Towards Localization via Data Embedding for TabPFN

Mykhailo Koshil[1] **Thomas Nagler**[2] **Matthias Feurer**[2] **Katharina Eggensperger**[1]

[1] University of Tübingen
first.last@uni-tuebingen.de

[2] Department of Statistics, LMU Munich
Munich Center for Machine Learning
first.last@stat.uni-muenchen.de

Abstract

In-context learning (ICL) using Prior-data fitted networks (PFNs) like TabPFN has shown significant promise in supervised tabular learning tasks. However, scalability is limited by the quadratic complexity of the transformer architecture's attention across training points provided as context. A recent theoretical analysis suggested localization to overcome this issue. In this work, we propose LE-TabPFN implementing a new localization method that performs nearest neighbor selection using the model's learned internal representations. We evaluate LE-TabPFN across six datasets, demonstrating superior performance over standard TabPFN when scaling to larger datasets. We also explore design choices and analyze the bias-variance trade-off, showing that it desirably reduces bias while maintaining manageable variance. This work opens up a pathway for scaling TabPFN and ICL methods in general to arbitrarily large tabular datasets.

1 Introduction

Prior-data fitted networks (PFNs; S. Müller et al., 2022) are a class of neural networks that are trained on synthetic prior data, i.e., tabular classification tasks, and perform in-context learning for new tasks. TabPFN (Hollmann et al., 2023), a specific implementation of PFNs for tabular data, has shown impressive performance, often rivaling state-of-the-art models such as random forests and gradient boosting (McElfresh et al., 2023). However, a fundamental limitation of TabPFN is its use of a transformer architecture (Vaswani et al., 2017), which scales quadratically with the number of training points due to the self-attention mechanism. TabPFN was trained on up to 1024 training data points, yet, in a scaling experiment, the model demonstrated improved performance up to 4096 data points (Hollmann et al., 2023). Nagler (2023) studied the underlying statistical foundations, conducted a bias-variance analysis of the PFN model, and found that improved performance for larger datasets is due to a reduction in variance, demonstrating this using a simple toy experiment. In this work, we extend this preliminary experimental study and propose LE-TabPFN. Concretely, we contribute a **practical and principled method to localize TabPFN**, an **exploration of its design decisions**, and **empirical analysis of its performance on** 6 **datasets**. Our localization method improves performance over TabPFN for large datasets and, thus, is a promising candidate method for scaling TabPFN and future in-context learning methods to arbitrarily large datasets.

2 Background

TabPFN (Hollmann et al., 2023) belongs to the broader class of prior-data fitted networks (PFNs, S. Müller et al., 2022). It is a foundation model that is pre-trained on synthetic supervised learning tasks to approximate $p(y|\mathbf{x}_*, \mathcal{D})$, i.e., $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n) = \mathcal{D}$ and a query point $\mathbf{x}_*$, for which we want to predict $\hat{y}_*$. TabPFN uses in-context learning, which, in contrast to traditional

3rd Table Representation Learning Workshop at NeurIPS (TRL@NeurIPS 2024).

machine learning models, does *not* require training, hyperparameter tuning, or gradient updates. Instead, a forward pass through the network yields predictions for a new, unseen task.

While TabPFN has demonstrated robust predictive performance, its reliance on a transformer-based self-attention mechanism leads to quadratic scaling of computation costs wrt context size. To overcome this, prior works studied dataset distillation and context optimization (Thomas et al., 2024; Feuer et al., 2024; Ma et al., 2024; Rundel et al., 2024). In contrast, LE-TabPFN is motivated by the theoretical analysis that, for TabPFN trained on a maximum of $1\,024$ data points performance improvement when presented with up to $4\,096$ data points (Hollmann et al., 2023) can be entirely attributed to a decrease in variance. Since the TabPFN architecture does not adequately localize the predictions around the test data point, the bias does not decrease with increasing dataset size (Nagler, 2023). Thus, **localization is key to scale TabPFN** and Nagler (2023) proposed a simple strategy. For each point $\mathbf{x}_*$

1. reduce the training set $\widetilde{\mathcal{D}}(\mathbf{x}_*)$ to the k nearest neighbors of $\mathbf{x}_*$ from $\mathcal{D}$,
2. apply TabPFN to predict the label corresponding to $\mathbf{x}_*$ using only $\widetilde{\mathcal{D}}(\mathbf{x}_*)$ as context.

