

---

# KA-GNN: KOLMOGOROV-ARNOLD GRAPH NEURAL NETWORKS FOR MOLECULAR PROPERTY PREDICTION

---

Longlong Li<sup>1,2,3</sup>, Yipeng Zhang<sup>3</sup>, Guanghui Wang<sup>1</sup>, and Kelin Xia<sup>3</sup>

<sup>1</sup>School of Mathematics, Shandong University, Jinan 250100, China

<sup>2</sup>Data Science Institute, Shandong University, Jinan 250100, China

<sup>3</sup>Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Singapore

Emails: longlee@mail.sdu.edu.cn, yipeng001@e.ntu.edu.sg, ghwang@sdu.edu.cn, xiakelin@ntu.edu.sg

## ABSTRACT

As key models in geometric deep learning, graph neural networks have demonstrated enormous power in molecular data analysis. Recently, a specially-designed learning scheme, known as Kolmogorov-Arnold Network (KAN), shows unique potential for the improvement of model accuracy, efficiency, and explainability. Here we propose the first non-trivial Kolmogorov-Arnold Network-based Graph Neural Networks (KA-GNNs), including KAN-based graph convolutional networks(KA-GCN) and KAN-based graph attention network (KA-GAT). The essential idea is to utilizes KAN's unique power to optimize GNN architectures at three major levels, including node embedding, message passing, and readout. Further, with the strong approximation capability of Fourier series, we develop Fourier series-based KAN model and provide a rigorous mathematical prove of the robust approximation capability of this Fourier KAN architecture. To validate our KA-GNNs, we consider seven most-widely-used benchmark datasets for molecular property prediction and extensively compare with existing state-of-the-art models. It has been found that our KA-GNNs can outperform traditional GNN models. More importantly, our Fourier KAN module can not only increase the model accuracy but also reduce the computational time. This work not only highlights the great power of KA-GNNs in molecular property prediction but also provides a novel geometric deep learning framework for the general non-Euclidean data analysis.

**Keywords** Kolmogorov-Arnold Network, Fourier series, Graph Neural Network, Molecular Property Prediction

## 1 Introduction

Drug development is a complex and costly process, typically requiring decades of time and substantial investment [1]. In this challenging landscape, artificial intelligence (AI) has become particularly valuable, significantly impacting the prediction of molecular properties and showing immense promise in drug design [2, 3]. AI has greatly advanced virtual screening processes, potentially reducing the time and investment required [4, 5, 6]. AI-based molecular models, which drive these advancements, generally fall into two categories: those based on molecular descriptors and end-to-end deep learning models [7].

The first category relies on molecular descriptors or fingerprints as input features for machine learning algorithms. This process, known as featurization or feature engineering, involves not only capturing physical, chemical, and biological properties but also incorporating a wide array of fingerprints based on molecular structure information. Among these, structure-based fingerprints, particularly those derived from topological data analysis methods, have proven highly effective in molecular representation and featurization [8, 9, 10, 11]. The integration of these fingerprints with learning models has achieved significant success in various stages of drug design, including protein-ligand binding affinity prediction [12, 13], protein mutation analysis [14, 15], and drug design [16, 17], among other areas.

The second category includes end-to-end deep learning models that utilize various molecular representations such as Simplified Molecular Input Line Entry System (SMILES) strings, images, or molecular graphs, and deploy architectures like Transformers, Convolutional Neural Networks (CNNs), and Graph Neural Networks (GNNs) for prediction [18, 19, 20, 21, 22, 23]. Among these, molecular graphs based on covalent bonds are predominantly employed to describe molecular topology. Geometric deep learning models based on these molecular graphs, such as Graph Convolutional Networks (GCNs) [24], graph autoencoders [25], and graph transformers [26], have been extensively used in molecular data analysis and drug design. Additionally, recent research has demonstrated that molecular descriptors based on non-covalent interactions perform exceptionally well in predicting protein-ligand and protein-protein binding affinities [27]. These observations imply that new geometric molecular graph representations could surpass traditional covalent-bond-based graphs. By integrating these geometry-based molecular graphs into Geometric Deep Learning (GDL) models, it is possible to enhance model performance and deepen the understanding of molecular geometry [28].

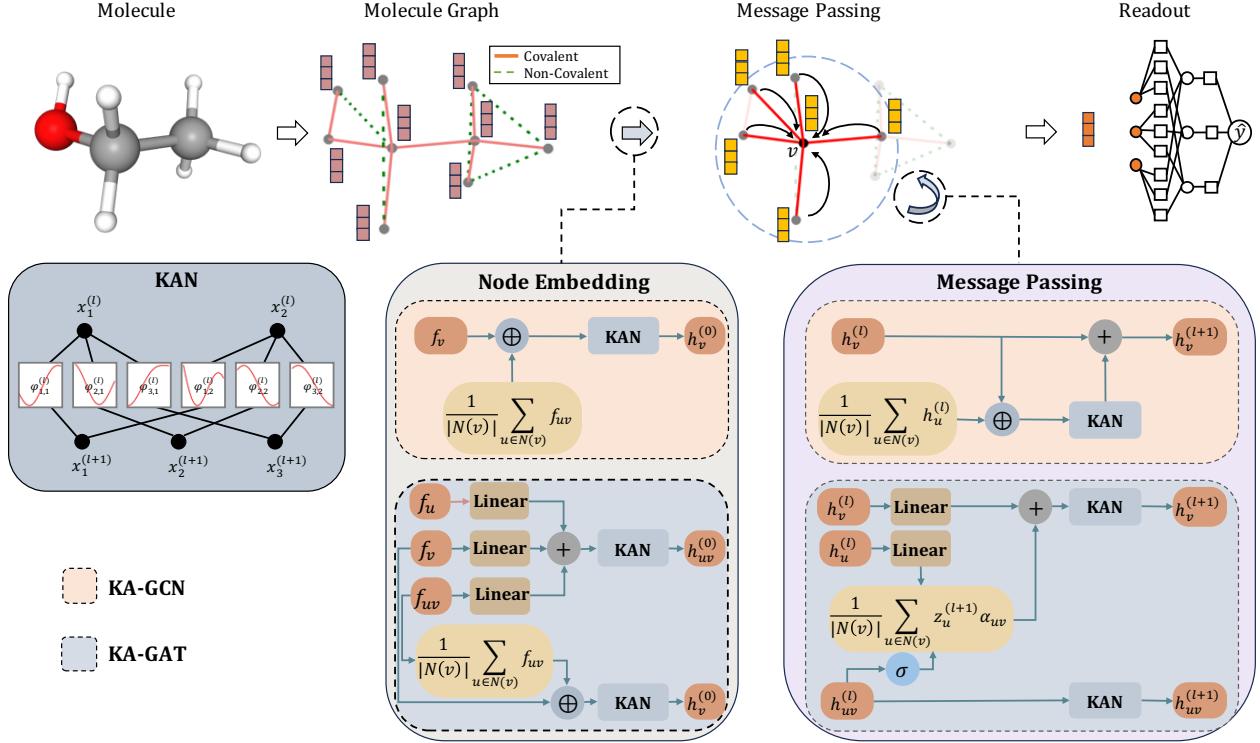


Figure 1: Overview of the KA-GNN model architecture. The flowchart illustrates the modified components within the GNN: node embedding, message-passing, pooling and prediction modules.

Kolmogorov-Arnold Networks (KANs), which are based on the Kolmogorov-Arnold representation theorem, are increasingly recognized as a potent alternative to Multi-layer Perceptrons (MLPs). KANs distinguished by their unique architecture that employs different learnable activation functions, eliminate traditional linear weight matrices and enhance model accuracy and efficiency, particularly in solving partial differential equations, as described by Liu et al. [29]. Recent research highlights the versatility and adaptability of KANs across various domains [30, 31, 32, 33]. One notable application is the integration of KANs with existing neural network models to enhance performance. For instance, Genet et al. [33] significantly improved multi-step time series forecasting by integrating KANs with Long Short-Term Memory networks (LSTMs). Cheon et al. [34] effectively classified remote sensing scenes by merging KANs with pre-trained Convolutional Neural Network (CNN) models. Kiamari et al. [35] demonstrated their multifunctionality by incorporating KANs into Graph Convolutional Networks (GCNs) for semi-supervised graph learning tasks. Furthermore, adaptations in the base functions of KANs to better suit neural network applications have led to significant enhancements. Bozorgasl et al. [36] improved the interpretability and performance of KANs by incorporating wavelet functions that more effectively capture the frequency components of data. Tashin et al. [37] employed KANs with Fourier transform basis functions for feature transformation before GNN processing, validating their utility in Small Molecule-Protein Interaction Predictions. Li et al. [38] used adaptive Radial Basis Functions

(RBFs) in KANs to enhance feature updating and replace MLPs in the prediction phase, demonstrating robust integration capabilities with various neural network frameworks. Kashefi et al. [39] applied Jacobi polynomials to design KAN layers for GNNs, effectively predicting fluid fields on irregular geometries. These advancements underscore the significant potential of KANs to refine neural network architectures. The ongoing exploration of how to further enhance KANs and integrate it into node feature embedding, message passing in different GNN frameworks, and the prediction phase remains a vital area of research.

