Computational Molecular Biology and Bioinformatics

Kolmogorov-Arnold GNNs

Malay Bhattacharyya

Associate Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

September, 2025

Kolmogorov-Arnold representation theorem

In real analysis and approximation theory, the Kolmogorov-Arnold representation theorem states that every multivariate continuous function $f : [0,1]^n \to \mathbb{R}$ can be represented as a superposition of continuous single-variable functions. To elaborate, if f is a multivariate continuous function, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition as follows.

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

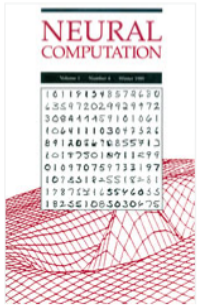
where the inner function ϕ satisfies $\phi_{q,p} : [0,1] \to \mathbb{R}$ and the outer function Φ satisfies $\Phi_q : \mathbb{R} \to \mathbb{R}$.

Kolmogorov-Arnold Networks – The debate

MIT Press Direct | Search... | Neural Computation | Advanced Search | Register

Neural Computation — Issues | Online Early | About | Submit | Subscribe

Volume 1, Issue 4
Winter 1989



December 01 1989

Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant

In Special Collection: CogNet

Federico Girosi, Tomaso Poggio

Check for updates

Author and Article Information

Neural Computation (1989) 1 (4): 465–469.

https://doi.org/10.1162/neco.1989.1.4.465 **Article history**

Cite | Permissions | Share

MIT Press Direct | Search... | Neural Computation | Advanced Search | Register

Neural Computation — Issues | Online Early | About | Submit | Subscribe

Volume 3, Issue 4
Winter 1991

No cover image available

December 01 1991

Kolmogorov's Theorem Is Relevant

In Special Collection: CogNet

Věra Kůrková

Check for updates

Author and Article Information

Neural Computation (1991) 3 (4): 617–622.

https://doi.org/10.1162/neco.1991.3.4.617 **Article history**

Previous Article | Next Article

Cite | Permissions | Share

Kolmogorov-Arnold networks

The Kolmogorov-Arnold Networks (KANs) have recently been proposed as promising alternatives to Multi-Layer Perceptrons (MLPs) [1]. They offer improved expressivity, parameter efficiency and interpretability. Unlike MLPs, KANs are inspired by the Kolmogorov-Arnold representation theorem.

While MLPs have fixed activation functions on nodes (denoting neurons), KANs have learnable activation functions on edges (weights). Interestingly, the KANs have no linear weights at all – every weight parameter is replaced by a univariate function parametrized as a spline.

Note: MLPs are inspired by the universal approximation theorem.