With that, bias and the overall error decrease when going beyond what TabPFN was trained for, giving rise to a strategy that allows the exploitation of large datasets. These results provide a theoretical foundation and motivation for exploring localization.

3 Method

We propose to base the **l**ocalization on learned **e**mbeddings, extracted at intermediate layers of TabPFN; thus, we dub our approach LE-TabPFN. We assume that the internal latent space provides more meaningful representations for localization than the raw feature space. In practice, this contains several design choices: (D1) A layer in TabPFN to read out the transformed representation. (D2) A separate and fixed context that is used to embed new data points into the intermediate representation (D3) A distance function between points in the embedded space.

Choice (D1) depends on the TabPFN architecture. The current implementation [1] uses an encoder-only architecture with 12 layers, followed by a 2-layer fully-connected neural network. The embedding is the output of the transformer encoder block at the respective layer averaged over the ensemble dimension. We chose the last encoder block unless specified otherwise. For this initial study, we chose the most straightforward possibilities for (D2) and (D3): a random context of size $1\,024$ and the Euclidean distance. By wrapping this around TabPFN's scikit-learn interface (Pedregosa et al., 2011), we localize the context on a per-query basis, scaling the features for each test point independently.

4 Exploratory Experiments

Next, we turn to experiments. We first validate our findings by running the same bias-variance decomposition from Nagler, 2023 with our method and then study the behavior of LE-TabPFN on 6 datasets wrt performance and impact of design decision.

4.1 Bias-Variance Decomposition

To validate our localization approach, we replicate the bias-variance decomposition experiment from Nagler (2023) and compute the bias-variance decomposition of the RMSE. For this, we first simulate 1000 datasets $\mathcal{D}_n$ from $p_0(1|X) = \frac{1}{2} + \sin(1^T\mathbf{X})/2$ with $Y \in \{0,1\}, X \sim \mathcal{N}(0, I_5)$, and apply TabPFN and LE-TabPFN. Then, we compute the average squared bias and variance over 1024 samples $X_{test} \sim \mathcal{N}(0, I_5)$. In contrast to the original experiment, which only used up to 4000 data points in a single dataset, we used up to 8192 data points. Also, we extend the experiment and not only use the raw features for localization, but also the learned embeddings as described in the previous section. Our results in Figure 1 confirm that the localization method reduces bias compared to the original TabPFN, while the increase in variance remains small. In addition, we observe that using the embedding for contextualization leads to lower bias and variance than using the raw features.

[1] See `https://github.com/automl/TabPFN`, version $0.1.9$

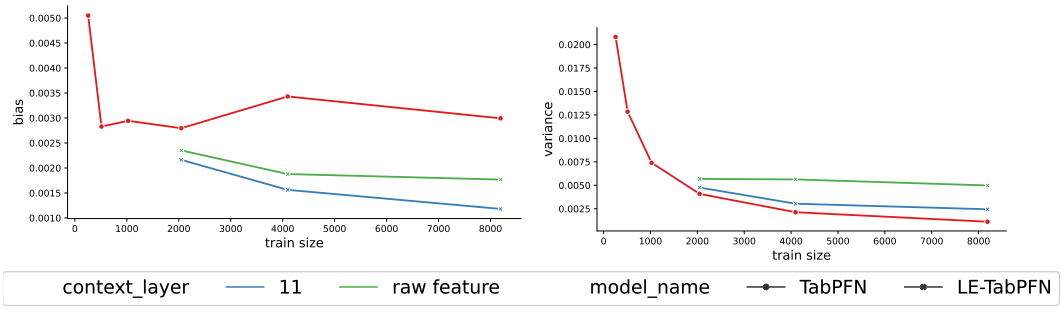

Figure 1: Bias-variance decomposition of the prediction error of TabPFN and LE-TabPFN with increasing training data size.

