A Survey on Kolmogorov-Arnold Network

SHRIYANK SOMVANSHI, Texas State University, San Marcos, United States
SYED AAQIB JAVED, Texas State University, San Marcos, United States
MD MONZURUL ISLAM, Texas State University, San Marcos, United States
DIWAS PANDIT, Texas State University, San Marcos, United States
SUBASISH DAS, Civil Engineering, Texas State University, San Marcos, United States

This review study explores the theoretical foundations, evolution, applications, and future potential of Kolmogorov–Arnold Networks (KAN), a neural network model inspired by the Kolmogorov–Arnold representation theorem. KANs set themselves apart from traditional neural networks by employing learnable, spline-parameterized functions rather than fixed activation functions, allowing for flexible and interpretable representations of high-dimensional functions. The review explores Kan's architectural strengths, including adaptive edge-based activation functions that enhance parameter efficiency and scalability across varied applications such as time series forecasting, computational biomedicine, and graph learning. Key advancements including Temporal-KAN (T-KAN), FastKAN, and Partial Differential Equation (PDE) KAN illustrate KAN's growing applicability in dynamic environments, significantly improving interpretability, computational efficiency, and adaptability for complex function approximation tasks. Moreover, the article discusses KANs integration with other architectures, such as convolutional, recurrent, and transformer-based models, showcasing its versatility in complementing established neural networks for tasks that require hybrid approaches. Despite its strengths, KAN faces computational challenges in high-dimensional and noisy data settings, sparking continued research into optimization strategies, regularization techniques, and hybrid models. This article highlights KANs expanding role in modern neural architectures and outlines future directions to enhance its computational efficiency, interpretability, and scalability in data-intensive applications.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Deep learning theory**; **Kolmogorov Arnold Networks (KAN)**; *Model interpretability*; • **Applied computing** → *Predictive analytics;*

Additional Key Words and Phrases: Kolmogorov-arnold network

ACM Reference Format:
Shriyank Somvanshi, Syed Aaqib Javed, Md Monzurul Islam, Diwas Pandit, and Subasish Das. 2025. A Survey on Kolmogorov-Arnold Network. *ACM Comput. Surv.* 58, 2, Article 55 (September 2025), 35 pages. https://doi.org/10.1145/3743128

Authors' Contact Information: Shriyank Somvanshi, Texas State University, San Marcos, Texas, United States; e-mail: shriyanksomvanshi@gmail.com; Syed Aaqib Javed, Texas State University, San Marcos, Texas, United States; e-mail: aaqib.ce@txstate.edu; Md Monzurul Islam, Texas State University, San Marcos, Texas, United States; e-mail: monzurul@txstate.edu; Diwas Pandit, Texas State University, San Marcos, Texas, United States; e-mail: zxh15@txstate.edu; Subasish Das, Civil Engineering, Texas State University, San Marcos, Texas, United States; e-mail: subasish@txstate.edu.

ACM 0360-0300/2025/09-ART55
https://doi.org/10.1145/3743128

1 Introduction

Traditional deep learning models such as **Convolutional Neural Networks** (**CNNs**), **Recurrent Neural Networks** (**RNNs**), and Transformers have achieved remarkable success across a wide range of tasks, from image recognition to natural language processing. However, despite their widespread adoption, these models face several limitations. They often require large amounts of training data and computational resources, exhibit limited interpretability, and can be prone to overfitting or sensitivity to distribution shifts. Moreover, their reliance on fixed activation functions and rigid architectural assumptions can constrain their ability to efficiently model complex, high-dimensional functions with minimal parameters.

In addition, many of these models function as black-box systems, making it difficult to understand or trust their predictions in safety-critical or regulated domains such as healthcare [1], transportation [2], and finance [3]. This lack of transparency has led to increased interest in **explainable AI** (**XAI**) and the development of models that are inherently interpretable rather than relying solely on post-hoc explanation tools. Furthermore, current deep learning architectures tend to be over-parameterized [4, 5] and computationally intensive [6], posing challenges for deployment in edge devices or resource-constrained environments. This inefficiency has motivated a broader exploration of architectures that can provide competitive accuracy with fewer parameters and reduced computational cost.

In response to these challenges, researchers have explored a variety of alternative neural network architectures that aim at improving efficiency, interpretability, and mathematical grounding. One such promising approach is the **Kolmogorov–Arnold Network** (**KAN**), which is inspired by the Kolmogorov–Arnold representation theorem. This theorem, developed by Andrey Kolmogorov and Vladimir Arnold [7], asserts that any multivariate continuous function can be represented as a finite sum of continuous univariate functions. This theoretical insight has inspired the design of KANs, which aim at decomposing complex functions into simpler, interpretable components using learnable univariate transformations. Unlike traditional neural networks that rely on fixed nonlinearities such as ReLU or sigmoid, KANs employ adaptive, spline-parametrized functions on edges instead of fixed activations. These spline-based activations allow for more flexible and precise function approximations, mitigating common issues such as vanishing gradients and saturation [8–10]. This adaptability enables KANs to capture complex, nonlinear relationships while maintaining a high degree of interpretability, making them particularly useful in domains that demand transparency and symbolic reasoning.

KAN architecture replaces traditional linear weights with adaptive spline-based activation functions, distinguishing them significantly from **Multi-Layer Perceptrons** (**MLPs**) [11]. Recent studies have demonstrated the effectiveness of KANs in applications such as predicting electrohydrodynamic pump parameters, achieving both accuracy and symbolic interpretability [13]. Emerging variants such as Chebyshev KAN, which incorporates Chebyshev polynomials, further extend the capacity for nonlinear function approximation [14].

Despite being less mainstream than CNNs or Transformers, KANs have gained attention as interpretable and data-efficient alternatives due to their fewer parameter requirements and dynamic function representation [15, 16]. However, significant research gaps persist, notably their challenges in scaling KANs to handle high-dimensional smooth functions and ensuring robustness across diverse datasets [16]. Recent advancements like **Wavelet KAN** (**Wav-KAN**) aim at addressing these issues by combining wavelet transforms with spline-based activations to enhance both interpretability and computational performance [23].

KANs continue to evolve, with newer architectures like Temporal KAN incorporating memory mechanisms for sequential data and outperforming traditional RNNs in time series forecasting [24]. The introduction of FastKAN, which approximates B-splines using Gaussian radial basis functions,

marks another step toward improved computational efficiency [26]. Comparative studies highlight KANs ability to reduce the number of parameters while achieving performance on par with CNNs and RNNs, as evidenced in tasks like image classification [29], and superior performance compared to transformers on smaller datasets [30]. These advancements position KANs as a scalable and efficient alternative to more complex architectures, particularly in environments constrained by data or computational resources. In practice, the distinctive edge-based activations of KANs enhance modularity and interpretability, facilitating their integration into neural architectures such as autoencoders and time series models [31, 32].

1.1 Research Questions

This review seeks to address several key research questions regarding KANs:

(1) **What are the primary theoretical developments in KANs, and how do they contribute to the broader landscape of neural network architectures?**
This question aims at exploring how the Kolmogorov–Arnold representation theorem has influenced the design of KANs and what theoretical innovations have emerged over time.
(2) **How have KANs been applied across various fields?**
By addressing this, the review examines the practical applications of KANs and compares their performance with traditional architectures such as MLPs, CNNs, RNNs, and transformers.
(3) **What are the key challenges and opportunities for KAN research, particularly in terms of scalability, computational efficiency, and robustness?**
This question focuses on the limitations KANs face in large-scale applications and complex datasets, identifying potential areas for future research and optimization.

The remainder of the article is organized as follows: Section 2 introduces the theoretical foundations of KANs, while Section 3 discusses major architectural extensions. Section 4 outlines training strategies. Section 5 presents applications, benchmarks, and comparative evaluations. Section 6 examines key challenges and limitations, followed by Section 7, which highlights emerging innovations and hybrid architectures. Section 8 explores future directions, and the final section concludes the study.

2 Historical Evolution of KAN

2.1 Kolmogorov–Arnold Theorem

The **Kolmogorov–Arnold representation theorem** (**KAT**) plays a key role in the theoretical foundation of function approximation in neural networks. Originally introduced by Andrey Kolmogorov in 1956 and later refined by Vladimir Arnold, the theorem states that any continuous function $f : [0,1]^n \to \mathbb{R}$ can be expressed as a finite superposition of continuous univariate functions [7, 118]. The formal statement of the theorem is as follows:

> **Kolmogorov–Arnold representation theorem.** Let $f : [0,1]^n \to \mathbb{R}$ be any continuous function. Then there exist continuous univariate functions $\phi_q : \mathbb{R} \to \mathbb{R}$ and $\varphi_{q,p} : [0,1] \to \mathbb{R}$ such that:

$$f(x_1, x_2, x_3, \ldots, x_n) = \sum_{q=1}^{2n+1} \phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(x_p) \right), \quad \text{for all } (x_1, \ldots, x_n) \in [0,1]^n. \tag{1}$$

Here, each $\phi_q : \mathbb{R} \to \mathbb{R}$ and $\varphi_{q,p} : [0,1] \to \mathbb{R}$ are continuous univariate functions that map input variables into a result through a sum of simpler functions. The term $\sum_{p=1}^{n} \varphi_{q,p}(x_p)$ represents how the univariate functions are combined [8, 24]. Equation (1) corresponds to a two-layer structure with a single hidden layer of width $2n + 1$, and serves as the foundational case for shallow KANs.

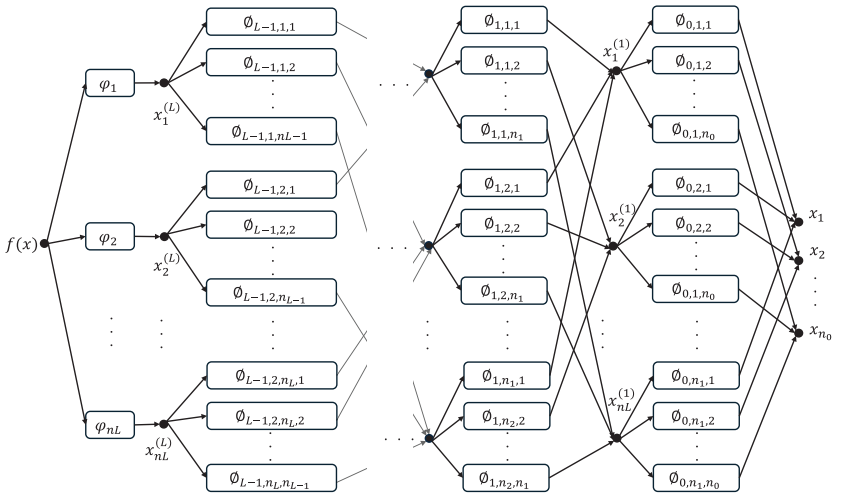

Fig. 1. Schematic representation of the KAN architecture. Each layer applies learnable univariate spline functions along the edges (i.e., between nodes), enabling edge-based transformations rather than traditional node-based activations. While the classical Kolmogorov–Arnold formulation applied outer functions only at the final stage, modern deep KAN architectures generalize this by applying such spline-based transformations at every layer, ensuring compositional and interpretable modeling across the entire network [8, 52].

Modern KAN architectures extend this principle into deeper, multilayer networks as shown in Figure 1, enabling more expressive and hierarchical function learning.

This result provided powerful insight: multivariate function approximation could be reduced to operations involving only univariate functions and addition, bypassing the drawback of dimensionality in theoretical terms [37]. The general significance of the KAT lies in its universality and constructive formulation. It establishes that any continuous multivariate function can be represented as a finite superposition of continuous functions of fewer variables. Arnold's [37] refinement provided concrete and constructive proof for the three-variable case, demonstrating that such functions can be decomposed into compositions of continuous functions of two variables. This decomposition highlights the interpretability and structural simplicity of the components involved. In the context of neural networks, KAT provided one of the earliest mathematical justifications for using layered structures to approximate high-dimensional functions [8, 37]. While the original formulation was non-constructive and not directly suited for implementation in conventional architectures, its philosophical underpinning inspired the design of networks that attempt to decompose multivariate relationships into univariate transformations [8].

Researchers such as Funahashi [42] provided formal proof that multilayer feedforward neural networks with non-polynomial activation functions specifically, bounded, monotone, continuous sigmoidal functions are universal approximators of continuous functions on compact domains. In his work, Funahashi also explored the connection to the Kolmogorov–Arnold-Sprecher theorem, using it to construct a four-layer network with nontrivial realization capabilities. In contrast, Girosi and Poggio [40] argued that Kolmogorov's theorem, while mathematically elegant, is not relevant for the practical analysis of neural networks due to the irregular and non-implementable nature of the function classes it implies.

Additionally, modern interpretations of the theorem have influenced the architecture of networks with compositional structures, especially those aiming at improving interpretability and reduce parameter complexity [8]. The decomposition into univariate functions aligns with

modular neural design philosophies and has inspired novel architectures like KANs, which explicitly draw upon the theorem's principles [8, 43]. Despite certain limitations, such as the lack of smoothness in inner functions in Kolmogorov's original construction the theorem remains a cornerstone in the theoretical landscape of function approximation and continues to influence both theoretical and practical advancements in neural network research.

2.2 From Theory to Practice: Bridging KAT and Neural Networks

KAT offers a powerful theoretical assurance: any continuous multivariate function can be expressed as a finite composition of univariate functions and addition [7]. This abstract result has far-reaching implications for the design of neural networks, as it suggests a fundamental pathway for approximating complex functions through simpler components.

However, KAT's original formulation is largely non-constructive and involves highly irregular inner functions, which are not compatible with gradient-based learning techniques common in machine learning [40]. As such, directly implementing the theorem within neural network architectures poses practical limitations, particularly in terms of smoothness, differentiability, and learnability.

To address these limitations, subsequent theoretical efforts have aimed at bridging KAT with computationally feasible models. Funahashi [42] proved that multilayer feedforward networks equipped with sigmoidal activation functions can approximate any continuous function on a compact domain, providing a more constructive alternative. Girosi and Poggio [40] further clarified the gap between the expressive power of KAT and the trainability of practical neural networks, highlighting the need for smoother and implementable function classes. These foundational insights laid the groundwork for neural architectures that draw on the compositional philosophy of KAT while adopting trainable, differentiable components suitable for modern optimization frameworks.

