Fast and Flexible: AI-Powered Tabular Modeling with TabPFN

Accurate Predictions on Small Data with a Tabular Foundation Model

SIH Seminar by Sebastian Haan

Outline

1. Introduction and Method Overview (15mins)
2. Questions (~2mins)
3. How to and Code Examples (Hands-on Notebook, 15mins)
4. Advanced Insights (Hands-on Notebooks, 15mins):
    - Accuracy of predicted uncertainties and probabilities
    - Model interpretability and feature importance
5. Conclusion and Discussion (10mins)

Main Sources

- TabPFN v1 (2023): https://arxiv.org/pdf/2207.01848v6 (https://github.com/PriorLabs/TabPFN/tree/v1.0.0)

- TabPFN v2 (2024): https://www.nature.com/articles/s41586-024-08328-6 (https://github.com/PriorLabs/TabPFN)

University of Freiburg (Freiburg, Germany)
PriorLabs, Freiburg
ELLIS Institute Tübingen (Tübingen, Germany)

The Challenge of Tabular Data

- **Tabular data is ubiquitous** across science and industry
- **Fundamental Prediction Task**: Generate predictions (and their uncertainties) based on given feature data (columns)/filling in missing data.
- **Limitations of Deep learning models:** have historically struggled with tabular and the prevalence of small, independent datasets. (76% of datasets on openml.org have less than 10,000 rows.)
- **Traditional Approaches:** Gradient-boosted decision trees have been the dominant approach for tabular data for the past 20 years
- **Need for a New Approach:** Tabular Prior-data Fitted Network (TabPFN) is a tabular foundation model designed for small-to-medium-sized datasets.

Introducing TabPFN

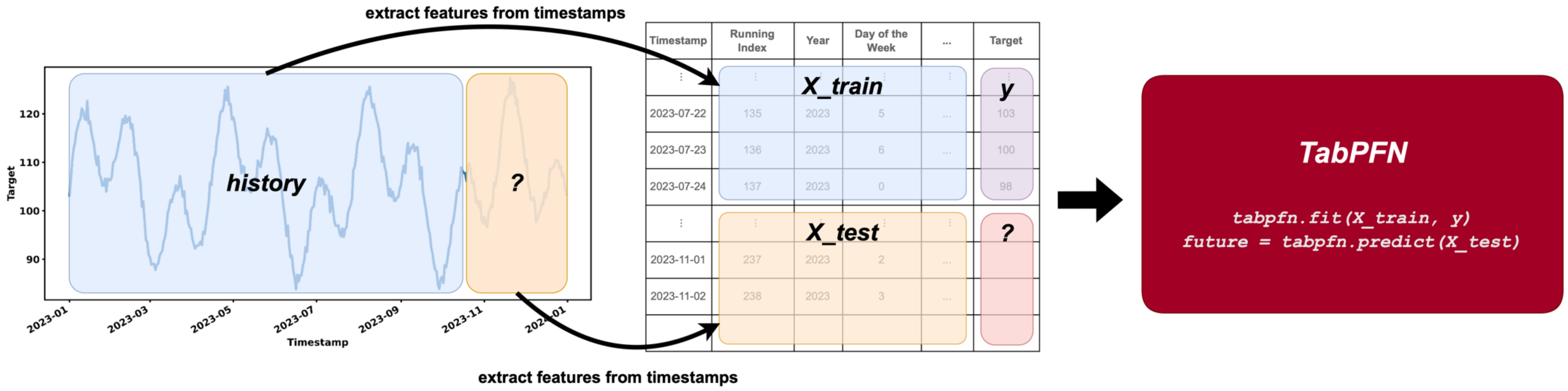
- **Outperforms** all previous methods on datasets with up to 10,000 samples.

- **In-Context Learning (ICL):** inspired by large language models, allowing it to learn from training data in a single forward pass and apply that to unseen test data.

- **Learns a tabular prediction algorithm** across millions of synthetic datasets.