4.2 Empirical Evaluation

Experimental Setup. We use a total of 6 datasets: three datasets previously used to demonstrate scaling effects in TabPFN (adult-census, electricity and eeg-eye-state, Thomas et al., 2024),[2] and three large datasets that only contain numerical features (Higgs, Covertype, and MiniBooNe) and less than 100 features to replicate the training setting of the TabPFN. We obtained the datasets provided by OpenML (Vanschoren et al., 2014) using OpenML-Python (Feurer et al., 2021). We utilize 3 folds and the entire test set of the respective OpenML tasks.

Does the localization allow scaling to arbitrary dataset sizes? We first investigate whether LE-TabPFN can leverage additional data to improve performance, as hypothesized. We compare median AUC over dataset size for LE-TabPFN (blue), TabPFN with up to 8 192 data points (black), TabPFN with random subsamples (red), and a random forest (purple; Breiman, 2001) in Figure 2. LE-TabPFN continues to improve with larger training sets, while standard TabPFN with random subsamples plateaus. LE-TabPFN also improves over TabPFN with up to 4 096 data points, which suggests that the reduction in bias outweighs the reduction in variance due to the increased number of data points.

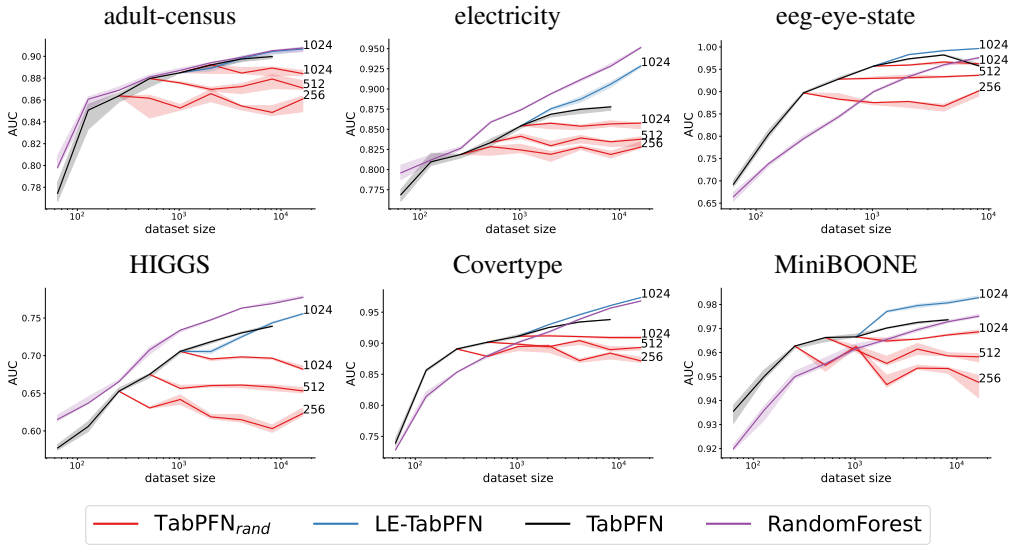

Figure 2: Median AUC over dataset size.

What is the impact of the readout layer? Next, we study whether later layers better capture the relation between data points, leading to embeddings that produce a better context and performance.

[2]We note that adult-census and electricity are suboptimal to examine TabPFN as they contain missing values and categorical features, two dataset characteristics that TabPFN was not trained on. We impute missing values with the per-feature mean.

