KAN: Kolmogorov–Arnold Networks

Ziming Liu[1] **Yixuan Wang**[2] Sachin Vaidya[1] Fabian Ruehle[3]
Jim Halverson[3] Marin Soljacic[1] Thomas Y. Hou[2] Max Tegmark[1]
[1]MIT [2]Caltech [3]Northeastern
ICLR April 24-28, 2025

Kolmogorov-Arnold Representation Theorem

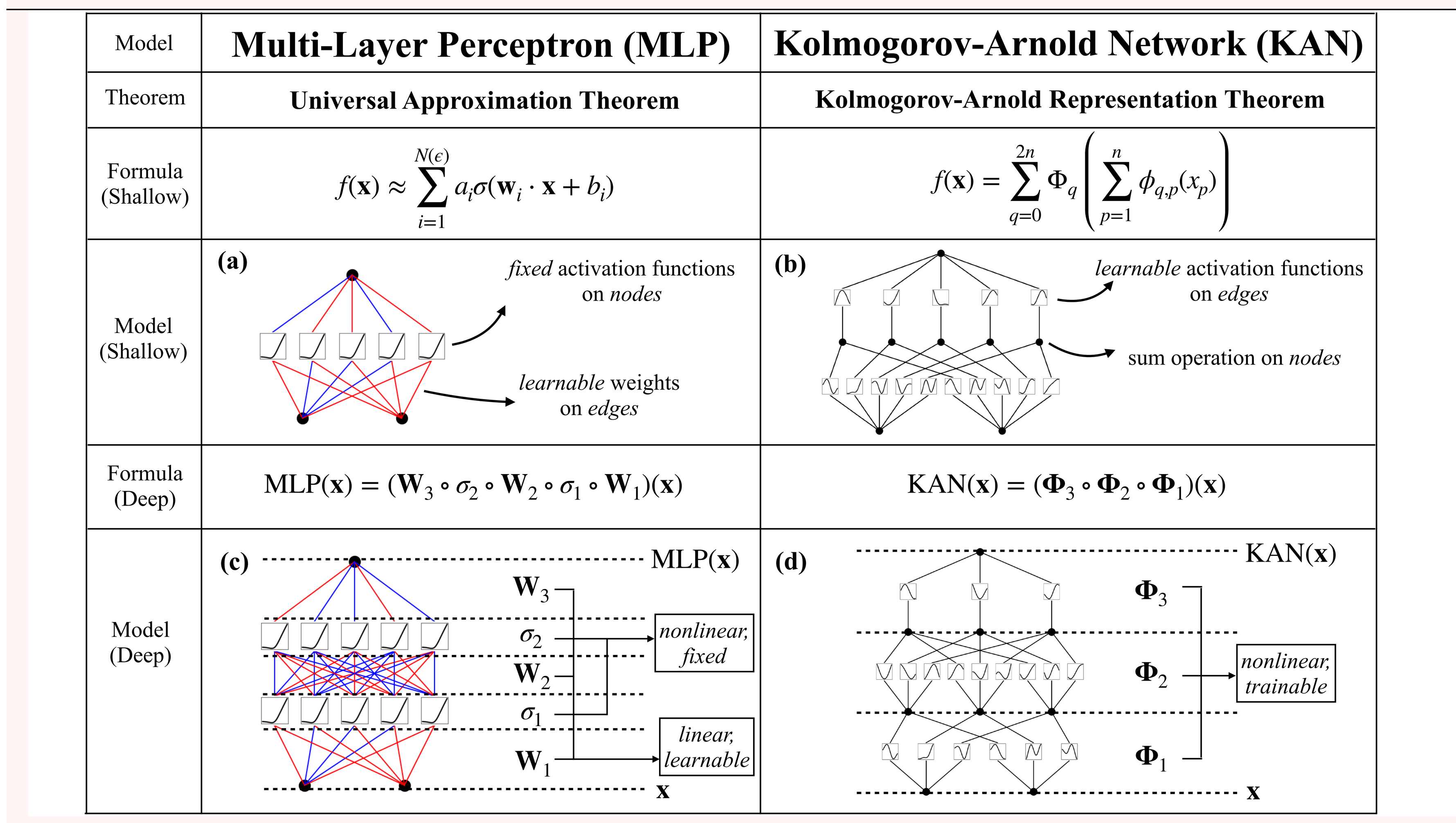
For a continuous $f : [0,1]^n \to \mathbb{R}$

$$f(\mathbf{x}) = f(x_1, \cdots, x_n) = \sum_{q=0}^{2n} \Phi_q\Big(\sum_{p=1}^{n} \phi_{q,p}(x_p)\Big). \tag{1}$$

where $\phi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ are continuous.

- Summing and composition of univariate functions. Potentially address the **curse of dimensionality** (COD).
- Φ_q and $\phi_{q,p}$ not necessarily smooth. In practice we may need more than two layers.

