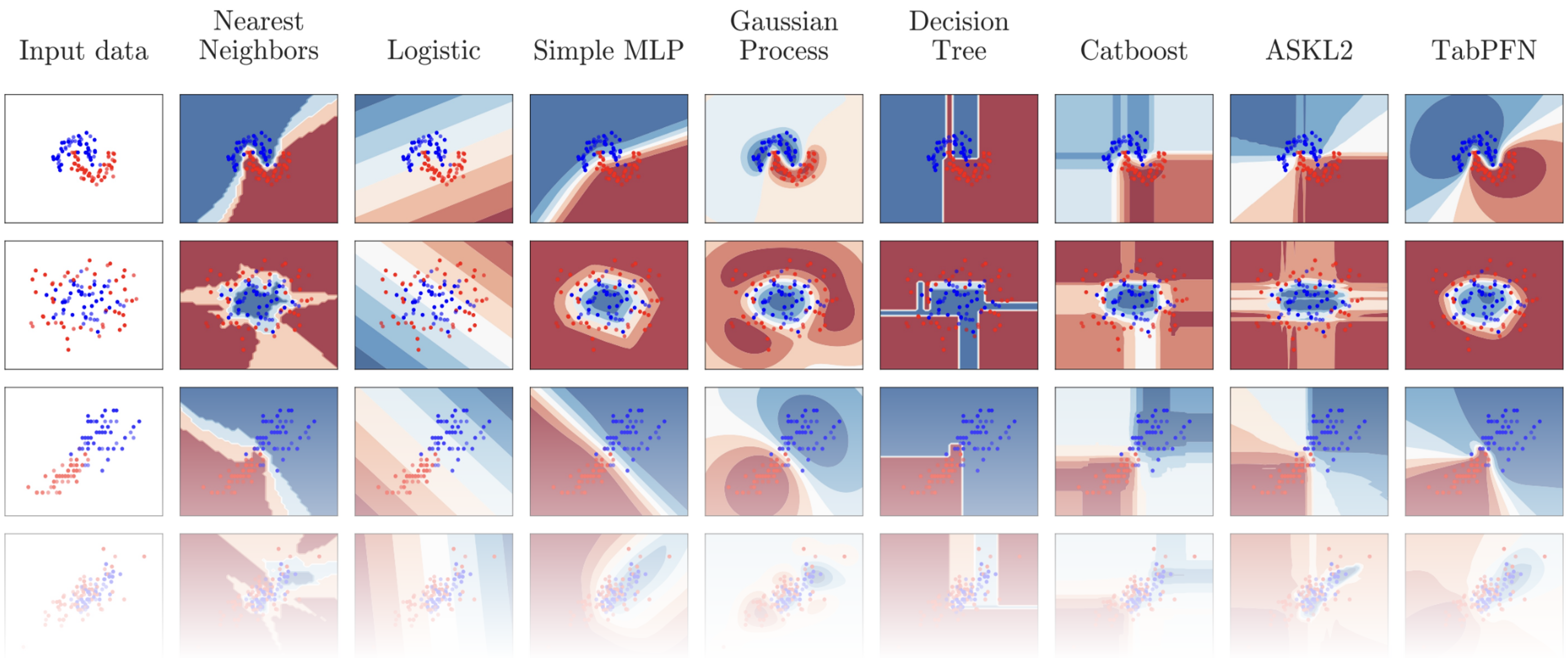


TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second

Noah Hollmann*, Samuel Müller*, Katharina Eggensperger and Frank Hutter

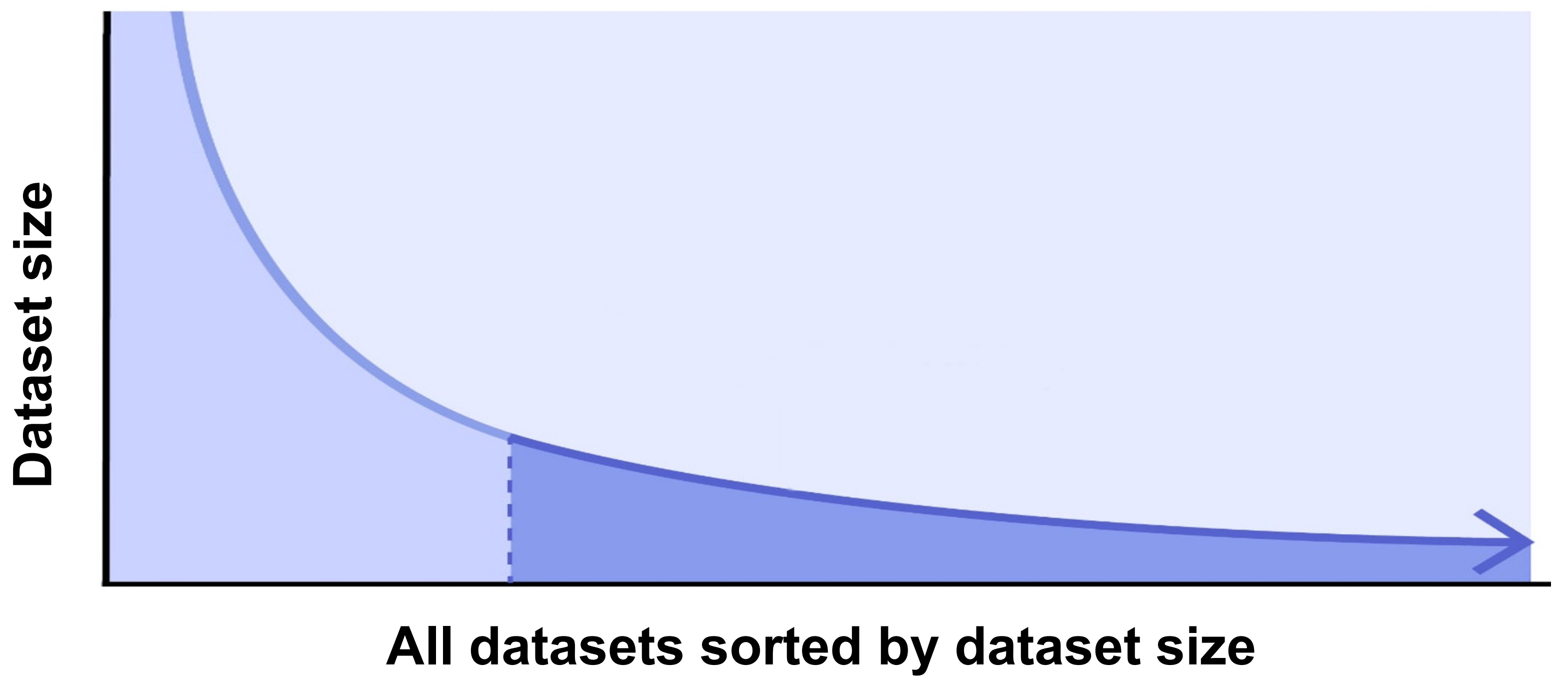
* Equal contribution

Plot based on scitkit-learn example by Gaël Varoquaux & Andreas Müller

Neural Networks for Small Tabular Datasets

Premises

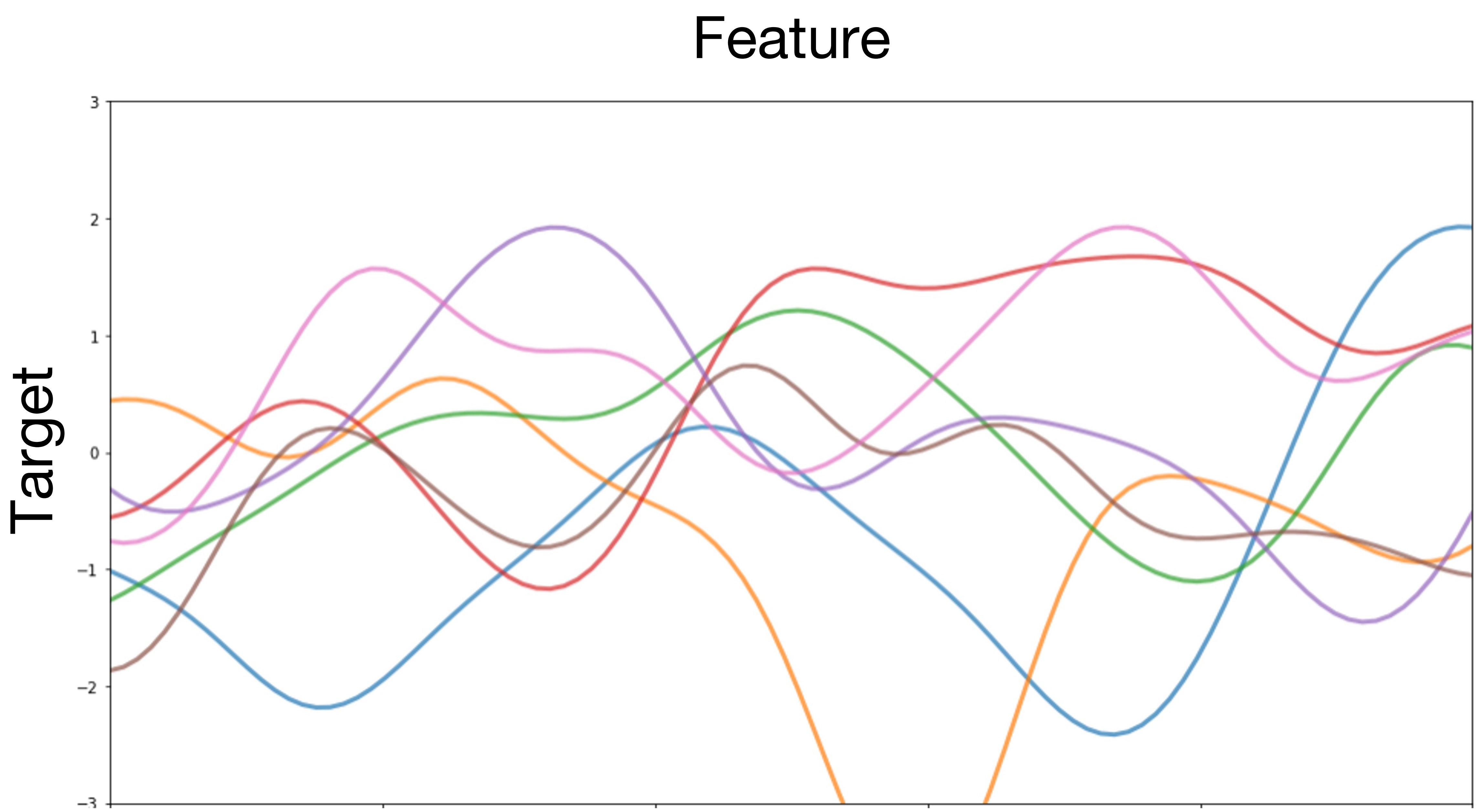
- Neural Networks excel for large amounts of data or given effective inductive biases (such as CNNs for images), but:
  - Overfit on small datasets unless regularized properly (e.g., [1])
  - For tabular data identifying effective inductive biases is challenging
- The long tail of Machine learning datasets: Many real-world datasets are small, but most ML research datasets are large (e.g., [2])