In this paper, we introduce the first non-trivial Kolmogorov-Arnold Network-based Graph Neural Networks (KA-GNNs), including KAN-based convolutional networks (KA-GCN) and KAN-based graph attention network (KA-GAT). Figure 1 outlines the general KA-GNNs architecture. Different from all the previous trivial KAN-based GNN models, which only replace the MLP in the readout part with a standard KAN module, we utilize KAN to optimize GNN architectures at three major levels, including node embedding, message passing, and readout. Further, a Fourier series-based KAN model has been developed and a rigorous theoretical prove of its robust approximation capability is given. Based on seven benchmark datasets, we have extensively validated our KA-GNNs and compared with state-of-the-art models. Our KA-GNNs have achieved both great accuracy and efficiency, providing a novel geometric deep learning framework for analyzing general non-Euclidean data.

## 2 Results

### 2.1 Kolmogorov-Arnold Network (KAN)

**Kolmogorov-Arnold Representation Theorem** The Kolmogorov–Arnold Representation Theorem (or Superposition Theorem) is a milestone in the field of real analysis and approximation theory. It states that every multivariate continuous function can be represented as superposition of the addition of continuous functions of one variable. This theorem not only solves Hilbert’s thirteenth problem itself but also generalizes it to a broader form.

Vladimir Arnold and Andrey Kolmogorov’s works [40] 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,

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

where  $n$  denote the number of variables of the function  $f$ .  $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$  and  $\phi_{q,p} : [0, 1] \rightarrow \mathbb{R}$  are continues function.

#### 2.1.1 Kolmogorov-Arnold Network (KAN)

Inspired by the Kolmogorov-Arnold representation theorem, Liu et al. [29] proposed a new deep learning architecture called the Kolmogorov-Arnold Network (KAN) as a promising alternative to the Multi-Layer Perceptron (MLP). To enhance the KAN’s representational capacity and leverage modern techniques (e.g., backpropagation) for training the networks, KAN transcends several limitations of the 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  $\mathbf{x}^{(l)} := (x_1^{(l)}, x_2^{(l)}, \dots, x_{n_l}^{(l)})$ , where  $n_l$  is the width of layer  $l$ . The activation value in layer  $l + 1$  is then simply the sum of all incoming post-activations:

$$x_j^{(l+1)} = \sum_{i=1}^{n_l} \phi_{j,i}^{(l)}(x_i^{(l)}), \quad j = 1, \dots, n_{l+1}. \quad (2)$$

Here,  $\phi_{j,i}^{(l)}$  for  $i = 1, \dots, n_l$  and  $j = 1, \dots, n_{l+1}$  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 1.

- Although many constructive proofs of the Kolmogorov-Arnold representation theorem indicate that the inner functions  $\phi_{q,p}$  in equation 1 are highly non-smooth [41, 42], KAN opts for smooth functions as pre-activation functions  $\phi_{j,i}^{(l)}$  to facilitate backpropagation. Liu et al. [29] selected functions based on B-splines.

### 2.1.2 Fourier-series KAN model

To optimize the network and avoid complex calculations, we propose to utilize Fourier series as the pre-activation functions for KAN [37] as in Eq.(2):

$$\phi_{j,i}^{(l)}(x) = \sum_{k=1}^K \left( A_{k,j,i}^{(l)} \cos(kx) + B_{k,j,i}^{(l)} \sin(kx) \right), \quad (3)$$

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  $N\left(0, \frac{1}{n_{l+1} \times K}\right)$ .

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

$$x_j^{(l+1)} = \sum_{i=1}^{n_l} \sum_{k=1}^K \left( A_{k,j,i}^{(l)} \cos(kx_i^{(l)}) + B_{k,j,i}^{(l)} \sin(kx_i^{(l)}) \right), \quad (4)$$

where  $\mathbf{x}^{(l)} = (x_1^{(l)}, x_2^{(l)}, \dots, x_{n_l}^{(l)})$  denotes the input vector of activation values in layer  $l$  of the KAN, and  $\mathbf{x}^{(l+1)} = (x_1^{(l+1)}, x_2^{(l+1)}, \dots, x_{n_{l+1}}^{(l+1)})$  represents the output vector. Eq (4) can be concisely expressed as:  $\mathbf{x}^{(l+1)} = \text{KAN}_l(\mathbf{x}^{(l)})$ , where  $\text{KAN}_l(\cdot)$  denotes the above KAN function in layer  $l$ . This network is integrated into a graph neural network, culminating in a novel architecture named KA-GNNs, which is employed for molecular property prediction tasks.

Given that the inner functions in Eq (1) of the Kolmogorov-Arnold representation theorem can exhibit significant non-smoothness, we do not use this theorem as the foundational theory for showing the approximation capability of our model. Instead, we base our approach on the extension of Carleson's theorem [43] regarding the convergence of Fourier series for multivariable functions, as proved by Fefferman in [44]:

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

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

where  $x \in [0, 2\pi]^n$  and  $L^2([0, 2\pi]^n)$  denotes the space of square-integrable functions on  $[0, 2\pi]^n$ , which consists of all functions  $f$  such that  $\int_{[0, 2\pi]^n} |f(\vec{\mathbf{x}})|^2 d\vec{\mathbf{x}} < \infty$ . We have

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

almost everywhere.

The above theorem demonstrates the strong approximation capability of Fourier series, which is why we adopt Fourier series as the foundational basis for our model. Therefore, we retain the architecture of the Kolmogorov-Arnold network but replace the pre-activation functions with Fourier series. We can further prove that this new KAN architecture provides the potential of robust approximation capability:

**Theorem 2** Let  $f \in L^2([0, 2\pi]^n)$  be a square-integrable function. For almost every  $\vec{\mathbf{x}} \in [0, 2\pi]^n$  and for any  $\epsilon > 0$ , there exist a positive integer  $K$  and a sequence of Fourier-based KAN functions  $\text{KAN}_l$  at multiple layers  $l = 0, 1, \dots, L$ , such that the number of harmonics in the pre-activation functions of these KAN functions does not exceed  $K$ , and

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

where  $\circ$  denotes function composition.

In summary, our model extends the Kolmogorov-Arnold Network (KAN) architecture by incorporating Fourier series as pre-activation functions to enhance approximation capability.

## 2.2 Kolmogorov-Arnold Network-based Graph Neural Network 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 represents an atom and an edge is formed among any two atoms if their distance is

Table 1: Features Type and Description in KA-GNN models. Here we list both atom, covalent bond, and non-covalent bond features.

Features Type		Description	Type	Size
Atom	CGCNN	Atomic number, Radius, and electronegativity	One-Hot	92
Covalent Bond	Bond Directionality	None, Beginwedge, Begindash, etc.	One-Hot	7
	Bond Type	Single, Double, Triple, or Aromatic.	One-Hot	4
	Bond Length	Numerical and square length of the bond.	Float	2
	In Ring	Indicates if the bond is part of a chemical ring.	One-Hot	2
Non-Covalent Bond	Atom charges	Atoms charges in Molecular ( $q_i, q_j, q_i \cdot q_j$ )	Float	3
	Distance between atoms	Distance between atoms ( $\frac{1}{d_{ij}}, \frac{1}{d_{ij}^6}, \frac{1}{d_{ij}^{12}}$ )	Float	3

within a cutoff distance (in our model, we use cutoff distance as 5 Å). Figure 1 **A** 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 angstroms are considered in our KA-GNN model. Our atomic features—comprising atomic number, radius, and electronegativity—are derived using Rdkit, following the approach in CGCNN [45] and PCNN [46]. Each node  $v \in V$  is associated with a feature vector  $f_v$ , which is a 92-dimensional vector composed of one-hot encoded representations of atomic properties, following the approach described in [45]. Similarly, each edge  $uv \in E$  is associated with a feature vector  $f_{uv}$ , 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, i.e., covalent bonds and non-covalent bonds, with different initial features. Detailed descriptions of the feature vectors are provided in Table 1.

