universität freiburg

* Private Repo, feel free to send access request!

HPO4TabPFN - Scaling Down TabPFN

Vladyslav Moroshan | Nastaran Alipour | Jude Mingay

University of Freiburg

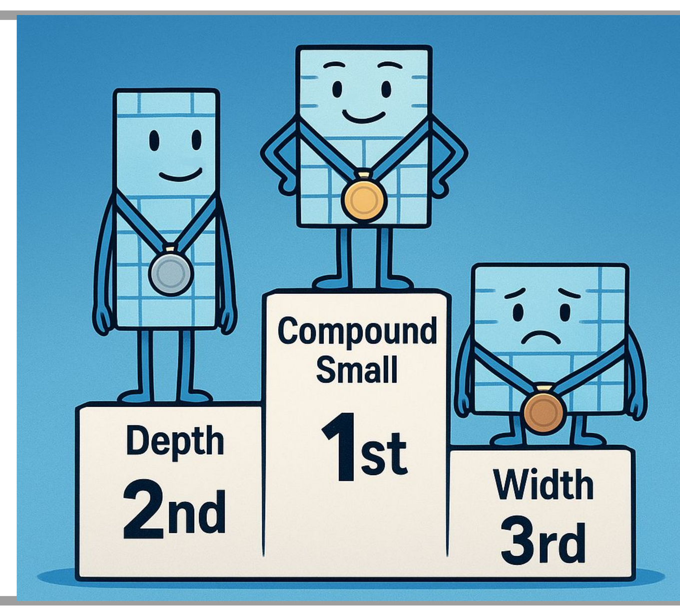
Supervisor: Johannes Hog

Introduction

How do you scale a TabPFN? What hyperparameters matter most?

This study explores the relationship between model size and optimal training configurations for **NanoTabPFN**, a lightweight proxy for TabPFN[1]. Using **random search** with no prior assumptions, we analyze how scaling affects both performance and hyperparameter sensitivity. The evaluation is on **TabArena**[2] benchmark.

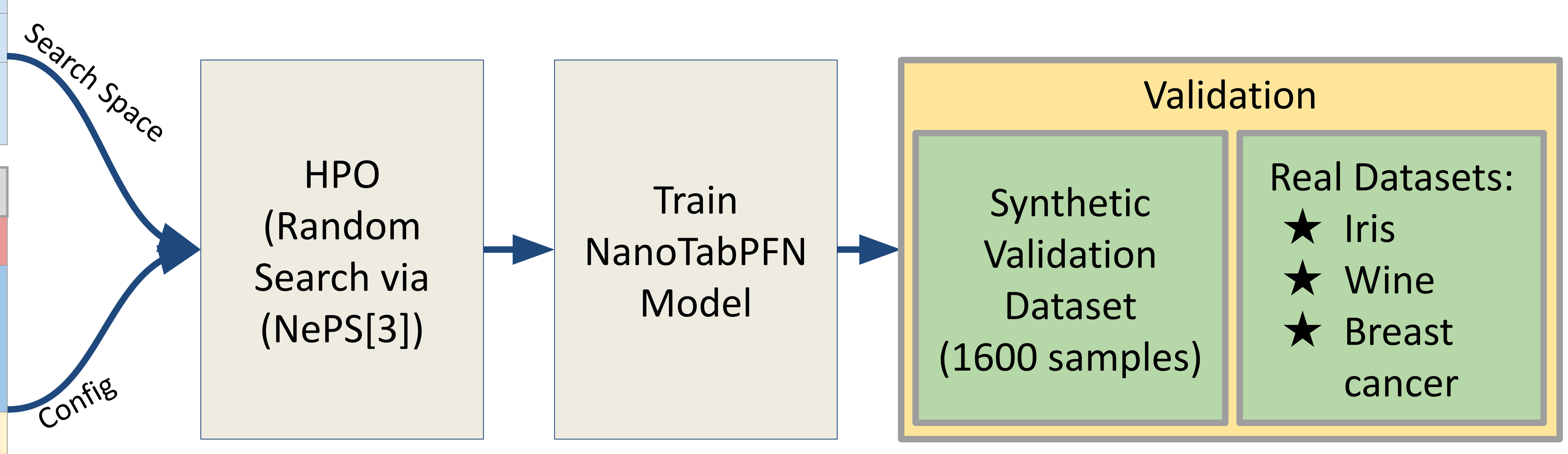
Our results reveal that smaller and deeper models can outperform shallower and larger configurations.