→ *Learning a general algorithm for problem solving*

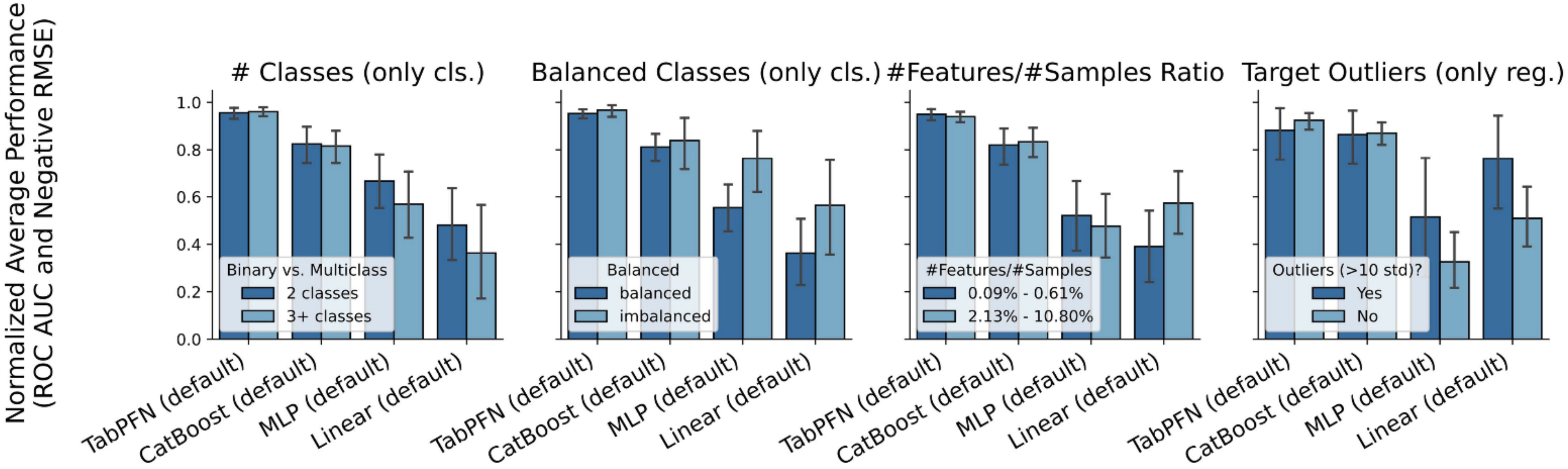
- **Foundation Model for Tabular Data:** TabPFN is a transformer-based foundation model and adapts its architecture for the 2D nature of tabular data.