### 2.2.1 KA-GCN Model

Our KA-GCN architecture has three major steps, including node embedding, message-passing, and readout. First, one KAN layer is employed in the initialization of node embeddings  $h_v^{(0)}, v \in V$  as follows,

$$h_v^{(0)} := \text{KAN} \left( f_v \oplus \left( \frac{1}{|N(v)|} \sum_{u \in N(v)} f_{uv} \right) \right), \quad (5)$$

where  $N(v)$  represents the set of the neighboring nodes of node  $v$  (excluding  $v$  itself),  $|N(v)|$  is the number of neighboring nodes, and  $\oplus$  represents vector concatenation.

Second, one or several KAN layers are employed in the message passing module. At the layer  $l$ , the KAN-based message passing module can be expressed as,

$$h_v^{(l+1)} := h_v^{(l)} + \text{KAN} \left( h_v^{(l)} \oplus \left( \frac{1}{|N(v)|} \sum_{u \in N(v)} h_u^{(l)} \right) \right). \quad (6)$$

Computationally, the first KAN layer maps initial feature vector  $f_v$  from dimension 113 to updated feature vector  $h_v$  with dimension 64. After that, the feature dimension will stay the same in the later message-passing process.

Third, one or several KAN layers are employed in the readout module. After  $L$ -th the message passing phases, we apply average pooling across all node features to obtain a latent vector for the molecular graph  $G$ . An one-layer or two-layer KAN model is employed in readout module for the final prediction as follows,

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

The cross-entropy is used as the loss function as follows,

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

where  $y_i$  denotes the actual label, and  $\hat{y}_i$  denotes the predicted label.

## 2.2.2 KA-GAT Model

Similar to the KA-GCN architecture, the KA-GAT model initializes the node and edge embeddings,  $h_v^{(0)}$  and  $h_{uv}^{(0)}$ , using a KAN layer as follows:

$$\begin{aligned} h_v^{(0)} &:= \text{KAN} \left( f_v \oplus \left( \frac{1}{|N(v)|} \sum_{u \in N(v)} f_{uv} \right) \right), \\ h_{uv}^{(0)} &:= \text{KAN} (\text{Linear}(f_v) + \text{Linear}(f_{uv}) + \text{Linear}(f_u)) \end{aligned} \quad (7)$$

Subsequently, the KA-GAT model uses these initial embeddings to compute attention scores that guide the aggregation of messages:

$$\begin{aligned} z_v^{(l+1)} &:= \text{Linear} (h_v^{(l)}) , z_u^{(l+1)} := \text{Linear} (h_u^{(l)}) , \\ \alpha_{uv} &:= \text{Softmax} (h_{uv}^{(l)}) , m_{uv}^{(l+1)} := z_u^{(l+1)} \cdot \alpha_{uv} \\ m_v^{(l+1)} &:= z_v^{(l+1)} + \frac{1}{|N(v)|} \left( \sum_{u \in N(v)} m_{uv}^{(l+1)} \right) \end{aligned} \quad (8)$$

Then, one or more KAN layers are used to update the node and edge features:

$$h_v^{(l+1)} := \text{KAN}(m_v^{(l+1)}), \quad h_{uv}^{(l+1)} := \text{KAN}(h_{uv}^{(l)}) \quad (9)$$

Finally, after message passing, the KA-GAT model employs the same readout module as used in the KA-GCN.

## 2.3 KA-GNNs for Molecular Property Analysis

### 2.3.1 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 [47]. Of these, three datasets—MUV, HIV, and BACE—are from the biophysics domain. The rest four datasets are from the physiology domain, including BBBP, Tox21, SIDER, and ClinTox.

In our comparative analysis across seven datasets, we evaluated a range of geometric deep learning models, including AttentiveFP [48], D-MPNN [49], Mol-GDL [28], N-Gram [50], PretrainGNN [51], GROVER [26], GraphMVP [52], MolCLR [53], GEM [54], Uni-mol [55], SMPT [56], and GNN-SKAN [38]. Additionally, we included our KAN-based GNN models, such as the GNN-SKAN [38]. Further details on these models can be found in Subsection Baseline Models.

The comparative results in Table 2 further confirm the superiority of the KA-GNNs model. 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 AUC for KA-GCN and KA-GAT showed improvements of approximately 7.95% and 7.68%.

### 2.3.2 Importance of Fourier series for KA-GNNs

In this research, we employed a Fourier-based Kolmogorov-Arnold Network (KAN) as the primary computational mechanism within our GNN architecture, which differs significantly from traditional MLP-based approaches. To assess the impact of different base functions on the prediction of molecular properties, we explored several base function including B-spline, polynomial, Fourier-series, in the KA-GCNs architecture. These models were tested within the traditional GCN framework as well as the GAT framework. The comparative results are summarized in Table 3. The Polynomial-base function can be expressed in Eq (10).

$$x_j^{(l+1)} = \sum_{i=1}^{n_l} \sum_{k=1}^K \left( C_{k,j,i}^{(l)} (x_i^{(l)})^k \right), \quad (10)$$

Table 2: Comparison of KA-GNNs with state-of-the-art geometric deep learning models in different molecular property datasets. The performance metric is the average of Receiver Operating Characteristic - Area Under the Curve (ROC-AUC). The standard deviation results are denoted as subscripts. The best model for each category is highlighted in bold, while the second-best performance is marked with underline. The notation - indicates that the results were not reported.