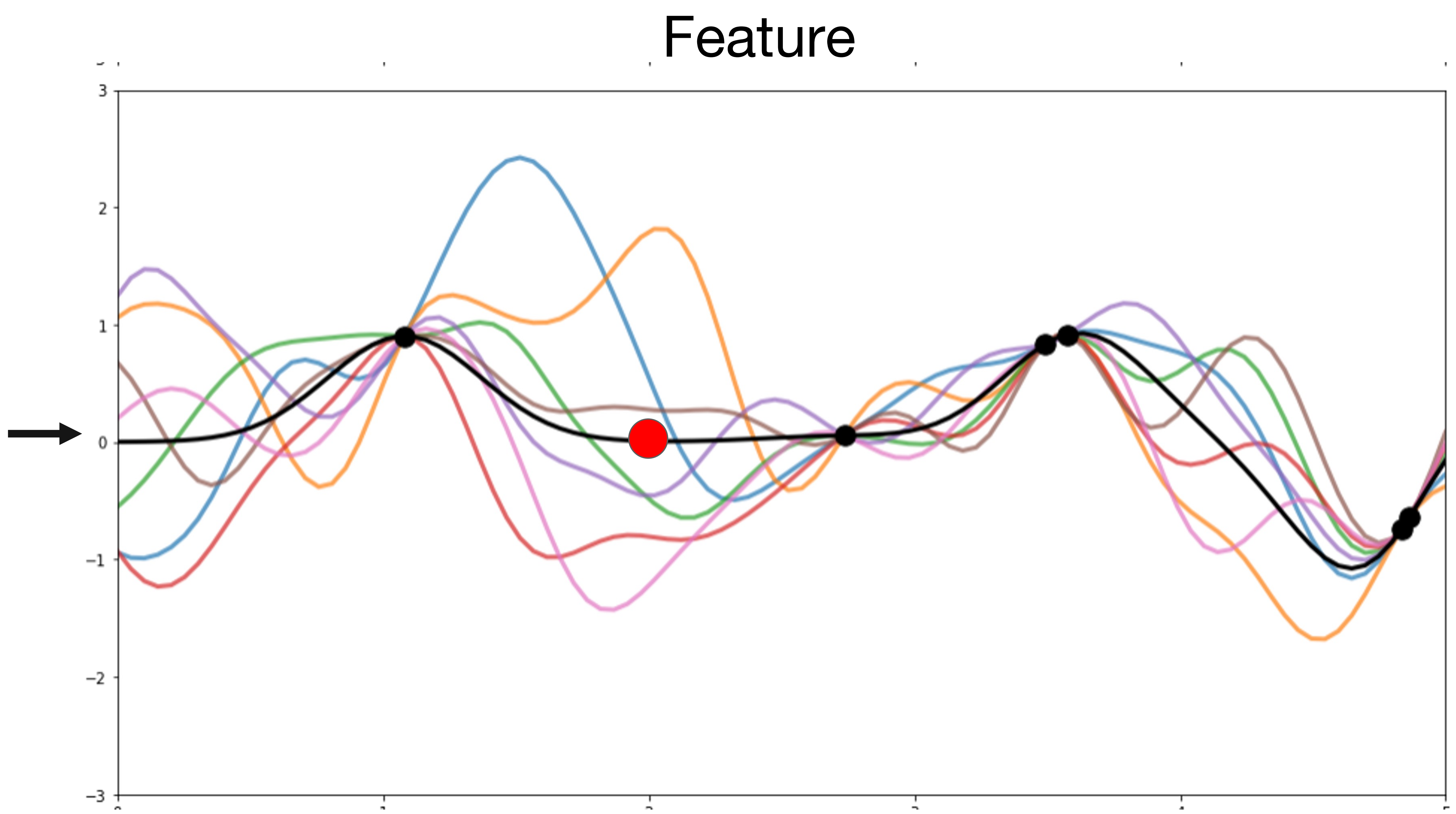


Idea: Model inductive biases and regularize using Bayesian priors.
Bayesian inference is enabled through Prior Fitted Networks.

[1] NeurIPS 2021: Well-tuned Simple Nets Excel on Tabular Datasets
[2] Nature Neuroscience: Big data from small data: data-sharing in the 'long tail' of neuroscience

Bayesian Supervised Learning





Prior $p(t)$:

With latents t: parameterizing functions

Posterior: $$p(t|D) = \frac{p(D|t)p(t)}{\int p(D|t)dt}$$

Posterior predictive distributuion: $$p(y|x, D) = \int p(y|x, t)p(t|D)dt$$

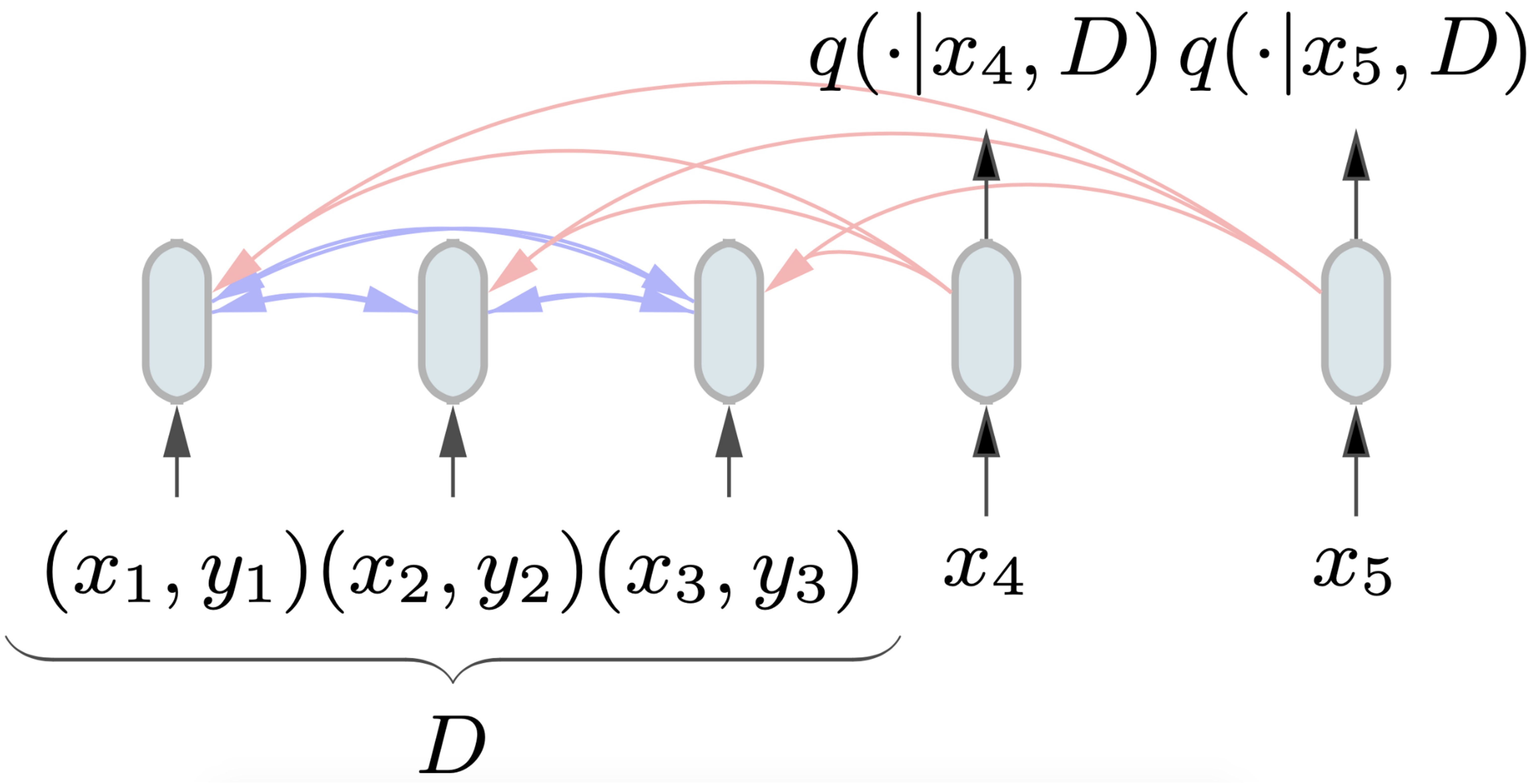
Intractable exactly!

Plot based on scitkit-learn example by
Jan Hendrik Metzen, Guillaume Lemaitre

Prior work

Prior-Data Fitted Networks (PFNs)

A Permutation-Invariant Transformer

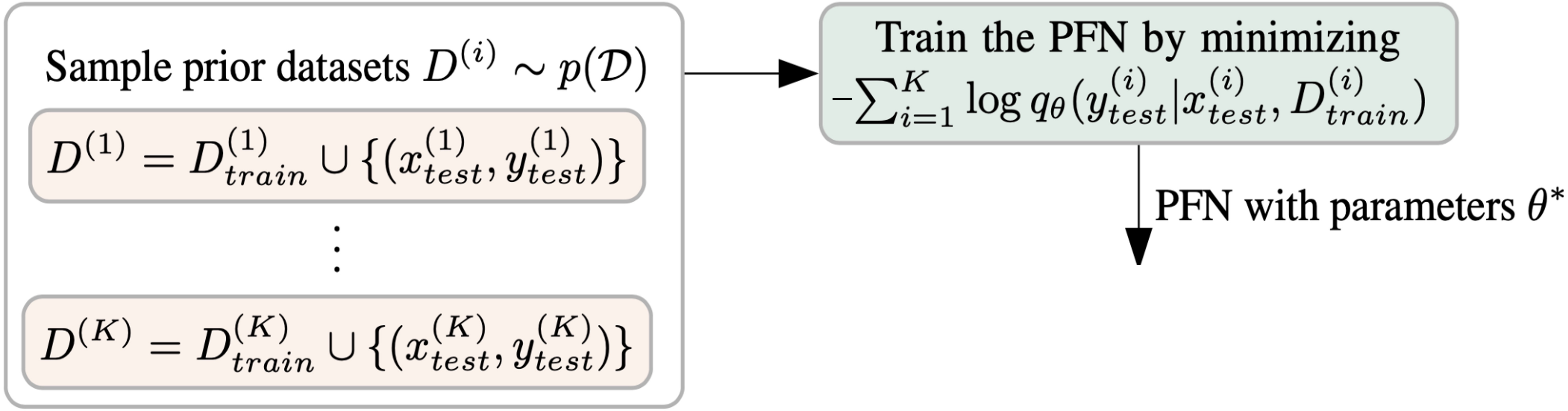


Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. **Transformers Can Do Bayesian Inference**. In International Conference on Learning Representations, 2022.

