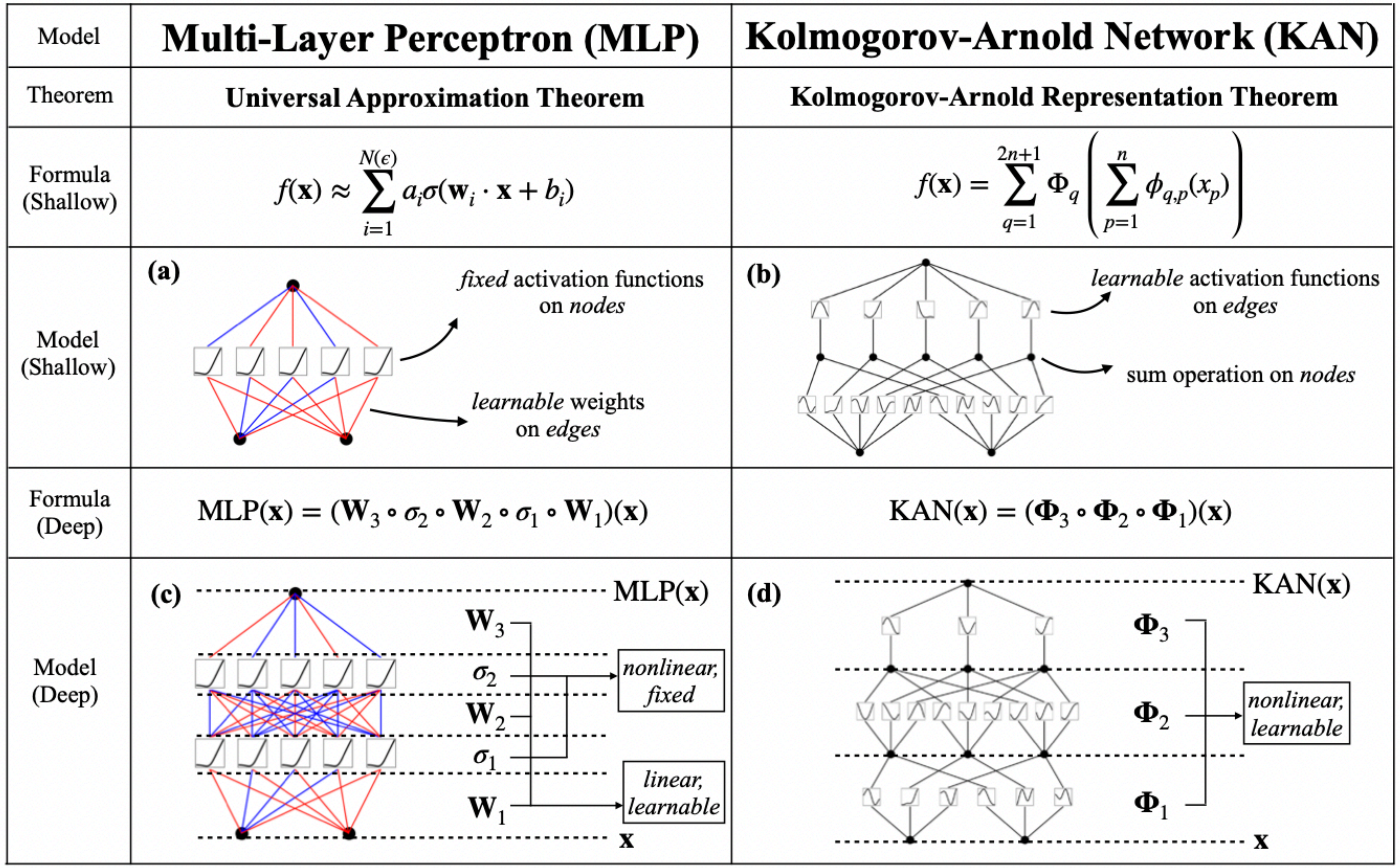
KANs meet Tabular DL

Георгий Булгаков, 3 курс ФПМИ
Данил Руденко, 3 курс ФПМИ

Научные руководители:
Глеб Молодцов и Даниил Медяков, МФТИ и BRAIn LAB

KAN’s Basics

Idea: learnable activations!



