TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models

Prior Labs Team[1]
[1]The list of contributors can be found in the appendix.

The first tabular foundation model, TabPFN, and its successor TabPFNv2 have impacted tabular AI substantially, with dozens of methods building on it and hundreds of applications across different use cases.
This report introduces TabPFN-2.5, the next generation of our tabular foundation model, scaling to 20× data cells compared to TabPFNv2. On industry standard benchmarks with up to 50,000 data points and 2,000 features, TabPFN-2.5 substantially outperforms tuned tree-based models and matches the accuracy of AutoGluon 1.4, a complex four-hour tuned ensemble that even includes the previous TabPFNv2.
For production use cases, we introduce a new distillation engine that converts TabPFN-2.5 into a compact MLP or tree ensemble, preserving most of its accuracy while delivering orders-of-magnitude lower latency and plug-and-play deployment.
This new release will immediately strengthen the performance of the many applications and methods already built on the TabPFN ecosystem.

Date: November 6, 2025
Website: https://priorlabs.ai/
Docs: https://docs.priorlabs.ai/overview
PyPI: `pip install tabpfn`
License: `TABPFN-2.5 License v1.0` (see Section 6 for details)
Contact: hello@priorlabs.ai

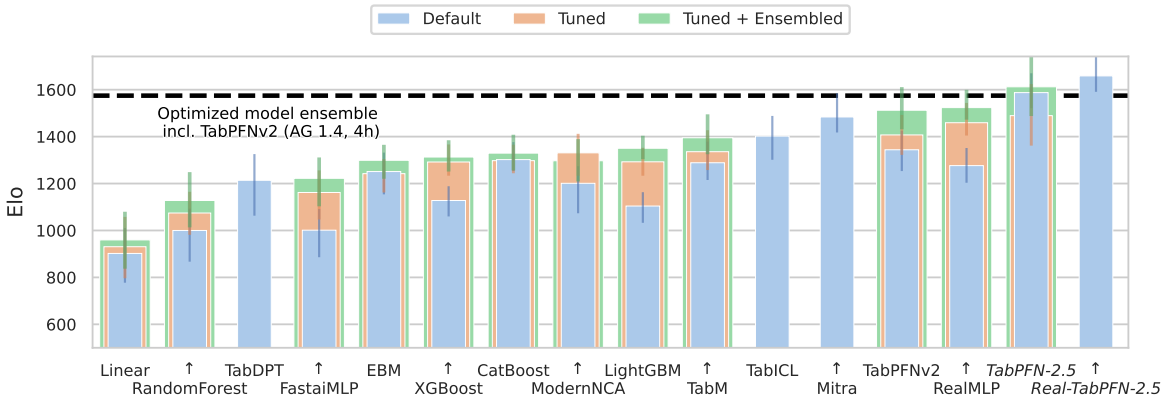


Figure 1: TabPFN-2.5 performance on the standard TabArena-lite benchmark, TabPFNv2 classification subset. TabPFN-2.5 outperforms any other model in a forward pass, and marks a strong leap from TabPFNv2. When fine-tuned on real data, Real-TabPFN-2.5 shows even stronger performance. The horizontal dotted line stands for AutoGluon 1.4 extreme mode tuned for 4 hours, an ensemble of models including TabPFNv2 [1].

1 Introduction

Tabular data is ubiquitous, forming the backbone of decision-making in countless domains, from finance to healthcare. For decades, traditional tabular machine learning—built on gradient-boosted trees [2–4], random forests [5], and linear or additive models—has been the workhorse of applied data science. Yet these methods remain limited: they require extensive dataset-specific tuning, often provide uncalibrated or unreliable uncertainty estimates without significant modification, and lack the generalization and transferability of modern foundation models.

Tabular foundation models (TFMs) offer a new paradigm. They address these limitations by pretraining on large synthetic distributions of tabular tasks and performing inference via in-context learning instead of gradient descent. They are training-free predictors meta-trained to yield strong calibration, without the need for time-consuming and labor-intensive hyperparameter tuning necessary for gradient-boosted trees. Their strong generalization makes them particularly attractive for data-scarce domains.

Our initial release, TabPFNv1 [6] served as a proof-of-concept that a transformer could learn a Bayesian-like inference algorithm, though it was limited to small (up to 1k samples), clean, numerical-only data. Our successor, TabPFNv2 [7], scaled this idea into a practical model for datasets up to 10,000 samples. TabPFNv2 handles the messy and heterogeneous data seen in the real world—including categorical features, missing values, and outliers.

This paper describes the next release of TabPFN: TabPFN-2.5. Our key contributions are:

- **SOTA Performance:** In a forward pass, TabPFN-2.5 outperforms tuned tree-based models (like XGBoost and CatBoost) and matches the accuracy of AutoGluon 1.4 tuned for 4 hours—a complex ensemble that includes all previous methods, even TabPFNv2.
- **Improved Scalability:** We scale the power of in-context learning to datasets of up to 50,000 samples (5x increase over TabPFNv2) and 2,000 features (4x increase), making TFMs viable for a much wider range of real-world problems [1].
- **Fast Inference:** We dramatically improve inference speed. We introduce TabPFN-as-MLP/TreeEns, a proprietary output engine, that yields an MLP or tree ensemble, combining most of TabPFN's accuracy with the low-latency inference and easy deployment of MLPs and tree ensembles.

We begin by surveying the growing ecosystem of TabPFN applications and extensions (Section 2). We then describe our methodological advances (Section 3) and present the experimental results (Section 4). We then discuss how to get the best speed out of TabPFN on common hardware (Section 5) as well as our non-commercial open-source license (Section 6). We conclude by discussing the remaining limitations and opportunities for future work (Sections 7). For installation and usage examples, see the online documentation at https://docs.priorlabs.ai/.

2 Ecosystem & Adoption

2.1 Community Adoption

Since its release, TabPFNv2 has become a widely used baseline for tabular ML. The *Nature* paper [7] has been cited in almost 400 papers within 10 months of its publication, and the open-source package has surpassed 2,000,000 downloads on PyPI[2]. Adoption spans both research and production, especially in settings with sparse data or frequent retraining requirements. This widespread adoption has matured TabPFN from a research model into a stable product. With feedback from our community of nearly 1,500 users on Discord, and hundreds of closed GitHub issues, we have shipped numerous stability fixes, and cross-platform device compatibility. In addition to commercial use-cases, we also collected 100 published use cases across a broad range of areas (please see Appendix B for a detailed list):

- **Healthcare and Life Sciences.** Adoption is strongest in healthcare (50+ published applications), driven by TabPFN's exceptional performance in data-scarce settings—a common challenge in medicine. Use cases span oncology, neurology, cardiology, and pharmacology, powering applications

[1] In exploratory runs, classification datasets up to ∼160k rows × 500 features and regression datasets up to ∼85k × 500 features fit into memory on an NVIDIA H100 (80 GB) using FP16 and FlashAttention-3. These configurations are outside our validated range and not included in reported benchmarks.

[2] Google Scholar entry (accessed Nov 6, 2025; Download stats for `tabpfn` on pepy.tech (accessed Nov 6, 2025): https://www.pepy.tech/project/tabpfn

like diagnosis, prognosis, and treatment response prediction from complex multimodal (clinical, imaging, omics) data.

- **Financial Services, Banking, and Insurance.** While we see strong commercial traction, public-facing use cases are rare due to the competitive, private nature of this industry (3 collected). Applications in this domain typically involve proprietary forecasting, uplift modeling, and risk assessments.
- **Energy and Utilities.** We've identified 14 published cases centered on complex forecasting and optimization. Key applications include environmental forecasting (algal blooms, wildfire risk), renewable-energy nowcasting, and process/asset optimization across water, oil & gas.
- **Manufacturing and Industrial.** The 13 diverse published use cases in this area highlight TabPFN's flexibility. Applications include anomaly detection in IIoT security, predictive maintenance for rotating machinery, physics-aware optimization for battery thermal modeling, and semiconductor test optimization.
- **Other Industries** Over 20 further applications demonstrate broad utility, spanning geoscience, agriculture, materials, and engineering. These range from microbiome classification and lunar regolith analysis to soil property modeling, fuel-blend optimization and crop yield forecasting.

2.2 A Foundational Layer for New Research

Beyond direct application, TabPFN now serves as a foundational layer for new research domains. Its ability to act as a powerful, pre-trained "algorithm-in-a-box" has unlocked new approaches to complex problems. We expect TabPFN-2.5 to directly boost performance in all these areas:

- **Time Series Forecasting**: TabPFN-TS [8] extends TabPFN to time-series forecasting by incorporating temporal context into its in-context learning mechanism, outperforming specialized time-series models without any retraining.
- **Node Classification in Graphs:** Various works [9, 10] represent graph nodes as tabular instances with relational and structural features, directly using tabular foundation models like TabPFN to solve the problem.
- **Data Streams:** TabPFNv2 was used for in-context learning on Evolving Data Streams [11]. TabPFN can *adapt to non-stationary data streams* online, without retraining, enabling continual learning in evolving environments.
- **Reinforcement Learning:** TabPFNv2 was used to replace gradient-based policy optimization with in-context optimization over trajectories, creating a powerful general-purpose optimizer for RL tasks [12].
- **Bayesian optimization:** GIT-BO [13] uses TabPFNv2 inside of high-dimensional Bayesian Optimization, as it enables efficient search in high-dimensional and heterogeneous design spaces.
- **Multimodal Learning & Encoding:** TabPFN is used to integrate tabular data with other modalities. It can serve as a *frozen tabular encoder* to generate robust embeddings for combination with data like images (e.g., in the TIME framework [14]), or handle modalities in a unified manner by adding modality-specific projectors [11].
- **Causal Inference:** Do-PFN [15], CausalPFN [16], and CausalFM [17] pre-train PFNs to predict interventional outcomes, and show strong performance in estimating causal effects.

