nature machine intelligence

Article

https://doi.org/10.1038/s42256-025-01087-7

# Kolmogorov–Arnold graph neural networks for molecular property prediction

Received: 18 December 2024

Accepted: 26 June 2025

Published online: 11 August 2025

Check for updates

Longlong Li[1,2,3,4], Yipeng Zhang[3,4], Guanghui Wang[1] & Kelin Xia[3] ✉

Graph neural networks (GNNs) have shown remarkable success in molecular property prediction as key models in geometric deep learning. Meanwhile, Kolmogorov–Arnold networks (KANs) have emerged as powerful alternatives to multi-layer perceptrons, offering improved expressivity, parameter efficiency and interpretability. To combine the strengths of both frameworks, we propose Kolmogorov–Arnold GNNs (KA-GNNs), which integrate KAN modules into the three fundamental components of GNNs: node embedding, message passing and readout. We further introduce Fourier-series-based univariate functions within KAN to enhance function approximation and provide theoretical analysis to support their expressiveness. Two architectural variants, KA-graph convolutional networks and KA-augmented graph attention networks, are developed and evaluated across seven molecular benchmarks. Experimental results show that KA-GNNs consistently outperform conventional GNNs in terms of both prediction accuracy and computational efficiency. Moreover, our models exhibit improved interpretability by highlighting chemically meaningful substructures. These findings demonstrate that KA-GNNs offer a powerful and generalizable framework for molecular data modelling, drug discovery and beyond.

Even with the huge successes in efficient experimental tools and state-of-the-art computational models, drug design and discovery is still a time-consuming and extremely costly process[1]. Recently, artificial intelligence, in particular artificial intelligence for sciences, has demonstrated its enormous potential and great power in scientific data analysis. The success of AlphFold models in protein structure prediction has fundamentally changed molecular structural and property analysis, and ushered in a new era of drug design and discovery[2–6]. In general, all molecule-based artificial intelligence models fall into two categories, that is, molecular feature-based machine learning and end-to-end deep learning[7]. The first category relies on molecular descriptors or fingerprints as input features for machine learning models. The key process is molecular featurization (or feature engineering), which extracts or generates molecular features from structural, physical, chemical or biological properties. Among them, structure-based descriptors or fingerprints, especially those derived from topological methods, have proved highly effective[8–11]. Integrating these topology-based molecular features with machine learning models has led to notable success in various drug design stages, such as protein–ligand binding affinity prediction[12,13], protein mutation analysis[14,15] and others[16,17].

The second category includes end-to-end deep learning models that use various molecular representations, such as simplified molecular input line entry system strings, molecule-based images and volumetric data, or molecular graphs, and adopt architectures such as Transformers, two-dimensional (2D) or three-dimensional (3D) convolutional neural networks (CNNs) and geometric deep learning (GDL)[18–23]. In particular, GDL models, such as graph convolutional networks (GCNs)[24], graph autoencoders[25], graph transformers[26] and so on, have been widely used in molecular data analysis and drug design. However, traditional covalent-bond-based molecular graph representations have various limitations, while incorporating non-covalent interactions has been shown to notably enhance performance[27]. These

[1]School of Mathematics, Shandong University, Jinan, China. [2]Data Science Institute, Shandong University, Jinan, China. [3]Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore. [4]These authors contributed equally: Longlong Li, Yipeng Zhang. ✉e-mail: xiakelin@ntu.edu.sg

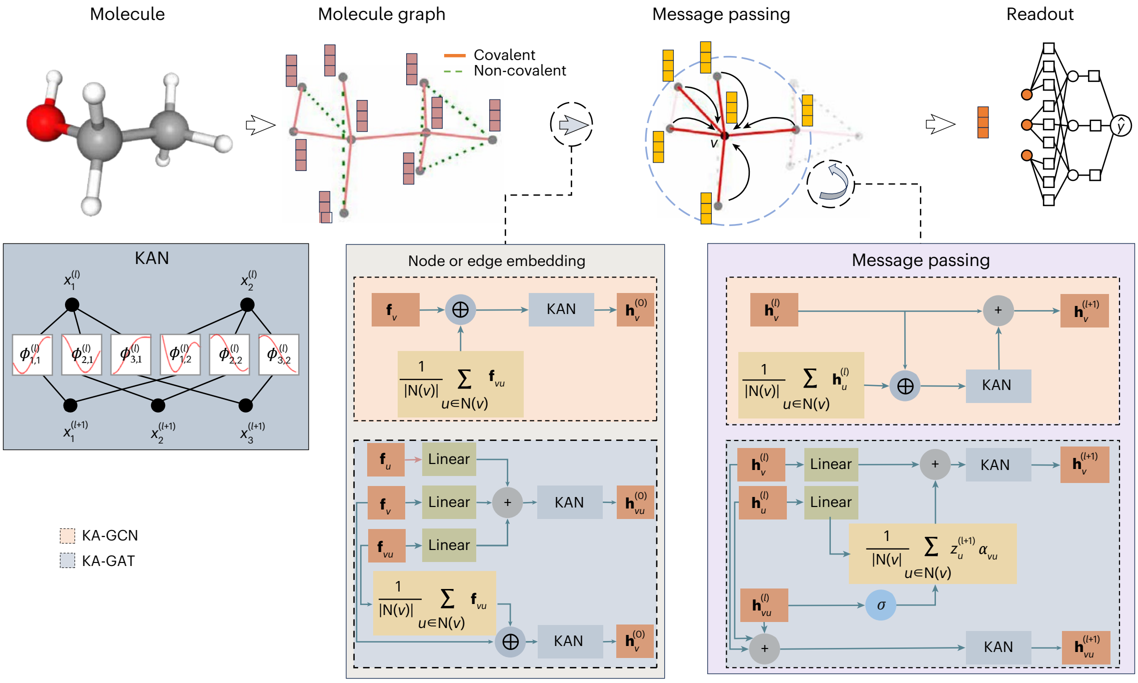

**Fig. 1 | Overview of the KA-GNN model architecture.** The flowchart illustrates the modified components within the GNN: node or edge embedding, message passing, pooling and prediction modules.

results indicate that new graph representations could outperform traditional covalent-bond-based ones. By integrating geometry and/or topology-based molecular graphs into GDL models, we can boost model performance and gain deeper insights into molecular structure and function[28].

Kolmogorov–Arnold networks (KANs), grounded in the Kolmogorov–Arnold representation theorem, are emerging as a compelling alternative to traditional multi-layer perceptrons (MLPs). Unlike conventional MLPs that use constant weights on edges and activation functions on nodes, KANs adopt learnable (basis) univariate functions on edges, enabling accurate and interpretable modelling of complex functions. They have shown promise not only in solving partial differential equations[29], but also across diverse application domains[30–33]. For instance, one study[33] achieved notable improvements in multi-step time series forecasting by combining KANs with long short-term memory networks. Another study[34] enhanced remote sensing scene classification by incorporating KANs into pretrained CNNs. The choice of univariate functions also greatly influences performance. A previous study[35] demonstrated that wavelet functions improve interpretability and frequency analysis. Similarly, another study[36] used Jacobi polynomials in their PointNet-KAN model to achieve strong results with a shallower, more efficient architecture.

More importantly, KAN modules have been embedded into graph neural networks (GNNs) to replace the MLPs used in node embedding, message passing and readout. For instance, GraphKAN[37] applied KANs in the embedding and readout parts. GKAN[38] and GraphKAN[39] used B-spline functions during message passing. KA-GNNs[30] and GNN-SKAN[40] used radial basis functions for message passing, with GNN-SKAN also improving the readout. All these approaches enhanced individual GNN components and outperformed their original models, demonstrating the power of KAN integration.