Basic figure from [1]

Core Theorem

Kolmogorov-Arnold theorem, [2]

Theorem 1 (Representation of continuous functions on a compact set). *Let $f\colon [0,1]^n \to \mathbb{R}$ be a continuous function of n variables defined on the unit cube. Then f can be represented in the form:*

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \Phi_{q,p}(x_p) \right),$$

where the functions $\Phi_{q,p}\colon [0,1] \to \mathbb{R}$ and $\Phi_q\colon \mathbb{R} \to \mathbb{R}$ are continuous for all $p = 1, \ldots, n$ and $q = 1, \ldots, 2n+1$.

Our Research

Why Tabular DL?

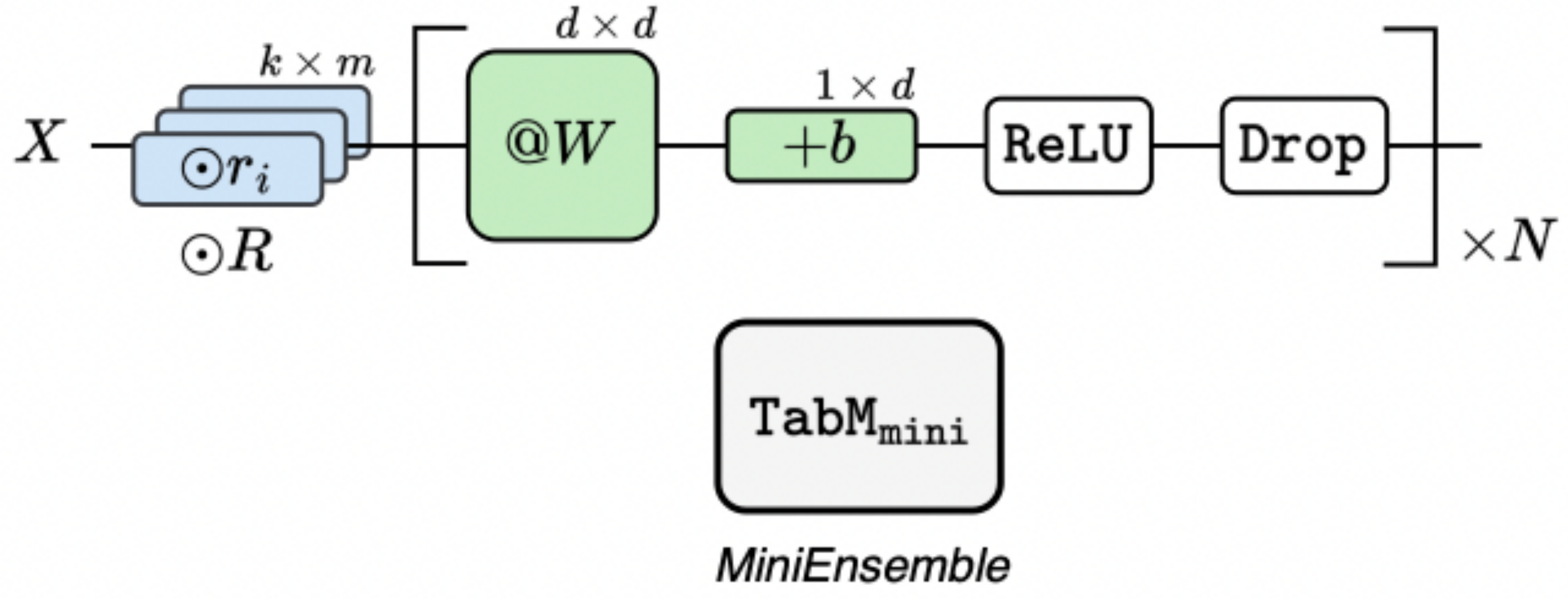
✳DL models began to outperform GBDT (CatBoost, XGBoost, LightGBM);

✳MLP-based models perform great ([3], [4]);