2.3 The TabPFN-Extensions Ecosystem

We maintain the TabPFN–Extensions repository (`https://github.com/PriorLabs/tabpfn-extensions`), which offers extensions around the core model, developed together with a growing community around TabPFN. These extensions leverage TabPFN capabilities for:

- **Interpretability.** SHAP values, feature selection, partial dependence.
- **Unsupervised Tasks.** Data generation, augmentation, outlier detection.

- **Advanced Modeling.** Many-class classification, regression-via-classifier.
- **Performance & Integration.** Lightweight HPO, ensembling, and integration with tree/forest baselines.

Figure 11 in the appendix provides a minimal workflow to help users pick the right components for their task.

3 Model Overview

TabPFN-2.5 follows the same general design as TabPFNv2 but introduces deeper architectures, richer synthetic priors, and new calibration and inference modules. We summarize only the key changes here.

Data. We improved our prior data generation substantially, broadened the set of distributions and scaled up to more data points and more features, while keeping the prediction tasks difficult. Like the original TabPFNv2, TabPFN-2.5 is trained purely on synthetically generated data. We also release a version that is fine-tuned on real data following Real-TabPFN [18]. It is trained on a curated corpus of **43 real-world tabular datasets** sourced from OpenML and Kaggle, deduplicated against all internal benchmarks and the full TabArena suite. We refer to this version as Real-TabPFN-2.5, and report strong improvement in Figures 3 and 4. See Appendix C for details on training and deduplication.

Architecture. We follow the alternating-attention transformer design of TabPFNv2, which attends across both data points and features to achieve permutation invariance, but introduces some changes:

- We increase the network depth from 12 to 18 layers for our regression model and 24 layers for our classification model.
- We simultaneously increase the feature group size (the number of features being embedded together), which allows for faster training and inference. We use a group size of 3 for TabPFN-2.5, compared to 2 for TabPFNv2.
- For our regression models, we found small improvement in replacing the linear encoder used in TabPFNv2 by a 2-layer MLP.
- Finally, we add 64 additional "thinking" rows to the input dataset of TabPFN-2.5, which are learned during pretraining. Inspired by results from the LLM literature [19, 20], these rows give additional computational capacity to the model and can also act as attention sinks to help the model ignore other rows [21].

Other core components from TabPFNv2—feature/sample dual attention, caching separation of training/test context, and positional feature embeddings—remain unchanged.

Preprocessing. We aggregate predictions across multiple dataset permutations and feature transformations to enhance robustness and generalization. In the updated TabPFN-2.5 configuration, additional feature transformations are introduced to enhance robustness against outlier-prone feature distributions and to increase the diversity among the individual estimators. Specifically, we combine robust scaling and soft clipping (following [22]) with quantile transformations and standard scaling to balance stability and sensitivity across features. Following TabPFNv2, we also include singular value decomposition (SVD) components as additional features in some of the estimators, capturing high-energy directions of variance that provide complementary global structure information.

Hyperparameter Tuning of TabPFN with TabPFN. TabPFN's hyperparameter space spans architectural, training, and prior-data parameters, making exhaustive grid search computationally infeasible. To explore this space efficiently, we adopted a surrogate-based optimization strategy.

We first trained ≈ 100 models on a broad but sparse grid of hyperparameter configurations drawn from plausible prior ranges and evaluated them on a curated in-house validation suite, producing a compact set of hyperparameter–performance pairs.

With ~ 50 hyperparameters and only 100 datapoints, direct interpolation was prone to overfitting. We therefore used a regression model well-suited for data-scarce structured prediction—our previous

TabPFNv2 model—as a surrogate to predict validation performance over a denser grid of 10,000 configurations. This self-referential "TabPFN-tunes-TabPFN" strategy efficiently surfaced promising regions of the search space for full, compute-intensive training runs.

Tuning custom metrics. TabPFN-2.5 adds new post-processing capabilities that enhance both calibration and metric-specific optimization. Our framework now supports tuning the classifier's decision threshold, enabling direct optimization of metrics beyond accuracy—such as the F1-score—by adjusting the operating point to the desired trade-off between precision and recall. For multiclass classification, it allows to apply temperature scaling to the final softmax outputs to improve probability calibration. This threshold tuning procedure can yield substantial performance improvements (see Appendix H). Unless otherwise noted, however, all classification results in this report are computed using uncalibrated, default scores, without temperature scaling or threshold tuning.

Reducing inference costs. Through optimized preprocessing, adoption of FlashAttention-3 [23], and parallel evaluation across multiple GPUs, TabPFN-2.5 scales inference to datasets with up to 50,000 rows and 2,000 features.

Creating fast, deployable models. To improve deployment flexibility, we developed a proprietary distillation engine that, given a training data set, outputs a multi-layer perceptron (TabPFN-2.5-as-MLP) or tree ensemble classifier (TabPFN-2.5-as-TreeEns) whose performance is close to the one of TabPFN on this dataset (see Figure 7). In contrast to TabPFN, this resulting MLP or tree ensemble classifier is dataset-specific, does not perform in-context learning, takes as input a single data point, and has extremely low latency and memory footprint for making predictions. Because it outputs a standard MLP or tree ensemble, it can be seamlessly integrated into existing production pipelines, including those constrained by latency, interpretability, or regulatory requirements that hinder a change in the class of models being deployed. This increases TabPFN-2.5's practical use in real-world decision systems. Other types of models could easily be supported.

Model	Rows	Feat.	Type	Depth	Inference mode
TabPFN-v1	1,000	100	Num.	8	ICL
TabPFN-v2	10,000	500	Mixed	12	ICL
TabPFN-2.5	50,000	2000	Mixed	18–24	ICL+MLP/Trees

Table 1: **Summary of TabPFN model variants.** Max Rows and Features are the recommended maximum sizes. Models fit larger datasets but are not built and evaluated for these settings.

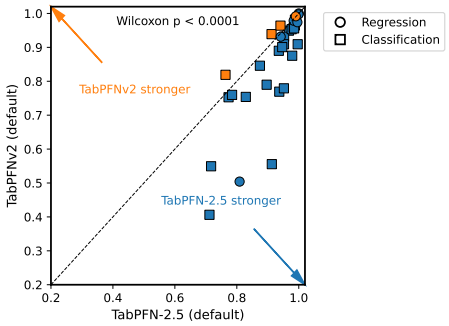


Figure 2: **TabPFN-2.5 clearly outperforms TabPFNv2.** We show normalized performance for each dataset of the TabPFNv2 subset of TabArena. TabPFN-2.5 often performs much better and is never much worse.

4 Experimental Results

We first demonstrate state-of-the-art performance on the industry standard benchmark TabArena and using our own benchmarking framework. Then, we report our advances to reduce inference latency. Finally, we demonstrate that TabPFN-2.5 yields new state-of-the-art performance for causal machine learning.

4.1 Performance on the Industry Standard Benchmark TabArena

TabArena [24] is the most curated tabular benchmark, based on the largest number of candidate datasets considered, and created by open-source contributors from a wide range of institutions. It will appear at the NeurIPS 2025 Datasets & Benchmarks track and is thus most up-to-date. We follow the paper's recommendation to benchmark on "TabArena-Lite", which is a cheaper but representative version of the full benchmark using only one test fold. The benchmark contains a set of 51 datasets selected from 1053 to be representative of real-world tabular data. See Erickson et al. [24] for the list of datasets.

Pushing the limit on medium-sized datasets. Figure 3 shows results for TabPFN-2.5 on TabArena-Lite with up to 10,000 data points and 500 features, demonstrating that TabPFN-2.5, in a forward pass, outperforms the wide range of existing tabular prediction methods. On classification, TabPFN-2.5 in a forward pass outperforms AutoGluon 1.4, an ensemble tuned for four hours and including best other methods (even TabPFNv2). Using our Real-TabPFN-2.5 variant fine-tuned on real datasets (deduplicated from TabArena datasets) widens the lead even further. On the other hand, our regression model benefits much more from tuning and outperforms AutoGluon 1.4 after being tuned for 60 configurations.

Scaling to larger datasets. Figure 4 shows a similar experiment with up to 50,000 data points and 2,000 features, clearly ranking TabPFN-2.5 as the best default model, and outperforming (for regression datasets) or approaching (for classification datasets) AutoGluon 1.4 (tuned for 4 hours) when tuned. Again, we highlight the very strong default performance of Real-TabPFN-2.5 on these larger classification datasets.

A significant improvement upon TabPFNv2. Comparing the default performance of TabPFN-2.5 and TabPFNv2, we see a big leap in performance in Figure 3. In addition, looking at performance on each dataset in TabArena (TabPFNv2 compatible subset) in Figure 2, we see that TabPFN-2.5 clearly outperforms TabPFNv2 on almost all datasets, and is never much worse. In Appendix G, we detail the results on TabArena-Lite, comparing TabPFN-2.5 to other foundation models like TabICL [25] or LimiX [26].