Key Advantages of TabPFN

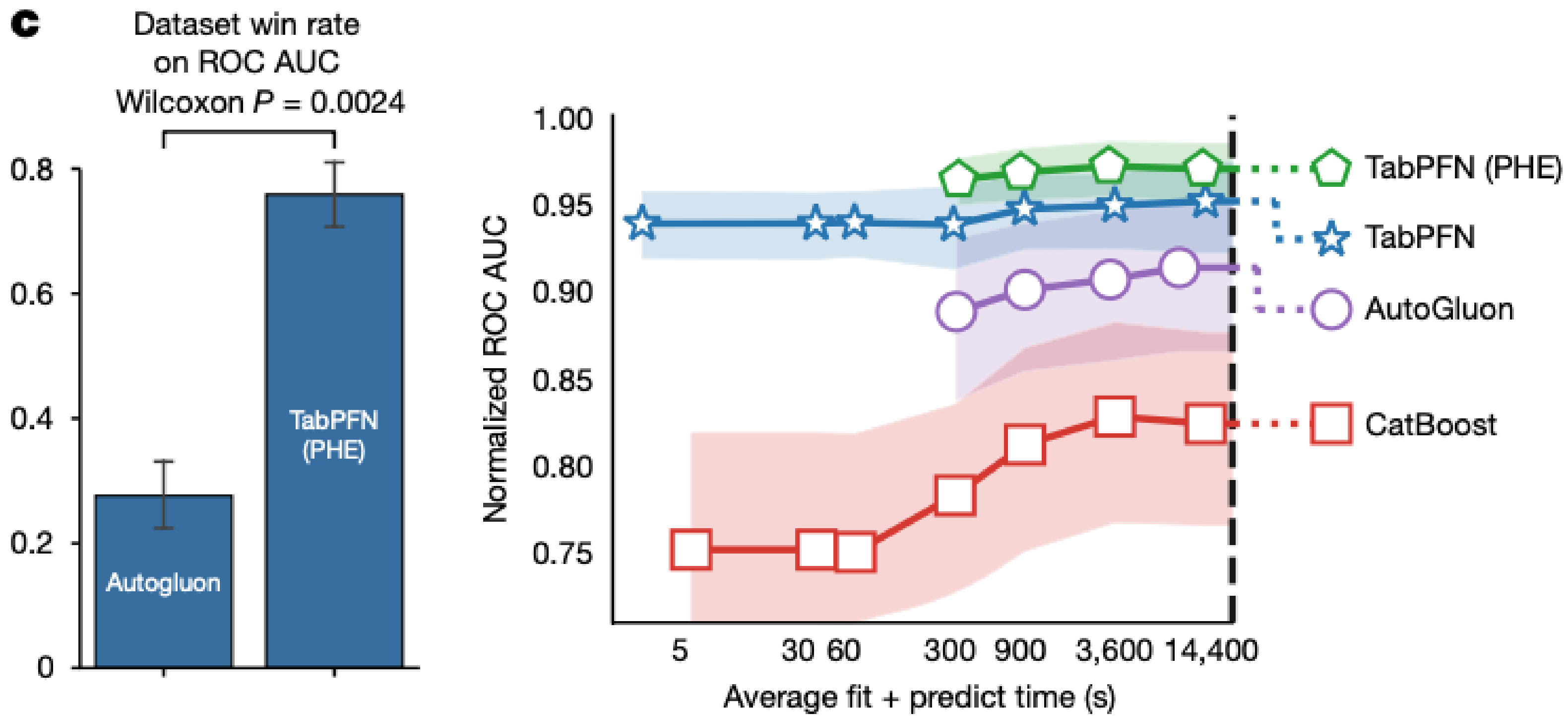
- **Accuracy:** Dominant performance on datasets with up to 10,000 samples and 500 features.
- **Speed:** In some cases it achieves "a speedup of 5,140x (classification) and 3,000x (regression)" compared to tuned state-of-the-art methods.
- **Robustness:** Handles missing values, categorical data, and outliers effectively. (added in v2).
- **Uncertainty Modeling:** Provides target (posterior) distribution, capturing prediction uncertainty and handling of multi-modal distributions
- **Interpretability**: Achieves high accuracy with simple, interpretable feature relationships
- **Generative Capabilities**: not just a predictive model but can also be used for fine-tuning, data generation, density estimation, and learning reusable embeddings.