Prior-Data Fitted Networks (PFNs)

Prior work

Prior-fitting

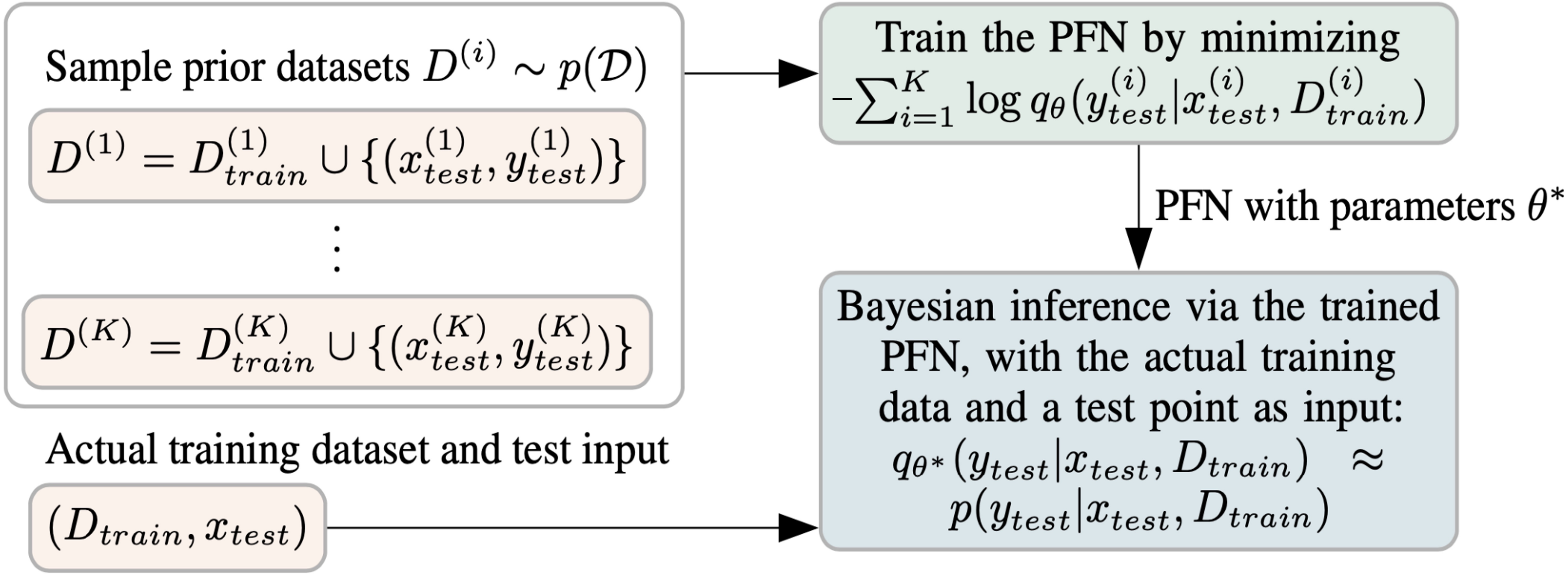


Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. **Transformers Can Do Bayesian Inference**. In International Conference on Learning Representations, 2022.

Prior-Data Fitted Networks (PFNs)

Prior work

Prior-fitting & Inference

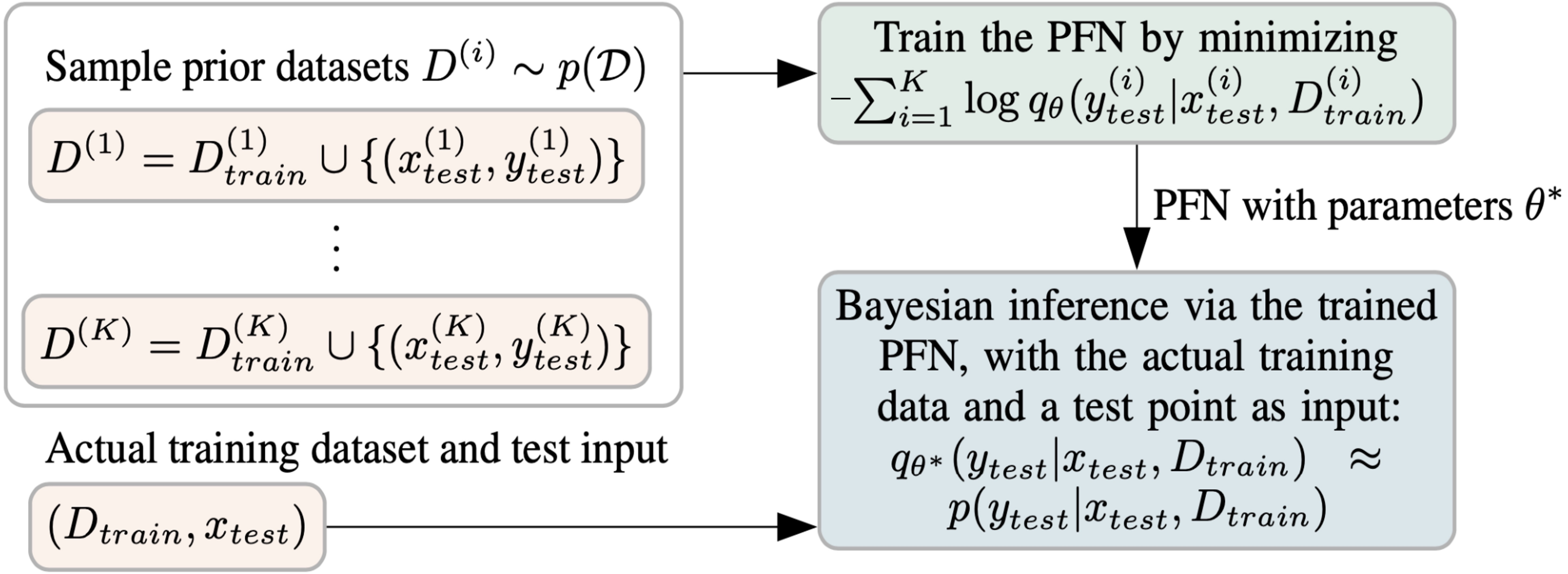


Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. **Transformers Can Do Bayesian Inference**. In International Conference on Learning Representations, 2022.