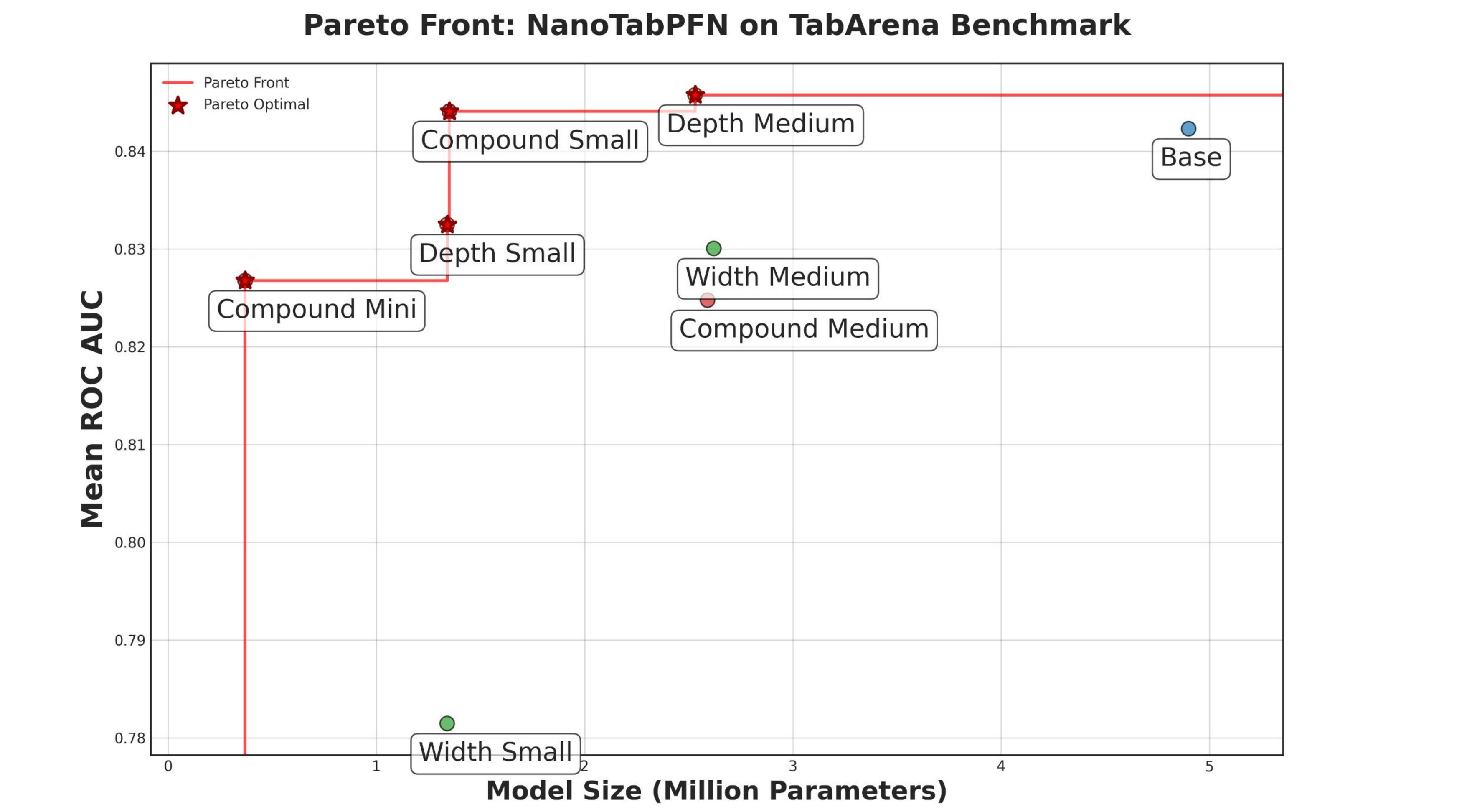
Methodology

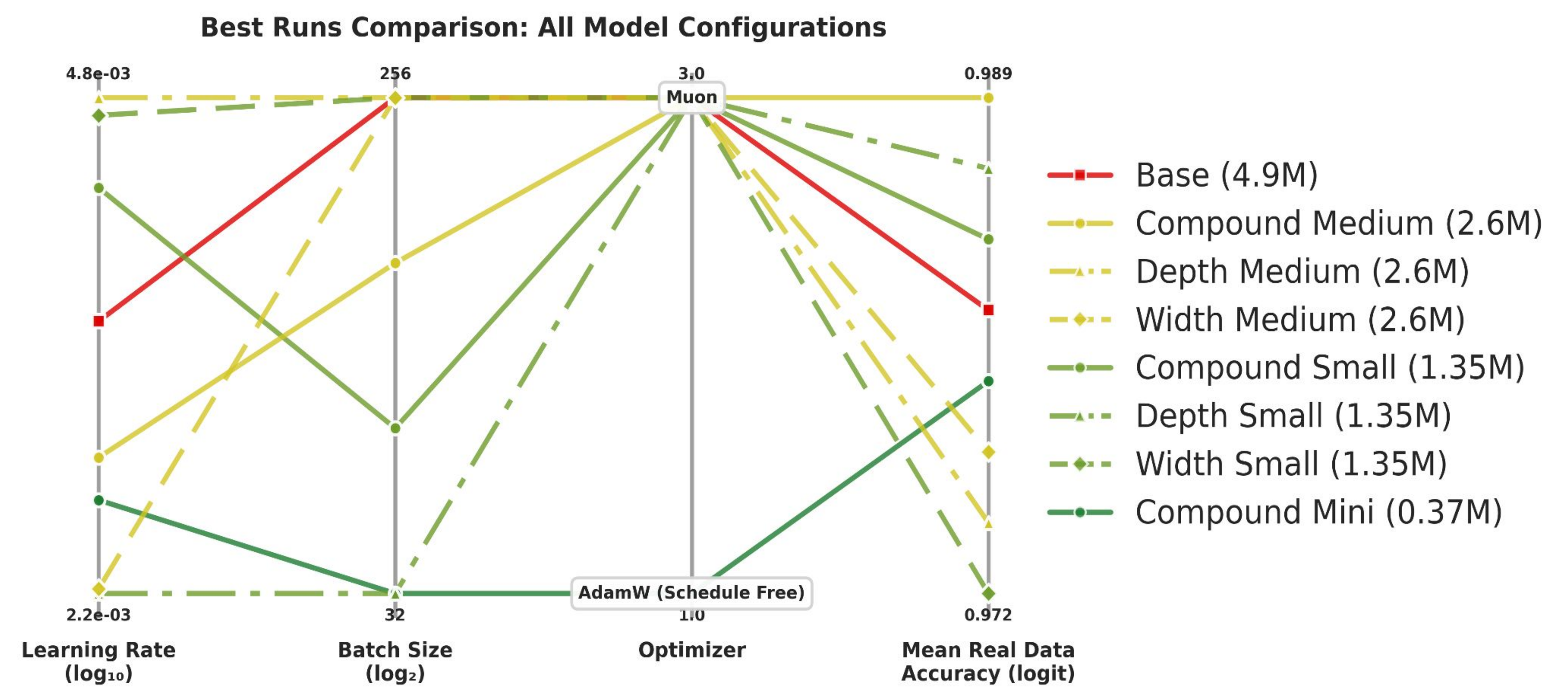
Parameter	Range	Type
Learning Rate	[0.00001, 0.01]	Log Scale
Batch Size	[32, 64, 128, 256]	Categorical
Optimizer	[AdamW, Muon[4], Lion[5], AdamW Schedule Free]	Categorical

Scale	Name	Layers	Embedding	MLP hidden	Params
Big	Base	8	192	768	4.9 M
Medium	Width	4	192	768	2.6 M
	Depth	8	140	560	
	Compound	6	160	640	
Small	Width	2	192	768	1.35 M
	Depth	8	100	400	
	Compound	4	140	560	
Mini	Compound	6	64	192	0.37 M



Results





SHAP Hyperparameter Importance

Hyperparameter	Base	Width Small	Width Medium	Depth Small	Depth Medium	Compound Small	Compound Medium	Compound Mini	Average
LR^2	47.6	51.0	48.9	37.8	47.6	50.4	43.0	45.1	46.4
Opt	37.2	37.4	38.8	40.2	31.4	36.9	38.2	32.4	36.6
BS^2	15.2	11.6	12.3	22.0	21.1	12.7	18.8	22.5	17.0

Model Configuration

Conclusions

- **Compound Small** (1.35 M) outperformed larger models.
- **Muon**[3] is the best optimizer.
- **Deeper models** slightly outperform wider ones.
- **Batch Size Correlates with Scale**
- **Learning rate** and **model scale** are the most critical hyperparameters.
- **Smaller models** benefit from **higher learning rates**.

References

[1] Noah Hollmann et al. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second, arXiv:2207.01848

[2] TabArena: A Living Benchmark for Machine Learning on Tabular Data arXiv:2506.16791

[3] Stoll, D., Mallik, N., Schrodi, S., Janowski, M., Garibov, S., Abou Chakra, T., Rogalla, D., Bergman, E., Hvarfner, C., Binxin, R., Kober, N., Vallaeys, T., & Hutter, F. (2023). Neural Pipeline Search (NePS) https://github.com/automl/neps

[4] Keller Jordan et al., Muon: An optimizer for hidden layers in neural networks, 2024.

[5] Symbolic Discovery of Optimization Algorithms, arXiv:2302.06675