2.3 KAN Architecture

KANs represent a novel approach to neural network design, grounded in the Kolmogorov–Arnold representation theorem [117]. The key architectural innovation of KAN lies in replacing traditional fixed linear weights with learnable univariate functions, which are typically parameterized using B-splines. This enables KANs to model complex nonlinear functions with improved accuracy and interoperability. The mathematical foundation of KANs is based on the Kolmogorov–Arnold representation as explained in Equation (1) above.

To operationalize this in modern networks, KANs use B-splines to parameterize the univariate functions. B-splines provide smooth, localized, and adaptive control over the function shape. Unlike fixed activations like ReLU, spline-based edge activations in KAN allow dynamic, trainable nonlinear mappings during training. By learning the control points of the splines, KANs balance the modeling of both global and local patterns, improving generalization and interpretability [8, 11, 119].

Liu et al. [8] extended the KAN to networks of arbitrary depth, building on the Kolmogorov–Arnold representation theorem by unifying the outer functions ϕ_q and inner functions $\varphi_{q,p}$ into a series of KAN layers. In this extended form, the network architecture is defined by an integer array $[n_0, n_1, \ldots, n_L]$, where n_L represents the number of neurons in the L-th layer. Each layer in the KAN is structured to transform an input vector x_l of n_l dimensions into an output vector x_{l+1} of n_{l+1} dimensions.

$$x_{l+1} = \begin{pmatrix} \varphi_{1,1,l} & \cdots & \varphi_{1,n_l,l} \\ \vdots & \ddots & \vdots \\ \varphi_{n_{l+1},1,l} & \cdots & \varphi_{n_{l+1},n_l,l} \end{pmatrix} x_l. \tag{2}$$

Here, the function matrix for each layer l is denoted by Φ_l, where

$$\Phi_l = \begin{pmatrix} \varphi_{l,1,1(\cdot)} & \cdots & \varphi_{l,1,n_l(\cdot)} \\ \vdots & \ddots & \vdots \\ \varphi_{l,n_{l+1},1(\cdot)} & \cdots & \varphi_{l,n_{l+1},n_l(\cdot)} \end{pmatrix}. \tag{3}$$

Within each lth layer, the activation value of a neuron in the next layer is computed as the sum of all post-activation values from the previous layer. This hierarchical composition enables KAN to approximate complex multivariate functions through a series of recursive transformations. Mathematically, this process can be described as

$$f(x) = \sum_{i_L=1}^{n_L} \phi_{i_L} \left(x_{i_L}^{(L)} \right). \tag{4}$$

where each intermediate representation $x_{i_L}^{(L)}$ is recursively defined by summing over non-linear transformations applied to outputs from the preceding layer:

$$x_{i_L}^{(L)} = \sum_{i_{L-1}=1}^{n_{L-1}} \varphi_{L-1,i_L,i_{L-1}} \left(x_{i_{L-1}}^{(L-1)} \right). \tag{5}$$

In Liu et al.'s implementation, each KAN layer consists of a combination of spline functions and SiLU activations, which enhances flexibility in function approximation. By leveraging a variety of basis functions, including Legendre and Chebyshev polynomials, as well as Gaussian radial distribution functions, KAN can efficiently capture complex relationships within the data while maintaining interpretability and computational efficiency.

A general KAN network comprised of L layers can be written as

$$\mathrm{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \Phi_{L-3} \circ \cdots \circ \Phi_1 \circ \Phi_0)\, x, \tag{6}$$

where each layer represents a transformation from one dimensionality to another, reducing the complexity of functions by stacking KAN layers. The transformation at each layer uses a matrix of univariate functions, rather than traditional weight matrices. This is formally given as

$$x_{l+1,j} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), \tag{7}$$

where $\phi_{l,j,i}$ are spline-based univariate activation functions. Each edge between neuron i in layer l and neuron j in layer $l+1$ carries its own learnable function $\phi_{l,j,i}$, which operates on the input $x_{l,i}$ and is summed at each receiving node.

While Equation (7) captures the inner transformation at each layer, it may be interpreted as outer functions only applying to the final layer. To clarify, modern deep KAN architectures extend the original two-layer Kolmogorov–Arnold formulation by applying these spline-based edge activations at every hidden layer. Outer functions (like Φ_L) are typically reserved for the final transformation, but the layered use of univariate spline activations across all levels ensures hierarchical compositional modeling [8, 52].

Figure 1 provides a schematic representation of the KAN architecture, encapsulating the layered transformation process discussed above. Each layer in the KAN framework applies learnable, spline-based functions to inputs, replacing traditional fixed weights with adaptable non-linear mappings; enabling KANs to approximate complex, non-linear functions by capturing intricate dependencies across multiple layers. This structure generalizes the original Kolmogorov–Arnold theorem into a modular, deep-learning framework. As a result, the KAN structure enhances symbolic

function discovery capabilities while simultaneously improving computational efficiency, making it suitable for high-dimensional and complex data modeling tasks.

Approximation of Complex Functions in High-Dimensional Spaces: KAN leverages this theorem by approximating the univariate functions $\varphi_{q,p}(x_p)$ and outer functions ϕ_q using learnable splines. This allows KAN to dynamically adapt to the data patterns, as opposed to traditional neural networks like MLPs, which have fixed activation functions. The learnable nature of the splines allows KAN to approximate even complex, non-smooth functions that are otherwise challenging to capture with fixed activation networks [8, 11, 119].

A notable aspect of KAN's architecture is the preference for multiplication over division when performing mathematical operations during function approximation. Avoiding division mitigates the risk of numerical instability, particularly when divisors approach zero, while multiplication ensures smooth gradients and reliable optimization. Recent research demonstrates that multiplication significantly enhances local convergence, especially when iterative methods like Newton-Kaczmarz are employed [139]. This insight simplifies the basis function design and avoids computational bottlenecks, further improving KAN's robustness in real-world applications.

KAN operates by replacing the weight matrices typically found in MLPs with these learnable univariate functions, transforming the output in a flexible manner. The univariate functions are parameterized as splines, which are piecewise polynomial functions that can adapt locally without losing global smoothness [59]. This behavior arises from the mathematical properties of spline functions. The spline-based activation functions used in KANs enable local adaptability through control points that define piecewise polynomial segments [29]. These segments are typically continuous up to a certain derivative order (e.g., C^1 or C^2), ensuring smooth global behavior while allowing flexible local variations. This dual property facilitates the modeling of non-linear, locally varying data while preserving overall function continuity and smoothness, which is critical for both approximation fidelity and interoperability.

Interpretability and Scaling:

KANs offer modestly enhanced interpretability over traditional MLPs by making the functional mappings more transparent. Each univariate function is represented as a B-spline, allowing for fine control and clear visualization of how individual variables contribute to the final function. This interpretability makes KANs well-suited for scientific discovery tasks, where the goal is not only to approximate a function but also to gain insights into its structure. Additionally, KANs utilize neural scaling laws that allow them to generalize well with fewer parameters compared to traditional deep learning models. The spline-based structure of KANs enables them to capture complex, high-dimensional relationships while mitigating the challenges associated with high-dimensional spaces, which often hinder models that rely solely on fixed activation functions [8].

3 KAN Extensions

3.1 Major Extensions of KANs

Over time, KANs have evolved into a range of specialized extensions, each adapting the core principle of univariate decomposition to address domain-specific and computationally intensive tasks. These extensions modify the base architecture, activation mechanisms, or learning strategies to improve expressiveness, scalability, and interpretability. The following subsections present key KAN variants, each tailored to meet unique challenges across areas such as symbolic regression, partial differential equation solving, hyperspectral image classification, time series forecasting, and graph-structured learning.

3.1.1 MultKAN. MultKAN augments the original KAN structure by incorporating multiplication nodes in addition to standard additive paths. This modification allows the network

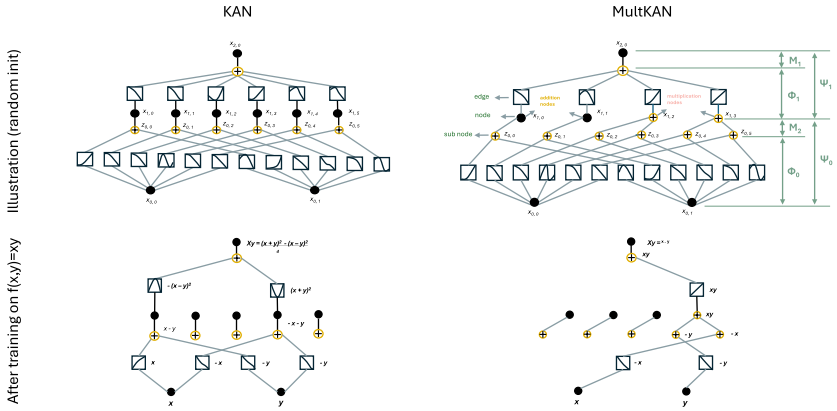

Fig. 2. Comparison of KAN and MultKAN diagrams [8].

to represent multiplicative relationships directly, thereby reducing the number of operations required for tasks such as symbolic function discovery. Figure 2 illustrates a structural comparison between the KAN and the enhanced MultKAN architecture. The top section displays the layout of each network, highlighting how MultKAN introduces additional multiplication layers, which allow it to capture multiplicative relationships more efficiently. The bottom section demonstrates the training outcomes on the function $f(x, y) = xy$, where KAN utilizes two addition nodes to approximate multiplication, while MultKAN achieves the same task with a single multiplication node. This adaptation enables MultKAN to directly represent multiplicative structures, thereby enhancing computational efficiency and interpretability in symbolic function discovery.

3.1.2 PDE-KAN. The **Partial Differential Equation KAN** (**PDE-KAN**) enhances the original KAN by incorporating physics-informed elements suited for solving differential equations. Unlike traditional KANs, which are limited to general function approximation, PDE-KANs employ physics-based constraints and loss functions, enhancing interpretability and accuracy in modeling physical phenomena described by PDEs. This adaptation enables PDE-KANs to tackle both forward and inverse problems in computational physics, making them more effective for high-complexity tasks involving boundary and initial conditions. By leveraging PDE forms such as the energy, strong, and inverse forms, PDE-KANs achieve improved convergence rates and solution accuracy, demonstrating significant advantages over conventional neural network approaches like MLPs [38, 66].

While PDE-KAN is designed to solve differential equations by incorporating physics-informed loss functions that enforce boundary and domain constraints [38], it retains the core KAN philosophy of using learnable univariate functions to preserve interpretability. This makes it particularly useful in simulations where physical consistency and symbolic reasoning are essential for validating model performance.

3.1.3 rKAN. **Rational KAN** (**rKAN**), proposed by Aghaei [65], extends KAN by using rational function bases specifically employing Padé approximations and rational Jacobi functions–to significantly improve performance in regression and classification tasks. This method enhances model accuracy and computational efficiency through optimized activation functions and more efficient parameter update mechanisms. In rKAN, the use of rational functions via polynomial divisions and mapped Jacobi functions enables it to capture sharp peaks and rapid changes in data more effectively than traditional KAN.

The rational Jacobi approach also extends function approximation over semi-infinite or infinite domains, improving rKAN's versatility for tasks requiring high precision across broad input spaces, such as physics-informed problems with complex boundary conditions. The overall rKAN formulation is expressed as

$$F(\xi) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(\xi_p) \right), \tag{8}$$

where $\varphi_{q,p}(\cdot)$ are rational functions based on Padé or Jacobi mappings. Here, $\Phi_q(\cdot)$ serves as an outer activation or aggregation function, providing flexibility in capturing complex interactions across input dimensions. This layered structure allows rKAN to generalize effectively across various types of data and tasks, offering significant improvements in precision and interpretability.

3.1.4 Wav-KAN. The Wav-KAN adapts the KAN architecture by incorporating wavelet functions as learnable activation functions, enabling nonlinear mapping of input spectral features and effectively capturing multi-scale spatial-spectral patterns through dilation and translation. This approach allows Wav-KAN to isolate significant patterns at various scales, enhancing its ability to filter out noise while retaining critical features, which is particularly useful in hyperspectral image classification. Using the Continuous Wavelet Transform and Discrete Wavelet Transform, Wav-KAN captures both high-frequency and low-frequency components, improving interpretability and robustness compared to traditional models.

As demonstrated by Seydi et al. [120], Wav-KAN significantly outperformed traditional MLP and Spline-KAN models, achieving notable improvements in classification accuracy on benchmark datasets such as Salinas, Pavia, and Indian Pines. This model also enhances computational efficiency by reducing the number of necessary parameters without sacrificing precision, making Wav-KAN an efficient and powerful solution for handling the high-dimensional, correlated nature of hyperspectral data.

3.1.5 T-KAN and MT-KAN. **Temporal KAN** (**T-KAN**) and **Multi-Task KAN** (**MT-KAN**) are specialized KAN variants developed for time series applications [60]. T-KAN, designed for univariate time series data, utilizes learnable univariate activation functions that dynamically adapt to nonlinear relationships and capture complex temporal patterns, allowing it to effectively handle variations across time. The architecture models relationships between consecutive time steps, predicting future values while tracking concept drift, a capability particularly valuable in financial forecasting and energy demand prediction. The T-KAN output at time $t + T$, denoted S_{t+T}, is given by

$$S_{t+T} = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p} \left(S_{t-h+p} \right) \right). \tag{9}$$

where S_{t-h+p} represents past observations, and $\varphi_{q,p}$ are spline-parametrized functions learned during training to model nonlinear temporal dependencies. Additionally, symbolic regression enhances T-KAN's interpretability by generating human-readable expressions that reveal the underlying dependencies between time steps.

Building on T-KAN's capabilities, MT-KAN extends this approach to handle multivariate time series by introducing a shared network structure that enables multi-task learning across related tasks. This model effectively captures inter-variable relationships, enhancing predictive accuracy in applications like electricity load forecasting and air quality monitoring, where multivariate dependencies are essential. MT-KAN improves performance by leveraging inter-task relationships, optimizing feature representations across tasks to boost prediction accuracy with reduced training data requirements. This makes MT-KAN an efficient choice for complex, interdependent time series data.