Prior work

Prior-Data Fitted Networks (PFNs)

Prior-fitting & Inference

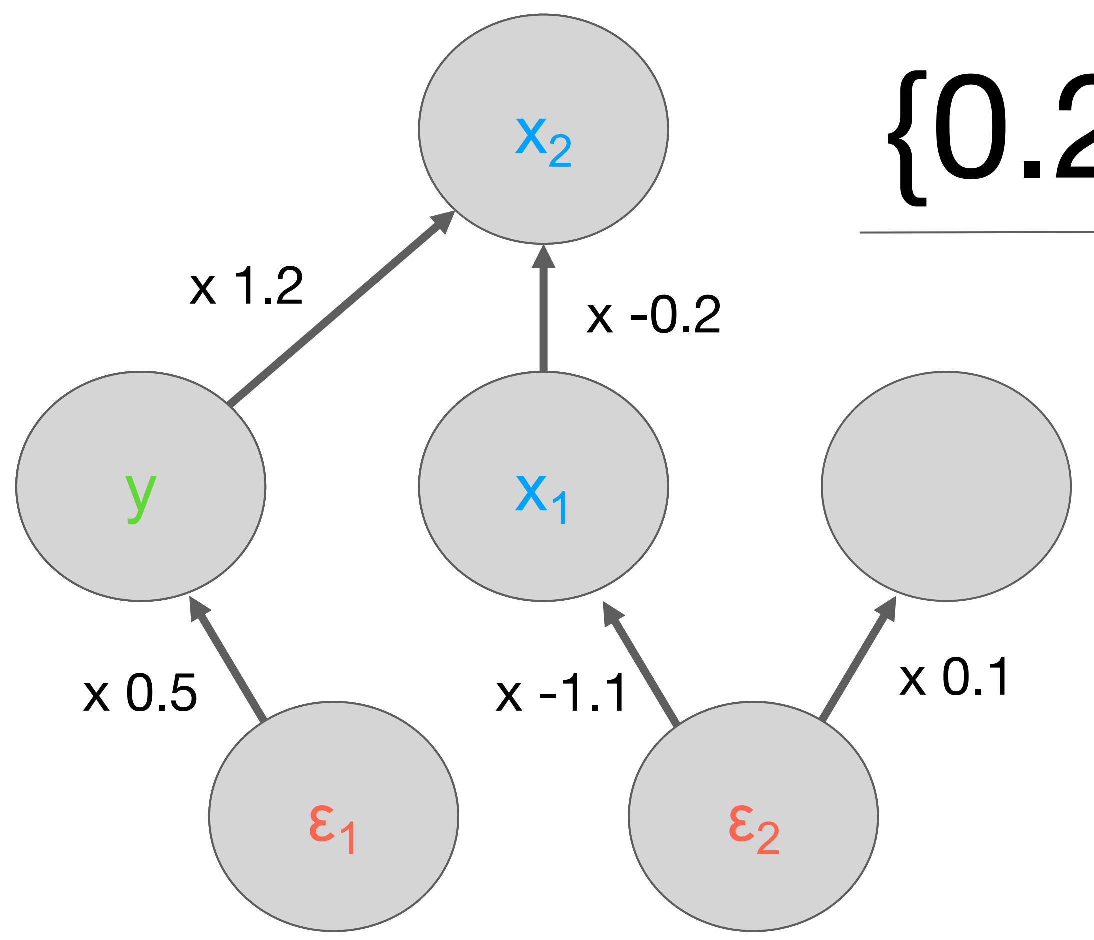


PFNs allow Bayesian inference for any prior we can sample from

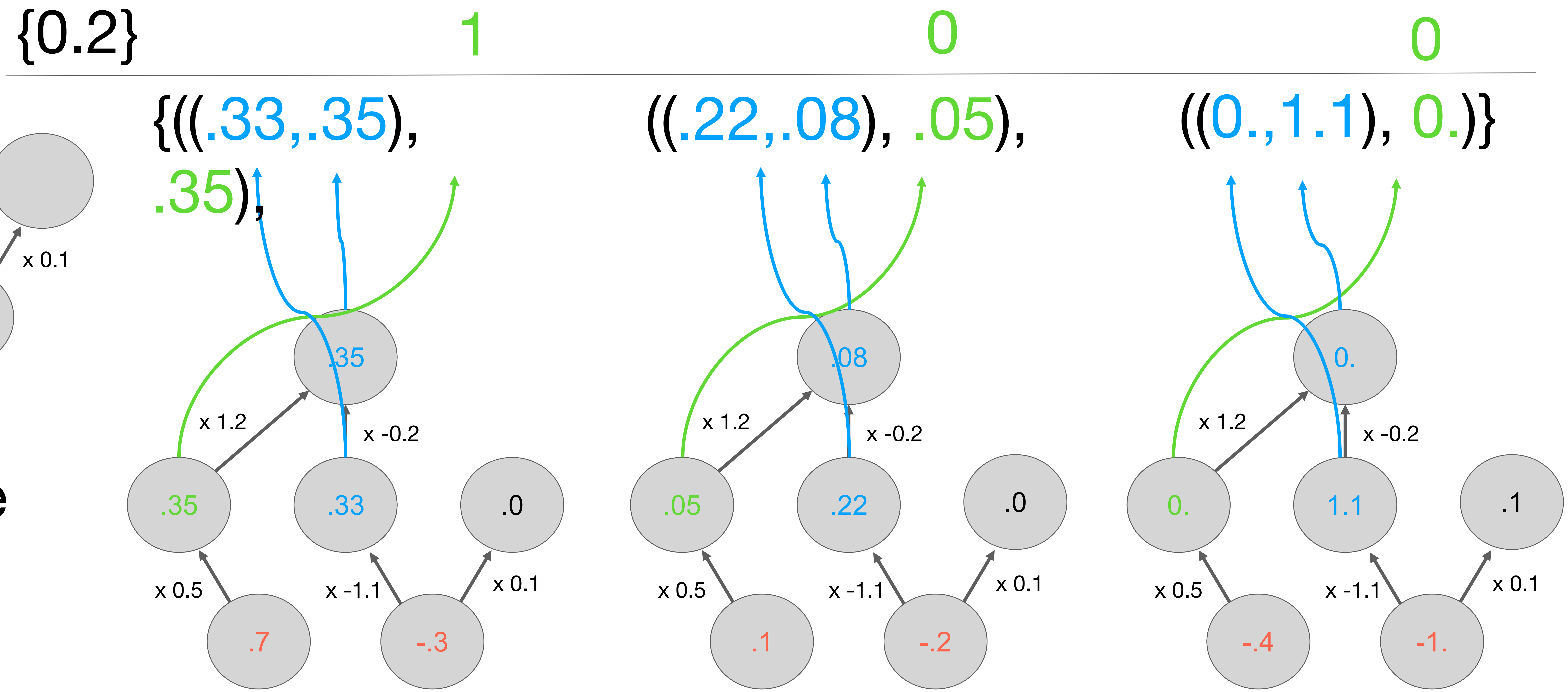
TabPFN Prior: Structural Causal Models

Building a synthetic dataset for training

Initialize:

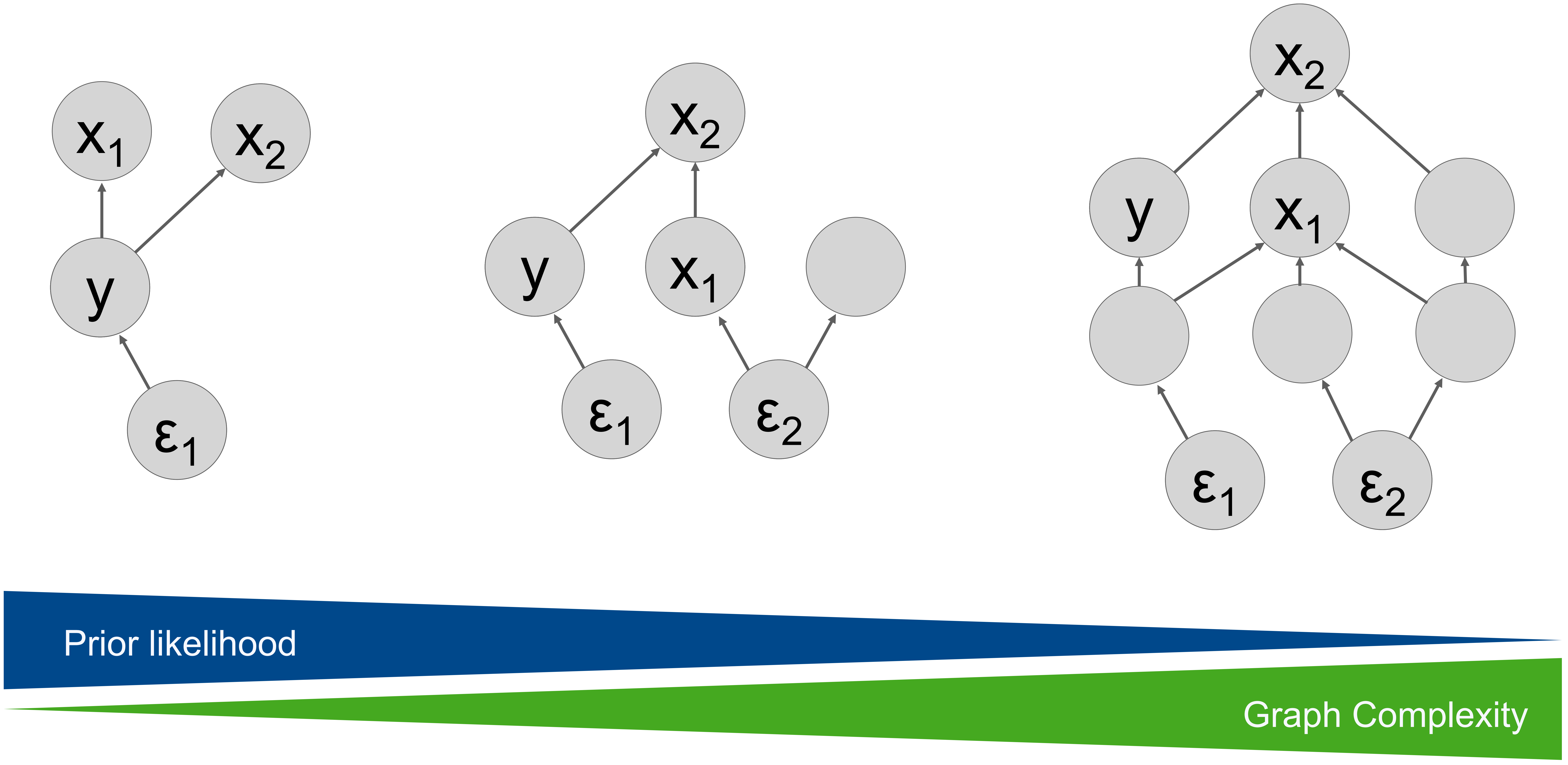


Build dataset:



Sample noise per example:

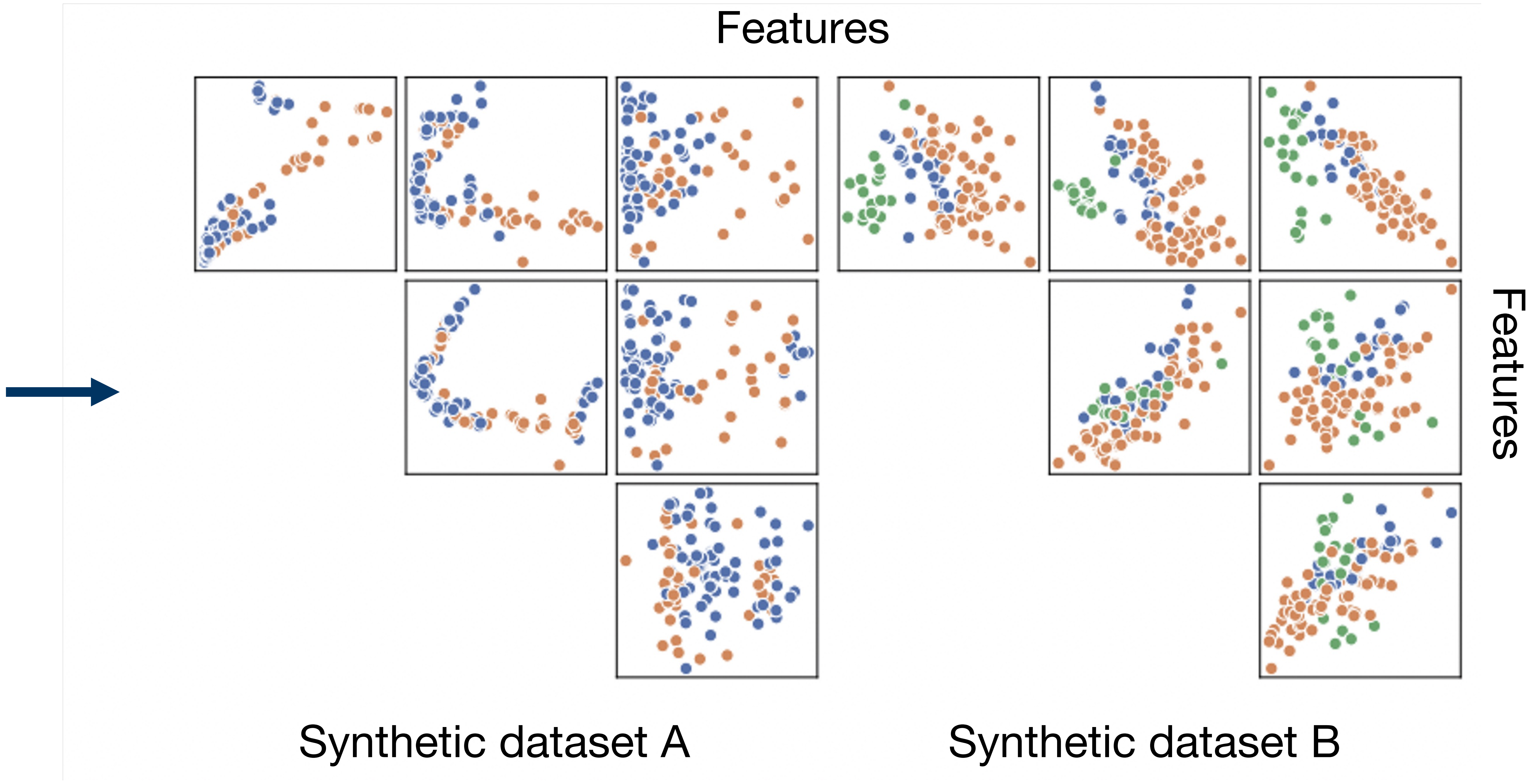
TabPFN Prior: Simplicity Principle



TabPFN Prior: Pior samples

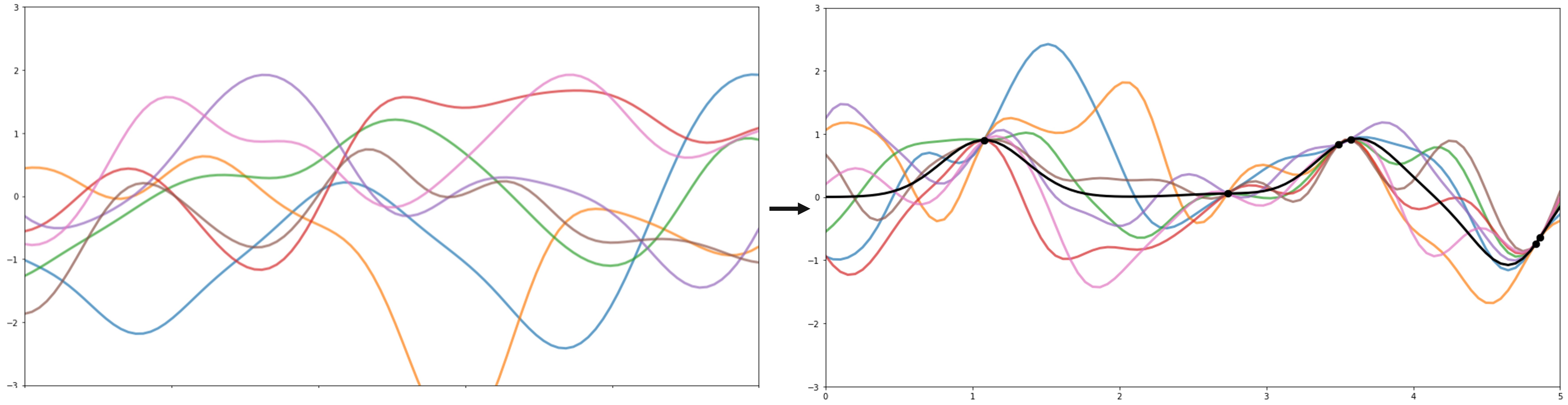
Data for meta-learning is purely synthetic

```
def create_random_scm(graph_nodes):
    for node, i in enumerate(g.nodes):
        g.nodes[node]['size'] = random.randint(...)
        g.nodes[node]['is_feature'] = False

    nodes_per_layer = list(nx.topological_generations(g))
    layers, node_ids_per_layer = [], []
    adj = nx.adjacency_matrix(g)

    for i, _ in enumerate(nodes_per_layer[:-1]):
        ..
```



Colors indicate target classes

Bayesian Supervised Learning



Prior $p(t)$: SCM prior

With latents t : parameterizing SCMs,
i.e. **weights, graph structure, activation function, etc**

Posterior: $$p(t|D) = \frac{p(D|t)p(t)}{\int p(D|t)dt}$$

Posterior predictive distribution: $$p(y|x,D) = \int p(y|x,t)p(t|D)dt$$

Plot based on scitkit-learn example by
Jan Hendrik Metzen, Guillaume Lemaitre

Simplifying AutoML for Real-time Training

AutoML pipeline

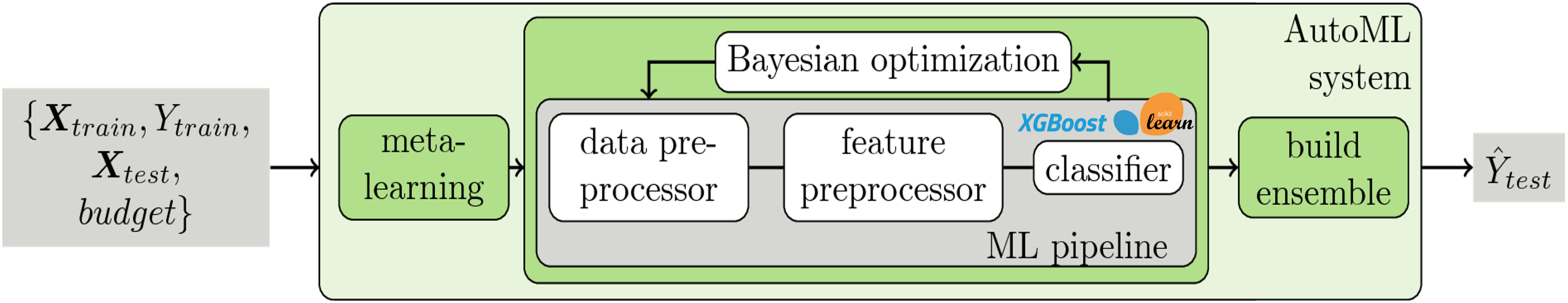
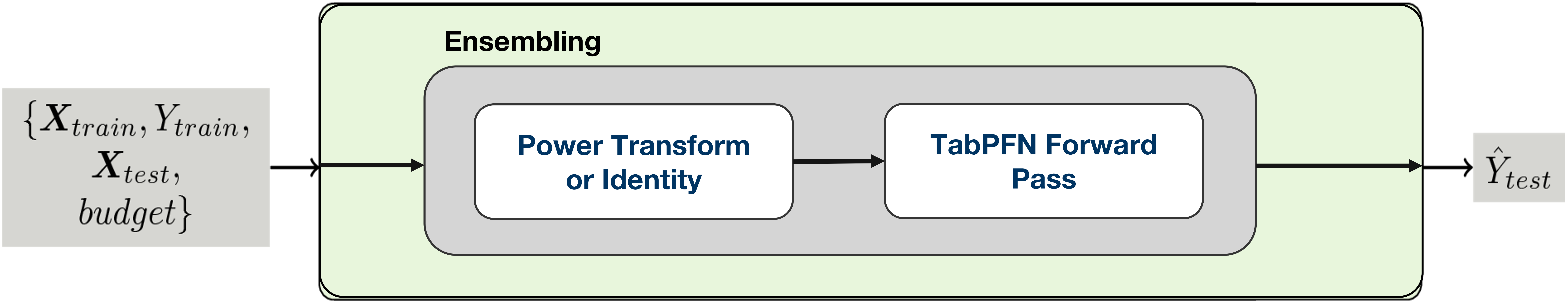