Performance and Evaluation



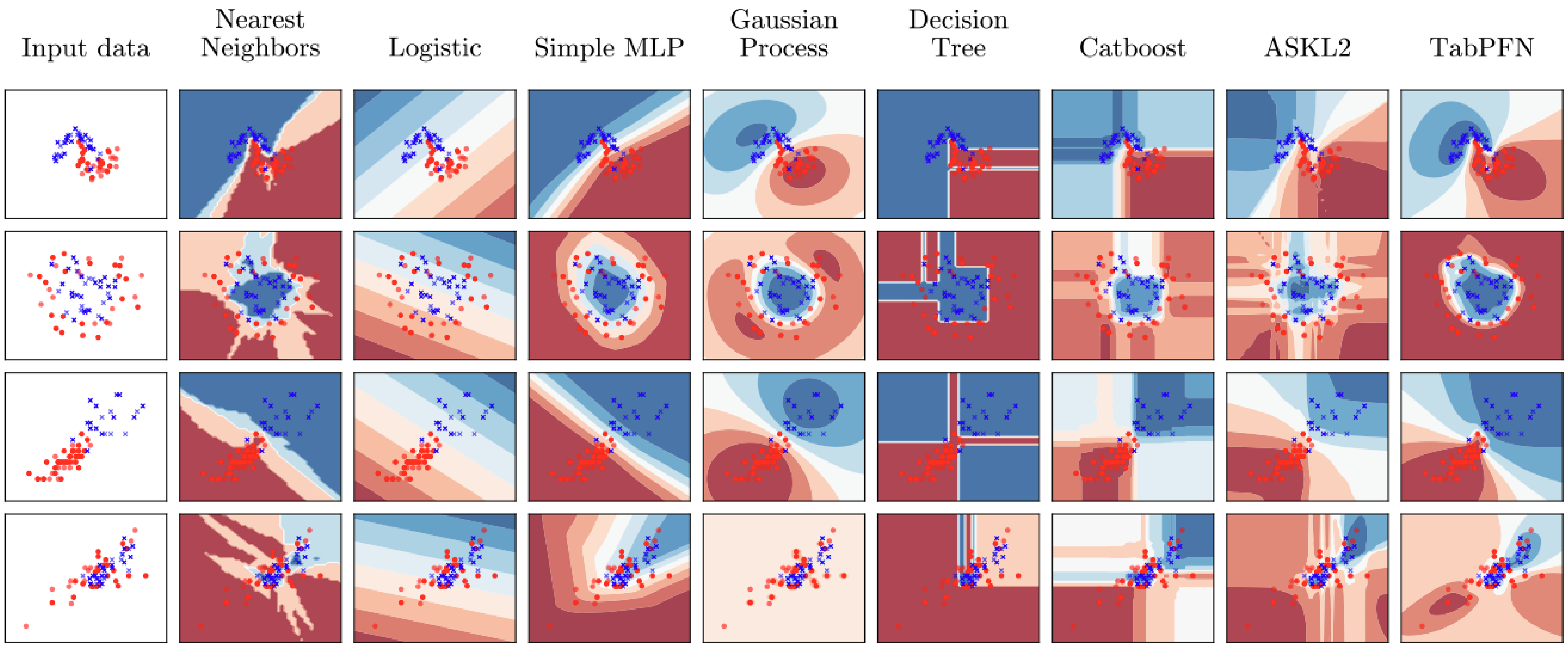
- Comparison with Baselines: TabPFN is compared against several state-of-the-art baselines, including tree-based methods (XGBoost, CatBoost, LightGBM), linear models, SVMs, and MLPs.
- Superior Performance: TabPFN outperforms tuned versions of baselines on most datasets, achieving "0.187" higher normalized ROC AUC than CatBoost in the default setting for classification and "0.051" higher normalized RMSE for regression.
- Robustness: It shows robustness against uninformative features, outliers and missing data.
- Reduced Sample Requirements: performs as well as the best baseline with half of training samples.