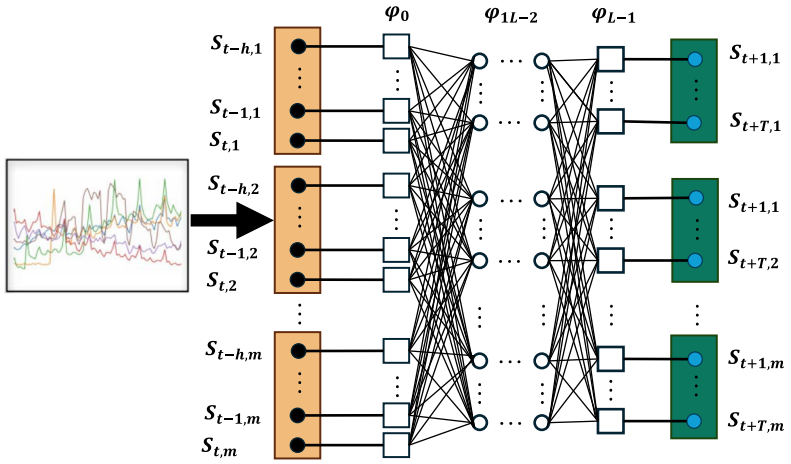

Fig. 3. MT-KAN architecture for multivariate time series [60].

Figure 3 shows the MT-KAN architecture for multivariate time series, where past values of multiple variables are processed through shared network layers $\Phi_0, \Phi_{1:L-2}, \Phi_{L-1}$ to capture temporal and cross-variable dependencies. This setup allows MT-KAN to model complex interactions between variables and improve forecasting accuracy for interdependent tasks. The final layer outputs future values for each variable over the predicted horizon.

3.1.6 SigKAN. SigKAN enhances KANs capabilities in time series prediction by incorporating path signatures, which capture essential geometric features of time series paths [32]. By integrating these signatures with KAN's output, SigKAN provides a more comprehensive representation of temporal data, effectively capturing complex time series patterns through iterated integrals. The core architecture of SigKAN includes a *learnable path signature layer* that computes path signatures for each input sequence, allowing the network to capture sequential dependencies and intricate path structures. These signatures are combined with the KAN output through a **Gated Residual KAN** (**GRKAN**) Layer, which modulates information flow by applying weighting mechanisms to enhance relevant features and suppress noise.

The output of SigKAN can be described by the following equation:

$$y = \psi \odot \text{KAN}(X), \tag{10}$$

where $\psi = \text{SoftMax}(\text{GRKAN}(S(X)))$ is the weighted output from the Gated Residual KAN layer, and $S(X)$ represents the path signature of the input sequence X. The path signature $S(X)$ includes iterated integrals of the input sequence, capturing complex temporal patterns across different scales.

Figure 4 illustrates the SigKAN architecture, where input data $X_n = (x_{n,t-1}, x_{n,t-2}, \ldots, x_{n,t-p})$ is first processed by learnable weight coefficients w_n to create $\hat{X}_n$. This representation passes through the SigLayer to compute path signatures $S_k(\hat{X}_n)$, capturing essential path features. The signatures are then fed into the GRKAN module, which weights and modulates this information. Simultaneously, $\hat{X}_n$ is passed to the KAN layer, and a Dropout layer is applied for regularization. The final output Y_t is obtained by combining the GRKAN-weighted signatures with the KAN output, offering a robust prediction that leverages both path signature geometry and KAN's functional approximation capabilities.

This integration of path signatures with KAN's functional approximation allows SigKAN to adapt dynamically to nonlinear relationships in time series, making it effective in tasks like financial modeling and multivariate forecasting. As demonstrated by Inzirillo and Genet [32], SigKAN

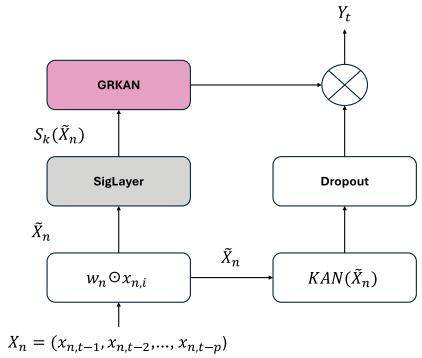

Fig. 4. SigKAN architecture [32].

significantly outperforms traditional KAN models, achieving both improved accuracy and a deeper understanding of temporal dependencies.

3.1.7 GKAN. **Graph KAN (GKAN)** extends the KAN framework to graph-structured data by introducing learnable univariate functions on graph edges, replacing the fixed convolutional structure of traditional **Graph Convolutional Networks (GCNs)** [64]. GCNs operate by iteratively aggregating and transforming feature information from local neighborhoods within a graph, effectively capturing both node features and graph topology. This approach, pioneered by [124], has proven effective for various applications, including node classification and recommendation systems. However, GCNs rely on fixed convolutional filters, which limits their flexibility in handling complex, heterogeneous graphs.

To address this limitation, GKAN introduces two primary architectures: Architecture 1, which aggregates node features before applying KAN layers, allowing learnable activation functions to capture complex local relationships, and Architecture 2, which places KAN layers between node embeddings at each layer before aggregation, allowing for dynamic adaptation to changes in graph structure. Formally, in Architecture 1, the embedding of nodes at layer $\ell + 1$ is represented as

$$H^{(\ell+1)}_{\text{Archit.1}} = \text{KANLayer}(\hat{A}H^{(\ell)}_{\text{Archit.1}}), \tag{11}$$

where $\hat{A}$ is the normalized adjacency matrix, and $H^{(0)}_{\text{Archit.1}} = X$ (input features). In Architecture 2, the process is reversed:

$$H^{(\ell+1)}_{\text{Archit.2}} = \hat{A}\,\text{KANLayer}(H^{(\ell)}_{\text{Archit.2}}). \tag{12}$$

This flexible setup enables GKAN to adapt effectively to large-scale and heterogeneous graph data by optimizing feature representation across evolving graph structures.

Figure 5 provides a visual comparison between the traditional GCN and the two proposed GKAN architectures. Part (a) illustrates the GCN setup, where convolutional layers are applied directly on node embeddings. Part (b) demonstrates GKAN Architecture 1, which aggregates node features before passing them through KAN layers, while Part (c) depicts Architecture 2, where KAN layers are applied to individual node embeddings prior to aggregation. This figure highlights how each GKAN architecture processes graph data differently, enabling greater adaptability and flexibility compared to standard GCNs.

4 Practical Considerations in Training KANs

While the previous section explored the architectural foundations and domain-specific extensions of KANs, their real-world effectiveness depends equally on practical training strategies. This

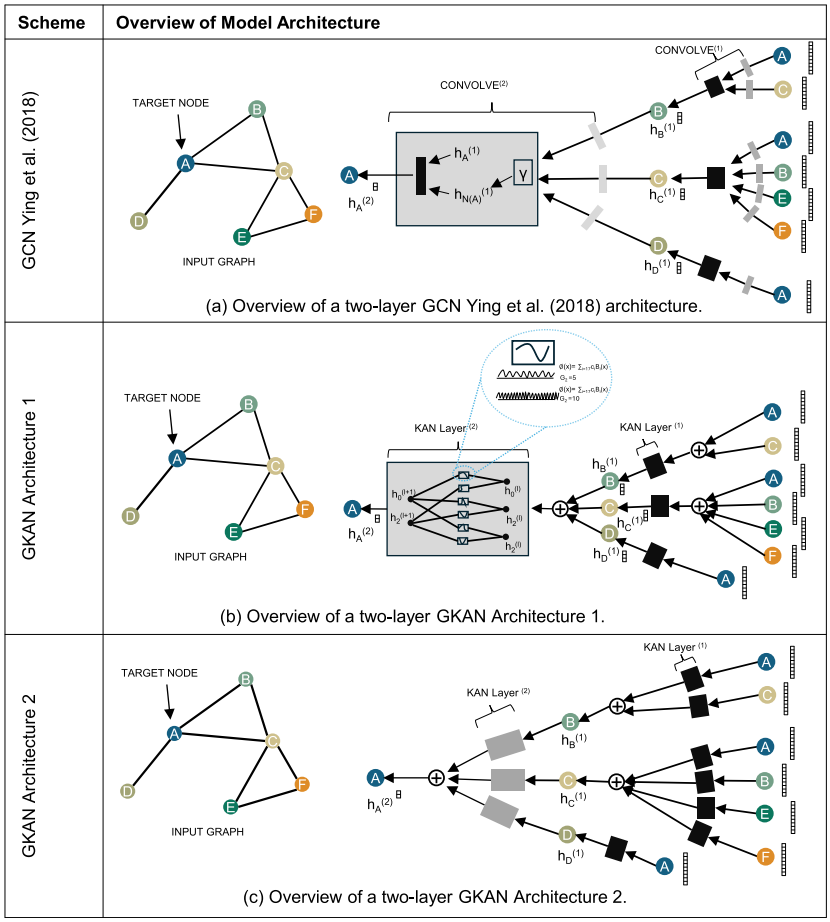

Fig. 5. Comparison of different GKAN model architectures [64].

section outlines key considerations including optimization techniques, initialization methods, and regularization approaches that enable stable training, generalization, and reliable deployment of KAN models. Table 1 provides a structured summary of recent KAN variants, highlighting their core architectures, training strategies, and distinctive features across four categories: interpretability, sequential modeling, optimization, and efficiency. This overview helps contextualize how various techniques are tailored to address performance bottlenecks and training challenges in different application settings.

4.1 Optimization Strategies

Optimization in KAN is essential due to the complexity of the high-dimensional function approximations they perform. KAN's power lies in its ability to decompose multivariate functions into univariate spline functions, but the effectiveness of this process depends on the optimization of these splines. Optimization adjusts the control points and knots of the splines to minimize errors between predicted and actual outputs, allowing the model to capture intricate data patterns. However, the high dimensionality and complexity of these approximations introduce several challenges:

Table 1. KAN-based Studies on Different Performance Measures

Model (Year) Source	Architecture	Training Process	Main Features
Interpretability			
KAN 2024 [8]	Learnable spline-based edge activations	Gradient descent with LBFGS, adaptive grid	High interpretability, efficient scaling laws
SKAN 2024 [15]	Structured, smooth nested functions	RMSE minimization, data-efficient, extrapolation in sparse data regions	High interpretability, scalable, effective in sparse data, smooth function representation
Rational KAN 2024 [65]	Rational basis functions (Padé, Jacobi)	Gradient descent (L-BFGS, Adam)	High accuracy, effective for physics-informed tasks and complex approximations
S-KAN 2024 [77]	Adaptive multi-activation nodes	Full training, selective, pruning	Flexible activation, robust fitting, improved generalization
MultKAN 2024 [92]	KAN layers with addition and multiplication nodes	Gradient descent, sparse regularization	High interpretability, modularity, handles multiplicative structures
RKAN 2024 [99]	Chebyshev polynomial-based KAN convolutions	SGD for small datasets, AdamW for large datasets	Improved feature representation, robust gradient flow, adaptable to CNNs, computationally efficient
LSin-SKAN 2024 [105]	Sine-based, single parameter	Gradient descent, stable, fast convergence	Efficient, high performance among SKAN variants
LCos-SKAN 2024 [105]	Cosine-based, single parameter	Gradient descent, moderate convergence, oscillates	Efficient, slightly lower accuracy than LArctan-SKAN
LArctan-SKAN 2024 [105]	Arctangent-based, internal scaling	Gradient descent, stable, fast convergence, high accuracy	Best accuracy, highly efficient, stable training
LSS-SKANs 2024 [108]	Single-parameterized shifted Softplus	Adam optimizer, fine-tuned learning rate	High efficiency, superior accuracy, strong interpretability, suitable for MNIST and similar tasks
Sequential			
iKAN 2024 [103]	Multi-encoder, KAN-based classifier, feature redistribution layer	Two-step: encoder and frozen KAN-based classifier training	High incremental learning performance, reduces catastrophic forgetting, supports heterogeneous data, uses local plasticity
HiPPO-KAN 2024 [109]	HiPPO-encoded, spline-based KAN	Gradient descent, MSE loss	Parameter-efficient, captures long dependencies, reduced lagging
WormKAN 2024 [114]	KAN-based encoder-decoder	Reconstruction loss, regularization, smoothness	High interpretability, concept drift detection
BiLSTMKANnet 2024 [115]	BiLSTM + DenseKAN layers	10-fold CV, GridSearchCV	Temporal dependency capture, interpretability, adaptable to sequence data
T-KAN 2024 [60]	Spline-based, learnable edge activations	Sliding window, gradient descent, pruning	High interpretability, concept drift detection
TKANs 2024 [24]	RKAN layers with LSTM gating, B-spline activations	Adam optimizer, RMSE loss, early stopping	High accuracy in long-term forecasting, stable training, effective memory management for sequential data

(Continued)

Table 1. Continued

Model (Year) Source	Architecture	Training Process	Main Features
Sequential			
MT-KAN 2024 [60]	Spline-based, learnable edge activations with cross-variable interactions	Gradient descent, pruning for efficiency	High interpretability, improved multivariate forecasting
SigKAN 2024 [32]	Gated Residual KAN, path signature layer	Adam optimizer, early stopping	Accurate short-term forecasting, stable, captures temporal dependencies in complex time series
TKAT 2024 [68]	Encoder-decoder with TKAN layers, self-attention	Adam optimizer, MSE loss, early stopping	High interpretability, captures temporal dependencies, suited for multivariate time series
Optimization			
FastKAN 2024 [26]	Gaussian RBFs with layer normalization	Benchmarked on MNIST, 20 epochs	3.3x faster than KAN, simplified implementation, retains accuracy for high-dimensional functions
FBKANs 2024 [74]	Domain decomposition, spline-based local KANs per subdomain	Combined data-driven and physics-informed loss, adaptive grids, parallel training	Scalable, noise-robust, compatible with enhancements
BSRBF-KAN 2024 [75]	Combines B-splines and Gaussian RBFs	15 epochs (MNIST), 25 (Fashion-MNIST), AdamW optimizer	High accuracy, fast convergence, adaptable activations
DKL-KAN 2024 [87]	Three-layer KAN with spline activation; KISS-GP and SKIP for scalability	Normalization, Adam optimizer, 2500 epochs on GPU	Captures discontinuities, excels on small datasets, reliable uncertainty estimates in sparse data
DropKAN 2024 [89]	Spline-based, post-activation masking	Adam optimizer, 2000 steps	Improved generalization, prevents co-adaptation, flexible scaling
Efficiency			
FKANs 2024 [91]	Spline-based, learnable activations	Federated averaging, local training	High interpretability, privacy-preserving, fast convergence, stable performance
MFKAN 2024 [110]	Low-fidelity, linear, and nonlinear KAN blocks	Pretrained low-fidelity; multifidelity training	Efficient with sparse high-fidelity data, adaptive linear and nonlinear modeling, data efficiency

— **Non-linear Parameter Space:** Unlike traditional neural networks, KAN involves adjusting spline parameters, which makes the optimization landscape non-linear and harder to navigate.
— **Exponential Growth in Parameter Space:** As the input dimensionality increases, the number of learnable parameters expands exponentially, making the model more prone to overfitting and slower convergence.
— **Computational Overhead:** The flexibility of learnable splines increases the computational burden during training, making optimization slower and more resource-intensive [119].