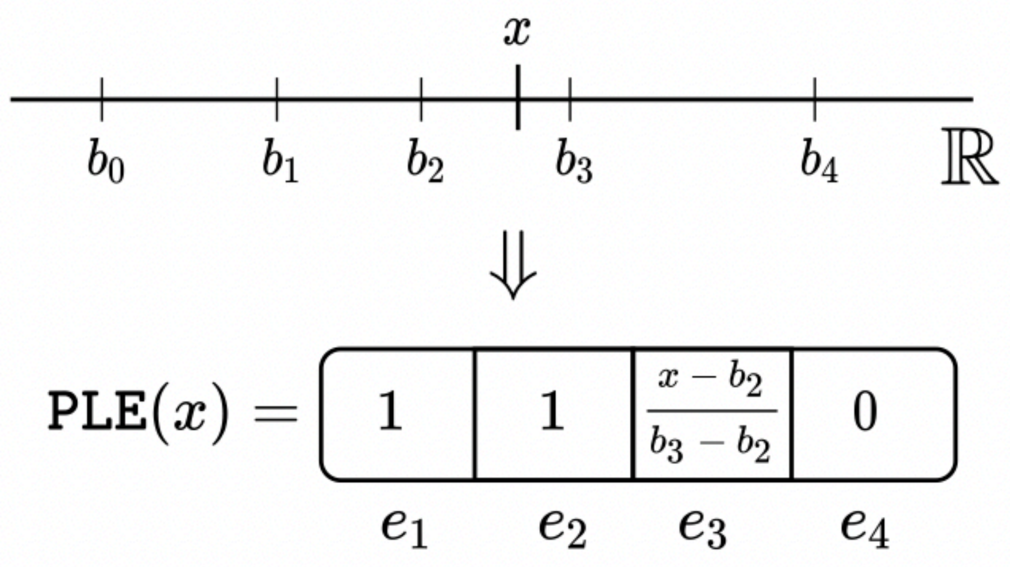
✳Tabular data often is low dimensional;

✳So, **KAN can surpass MLP!**

CatBoost



$\mathrm{TabM}_{\mathrm{mini}}$, [3]



Piecewise Linear Encoding, [4]

What’s done?

Experiments and progress

✓ **<u>Built the pipeline for our experiments (…);</u>**

✓ BatchNorm and Dropout in KAN;

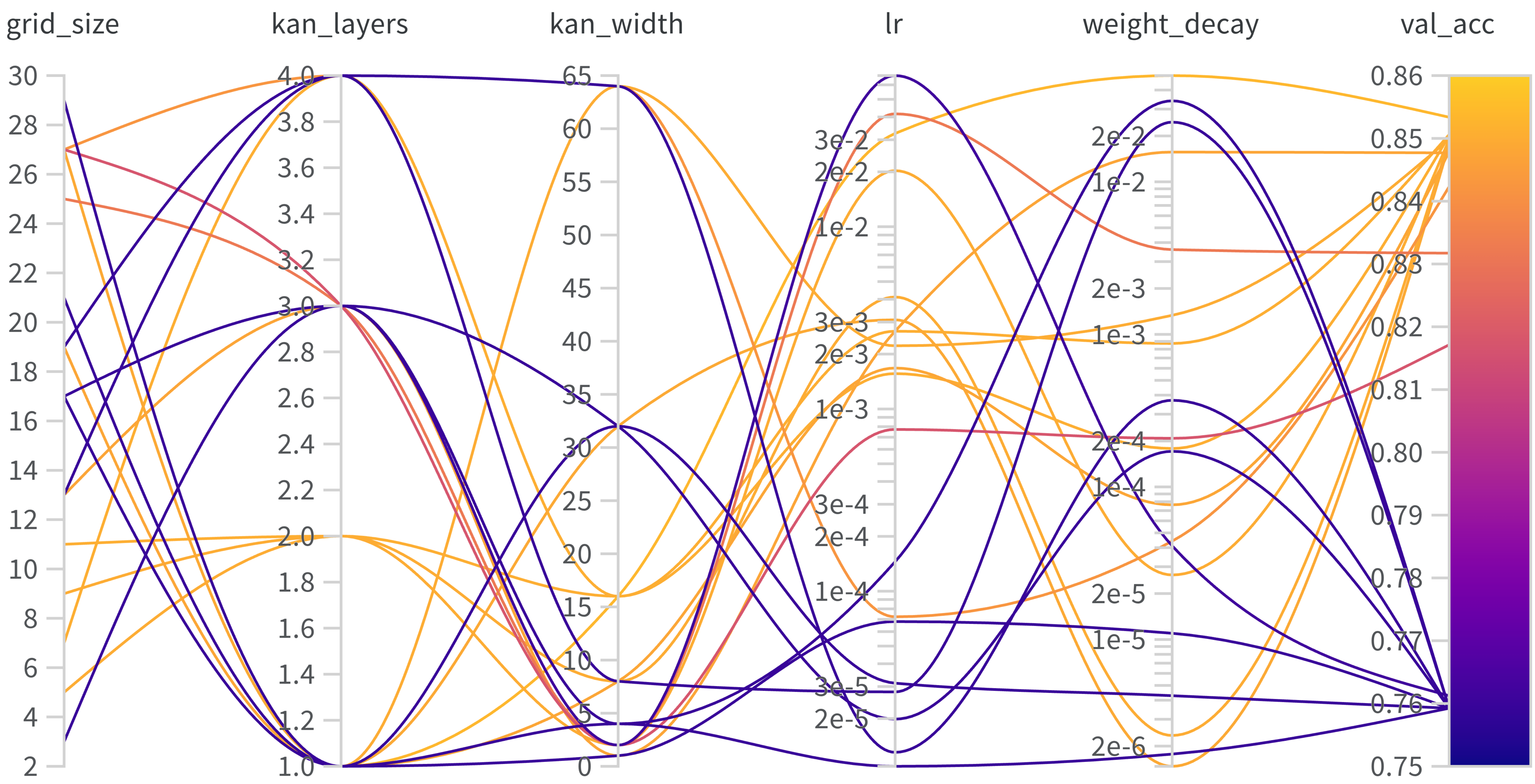
✓ Different KAN-based approaches (KAN [1], efficientKAN [5], ChebyKAN [6], FastKAN [7]);