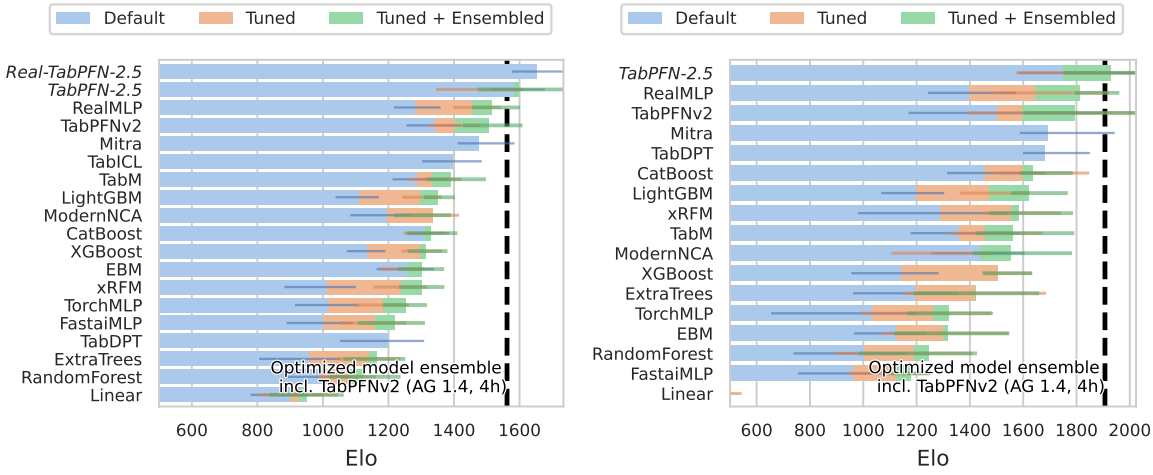


Figure 3: TabArena-Lite results on **classification** (left) and **regression** (right), restricted to datasets with less than **10K training samples and 500 features**. Note that tuning for TabPFN-2.5 is only based on 60 random configs compared to 200 for the baselines. The vertical dotted line stands for AutoGluon 1.4 extreme mode tuned for 4 hours, an ensemble of models including TabPFNv2 [1].

4.2 Performance on Internal Benchmarks

A diverse internal benchmark. In addition to the public TabArena benchmark, we built our own benchmarking framework using proprietary data. It includes over 100 use cases from healthcare, finance, insurance, retail and manufacturing. This benchmark focuses on comparing to gradient-boosted decision

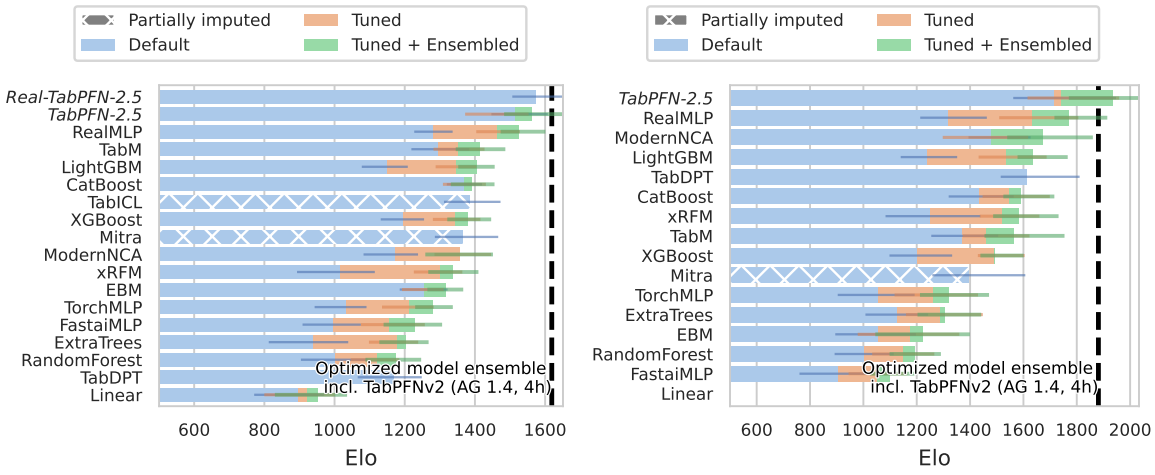


Figure 4: TabArena-Lite results on **classification** (left) and **regression** (right), restricted to datasets with less than **50,000 training samples and 2,000 features**. Note that tuning for TabPFN-2.5 is only based on 60 random configs compared to 200 for the baselines. The vertical dotted line stands for AutoGluon 1.4 extreme mode tuned for 4 hours, an ensemble of models including TabPFNv2 [1].

tree libraries that are frequently used in industry (XGBoost [2], CatBoost [3], LightGBM [4]), both in their default version and tuned for one hour. In all cases, we show the results of three standard gradient-boosted tree libraries (LightGBM, XGBoost and CatBoost). We tune all of the baselines for 1hr, using random search on the established search spaces from [7]. TabPFN is tuned using our AutoTabPFN system, resulting in a tuned and ensembled model.

TabPFN-2.5 shows strong results up to 50,000 samples and 2,000 features. Figure 5 and Figure 6 show results on our internal benchmark for classification and regression datasets with up to 50k data points and 500 features. We can see on these figures that TabPFN outperforms in one forward pass all our tuned baselines. In Section F, we also show strong results on datasets with 500 to 2,000 features, and provide more details on how we normalize the performance of each model across datasets.

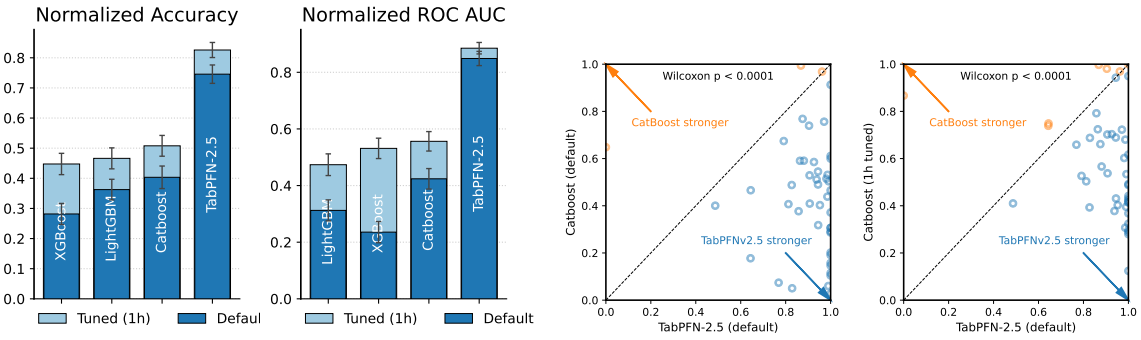


Figure 5: Results from our internal benchmark on **classification datasets with up to 50k data points**. More details on the normalization is available in Appendix F. In the scatter plots (right), each point represents a different dataset from our internal benchmark, and the axes measure the normalized performance of TabPFN-2.5 and CatBoost (either default or tuned for 1 hour) on this dataset.

4.3 Measuring TabPFN-2.5 Training and Inference Speed

Figure 8 shows how TabPFN-2.5 classification speed scales with training set size, when using one or four GPUs, as we vary the number of rows and columns in the dataset. The time measured includes both the time to process the training rows (equivalent to the combination of "training" a classical ML model) and "prediction" time on test rows. We can observe the expected scaling in $\mathcal{O}(r^2 \min(c, 500) + r \min(c, 500)^2)$,

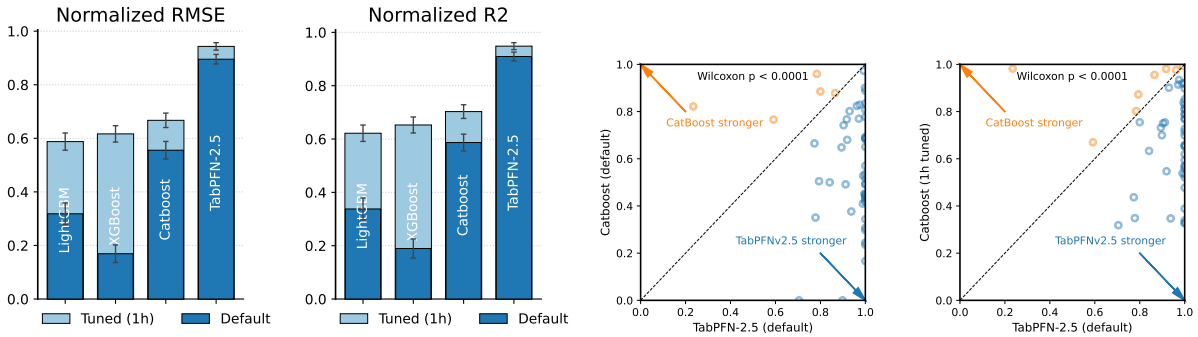


Figure 6: Results from our internal benchmark on **regression datasets with up to 50k data points**. More details on the normalization is available in Appendix F. In the scatter plots (right), each point represent a different dataset from our internal benchmark, and the axis measure the normalized performance of TabPFN-2.5 and CatBoost (either default or tuned for 1 hour) on this dataset

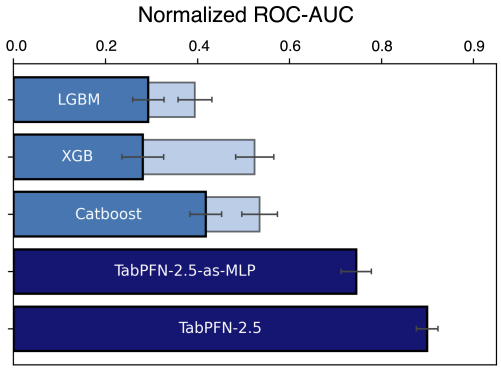
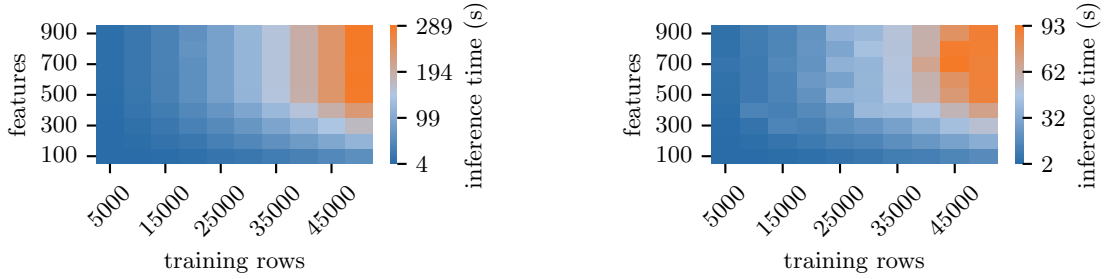


Figure 7: **TabPFN-as-MLP still outperforms tree-based models.** For baseline, light blue represents performance when tuned for 1 hour, and darker blue default performance. For TabPFN, we report default performance.

where r is the number of rows and c is the number of columns, due to dual attention over rows and capped per-estimator feature subsampling at 500 features. Section 5 contains results for regression, and performance on common models of GPU, for reference. The inference speed reported here reflects the latency of the full in-context learning model.