Given the challenges of optimizing KAN, several techniques have been employed to improve convergence and performance:

— **Gradient Descent and its Variants:** Since KAN parameters (spline control points) are optimized using gradient-based techniques, variants of gradient descent like **Stochastic Gradient Descent (SGD)** with momentum are often used to help smooth the optimization path and avoid local minima. These techniques mitigate the difficulties of non-linear optimization by using momentum to escape saddle points [119].

— **Adam Optimizer:** Adam is another popular optimization technique that combines momentum and adaptive learning rates, making it highly effective for training KAN models. Adam's ability to adjust the learning rate for each parameter individually is beneficial for optimizing KAN's complex spline functions [119].
— **Regularization Techniques:** To prevent overfitting in high-dimensional spaces, L2 regularization and dropout are commonly used. These techniques help constrain the flexibility of the splines and prevent them from fitting the noise in the training data [119].
— **Batch Normalization:** To stabilize training and speed up convergence, batch normalization is often applied in KAN layers. This technique helps address the vanishing or exploding gradient problem, which is common in deep and complex networks [119].

Convergence during KAN training is a known challenge due to the high dimensionality and non-linear optimization landscape [25, 76]. Several articles have highlighted specific issues related to sensitivity to initialization and slow convergence in high-dimensional spaces:

— **Sensitivity to Initialization:** Poor initialization of spline parameters can lead to suboptimal convergence or cause the optimization to get stuck in local minima. Research has shown that careful initialization strategies, such as He initialization or Xavier initialization, can mitigate these issues by providing a better starting point for optimization [8, 119].
— **Slow Convergence:** Due to the large number of learnable parameters in KAN (especially when dealing with high-dimensional data), optimization can converge very slowly. This is compounded by the need to optimize both the shape and position of the splines, making the process more complex than standard neural networks. Advanced techniques such as second-order optimizers (e.g., L-BFGS) have been proposed to speed up convergence by leveraging curvature information of the loss landscape [8, 119].
— **Regularization and Dropout:** Overfitting is a common issue due to the flexible nature of splines. To address this, researchers have proposed dropout during training, which helps improve generalization by randomly removing units and their connections, reducing overfitting risks [119].
— **Optimization Instabilities:** Researchers have also discussed optimization instabilities in KAN, especially in high-dimensional spaces. One common issue is the tendency of optimization algorithms to converge to local minima rather than global ones, particularly when dealing with spline-based architectures. Early stopping and learning rate schedules have been recommended to help address these convergence difficulties by preventing overfitting and ensuring smoother optimization [8, 119].

KANs incorporate unique optimization strategies due to their spline-based architecture and dynamic activation functions. The key optimization approaches are listed below:

(1) **Spline-based Learnable Weights:** The weights in KANs are not fixed linear transformations as in traditional models but are instead learnable spline functions. These weights are optimized using gradient-based methods like backpropagation, similar to other neural networks. However, specific adjustments are required to handle the B-spline functions used in KANs. To address this:
   — Spline grid updates are implemented during training to adjust the locations of knots in B-splines, allowing the splines to adapt dynamically as the network learns.
   — Gradient descent is employed to optimize the coefficients of the spline functions, with methods such as SGD or Adam being commonly used.
(2) **Variance-preserving Initialization:** KAN models require careful initialization of the spline coefficients to ensure stable training. A variance-preserving initialization is often used to maintain the variance of activations across layers. This ensures that the model does not

Table 2. Optimization Strategies in KAN vs. Other Models

Optimization Feature	KAN	CNN	RNN	MLP	Transformer
Weight Representation	Learnable spline-based weights [8, 92]	Fixed linear weights [12]	Fixed linear weights [93]	Fixed linear weights (matrix multiplications) [128]	Fixed linear weights (query/key/value projections) [127]
Activation Functions	Spline-based activation on edges [8, 92]	Non-linear activations on nodes (e.g., ReLU) [94]	Non-linear activations with gated mechanisms [95]	Non-linear activations on nodes (e.g., ReLU, sigmoid) [96, 97]	Non-linear activations (e.g., GELU, ReLU) [127]
Initialization	Variance-preserving initialization of spline grids [17]	Random initialization [19]	Orthogonal or identity initialization for stability [20]	He/Xavier initialization (scaled by layer size) [21]	Xavier/Glorot or scaled initialization (for attention layers) [127]
Residual Connections	Residual activation with spline-basis [99]	Common (e.g., ResNets) [100]	Less common; more typical in LSTMs [101]	Less common (but used in deep MLPs) [102]	Essential (pre-norm/post-norm residual connections) [127]
Computational Complexity	Efficient with spline optimizations (e.g., FastKAN) [47, 62]	Moderate (requires large networks for performance) [27]	High (requires backpropagation through time) [127]	Moderate (scales with width/depth) [12]	High (quadratic in sequence length due to self-attention) [127]

suffer from vanishing or exploding gradients during optimization, which is crucial given the non-linear nature of KANs.

(3) **Residual Activation Functions:** To improve convergence during training, KANs often employ residual activations. These are combinations of basic activation functions, $b(x)$ (e.g., SiLU) and learned spline functions. This structure is inspired by the residual connections used in CNNs, which have proven effective in accelerating convergence and mitigating vanishing gradient issues:

$$\phi(x) = w_b \cdot b(x) + w_s \cdot \text{spline}(x). \tag{13}$$

The weights w_b and w_s control the contributions of the basic activation and the spline, respectively. This structure introduces non-linearity via the basic activation, while the spline models more complex, fine-grained relationships in the data. Together, they improve convergence, allowing KANs to learn smoothly and effectively balance both components during training.

(4) **Efficient Splines for Faster Computation:** Given that traditional B-splines can be computationally expensive, some variants of KAN, like FastKAN, replace B-splines with more efficient Gaussian RBFs. This reduces the computational complexity and speeds up both forward and backward passes during optimization.

Table 2 compares the key optimization features of CNNs, RNNs, and KANs. CNNs, introduced by Yann LeCun [125] in 1989, employ fixed linear weights and non-linear activation functions (such as ReLU), making them highly effective for image recognition tasks like classification and object detection. RNNs, conceptualized by John Hopfield [126] in 1982 and later refined by Ronald J. Williams and David E. Rumelhart [128], are designed for sequential data processing, using fixed linear weights and non-linear gated mechanisms to model temporal dependencies in tasks such as language modeling and time series forecasting. In contrast, KANs extend neural network capabilities by incorporating learnable spline-based weights and activations, providing greater flexibility and interpretability compared to CNNs and RNNs.

KANs also utilize variance-preserving initialization and offer computational efficiency, especially when optimized with methods like FastKAN. Meanwhile, CNNs use random initialization, and RNNs often rely on orthogonal or identity initialization to maintain stability, with the latter being more computationally intensive due to backpropagation through time. Additionally, KANs leverage residual activations through spline bases, while residual connections are common in CNNs (e.g., ResNets) and less typical in RNNs, though they appear more frequently in models like LSTMs.

4.2 Regularization Techniques

KANs utilize several regularization techniques to mitigate overfitting, a critical challenge in machine learning models. The most common techniques include the following:

(1) **Dropout:** Dropout is a popular regularization method where randomly selected neurons are ignored during training. This prevents the network from becoming too reliant on specific neurons, which can lead to overfitting. By dropping out a random fraction of neurons, the network is forced to learn more general features that are robust to different inputs. Although dropout is common in traditional neural networks, its implementation in KANs depends on their architecture and how the inner and outer functions are modeled.

$$h_i = \text{Dropout}(f_i(x), p), \tag{14}$$

where h_i is the hidden layer output after dropout, $f_i(x)$ is the function applied by the hidden layer, and p is the dropout probability.

(2) **Weight Regularization (L2 Norm or Ridge Regularization):** In KANs, weight regularization is applied to penalize large weights, which could indicate overfitting. The L2 norm, also known as Ridge regularization, adds a penalty to the loss function proportional to the square of the magnitude of the weights. This technique encourages the network to maintain smaller weights, resulting in a smoother model that is less likely to overfit the training data.

$$\text{Loss} = \text{MSE} + \lambda \sum w_i^2, \tag{15}$$

where λ is the regularization strength, and w_i are the network weights.

(3) **Early Stopping:** Early stopping is a regularization technique that monitors the model's performance on the validation dataset during training. If the model's performance on the validation set starts to degrade while continuing to improve on the training set, training is stopped. This prevents the model from becoming overly fitted to the training data. KAN models can implement this technique, as shown in various deep learning tasks [65].

(4) **Constraints on Inner and Outer Functions:** In KANs, constraints are often applied to the inner and outer functions to ensure they maintain certain properties that improve generalization. For example, the smoothness or continuity of the functions is often controlled through constraints on their derivatives. These constraints help the model avoid fitting noise and ensure that the functions remain interpretable and generalizable [129].

(5) **Gradient Clipping:** Gradient clipping prevents the problem of exploding gradients during backpropagation by limiting the size of gradients during training. This ensures that the training process remains stable and prevents the model from converging to sharp, overfitting-prone solutions. Though not unique to KANs, gradient clipping is another useful regularization technique in complex architectures like these [129].

(6) **Sparsity Constraints:** KANs can also incorporate sparsity regularization, where additional terms are added to the loss function to encourage sparse activations. This means that only a few neurons are active at a time, reducing the model's complexity and helping prevent overfitting by making the model more interpretable and efficient [51].

5 Applications, Benchmarks, and Comparative Evaluation of KANs

5.1 Applications Across Scientific and Engineering Domains

The rapid advancement of KAN architectures has given rise to a diverse range of model variants, each tailored to address specific computational challenges and domain-specific requirements. While earlier sections focused on training strategies and optimization techniques, it is equally important to understand how these innovations translate into practical applications. This section

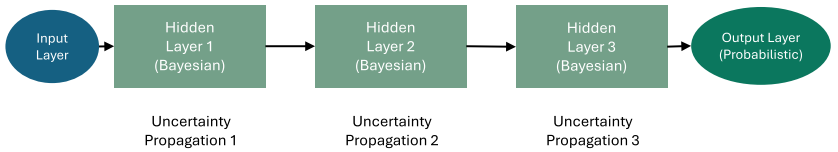

Fig. 6. Hierarchical Bayesian structure in BKANs, illustrating uncertainty propagation through each layer to produce probabilistic outputs for robust decision-making [130].

explores the deployment of KAN models across a broad range of fields-including time series forecasting, computational biomedicine, graph learning, physics-informed modeling, signal processing, computer vision, and transportation systems highlighting their versatility, effectiveness, and growing impact in real-world scenarios.

To help readers navigate the landscape of KAN applications, Table 3 provides a structured summary of key KAN-based models. It categorizes them across some key application domains. Each entry highlights the core architecture, training strategy, and primary strengths of the model, illustrating the versatility and impact of KANs in real-world research problems.

5.1.1 Time Series Forecasting. KANs and their variants (e.g., T-KAN, MT-KAN, SigKAN) have shown strong performance in time-series tasks such as satellite traffic, cryptocurrency forecasting, electricity load prediction, and air quality monitoring [16, 24, 32, 60]. Their adaptive splines efficiently capture temporal dependencies and concept drift with fewer parameters than traditional models. TKANs further incorporate memory mechanisms (like LSTM) for improved sequential learning in domains such as finance and healthcare.

5.1.2 Computational Biomedicine and Survival Analysis. KANs are increasingly used in medical applications due to their reliability, interpretability, and ability to operate with limited or imbalanced data. **BKANs** (**Bayesian KANs**) model, as shown in Figure 6, both aleatoric and epistemic uncertainty, improving diagnostic confidence in clinical settings [130]. TKANs assist in patient monitoring and disease progression modeling [24]. CoxKAN supports symbolic hazard function modeling for survival analysis and biomarker identification [131].

5.1.3 Graph Learning and Networked Data. Extensions such as GKAN, KAGNN, and KAGCN replace fixed convolutions with spline-based learnable functions to improve node representation, classification, and regression in graphs [55, 56]. Applications include social networks, molecular property prediction, and knowledge graphs, with improved modularity and interpretability over conventional GCNs.

5.1.4 Physics-informed Modeling and Scientific Computing. Physics-informed KANs (PIKANs and PDE-KANs) are applied to solve partial differential equations by incorporating domain-specific constraints and adaptive spline bases. cPIKANs and FBKANs address sparse PDE observations in quantum mechanics and fluid dynamics with improved convergence and accuracy compared to traditional PINNs [49, 90, 133].

5.1.5 Signal Processing and Spectral Learning. KANs demonstrate noise-robust capabilities in signal processing and hyperspectral imaging. Wav-KAN preserves spectral fidelity through wavelet-domain decomposition and excels in hyperspectral classification (e.g., Indian Pines) [23]. Kernel-filtered KANs have shown resilience in noisy time series and environmental sensing contexts [54].