Here we propose KA-GNN, a unified framework that fully integrates KANs into all three core components of GNNs, including node embedding, message passing and readout. Fourier-based univariate functions are developed to effectively capture both low-frequency and high-frequency structural patterns in graphs, enhancing the expressiveness of feature embedding and message aggregation. A theoretical proof is provided to demonstrate the strong approximation capabilities of our Fourier-KAN model. We further design two variants: KA-GCN (KAN-augmented GCN) and KAN-augmented graph attention network (KA-GAT), as shown in Fig. 1. Extensive experiments across seven benchmark datasets validate KA-GNN's superior accuracy and computational efficiency, establishing it as a promising new paradigm in GDL for non-Euclidean data.

## Results

### Fourier-based KAN layer

KANs are inspired by the Kolmogorov–Arnold superposition theorem, which states that any multivariate continuous function can be expressed as a finite composition of univariate functions and additions. Accordingly, KANs generalize this by constructing each layer as a sum of learnable univariate functions, consistent with the superposition form. Each function acts as a learnable pre-activation, offering a flexible alternative to fixed nonlinearities (for example, ReLU) in MLPs. This enables KANs to use fewer parameters and exhibit smoother gradients, yielding more compact and accurate function approximations.

In this work, we extend the KANs by adopting the Fourier series as the basis for its pre-activation functions (equations (4) and (5)). Compared with previous KAN implementations that use B-spline functions, the Fourier-based formulation enables both low-frequency and high-frequency structural patterns in graphs to be effectively captured. The use of global trigonometric functions allows for smooth, compact representations that benefit gradient flow and parameter efficiency.

We theoretically establish that the proposed Fourier-based KANs architecture possesses strong approximation capability. Although this structure is inspired by the Kolmogorov–Arnold representation theorem, the univariate functions involved in that theorem can be highly non-smooth, limiting its suitability as a theoretical foundation. To rigorously ground the expressive power of our model, we instead base our theoretical foundation on Carleson's convergence theorem and Fefferman's multivariate extension[41,42]:

**Theorem 1.** *Let* $Z^n$ *denote the n-dimensional integer lattice, and* $Z_N^n = \{1, 2, \dots, N\}^n \subset Z^n$*. For a square-integrable function* $f \in L^2([0, 2\pi]^n)$ *and its Fourier expansion:*

$$f(\mathbf{x}) \sim \sum_{\mathbf{k}\in Z^n} \left(a_{\mathbf{k}} \cos(\mathbf{k}\cdot\mathbf{x}) + b_{\mathbf{k}} \sin(\mathbf{k}\cdot\mathbf{x})\right),$$

*where* $x \in [0, 2\pi]^n$, *we have*

$$f(\mathbf{x}) = \lim_{N\to\infty} \sum_{\mathbf{k}\in Z_N^n} \left(a_{\mathbf{k}} \cos(\mathbf{k}\cdot\mathbf{x}) + b_{\mathbf{k}} \sin(\mathbf{k}\cdot\mathbf{x})\right)$$

*almost everywhere.*

Motivated by the strong approximation guarantees of Fourier series, we adopt them as foundational components of our model. We retain the KAN structure but replace pre-activations with Fourier series. The following result further supports its expressive power:

**Theorem 2.** *Let* $f \in L^2([0, 2\pi]^n)$ *be a square-integrable function. For almost every* $\mathbf{x} \in [0, 2\pi]^n$ *and any* $\epsilon > 0$, *there exists a positive integer* $K$ *and a sequence of Fourier-based* KANs: $\mathrm{KAN}_l$, *for* $l = 0, 1, \ldots, L$, *such that the number of harmonics in pre-activation functions is bounded* by $K$, *and*

$$|f(\mathbf{x}) - \mathrm{KAN}_L \circ \mathrm{KAN}_{L-1} \circ \cdots \circ \mathrm{KAN}_0(\mathbf{x})| < \epsilon,$$

*where* $\circ$ *denotes function composition.*

These theoretical results show that the Fourier-based KAN architecture can approximate any square-integrable multivariate function, providing strong expressive power and theoretical guarantees. To empirically validate this, we compare its fitting performance with a standard two-layer MLP across six representative functions. Results in Supplementary Fig. 1 demonstrate the superior approximation capability of our approach, confirming the practical benefits of incorporating Fourier series into the KAN framework.

### KA-GNNs

Inspired by the theoretical strengths of the Fourier-based KAN layer, we develop a class of GNNs, known as KA-GNNs, that systematically integrate Fourier-based KAN modules across the entire GNN pipeline, including node embedding initialization, message passing and graph-level readout. This integration replaces conventional MLP-based transformations with Fourier-based KAN modules, yielding a unified, fully differentiable architecture with enhanced representational power and improved training dynamics. By leveraging adaptive, data-driven nonlinear mappings over fixed activations, KA-GNNs construct richer node embeddings, modulate feature interactions during message passing and capture more expressive graph-level representations.

To instantiate the KA-GNNs framework, we design two variants, that is, KA-GCN and KA-GAT, which respectively integrate Fourier-based KAN modules into GCN and GAT backbones. In KA-GCN, each node's initial embedding is computed by passing the concatenation of its atomic features (for example, atomic number, radius) and the average of its neighbouring bond features (for example, bond type, bond length) through a KAN layer. This design encodes both atomic identity and local chemical context via data-dependent trigonometric transformations. The message-passing layers follow the GCN scheme, while node features are updated via residual KANs, replacing traditional MLPs.

KA-GAT introduces an expressive design by also incorporating edge embeddings. Both node and edge features are initialized using KAN layers. Each edge embedding is formed by fusing its bond features with features of the two endpoint atoms, enabling context-aware bond representations. During message passing, attention scores are computed from edge embeddings, and both node and edge features are iteratively updated via KAN layers. This supports adaptive attention and rich feature interactions across covalent and non-covalent bonds. By replacing static aggregation and fixed MLPs with KAN-driven mappings, KA-GCN and KA-GAT achieve more flexible and expressive molecular graph modelling.

### KA-GNNs for molecular property prediction

To apply KA-GCN and KA-GAT to molecular data, we represent each molecule as a graph $G = (V, E)$, where atoms are nodes ($V$) and both covalent and non-covalent interactions are edges ($E$). Specifically, an edge connects two atoms if they are bonded or spatially close within a 5-Å cut-off, capturing both local bonds and long-range dependencies. Each atom is encoded with a 92-dimensional vector following crystal graph convolutional neural networks[43], incorporating atomic number, group, period, electronegativity and covalent radius. Each bond is encoded with a 21-dimensional vector: covalent bonds are represented using chemical descriptors, while non-covalent bonds use spatial descriptors. These two bond types are treated as distinct edge categories with different initial encoding settings. This enriched graph representation equips KA-GCN and KA-GAT with both chemical and spatial context, allowing Fourier-based KAN layers to effectively model complex interatomic relationships.

### Performance and comparison of KA-GNNs

To evaluate the performance of our proposed KA-GNN models, we consider seven widely used benchmark datasets from MoleculeNet[44]. Of these, three datasets, including MUV, HIV and BACE, are from the biophysics domain, while the remaining four including BBBP, Tox21, SIDER and ClinTox, belong to the physiology domain.

In our comparative analysis across seven datasets, we evaluated a range of GDL models, including AttentiveFP[45], D-MPNN[46], Mol-GDL[28], N-Gram[47], PretrainGNN[48], GraphMVP[49], MolCLR[50], GEM[51], Uni-mol[52] and SMPT[53]. In addition to our KA-GNN models, we also include other KAN-based GNN architectures, such as GNN-SKAN[40], GraphKAN[39] and KA-GNNs[30].