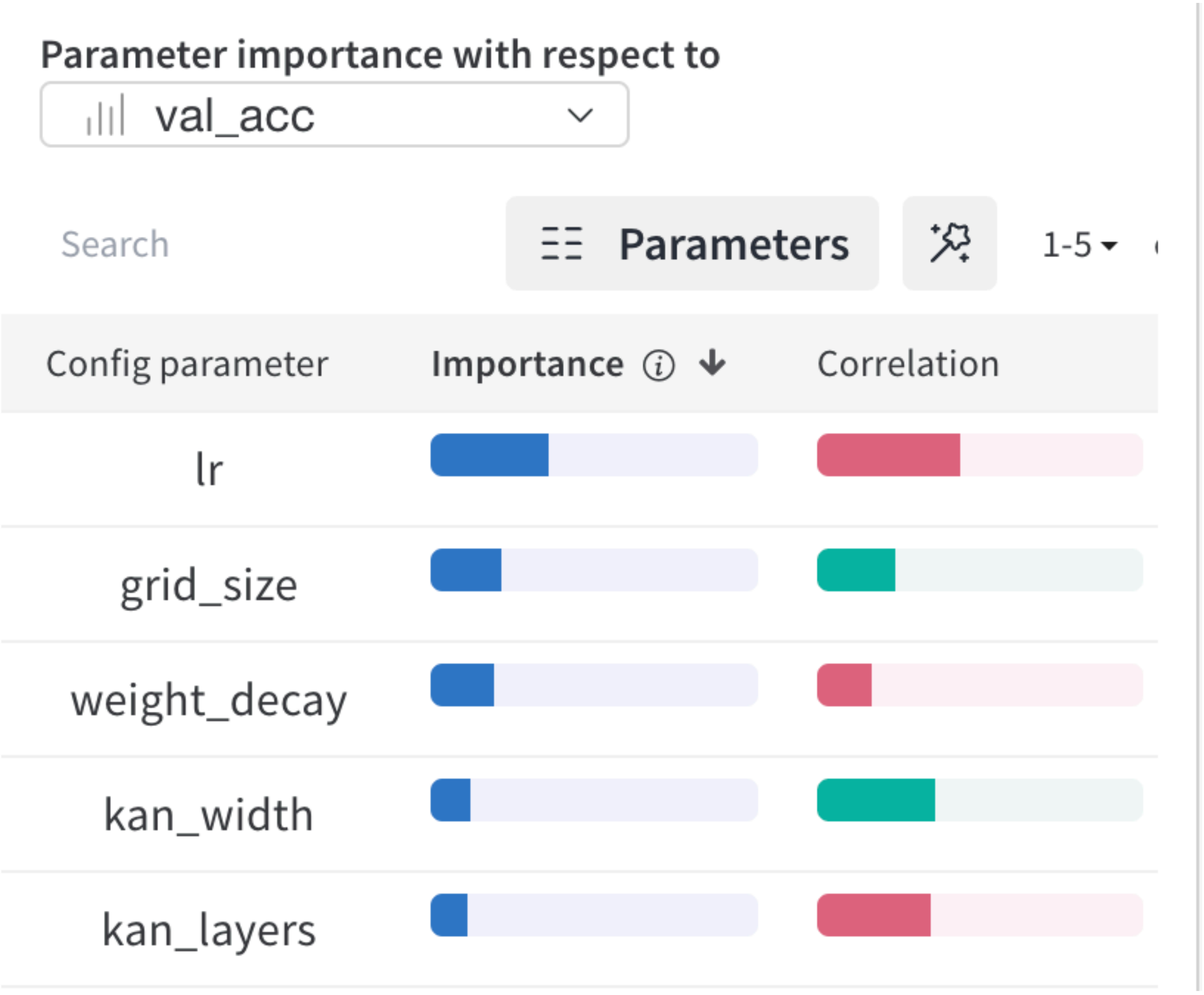
✓ KAN with Piecewise Linear Embedding and Periodic Embedding ([3]);