5.1.6 Power Systems and Tabular Data. KANs perform well on structured tabular data, offering compactness and interpretability. FastKAN, which replaces B-splines with Gaussian kernels,

Table 3. KAN-based Studies in Different Applied Research Areas

Model (Year) Source	Architecture	Training Process	Main Features
Physics-Informed Modeling			
KINN 2024 [38]	Spline-based, B-spline activations	Gradient descent, meshless sampling, triangular integration	High interpretability, efficient for PDEs, handles multi-frequency components, low spectral bias
PDE-KAN 2024 [38]	Spline-based, B-spline activations with tanh normalization	Meshless sampling, triangular integration	Low spectral bias, efficient for multi-frequency, adaptable to complex PDEs
PIKANs 2024 [66]	Single-layer KAN, polynomial activations (e.g., Chebyshev)	Adam optimizer, RBA for loss adjustment	High accuracy with fewer parameters, adjustable polynomial order, enhanced stability with double precision
DeepOKANs 2024 [66]	Branch [128, 100, 100, 100], Trunk [4, 100, 100, 100], Chebyshev polynomials	Adam optimizer, 200k iterations, L2 regularization	Robust to noise, strong in complex tasks, higher computational cost
RLDK 2024 [67]	Spline-based, compact architecture with learnable edge activations	LBFGS optimizer with custom loss (reconstruction and prediction)	High parameter efficiency, fast training, suitable for real-time control, data-efficient
fKAN 2024 [76]	Fractional Jacobi Neural Block (fJNB) with trainable α, β, γ	Adam and L-BFGS optimizers	High adaptability, improved accuracy, suitable for deep learning and physics-informed tasks
PIKANs 2024 [90]	Adaptive KAN with spline-based activations	Adaptive gradient descent with grid updates	High accuracy, efficient PDE solver, customizable basis functions
KAN-ODE 2024 [98]	RBF-based, learnable Swish activations	Gradient descent, adjoint method, pruning	High interpretability, efficient on sparse data, symbolic learning
EPi-cKAN 2024 [112]	Chebyshev polynomial-based, interconnected sub-networks	Physics-informed MSE loss; step-decay learning rate	Accurate stress-strain predictions; combines physics and data-driven insights; efficient parameter use
Computer Vision and Forensics			
Wav-KAN 2024 [23]	Wavelet-based, learnable edge functions	Batch norm, AdamW optimizer, grid search for wavelets	High interpretability, multi-resolution analysis, noise robustness, efficient for high-dimensional data
C-KAN 2024 [29]	Spline-based, learnable convolutions	Gradient descent with regularization, grid updates	High efficiency, adaptable activations, competitive accuracy
KCN 2024 [71]	Spline-based edge activations	Gradient descent with backpropagation; layer freezing	High accuracy, parameter-efficient, adaptable to complex data
KANICE 2024 [79]	ICBs with 3x3 and 5x5 convolutions, KANLinear spline layers	25 epochs on image datasets, batch normalization	Adaptive feature extraction, universal approximation, efficient in KANICE-mini
Kaninfradet3D 2024 [107]	KAN layers, cross-attention, KANvtransform, ConKANfuser	AdamW optimizer, staged training, Cosine Annealing	Enhanced feature fusion, high-dimensional data handling, improved 3D detection accuracy
PointNet-KAN 2024 [113]	Shared KAN layers, Jacobi polynomials, permutation invariance	Adam optimizer, batch norm, max pooling	Competitive, efficient for 3D point cloud classification and segmentation

(Continued)

Table 3. Continued

Model (Year) Source	Architecture	Training Process	Main Features
Graph Learning and Networked Data			
GraphKAN 2024 [61]	Spline-based, learnable edge activations	Cosine Annealing, LayerNorm, 200 epochs	High accuracy, effective for few-shot classification
GKAN 2024 [64]	Learnable spline functions on edges	Semi-supervised with backpropagation	Efficient, accurate, adjustable parameters for large-scale graph data
KAGNNs 2024 [55]	KAN-based, spline-driven updates	Gradient descent with early stopping	High interpretability, strong expressiveness, suited for graph regression and classification
FourierKAN-GCF 2024 [73]	Fourier-based GCN with Fourier KAN replacing MLP	BPR loss, grid search, message and node dropout	Efficient, strong interaction representation, robust, adaptable, easier training than spline-KAN
KA-GNN 2024 [111]	Fourier-based, learnable edge activations, 5Å cut-off for bonds	Cross-entropy loss, Fourier message passing	High interpretability, parameter-efficient, robust molecular modeling
Quantum Computing			
KANQAS 2024 [70]	Spline-based, learnable activations in Double Deep Q-Network (DDQN)	Gradient descent with RL in DDQN	High interpretability, parameter-efficient, effective for quantum state prep and quantum chemistry
QCKANnet 2024 [115]	Conv1DKAN + Quantum layers	Keras tuner, cross-validation	Efficient pattern recognition, quantum-classical hybrid, scalable
QDenseKANnet 2024 [115]	DenseKAN layers + Quantum circuits	Hyperband tuning, 10-fold CV	Nonlinear modeling, high accuracy, robust for complex data
QKAN 2024 [116]	Block-encoded layers with Chebyshev activations	Quantum circuits, parameterized learning	Handles high-dimensional data, efficient for complex approximations
Specialized Areas			
KAN2CD 2024 [25]	Two-level KANs with learnable embeddings	Adam optimizer, B-spline KANs for efficiency	High interpretability, efficient training, competitive accuracy, suitable for cognitive diagnosis
KAN-EEG 2024 [69]	Spline-based, learnable edge activations	100 epochs on EEG data, with gradient descent and epoch-based convergence	High interpretability, efficient, adaptable across datasets, suitable for on-device deployment
U-KAN 2024 [72]	Encoder-decoder with tokenized KAN layers	Cross-entropy and Dice loss for segmentation; MSE for generation	High interpretability, efficient, adaptable for segmentation and generative tasks, improved accuracy
COEFF-KANs 2024 [86]	MoLFormer for chemical embeddings, followed by KAN layers	Fine-tuning with AdamW optimizer	High interpretability, data-efficient learning, complex relationship modeling, enhanced by CIDO method
CaLMPhosKAN 2024 [88]	Fused codon and amino acid embeddings; ConvBiGRU; Wavelet-based KAN (DoG wavelet)	Binary cross-entropy loss, Adam optimizer, 10-fold cross-validation	High accuracy for phosphorylation site prediction, effective for disordered regions, rich feature representation
SCKansformer 2024 [104]	KAN-based Kansformer, SCConv Encoder, GLAE	Adam optimizer, data augmentation, 100 epochs	High interpretability, reduced redundancy, effective global-local feature capture, fine-grained classification

(Continued)

Table 3. Continued

Model (Year) Source	Architecture	Training Process	Main Features
Specialized Areas			
KANsformer 2024 [106]	Transformer encoder, KAN decoder with splines	Unsupervised, Adam	Scalable, real-time inference, interpretable, supports transfer learning

reduces computational overhead while maintaining accuracy in power grid monitoring and tabular classification tasks [26, 132]. These models outperform MLPs in parameter efficiency and generalization.

5.1.7 Computer Vision and Forensics. Hybrid architectures such as KAN-Mixer and KANICE integrate CNN layers with spline-based components for robust visual recognition and forensic applications [78, 79]. These models are applied in object detection and the classification of AI-generated imagery, where interpretability and spatial awareness are crucial.

5.1.8 Transportation and Mobility. TrafficKAN-GCN combines KAN's nonlinear modeling with GCNs for spatial-temporal traffic forecasting and safety prediction [81]. The model showed strong performance on real-world datasets, including managing congestion and disruptions during incidents like the Francis Scott Key Bridge collapse.

This domain-based organization highlights the versatility of KANs across real-world applications, from modeling complex temporal signals to advancing biomedical diagnostics and physics-based simulations. The architecture's adaptability and interpretability position it as a promising alternative to conventional neural network models in both research and deployment settings.

5.2 Critical Evaluation and Comparative Analysis

Comparison with MLPs: As MLPs form the foundation of many neural architectures, it is important to understand how KANs relate to and diverge from them. Both KANs and MLPs are universal function approximators, but they differ significantly in how they model functional dependencies. MLPs rely on fixed-weight matrices and nonlinear activations (such as ReLU or SiLU), making them efficient but often opaque in terms of interpretability [8]. In contrast, KANs replace these fixed weights with learnable univariate functions (typically spline-based) assigned to each edge in the network [8]. This edge-based functional representation allows for symbolic expressions that are often considered more interpretable in certain scientific and regression contexts. FastKAN, for instance, achieves similar or better accuracy than MLPs on image classification tasks like MNIST while being significantly more parameter-efficient and interpretable [8, 47]. However, MLPs benefit from decades of optimized training strategies, better toolchain support, and stable convergence in large-scale applications, making them a more accessible baseline for general-purpose deep learning.

KANs have also been empirically shown to outperform MLPs in real-world regression and time-series forecasting problems. For example, in electrohydrodynamic pump modeling, KANs achieved higher predictive accuracy and interpretability than both MLPs and Random Forests by capturing nonlinear dynamics more effectively [13]. Similarly, in satellite traffic prediction, KANs outperformed MLPs with fewer parameters, reinforcing their strength in data-efficient forecasting [16]. **Temporal KANs (TKAN)** further integrate memory-based structures to outperform MLPs and RNNs in long-horizon multistep time series prediction [24]. Additionally, recent theoretical analyses demonstrate that KANs achieve lower generalization bounds than MLPs in compositional

or low-dimensional settings, providing both interpretability and robustness in data-scarce environments [119]. Nevertheless, the interpretability advantage of KANs may not generalize across all domains; in complex expressions or perceptual tasks like vision, symbolic forms can still be opaque, and interpretability remains context-dependent.

Comparisons with CNNs and Transformers: While KANs were originally proposed as an interpretable alternative to MLPs [8, 92], their capabilities now extend to domains traditionally dominated by CNNs and Transformer-based models. Wav-KAN and RKAN have demonstrated strong performance in structured imaging tasks and physics-informed modeling, leveraging spline-based edge activations to outperform CNNs in terms of parameter efficiency and symbolic expressiveness [48, 65]. For example, Wav-KAN achieves enhanced spectral decomposition in hyperspectral image classification, while RKAN has shown advantages in PDE-driven simulation tasks.

KAN-based hybrids with Transformers have also emerged in recent work. Models like KANsformer [121, 122] and ViKANformer [123] demonstrate that KANs can replace MLP blocks within Vision Transformers to improve interpretability and adaptability. These hybrids maintain self-attention structures while introducing functional modularity, which results in competitive accuracy and increased transparency for vision tasks.

Key Advantages: One of the most prominent advantages of KANs is their interpretability. Models such as MultKAN and SigKAN can uncover symbolic representations of learned functions, which is highly beneficial for domains like medicine, chemistry, and scientific modeling [32, 33]. This symbolic bias also contributes to superior data efficiency: in medical image segmentation, for instance, Semi-KAN and UNetVL outperform CNN-based U-Nets and Transformer-based models in low-data settings by leveraging fewer but more expressive parameters [34, 35]. In privacy-sensitive environments like federated learning, Fed-KAN maintains accuracy while providing transparent decision processes across nodes [36]. KANs also excel in scientific discovery; FastKAN, rKAN, and FourierKAN-GCF have been applied to learn symbolic forms of PDEs, simulate molecular dynamics, and extract governing physical laws with generalizability across regimes [26, 65, 73].

Core Limitations: Despite these strengths, KANs face challenges. Scalability is a notable issue: their computational complexity increases sharply with dimensionality due to spline parameterization and adaptive grid learning, as seen in molecular dynamics modeling [52] and tumor inpainting [28]. Training stability is also an ongoing concern. Unlike MLPs and CNNs, KANs require specialized techniques such as control point tuning, grid pruning, or wavelet adaptation. Approaches like DropKAN and LoTRA address this with dropout-based regularization and tensor-rank compression, though they do not fully resolve training instability [63, 89].

Benchmarking and Adoption: A further issue is the lack of standardized benchmarks. Many KAN variants are evaluated on small-scale or synthetic datasets with limited comparability. The absence of public toolkits and pre-trained models makes it difficult to assess KANs against mainstream CNNs (e.g., EfficientNet) or Transformer variants (e.g., Swin, ViT). Recent exceptions like KANICE and ViKANformer [79, 123] help bridge this gap by offering open implementations and reproducible results, but more community-driven benchmarks are needed.

Tradeoffs and Hybrid Potential: KANs embody tradeoffs between expressiveness, interpretability, and computational overhead. Their spline-based activations enable local adaptivity and symbolic insight but come at the cost of increased training complexity. In contrast, traditional deep models provide stability and toolchain maturity at the cost of being black-box in nature. KANs tend to generalize well on structured or low-data problems but struggle in high-noise, unstructured domains where overfitting may occur [22]. Hybrid approaches such as SCKansformer and KANMixer have begun to address this by integrating KAN layers into CNN or Transformer pipelines [78, 104], offering a pragmatic middle ground for practical deployment.

In summary, KANs offer a promising paradigm for interpretable and scientifically grounded machine learning, particularly in data-constrained or knowledge-driven tasks. However, to gain broader adoption, they must overcome issues of training stability, scalability, and tooling. A concerted effort to create robust benchmarks, share reproducible implementations, and improve optimizer support will be crucial in the coming years.

6 Challenges and Limitations

6.1 Computational Complexity

KANs have garnered increasing attention as a promising alternative to traditional neural network architectures, such as MLPs, due to their unique ability to decompose multivariate functions into simpler, univariate components. Despite these advantages, the implementation of KANs in high-dimensional spaces presents significant computational challenges, particularly in terms of optimization, training time, and resource usage. One of the primary issues is the non-convex nature of the optimization problem, which leads to slower convergence and longer training times compared to conventional neural networks. For instance, Bresson et al. [55] noted that KANs struggle with high computational complexity in graph learning tasks, which increases training time when compared to MLPs.

Additionally, KANs are particularly sensitive to noise in the training data, which further exacerbates computational inefficiencies. Shen et al. [54] highlighted how the presence of noise significantly degrades KAN performance, necessitating costly noise-reduction techniques such as kernel filtering and oversampling, which further inflate the computational overhead. Similarly, Nagai and Okumura [52] explored the use of KANs in molecular dynamics simulations and found that while KANs can reduce computational costs for low-dimensional tasks, their scalability is limited in high-dimensional settings due to the complexity of spline interpolation used for function approximation.

Moreover, KANs tend to demand more computational resources than black-box models like MLPs. As Sun [44] pointed out, KANs often require additional computational power and training time to handle complex, high-dimensional tasks, even though they offer advantages in interpretability and symbolic function generation. In addition, hardware implementation of KANs poses another significant challenge. A study by Tran et al. [45] revealed that KANs consume substantially higher hardware resources than MLPs when implemented on FPGA-based systems, particularly in high-dimensional classification tasks. While KANs may offer comparable accuracy for simpler datasets, their resource consumption renders them inefficient for complex, large-scale applications.

To address some of these challenges, Liu et al. [92] proposed KAN 2.0, which introduced the MultKAN model incorporating multiplication nodes to reduce computational complexity and enhance interpretability. While this design helps mitigate training time and improves symbolic clarity, KANs still require longer training durations and consume more resources than conventional architectures due to their reliance on symbolic regression and specialized activation functions. In the context of dynamical systems, Koenig et al. [98] applied KAN-ODEs, demonstrating superior accuracy and interpretability compared to traditional Neural ODEs. However, their implementation requires customized optimization strategies, adding further computational overhead.