To ensure fair comparisons, we adopt consistent experimental settings across all models. Details of the hyperparameter configurations and input feature constructions are provided in Supplementary Tables 1 and 2.

The comparative results in Table 1 further confirm the superiority of the KA-GNN models. Our model achieves state-of-the-art performance across all benchmark datasets, particularly excelling on complex and challenging datasets such as ClinTox and MUV. These results demonstrate the robust capability of our model in handling molecular data. Notably, in the BBBP dataset, the area under the curve (AUC) for KA-GCN and KA-GAT showed improvements of approximately 7.95% and 7.68%.

### Importance of Fourier series for KA-GNNs

We use a Fourier-based KAN as the core module in our GNN architecture, departing from traditional MLP-based designs. To assess the influence of basis functions on molecular property prediction, we evaluate B-spline, polynomial and Fourier-series formulations within both KA-GCN and KA-GAT frameworks. All variants are benchmarked under identical GCN and GAT settings. The polynomial transformation is defined in equation (1):

$$x_j^{(l+1)} = \sum_{i=1}^{n_l} \sum_{k=0}^{K} \left( C_{k,j,i}^{(l)} \left(x_i^{(l)}\right)^k \right), \tag{1}$$

where $C_{k,j,i}^{(l)}$ denotes learnable parameters initialized from $\mathcal{N}(0, \frac{1}{n_{l+1}\times K})$, and $\mathbf{x}^{(l)} = (x_1^{(l)}, x_2^{(l)}, \ldots, x_{n_l}^{(l)})$ is the activation vector at layer $l$.

Table 2 shows consistent performance gains across datasets, validating the effectiveness of KA-GNNs for molecular tasks. The superior expressiveness of Fourier-based KANs underpins these results, supported theoretically in Theorem 2 and empirically in our function-fitting experiments. Further, our ablation study reveals that Fourier-based KANs improve not only accuracy but also feature embedding, message passing and final prediction stages.

We further compare standard GCN/GAT models with their KA-GNN counterparts (using Fourier series) under the same molecular graphs. As shown in Table 3, KA-GNNs consistently outperform baselines,

**Table 1 | Comparison of KA-GNNs with state-of-the-art GDL models across various molecular property prediction datasets**

| Model | BACE | BBBP | ClinTox | SIDER | Tox21 | HIV | MUV |
|---|---|---|---|---|---|---|---|
| Number of molecules | 1,513 | 2,039 | 1,478 | 1,427 | 7,831 | 41,127 | 93,808 |
| Number of average atoms | 65 | 46 | 50.58 | 65 | 36 | 46 | 43 |
| Number of tasks | 1 | 1 | 2 | 27 | 12 | 1 | 17 |
| D-MPNN[46] | 0.809$_{(0.006)}$ | 0.710$_{(0.003)}$ | 0.906$_{(0.007)}$ | 0.570$_{(0.007)}$ | 0.759$_{(0.007)}$ | 0.771$_{(0.005)}$ | 0.786$_{(0.014)}$ |
| AttentiveFP[45] | 0.784$_{(0.022)}$ | 0.663$_{(0.018)}$ | 0.847$_{(0.003)}$ | 0.606$_{(0.032)}$ | 0.781$_{(0.005)}$ | 0.757$_{(0.014)}$ | 0.786$_{(0.015)}$ |
| N-GramRF[47] | 0.779$_{(0.015)}$ | 0.697$_{(0.006)}$ | 0.775$_{(0.040)}$ | 0.668$_{(0.007)}$ | 0.743$_{(0.009)}$ | 0.772$_{(0.004)}$ | 0.769$_{(0.002)}$ |
| N-GramXGB[47] | 0.791$_{(0.013)}$ | 0.691$_{(0.008)}$ | 0.875$_{(0.027)}$ | 0.655$_{(0.007)}$ | 0.758$_{(0.009)}$ | 0.787$_{(0.004)}$ | 0.748$_{(0.002)}$ |
| PretrainGNN[48] | 0.845$_{(0.007)}$ | 0.687$_{(0.013)}$ | 0.726$_{(0.015)}$ | 0.627$_{(0.008)}$ | 0.781$_{(0.006)}$ | 0.799$_{(0.007)}$ | 0.813$_{(0.021)}$ |
| GraphMVP[49] | 0.812 $_{(0.009)}$ | 0.724 $_{(0.016)}$ | 0.791 $_{(0.028)}$ | 0.639 $_{(0.012)}$ | 0.759 $_{(0.005)}$ | 0.770 $_{(0.012)}$ | 0.777 $_{(0.006)}$ |
| MolCLR[50] | 0.824 $_{(0.009)}$ | 0.722 $_{(0.021)}$ | 0.912 $_{(0.035)}$ | 0.589 $_{(0.014)}$ | 0.750 $_{(0.002)}$ | 0.781 $_{(0.005)}$ | 0.796 $_{(0.019)}$ |
| GEM[51] | 0.856$_{(0.011)}$ | 0.724$_{(0.004)}$ | 0.901$_{(0.013)}$ | 0.672$_{(0.004)}$ | 0.781$_{(0.001)}$ | 0.806$_{(0.009)}$ | 0.817$_{(0.005)}$ |
| Mol-GDL[28] | 0.863$_{(0.019)}$ | 0.728$_{(0.019)}$ | 0.966$_{(0.002)}$ | 0.831$_{(0.002)}$ | 0.794$_{(0.005)}$ | 0.808$_{(0.007)}$ | 0.675$_{(0.014)}$ |
| Uni-mol[52] | 0.857 $_{(0.002)}$ | 0.729 $_{(0.006)}$ | 0.919 $_{(0.018)}$ | 0.659 $_{(0.013)}$ | 0.796 $_{(0.005)}$ | 0.808 $_{(0.003)}$ | 0.821 $_{(0.013)}$ |
| SMPT[53] | 0.873 $_{(0.015)}$ | 0.734 $_{(0.003)}$ | 0.927 $_{(0.002)}$ | 0.676 $_{(0.050)}$ | 0.797 $_{(0.001)}$ | 0.812 $_{(0.001)}$ | 0.822 $_{(0.008)}$ |
| GNN-SKAN[40] | 0.747$_{(0.009)}$ | 0.676$_{(0.014)}$ | - | 0.614$_{(0.005)}$ | 0.747$_{(0.005)}$ | 0.786$_{(0.015)}$ | - |
| GraphKAN[39] | 0.823$_{(0.011)}$ | 0.731$_{(0.017)}$ | 0.984$_{(0.003)}$ | 0.837$_{(0.001)}$ | 0.753$_{(0.007)}$ | 0.711$_{(0.016)}$ | 0.715$_{(0.014)}$ |
| KA-GNNs[30] | 0.752$_{(0.011)}$ | 0.721$_{(0.003)}$ | 0.972$_{(0.001)}$ | 0.831$_{(0.004)}$ | 0.730$_{(0.012)}$ | 0.717$_{(0.018)}$ | 0.701$_{(0.006)}$ |
| KA-GCN | **0.890**$_{(0.014)}$ | **0.787**$_{(0.014)}$ | **0.992**$_{(0.005)}$ | 0.842$_{(0.001)}$ | 0.799$_{(0.005)}$ | 0.821$_{(0.005)}$ | **0.834**$_{(0.009)}$ |
| KA-GAT | 0.884$_{(0.004)}$ | 0.785$_{(0.021)}$ | 0.991$_{(0.005)}$ | **0.847**$_{(0.002)}$ | 0.800$_{(0.006)}$ | **0.823**$_{(0.002)}$ | 0.834$_{(0.010)}$ |

Note that bold values indicate the best-performing models; underlined values indicate the second best. All metrics are average receiver operating characteristic-AUC and standard deviations are shown as subscripts.