Model	BACE	BBBP	ClinTox	SIDER	Tox21	HIV	MUV
No. molecules	1513	2039	1478	1427	7831	41127	93808
No. avg atoms	65	46	50.58	65	36	46	43
No. tasks	1	1	2	27	12	1	17
D-MPNN [49]	<b>0.809</b> <sub>(0.006)</sub>	<b>0.710</b> <sub>(0.003)</sub>	<b>0.906</b> <sub>(0.007)</sub>	<b>0.570</b> <sub>(0.007)</sub>	<b>0.759</b> <sub>(0.007)</sub>	<b>0.771</b> <sub>(0.005)</sub>	<b>0.786</b> <sub>(0.014)</sub>
AttentiveFP [48]	<b>0.784</b> <sub>(0.022)</sub>	<b>0.663</b> <sub>(0.018)</sub>	<b>0.847</b> <sub>(0.003)</sub>	<b>0.606</b> <sub>(0.032)</sub>	<b>0.781</b> <sub>(0.005)</sub>	<b>0.757</b> <sub>(0.014)</sub>	<b>0.786</b> <sub>(0.015)</sub>
N-GramRF [50]	<b>0.779</b> <sub>(0.015)</sub>	<b>0.697</b> <sub>(0.006)</sub>	<b>0.775</b> <sub>(0.040)</sub>	<b>0.668</b> <sub>(0.007)</sub>	<b>0.743</b> <sub>(0.009)</sub>	<b>0.772</b> <sub>(0.004)</sub>	<b>0.769</b> <sub>(0.002)</sub>
N-GramXGB [50]	<b>0.791</b> <sub>(0.013)</sub>	<b>0.691</b> <sub>(0.008)</sub>	<b>0.875</b> <sub>(0.027)</sub>	<b>0.655</b> <sub>(0.007)</sub>	<b>0.758</b> <sub>(0.009)</sub>	<b>0.787</b> <sub>(0.004)</sub>	<b>0.748</b> <sub>(0.002)</sub>
PretrainGNN [51]	<b>0.845</b> <sub>(0.007)</sub>	<b>0.687</b> <sub>(0.013)</sub>	<b>0.726</b> <sub>(0.015)</sub>	<b>0.627</b> <sub>(0.008)</sub>	<b>0.781</b> <sub>(0.006)</sub>	<b>0.799</b> <sub>(0.007)</sub>	<b>0.813</b> <sub>(0.021)</sub>
GROVE_base [26]	<b>0.821</b> <sub>(0.007)</sub>	<b>0.700</b> <sub>(0.001)</sub>	<b>0.812</b> <sub>(0.030)</sub>	<b>0.648</b> <sub>(0.006)</sub>	<b>0.743</b> <sub>(0.001)</sub>	<b>0.625</b> <sub>(0.009)</sub>	<b>0.673</b> <sub>(0.018)</sub>
GROVE_large [26]	<b>0.810</b> <sub>(0.014)</sub>	<b>0.695</b> <sub>(0.001)</sub>	<b>0.762</b> <sub>(0.037)</sub>	<b>0.654</b> <sub>(0.001)</sub>	<b>0.735</b> <sub>(0.001)</sub>	<b>0.682</b> <sub>(0.011)</sub>	<b>0.673</b> <sub>(0.018)</sub>
GraphMVP [52]	<b>0.812</b> <sub>(0.009)</sub>	<b>0.724</b> <sub>(0.016)</sub>	<b>0.791</b> <sub>(0.028)</sub>	<b>0.639</b> <sub>(0.012)</sub>	<b>0.759</b> <sub>(0.005)</sub>	<b>0.770</b> <sub>(0.012)</sub>	<b>0.777</b> <sub>(0.006)</sub>
MoICLR [53]	<b>0.824</b> <sub>(0.009)</sub>	<b>0.722</b> <sub>(0.021)</sub>	<b>0.912</b> <sub>(0.035)</sub>	<b>0.589</b> <sub>(0.014)</sub>	<b>0.750</b> <sub>(0.002)</sub>	<b>0.781</b> <sub>(0.005)</sub>	<b>0.796</b> <sub>(0.019)</sub>
GEM [54]	<b>0.856</b> <sub>(0.011)</sub>	<b>0.724</b> <sub>(0.004)</sub>	<b>0.901</b> <sub>(0.013)</sub>	<b>0.672</b> <sub>(0.004)</sub>	<b>0.781</b> <sub>(0.001)</sub>	<b>0.806</b> <sub>(0.009)</sub>	<b>0.817</b> <sub>(0.005)</sub>
Mol-GDL [28]	<b>0.863</b> <sub>(0.019)</sub>	<b>0.728</b> <sub>(0.019)</sub>	<b>0.966</b> <sub>(0.002)</sub>	<b>0.831</b> <sub>(0.002)</sub>	<b>0.794</b> <sub>(0.005)</sub>	<b>0.808</b> <sub>(0.007)</sub>	<b>0.675</b> <sub>(0.014)</sub>
Uni-mol[55]	<b>0.857</b> <sub>(0.002)</sub>	<b>0.729</b> <sub>(0.006)</sub>	<b>0.919</b> <sub>(0.018)</sub>	<b>0.659</b> <sub>(0.013)</sub>	<b>0.796</b> <sub>(0.005)</sub>	<b>0.808</b> <sub>(0.003)</sub>	<b>0.821</b> <sub>(0.013)</sub>
SMPT[56]	<b>0.873</b> <sub>(0.015)</sub>	<b>0.734</b> <sub>(0.003)</sub>	<b>0.927</b> <sub>(0.002)</sub>	<b>0.676</b> <sub>(0.050)</sub>	<b>0.797</b> <sub>(0.001)</sub>	<b>0.812</b> <sub>(0.001)</sub>	<b>0.822</b> <sub>(0.008)</sub>
GNN-SKAN [38]	<b>0.747</b> <sub>(0.009)</sub>	<b>0.676</b> <sub>(0.014)</sub>	-	<b>0.614</b> <sub>(0.005)</sub>	<b>0.747</b> <sub>(0.005)</sub>	<b>0.786</b> <sub>(0.015)</sub>	-
<b>KA-GCN</b>	<b>0.890</b> <sub>(0.014)</sub>	<b>0.787</b> <sub>(0.014)</sub>	<b>0.989</b> <sub>(0.003)</sub>	<b>0.842</b> <sub>(0.001)</sub>	<b>0.799</b> <sub>(0.005)</sub>	<b>0.821</b> <sub>(0.005)</sub>	<b>0.834</b> <sub>(0.009)</sub>
<b>KA-GAT</b>	<b>0.884</b> <sub>(0.004)</sub>	<b>0.785</b> <sub>(0.021)</sub>	<b>0.991</b> <sub>(0.005)</sub>	<b>0.847</b> <sub>(0.002)</sub>	<b>0.800</b> <sub>(0.006)</sub>	<b>0.823</b> <sub>(0.002)</sub>	<b>0.834</b> <sub>(0.010)</sub>

Table 3: Comparison of the performance of KA-GNN (and KA-GAT) models with three types of base functions, 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	<b>0.771</b> <sub>(0.012)</sub>	<b>0.853</b> <sub>(0.027)</sub>	<b>0.890</b> <sub>(0.014)</sub>	<b>0.808</b> <sub>(0.009)</sub>	<b>0.8319</b> <sub>(0.007)</sub>	<b>0.884</b> <sub>(0.004)</sub>
BBBP	<b>0.723</b> <sub>(0.008)</sub>	<b>0.654</b> <sub>(0.009)</sub>	<b>0.787</b> <sub>(0.014)</sub>	<b>0.657</b> <sub>(0.004)</sub>	<b>0.708</b> <sub>(0.005)</sub>	<b>0.785</b> <sub>(0.021)</sub>
ChinTox	<b>0.973</b> <sub>(0.006)</sub>	<b>0.981</b> <sub>(0.004)</sub>	<b>0.989</b> <sub>(0.003)</sub>	<b>0.948</b> <sub>(0.003)</sub>	<b>0.983</b> <sub>(0.003)</sub>	<b>0.991</b> <sub>(0.005)</sub>
SIDER	<b>0.824</b> <sub>(0.003)</sub>	<b>0.832</b> <sub>(0.006)</sub>	<b>0.842</b> <sub>(0.001)</sub>	<b>0.825</b> <sub>(0.004)</sub>	<b>0.836</b> <sub>(0.001)</sub>	<b>0.847</b> <sub>(0.002)</sub>
Tox21	<b>0.724</b> <sub>(0.005)</sub>	<b>0.715</b> <sub>(0.004)</sub>	<b>0.799</b> <sub>(0.005)</sub>	<b>0.731</b> <sub>(0.012)</sub>	<b>0.753</b> <sub>(0.004)</sub>	<b>0.800</b> <sub>(0.006)</sub>
HIV	<b>0.753</b> <sub>(0.007)</sub>	<b>0.804</b> <sub>(0.010)</sub>	<b>0.821</b> <sub>(0.005)</sub>	<b>0.744</b> <sub>(0.018)</sub>	<b>0.818</b> <sub>(0.006)</sub>	<b>0.823</b> <sub>(0.002)</sub>
MUV	<b>0.638</b> <sub>(0.008)</sub>	<b>0.787</b> <sub>(0.012)</sub>	<b>0.834</b> <sub>(0.009)</sub>	<b>0.792</b> <sub>(0.013)</sub>	<b>0.797</b> <sub>(0.011)</sub>	<b>0.834</b> <sub>(0.010)</sub>

where,  $C_{k,j,i}^{(l)}$  is learnable parameters initially sampled from a normal distribution  $\mathcal{N}\left(0, \frac{1}{n_{l+1} \times K}\right)$ .  $\mathbf{x}^{(l)} = (x_1^{(l)}, x_2^{(l)}, \dots, x_{n_l}^{(l)})$  denotes the input vector of activation values in layer  $l$  of the Polynomial-series based KAN.

The data in the table illustrate that the KA-GCN and KA-GAT models, which incorporate Fourier-based methods, consistently outperform other models across all tested datasets. This underscores the substantial advantages of integrating Fourier-based approaches in enhancing the predictive accuracy and generalizability of GNN architectures. The superiority of these models is largely attributable to the powerful approximation capabilities of Fourier-based KAN, which we have substantiated in Theorem 2 and will further validate through subsequent function fitting evaluations. Furthermore, our ablation study not only validates our hypothesis that Fourier-based KAN provides a robust framework for improving molecular property prediction, significantly enhancing everything from molecular feature initialization to message passing in graph neural networks and prediction computations, but also compares the operational efficiency of models under different base functions. As shown in Figure 2 A and B, the results demonstrate that Fourier-based KAN significantly outperforms B-spline KAN in terms of efficiency. The Fourier-based models also clearly surpass the GCN and GAT models that utilize these other two base functions.