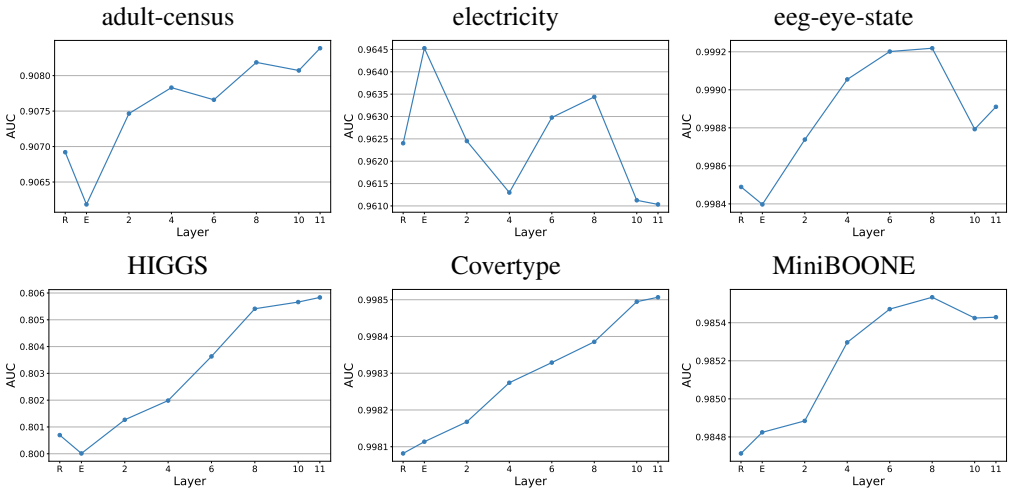

Figure 3: Performance of LE-TabPFN as a function of the readout layer. -1 is the raw data, 0 is the embedding layer of the transformer, and positive numbers are the respective encoder blocks.

Table 1: We report AUC in percentage for Random Forests and different variations ofLE-TabPFN and TabPFN. Top: Using 8 192 training data points. Middle: Using all training data points. Bottom: LE-TabPFN using a remote context (see Section 4.2).

#train	model	context layer	CoverType	Higgs	MiniBooNe	adult-census	eeg-eye-state	electricity
8 192	TabPFN$_{rand}$		0.8928	0.6519	0.9597	0.8707	0.9307	0.8355
	TabPFN		0.9382	0.7395	0.9738	0.8992	0.9611	0.8751
	RandomForest		0.9563	**0.7709**	0.9729	**0.9037**	0.9743	**0.9299**
	LE-TabPFN	raw feature	0.9539	0.7428	0.9813	0.9012	0.9971	0.9042
		0	0.9582	0.7364	**0.9814**	0.9014	**0.9972**	0.9083
		11	**0.9610**	0.7442	0.9812	0.9024	0.9970	0.9047
full	TabPFN$_{rand}$		0.9080	0.6914	0.9651	0.8866	0.9265	0.8515
	RandomForest		0.9973	**0.8156**	0.9804	0.9072	0.9851	**0.9718**
	LE-TabPFN	raw features	0.9981	0.8007	0.9847	0.9069	0.9985	0.9624
		0	0.9981	0.8000	0.9848	0.9062	0.9984	0.9645
		11	**0.9985**	0.8058	**0.9854**	**0.9084**	**0.9989**	0.9610
full	Remote context TabPFN	raw features	0.4441	0.5542	0.6065	0.7693	0.4559	0.6034
		0	0.3337	0.4381	0.1213	0.2963	0.4279	0.2770
		11	0.2775	0.3422	0.1686	0.1628	0.0972	0.2503

Figure 3 shows that computing the neighborhood based on internal embeddings improves over using the raw feature space. However, the absolute difference in AUC is surprisingly small (with the largest difference for CoverType). This suggests that the original feature space remains highly informative for these datasets.