**Table 2 | Comparison of the performance of KA-GNN (KA-GCN and KA-GAT) models with three types of base function, including B-spline, polynomial and Fourier series**

| Dataset | KA-GCN models | | | KA-GAT models | | |
|---|---|---|---|---|---|---|
| | B-spline | Polynomial | Fourier series | B-spline | Polynomial | Fourier series |
| BACE | 0.771$_{(0.012)}$ | 0.853$_{(0.027)}$ | **0.890**$_{(0.014)}$ | 0.808$_{(0.009)}$ | 0.8319$_{(0.007)}$ | **0.884**$_{(0.004)}$ |
| BBBP | 0.723$_{(0.008)}$ | 0.654$_{(0.009)}$ | **0.787**$_{(0.014)}$ | 0.657$_{(0.004)}$ | 0.708$_{(0.005)}$ | **0.785**$_{(0.021)}$ |
| ClinTox | 0.973$_{(0.006)}$ | 0.981$_{(0.004)}$ | **0.992**$_{(0.005)}$ | 0.948$_{(0.003)}$ | 0.983$_{(0.003)}$ | **0.991**$_{(0.005)}$ |
| SIDER | 0.824$_{(0.003)}$ | 0.832$_{(0.006)}$ | **0.842**$_{(0.001)}$ | 0.825$_{(0.004)}$ | 0.836$_{(0.001)}$ | **0.847**$_{(0.002)}$ |
| Tox21 | 0.724$_{(0.005)}$ | 0.715$_{(0.004)}$ | **0.799**$_{(0.005)}$ | 0.731$_{(0.012)}$ | 0.753$_{(0.004)}$ | **0.800**$_{(0.006)}$ |
| HIV | 0.753$_{(0.007)}$ | 0.804$_{(0.010)}$ | **0.821**$_{(0.005)}$ | 0.744$_{(0.018)}$ | 0.818$_{(0.006)}$ | **0.823**$_{(0.002)}$ |
| MUV | 0.638$_{(0.008)}$ | 0.787$_{(0.012)}$ | **0.834**$_{(0.009)}$ | 0.792$_{(0.013)}$ | 0.797$_{(0.011)}$ | **0.834**$_{(0.010)}$ |

Note that bold values indicate the best-performing models. All metrics are average receiver operating characteristic-AUC. Standard deviations are shown as subscripts.

highlighting Fourier KANs' enhanced ability to capture chemical structure representations.

To investigate the architectural drivers behind KA-GNNs' performance, we conducted an extensive ablation study examining input normalization, KAN modularity, graph construction (for example, non-covalent bonds), regression capabilities, scalability and hyperparameter sensitivity (Supplementary Tables 3–8).

Notably, we tested six normalization schemes for node and edge features. KA-GNNs maintained strong performance across all, underscoring their robustness to input scale variation.

## Efficiency of Fourier-based KAN in GNNs

As shown in Fig. 2a,b, Fourier-based KAN models consistently achieve lower runtime than both B-spline KANs and MLP-based GCN/GATs. This efficiency arises from the global nature of sinusoidal basis functions, which enable compact function approximation with fewer trainable parameters.

Unlike MLPs with large dense weight matrices, KANs use fixed basis functions with lightweight learnable coefficients, notably reducing computational cost. Although B-spline KANs also have lower parameter counts, their localized control points and interpolation steps introduce overhead. By contrast, Fourier bases offer smooth, global approximations without such burdens.

**Table 3 | Comparison of GCN/GAT models and KA-GCN/KA-GAT with Fourier series**

| Dataset | GCN | KA-GCN | GAT | KA-GAT |
|---|---|---|---|---|
| BACE | 0.835$_{(0.014)}$ | **0.890**$_{(0.014)}$ | 0.834$_{(0.012)}$ | **0.884**$_{(0.004)}$ |
| BBBP | 0.735$_{(0.011)}$ | **0.787**$_{(0.014)}$ | 0.707$_{(0.007)}$ | **0.785**$_{(0.021)}$ |
| ClinTox | 0.979$_{(0.004)}$ | **0.992**$_{(0.005)}$ | 0.983$_{(0.006)}$ | **0.991**$_{(0.005)}$ |
| SIDER | 0.834$_{(0.001)}$ | **0.842**$_{(0.001)}$ | 0.836$_{(0.002)}$ | **0.847**$_{(0.002)}$ |
| Tox21 | 0.747$_{(0.006)}$ | **0.799**$_{(0.005)}$ | 0.751$_{(0.007)}$ | **0.800**$_{(0.006)}$ |
| HIV | 0.762$_{(0.005)}$ | **0.821**$_{(0.005)}$ | 0.761$_{(0.003)}$ | **0.823**$_{(0.002)}$ |
| MUV | 0.741$_{(0.006)}$ | **0.834**$_{(0.009)}$ | 0.784$_{(0.019)}$ | **0.834**$_{(0.010)}$ |

Note that bold values indicate the best-performing models. All metrics are average receiver operating characteristic-AUC. Standard deviations are shown as subscripts.

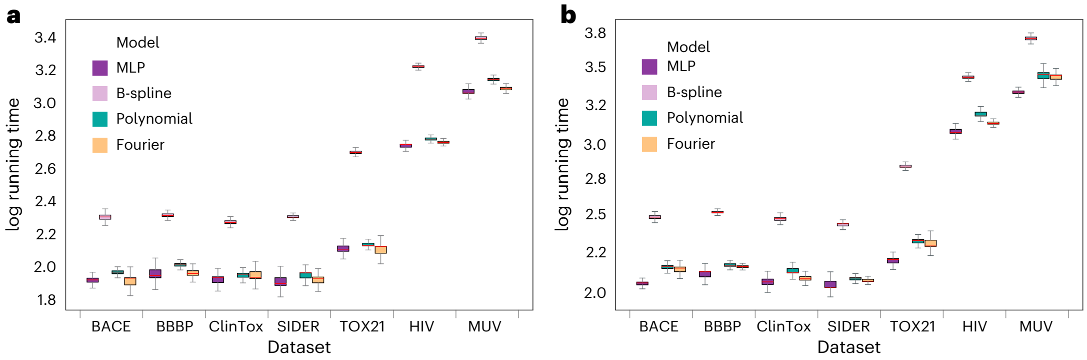

**Fig. 2 | Efficiency comparison of MLP-based models and KA-GNNs with different basis functions across seven benchmark datasets. a**, KA-GCN models. **b**, KA-GAT models. Boxplots showing the distribution of $\log_{10}$ running times (in seconds) for MLP-based models and KA-GNNs using different basis functions across seven datasets. Each box represents the interquartile range, the centre line indicates the median and whiskers extend to 1.5× interquartile range. Individual data points ($n = 3$) are overlaid as dots.

Fourier-based KA-GNNs also outperform traditional GCNs and GATs in efficiency. While GCNs scale with node and layer count, and GATs with attention heads, Fourier-KAN simplifies message passing by eliminating redundant parameter matrices. This accelerates convergence and lowers training time, making it highly suitable for large-scale graph learning.

### Interpretation of KA-GAT analysis

The interpretability of the KAN module is a key advantage. To demonstrate this, we analyse KA-GAT's prediction for the molecule Fc1ccc(NC(=O)c2ncc(cc2)C#N)cc1[C@]1(N=C(OCC1)N)C. Figure 3a,b shows its 2D and 3D structures (via MolView) as chemical references. Figure 3c provides a saliency map highlighting atom- and bond-level contributions. Modelling the molecule as a graph $G = (V, E)$ with feature matrices $f_V$ and $f_E$, and using a trained model $f_\theta$ targeting class $c$, the output score is $s = f_\theta(G, f_V, f_E)_c$. Using input gradients, the $L^1$ norm quantifies each input's influence:

$$\mathrm{Saliency}(v) = \sum_{k=1}^{d_1} \left| \frac{\partial s}{\partial \mathbf{f}_v[k]} \right|, \quad \mathrm{Saliency}(e) = \sum_{k=1}^{d_2} \left| \frac{\partial s}{\partial \mathbf{f}_e[k]} \right|.$$

Here, $d_1$ and $d_2$ is the dimensional of node feature and edge feature, respectively, and $\mathbf{f}_v[k]$ and $\mathbf{f}_e[k]$ is the $k$th element of $\mathbf{f}_v$ and $\mathbf{f}_e$. These gradients measure prediction sensitivity to input perturbations.