Furthermore, in this study, we also compared the traditional GCN and GAT models with our KA-GNNs (KA-GCN and KA-GAT), on the same feature molecular graphs. Table 4 illustrates that KA-GNNs demonstrate significant

Table 4: Comparison of GCN/GAT models and KA-GCN/KA-GAT with Fourier series.

Dataset	GCN	KA-GCN	GAT	KA-GAT
BACE	0.835 <sub>(0.014)</sub>	<b>0.890</b> <sub>(0.014)</sub>	0.834 <sub>(0.012)</sub>	<b>0.884</b> <sub>(0.004)</sub>
BBBP	0.735 <sub>(0.011)</sub>	<b>0.787</b> <sub>(0.014)</sub>	0.707 <sub>(0.007)</sub>	<b>0.785</b> <sub>(0.021)</sub>
ClinTox	0.979 <sub>(0.004)</sub>	<b>0.989</b> <sub>(0.003)</sub>	0.983 <sub>(0.006)</sub>	<b>0.991</b> <sub>(0.005)</sub>
SIDER	0.834 <sub>(0.001)</sub>	<b>0.842</b> <sub>(0.001)</sub>	0.836 <sub>(0.002)</sub>	<b>0.847</b> <sub>(0.002)</sub>
Tox21	0.747 <sub>(0.006)</sub>	<b>0.799</b> <sub>(0.005)</sub>	0.751 <sub>(0.007)</sub>	<b>0.800</b> <sub>(0.006)</sub>
HIV	0.762 <sub>(0.005)</sub>	<b>0.821</b> <sub>(0.005)</sub>	0.761 <sub>(0.003)</sub>	<b>0.823</b> <sub>(0.002)</sub>
MUV	0.741 <sub>(0.006)</sub>	<b>0.834</b> <sub>(0.009)</sub>	0.784 <sub>(0.019)</sub>	<b>0.834</b> <sub>(0.010)</sub>

performance improvements across all datasets. This confirms that the incorporation of Fourier series significantly enhances the models’ ability to recognize and express the inherent periodicity and complexity in chemical structures.

## Discussion

We attribute the effectiveness and power of the KA-GNNs model to three core innovations. First, in constructing molecular graphs, we incorporate edges not only from traditional covalent bonds but also from non-covalent interactions based on a cut-off distance. This approach significantly enhances the model’s understanding of molecular structures, enabling it to capture more molecular properties than traditional covalent interactions alone. Second, we integrate the Kolmogorov-Arnold Network (KAN) into our model. KAN significantly reduces the number of parameters while providing high interpretability and enhanced expressive power, leading to superior performance of the KA-GNNs model. Third, we replace the pre-activation function in the KAN model from the original B-splines to Fourier series functions. Fourier series functions, widely used in signal processing, have proven effective in our model for analyzing and learning high-dimensional graphical data. By learning and optimizing the coefficients of these Fourier series functions for different variables, KA-GNNs improves the accuracy and stability of predictions when dealing with complex chemical and biological data structures. These three innovations collectively contribute to the outstanding performance of KA-GNNs, making it a powerful tool for molecular property prediction.

However, our KA-GNNs still have limitations. One of the key challenges is their interpretability. Even though KANs have been designed for excellent interpretability and indeed show good results in PDE examples. In our study, the pruning process however does not generate biologically meaningful superposition structures, that are easy to interpretate.

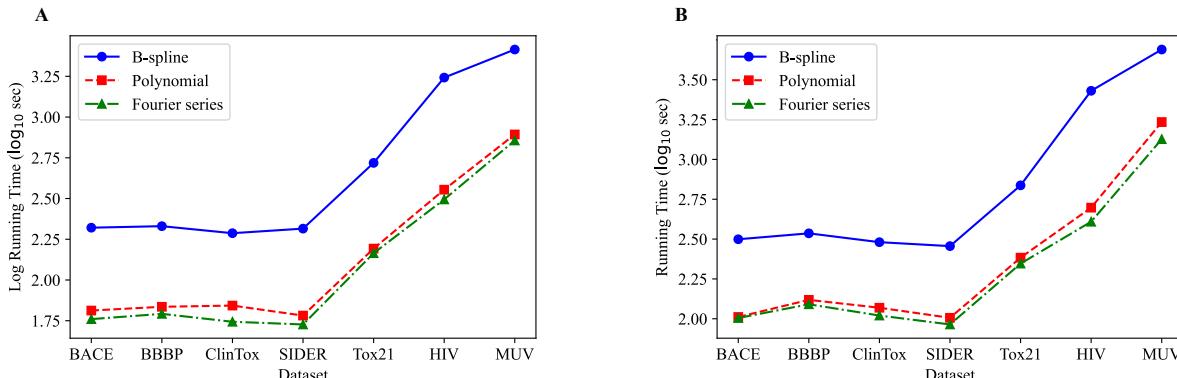


Figure 2: The comparison the computational efficiency of KA-GNNs based on B-spline, polynomial, and Fourier series. A. Running time of KA-GCN model for 100 epochs across different datasets. B. Running time of KA-GAT Model for 100 epochs across different datasets.

### 3 Methods

#### 3.1 Approximation Capability of Fourier KAN

Given that the inner functions in Eq (1) of the Kolmogorov-Arnold representation theorem can exhibit significant non-smoothness, we do not use this theorem as the foundational theory for showing the approximation capability of our model. Instead, we base our approach on the extension of Carleson's theorem [43] regarding the convergence of Fourier series for multivariable functions, as proved by Fefferman in [44]:

**Theorem 3** *Let  $\mathbf{Z}^n$  denote the  $n$ -dimensional integer lattice, and let  $Z_N^n = \{1, 2, \dots, N\}^n \subset \mathbf{Z}^n$ . Then for the function  $f \in L^2([0, 2\pi]^n)$  and its Fourier expansion:*

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

where  $x \in [0, 2\pi]^n$  and  $L^2([0, 2\pi]^n)$  denotes the space of square-integrable functions on  $[0, 2\pi]^n$ , which consists of all functions  $f$  such that  $\int_{[0, 2\pi]^n} |f(\vec{x})|^2 d\vec{x} < \infty$ . We have

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

almost everywhere.

The above theorem demonstrates the strong approximation capability of Fourier series, which is why we adopt Fourier series as the foundational basis for our model. Therefore, we retain the architecture of the Kolmogorov-Arnold network but replace the pre-activation functions with Fourier series. We can further prove that this new KAN architecture provides the potential of robust approximation capability:

**Theorem 4** *Let  $f \in L^2([0, 2\pi]^n)$  be a square-integrable function. For almost every  $\vec{x} \in [0, 2\pi]^n$  and for any  $\epsilon > 0$ , there exist a positive integer  $K$  and a sequence of Fourier-based KAN functions  $\text{KAN}_l$  at multiple layers  $l = 0, 1, \dots, L$ , such that the number of harmonics in the pre-activation functions of these KAN functions does not exceed  $K$ , and*

$$|f(\vec{x}) - \text{KAN}_L \circ \text{KAN}_{L-1} \circ \dots \circ \text{KAN}_0(\vec{x})| < \epsilon,$$

where  $\circ$  denotes function composition.

**Theorem 5** *Let  $\mathbf{Z}^n$  denote the  $n$ -dimensional integer lattice, and let  $Z_N^n = \{1, 2, \dots, N\}^n \subset \mathbf{Z}^n$ . Then for the function  $f \in L^2([0, 2\pi]^n)$  and its Fourier expansion:*

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

where  $x \in [0, 2\pi]^n$  and  $L^2([0, 2\pi]^n)$  denotes the space of square-integrable functions on  $[0, 2\pi]^n$ , which consists of all functions  $f$  such that  $\int_{[0, 2\pi]^n} |f(\vec{x})|^2 d\vec{x} < \infty$ .

We have

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

almost everywhere.

**Proof.** The theorem follows as a particular instance of the multidimensional Carleson theorem, as established in [44].