Comparison with tuned ensemble methods

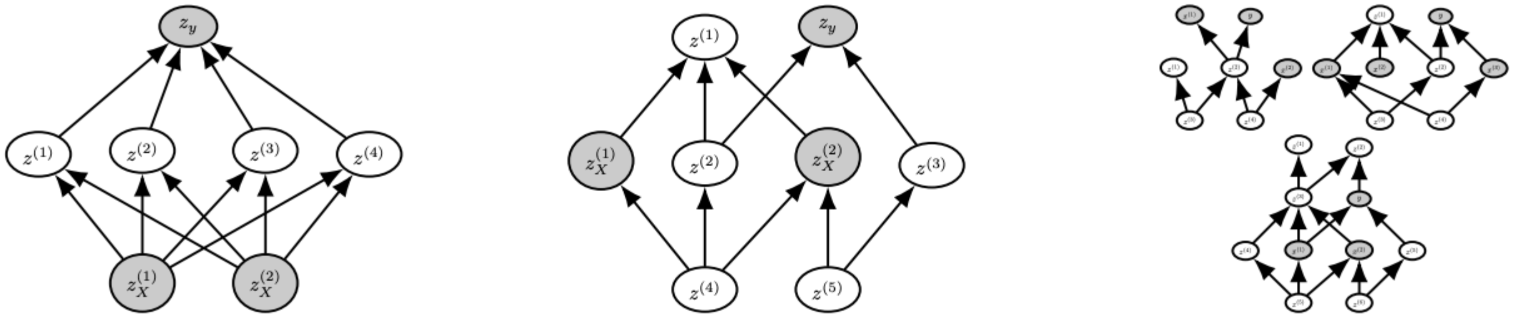


Robustness across datasets and performance comparison across tuned ensembles

Comparison Decision Boundaries



TABPFN’s Synthetic Data Generation Process



(a) A BNN (b) An SCM (c) SCMs sampled from the prior

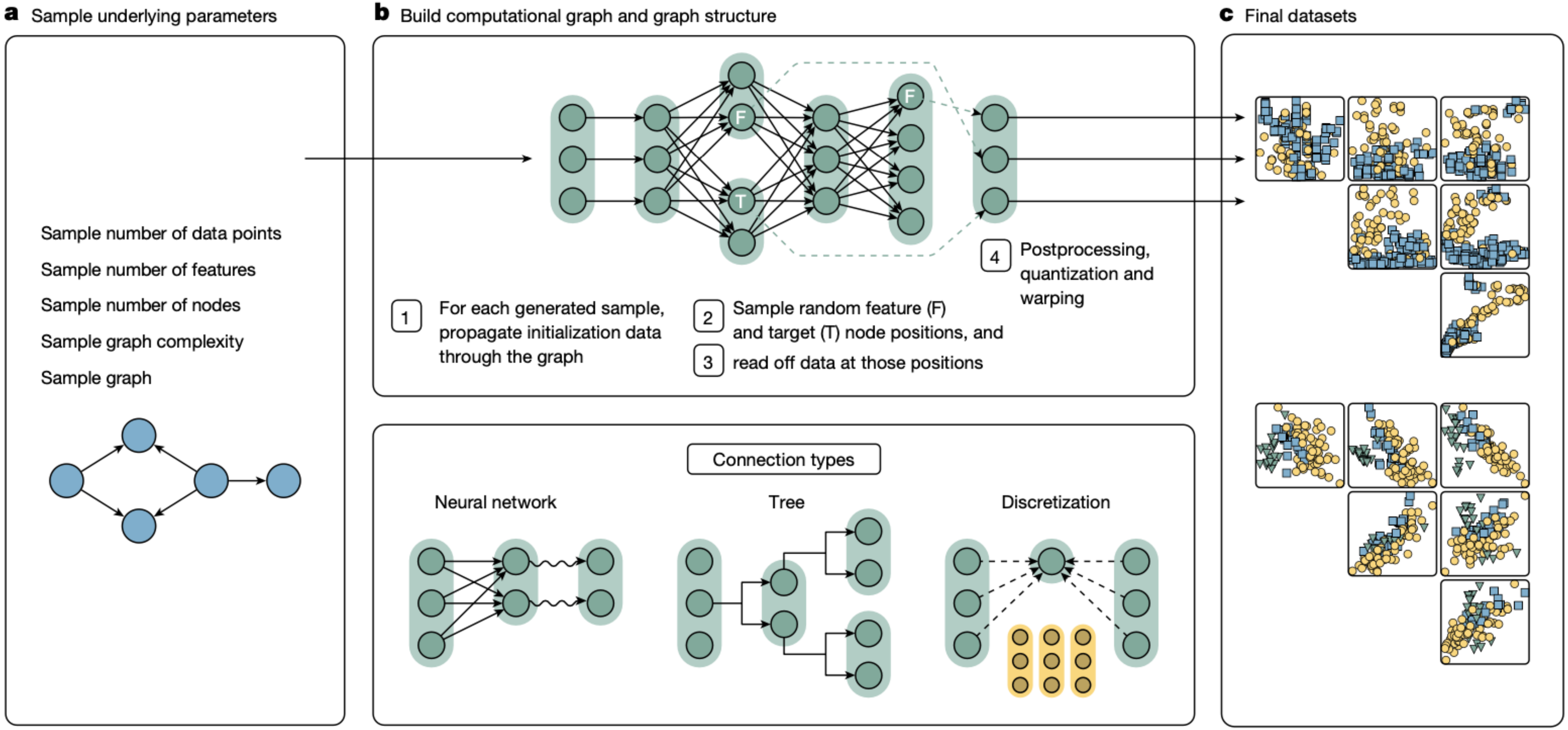
Structural Causal Models (SCMs): for representing causal relationships

Hyperparameter Control: samples hyperparameters like dataset size, number of features