To investigate this further, we study whether using a "remote" context – comprising the most distant data points – rather than a local one results in degraded performance. The last row in Table 1 confirms that performance declines drastically when using the last layer compared to the first, with AUC dropping below the chance level of 0.5. This indicates that later layers indeed capture different information compared to earlier ones. We suspect the raw features are "too informative" for our datasets, and Euclidean distances in the original space are good enough for localization. To examine this, we augment the dataset with random features, simulating irrelevant features (as is typical for real-world tabular data). This reduces the meaningfulness of distances in the raw feature space. Results in Figure 4 support this hypothesis: when we add random features, the performance improves as we use embeddings from later layers, outperforming both earlier layers and raw features. This suggests that learned representations are more informative for building a local context.

How does TabPFN perform compared to LE-TabPFN? Lastly, we draw a quantitative comparison between TabPFN, using $1\,024$ randomly subsampled datapoints (TabPFN$_{rand}$) , a random forest (RF) trained on all data, and our LE-TabPFN, on subsamples of $8\,192$ data points, and the full datasets. We give all results in Table 1 and can observe that LE-TabPFN improves over TabPFN$_{rand}$ on all studied datasets. Also, while TabPFN is inferior to the RF on all studied datasets, LE-TabPFN is superior to

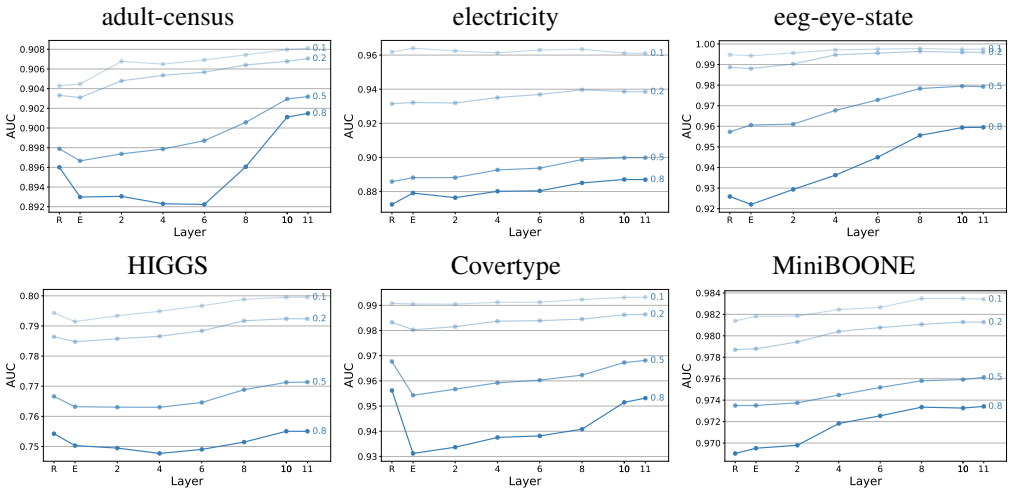

Figure 4: Performance (AUC) of LE-TabPFN as a function of the readout layer when adding various amounts of random features ($X \in \{0.1, 0.2, 0.5, 0.8\}$ stands for $X * n_{features}$ random features that are added to the original dataset to reduce the meaningfulness of the original representation).

RF on four out of six datasets and almost closes the performance gap on the remaining two datasets. As indicated by the experiment before, we do not find a strong impact of the readout layers on these datasets. Overall, we can see that localization can scale TabPFN to arbitrary training sizes.

5 Conclusion and Future Work

We demonstrated that the localization principle is a powerful paradigm for scaling TabPFN to supervised learning tasks with more than 1 024 training points. In the future, we plan to (1) include the localization in the pre-training step, as suggested by Nagler (2023), (2) optimize our approach for inference speed, and (3) study localization for other ICL models, such as TabLLM (Hegselmann et al., 2023) or MotherNet (A. Müller et al., 2023). Furthermore, we want to (4) extend this proof-of-concept to a large-scale comparison, including a similar idea motivated by RAG (Thomas et al., 2024), and other methods aiming to scale TabPFN by learning a single, static context (Feuer et al., 2024; Rundel et al., 2024; Ma et al., 2024).