**Theorem 6** *Let  $f \in L^2([0, 2\pi]^n)$  be a square-integrable function. For almost every  $\vec{x} \in [0, 2\pi]^n$  and for any  $\epsilon > 0$ , there exist a positive integer  $K$  and a sequence of Fourier-based KAN functions  $\text{KAN}_l$  at multiple layers  $l = 0, 1, \dots, L$ , such that the number of harmonics in the pre-activation functions of these KAN functions does not exceed  $K$ , and*

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

where  $\circ$  denotes function composition.

**Proof.** Based on **Theorem 1**, there exists a number  $N$  such that

$$\left| f(\vec{x}) - \sum_{\vec{k} \in Z_N^n} (a_{\vec{k}} \cos(\vec{k} \cdot \vec{x}) + b_{\vec{k}} \sin(\vec{k} \cdot \vec{x})) \right| < \frac{\epsilon}{2}. \quad (11)$$

Furthermore, there exists  $\delta > 0$  such that for any  $\vec{k} \in Z_N^n$  and any  $y \in \mathbb{R}$  with  $\|y - \vec{k} \cdot \vec{x}\| < \delta$ , we have

$$|\cos(y) - \cos(\vec{k} \cdot \vec{x})| < \frac{\epsilon}{4N^n a_{\vec{k}}}, \quad (12)$$

and

$$|\sin(y) - \sin(\vec{k} \cdot \vec{x})| < \frac{\epsilon}{4N^n b_{\vec{k}}}. \quad (13)$$

Let  $S(N_1, id)(x)$  denote the partial sum of the first  $N_1$  terms of the Fourier expansion of  $f$ , where  $N_1$  is such that

$$|x - S(N_1, id)(x)| < \frac{\delta}{nN}, \quad (14)$$

and  $id : x \mapsto x$  is the identity function on  $[0, 2\pi]$ .

To prove the result, select  $K = \max\{N, N_1\}$ . We construct KAN functions  $\text{KAN}_0$  and  $\text{KAN}_1$  in two layers to achieve  $|f(\vec{x}) - \text{KAN}_1 \circ \text{KAN}_0(\vec{x})| < \epsilon$ . In the following proof, we denote the input vector at layer  $l$  by  $\mathbf{x}^{(l)} = (x_1^{(l)}, x_2^{(l)}, \dots)$ , and let the initial input vector  $\mathbf{x}^{(0)} = \vec{x}$

For  $Z_N^n = \{\vec{k}_1, \vec{k}_2, \dots, \vec{k}_{N^n}\}$  with  $\vec{k}_j = (k_{j,1}, k_{j,2}, \dots, k_{j,n})$ , define the pre-activation function at layer 0 as

$$\phi_{j,i}^{(0)}(x) = k_{j,i} S(N_1, id)(x), \quad i = 1, 2, \dots, n, j = 1, 2, \dots, N^n.$$

Then the  $j$ -th neuron at layer 1 is  $x_j^{(1)} = \sum_{i=1}^n k_{j,i} S(N_1, id)(x_i^{(0)})$ . From (14), we have

$$|x_j^{(1)} - \vec{k}_j \cdot \mathbf{x}^{(0)}| < \delta, \quad \text{for all } j = 1, 2, \dots, N^n. \quad (15)$$

At layer 1, define the pre-activation function as

$$\phi_{j,i}^{(1)}(x) = a_{\vec{k}_i} \cos(x) + b_{\vec{k}_i} \sin(x), \quad i = 1, 2, \dots, N^n, j = 1.$$

The single neuron at layer 2 is

$$x_1^{(2)} = \sum_{i=1}^{N^n} \left( a_{\vec{k}_i} \cos(x_i^{(1)}) + b_{\vec{k}_i} \sin(x_i^{(1)}) \right).$$

Using (12), (13), and (15), we get

$$\left| x_1^{(2)} - \sum_{\vec{k} \in Z_N^n} (a_{\vec{k}} \cos(\vec{k} \cdot \mathbf{x}^{(0)}) + b_{\vec{k}} \sin(\vec{k} \cdot \mathbf{x}^{(0)})) \right| < \frac{\epsilon}{2}.$$

Finally, using the inequality (11), we conclude

$$|f(\vec{x}) - x_1^{(2)}| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon,$$

Notice that  $x_1^{(2)} = \text{KAN}_1 \circ \text{KAN}_0(\mathbf{x}^{(0)})$  by definition, we complete the proof.

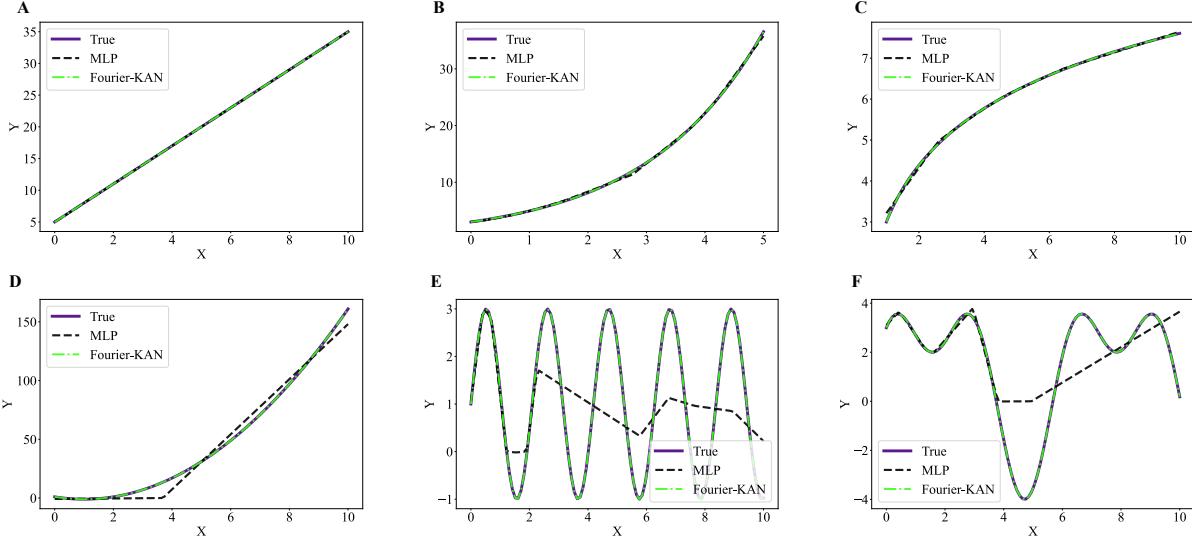


Figure 3: Fourier-series KAN and two-layers MLP fit six different functions: A. Linear function  $y = 3x + 5$ ,  $K = 200$  in Fourier-series KAN; B. Exponential function  $y = 3 \exp(\frac{1}{2}x)$ ,  $K = 120$  in Fourier-series KAN; C. Logarithmic function  $y = 2 \log(x) + 3$ ,  $K = 100$  in Fourier-series KAN; D. Polynomial function  $y = 2x^2 - 4x + 1$ ,  $K = 500$  in Fourier-series KAN; E. Sin function  $y = 2 \sin(3x) + 1$ ,  $K = 100$  in Fourier-series KAN; F. Sin and Cos function  $y = 3 \sin(x) + 2 \cos(2x) + 1$ ,  $K = 100$  in Fourier-series KAN.

In contrast to the multilayer perceptron (MLP) approach, which increases the number of layers to enhance the model’s ability to fit the data, the Fourier KAN model enhances its fitting ability by adjusting the number of Fourier basis functions (controlled by the  $K$  parameter) and the option to include a bias term to decompose complex functions into simpler sine and cosine waveforms. This strategic decomposition not only improves the model’s expressiveness but also reduces the computational burden. We evaluated the fitting ability of two-layers MLP and one-layer Fourier series KAN on six different function types, and the results are shown in Figure 3. Our analysis shows that Fourier KAN captures the underlying properties of the function more skillfully and exhibits superior expressiveness, especially in the ability to fit periodic functions.