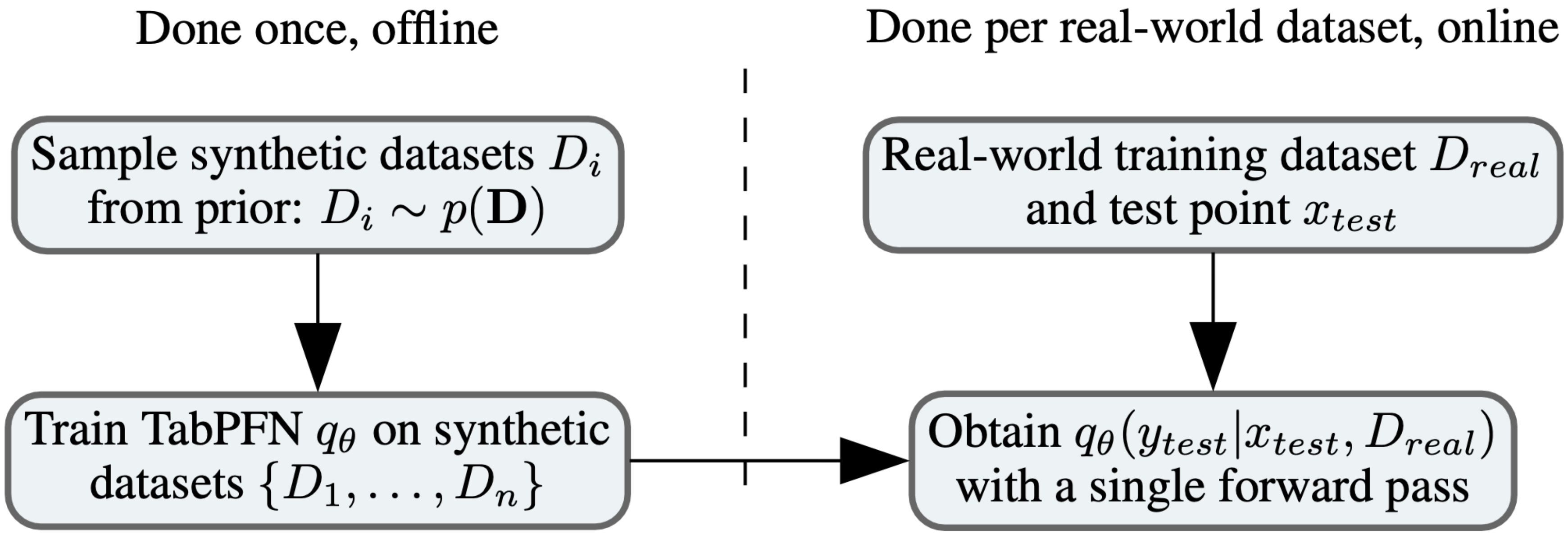
Diverse Mappings: The data generation process employs neural networks, decision trees, and discretization mechanisms, and gaussian noise allowing the model to represent complex relationships

