# 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

## Kolmogorov-Arnold Representation Theorem

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

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

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).
- $\Phi_q$ and $\phi_{q,p}$ not necessarily smooth. In practice we may need more than two layers (but normally it suffices with 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) $\mathrm{MLP}(\mathbf{x})$: $\mathbf{W}_3$, $\sigma_2$, $\mathbf{W}_2$, $\sigma_1$, $\mathbf{W}_1$, $\mathbf{x}$; *nonlinear, fixed*; *linear, learnable* | (d) $\mathrm{KAN}(\mathbf{x})$: $\mathbf{\Phi}_3$, $\mathbf{\Phi}_2$, $\mathbf{\Phi}_1$, $\mathbf{x}$; *nonlinear, trainable* |

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

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

In particular for $L^2$ or RMSE, we have the scaling table

| Paper | Idea | Scaling exponent $\alpha$ |
|---|---|---|
| Sharma & Kaplan [5] | Intrinsic dimensionality | $(k+1)/d$ |
| Michaud et al. [2] | maximum arity | 1 |
| Poggio et al. [3] | compositional sparsity | $m/2$ |
| Ours | K-A representation | $k+1$ |

Table 1. Scaling exponents from different theories $\ell \propto N^{-\alpha}$. $\ell$: test RMSE loss, $N$: number of model parameters, $d$: input intrinsic dimension, $k$: order of piecewise polynomial, $m$: derivative order as in function class $W_m$.

Leveraging the 1D structure to get better scaling laws

## Advantages of KAN Architecture

1. **Interpretability:** KANs with much smaller network size and fewer trainable parameters is normally comparable in performance to MLPs. This makes them ideal for model reduction or symbolic regression.
2. **Accuracy:** KANs can achieve much smaller error due to mesh refinement of splines, with better scaling laws.

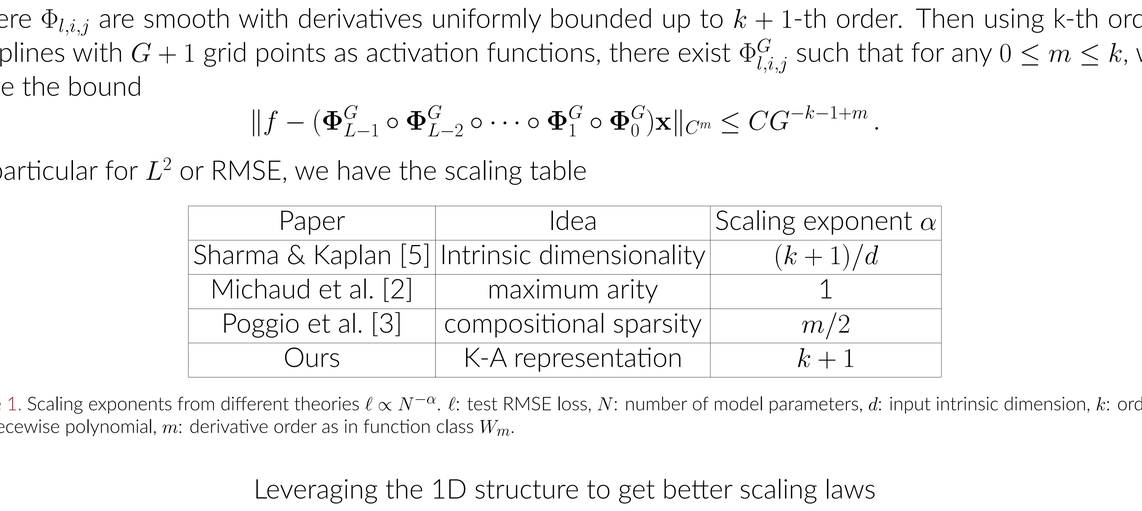


Figure 1. Should I used KANs or MLPs?

## Other Applications

- (What we have done:) Data regression; Anderson localization; Imaging.
- (Future works:) Large Language Models.

## Fitting Special Functions

We compare KANs and MLPs on four toy example, among which are exponential and Bessel functions.