Kolmogorov–Arnold Networks (KANs)



Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=0}^{2n} \Phi_q\left(\sum_{p=1}^{n} \phi_{q,p}(x_p)\right)$
Model (Shallow)	(a) *fixed* activation functions on *nodes*; *learnable* weights on *edges*	(b) *learnable* activation functions on *edges*; sum operation on *nodes*
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)	(c) MLP(x); $\mathbf{W}_3$, σ_2, $\mathbf{W}_2$, σ_1, $\mathbf{W}_1$; *nonlinear, fixed*; *linear, learnable*; x	(d) KAN(x); $\mathbf{\Phi}_3$, $\mathbf{\Phi}_2$, $\mathbf{\Phi}_1$; *nonlinear, trainable*; x

We parametrize the learnable activation functions by B-splines.

Approximation Theory

Suppose that a function $f(\mathbf{x})$ admits a smooth representation

$$f = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)\mathbf{x}\,, \tag{2}$$

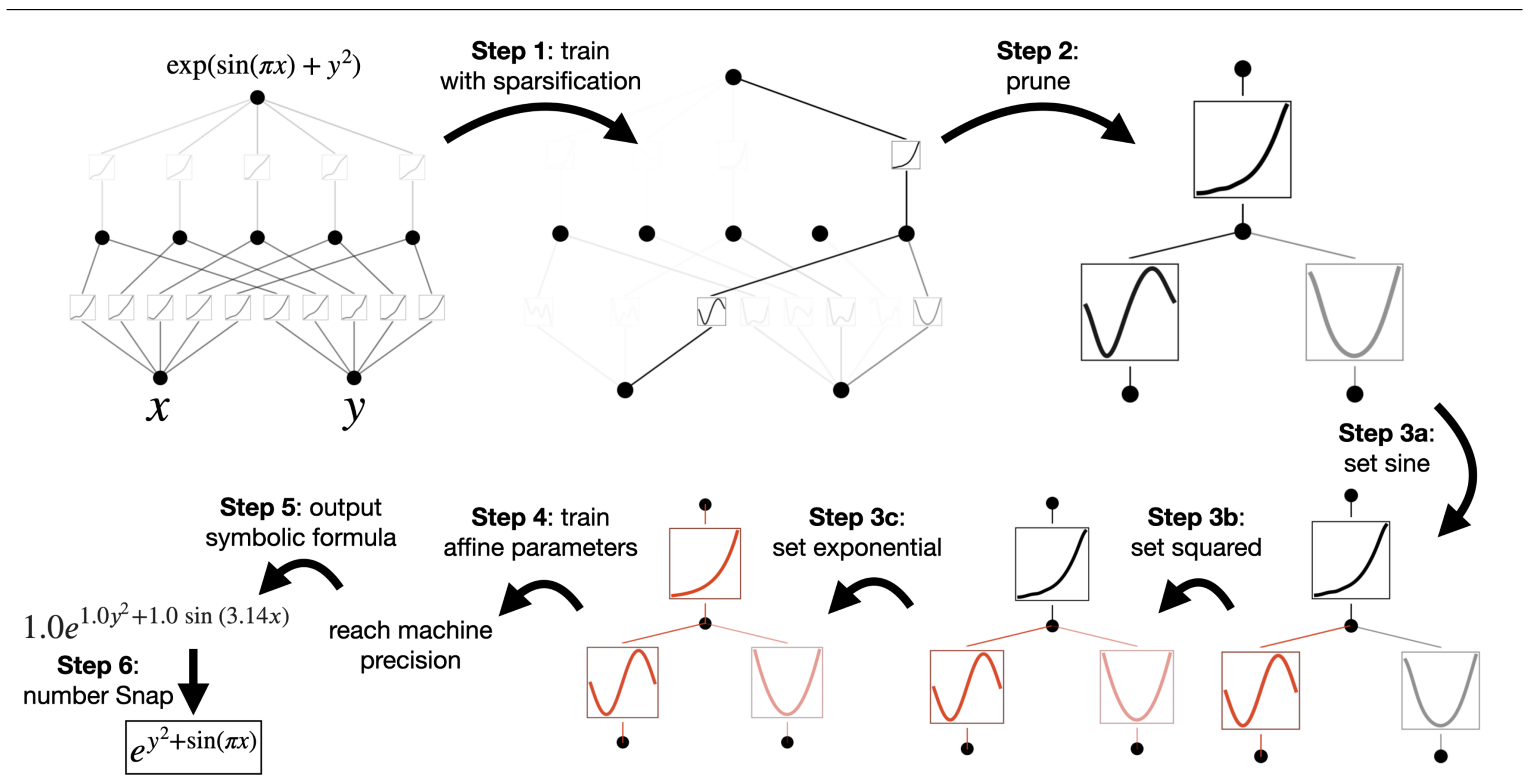
where $\Phi_{l,i,j}$ are smooth with derivatives uniformly bounded up to $k+1$-th order. Then using k-th order B-splines with $G+1$ grid points as activation functions, there exist $\Phi^G_{l,i,j}$ such that for any $0 \le m \le k$, we have the bound

$$\|f - (\mathbf{\Phi}^G_{L-1} \circ \mathbf{\Phi}^G_{L-2} \circ \cdots \circ \mathbf{\Phi}^G_1 \circ \mathbf{\Phi}^G_0)\mathbf{x}\|_{C^m} \le CG^{-k-1+m}\,. \tag{3}$$

In particular for L^2 or RMSE, we have the scaling law $k+1$. Informally, such functions are dense in the class of continuous functions, by [1].