To extract key substructures, we retain nodes and edges above the 40th percentile of normalized saliency, yielding a 16-node, 11-edge high-attribution subgraph. Functional groups are defined using SMARTS patterns, with group-level saliency computed as:

$$\mathrm{Saliency}_{\mathrm{group}}(g) = \frac{1}{|V_g|} \sum_{u \in V_g} \mathrm{Saliency}_V(u),$$

where $V_g \subseteq V$ are the atoms in group g. In this case, fluoro (–F) ranks highest (2.9535), followed by amide (–CONH–, 2.4133), aromatic rings (2.3251) and amine (–NH–, 1.1992), reflecting the model's attention to electron-withdrawing and resonance-stabilizing groups.

Figure 3d shows subgraph importance from a KA-GAT-adapted GNNExplainer. Soft masks $m_V \in [0, 1]^{|V|}$, $m_E \in [0, 1]^{|E|}$ are applied element-wise to features. The loss balances output fidelity and sparsity:

$$\mathcal{L} = \left\| f_\theta(G, f_V, f_E) - f_\theta(G, f_V \odot m_V, f_E \odot m_E) \right\|^2 + \lambda_V \parallel m_V \|_1 + \lambda_E \parallel m_E \|_1$$

where $\lambda_V$ and $\lambda_E$ are regularization terms. The resulting subgraph $G' \subseteq G$ captures the most influential atoms and bonds.

In summary, KA-GAT identifies chemically relevant features, such as fluoro and amide groups, yielding interpretable and domain-aligned predictions that support its use in drug discovery.

## Discussion

Our KA-GNNs have synergistically incorporated KAN module into GNN architecture and have three core innovations. First, in molecular graph construction, we incorporate both covalent and non-covalent interactions by introducing edges based on a cut-off distance. This approach allows KA-GNN to capture a broader range of molecular interactions beyond traditional covalent bonds, leading to a more comprehensive understanding of molecular structures and their properties. Second, we integrate the KAN into the GNN framework, notably reducing the number of parameters while enhancing interpretability and expressive power. Unlike traditional MLP-based message-passing schemes, KAN offers a more flexible function approximation mechanism, leading to improved predictive accuracy with fewer trainable parameters. Third, we introduce a Fourier-based activation function in KAN, replacing the original B-spline functions. Fourier series, widely used in signal processing and function approximation, enable KA-GNN to better capture high-frequency patterns in graph-structured data. By learning and optimizing Fourier coefficients, KA-GNN achieves higher accuracy and stability, particularly in modelling complex molecular interactions and biochemical systems.

Even with the promising performance, our KA-GNNs can be further improved, in particular, the alignment of the learned Fourier-based representations with explicit chemical and biological principles remains a great challenge. Although saliency maps and subgraph importance analysis have demonstrated that our KA-GNN effectively identifies key molecular substructures, critical atomic interactions and functionally relevant subgraphs, an insightful feature importance analysis that connects an individual or frequency-related Fourier function to physical or chemical meanings is absent. In fact, we do not obtain any 'meaningful' results using KAN-based pruning approach for feature importance analysis of our learned Fourier functions. An exciting direction for future research involves incorporating domain-specific constraints, leveraging contrastive learning strategies, or applying physico-chemically informed Fourier functions to improve the interpretability of KA-GNN's learned representations and their alignment with expert domain knowledge.

The properly designed Fourier components may have huge potential to effectively characterize and represent deep physical and chemical properties of molecules. From the physical and chemical perspective, quantum mechanics models, such as Schrodinger equation, density function theory, molecular orbital theory and so on, all involve

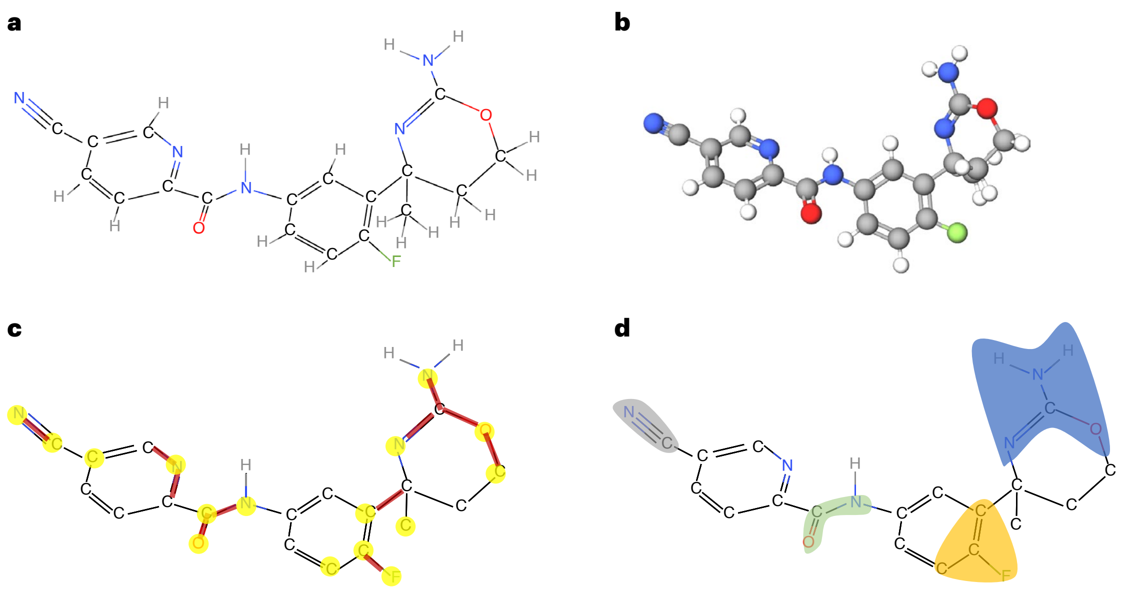

**Fig. 3 | KA-GAT interpretation of the molecule Fc1ccc(NC(=O)c2ncc(cc2)C#N)cc1[C@]1(N=C(OCC1)N)C. a**, Two-dimensional structure visualized using MolView. **b**, Three-dimensional structure generated by expanding the chiral centre [C@] and optimizing molecular geometry. **c**, Saliency map highlighting the atoms and bonds most influential to the model's prediction (a brighter colour indicates higher importance). **d**, Subgraph importance derived from a mask-based GNNExplainer, highlighting key functional groups influencing the prediction.

harmonic or wave-like properties that are mathematically represented as a series of Fourier-type functions. From the learning perspective, spherical harmonics have already been widely used in GDL models to characterize equivariant properties of molecules. These spherical harmonics basis functions are exactly Fourier-type functions. Finally, the Fourier components can be used in not only frequency perturbation analysis, but also characterizing various quantum descriptors such as highest occupied molecular orbital and lowest unoccupied molecular orbital gaps, atomic partial charges or symmetry-adapted basis functions. These studies may help to bridge the gap between graph representations and molecular functions and properties.