4.4 Fast Inference with TabPFN-2.5-as-MLP

We benchmark TabPFN-2.5-as-MLP against tuned LightGBM, XGBoost, and CatBoost models , as well as the standard TabPFN-2.5 model, on our curated collection of internal open source datasets with less than 10k data points. Figure 7 illustrates representative test-split performance. Empirically, TabPFN-2.5-as-MLP offers competitive accuracy while reducing inference cost, making it attractive for high-throughput or resource-constrained deployment scenarios.



(a) one H100 GPU

(b) four H100 GPUs

Figure 8: Time taken, in seconds, to fit TabPFN-2.5 classification models on various training set sizes, and then make predictions on 500 test rows. Figure 16 in Section 5 reports results for regression, alongside performance on A100 and T4 GPUs.

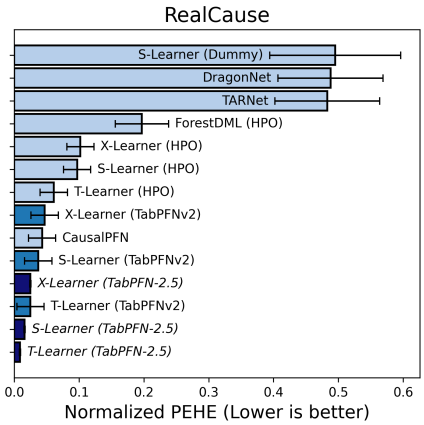


Figure 9: PFN-based CATE estimators dominate RealCause, outperforming specialized tree- and deep-learning-based methods for causal inference. Choice of propensity and outcome model is important for CATE estimation.

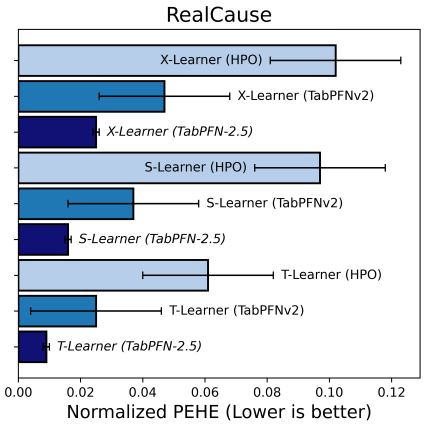


Figure 10: Improvements in base model predictive performance transfer to improved performance in CATE estimation. Our new model, TabPFN-2.5, is the strongest choice of base model for all meta-learners.

4.5 TabPFN for Causal Inference

RealCause Benchmark. To systematically evaluate TabPFN's potential as a causal estimator, we leverage the RealCause benchmark [27], a semi-synthetic benchmark which begins with real-world randomized control trial (RCT) data and synthetically creates observable confounding effects.[3] We measure the Precision in Estimating Heterogeneous Effects (PEHE), which corresponds to the root-mean-squared error between predicted and RealCause's ground-truth CATE values[4]. In Figure 9, we show that PFN-based methods for CATE-estimation dominate the leaderboard, occupying the first seven positions. TabPFN-2.5 applied as a T-Learner, a simple two-model approach that fits a separate model to the treatment and control observations, achieves the strongest overall performance, outperforming specialized tree- and deep-learning-based methods [28]. We also observe in Figure 10 that for each of our three meta-learners, TabPFN-2.5 performs better out-of-the-box than TabPFNv2 and HPO[5]. This result shows that improvements in base model predictive performance transfer to the problem of causal inference.

Foundation Models for Causal Inference. While we show strong results in unconfounded settings, real-world causal inference often involves imperfect data and latent confounders. A growing line of work aims to pre-train PFNs explicitly for causal reasoning—for example, predicting interventional outcomes or learning causal structures directly [15–17, 30, 31]. We view this as one of the most exciting frontiers for foundation models: extending TabPFN's reasoning from predicting *what is* to inferring *what would happen if*, and ultimately, *understanding why*.

5 How to Get Optimal Fit + Predict Speed from TabPFN-2.5

To achieve good performance, we recommend the following:

- **Use a dedicated GPU or GPUs**: We recommend NVIDIA H100 or A100 GPUs. Any dedicated GPU supported by PyTorch is compatible, but some models may not have enough memory for larger datasets or perform slowly. Integrated GPUs, MPS (Apple Silicon), and CPUs are also supported, but are only suitable for small datasets.
- **Use multiple GPUs**: For larger datasets, fit + predict time can be dramatically reduced by parallelizing inference over several GPUs. To enable this, set the `device` parameter of `TabPFNClassifier` and `TabPFNRegressor`.

[3] Descriptions of the ACIC-2016, IHDP, and Lalonde-PSID and Lalonde-CPS datasets are provided in Appendix Table 3.
[4] For a description of the CATE estimation task and common estimators, please refer to Appendix D.
[5] Hyperparameter optimization is run for 60 seconds on an H100 per propensity and outcome model using FLAML [29].

- **Use batch inference**: Unless the fitted-model cache is enabled (see below), the model is retrained each time `.predict()` is called. This means that it is much faster to make a prediction for all your test points in a single `.predict()` call. If you run out of memory, split the test points into batches of 1000 to 10000 and call `.predict()` for each batch.
- **Use PyTorch 2.8 or above**: TabPFN-2.5 also supports earlier versions of PyTorch, but these may have lower performance.
- **For small datasets, enable the fitted-model cache**: This is an experimental feature that trains and stores the model during `.fit()`, making subsequent `.predict()` calls fast by using a KV-Cache. It is enabled by setting the `fit_mode` parameter of `TabPFNClassifier` and `TabPFNRegressor` to `fit_with_cache`. However, with this setting classification models will consume approximately 6.1 KB of GPU memory and 48.8 KB of CPU memory per cell in the training dataset (regression models about 25% less), thus it is currently only suitable for small training datasets. For larger datasets and CPU-based inference, we recommend the TabPFN-as-MLP/Tree output engine.
- If speed is important for your application, you may consider optimizing the `memory_saving_mode` and `n_preprocessing_jobs` parameters of `TabPFNClassifier` and `TabPFNRegressor`. See the code documentation for further information.

Figure 16 in the appendix shows the inference latency you can expect for three common models of GPU, when using one or four GPUs. It also shows the maximum dataset size that fits in memory for each GPU.

6 License and Availability

We release TabPFN-2.5 under our `TABPFN-2.5 License v1.0` designed to be permissive for research and internal evaluation. It *explicitly allows* testing, evaluation, and internal benchmarking, so an organization can download the model and run preliminary assessments on its own datasets.

The key restriction is that the model, its derivatives, and its outputs cannot be used for any commercial or production purpose. This includes, but is not limited to, revenue-generating products, competitive benchmarking for procurement, client deliverables, or using the model's results for internal commercial decision-making.

For all production use cases, we offer a *Commercial Enterprise License*. This provides access to our proprietary high-speed inference engine, dedicated support, integration tooling, and other internal models.

Please contact us at sales@priorlabs.ai for commercial licensing inquiries. The full non-commercial mode license text can be found at https://huggingface.co/Prior-Labs/tabpfn_2_5/blob/main/LICENSE.

7 Conclusion and The Road Ahead

We are excited about this release. Taken together, our experiments on public (TabArena) and private benchmarks demonstrate that TabPFN-2.5 sets a new state-of-the-art for tuning-free tabular models. In a single forward pass, it matches the performance of complex 4-hour-tuned ensembles - ensembles that even include our previous TabPFNv2 - for datasets up to 50,000 data points and 2,000 features. This advantage holds for classification, regression, and sophisticated downstream tasks like causal inference.

While we have pushed the boundary to 50,000 samples, the next step is scaling to datasets with millions of rows. We are actively developing new techniques—including retrieval, fine-tuning, and novel architectures—and anticipate that systems based on Tabular Foundation Models (TFMs) will define state-of-the-art performance for datasets with millions of data points within the next year.

Our broader vision beyond this release is to tackle the entire stack of problems with tabular-like data, including time series, multimodal tabular data, causal inference, unsupervised tasks, integration of domain knowledge and decision support, ultimately building the core intelligence engine for reasoning over structured and multimodal data.

References

[1] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[3] 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.

[4] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf.

[5] Random forests. 45(1):5–32, 2001. URL http://dx.doi.org/10.1023/A%3A1010933404324.