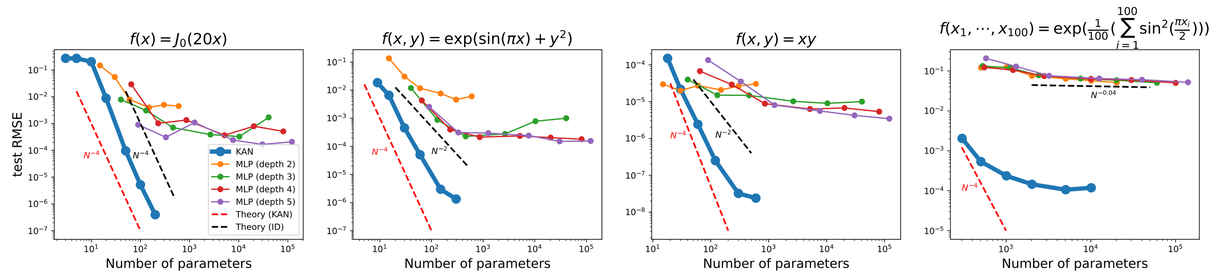


Figure 2. KANs can almost saturate the fastest scaling law predicted by our theory ($\alpha = 4$), while MLPs scales slowly and plateau quickly.

## Solving Partial Differential Equations (PINNs)

We consider a Poisson equation with zero Dirichlet boundary data. For $\Omega = [-1,1]^2$, consider the PDE

$$\begin{aligned} u_{xx} + u_{yy} &= f \quad \text{in } \Omega\,, \\ u &= 0 \quad \text{on } \partial\Omega\,. \end{aligned} \tag{4}$$

$f = -\pi^2(1+4y^2)\sin(\pi x)\sin(\pi y^2) + 2\pi \sin(\pi x)\cos(\pi y^2)$ for which $u = \sin(\pi x)\sin(\pi y^2)$ is the true solution.

We use the framework of physics-informed neural networks (PINNs) [4] with loss prescribed by

$$\text{loss}_{\text{pde}} = \alpha \text{loss}_i + \text{loss}_b := \frac{0.01}{n_i} \sum_{i=1}^{n_i} |u_{xx}(z_i) + u_{yy}(z_i) - f(z_i)|^2 + \frac{1}{n_b} \sum_{i=1}^{n_b} u^2\,,$$

where we use $\text{loss}_i$ to denote the interior loss, discretized and evaluated by a uniform sampling of $n_i$ points $z_i = (x_i, y_i)$ inside the domain, and similarly we use $\text{loss}_b$ to denote the boundary loss, discretized and evaluated by a uniform sampling of $n_b$ points on the boundary.

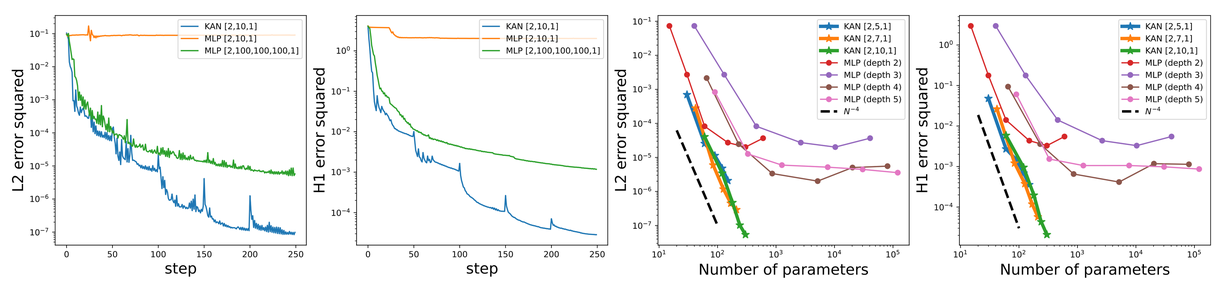


Figure 3. The PDE example. We plot L2 squared and H1 squared losses between the predicted solution and ground truth solution. First and second: training dynamics of losses. Third and fourth: scaling laws of losses against the number of parameters. KANs converge faster, achieve lower losses, and have steeper scaling laws than MLPs.