Kolmogorov-Arnold networks

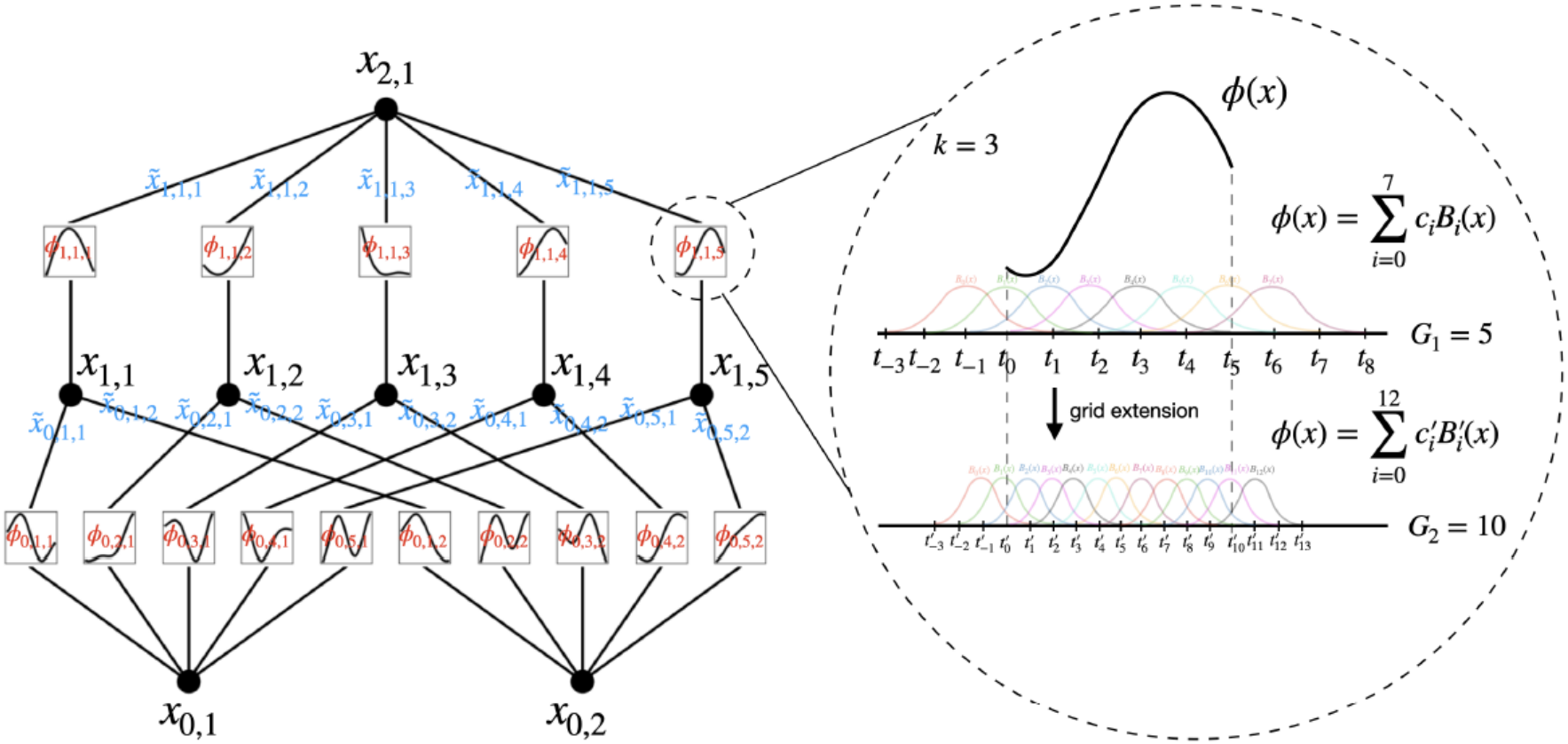
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
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, learnable*; x

Kolmogorov-Arnold networks

KANs are merely combinations of splines (smooth curves) and MLPs, utilizing their respective advantages while avoiding their disadvantages.

Splines are accurate for low-dimensional functions, easy to adjust locally, and able to switch between different resolutions. However, splines have a serious curse of dimensionality (COD) problem, because of their inability to exploit compositional structures. MLPs, on the other hand, suffer less from COD thanks to their feature learning, but are less accurate than splines in low dimensions, because of their inability to optimize univariate functions.

Making the Kolmogorov-Arnold networks deep



An 2-layer KAN with input dimension $n = 2$, width $2n + 1$ in the middle layer, and a single output.

Making the Kolmogorov-Arnold networks deep

The 2-Layer KAN has a shape $[n, 2n+1, 1]$. This can be made deeper by considering a general KAN network as a composition of L layers. Given an input vector $x_0 \in \mathbb{R}_0^n$, the output of a general KAN is given by

$$KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)x.$$

Every neuron has a pre-activation (on the incoming edge) and a post-activation (on the outgoing edge) function. The activation value of a neuron is simply the sum of all incoming post-activations.

Note: All the operations within KAN are differentiable, so we can train KANs with back propagation.

Combining Kolmogorov-Arnold networks with GNNs

To combine the strengths of both KANs and graph neural networks (GNNs), Kolmogorov-Arnold GNNs (KA-GNNs) have been proposed very lately [2].