[6] 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.

[7] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

[8] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. The tabular foundation model tabpfn outperforms specialized time series forecasting models based on simple features. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

[9] Adrian Hayler, Xingyue Huang, İsmail İlkan Ceylan, Michael Bronstein, and Ben Finkelshtein. Bringing graphs to the table: Zero-shot node classification via tabular foundation models. *arXiv preprint arXiv:2509.07143*, 2025. doi: 10.48550/arXiv.2509.07143. URL https://arxiv.org/abs/2509.07143.

[10] Dmitry Eremeev, Gleb Bazhenov, Oleg Platonov, Artem Babenko, and Liudmila Prokhorenkova. Turning tabular foundation models into graph foundation models, 2025. URL https://arxiv.org/abs/2508.20906.

[11] Afonso Lourenço, João Gama, Eric P. Xing, and Goreti Marreiros. In-context learning of evolving data streams with tabular foundational models. *arXiv preprint arXiv:2502.16840*, 2025. doi: 10.48550/arXiv.2502.16840. URL https://arxiv.org/abs/2502.16840.

[12] David Schiff, Ofir Lindenbaum, and Yonathan Efroni. Gradient free deep reinforcement learning with tabpfn. *arXiv preprint arXiv:2509.11259*, 2025. doi: 10.48550/arXiv.2509.11259. URL https://arxiv.org/abs/2509.11259.

[13] Rosen Ting-Ying Yu, Cyril Picard, and Faez Ahmed. Git-bo: High-dimensional bayesian optimization with tabular foundation models. *arXiv preprint arXiv:2505.20685*, 2025. doi: 10.48550/arXiv.2505.20685. URL https://arxiv.org/abs/2505.20685.

[14] Jiaqi Luo, Yuan Yuan, and Shixin Xu. Time: Tabpfn-integrated multimodal engine for robust tabular-image learning, 2025. URL https://arxiv.org/abs/2506.00813.

[15] Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation. *arXiv preprint arXiv:2506.06039*, 2025.

[16] Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Benson Li, Junwei Ma, Jesse C. Cresswell, and Rahul G. Krishnan. Causalpfn: Amortized causal effect estimation via in-context learning, 2025. URL https://arxiv.org/abs/2506.07918.

[17] Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation models for causal inference via prior-data fitted networks, 2025. URL https://arxiv.org/abs/2506.10914.

[18] 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.

[19] William Merrill and Ashish Sabharwal. Exact expressive power of transformers with padding. *CoRR*, abs/2505.18948, 2025. URL https://arxiv.org/abs/2505.18948. arXiv pre-print.

[20] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations (ICLR) 2024 Poster*, 2024. URL https://openreview.net/forum?id=ph04CRkPdC. Poster paper; published 16 Jan 2024, last modified 17 Mar 2024.

[21] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR) 2024*, 2024. URL https://arxiv.org/abs/2309.16588. arXiv preprint arXiv:2309.16588v2, submitted 28 Sep 2023, revised 12 Apr 2024.

[22] David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/2ee1c87245956e3eaa71aaba5f5753eb-Abstract-Conference.html.

[23] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/7ede97c3e082c6df10a8d6103a2eebd2-Abstract-Conference.html.

[24] Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, Frank Hutter, et al. Tabarena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*, 2025.

[25] 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.

[26] Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, Ningbo Dai, Renzhe Xu, Shuyang Li, Tianyang Zhang, Yue He, Yuanrui Wang, Yunjia Zhang, Zijing Xu, Dongzhe Li, Fang Gao, Hao Zou, Jiandong Liu, Jiashuo Liu, Jiawei Xu, Kaijie Cheng, Kehan Li, Linjun Zhou, Qing Li, Shaohua Fan, Xiaoyu Lin, Xinyan Han, Xuanyue Li, Yan Lu, Yuan Xue, Yuanyuan Jiang, Zimu Wang, Zhenlei Wang, and Peng Cui. Limix: Finetuning tabular foundation models via limited mixture adaptation. *arXiv preprint arXiv:2509.03505*, 2025.

[27] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. Realcause: Realistic causal inference benchmarking. *CoRR*, abs/2011.15007, 2020. URL https://arxiv.org/abs/2011.15007.

[28] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests, 2017. URL https://arxiv.org/abs/1510.04342.

[29] Chi Wang and Qingyun Wu. FLO: fast and lightweight hyperparameter optimization for automl. *CoRR*, abs/1911.04706, 2019. URL http://arxiv.org/abs/1911.04706.

[30] Anish Dhir, Cristiana Diaconu, Valentinian Mihai Lungu, James Requeima, Richard E. Turner, and Mark van der Wilk. Estimating interventional distributions with uncertain causal graphs through meta-learning, 2025. URL https://arxiv.org/abs/2507.05526.

[31] Andreas Sauter, Saber Salehkaleybar, Aske Plaat, and Erman Acar. Activa: Amortized causal effect estimation via transformer-based variational autoencoder, 2025. URL https://arxiv.org/abs/2503.01290.

[32] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[33] Rickard Karlsson and Jesse Krijthe. Detecting hidden confounding in observational data using multiple environments. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 44280–44309. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/89e541b817ea043a971840a926e12b37-Paper-Conference.pdf.

[34] Alicia Curth, David Svensson, Jim Weatherall, and Mihaela Van Der Schaar. Really doing great at estimating cate? a critical look at ml benchmarking practices in treatment effect estimation. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*, 2021.

[35] Miruna Oprescu, Vasilis Syrgkanis, Keith Battocchi, Maggie Hei, and Greg Lewis. Econml: A machine learning library for estimating heterogeneous treatment effects. In *33rd Conference on Neural Information Processing Systems*, page 6, 2019.

[36] Toon Vanderschueren, Tim Verdonck, Mihaela van der Schaar, and Wouter Verbeke. AutoCATE: End-to-end, automated treatment effect estimation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QbOoz74GNO.

[37] Qiong Zhang, Yan Shuo Tan, Qinglong Tian, and Pengfei Li. Tabpfn: One model to rule them all? *arXiv preprint arXiv:2505.20003*, 2025.

[38] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, 2019.

A Contributors

Model dev & Deployment

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Brendan Roof, Phil Jund, Benjamin Jäger, Adrian Hayler, Dominik Safaric, Simone Alessi, Felix Jablonski, Mihir Manium, Rosen Yu, Anurag Garg, Jake Robertson, Shi Bin (Liam) Hoo, Vladyslav Moroshan, Magnus Bühler, Lennart Purucker, Noah Hollmann, Frank Hutter

Distribution & Product

Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Sauraj Gambhir

B TabPFN Use Case Overview

TabPFNv2 has been applied to a broad set of use cases. We now list 100 published use cases across different industries.

Healthcare and Life Sciences

We collected 50 published TabPFN use cases in this area, by far more than in any other area; we attribute this partly to the scarcity of data in healthcare and life sciences, and partly to the open publishing culture in this area. Use cases span oncology, neurology, cardiology, psychiatry, nephrology, and pharmacology. Applications include diagnosis, prognosis, and treatment response prediction from multimodal clinical, imaging, and omics data, often under severe data scarcity.

1. TabPFN was applied to distinguish cancer patients from healthy individuals using immune system profiles from peripheral blood, facilitating predictions of immunotherapy responses. Link
2. A machine learning model employing TabPFN was developed for non-invasive diagnostic prediction of minimal change disease in patients with nephrotic syndrome, utilizing clinical biomarkers. Link
3. TabPFN was integrated into a system for analyzing T-cell receptor repertoires combined with clinical biomarkers to forecast immunotherapy outcomes in cancer patients, as explored by researchers at BostonGene. Link
4. TabPFN enabled early detection of stillbirth risks through analysis of cardiotocography data, supporting improved prenatal care. Link
5. Predictive modeling for postoperative outcomes following anterior cervical corpectomy utilized TabPFN to assess patient demographics and surgical parameters. Link
6. A hybrid model incorporating TabPFN was introduced to predict dementia progression in Parkinson's disease patients, handling small datasets and missing values effectively. Link
7. A machine learning model based on TabPFN was developed to predict 90-day unfavorable outcomes in stroke patients with distal vessel occlusions using CT perfusion imaging. Link
8. TabPFN was utilized in chemoproteomics for identifying small-molecule fragment-protein interactions, aiding ligand discovery in drug development. Link
9. TabPFN facilitated the prediction of non-invasive ventilation outcomes in patients with acute hypoxemic respiratory failure, supporting early identification of treatment failures. Link
10. An interpretable Transformer-based model leveraging TabPFN was created to predict intravenous immunoglobulin resistance in pediatric patients with Kawasaki disease. Link
11. TabPFN was used to combine clinical, MR morphological, and delta-radiomics features to predict lymphovascular invasion in invasive breast cancer patients. Link
12. TabPFN is proposed to predict mental health trajectories through digital phenotyping, enabling proactive and personalized interventions in precision psychiatry. Link

13. TabPFN contributed to cardiovascular disease risk stratification using clinical features from a large patient cohort, incorporating interpretability techniques. Link

14. TabPFN outperformed traditional machine learning models for early prediction of acute kidney injury in hospitalized patients, demonstrating generalizability across datasets. Link

15. TabPFN was integrated into a framework for predicting postoperative mobility and discharge destinations in older adults using sensor data. Link

16. TabPFN supported the prediction of infant temperament from maternal mental health data, aiding early identification of at-risk infants. Link

17. TabPFN was employed to characterize clinical risk profiles for complications in type 2 diabetes mellitus patients, focusing on neuropathy and retinopathy. Link