◉ KAN combined with MLP

◉ Various optimizers (AdamW [8], AdEMAmix [9], Muon [10], MARS [11])

Hyperparameters





Example: tuning KAN on Adult dataset

BatchNorm mystery

First series

Model	adult ↑	gesture ↑	california ↓ (MSI	churn ↑	eye ↑
BatchNorm	**0,843 +- 0,003**	**0,561 +- 0,008**	**0,654 +- 0,045**	**0,854 +- 0,004**	**0,592 +- 0,008**
Dropout 0,5	0,808 +- 0,004	0,487 +- 0,003	0,897 +- 0,070	0,796 +- 0,002	0,457 +- 0,008
Vanila	0,811 +- 0,005	0,489 +- 0,003	0,893 +- 0,075	0,796 +- 0,002	0,476 +- 0,009

Second series

Model	adult ↑	gesture ↑	california ↓	churn ↑	house ↓
KAN	**0,854 +- 0,001**	0,539 +- 0,005	**0,427 +- 0,006**	0,854 +- 0,004	0,696 +- 0,005
KAN PLR	**0,869 +- 0,001**	0,557 +- 0,006	0,409 +- 0,004	0,856 +- 0,001	0,655 +- 0,006
KAN PLE-Q	**0,859 +- 0,001**	0,571 +- 0,010	**0,395 +- 0,003**	0,850 +- 0,011	**0,626 +- 0,004**
BatchNorm KAN	0,845 +- 0,002	**0,549 +- 0,005**	0,517 +- 0,010	**0,860 +- 0,004**	**0,688 +- 0,028**
BatchNorm KAN PLR	0,867 +- 0,001	**0,564 +- 0,005**	**0,408 +- 0,004**	**0,858 +- 0,002**	**0,654 +- 0,012**
BatchNorm KAN PLE	0,853 +- 0,001	**0,586 +- 0,004**	0,396 +- 0,006	**0,862 +- 0,002**	0,627 +- 0,005

Explanation: Data preprocessing

BatchNorm & Dropout conclusion

❌ **Dropout** is **not** needed

❓ **BatchNorm** is **not always** useful.

Model	adult ↑	gesture ↑	california ↓	churn ↑	house ↓
KAN	**0,854 +- 0,001**	0,539 +- 0,005	**0,427 +- 0,006**	0,854 +- 0,004	0,696 +- 0,005
KAN PLR	**0,869 +- 0,001**	0,557 +- 0,006	0,409 +- 0,004	0,856 +- 0,001	0,655 +- 0,006
KAN PLE-Q	**0,859 +- 0,001**	0,571 +- 0,010	**0,395 +- 0,003**	0,850 +- 0,011	**0,626 +- 0,004**
BatchNorm KAN	0,845 +- 0,002	**0,549 +- 0,005**	0,517 +- 0,010	**0,860 +- 0,004**	**0,688 +- 0,028**
BatchNorm KAN PLR	0,867 +- 0,001	**0,564 +- 0,005**	**0,408 +- 0,004**	**0,858 +- 0,002**	**0,654 +- 0,012**
BatchNorm KAN PLE-	0,853 +- 0,001	**0,586 +- 0,004**	0,396 +- 0,006	**0,862 +- 0,002**	0,627 +- 0,005