Acknowledgments and Disclosure of Funding

Katharina Eggensperger and Mykhailo Koshil acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645 and by the Baden-Württemberg Ministry of Science and the Federal Ministry of Education and Research (BMBF) as part of the Excellence Strategy of the German Federal and State Governments. The authors also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Mykhailo Koshil.

References

Müller, S., N. Hollmann, S. Arango, J. Grabocka, and F. Hutter (2022). "Transformers Can Do Bayesian Inference". In: *Proceedings of the International Conference on Learning Representations (ICLR'22)*. Published online: `iclr.cc`.

Hollmann, N., S. Müller, K. Eggensperger, and F. Hutter (2023). "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second". In: *International Conference on Learning Representations (ICLR'23)*. Published online: `iclr.cc`.

McElfresh, D., S. Khandagale, J. Valverde, V. Prasad C, G. Ramakrishnan, M. Goldblum, and C. White (2023). "When Do Neural Nets Outperform Boosted Trees on Tabular Data?" In: *Proceedings of the 37th International Conference on Advances in Neural Information Processing Systems (NeurIPS'23)*. Curran Associates, pp. 76336–76369.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin (2017). "Attention is All you Need". In: *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'17)*. Ed. by I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc.

Nagler, T. (2023). "Statistical Foundations of Prior-Data Fitted Networks". In: *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 25660–25676.

Thomas, V., J. Ma, R. Hosseinzadeh, K. Golestan, G. Yu, M. Volkovs, and A. Caterini (2024). "Retrieval & Fine-Tuning for In-Context Tabular Models". In: *1st ICML Workshop on In-Context Learning*.

Feuer, B., R. Schirrmeister, V. Cherepanova, C. Hegde, F. Hutter, M. Goldblum, N. Cohen, and C. C. White (2024). "TuneTables: Context Optimization for Scalable Prior-Data Fitted Networks". In: *Accepted for Publication at the 37th Conference on Neural Information Processing Systems*.

Ma, J., V. Thomas, G. Yu, and A. Caterini (2024). "In-Context Data Distillation with TabPFN". In: *arXiv:2402.06971 [cs.LG]*.

Rundel, D., J. Kobialka, C. von Crailsheim, M. Feurer, T. Nagler, and D. Rügamer (2024). "Interpretable Machine Learning for TabPFN". In: *Explainable Artificial Intelligence*. Ed. by L. Longo, S. Lapuschkin, and C. Seifert. Vol. 2154, pp. 465–476.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Vanschoren, J., J. van Rijn, B. Bischl, and L. Torgo (2014). "OpenML: Networked Science in Machine Learning". In: *SIGKDD Explorations* 15.2, pp. 49–60.

Feurer, M., J. van Rijn, A. Kadra, P. Gijsbers, N. Mallik, S. Ravi, A. Müller, J. Vanschoren, and F. Hutter (2021). "OpenML-Python: an extensible Python API for OpenML". In: *Journal of Machine Learning Research* 22.100. Ed. by B. Kegl, pp. 1–5.

Breiman, L. (2001). "Random Forests". In: *Machine Learning Journal* 45, pp. 5–32.

Hegselmann, S., A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag (2023). "TabLLM: Few-shot Classification of Tabular Data with Large Language Models". In: *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 5549–5581.

Müller, A., C. Curino, and R. Ramakrishnan (2023). *MotherNet: A Foundational Hypernetwork for Tabular Classification*. arXiv: `2312.08598 [cs.LG]`. URL: `https://arxiv.org/abs/2312.08598`.

A Datasets used

Table 2: List of datasets used in the experiments

Name	OpenML Task ID	#Features	#Instances	#Classes
CoverType	7593	54	581 012	7
Higgs	360114	28	1 000 000	2
MiniBooNe	168335	50	130 064	2
adult-census	3953	15	32 561	2
eeg-eye	14951	14	14 980	2
electricity	219	8	45 312	2