18. TabPFN was extended with a longitudinal-to-cross-sectional transformation to forecast Alzheimer's disease progression on neuroimaging datasets. Link

19. TabPFN supported uncertainty calibration evaluation in medical data using variational techniques. Link

20. TabPFN was applied to predict tumor response to chemotherapy in cholangiocarcinoma patients using RNA expression landscapes. Link

21. TabPFN was incorporated into a generative model framework for tasks like data augmentation and imputation in biomedicine. Link

22. TabPFN facilitated the prediction of gallstone malignancy risks through analysis of associated disease factors. Link

23. TabPFN was used in classifying tuberculosis treatment outcomes based on clinical and sociodemographic data from national registries. Link

24. TabPFN contributed to early prediction of gestational diabetes using cell-free DNA and genetic scores from early pregnancy blood samples. Link

25. TabPFN was used for predicting schizophrenia based on sense of agency features, emphasizing interpretability. Link

26. TabPFN was integrated into a physiologically-based pharmacokinetic model for predicting dissolution and absorption of amorphous solid dispersions in drug development. Link

27. TabPFN enabled classification of respiratory diseases from sound data, addressing clinical spectrum diversity. Link

28. TabPFN was applied to small-data tabular learning in drug discovery, handling data scarcity and distribution shifts. Link

29. TabPFN facilitated prediction of coronary heart disease risk in patients with cardiovascular-kidney-metabolic syndrome, optimizing evaluation in small samples. Link

30. TabPFN was used to predict success of allogeneic stem cell mobilization in donors, aiding transplant therapies. Link

31. TabPFN contributed to predicting manual strength using anthropometric data, focusing on accuracy and interpretability. Link

32. TabPFN supported uncertainty-guided model selection for biomolecule efficacy prediction, enhancing ensemble optimization in drug discovery, as studied at GSK. Link

33. TabPFN was utilized in a multitask deep learning framework for optimizing in vitro fertilization decisions, including embryo transfer and pregnancy prediction. Link

34. TabPFN enabled a framework for early Long COVID detection through causal gene identification and interpretability. Link

35. TabPFN was used in a foundation model approach for neoadjuvant therapy recommendations in breast cancer, integrating multi-omics data. Link

36. TabPFN facilitated prediction of recurrence and progression in oral potentially malignant disorder patients post-surgery. Link

37. TabPFN supported prediction of occult lymph node metastasis in non-small cell lung cancer patients treated with stereotactic ablative radiotherapy. Link

38. TabPFN was used in stroke diagnosis, addressing dataset imbalance and model interpretability for clinical decisions. Link

39. TabPFN was integrated into a multimodal thesis framework for clinical predictions using tabular and phenotypic data from large-scale projects. Link

40. TabPFN was used to predict diabetes-related hypo- and hyperglycemia during hemodialysis using continuous glucose monitoring data, facilitating improved patient management. Link

41. TabPFN was applied to CorvisST biomechanical indices to classify corneal disorders, improving diagnostic accuracy in ophthalmology. Link

42. TabPFN was incorporated into a non-invasive sleep staging framework using respiratory sound features, advancing passive sleep monitoring. Link

43. TabPFN supported prediction of vancomycin blood concentrations to optimize antimicrobial dosing strategies in clinical practice. Link

44. TabPFN was used to predict negative self-rated oral health in adults, identifying risk factors for targeted public-health interventions. Link

45. TabPFN was extended to many features to enable robust analysis of high-dimensional biomedical data, improving model stability and interpretability in clinical applications.

46. TabPFN supported multi-omics fusion for neoadjuvant therapy recommendation in breast cancer, improving personalized treatment strategies. Link

47. TabPFN supported uncertainty-guided model selection for siRNA efficacy prediction, advancing molecular screening and drug discovery workflows. Link

48. TabPFN was used to classify respiratory diseases from sound recordings, contributing to non-invasive respiratory diagnostics. Link

49. TabPFN enhanced small-data learning in drug discovery, improving predictive performance under severe data scarcity. Link

50. TabPFN predicted gastrointestinal bleeding risk in pediatric Henoch–Schönlein purpura patients, supporting early clinical intervention. Link

Financial Services, Banking, and Insurance

While we have seen strong customer interest in this area, this is not reflected by the relatively few published use cases (only 3) we managed to collect; we attribute this to the domain's competitive nature and disinclination to publish.

1. TabPFN was applied to usage-based premium calculations in actuarial science, leveraging driving behavior data from IoT devices. Link

2. TabPFN facilitated cross-selling of health insurance products through deep learning analysis of customer data. Link

3. TabPFN was used in corporate bond recovery rate prediction for credit risk management. Link

Energy and Utilities

We collected 14 use cases focused on environmental forecasting (algal blooms, wildfire, rainfall), renewable-energy nowcasting, process/asset optimization across water, oil & gas, and materials.

1. TabPFN was employed to predict river algal blooms through multi-classification of chlorophyll-a concentrations, aiding water management. Link
2. TabPFN facilitated wildfire propagation prediction in Canadian conifer forests, classifying fire types for environmental risk assessment. Link
3. TabPFN was integrated into a machine learning framework for optimizing energy consumption at wastewater treatment plants. Link
4. TabPFN supported rainfall forecast post-processing using historical error patterns from environmental data. Link
5. TabPFN enabled solar forecast error adjustment, particularly during rapid weather changes, as developed by Open Climate Fix. Link
6. TabPFN was applied to predict ash fusibility in high-alkali coal for improved energy production. Link
7. TabPFN contributed to predicting Henry coefficients for alkanes in zeolites, aiding hydroisomerization in sustainable fuel production. Link
8. TabPFN facilitated shape-selectivity modeling in zeolites for long-chain alkane hydroisomerization, optimizing catalyst design. Link
9. TabPFN was used in an integrated framework for estimated ultimate recovery prediction and fracturing optimization in shale gas reservoirs. Link
10. TabPFN supported core data augmentation for enhanced reservoir parameter prediction in oil and gas exploration. Link
11. TabPFN was employed to optimize energy performance in multistage centrifugal pumps through entropy generation analysis. Link
12. TabPFN contributed to physics-informed regression for evaluating solar-reflective materials in facade temperature modeling. Link
13. TabPFN was applied to generate advanced global heat flow maps at 0.2° resolution, integrating high-resolution geophysical data to improve geothermal resource modeling. Link
14. TabPFN contributed to FuelCast, standardizing benchmarks for ship fuel consumption prediction and improving efficiency in maritime operations. Link

Manufacturing and Industrial

We collected 13 diverse use cases including anomaly detection, predictive maintenance, physics-aware optimization—spanning IIoT security, rotating machinery, semiconductor testing, geotechnical/optical sensing, machining, battery thermal modeling, and concrete mix design.

1. TabPFN enabled early fault classification in rotating machinery, addressing data scarcity in industrial scenarios. Link
2. TabPFN facilitated microcontroller performance prediction, aiding semiconductor screening with minimal supervision, as studied at Infineon Technologies. Link
3. TabPFN was applied to caisson inclination prediction in ultra-deep construction, combining data denoising techniques. Link
4. TabPFN supported event classification in phase-sensitive optical time-domain reflectometry systems for distributed fiber sensing. Link

5. TabPFN was integrated into an adaptive ensemble for intrusion detection in Industrial Internet of Things networks. Link
6. TabPFN enabled a random forest-based framework for attack recognition in Internet of Things networks, improving interpretability. Link
7. TabPFN facilitated geotechnical site characterization for predicting soil strength and imputing mechanical parameters. Link
8. TabPFN was used in cryogenic-assisted abrasive waterjet machining for improving surface integrity in titanium alloys. Link
9. TabPFN supported in-context learning for thermal behavior prediction in nano-phase change materials for battery systems. Link
10. TabPFN was applied to explainable strength evaluation in multicomponent concrete mixtures. Link
11. TabPFN was integrated into a multimodal fusion framework linking microstructure to friction behavior in martensitic stainless steel, improving wear resistance in materials engineering applications. Link
12. TabPFN supported multiscale modeling to predict soil salinity in arid farmland, advancing sustainable agricultural management in regions such as Xinjiang. Link
13. TabPFN was used in explainable modeling of multicomponent concrete strength, identifying key material factors and informing construction practices. Link

Other Industries

We collected 20 further heterogeneous TabPFN applications spanning geoscience, agriculture, materials, and engineering domains—ranging from microbiome classification and lunar regolith analysis to soil property modeling, crop yield and phenology forecasting, fuel-blend optimization, and spatial regression.