### 3.2 Benchmark Datasets

For an extensive validation of our KA-GNN model, we used seven benchmark datasets from MoleculeNet [47]. Three datasets are from the biophysics domain: MUV, HIV, and BACE. MUV, a subset of PubChem BioAssay refined using nearest neighbor analysis, is designed for validating virtual screening techniques. The HIV dataset measures the ability of molecules to inhibit HIV replication. BACE includes both quantitative (IC50) and qualitative (binary) binding results for inhibitors of human  $\beta$ -secretase 1 (BACE-1). The remaining four datasets are from the physiology domain: BBBP, Tox21, SIDER, and ClinTox. BBBP contains binary labels for blood-brain barrier penetration. Tox21 provides qualitative toxicity measurements for 12 biological targets. SIDER is a database of marketed drugs and their adverse drug reactions (ADRs), grouped into 27 system organ classes. ClinTox includes qualitative data on FDA-approved drugs and those that failed clinical trials due to toxicity.

The original data in these datasets are SMILES strings of the molecules. During data preprocessing, we use the Merck molecular force field (MMFF94) function from the RDKit package to generate 3D molecular structures from the SMILES strings. Based on the generated structures, we construct graphs for the GNN in our KA-GNN model to predict the relevant properties. For evaluation, we use the Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) metric. Additionally, we employ the scaffold splitting method [57] to divide the datasets into training, validation, and test sets in a ratio of 8:1:1.

Table 5: KA-GCN and KA-GAT parameters for different datasets.

Dataset	Model	Batch size	LR	K	Layers	Heads	Epochs
BACE	KA-GNN	128	1e-4	1	3	-	500
BACE	KA-GAT	64	1e-4	2	2	4	500
BBBP	KA-GNN	128	1e-4	2	1	-	500
BBBP	KA-GAT	128	1e-4	5	2	2	500
ChinTox	KA-GNN	128	1e-4	2	2	-	500
ChinTox	KA-GAT	128	1e-4	5	2	2	500
SIDER	KA-GNN	128	1e-4	2	1	-	500
SIDER	KA-GAT	128	1e-4	4	2	2	500
Tox21	KA-GNN	512	1e-4	2	2	-	500
Tox21	KA-GAT	512	1e-4	3	2	2	500
HIV	KA-GNN	512	1e-4	2	2	-	500
HIV	KA-GAT	512	1e-4	2	2	4	500
MUV	KA-GNN	512	1e-4	2	2	-	500
MUV	KA-GAT	512	1e-4	3	2	4	500

### 3.3 Baseline Models

Our KA-GNN model has been evaluated alongside a selection of state-of-the-art GNN architectures, including both pre-trained and non-pre-trained models. These encompass MPNN frameworks like D-MPNN [49], attention-driven models such as AttentiveFP [48], and multi-scale approaches exemplified by Mol-GDL [28]. Importantly, our comparisons also extend to geometry-focused graph models that leverage pre-training, including N-Gram [50], PretrainGNN [51], GROVER [26], GraphMVP [52], MolCLR [53], GEM [54], Uni-mol [55], SMPT [56], and GNN-SKAN[38]. Table 5 is the hyperparameters setting for various datasets, including model type, batch size, learning rate (LR), number of harmonics ( $K$ ), number of message passing layers, number of attention heads (for GAT), and number of epochs.

Table 6: Hyperparameter Sensitivity Analysis for KA-GNN.

Batch size	64	128	256	512
BACE	0.887	<b>0.890</b>	-	-
BBBP	0.696	<b>0.787</b>	-	-
ChinTox	<b>0.992</b>	0.989	-	-
SIDER	0.841	<b>0.842</b>	-	-
Tox21	0.772	0.774	0.769	<b>0.799</b>
HIV	0.754	0.768	0.778	<b>0.821</b>
MUV	0.686	0.696	0.725	<b>0.834</b>
LR	1e-3	5e-4	1e-4	5e-5
BACE	0.806	0.818	<b>0.890</b>	0.858
BBBP	0.692	0.717	<b>0.787</b>	0.736
ChinTox	0.984	0.979	<b>0.991</b>	0.989
SIDER	0.831	0.837	<b>0.842</b>	0.835
Tox21	0.725	0.763	<b>0.799</b>	0.764
HIV	0.754	0.779	<b>0.821</b>	0.819
MUV	0.708	0.723	<b>0.834</b>	0.801
#Layers	0	1	2	3
BACE	0.674	0.719	0.822	<b>0.890</b>
BBBP	0.720	<b>0.787</b>	0.734	0.672
ChinTox	0.973	0.976	<b>0.989</b>	0.984
SIDER	0.838	0.830	<b>0.842</b>	0.829
Tox21	0.702	0.721	<b>0.799</b>	0.760
HIV	0.718	0.763	<b>0.821</b>	0.754
MUV	0.700	0.756	<b>0.834</b>	0.710

### 3.4 Hyperparameter Sensitivity Analysis

Table 6 presents an analysis of the sensitivity of various hyperparameters, including batch size, learning rate (LR), and network depth, on the performance of molecular property prediction models. The results suggest that selecting an optimal batch size and a lower learning rate tends to enhance model performance. However, increasing the number of layers does not consistently yield improvements and may even result in overfitting or diminishing returns. These findings highlight the critical role of hyperparameter tuning in achieving optimal model performance.

## 4 Conclusion

In this study, we propose the first non-trivial Kolmogorov-Arnold Network-based Graph Neural Networks (KA-GNNs). Our KA-GNNs utilize KAN’s unique power to optimize GNN architectures at three major levels, including node embedding, message passing, and readout. We develop Fourier series-based KAN model and provide a rigorous mathematical prove of the robust approximation capability of the Fourier KAN architecture. We compare the effects of various pre-activation function choices on the expressive power of the KAN model in terms of model performance and computational efficiency, identifying Fourier series as the optimal pre-activation function. Based on widely-used benchmark datasets for molecular property prediction, we extensively compare with existing state-of-the-art models, and find that our KA-GNNs can outperform traditional GNN models. This work not only highlights the great power of KA-GNNs in molecular property prediction but also provides a novel geometric deep learning framework for the general non-Euclidean data analysis.

## References

- [1] James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.
- [2] Yang Zhang, Caiqi Liu, Mujiexin Liu, Tianyuan Liu, Hao Lin, Cheng-Bing Huang, and Lin Ning. Attention is all you need: utilizing attention in AI-enabled drug discovery. *Briefings in bioinformatics*, 25(1):bbad467, 2024.
- [3] HCS Chan, H Shan, T Dahoun, H Vogel, and S Yuan. Advancing drug discovery via artificial intelligence. *Trends in Pharmacological Sciences*, 40(8):592–604, 2019.
- [4] KA Carpenter and X Huang. Machine learning-based virtual screening and its applications to Alzheimer’s drug discovery: a review. *Current Pharmaceutical Design*, 24(28):3347–3358, 2018.
- [5] EHB Maia, LC Assis, TA De Oliveira, AM Da Silva, and AG Taranto. Structure-based virtual screening: from classical to artificial intelligence. *Frontiers in Chemistry*, 8:343, 2020.
- [6] T Zhao, Y Hu, LR Valsdottir, T Zang, and J Peng. Identifying drug–target interactions based on graph convolutional network and deep neural network. *Briefings in Bioinformatics*, 22(2):2141–2150, 2021.
- [7] Jun Xia, Lecheng Zhang, Xiao Zhu, Yue Liu, Zhangyang Gao, Bozhen Hu, Cheng Tan, Jiangbin Zheng, Siyuan Li, and Stan Z Li. Understanding the limitations of deep models for molecular property prediction: Insights and solutions. *Advances in Neural Information Processing Systems*, 36:64774–64792, 2023.
- [8] C Durán, S Daminelli, and JM Thomas. Pioneering topological methods for network-based drug–target prediction by exploiting a brain-network self-organization theory. *Briefings in Bioinformatics*, 19(6):1183–1202, 2018.
- [9] Duc Duy Nguyen, Zixuan Cang, and Guo-Wei Wei. A review of mathematical representations of biomolecular data. *Physical Chemistry Chemical Physics*, 22(8):4343–4367, 2020.
- [10] Stephen Bonner et al. Implications of topological imbalance for representation learning on biomedical knowledge graphs. *Briefings in Bioinformatics*, 23(5):bbac279, 2022.
- [11] Z Meng and K Xia. Persistent spectral-based machine learning (PerSpect ML) for protein-ligand binding affinity prediction. *Science Advances*, 7(19):eabc5329, 2021.
- [12] DD Nguyen and GW Wei. AGL-Score: Algebraic Graph Learning Score for Protein-Ligand Binding Scoring, Ranking, Docking, and Screening. *Journal of Chemical Information and Modeling*, 59(7):3291–3304, 2019.
- [13] NA Szulc, Z Mackiewicz, and JM Bujnicki. Structural interaction fingerprints and machine learning for predicting and explaining binding of small molecule ligands to RNA. *Briefings in Bioinformatics*, 24(4):bbad187, 2023.
- [14] J Chen, R Wang, M Wang, and GW Wei. Mutations strengthened SARS-CoV-2 infectivity. *Journal of Molecular Biology*, 432(19):5212–5226, 2020.