Recent developments in rKAN, proposed by Aghaei [65], show promise in improving computational efficiency through the use of rational functions for smoother function approximation. Nevertheless, these benefits come with added design and training complexity, particularly in high-dimensional settings. Schmidt-Hieber [11] revisited the Kolmogorov–Arnold representation theorem and noted that, while KANs theoretically excel in high-dimensional function decomposition, their practical scalability remains constrained due to the increased number of parameters and

complex activation functions, leading to higher resource consumption compared to deep ReLU networks.

Despite these computational limitations, KANs continue to demonstrate strong performance in tasks that prioritize interpretability, compositional modeling, and efficiency. For example, Convolutional KANs achieve accuracy comparable to CNNs on datasets such as MNIST and Fashion-MNIST, while using significantly fewer parameters [29]. Likewise, GKAN models show improved accuracy and symbolic interpretability in node classification and link prediction tasks [56], reinforcing their value in domain-specific applications with constrained computational budgets. These cases highlight that while KANs may incur higher computational overhead than traditional MLPs-especially in high-dimensional settings, they offer significant advantages when modularity, symbolic reasoning, and transparency are prioritized over raw training speed and scalability.

6.2 Generalization and Overfitting

While KANs excel in interpretability and symbolic reasoning, generalization remains a nuanced challenge. Zhang and Zhou [132] established theoretical generalization bounds based on operator norms for KANs using basis-function activations, validating their robustness across simulated and real-world datasets. However, the flexibility that allows KANs to fit complex relationships can also lead to overfitting-especially in small datasets without regularization.

Recent work by Alter et al. [51] has highlighted KANs' resistance to adversarial attacks due to their univariate decomposition, while still cautioning that regularization and pruning are necessary to prevent overfitting. Koenig et al. [98] proposed sparse regularization and symbolic constraints in KAN-ODEs, yielding improved generalization in dynamic systems over traditional neural ODEs.

In graph-based domains, Carlo et al. [56] and Bresson et al. [55] showed that spline activations in GKANs help reduce model complexity and overfitting by enforcing local smoothness and transparency. Similarly, Samadi et al. [15] demonstrated that embedding domain-specific structural knowledge in smooth KANs enables them to generalize from fewer data points, reducing training needs while maintaining accuracy.

To mitigate overfitting and underfitting issues, several techniques have been applied to enhance Kan's robustness. Sparse regularization, entropy regularization, and grid extension techniques are commonly employed to prevent overfitting while improving interpretability and performance on smaller datasets [8, 92]. Liu et al. [92] introduced MultKAN, an augmented version of KANs incorporating multiplicative layers, which not only enhances model capacity but also improves interpretability by uncovering modular structures in the data. Compared to MLPs and CNNs, KANs excel in small-scale tasks due to their spline-based architecture, allowing for precise function fitting without excessive model complexity. Empirical studies demonstrate that KANs often outperform traditional architectures, particularly in low-dimensional settings, graph learning, and scientific discovery, where their ability to model compositional and univariate structures proves advantageous [56].

6.3 Interpretability Limitations

While KANs offer greater interpretability than traditional deep neural networks due to their spline-based and symbolic structure [8, 52], they are still less transparent than inherently interpretable models like decision trees, which provide explicit, rule-based mappings. Recent studies acknowledge that although KANs improve the transparency of deep models by using edge-based univariate functions, they still require auxiliary techniques such as visualization or explainable AI tools to make outputs accessible in complex applications [51, 52]. This balance can lead to confusion, as KANs are often presented as interpretable but in practice, their interpretability is partial and

task-dependent. In many cases, the resulting symbolic functions can still be mathematically dense or unintuitive, limiting their practical interpretability.

To address this, several techniques have been introduced. Xu et al. [60] applied symbolic regression to decode temporal dependencies in T-KAN and MT-KAN, while Galitsky [134] explored logic-based augmentations to improve NLP model explanations. Despite these innovations, the interpretability of KANs still lags behind that of simpler statistical models, particularly in domains requiring strict transparency such as healthcare and finance.

XAI techniques like **SHapley Additive exPlanations** (**SHAP**) [57] and **Local Interpretable Model-agnostic Explanations** (**LIME**) [58] offer complementary insight by decomposing predictions into feature contributions, helping bridge the interpretability gap [55, 56]. In GNN applications, for instance, SHAP has been used to quantify node-level importance in GKAN, enhancing transparency in node classification and graph-level predictions.

While KANs offer built-in interpretability through symbolic regression and spline functions, they fall short of inherently interpretable models like decision trees, which provide explicit feature-rule mappings. This is particularly critical in domains like finance and healthcare, where model decisions must be transparent and subject to audit. Augmenting KANs with SHAP or LIME can bridge this gap, offering localized and global explanations without retraining the model. In high-stakes applications such as physics-based modeling, Koenig et al. [98] showed that embedding symbolic constraints into KAN-ODEs improves clarity, yet the underlying complexity of function compositions still necessitates additional visualization or constraint-based explainability tools. As such, the interpretability benefits of KANs should be viewed as relative and context-dependent rather than universally guaranteed.

7 Emerging Innovations and Hybrid Architectures

7.1 Developments and Emerging Applications

Since their introduction in 2024, KANs have seen rapid evolution through architectural enhancements and hybrid integrations, significantly broadening their applicability across domains. In graph learning, models such as KAGCN and KAGIN have demonstrated the superiority of KAN-based architectures over MLPs, particularly in tasks like graph regression, where improved node feature updates lead to higher accuracy and efficiency [55]. Similarly, in transfer learning, Shen and Younes [135] integrated KANs into ResNet-50 by replacing its linear probing layer, resulting in improved generalization on complex visual data. Extending this idea, the **Residual KAN** (**RKAN**) incorporates Chebyshev polynomial-based convolutions into ResNet and DenseNet backbones, showing enhanced feature extraction with competitive computational efficiency [99].

KAN autoencoders have proven effective for representation learning, achieving competitive reconstruction performance on datasets like MNIST due to their ability to capture fine-grained data dependencies through edge-based activations [31]. While these strengths are notable, KANs also face limitations-particularly their sensitivity to noise. To mitigate this, Shen et al. [54] introduced kernel filtering and oversampling techniques to improve robustness, which is essential for real-world deployment.

In sequential modeling, KANs have shown high predictive accuracy in time series forecasting tasks such as satellite traffic prediction by learning dynamic activation patterns that adapt over time [16]. Beyond traditional machine learning settings, recent works have applied KANs to energy systems. For instance, in electric vehicle battery **state-of-charge** (**SoC**) estimation, KANs outperform both **artificial neural networks** (**ANNs**) and hybrid Barnacles Mating Optimizer-based deep learning models through superior handling of high-dimensional and nonlinear input data [136]. Their edge-based activation architecture allows them to efficiently model voltage, current,

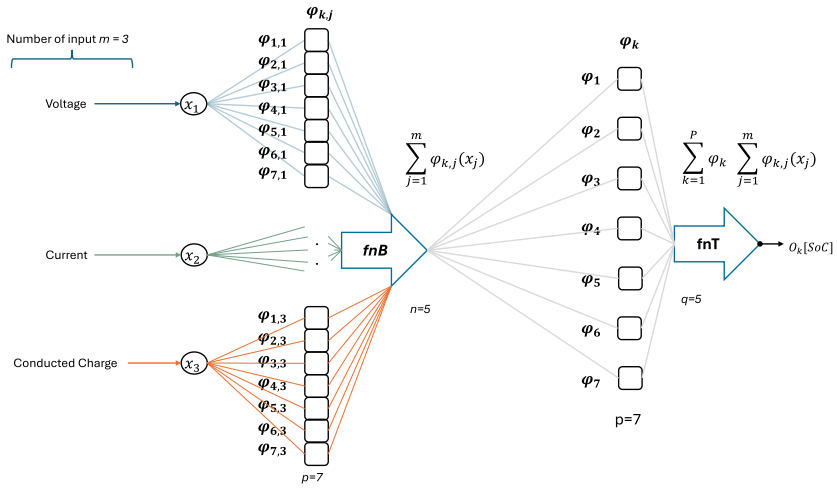

Fig. 7. KAN Architecture for Estimating Battery SoC [136].

and charge dynamics, as illustrated in Figure 7. A related study in commercial building energy forecasting also confirms KANs' strength in capturing complex temporal and physical patterns, achieving better accuracy and efficiency than comparable ANN-based solutions [137].

In the biomedical domain, the **Smooth KAN** (**SKAN**) model proposed by Samadi et al. [15] incorporates structural priors to improve interpretability and reduce the amount of data required for training. This architecture is particularly effective in computational biomedicine, addressing convergence issues while offering symbolic insight into disease modeling and progression [15]. In remote sensing, Jamali et al. [30] proposed a hybrid KAN model combining 1D, 2D, and 3D modules, which outperformed CNNs and vision transformers in hyperspectral image classification by enhancing feature discrimination in low-dimensional spectral data.

In adversarial robustness research, Alter et al. [51] illustrate that KANs have lower Lipschitz constants compared to MLPs, providing greater resistance to perturbations in adversarial conditions. While KANs are known to be sensitive to random noise in raw input data, this robustness against adversarial attacks arises from their edge-based activation structure, which smooths function transitions and limits output volatility under worst-case (structured) perturbations. Their study empirically demonstrates that KANs outperform deep networks in both white-box and black-box adversarial settings. This contrast stems from the difference between stochastic noise-random and unstructured and adversarial perturbations, which are deliberately crafted to mislead models. As a result, KANs may require noise-reduction techniques for noisy environments but still exhibit resilience in adversarial scenarios, making them promising for security-sensitive applications.

7.2 Integration with Other Models

To explore the benefits of integrating KAN with models like GNNs, CNNs, RNNs, and Reinforcement Learning, recent studies have extended KANs structure across these architectures. KANs, grounded in the Kolmogorov–Arnold representation theorem, compose models from simpler, interpretable functions rather than dense weight matrices, promoting transparency. This structure is especially valuable in fields requiring interpretability, including graph data analysis, molecular property prediction, and network classification [56, 111].

Recent studies have explored the combination of KAN with convolutional layers, primarily for applications in image classification and time-series forecasting. For instance, **Convolutional**

KANs (C-KAN) introduce convolutional layers that enhance KAN's feature extraction capability for complex, high-dimensional data such as images and time series, showing improved prediction accuracy and resilience to non-stationarity in data but without extending to GNNs, RNNs, or Reinforcement Learning applications [29, 82]. **KAN with Interactive Convolutional Elements (KANICE)**, a KAN architecture incorporating **Interactive Convolutional Blocks (ICBs)**, also aims at improving CNN adaptability by dynamically adjusting feature extraction across varying input distributions. While highly effective for image classification, this approach has not yet extended to domain-specific tasks like graph analysis or sequential processing [79]. Additionally, Activation Space **Selectable KAN (S-KAN)** introduces selectable activation modes to increase model adaptability across general data-fitting tasks and standard image classification datasets, showing promising results yet without addressing integrations with GNNs or sequential models such as RNNs [77].

In physics-informed contexts, KANs have been adapted to **Physics-Informed Neural Networks (PINNs)**, creating **Kolmogorov–Arnold-Informed Networks (KINNs)** for solving PDEs with greater accuracy than traditional neural networks. This application leverages KAN's spline-based architecture to improve convergence speed and parameter efficiency, beneficial in computational physics tasks, though it remains limited to PDE-focused applications and does not integrate GNN or sequential processing capabilities [38, 138]. Further extending KANs framework, Wang et al. [38] explore its integration within the energy form of PDEs, enabling KANs to enhance traditional PINN architectures without exploring graph data or real-time applications like reinforcement learning.

Studies, such as by Kilani [59], have underscored the versatility of KANs, highlighting successful applications in temporal data analysis and multi-step time series forecasting through hybrid KAN-RNN architectures, albeit on a foundational level. Integrations with advanced reinforcement learning or specialized GNN architectures have not been explored, suggesting the potential for KAN frameworks to enhance model interpretability and parameter efficiency across these domains in future work. In summary, current research indicates that the integration of KAN with other neural network architectures like CNNs has led to parameter-efficient, adaptable models suited to high-dimensional applications in image processing and time-series analysis. However, significant gaps remain in exploring KAN's potential with GNNs, Reinforcement Learning, and RNNs for domain-specific tasks such as graph data analysis and sequential decision-making.

8 Future Directions

Despite their growing prominence and wide applicability, KANs still face several unresolved challenges that limit their broader adoption. In this section, we identify key research gaps and outline future directions grouped thematically across architectural design, optimization strategies, scalability, benchmarking, and interdisciplinary applications.

8.1 Architectural Enhancements and Scalability

While KANs offer symbolic interpretability through spline-based edge functions, their reliance on dense, learnable univariate functions introduces significant computational overhead. This limits their scalability in high-dimensional tasks, such as molecular simulations or multi-channel image processing [52]. Future research must explore architectural innovations that reduce complexity while preserving interpretability. Promising directions include:

— Designing modular KAN blocks with sparsity constraints or low-rank spline approximations.
— Incorporating depth-efficient constructs such as skip connections and residual layers, as proposed in RKAN [99].

— Exploring dynamic KAN structures that adaptively prune or grow activation paths during training to improve scalability without compromising symbolic capacity.

8.2 Optimization and Training Strategies

KANs currently lack stable and efficient training pipelines, particularly when applied to noisy or high-variance datasets. While early variants such as DropKAN [89] and LoTRA [63] offer promising directions, further advancements in robust optimization techniques are required. Future efforts should focus on developing adaptive gradient optimizers that are specifically tailored to spline-based architectures. Additionally, the integration of symbolic regularization methods and entropy-based constraints may enhance both convergence stability and interpretability. Improvements in initialization strategies and the incorporation of gradient clipping schemes are also essential for achieving stable training, especially in deeper KAN architectures.

8.3 Standardization and Benchmarking

A significant barrier to the widespread adoption of KANs lies in the absence of standardized datasets, benchmarking protocols, and open-source toolkits. Currently, many models are evaluated on small, domain-specific datasets, which hinders fair cross-model comparisons. To promote reproducibility and encourage broader usage, the KAN research community should work toward establishing shared benchmarking tasks across application domains such as time series forecasting, symbolic regression, scientific computing, and medical diagnostics. Furthermore, releasing publicly available toolchains with pre-trained models and well-documented hyperparameter search spaces such as those provided by KANICE [79] and ViKANformer [123] can help lower the entry barrier for new researchers. In addition, diagnostic evaluation metrics that jointly assess symbolic fidelity and predictive performance will be vital for measuring progress in this field.