Different KANs

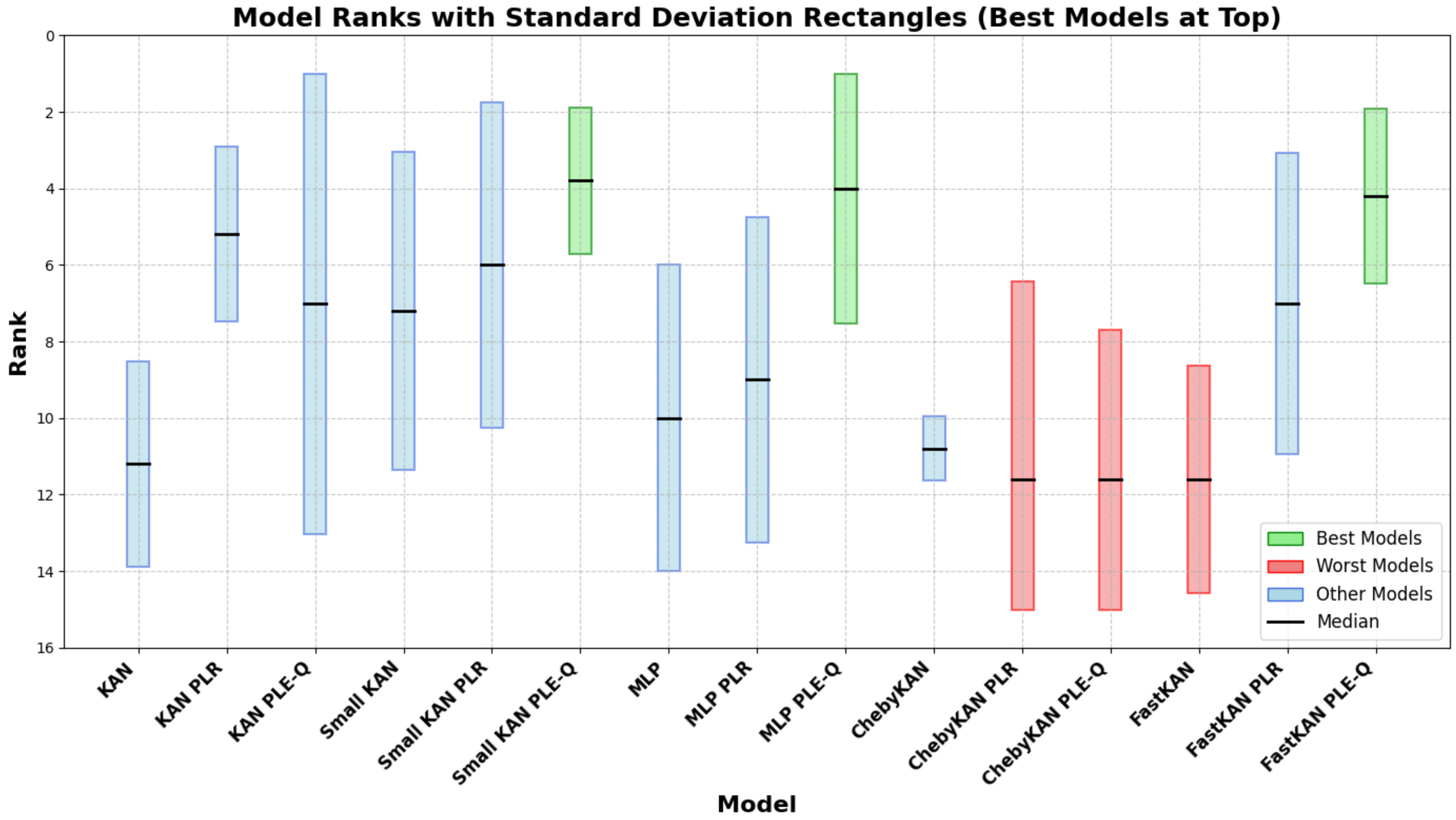
Which base function is the best?