1. TabPFN was modified for microbiome data classification in metagenomics, matching species abundance patterns with synthetic priors. Link
2. TabPFN enabled lunar regolith analysis for classifying meteorite compositions from spectral data. Link
3. TabPFN facilitated winter wheat yield forecasting in agricultural regions by integrating climate and remote sensing data. Link
4. TabPFN was applied to flood impact assessment on housing prices by geographic areas. Link
5. TabPFN showed the strongest performance on 31 predictive soil modeling datasets containing 30 to 460 samples. Link
6. TabPFN was applied to shallow natural gas hazard prediction in tunnel construction. Link
7. TabPFN supported automated feature engineering for energy consumption forecasting in domain-specific applications. Link
8. TabPFN enabled Australian rice phenology prediction using remote sensing and weather data for crop management. Link
9. TabPFN was applied to a multi-stage framework for predicting fuel blend properties through automated feature engineering. Link
10. TabPFN enabled kriging prior regression for incorporating spatial context in soil mapping predictions. Link
11. TabPFN was applied to predicting electric vehicle crash severity using deep learning models. Link
12. TabPFN enhanced clone-type recognition across programming languages through metrics-driven analysis, improving stability and interpretability in software engineering. Link

13. TabPFN was used to predict biomass-derived hard carbon performance in sodium-ion batteries, facilitating material selection for energy storage systems. Link

14. TabPFN informed the development of TabImpute, enabling efficient zero-shot imputation for missing tabular data and improving preprocessing pipelines. Link

15. TabPFN supported a target-specific framework for predicting fuel blend properties, optimizing formulation strategies via automated feature engineering. Link

16. TabPFN, alongside TabICL and related foundation models, was evaluated for intrusion detection, improving cybersecurity performance in IoT networks. Link

17. TabPFN supported continual learning for tabular data streams in resource-constrained environments. Link

18. TabPFN was adapted for high-dimensional data through continued pre-training, enhancing robustness in noisy environments. Link

19. TabPFN contributed to assessing robustness of language models for data fitting under irrelevant variations. Link

20. TabPFN enabled fast zero-shot imputation for missing data across diverse domains. Link

C Data Contamination and Deduplication for Real-TabPFN-2.5

To ensure fair evaluation and eliminate data contamination, we implemented an enhanced multi-tiered deduplication and filtering pipeline for Real-TabPFN-2.5. While based on the methodology used for Real-TabPFN [18], the process was extended to deduplicate the training datasets against all internal benchmarks, our curated in-house validation suite, and the public TabArena benchmark [24]. Our deduplication procedure combines automated cross-referencing of dataset identifiers, feature schemas, and row- and column-level hashes with manual metadata inspection to ensure that no training dataset overlaps with, or is derived from, any evaluation dataset. Datasets failing these criteria were excluded from the final training corpus.

C.1 Training Datasets

The following table lists the datasets curated for fine-tuning, along with their sources and access links.

Name	Source
artificial-characters	OpenML
BNG(breast-w)	OpenML
BNG(tic-tac-toe)	OpenML
connect_4	OpenML
eeg-eye-state	OpenML
Employee-Turnover-at-TECHCO	OpenML
eye_movements	OpenML
FOREX_eurpln-hour-High	OpenML
gas-drift	OpenML
higgs	OpenML
Intersectional-Bias-Assessment-(Training-Data)	OpenML
law-school-admission-binary	OpenML
Medical-Appointment	OpenML
microaggregation2	OpenML
fried	OpenML
mushroom	OpenML
NewspaperChurn	OpenML
nursery	OpenML
WBCAtt	OpenML
Internet Firewall Data	OpenML

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
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-comma-sep	Kaggle
ip-network-traffic-flows-labeled-with-87-apps	Kaggle
Janatahack cross-sell prediction	Kaggle
L&T Vehicle Loan Default Prediction	Kaggle
League of Legends Diamond Games (First 15 Minutes)	Kaggle
Richter's Predictor Modeling Earthquake Damage	Kaggle
Server Logs - Suspicious	Kaggle
Sloan Digital Sky Survey DR14	Kaggle
Sloan Digital Sky Survey DR16	Kaggle
Term Deposit Prediction Data Set	Kaggle
trajectory-based-ship-classification	Kaggle
Travel Insurance	Kaggle

D Details on Causal Inference Results

Causal Inference Most real-world decision problems ultimately hinge on causal questions—understanding what would happen if we intervened, rather than merely observing correlations. Estimating Conditional Average Treatment Effects (CATEs) is one of the central ways to answer these "what-if" questions: how would an individual's outcome change if a treatment were applied versus withheld?

Unconfounded Settings. Many causal inference methods require *unconfoundedness*, which broadly states that there are no features not included in the dataset that influence both the *treatment* variable and the outcome [32]. While recent studies have begun to challenge the validity and verifiability of this assumption [15, 33], there are presently a wide variety of causal inference methods designed for the unconfounded setting [34, 35].

Importance of Base Model. Recent empirical findings have shown that when unconfoundedness holds, CATE estimation can be framed as an AutoML problem [36], as many CATE estimators require a choice of classification or regression model to approximate the likelihood (propensity) of a treatment and an outcome given an individual's features. Parallel studies [15, 37] have shown that TabPFN is an especially strong choice for meta-learners such as the X-, T-, and S-Learner [38], hypothesizing that its strong performance in tabular prediction transfers to the problem of causal inference.

Table 3: Description of causal inference datasets in the RealCause benchmark.

Characteristic	ACIC-2016	IHDP	Lalonde-CPS	Lalonde-PSID
Realizations	10	100	100	100
Samples	4,802	747	16,177	2,675
Features	58	25	8	8

E The TabPFN Ecosystem

Figure 11 provides a minimal user workflow through components in the TabPFN–Extensions ecosystem.

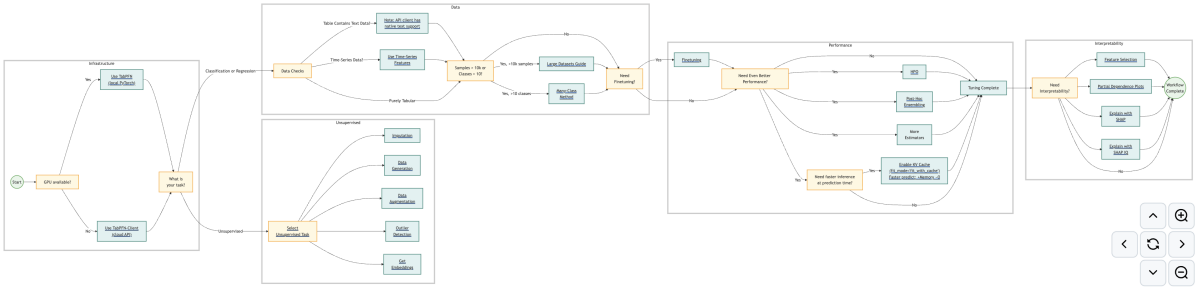


Figure 11: A minimal user workflow through components in the TabPFN–Extensions ecosystem.

F Additional Internal Benchmark Details

F.1 Details on the normalization

For benchmarking, we normalize scores per dataset to enable averaging and clearer comparison across datasets, ensuring that differences in dataset difficulty do not bias comparisons. For each dataset, we linearly scale scores between 0 (worse model on this dataset) and 1 (best model). For each model, the default and tuned versions are considered as two different models for the normalization. Bar heights show the mean normalized performance, and error bars denote the standard error of the mean (SEM) across datasets, reflecting uncertainty from dataset variability.

F.2 Additional results on large features

In Figure 12, we show results on an internal set of datasets containing from 500 to 2,000 features showing strong default performance.

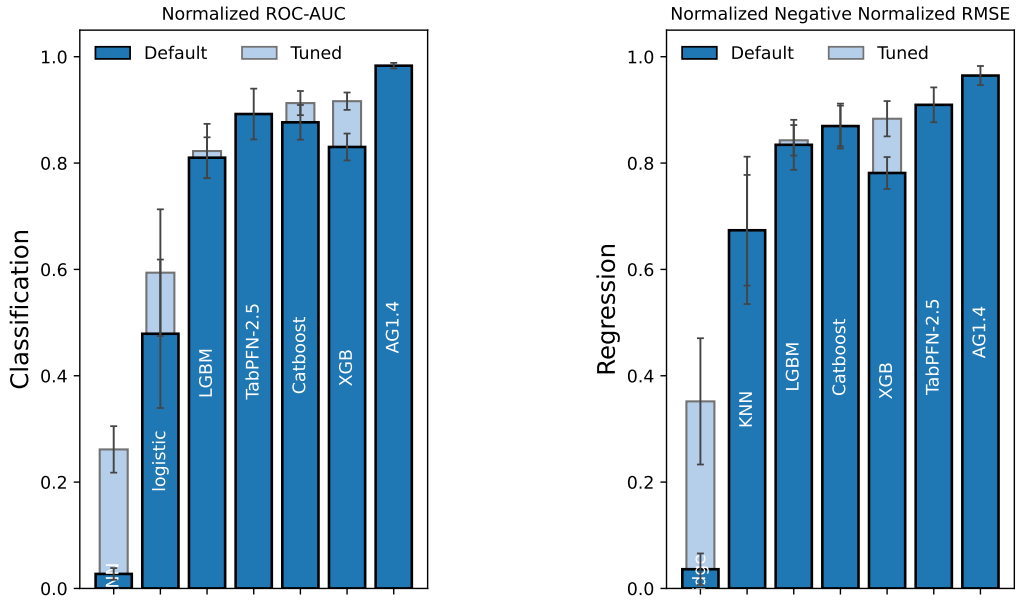


Figure 12: **TabPFN-2.5 default performs well up to 2,000 features**. In our internal benchmark on datasets from 500 features to 2,000 features, we can see that for both classification (left) and regression (right), the default TabPFN-2.5 outperforms any other default model and is better than any tuned single model for regression.

G Detailed TabArena Results

In addition to the results shown in Section 4, we compare our TabPFN-2.5 model to other foundation models in more detail below. In Figure 13, we show that TabPFN-2.5 outperforms TabICL on datasets compatible with both models, and in Figure 14, we show much better performance when compared to LimiX's results on datasets with less than 50,000 samples and 2,000 features, which corresponds to the datasets on which the TabArena maintainers could run LimiX at the time of writing.