Leveraging the 1D structure to get better scaling laws

KAN for Interpretability: Symbolic Training [2]



Function Fitting

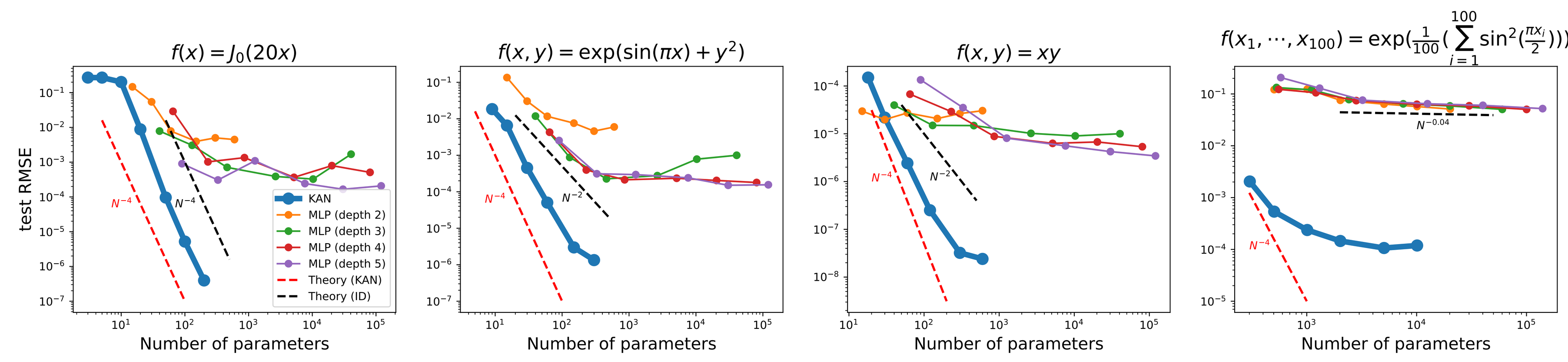


Figure 1. KANs almost saturate the fastest scaling law by theory ($\alpha = 4$), while MLPs scale slowly and plateau.

Image Fitting

Figure 2. KANs outperform MLP with frequency encoding tricks, due to the ability to capture high frequency [4].

Scaling up KANs

Problem	Model	PSNR / L2² Error	Training Time (s)
Image Fitting	KAN [2,128,128,128,128,1], G=[100,10,10,10,10]	45.76	1809
Image Fitting	MLP [2,404,404,404,404,1]	22.09	182
Image Fitting	SIREN 1 [2,128,128,128,128,1]	27.34	254
Image Fitting	SIREN 2 [2,404,404,404,404,1]	30.79	407
Image Fitting	MLP_RFF [2,404,404,404,404,1]	26.26	195
Allen-Cahn	KAN [2,5,5,1], G=5	3.4×10^{-3}	2801
Allen-Cahn	MLP [2,128,128,128,1]	1.5×10^{-1}	478
Allen-Cahn	MLP [2,128,128,128,1] (10x training)	3.9×10^{-4}	4766
Darcy Flow	KAN [2,10,1], G=20	3.9×10^{-4}	66
Darcy Flow	KAN [2,100,1], G=10	4.3×10^{-6}	107
Darcy Flow	KAN [2,10,10,10,10,1], G=5	8.5×10^{-5}	123
Darcy Flow	MLP [2,128,128,128,1]	3.0×10^{-5}	30
Darcy Flow	MLP [2,128,128,128,1] (10x training)	4.5×10^{-6}	277
Darcy Flow	MLP_RFF [2,128,128,128,1]	5.9×10^{-6}	31

Figure 3. KANs can scale up on GPUs and are fast. Examples of image fitting and PDE solving using Adam.

References

[1] Ming-Jun Lai and Zhaiming Shen.
The kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions.
arXiv preprint arXiv:2112.09963, 2021.

[2] Ziming Liu, Pingchuan Ma, Yixuan Wang, Wojciech Matusik, and Max Tegmark.
Kan 2.0: Kolmogorov-arnold networks meet science.
arXiv preprint arXiv:2408.10205, 2024.

[3] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark.
Kan: Kolmogorov-arnold networks.
arXiv preprint arXiv:2404.19756, ICLR 2025 oral, 2024.

[4] Yixuan Wang, Jonathan W Siegel, Ziming Liu, and Thomas Y Hou.
On the expressiveness and spectral bias of kans.
arXiv preprint arXiv:2410.01803, ICLR 2025, 2024.

https://roywangyx.github.io/ roywang@caltech.edu