Model	Function	Rank	Train Time Ratio	Test Time Ratio
KAN	**B-splines**	**2,8 ± 1,3**	2,73 ± 0,82	3,81 ± 0,94
Small KAN	**B-splines**	**1,8 ± 1,3**	2,6 ± 0,62	3,65 ± 0,61
MLP	**--**	3,0 ± 1,87	**1 ± 0**	**1 ± 0**
ChebyKAN	**Chebyshev Polinomials**	3,2 ± 1.3	1,39 ± 0,28	1,55 ± 0.27
FastKAN	**RBF**	4 ± 1	1,72 ± 0,48	2,02 ± 0.54

Our comparison of different models on 5 datasets

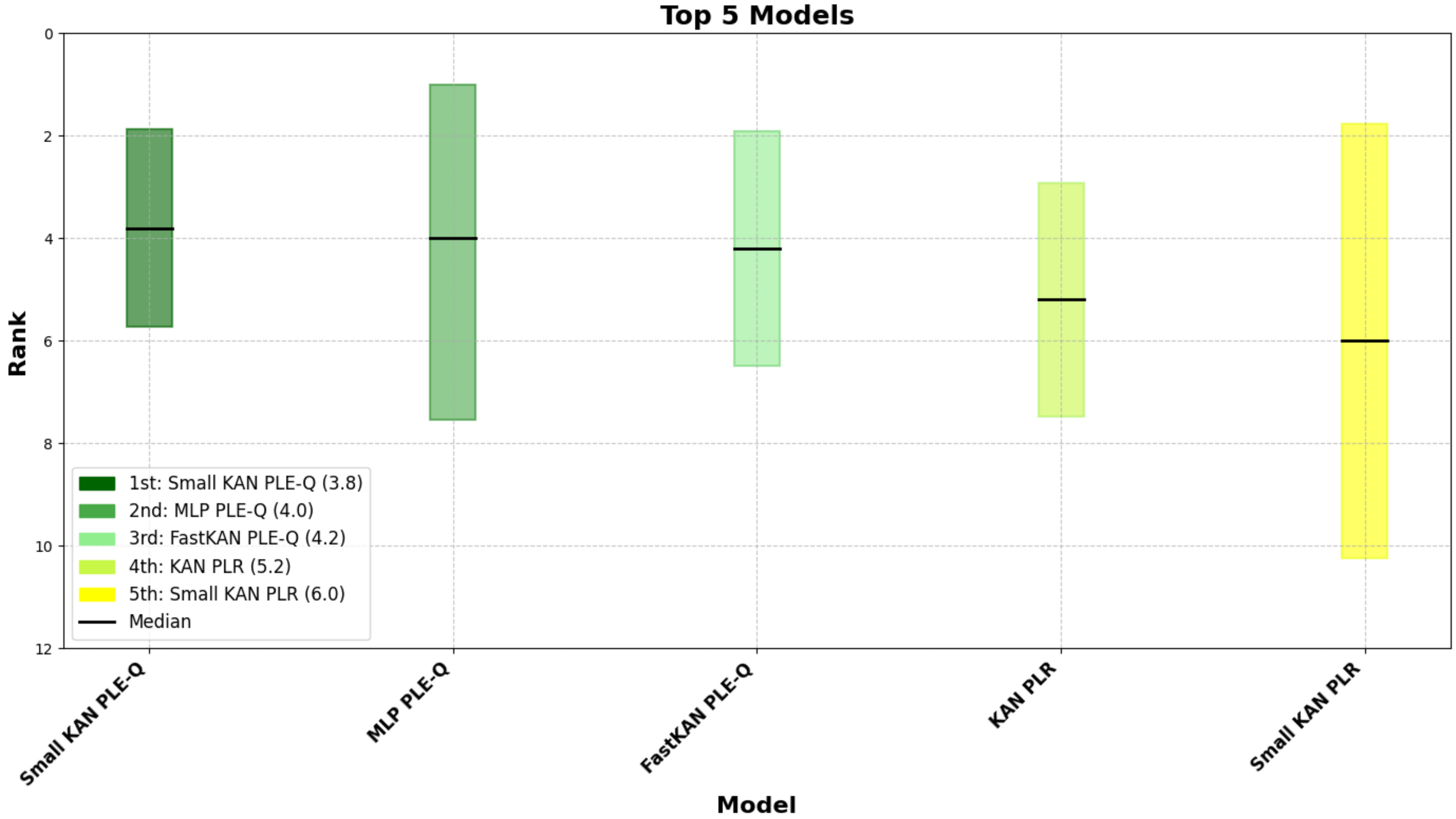
General comparison

All models



General Comparison

Best models



Time issues

KANs are much slower

Model	Train Time Ratio	Test Time Ratio
Small KAN PLE-Q	3,72 ± 0,82	4,38 ± 0,47
MLP PLE-Q	1,14 ± 0.18	1,33 ± 0,15
FastKAN PLE-Q	**1,99 ± 0,39**	**2,45 ± 0,31**
KAN PLR	4,92 ± 1,01	5,89 ± 0,95
Small KAN PLR	3,95 ± 0,82	4,53 ± 0,65

Best models time comparison

Conclusion: FastKAN is very promising for ensembles.