- [15] R Wang, Y Hozumi, C Yin, and GW Wei. Mutations on COVID-19 diagnostic targets. *Genomics*, 112(6):5204–5213, 2020.
- [16] K Gao, DD Nguyen, M Tu, and GW Wei. Generative network complex for the automated generation of drug-like molecules. *Journal of Chemical Information and Modeling*, 60(12):5682–5698, 2020.
- [17] Bo-Wei Zhao, Xiao-Rui Su, Peng-Wei Hu, Yu-Peng Ma, Xi Zhou, and Lun Hu. A geometric deep learning framework for drug repositioning over heterogeneous information networks. *Briefings in Bioinformatics*, 23(6):bbac384, 2022.
- [18] P Li, Y Li, and CY Hsieh. TrimNet: learning molecular representation from triplet messages for biomedicine. *Briefings in Bioinformatics*, 22(4):bbaa266, 2021.
- [19] Xiao-Shuang Li, Xiang Liu, Le Lu, Xian-Sheng Hua, Ying Chi, and Kelin Xia. Multiphysical graph neural network (MP-GNN) for COVID-19 drug design. *Briefings in Bioinformatics*, 23(4):bbac231, 2022.
- [20] C Kang, H Zhang, and Z Liu. LR-GNN: A graph neural network based on link representation for predicting molecular associations. *Briefings in Bioinformatics*, 23(1):bbab513, 2022.
- [21] Hanxuan Cai, Huimin Zhang, Duancheng Zhao, Jingxing Wu, and Ling Wang. FP-GNN: A versatile deep learning architecture for enhanced molecular property prediction. *Briefings in Bioinformatics*, 23(6):bbac408, 2022.
- [22] Hui Liu, Yibiao Huang, Xuejun Liu, and Lei Deng. Attention-wise masked graph contrastive learning for predicting molecular property. *Briefings in Bioinformatics*, 23(5):bbac303, 2022.
- [23] R Zhang, Y Lin, and Y Wu. MvMRL: A Multi-View Molecular Representation Learning Method for Molecular Property Prediction. *Briefings in Bioinformatics*, 25(4):bbae298, 2024.
- [24] R Mercado, T Rastemo, E Lindelöf, G Klambauer, O Engkvist, H Chen, and EJ Bjerrum. Graph networks for molecular design. *Machine Learning: Science and Technology*, 2(2):025023, 2021.
- [25] Q Liu, M Allamanis, M Brockschmidt, and A Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in Neural Information Processing Systems*, 31, 2018.
- [26] Y Rong, Y Bian, T Xu, W Xie, Y Wei, W Huang, and J Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.
- [27] M Wang, Z Cang, and GW Wei. A topology-based network tree for the prediction of protein-protein binding affinity changes following mutation. *Nature Machine Intelligence*, 2(2):116–123, 2020.
- [28] C Shen, J Luo, and K Xia. Molecular geometric deep learning. *Cell Reports Methods*, 3(11), 2023.
- [29] Z Liu, Y Wang, S Vaidya, F Ruehle, J Halverson, M Soljačić, TY Hou, and M Tegmark. Kan: Kolmogorov-Arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.
- [30] R Bresson, G Nikolenz, and G Panagopoulos. Kagnns: Kolmogorov-Arnold networks meet graph learning. *arXiv preprint arXiv:2406.18380*, 2024.
- [31] BC Koenig, S Kim, and S Deng. KAN-ODEs: Kolmogorov–Arnold Network Ordinary Differential Equations for Learning Dynamical Systems and Hidden Physics. *Computer Methods in Applied Mechanics and Engineering*, 432:117397, 2024.
- [32] M Liu, S Bian, and B Zhou. ikan: Global Incremental Learning with KAN for Human Activity Recognition Across Heterogeneous Datasets. *Proceedings of the 2024 ACM International Symposium on Wearable Computers*, pages 89–95, 2024.
- [33] R Genet and H Inzirillo. Tkan: Temporal kolmogorov-arnold networks. *arXiv preprint arXiv:2405.07344*, 2024.
- [34] M Cheon. Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing. *arXiv preprint arXiv:2406.00600*, 2024.
- [35] M Kiamari, M Kiamari, and B Krishnamachari. GKAN: Graph Kolmogorov-Arnold Networks. *arXiv preprint arXiv:2406.06470*, 2024.
- [36] Z Bozorgasl and H Chen. Wav-kan: Wavelet kolmogorov-arnold networks. *arXiv preprint arXiv:2405.12832*, 2024.
- [37] T Ahmed and MHR Sifat. GraphKAN: Graph Kolmogorov Arnold Network for Small Molecule-Protein Interaction Predictions. In *Proceedings of the ICML’24 Workshop on Machine Learning for Life and Material Science: From Theory to Industry Applications*, 2024.
- [38] Ruifeng Li, Mingqian Li, Wei Liu, and Hongyang Chen. GNN-SKAN: Harnessing the Power of SwallowKAN to Advance Molecular Representation Learning with GNNs. *arXiv preprint arXiv:2408.01018*, 2024.

- [39] Ali Kashefi. PointNet with KAN versus PointNet with MLP for 3D Classification and Segmentation of Point Sets. *arXiv preprint arXiv:2410.10084*, 2024.
- [40] AN Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk*, 114:953–956, 1957.
- [41] DA Sprecher. A numerical implementation of Kolmogorov’s superpositions. *Neural Networks*, 9(5):765–772, 1996.
- [42] J Braun and M Griebel. On a constructive proof of Kolmogorov’s superposition theorem. *Constructive Approximation*, 30:653–675, 2009.
- [43] L Carleson. On convergence and growth of partial sums of Fourier series. *Acta Mathematica*, 116:135–157, 1966.
- [44] C Fefferman. On the convergence of multiple Fourier series. *Bulletin of the American Mathematical Society*, 77:744–745, 1971.
- [45] T Xie and JC Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120(14):145301, 2018.
- [46] L Li, Xiang Liu, and G Wang. Path Complex Neural Network for Molecular Property Prediction. In *Proceedings of the ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024.
- [47] Z Wu, B Ramsundar, EN Feinberg, J Gomes, C Geniesse, AS Pappu, K Leswing, and V Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.
- [48] Z Xiong, D Wang, X Liu, F Zhong, X Wan, X Li, Z Li, X Luo, K Chen, H Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2019.
- [49] K Yang, K Swanson, W Jin, C Coley, P Eiden, H Gao, A Guzman-Perez, T Hopper, B Kelley, M Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.
- [50] S Liu, MF Demirel, and Y Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in Neural Information Processing Systems*, 32, 2019.
- [51] W Hu, B Liu, J Gomes, M Zitnik, P Liang, V Pande, and J Leskovec. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*, 2019.
- [52] S Liu, H Wang, W Liu, J Lasenby, H Guo, and J Tang. Pre-training Molecular Graph Representation with 3D Geometry. In *International Conference on Learning Representations*, 2022.
- [53] Y Wang, J Wang, Z Cao, and AB Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.
- [54] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.
- [55] G Zhou, Z Gao, Q Ding, H Zheng, H Xu, Z Wei, L Zhang, and G Ke. Uni-Mol: A Universal 3D Molecular Representation Learning Framework. *The Eleventh International Conference on Learning Representations*, 2023.
- [56] Y Li, W Wang, J Liu, and C Wu. Pre-training molecular representation model with spatial geometry for property prediction. *Computational Biology and Chemistry*, 109:108023, 2024.
- [57] B. Ramsundar, P. Eastman, P. Walters, and V. Pande. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. O’Reilly Media, Inc., Sebastopol, CA, 1st edition, 2019.