## Conclusion

In this study, we propose the non-trivial KAN-based GNNs. Our KA-GNNs use KAN's unique power to optimize GNN architectures at three core components, including node embedding, message passing and readout. We develop a Fourier-series-based KAN model and provide a rigorous theoretical justification of its approximation capability. We further compare different choices of pre-activation univariate functions in terms of both model performance and computational efficiency, and identify the Fourier series as the most effective choice. Extensive experiments on widely used molecular property prediction benchmarks demonstrate that our KA-GNNs consistently outperform traditional GNN models. This work not only highlights the potential of KA-GNNs for molecular tasks, but also contributes a GDL framework for general non-Euclidean data analysis.

## Methods

### Kolmogorov–Arnold networks and their theoretical background

The Kolmogorov–Arnold representation theorem (or superposition theorem) is a milestone in the field of real analysis and approximation theory[54]. It states that every multivariate continuous function can be represented as a superposition of the addition of continuous functions of one variable. This theorem not only solves Hilbert's 13th problem itself but also generalizes it to a broader form.

Vladimir Arnold and Andrey Kolmogorov's studies[54] prove that arbitrary multivariate continuous function $f$ can be written as a finite composition of continuous functions of a single variable and the binary operation of addition. More specifically,

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

where $n$ denotes the number of variables of the multivariate function $f$, and $\Phi_q:\mathbb{R}\to\mathbb{R}$ and $\phi_{q,p}:[0,1]\to\mathbb{R}$ are continuous single-variable functions.

**KAN.** Inspired by the Kolmogorov–Arnold representation theorem, one study[29] proposed a new deep learning architecture called the KAN as a promising alternative to the MLP. To enhance the representational capacity and leverage modern techniques (for example, backpropagation) for training the networks, KAN generalizes the setting in Kolmogorov–Arnold representation theorem:

- KAN does not adhere to the original depth-2 width $(2n+1)$ representation; instead, it generalizes to arbitrary widths and depths. Specifically, let the activation values in layer $l$ be denoted by $\mathbf{x}^{(l)}:=(x_1^{(l)},x_2^{(l)},\cdots,x_{n_l}^{(l)})$, where $n_l$ is the width of layer $l$. The activation value in layer $l+1$ is then simply the sum all incoming postactivations:

$$x_j^{(l+1)}=\sum_{i=1}^{n_l}\phi_{j,i}^{(l)}\left(x_i^{(l)}\right),\quad j=1,\ldots,n_{l+1}. \qquad (3)$$

  Here, $\phi_{j,i}^{(l)}$ for $i=1,\ldots,n_l$ and $j=1,\ldots,n_{l+1}$ are the pre-activation functions in layer $l$. The roles of these functions in KAN are analogous to the roles of the inner functions $\phi_{q,p}$ in equation (2).
- Although many constructive proofs of the Kolmogorov–Arnold representation theorem indicate that the inner functions $\phi_{q,p}$ in equation (2) are highly non-smooth[55,56], KAN opts for smooth functions as pre-activation functions $\phi_{j,i}^{(l)}$ to facilitate backpropagation. In ref. 29 the authors selected functions on the basis of B-splines.

**Fourier-series KAN model.** To optimize the network and avoid complex calculations, we propose to use Fourier series as the pre-activation functions for KAN[37] as in equation (3):

$$\phi_{j,i}^{(l)}(x)=\sum_{k=1}^{K}\left(A_{k,j,i}^{(l)}\cos(kx)+B_{k,j,i}^{(l)}\sin(kx)\right),\tag{4}$$

where $K$ is the number of harmonics, and $A_{k,j,i}^{(l)}$ and $B_{k,j,i}^{(l)}$ are learnable parameters initially sampled from a normal distribution $\mathcal{N}\left(0,\frac{1}{n_{l+1}\times K}\right)$.

Consequently, the activation value at the $j$th neuron in layer $l+1$ can be obtained by,

$$x_j^{(l+1)}=\sum_{i=1}^{n_l}\sum_{k=1}^{K}\left(A_{k,j,i}^{(l)}\cos\left(kx_i^{(l)}\right)+B_{k,j,i}^{(l)}\sin\left(kx_i^{(l)}\right)\right),\tag{5}$$

where $\mathbf{x}^{(l)}=(x_1^{(l)},x_2^{(l)},\ldots,x_{n_l}^{(l)})$ denotes the input vector of activation values in layer $l$ of the KAN, and $\mathbf{x}^{(l+1)}=(x_1^{(l+1)},x_2^{(l+1)},\ldots,x_{n_{l+1}}^{(l+1)})$ represents the output vector. Equation (5) can be concisely expressed as: $\mathbf{x}^{(l+1)}=\mathrm{KAN}_l(\mathbf{x}^{(l)})$, where $\mathrm{KAN}_l(\cdot)$ denotes the above KAN function in layer $l$. This network is integrated into GNNs, resulting in a architecture named KA-GNNs.

## KAN-based GNN models

In our KA-GNN models, a molecule is represented as a graph $G=(V,E)$, where $V$ denotes the set of nodes and $E$ denotes the set of edges. Each node $v\in V$ is associated with a feature vector $\mathbf{f}_v\in\mathbb{R}^{d_1}$, and each edge $uv\in E$ is associated with a feature vector $\mathbf{f}_{uv}\in\mathbb{R}^{d_2}$. Here, $uv$ denotes the edge that connecting node $u$ and $v$, and $d_1$ and $d_2$ denote the dimensions of node feature and edge feature, respectively. Each node represents an atom and an edge is formed among any two atoms if their distance is within a cut-off distance (in our model, we use cut-off distance as 5 Å). Figure 1a illustrates the complex interactions within the molecule, highlighting both the covalent bonds (solid lines) and non-covalent cut-off bonds (dashed lines) with distances less than 5 Å are considered in our KA-GNN model. Our atomic features, comprising atomic number, radius and electronegativity, are derived using Rdkit, following the approach in crystal graph convolutional neural networks[43] and path complex neural networks[57]. Each node $v\in V$ is associated with a feature vector $\mathbf{f}_v$, which is a 92-dimensional vector composed of one-hot encoded representations of atomic properties, following the approach described in ref. 43. Similarly, each edge $uv\in E$ is associated with a feature vector $\mathbf{f}_{uv}$, which is a 21-dimensional vector incorporating both chemical and geometrical information of the bond $uv$. The edges in our model can be classified into two types, that is, covalent bonds and non-covalent bonds, with different initial features. Detailed descriptions of the feature vectors are provided in the Supplementary Information.

**KA-GCN model.** 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 (for example, ReLU), KAN dynamically learns nonlinear transformations, allowing for more expressive feature propagation and improved adaptability to complex molecular graphs.

**(1) Node embedding initialization**

Instead of using a simple linear transformation, KA-GCN enhances initial node embeddings by incorporating both node and neighbourhood edge features through KAN:

$$h_v^{(0)}:=\mathrm{KAN}\left(\mathbf{f}_v\oplus\left(\frac{1}{|N(v)|}\sum_{u\in N(v)}\mathbf{f}_{vu}\right)\right),\tag{6}$$

where $N(v)$ represents the set of neighbouring nodes (excluding $v$ itself) with $|N(v)|$ the total number of the neighbouring nodes and $\oplus$ represents vector concatenation.

**(2) KAN-enhanced message passing**

At each layer $l$, KA-GCN refines node representations by aggregating information from neighbours using a residual KAN transformation:

$$h_v^{(l+1)}:=h_v^{(l)}+\mathrm{KAN}\left(h_v^{(l)}\oplus\left(\frac{1}{|N(v)|}\sum_{u\in N(v)}h_u^{(l)}\right)\right).\tag{7}$$

Here, KAN replaces the linear transformation in standard GCNs, dynamically adjusting feature interactions instead of applying a fixed-weight sum.