8.4 Interdisciplinary Applications and Generalization

KANs hold substantial promise for domains that prioritize interpretability and domain-specific alignment, such as medicine, physics, and finance. However, their capacity to generalize under conditions of data scarcity or adversarial noise remains insufficiently explored. Emerging models such as BKAN [130], CoxKAN [131], and Fed-KAN [36] have illustrated the potential for KANs in specialized contexts. Building on this foundation, future work should investigate the design of multi-modal KANs capable of handling structured (tabular), sequential (time series), and spatial (image) data simultaneously. Additionally, extending KANs into reinforcement learning and causal inference frameworks could unlock new capabilities, particularly in scenarios where model transparency and decision traceability are essential. Research should also focus on the development of theoretical generalization bounds under variable noise levels and sparse feature settings, especially in applications such as healthcare diagnostics, genomic modeling, and geospatial analytics.

8.5 Open Theoretical Questions

Despite their origins in the Kolmogorov–Arnold representation theorem, the theoretical foundation of KANs particularly in deep and multilayer settings remains only partially understood. Several key open questions persist. These include the need for formal convergence guarantees and generalization bounds under stochastic or non-smooth data regimes. It is also important to analyze the expressivity and sample complexity tradeoffs between KANs and other popular architectures such as MLPs and Transformers. Furthermore, there is limited theoretical understanding of how spline-based activation functions behave in deep compositional hierarchies, and how these dynamics influence learning efficiency and representation capacity.

In summary, KANs represent a promising new direction for symbolic and interpretable learning, but significant progress is still needed to scale them effectively to real-world problems. Advancements in architecture, optimization strategies, benchmarking practices, and theoretical analysis will be critical for unlocking the full potential of KANs in both scientific and industrial AI applications.

9 Conclusions

KANs stand out as a promising and theoretically grounded alternative to conventional neural networks, leveraging the Kolmogorov–Arnold representation theorem to decompose high-dimensional, multivariate functions into simpler, univariate components. This review highlighted KANs' unique strengths, including learnable spline-based activation functions, which allow precise function approximation with fewer parameters, enhancing interpretability and efficiency in applications such as time-series forecasting, graph learning, and physics-informed modeling. Recent advancements, including models like T-KAN and Wav-KAN, have showcased KAN's adaptability to dynamic and spectral data, demonstrating competitive or superior performance in efficiency and predictive power over traditional models.

Despite these strengths, KANs face challenges in scalability, computational complexity, and noise sensitivity in high-dimensional environments. Integrating KAN with other architectures, such as CNNs, RNNs, and GNNs, is explored as a potential avenue to enhance flexibility and address scalability limitations. Such hybrid architectures could allow KANs to inherit the interpretability benefits of their design while leveraging the flexibility and efficiency of other models, making them viable for applications across domains with high data demands.

This systematic review underscores KAN's growing relevance in scientific and applied research, proposing future directions that focus on optimizing scalability, improving noise robustness, and developing efficient training strategies. These advancements could strengthen KAN's role in data-driven applications, contributing to the ongoing development of transparent, efficient neural networks well-suited for complex function approximation in various fields.

Future research could focus on optimizing KAN's computational efficiency, exploring hybrid architectures with other models (e.g., CNNs, RNNs, GNNs), and developing robust benchmarks to facilitate interdisciplinary applications. Addressing these challenges could position KANs as a powerful tool for transparent and scalable neural networks in complex, data-driven domains.

Acknowledgments

We extend our sincere gratitude to Khaled Aly Abousabaa for his assistance in preparing the images for this article.

References

[1] P. N. Srinivasu, N. Sandhya, R. H. Jhaveri, and R. Raut. 2022. From blackbox to explainable AI in healthcare: Existing tools and case studies. *Mobile Information Systems* 2022, 1 (2022), 8167821.

[2] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, and I. Linkov. 2022. An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks. *IEEE Transactions on Intelligent Transportation Systems* 24, 1 (2022), 1000–1014.

[3] S. N. Cohen, D. Snow, and L. Szpruch. 2023. *Black-box Model Risk in Finance*. Cambridge University Press.

[4] Z. Chen, L. Demetrio, S. Gupta, X. Feng, Z. Xia, A. E. Cinà, M. Pintor, L. Oneto, A. Demontis, B. Biggio, et al. 2024. Over-parameterization and adversarial robustness in neural networks: An overview and empirical analysis. *arXiv preprint arXiv:2406.10090*.

[5] F. He and D. Tao. 2020. Recent advances in deep learning theory. *arXiv preprint arXiv:2012.10931*.

[6] A. Shrestha and A. Mahmood. 2019. Review of deep learning algorithms and architectures. *IEEE Access* 7 (2019), 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

[7] A. N. Kolmogorov. 1956. On the representation of continuous functions of several variables by superpositions of continuous functions of fewer variables. *Doklady Akademii Nauk SSSR* 108, 2 (1956), 179–182.

[8] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark. 2024. KAN: Kolmogorov -Arnold networks. *arXiv preprint arXiv:2404.19756*.

[9] P. Kidger, J. Morrill, J. Foster, and T. Lyons. 2020. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems* 33 (2020), 6696–6707.

[10] M. Unser. 1999. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine* 16, 6 (1999), 22–38.

[11] Johannes Schmidt-Hieber. 2021. The kolmogorov–arnold representation theorem revisited. *Neural Networks* 13, 7 (2021), 119–126.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

[13] Y. Peng, Y. Wang, F. Hu, M. He, Z. Mao, X. Huang, and J. Ding. 2024. Predictive modeling of flexible EHD pumps using Kolmogorov -Arnold networks. *Biomimetic Intelligence and Robotics* 4, 4 (2024), 100184.

[14] S. S. Sidharth, K. A. R., A. K. P., et al. 2024. Chebyshev polynomial-based Kolmogorov -Arnold networks: An efficient architecture for nonlinear function approximation. *arXiv preprint arXiv:2405.07200*.

[15] M. E. Samadi, Y. Müller, and A. Schuppert. 2024. Smooth Kolmogorov -Arnold networks enabling structural knowledge representation. *arXiv preprint arXiv:2405.11318*.

[16] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus. 2024. Kolmogorov -Arnold networks (KANs) for time series analysis. *arXiv preprint arXiv:2405.08790*.

[17] Xingyi Yang and Xinchao Wang. 2024. Kolmogorov-arnold transformer. *The Proceedings of the 13th International Conference on Learning Representations*.

[18] Yunhao Wang, Shuai Ma, Dahua Gao, and Guangming Shi. 2024. Swin kansformer-based semantic communication systems for wireless image transmission. In *Proceedings of the 2024 IEEE/CIC International Conference on Communications in China*. pages 265–270.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *Proceedings of the IEEE International Conference on Computer Vision*. pages 1026–1034.

[20] Q. V. Le, N. Jaitly, and G. E. Hinton. 2015. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.

[21] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. pages 249–256.

[22] G. Dudek and T. Rodak. 2025. HKAN: Hierarchical Kolmogorov -Arnold network without backpropagation. *arXiv preprint arXiv:2501.18199*.

[23] Z. Bozorgasl and H. Chen. 2024. Wav-KAN: Wavelet Kolmogorov -Arnold networks. *arXiv preprint arXiv:2405.12832*.

[24] R. Genet and H. Inzirillo. 2024. TKAN: Temporal Kolmogorov -Arnold networks. *arXiv preprint arXiv:2405.07344*.

[25] S. Yang, L. Qin, and X. Yu. 2024. Endowing interpretability for neural cognitive diagnosis by efficient Kolmogorov -Arnold networks. *arXiv preprint arXiv:2405.14399*.

[26] Z. Li. 2024. Kolmogorov -Arnold networks are radial basis function networks. *arXiv preprint arXiv:2405.06721*.

[27] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE* 105, 12 (2017), 2295–2329.

[28] T. Tao, Z. Wang, H. Zhang, T. N. Arvanitis, and L. Zhang. 2025. DiffKAN-Inpainting: KAN-based diffusion model for brain tumor inpainting. In *Proceedings of the 2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*. IEEE, 1–4.

[29] A. D. Bodner, A. S. Tepsich, J. N. Spolski, and S. Pourteau. 2024. Convolutional Kolmogorov -Arnold networks. *arXiv preprint arXiv:2406.13155*.

[30] A. Jamali, S. K. Roy, D. Hong, B. Lu, and P. Ghamisi. 2024. How to learn more? Exploring Kolmogorov -Arnold networks for hyperspectral image classification. *Remote Sensing* 16, 21 (2024), 4015.

[31] M. Moradi, S. Panahi, E. Bollt, and Y.-C. Lai. 2024. Kolmogorov -Arnold network autoencoders. *arXiv preprint arXiv:2410.02077*.

[32] H. Inzirillo and R. Genet. 2024. SigKAN: Signature-weighted Kolmogorov -Arnold networks for time series. *arXiv preprint arXiv:2406.17890*.

[33] A. Pal and S. Nagarajaiah. 2024. KAN/MultKAN with physics-informed spline fitting (KAN-PISF) for ordinary/partial differential equation discovery of nonlinear dynamic systems. *arXiv preprint arXiv:2411.11801*.

[34] X. Guo, T. Dam, R. Dhamdhere, G. Modanwal, and A. Madabhushi. 2025. UNetVL: Enhancing 3D medical image segmentation with Chebyshev KAN powered Vision-LSTM. *arXiv preprint arXiv:2501.07017*.

[35] Z. Ye, X. Niu, X. Wu, W. Yi, Y. Chang, and L. Lu. 2025. Semi-KAN: KAN provides an effective representation for semi-supervised learning in medical image segmentation. *arXiv preprint arXiv:2503.14983*.
[36] E. Zeydan, C. J. Vaca-Rubio, L. Blanco, R. Pereira, M. Caus, and K. Dev. 2025. Fed-KAN: Federated learning with Kolmogorov–Arnold networks for traffic prediction. *arXiv preprint arXiv:2503.00154*.
[37] V. I. Arnold. 1959. On the representation of continuous functions of three variables by superpositions of continuous functions of two variables. *Matematicheskii Sbornik (N.S.)* 48, 90 (1959), 3–74.
[38] Y. Wang, J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, and Y. Liu. 2024. Kolmogorov–Arnold informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov–Arnold networks. *arXiv preprint arXiv:2406.11045*.
[39] Stanislav Selitskiy. 2022. Kolmogorov's gate non-linearity as a step toward much smaller artificial neural networks. In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022)* 1 (2022), 492–499.
[40] Federico Girosi and Tomaso Poggio. 1989. Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation* 1, 4 (1989), 465–469.
[41] Hadrien Montanelli and Haizhao Yang. 2020. Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem. *Neural Networks* 129 (2020), 1–6.
[42] Ken-Ichi Funahashi. 1989. On the approximate realization of continuous mappings by neural networks. *Neural networks* 2, 3 (1989), 183–192.
[43] Yuan Gao, Zehuan Hu, Wei-An Chen, Mingzhe Liu, and Yingjun Ruan. 2025. A revolutionary neural network architecture with interpretability and flexibility based on Kolmogorov–Arnold for solar radiation and temperature forecasting. *Applied Energy* 378 (2025), 124844.
[44] Josh Qixuan Sun. 2024. Evaluating kolmogorov–arnold networks for scientific discovery: A simple yet eEffective approach. *OSF Preprint*, 2024.
[45] T. X. H. Le, T. D. Tran, H. L. Pham, V. T. D. Le, T. H. Vu, Y. Nakashima, and others. 2024. Exploring the limitations of Kolmogorov–Arnold networks in classification: Insights to software training and hardware implementation. In *Proceedings of the 2024 Twelfth International Symposium on Computing and Networking Workshops (CANDARW)*. IEEE, 110–116.
[46] I. Drokin. 2024. Kolmogorov–Arnold convolutions: Design principles and empirical studies. *arXiv preprint arXiv:2407.01092*.
[47] Z. Li. 2024. Kolmogorov–Arnold networks are radial basis function networks. *arXiv preprint arXiv:2405.06721*.
[48] Z. Bozorgasl and H. Chen. 2024. Wav-KAN: Wavelet Kolmogorov–Arnold networks. *arXiv preprint arXiv:2405.12832*.
[49] J. D. Toscano, T. Käufer, Z. Wang, M. Maxey, C. Cierpka, and G. E. Karniadakis. 2024. Inferring turbulent velocity and temperature fields and their statistics from Lagrangian velocity measurements using physics-informed Kolmogorov–Arnold networks. *arXiv preprint arXiv:2407.15727*.
[50] H.-T. Ta, D.-Q. Thai, A. B. S. Rahman, G. Sidorov, and A. Gelbukh. 2024. FC-KAN: Function combinations in Kolmogorov–Arnold networks. *arXiv preprint arXiv:2409.01763*.
[51] T. Alter, R. Lapid, and M. Sipper. 2024. On the robustness of Kolmogorov–Arnold networks: An adversarial perspective. *arXiv preprint arXiv:2408.13809*.
[52] Y. Nagai and M. Okumura. 2024. Kolmogorov–Arnold networks in molecular dynamics. *arXiv preprint arXiv:2407.17774*.
[53] J. Wang, P. Cai, Z. Wang, H. Zhang, and J. Huang. 2024. CEST-KAN: Kolmogorov–Arnold networks for CEST MRI data analysis. *arXiv preprint arXiv:2406.16026*.
[54] H. Shen, C. Zeng, J. Wang, and Q. Wang. 2024. Reduced effectiveness of Kolmogorov–Arnold networks on functions with noise. *arXiv preprint arXiv:2407.14882*.
[55] R. Bresson, G. Nikolentzos, G. Panagopoulos, M. Chatzianastasis, J. Pang, and M. Vazirgiannis. 2024. KAGNNs: Kolmogorov–Arnold networks meet graph learning. *arXiv preprint arXiv:2406.18380*.
[56] G. De Carlo, A. Mastropietro, and A. Anagnostopoulos. 2024. Kolmogorov–Arnold graph neural networks. *arXiv preprint arXiv:2406.18354*.
[57] S. M. Lundberg and S. Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* 30 (2017), 4765–4774.
[58] M. T. Ribeiro, S. Singh, and C. Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.
[59] B. H. Kilani. 2024. Kolmogorov–arnold networks: Key developments and uses. *Qeios*. DOI: 10.32388/7NNCAA
[60] K. Xu, L. Chen, and S. Wang. 2024. Kolmogorov–Arnold networks for time series: Bridging predictive power and interpretability. *arXiv preprint arXiv:2406.02496*.
[61] F. Zhang and X. Zhang. 2024. GraphKAN: Enhancing feature extraction with graph Kolmogorov–Arnold networks. *arXiv preprint arXiv:2406.13597*.