Different Optimizers

🏗️Results are coming...🏗️

- AdamW [8]
- AdEMAmix [9]
- Muon [10]
- MARS [11]

Future work

And current tasks

- Make efficient ensemble of KANs (like TabM, [3]);
- Analyse embeddings in ensembles;
- Test different optimizers;
- Analyse hyperparameters more.

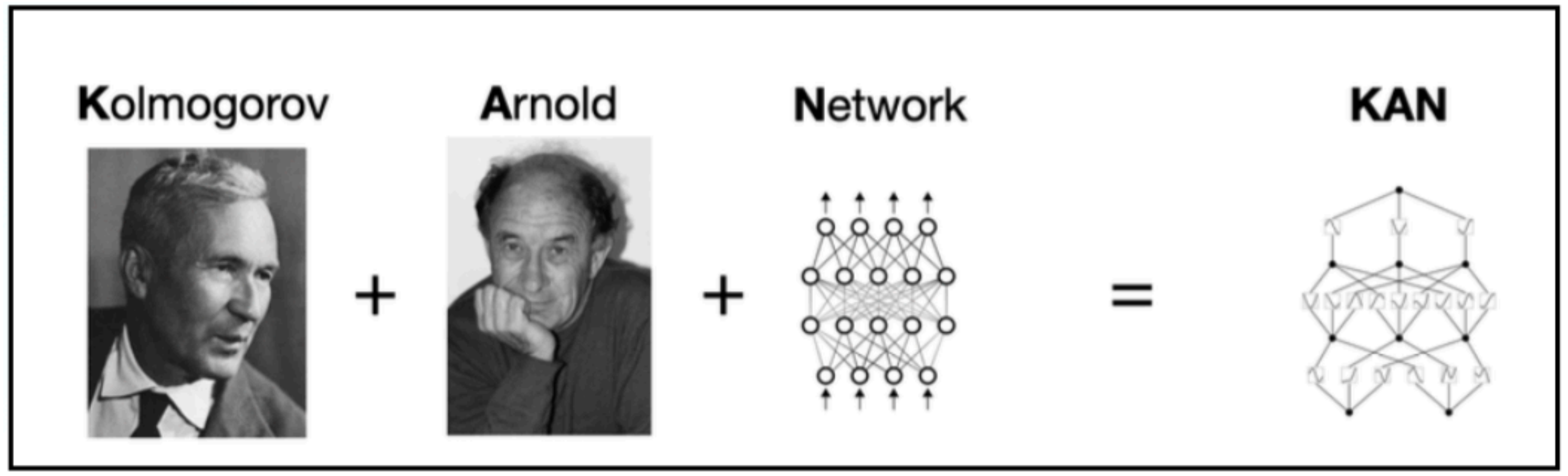


Figure from [1]

Conclusion

- KANs are promising alternative to MLP in Tabular DL
- However, without proper improvements MLPs are not worse;
- **TabM-KAN** could be very accurate!
- Type of KAN is important;
- Time issues need more analysis and optimization.

References

- [1] Ziming Liu et al., *KAN: Kolmogorov-Arnold Networks.*
- [2] Wikipedia, *Kolmogorov-Arnold Representation Theorem*.
- [3] Y.Gorishniy et al., *TabM: Advancing Tabular Deep Learning with Parameter-Efficient Ensembling*.
- [4] Y.Gorishniy et al., *On Embeddings for Numerical Features in Tabular Deep Learning* .
- [5] Bleatan, *efficient-kan, https://github.com/Blealtan/efficient-kan*.
- [6] SynodicMonth, *ChebyKAN, https://github.com/SynodicMonth/ChebyKAN*.
- [7] ZiyaoLI, *fast-kan, https://github.com/ZiyaoLi/fast-kan*.
- [8] Ilya Loshchilov, Frank Hutter, *Decoupled Weight Decay Regularization*.
- [9] Matteo Pagliardini et al., The AdEMAMix Optimizer: Better, Faster, Older.
- [10] Jingyuan Liu et al., *Muon is Scalable for LLM Training*.
- [11] Huizhuo Yan et al., *MARS: Unleashing the Power of Variance Reduction for Training Large Models*.

Thanks for attention!