Image credit Biedenkapp et al. (https://www.automl.org/blog-2nd-automl-challenge/)

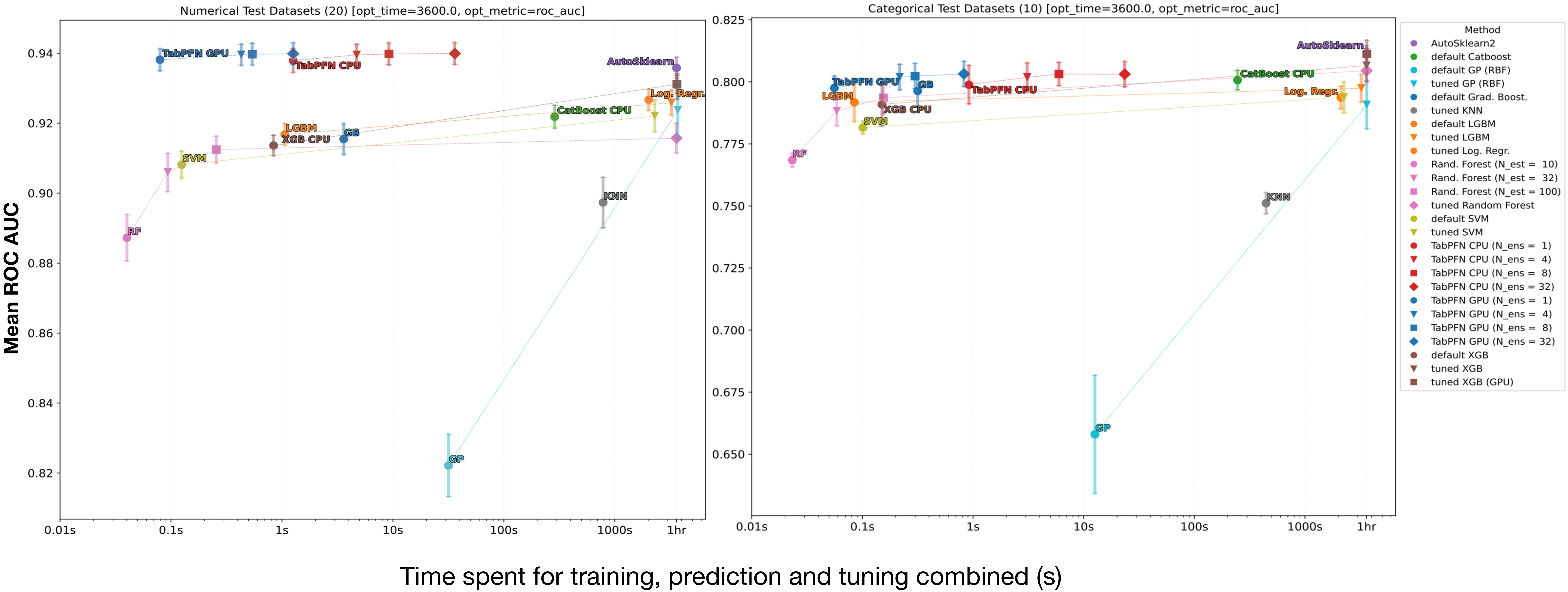
Simplifying AutoML for Real-time Training

TabPFN pipeline



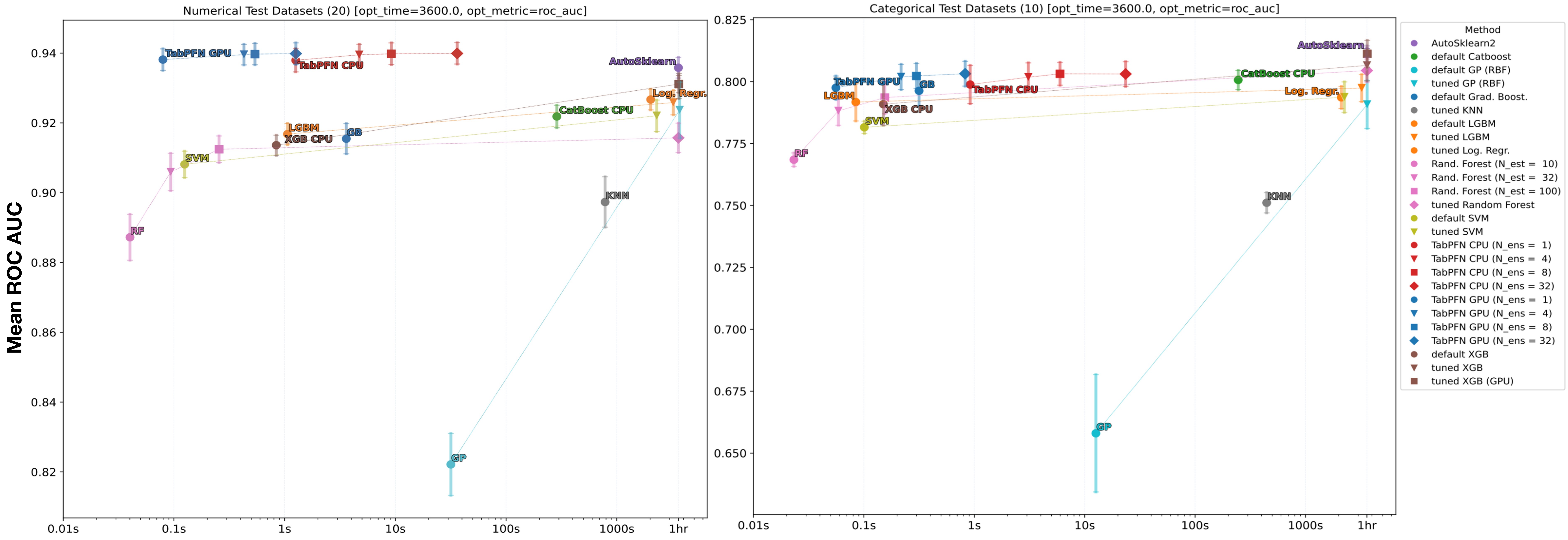
The TabPFN

Results on OpenML-CC18 suite subset with < 1000 examples



The TabPFN

Results on OpenML-CC18 suite subset with < 1000 examples



Additional evaluation on 149 validation datasets in our work!

Future Work

Improving predictions: Categorical/Missing data

Larger applicability: Generalizations to non-tabular data and regression tasks, architectural changes to remove quadratic memory scaling

Application of TabPFN: Instantaneous tabular predictions for small datasets in production: exploratory data analysis, novel feature engineering, active learning

/automl/TabPFN

pypi /tabpfn

colab

Conclusions

/automl/TabPFN

pypi /tabpfn

colab

TabPFN is fully learned: We do not specify the learning algorithm, but only prior assumptions from which the PFN learns a general prediction method

Strong performance on small datasets: 1000 training samples, 100 features, 10 classes; stronger results for continuous data

Training and prediction in < 1s: TabPFN can be parallelized on GPU using standard matrix operations

We thank Eddie Bergman, Ravin Kohli, Matthias Feurer, Robert Bosch GmbH