## AI for Math: Knot Theory

| Method | Architecture | Parameter Count | Accuracy |
|---|---|---|---|
| Deepmind's MLP | 4 layer, width-300 | $\sim 3 \times 10^5$ | 78% |
| KANs | 2 layer, $[17, 1, 14]$ ($G = 3$, $k = 3$) | $\sim 200$ | 81.6% |

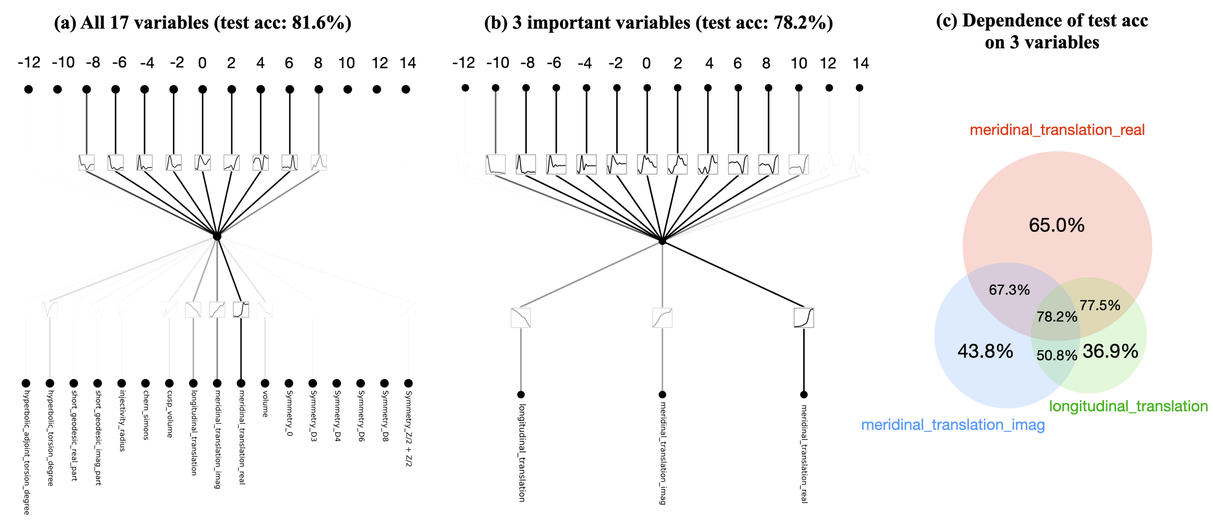


Figure 4. Knot dataset, supervised mode. With KANs, we rediscover Deepmind's results that signature is mainly dependent on meridinal translation.

In [1], supervised learning and human domain experts were combined to arrive at a new theorem relating algebraic and geometric topological invariants. Deepmind's main results for the knot theory dataset are: (1) They use network attribution methods to find the signature $\sigma$ is mostly dependent on meridinal distance $\mu$ (real $\mu_r$, imag $\mu_i$) and longitudinal distance $\lambda$. (2) Human scientists identified that $\sigma$ has high correlation with the $\text{slope} \equiv \text{Re}(\frac{\lambda}{\mu}) = \frac{\lambda \mu_r}{\mu_r^2 + \mu_i^2}$ and derived a bound for $|2\sigma - \text{slope}|$. KANs not only rediscover these results with much smaller networks and much more automation, but also present new results and insights.

## References

[1] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al.
Advancing mathematics by guiding human intuition with ai.
*Nature*, 600(7887):70–74, 2021.

[2] Eric J Michaud, Ziming Liu, and Max Tegmark.
Precision machine learning.
*Entropy*, 25(1):175, 2023.

[3] Tomaso Poggio, Andrzej Banburski, and Qianli Liao.
Theoretical issues in deep networks.
*Proceedings of the National Academy of Sciences*, 117(48):30039–30045, 2020.

[4] Maziar Raissi, Paris Perdikaris, and George E Karniadakis.
Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
*Journal of Computational physics*, 378:686–707, 2019.

[5] Utkarsh Sharma and Jared Kaplan.
A neural scaling law from the dimension of the data manifold.
*arXiv preprint arXiv:2004.10802*, 2020.