→ **Massive corpus of around 100 million synthetic datasets** for model

Overview of TabPFN Prior

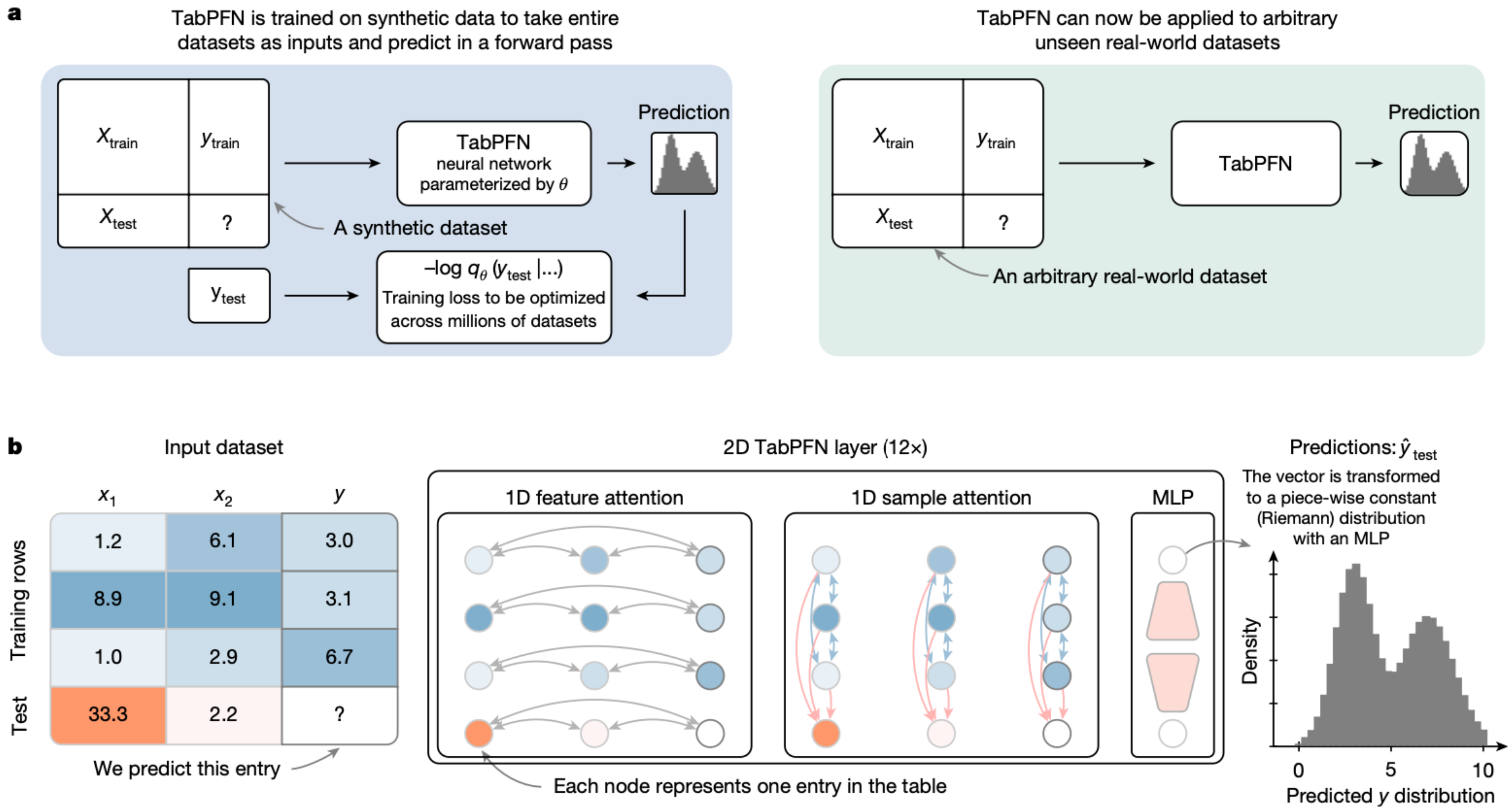


Bayesian Framework



- Can be viewed as **approximating Bayesian prediction**

Inference

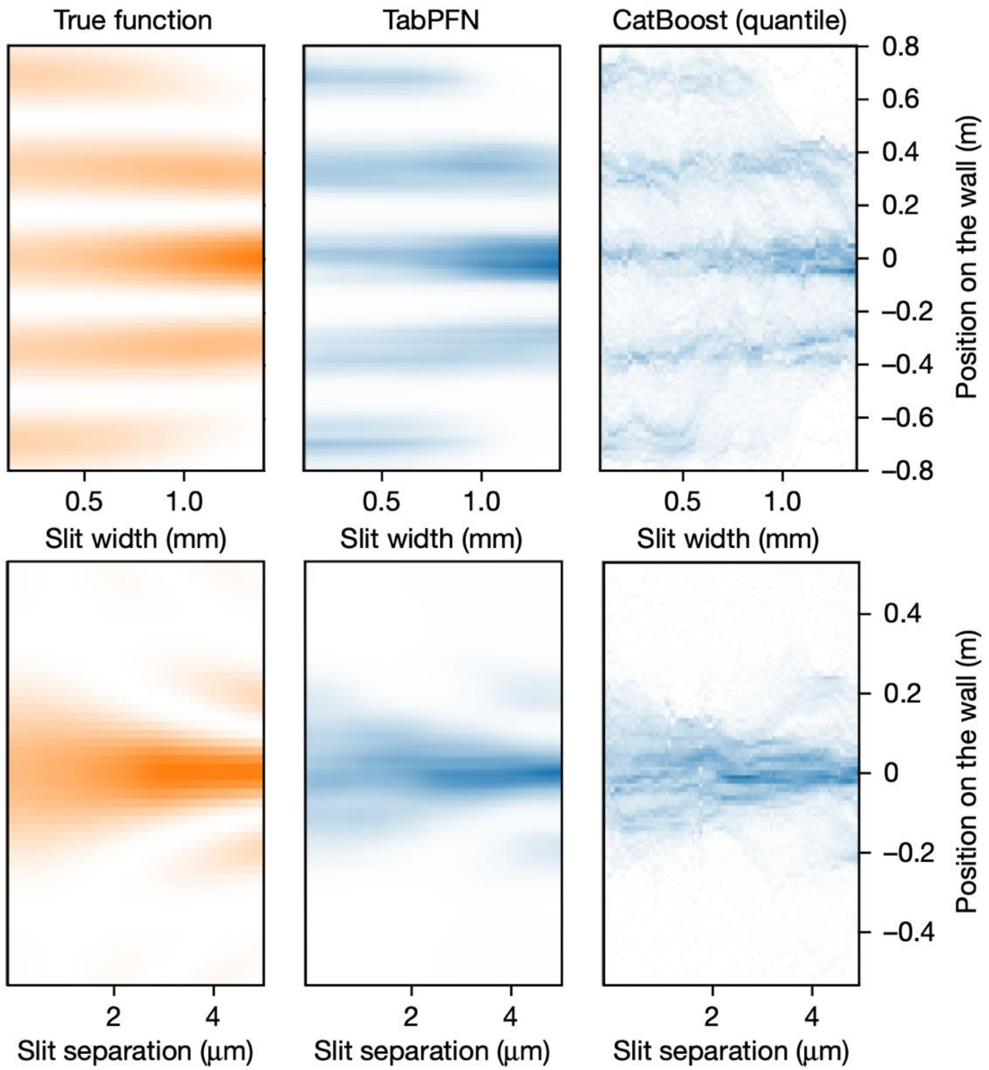


a
TabPFN is trained on synthetic data to take entire datasets as inputs and predict in a forward pass
X_{train}
y_{train}
X_{test}
?
TabPFN
neural network parameterized by θ
Prediction
A synthetic dataset
y_{test}
$-\log q_\theta (y_{test} | ...)$
Training loss to be optimized across millions of datasets
TabPFN can now be applied to arbitrary unseen real-world datasets
X_{train}
y_{train}
X_{test}
?
TabPFN
Prediction
An arbitrary real-world dataset
b
Input dataset
x_1
x_2
y
Training rows
Test
1.2
6.1
3.0
8.9
9.1
3.1
1.0
2.9
6.7
33.3
2.2
?
We predict this entry
2D TabPFN layer (12×)
1D feature attention
1D sample attention
MLP
Each node represents one entry in the table
Predictions: $\hat{y}_{test}$
The vector is transformed to a piece-wise constant (Riemann) distribution with an MLP
Density
0
5
10
Predicted y distribution

Example: Double Split Experiment

- TabPFN: able to accurately model the complex light patterns… in a single pass!

- Traditional methods (such as CatBoost) struggle because need extra steps and finetuning



How to Examples

Notebooks at: SIH github address TabPFN-seminar

- **TabPFN_Intro.ipynb:** Basic usage for classification and regression

- **TabPFN_UncertaintyEval.ipynb:** Testing accuracy of predictive probabilities and quantiles

- **TabPFN_Insights.ipynb:** Explain model prediction with SHAP and Feature Selection

Limitations

Dataset Size: Primarily designed for smaller to medium sized datasets

Inference Speed: Inference speed is slower compared to optimized traditional models.