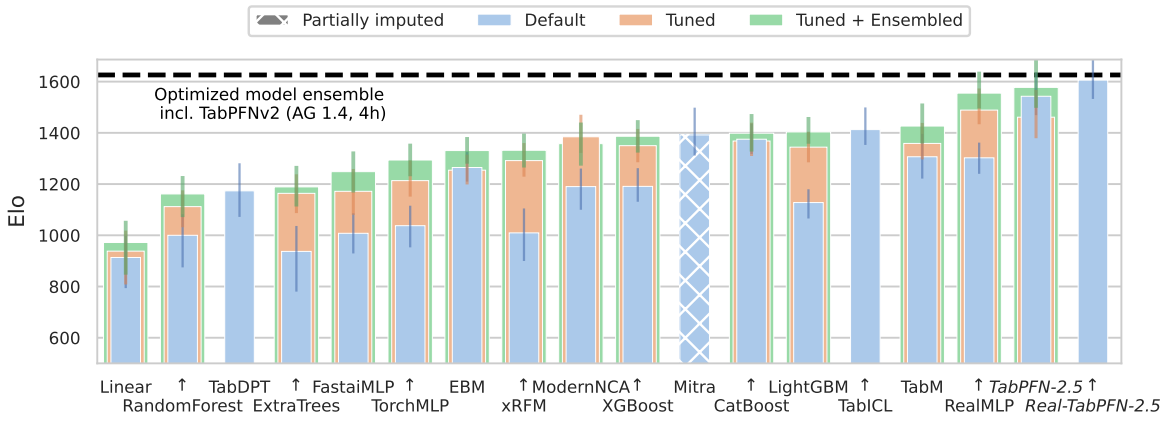


Figure 13: **Comparison with TabICL** [25]. In this plot, we show the performance of TabPFN-2.5 and TabICL on a TabArena-lite subset compatible with both models, restricting to **classification datasets with less than 50K training samples and less than 500 features**. On this subset, we see that TabPFN-2.5 significantly outperforms TabICL.

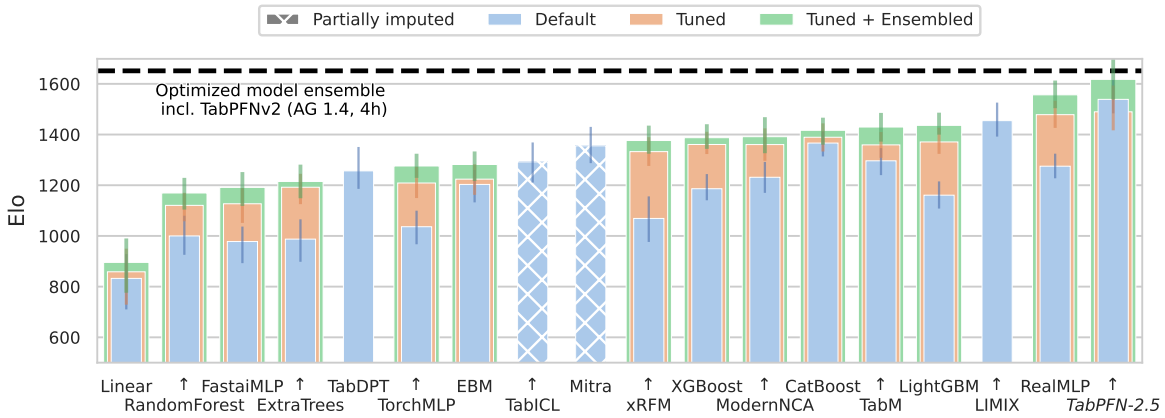


Figure 14: **Comparison with LimiX** [26]. In this plot, we show the performance of TabPFN-2.5 and LimiX on datasets from TabArena-Lite with **less than 50,000 training samples and less than 2,000 features**. On this subset, we see that TabPFN-2.5 significantly outperforms LimiX. Note that these results are still unverified by the original authors at the time of writing and thus not included in the main paper results.

H Results with Tuned Decision Thresholds

Starting with TabPFN-2.5, our framework supports tuning the decision threshold to optimize for specific metrics. Figure 15 quantifies the performance gains that this procedure can yield, illustrating substantial improvement in F1-score for several imbalanced datasets when tuning the threshold.

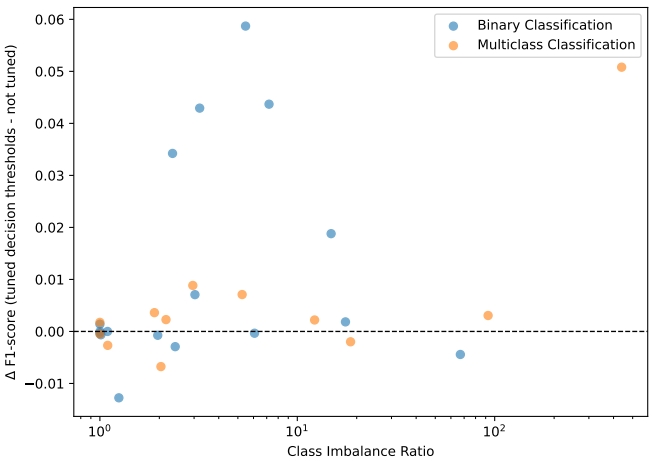


Figure 15: Absolute F1-score improvement from decision threshold tuning. The plot shows the difference in F1-score (macro) between a model with an optimized decision threshold and the same model using a default (untuned) threshold. This demonstrates the effectiveness of the tuning procedure for metric-specific optimization.

I Supplementary Inference Time Details

Figure 16 shows the inference latency you can expect for three common models of GPUs. Figure 17 shows that the time scales linearly with the number of test rows.

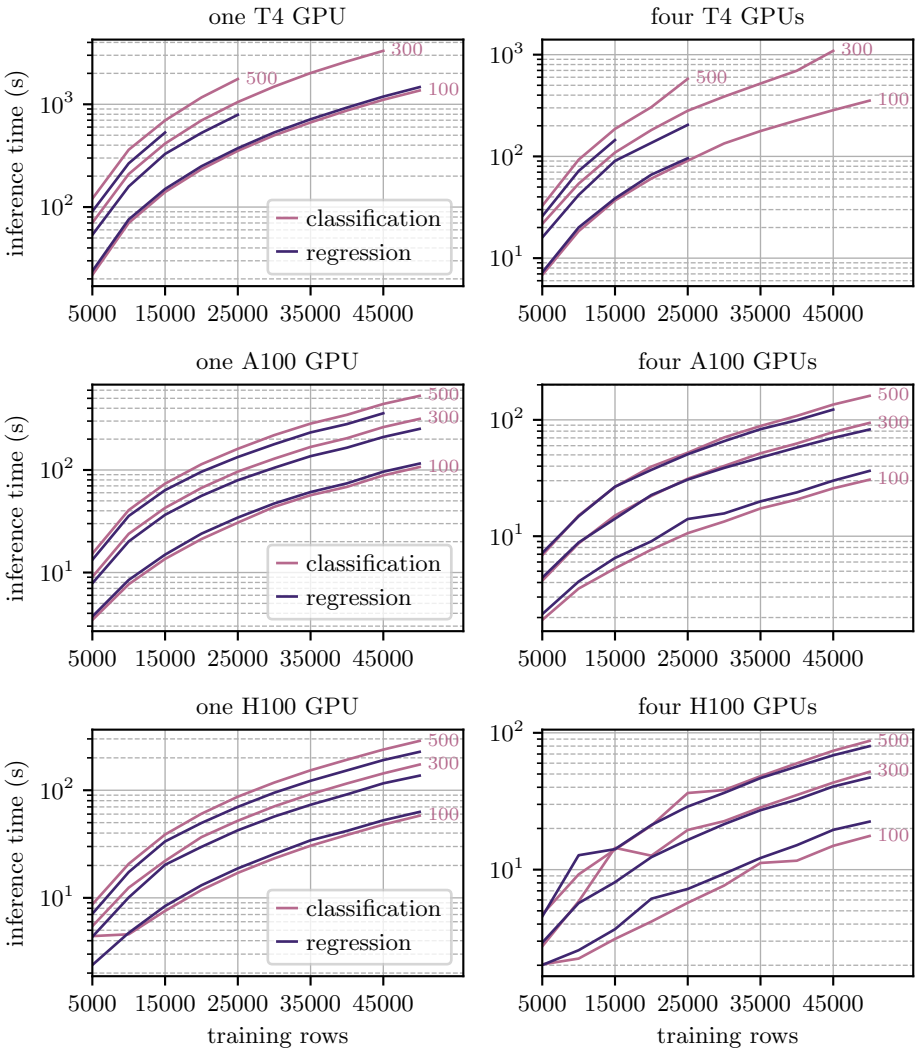


Figure 16: Time taken, in seconds, to train TabPFN-2.5 models on various training set sizes, and then make predictions on 500 test rows, using three common models of NVIDIA GPU: T4 15GB, A100 SXM 40GB, H100 SXM 80GB. Performance is shown for 100, 300, and 500 features. Datasets with more than 500 features have the same performance as datasets with 500, as each estimator will subsample to 500 features. Incomplete lines indicate that the GPU had insufficient memory for that dataset size.

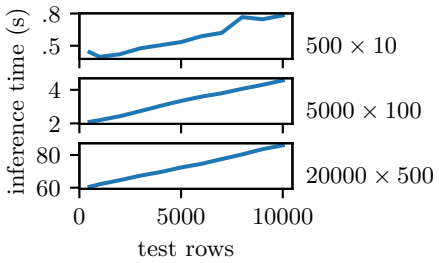


Figure 17: The time taken by TabPFN-2.5 to train and predict scales linearly in the test set size, shown here for a classification model trained on datasets of 500 rows × 10 features, 5,000 rows × 100 features, and 20,000 rows × 500 features. Measured on one H100 GPU.