Unlike the existing approaches (which embed KAN modules into GNNs by replacing MLPs in single steps), KA-GNNs integrate KAN modules into all the three fundamental components of GNNs – node embedding, message-passing and readout. It further introduces Fourier-series-based univariate functions within KAN to enhance function approximation and provide theoretical analysis to support their expressiveness.

KAN-based GNN models

Two architectural variants, namely KA-graph convolutional networks (KA-GCN) and KA-augmented graph attention networks (KA-GAT), are developed and evaluated across seven molecular benchmarks.

Motivated by the strong approximation guarantees of Fourier series, they are adopted as foundational components of the model. The proposed model retain the KAN structure but replaces pre-activations with Fourier series.

KAN-based GNN models for molecular property prediction

In this model, a molecule is represented as a graph $G = (V, E)$, where V denotes the set of nodes (denoting atoms) and E denotes the set of edges (formed among two atoms if they are close, cut-off distance is taken as 5Å). Each node $v \in V$ is associated with a feature vector $f_v \in \mathbb{R}^{d_1}$, and each edge $uv \in E$ is associated with a feature vector $f_{uv} \in \mathbb{R}^{d_2}$.

Each node has a one-hot encoded 92-dimensional feature vector comprising atomic number, radius and electronegativity, which are derived using Rdkit, following the approach in crystal GCN [3] and path complex NN [4]. Similarly, each edge has a 21-dimensional feature vector incorporating both chemical and geometrical information of the bond between the associated molecules. The edges in the model can be classified into two types – covalent bonds and non-covalent bonds – with different initial features.

The KA-GCN model

The KA-GCN extends the standard GCN by incorporating KAN layers at key stages of the model, replacing traditional MLP-based transformations. Unlike conventional MLPs, which apply fixed activation functions (e.g., ReLU), KAN dynamically learns nonlinear transformations, allowing for more expressive feature propagation and improved adaptability to complex molecular graphs. It incorporates KAN in the following steps:

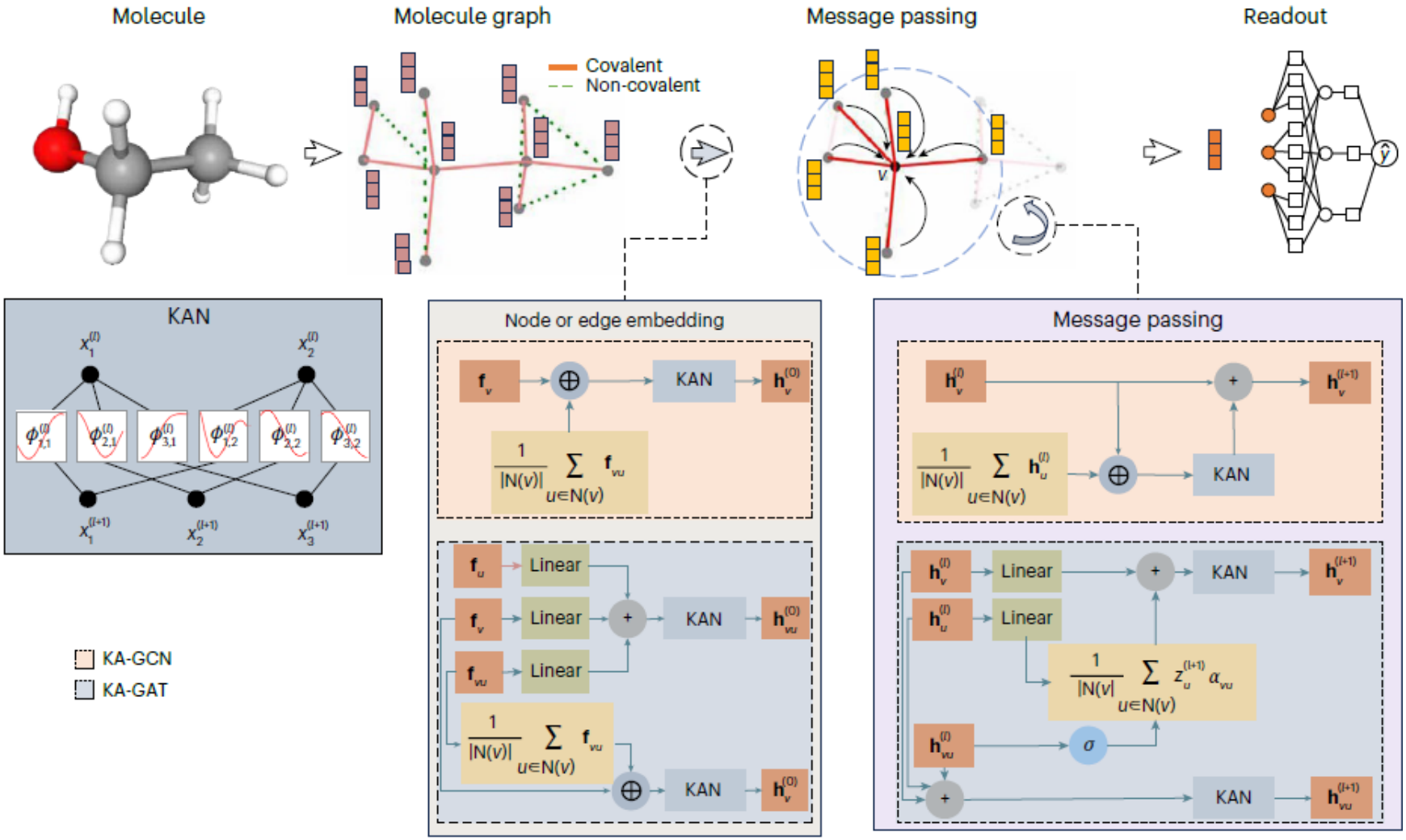
- Node embedding initialization:
  $h_v^{(0)} := KAN\left(\mathbf{f}_v \oplus \left(\frac{1}{|N(v)|}\sum_{u\in N(v)} \mathbf{f}_{vu}\right)\right)$
- KAN-enhanced message passing:
  $h_v^{(l+1)} := h_v^{(l)} + KAN\left(h_v^{(l)} \oplus \ldots \left(\frac{1}{|N(v)|}\sum_{u\in N(v)} h_u^{(l)}\right)\right)$
- Readout and prediction: $\hat{y} := KAN\left(\frac{1}{|V|}\sum_{v\in V} h_v^{(L)}\right)$

The KA-GAT model

Similar to KA-GCN, the KA-GAT model replaces MLP-based transformations in GAT with KAN, enabling adaptive attention mechanisms. Standard GAT uses a fixed attention mechanism that relies on static parameterized functions, whereas KA-GAT learns more flexible, data-dependent attention mechanisms. It incorporates KAN in the following steps:

- Adaptive initialization:
  $h_v^{(0)} := KAN\left(\mathbf{f}_v \oplus \left(\frac{1}{|N(v)|}\sum_{u\in N(v)} \mathbf{f}_{vu}\right)\right),$
  $h_{vu}^{(0)} := KAN(W_{hv}\mathbf{f}_v + W_e\mathbf{f}_{vu} + W_{tv}\mathbf{f}_u)$
- KAN-enhanced attention mechanism
- KAN-based feature update
- Readout and final prediction

The overall framework



References

1. Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y. and Tegmark, M., KAN: Kolmogorov–Arnold Networks, Proceedings of the 13th International Conference on Learning Representations, pp. 1-47, 2025.
2. Li, L., Zhang, Y., Wang, G. and Xia, K., Kolmogorov–Arnold graph neural networks for molecular property prediction. Nature Machine Intelligence, 7:1346-1354, 2025.
3. Xie, T. and Grossman, J.C., Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Physical review letters, 120(14):145301, 2018.
4. Li, L., Xiang, L.I.U., Wang, G., Wang, Y.G. and Xia, K., Path complex neural network for molecular property prediction. In ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling, 2024.