**(3) Readout and prediction**

After $L$ message-passing layers, we use KAN for graph-level representation aggregation:

$$\hat{y}:=\mathrm{KAN}\left(\frac{1}{|V|}\sum_{v\in V}h_v^{(L)}\right).$$

Compared with a simple MLP classifier, the KAN-based readout dynamically learns complex patterns in the molecular graph representation.

For classification tasks, we use the cross-entropy loss:

$$\mathcal{L}:=-\sum_i\left(y_i\log(\hat{y}_i)+(1-y_i)\log(1-\hat{y}_i)\right).$$

Overall, KA-GCN enhances expressiveness by allowing adaptive, data-driven feature transformations, addressing the rigidity of fixed activation functions in standard GCNs.

**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.

**(1) Adaptive initialization**

The initial embeddings for nodes and edges are enhanced using KAN layers:

$$\begin{aligned}h_v^{(0)}&:=\mathrm{KAN}\left(\mathbf{f}_v\oplus\left(\frac{1}{|N(v)|}\sum_{u\in N(v)}\mathbf{f}_{vu}\right)\right),\\ h_{vu}^{(0)}&:=\mathrm{KAN}\left(W_{hv}\mathbf{f}_v+W_e\mathbf{f}_{vu}+W_{tv}\mathbf{f}_u\right).\end{aligned}\tag{8}$$

KAN enables a nonlinear fusion of atomic and bond-level features, allowing more expressive initial representations. Here $W_{hv}$, $W_e$ and $W_{tv}$ are weight matrices to scale the input feature into same dimension.

**(2) KAN-enhanced attention mechanism**

Instead of a fixed attention computation, KA-GAT learns adaptive attention scores via:

$$\begin{aligned}z_v^{(l+1)}&:=Wh_v^{(l)},\quad z_u^{(l+1)}:=Wh_u^{(l)},\\ \alpha_{vu}&:=\mathrm{Softmax}\left(h_{vu}^{(l)}\right),\quad m_{vu}^{(l+1)}:=z_u^{(l+1)}\cdot\alpha_{vu},\\ m_v^{(l+1)}&:=z_v^{(l+1)}+\frac{1}{|N(v)|}\sum_{u\in N(v)}m_{vu}^{(l+1)}.\end{aligned}\tag{9}$$

Here $W$ is a value matrix. The use of KAN here enables adaptive, context-dependent attention scores, unlike standard GAT, which uses a fixed-weight attention mechanism.

**(3) KAN-based feature update**

After computing attention-based messages, we apply KAN for feature refinement:

$$h_v^{(l+1)}:=\mathrm{KAN}\left(m_v^{(l+1)}\right),\quad h_{vu}^{(l+1)}:=\mathrm{KAN}\left(h_{vu}^{(l)}+h_u^{(l)}+h_v^{(l)}\right).\tag{10}$$

This ensures that node and edge features are updated in a nonlinear, data-driven manner.

**(4) Readout and final prediction**

Similar to KA-GCN, KA-GAT uses KAN-based graph-level feature aggregation, allowing a flexible representation of the molecular graph.

### Key differences and contributions

KA-GNN seamlessly integrate KAN modules into the three fundamental components of GNNs, to notably enhance the model accuracy and interpretability.

- End-to-end KAN integration at all three core components: we systematically embed KAN modules into the entire GNN pipeline-including node embedding, message passing and readout, resulting in a unified, fully differentiable architecture with enhanced learning capacity.
- Fourier-based KAN formulation: we introduce a KAN variant based on the Fourier series, replacing traditional B-spline or RBF-based approaches. Our Fourier-KAN offers flexible function approximation, enabling the model to capture both smooth and highly nonlinear patterns in node features and attention weights, with provable universal approximation capabilities.
- Adaptive feature transformation and attention modelling: in KA-GAT, we replace conventional MLP-based attention mechanisms, with KAN modules, enabling more flexible modelling of both covalent and non-covalent interactions. Both KA-GCN and KA-GAT benefit from richer nonlinear transformation capabilities, surpassing ReLU-MLPs in expressiveness.
- Interpretability and domain alignment: in addition to accuracy gains, KA-GNN offers improved interpretability. As illustrated in Fig. 3, the model consistently highlights functionally important subgraphs, providing chemically meaningful explanations that outperform baseline GNNs in aligning with expert-curated knowledge.

## Data availability

All data used in this study are available via GitHub at https://github.com/LongLee220/KA-GNN and via Zenodo at https://doi.org/10.5281/zenodo.15003782 (ref. 58) or can be downloaded from the original dataset repository at https://moleculenet.org/datasets-1. Source data are provided with this paper.

## Code availability

All code is available via GitHub at https://github.com/LongLee220/KA-GNN and via Zenodo at https://doi.org/10.5281/zenodo.15003782 (ref. 58).

## References

1. Hughes, J. P., Rees, S., Kalindjian, S. B. & Philpott, K. L. Principles of early drug discovery. *Br. J. Pharmacol.* **162**, 1239–1249 (2011).
2. Zhang, Y. et al. Attention is all you need: utilizing attention in AI-enabled drug discovery. *Brief. Bioinform.* **25**, bbad467 (2024).
3. Chan, H., Shan, H., Dahoun, T., Vogel, H. & Yuan, S. Advancing drug discovery via artificial intelligence. *Trends Pharmacol. Sci.* **40**, 592–604 (2019).
4. Carpenter, K. & Huang, X. Machine learning-based virtual screening and its applications to Alzheimer's drug discovery: a review. *Curr. Pharm. Des.* **24**, 3347–3358 (2018).
5. Maia, E., Assis, L., De Oliveira, T., Da Silva, A. & Taranto, A. Structure-based virtual screening: from classical to artificial intelligence. *Front. Chem.* **8**, 343 (2020).
6. Zhao, T., Hu, Y., Valsdottir, L., Zang, T. & Peng, J. Identifying drug–target interactions based on graph convolutional network and deep neural network. *Brief. Bioinform.* **22**, 2141–2150 (2021).
7. Xia, J. et al. Understanding the limitations of deep models for molecular property prediction: Insights and solutions. *Adv. Neural Inf. Process. Syst.* **36**, 64774–64792 (2023).
8. Durán, C., Daminelli, S. & Thomas, J. Pioneering topological methods for network-based drug–target prediction by exploiting a brain-network self-organization theory. *Brief. Bioinform.* **19**, 1183–1202 (2018).
9. Nguyen, D. D., Cang, Z. & Wei, G.-W. A review of mathematical representations of biomolecular data. *Phys. Chem. Chem. Phys.* **22**, 4343–4367 (2020).
10. Bonner, S. et al. Implications of topological imbalance for representation learning on biomedical knowledge graphs. *Brief. Bioinform.* **23**, bbac279 (2022).
11. Meng, Z. & Xia, K. Persistent spectral-based machine learning (PerSpect ML) for protein–ligand binding affinity prediction. *Sci. Adv.* **7**, eabc5329 (2021).
12. Nguyen, D. & Wei, G. AGL-score: algebraic graph learning score for protein-ligand binding scoring, ranking, docking, and screening. *J. Chem. Inf. Model.* **59**, 3291–3304 (2019).
13. Szulc, N., Mackiewicz, Z. & Bujnicki, J. Structural interaction fingerprints and machine learning for predicting and explaining binding of small molecule ligands to RNA. *Brief. Bioinform.* **24**, bbad187 (2023).
14. Chen, J., Wang, R., Wang, M. & Wei, G. Mutations strengthened SARS-CoV-2 infectivity. *J. Mol. Biol.* **432**, 5212–5226 (2020).
15. Wang, R., Hozumi, Y., Yin, C. & Wei, G. Mutations on COVID-19 diagnostic targets. *Genomics* **112**, 5204–5213 (2020).
16. Gao, K., Nguyen, D., Tu, M. & Wei, G. Generative network complex for the automated generation of drug-like molecules. *J. Chem. Inf. Model.* **60**, 5682–5698 (2020).
17. Zhao, B.-W. et al. A geometric deep learning framework for drug repositioning over heterogeneous information networks. *Brief. Bioinform.* **23**, bbac384 (2022).
18. Li, P., Li, Y. & Hsieh, C. TrimNet: learning molecular representation from triplet messages for biomedicine. *Brief. Bioinform.* **22**, bbaa266 (2021).
19. Li, X.-S. et al. Multiphysical graph neural network (MP-GNN) for COVID-19 drug design. *Brief. Bioinform.* **23**, bbac231 (2022).
20. Kang, C., Zhang, H. & Liu, Z. LR-GNN: a graph neural network based on link representation for predicting molecular associations. *Brief. Bioinform.* **23**, bbab513 (2022).
21. Cai, H., Zhang, H., Zhao, D., Wu, J. & Wang, L. FP-GNN: a versatile deep learning architecture for enhanced molecular property prediction. *Brief. Bioinform.* **23**, bbac408 (2022).
22. Liu, H., Huang, Y., Liu, X. & Deng, L. Attention-wise masked graph contrastive learning for predicting molecular property. *Brief. Bioinform.* **23**, bbac303 (2022).
23. Zhang, R., Lin, Y. & Wu, Y. MvMRL: a multi-view molecular representation learning method for molecular property prediction. *Brief. Bioinform.* **25**, bbae298 (2024).
24. Mercado, R. et al. Graph networks for molecular design. *Mach. Learn. Sci. Technol.* **2**, 025023 (2021).
25. Liu, Q., Allamanis, M., Brockschmidt, M. & Gaunt, A.L. Constrained graph variational autoencoders for molecule design. In *Proc. 32nd International Conference on Neural Information Processing Systems* 7806–7815 (Curran Associates, 2018).
26. Rong, Y. et al. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **33**, 12559–12571 (2020).
27. Wang, M., Cang, Z. & Wei, G. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. *Nat. Mach. Intell.* **2**, 116–123 (2020).
28. Shen, C., Luo, J. & Xia, K. Molecular geometric deep learning. *Cell Rep. Methods* **3**, 100621 (2023).
29. Liu, Z. et al. KAN: Kolmogorov–Arnold networks. In *Proc. 13th International Conference on Learning Representations* 24–28 (OpenReview, 2025).
30. Bresson, R. et al. KAGNNs: Kolmogorov–Arnold networks meet graph learning. *Trans. Mach. Learn. Res.* (2025).
31. Koenig, B., Kim, S. & Deng, S. KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics. *Comput. Methods Appl. Mech. Eng.* **432**, 117397 (2024).