Memory Usage: Memory usage scales linearly with dataset size.

Scalability: The evaluation focused on datasets with up to 10,000 samples and 500 features, and "scalability to larger datasets requires further study".

No replacement for tradition methods: in some cases traditional methods might be a better fit.

References

TabPFN v1 (2023): https://arxiv.org/pdf/2207.01848v6 (https://github.com/PriorLabs/TabPFN/tree/v1.0.0)

TabPFN v2 (2024): https://www.nature.com/articles/s41586-024-08328-6
(https://github.com/PriorLabs/TabPFN)

"**Fitting TabPFN Models in R Using Reticulate**" by Måns Thulin: Although focused on R, this blog post demonstrates how to utilize TabPFN through Python's reticulate package: https://mansthulin.se/posts/tabpfn

Experimental R Package: https://github.com/PriorLabs/R-tabpfn

TABPFN-TS for timeseries analysis
https://github.com/liam-sbhoo/tabpfn-time-series

TabPFN Client **https://github.com/automl/tabpfn-client** Easy-to-use API client for cloud-based inference

Webinterface
https://ux.priorlabs.ai/

Conclusions

Paradigm Shift: TabPFN represents a **major change** in tabular data modeling by leveraging in-context learning to autonomously discover efficient algorithms, outperforming traditional methods on small datasets.

Foundation Model Potential: TabPFN's ability to learn from synthetic data opens new possibilities for tabular data analysis and facilitates capabilities such as **data generation, density estimation, and fine-tuning.**

Great for **generating multi-modal predictive distributions**, not limited to point estimates or assuming normal distributions.

Future Directions: Future research could explore scaling TabPFN to **larger datasets and creating specialized priors** for various data types and modalities (e.g., ECG, neuroimaging data53 and genetic data), or integrating in other AI system.