32. Liu, M., Bian, S., Zhou, B. & Lukowicz, P. ikan: Global incremental learning with kan for human activity recognition across heterogeneous datasets. In *Proc. International Symposium on Wearable Computers* 89–95 (ACM, 2024).
33. Genet, R. & Inzirillo, H. Tkan: temporal Kolmogorov–Arnold networks. Preprint at https://arxiv.org/abs/2405.07344 (2024).
34. Cheon, M. Kolmogorov–Arnold network for satellite image classification in remote sensing. Preprint at https://arxiv.org/abs/2406.00600 (2024).
35. Bozorgasl, Z. & Chen, H. Wav-kan: wavelet Kolmogorov–Arnold networks. Preprint at https://arxiv.org/abs/2405.12832 (2024).
36. Kashefi, A. PointNet with KAN versus PointNet with MLP for 3D classification and segmentation of point sets. Preprint at https://arxiv.org/abs/2410.10084 (2024).
37. Ahmed, T. & Sifat, M. GraphKAN: graph Kolmogorov Arnold network for small molecule-protein interaction predictions. In *Proc. Workshop on Machine Learning for Life and Material Science: From Theory to Industry Applications* (ICML, 2024).
38. Kiamari, M., Kiamari, M. & Krishnamachari, B. GKAN: graph Kolmogorov–Arnold networks. Preprint at https://arxiv.org/abs/2406.06470 (2024).
39. Zhang, F. & Zhang, X. Graphkan: enhancing feature extraction with graph Kolmogorov Arnold networks. Preprint at https://arxiv.org/abs/2406.13597 (2024).
40. Li, R., Li, M., Liu, W. & Chen, H. GNN-SKAN: harnessing the power of SwallowKAN to advance molecular representation learning with GNNs. Preprint at https://arxiv.org/abs/2408.01018 (2024).
41. Carleson, L. On convergence and growth of partial sums of Fourier series. *Acta Math.* **116**, 135–157 (1966).
42. Fefferman, C. On the convergence of multiple Fourier series. *Bull. Am. Math. Soc.* **77**, 744–745 (1971).
43. Xie, T. & Grossman, J. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.* **120**, 145301 (2018).
44. Wu, Z. et al. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
45. Xiong, Z. et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J. Med. Chem.* **63**, 8749–8760 (2019).
46. Yang, K. et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
47. Liu, S., Demirel, M.F. & Liang, Y. N-gram graph: simple unsupervised representation for graphs, with applications to molecules. In *Proc. 33rd International Conference on Neural Information Processing Systems* 8464–8476 (Curran Associates, 2019).
48. Hu, W. et al. Strategies for pre-training graph neural networks. In *Proc. International Conference on Learning Representations* (ICLR, 2019).
49. Liu, S. et al. Pre-training molecular graph representation with 3D geometry. In *Proc. International Conference on Learning Representations* (ICLR, 2022).
50. Wang, Y., Wang, J., Cao, Z. & Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
51. Fang, X. et al. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell.* **4**, 127–134 (2022).
52. Zhou, G. et al. Uni-mol: a universal 3D molecular representation learning framework. In *Proc. Eleventh International Conference on Learning Representations* (ICLR, 2023).
53. Li, Y., Wang, W., Liu, J. & Wu, C. Pre-training molecular representation model with spatial geometry for property prediction. *Comput. Biol. Chem.* **109**, 108023 (2024).
54. Kolmogorov, A. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk* **114**, 953–956 (1957).
55. Sprecher, D. A numerical implementation of Kolmogorov's superpositions. *Neural Netw.* **9**, 765–772 (1996).
56. Braun, J. & Griebel, M. On a constructive proof of Kolmogorov's superposition theorem. *Constr. Approx.* **30**, 653–675 (2009).
57. Li, L., Liu, X. & Wang, G. Path complex neural network for molecular property prediction. In *Proc. Workshop on Geometry-grounded Representation Learning and Generative Modeling* (ICML, 2024).
58. Longlong Li, Y. Z. & Xia, K. KA-GNN: Kolmogorov–Arnold graph neural networks for molecular property prediction. *Zenodo* https://doi.org/10.5281/zenodo.15003782 (2025).

## Acknowledgements

This work was supported in part by the Singapore Ministry of Education Academic Research Fund Tier 1 grant number RG16/23 (to K.X.), Tier 2 grant numbers MOE-T2EP20120-0010 (to K.X.) and MOE-T2EP20221-0003 (to K.X.) and the Program of China Scholarship Council grant number 202306220143 (to L.L.).

## Author contributions

K.X. conceived and designed the study. L.L. performed the calculations. G.W. provided computational resources. Y.Z. carried out the theoretical analysis. L.L., Y.Z. and K.X. jointly contributed to writing and revising the paper.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s42256-025-01087-7.

**Correspondence and requests for materials** should be addressed to Kelin Xia.

**Peer review information** *Nature Machine Intelligence* thanks the anonymous reviewers for their contribution to the peer review of this work.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.