[62] C. Coffman and L. Chen. 2025. MatrixKAN: Parallelized Kolmogorov–Arnold network. *arXiv preprint arXiv:2502.07176*.
[63] Y. Gao, M. K. Ng, and V. Y. F. Tan. 2025. Low tensor-rank adaptation of Kolmogorov–Arnold networks. *arXiv preprint arXiv:2502.06153*.
[64] M. Kiamari, M. Kiamari, and B. Krishnamachari. 2024. GKAN: Graph Kolmogorov–Arnold networks. *arXiv preprint arXiv:2406.06470*.
[65] A. A. Aghaei. 2024. RKAN: Rational Kolmogorov–Arnold networks. *arXiv preprint arXiv:2406.14495*.
[66] K. Shukla, J. D. Toscano, Z. Wang, Z. Zou, and G. E. Karniadakis. 2024. A comprehensive and fair comparison between MLP and KAN representations for differential equations and operator networks. *Computer Methods in Applied Mechanics and Engineering* 431 (2024).
[67] G. Nehma and M. Tiwari. 2024. Leveraging KANs for enhanced deep Koopman operator discovery. *arXiv preprint arXiv:2406.02875*.
[68] R. Genet and H. Inzirillo. 2024. A temporal Kolmogorov-Arnold Transformer for time series forecasting. *arXiv preprint arXiv:2407.20016*.
[69] L. F. Herbozo Contreras, J. Cui, L. Yu, Z. Huang, A. Nikpour, and O. Kavehei. 2025. KAN–EEG: towards replacing backbone–MLP for an effective seizure detection system. *Royal Society Open Science* 12, 3 (2025), 240999. https://doi.org/10.1098/rsos.240999
[70] A. Kundu, A. Sarkar, and A. Sadhu. 2024. KANQAS: Kolmogorov-Arnold network for quantum architecture search. *EPJ Quantum Technology* 11, 1 (2024), 76.
[71] M. Cheon. 2024. Kolmogorov-arnold network for satellite image classification in remote sensing. *arXiv preprint arXiv:2406.00600*.
[72] C. Li, X. Liu, W. Li, C. Wang, H. Liu, Y. Liu, Z. Chen, and Y. Yuan. 2025. U-KAN makes strong backbone for medical image segmentation and generation. In *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 5 (2025), 4652–4660.
[73] J. Xu, Z. Chen, J. Li, S. Yang, W. Wang, X. Hu, and E. C.-H. Ngai. 2024. FourierKAN-GCF: Fourier kolmogorov-arnold network–An effective and efficient feature transformation for graph collaborative filtering. *arXiv preprint arXiv:2406.01034*.
[74] A. A. Howard, B. Jacob, S. H. Murphy, A. Heinlein, and P. Stinis. 2024. Finite basis kolmogorov-arnold networks: domain decomposition for data-driven and physics-informed problems. *arXiv preprint arXiv:2406.19662*.
[75] H.-T. Ta. 2024. BSRBF-KAN: A combination of B-splines and radial basis functions in kolmogorov-arnold networks. In *Proceedings of the International Symposium on Information and Communication Technology*. Springer, 3–15.
[76] Alireza Afzal Aghaei. 2025. fKAN: Fractional kolmogorov–arnold networks with trainable jacobi basis functions. *Neurocomputing* 623 (2025), 129414.
[77] Zhuoqin Yang, Jiansong Zhang, Xiaoling Luo, Zheng Lu, and Linlin Shen. 2024. Activation space selectable kolmogorov–arnold networks. *arXiv preprint arXiv:2408.08338*.
[78] Minjong Cheon. 2024. Demonstrating the efficacy of kolmogorov–arnold networks in vision tasks. *arXiv preprint arXiv:2406.14916*.
[79] Md Meftahul Ferdaus, Mahdi Abdelguerfi, Elias Ioup, David Dobson, Kendall N. Niles, Ken Pathak, and Steven Sloan. 2024. KANICE: Kolmogorov -arnold networks with interactive convolutional elements. In *Proceedings of the 4th International Conference on AI-ML Systems*. 1–10.
[80] Taharim Rahman Anon and Jakaria Islam Emon. 2024. Detecting the undetectable: Combining kolmogorov–arnold networks and MLP for AI-Generated image detection. *arXiv preprint arXiv:2408.09371*.
[81] Jiayi Zhang, Yiming Zhang, Yuan Zheng, Yuchen Wang, Jinjiang You, Yuchen Xu, Wenxing Jiang, and Soumyabrata Dev. 2025. TrafficKAN-GCN: Graph convolutional-based kolmogorov–arnold network for traffic flow optimization. *arXiv preprint arXiv:2503.03276*.
[82] Ioannis E. Livieris. 2024. C-kan: A new approach for integrating convolutional layers with kolmogorov–arnold networks for time-series forecasting. *Mathematics* 12, 19 (2024), 3022.
[83] Décio Alves, Fábio Mendonça, Sheikh Shanawaz Mostafa, and Fernando Morgado-Dias. 2024. On the use of kolmogorov–arnold networks for adapting wind numerical weather forecasts with explainability and interpretability: Application to madeira international airport. *Environmental Research Communications* 6, 10 (2024), 105008.
[84] Chang Dong, Liangwei Zheng, and Weitong Chen. 2024. Kolmogorov–arnold networks (KAN) for time series classification and robust analysis. In *Proceedings of the International Conference on Advanced Data Mining and Applications*. Springer, 342–355.
[85] János Hollósi, Áron Ballagi, Gábor Kovács, Szabolcs Fischer, and Viktor Nagy. 2024. Detection of bus driver mobile phone usage using kolmogorov-arnold networks. *Computers* 13, 9 (2024), 218.
[86] Xinhe Li, Zhuoying Feng, Yezeng Chen, Weichen Dai, Zixu He, Yi Zhou, and Shuhong Jiao. 2024. Coeff-KANs: A paradigm to address the electrolyte field with KANs. *arXiv preprint arXiv:2407.20265*.

[87] Shrenik Zinage, Sudeepta Mondal, and Soumalya Sarkar. 2024. DKL-KAN: Scalable deep kernel learning using kolmogorov–arnold networks. *arXiv preprint arXiv:2407.21176*.
[88] Pawel Pratyush, Callen Carrier, Suresh Pokharel, Hamid D. Ismail, Meenal Chaudhari, and Dukka B. KC. 2024. Calmphoskan: Prediction of general phosphorylation sites in proteins via fusion of codon aware embeddings with amino acid aware embeddings and wavelet-based kolmogorov arnold network. *bioRxiv* (2024), 2024–07.
[89] Mohammed Ghaith Altarabichi. 2024. DropKAN: Regularizing KANs by masking post-activations. *arXiv preprint arXiv:2407.13044*.
[90] Spyros Rigas, Michalis Papachristou, Theofilos Papadopoulos, Fotios Anagnostopoulos, and Georgios Alexandridis. 2024. Adaptive training of grid-dependent physics-informed kolmogorov–arnold networks. *IEEE Access*. IEEE.
[91] Engin Zeydan, Cristian J. Vaca-Rubio, Luis Blanco, Roberto Pereira, Marius Caus, and Abdullah Aydeger. 2024. F-KANs: federated kolmogorov–arnold networks. *arXiv preprint arXiv:2407.20100*.
[92] Z. Liu, P. Ma, Y. Wang, W.Matusik, and M. Tegmark. 2024. KAN 2.0: Kolmogorov-arnold networks meet science. *arXiv preprint arXiv:2408.10205*.
[93] Jeffrey L. Elman. Finding structure in time. *Cognitive Science* 14, 2 (1990), 179–211.
[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* 60, 6 (June 2017), 84–90. https://doi.org/10.1145/3065386
[95] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[96] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2, 4 (1989), 303–314.
[97] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. 315–323.
[98] Benjamin C. Koenig, Suyong Kim, and Sili Deng. 2024. KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics. *Computer Methods in Applied Mechanics and Engineering* 432 (2024), 117397.
[99] Ray Congrui Yu, Sherry Wu, and Jiang Gui. 2024. Residual kolmogorov–arnold network for enhanced deep learning. *arXiv preprint arXiv:2410.05500*.
[100] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[101] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
[102] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. MLP-Mixer: An all-MLP architecture for vision. *Advances in Neural Information Processing Systems* 34 (2021), 24261–24272.
[103] Mengxi Liu, Sizhen Bian, Bo Zhou, and Paul Lukowicz. 2024. ikan: Global incremental learning with kan for human activity recognition across heterogeneous datasets. In *Proceedings of the 2024 ACM International Symposium on Wearable Computers*. 89–95.
[104] Y. Chen, Z. Zhu, S. Zhu, L. Qiu, B. Zou, F. Jia, Y. Zhu, C. Zhang, Z. Fang, F. Qin, J. Fan, C. Wang, Y. Gao, and G. Yu. 2024. SCKansformer: Fine-grained classification of bone marrow cells via kansformer backbone and hierarchical attention mechanisms. *IEEE Journal of Biomedical and Health Informatics* 29 (2024), 558–571. DOI: 10.1109/JBHI.2024.3471928
[105] Zhijie Chen and Xinglin Zhang. 2024. Larctan-SKAN: Simple and efficient single-parameterized kolmogorov–arnold networks using learnable trigonometric function. *arXiv preprint arXiv:2410.19360*.
[106] Xinke Xie, Yang Lu, Chong-Yung Chi, Wei Chen, Bo Ai, and Dusit Niyato. 2024. Kansformer for scalable beamforming. *arXiv preprint arXiv:2410.20690*.
[107] Pei Liu, Nanfang Zheng, Yiqun Li, Junlan Chen, and Ziyuan Pu. 2024. KANInfraDet3D: A road-side camera-LiDAR fusion 3D perception model based on nonlinear feature extraction and intrinsic correlation. *arXiv preprint arXiv:2410.15814*.
[108] Zhijie Chen and Xinglin Zhang. 2024. LSS-SKAN: Efficient kolmogorov-arnold networks based on single-parameterized function. *arXiv preprint arXiv:2410.14951*.
[109] SangJong Lee, Jin-Kwang Kim, JunHo Kim, TaeHan Kim, and James Lee. 2024. HIPPO-KAN: Efficient KAN model for time series analysis. *arXiv preprint arXiv:2410.14939*.
[110] Amanda A. Howard, Bruno Jacob, and Panos Stinis. 2024. Multifidelity kolmogorov-arnold networks. *arXiv preprint arXiv:2410.14764*.
[111] Longlong Li, Yipeng Zhang, Guanghui Wang, and Kelin Xia. 2024. KA-GNN: Kolmogorov-arnold graph neural networks for molecular property prediction. *arXiv preprint arXiv:2410.11323*.

[112] Farinaz Mostajeran and Salah A. Faroughi. 2024. EPI-CKANs: Elasto-plasticity informed kolmogorov-arnold networks using chebyshev polynomials. *arXiv preprint arXiv:2410.10897*.
[113] Ali Kashefi. 2024. PointNet with KAN versus PointNet with MLP for 3D classification and segmentation of point sets. *arXiv preprint arXiv:2410.10084*.
[114] Kunpeng Xu, Lifei Chen, and Shengrui Wang. 2024. Are KAN effective for identifying and tracking concept drift in time series? *arXiv preprint arXiv:2410.10041*.
[115] Md Abrar Jahin, Md Akmol Masud, MF Mridha, Zeyar Aung, and Nilanjan Dey. 2024. KACQ-DCNN: Uncertainty-aware interpretable Kolmogorov-Arnold classical-quantum dual-channel neural network for heart disease detection. *arXiv preprint arXiv:2410.07446*.
[116] Petr Ivashkov, Po-Wei Huang, Kelvin Koor, Lirandë Pira, and Patrick Rebentrost. 2024. QKAN: Quantum kolmogorov-arnold networks. *arXiv preprint arXiv:2410.04435*.
[117] Andreï Nikolaevich Kolmogorov. 1961. *On the Representation of Continuous Functions of Several Variables by Superpositions of Continuous Functions of a Smaller Number of Variables*. American Mathematical Society.
[118] Vladimir I. Arnold. 2009. On the representation of functions of several variables as a superposition of functions of a smaller number of variables. In *Collected Works: Representations of Functions, Celestial Mechanics and KAM Theory*. 1957–1965.
[119] Yuntian Hou, Tianrui Ji, and Di Zhang. 2024. A comprehensive survey on kolmogorov arnold networks (KAN). *arXiv preprint arXiv:2407.11075*.
[120] Seyd Teymoor Seydi, Zavareh Bozorgasl, and Hao Chen. 2024. Unveiling the power of wavelets: A wavelet-based kolmogorov-arnold network for hyperspectral image classification. *arXiv preprint arXiv:2406.07869*.
[121] Yanlin Wu, Tao Li, Zhihong Wang, Hong Kang, and Along He. 2024. TransUKAN: Computing-efficient hybrid KAN-transformer for enhanced medical image segmentation. *arXiv preprint arXiv:2409.14676*.
[122] Shiyi Chen, Chunlong He, and Xingquan Li. 2024. Kansformer: An efficient network for CSI feedback in FDD massive MIMO system. Available at SSRN 5168783.
[123] S. Shreyas and M. Akshath. 2025. ViKANformer: Embedding kolmogorov arnold networks in vision transformers for pattern-based learning. *arXiv preprint arXiv:2503.01124*.
[124] T. N. Kipf and M. Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
[125] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1, 4 (1989), 541–551.
[126] John J. Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79, 8 (1982), 2554–2558.
[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)* 30 (2017), 5998–6008.
[128] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.
[129] L. F. Guilhoto and P. Perdikaris. 2024. Deep learning alternatives of the Kolmogorov superposition theorem. *arXiv preprint arXiv:2410.01990*.
[130] M. M. Hassan. 2024. Bayesian Kolmogorov Arnold Networks (Bayesian_KANs): A probabilistic approach to enhance accuracy and interpretability. *arXiv preprint arXiv:2408.02706*.
[131] W. Knottenbelt, Z. Gao, R. Wray, W. Z. Zhang, J. Liu, and M. Crispin-Ortuzar. 2024. CoxKAN: Kolmogorov-arnold networks for interpretable, high-performance survival analysis. *arXiv preprint arXiv:2409.04290*.
[132] X. Zhang and H. Zhou. 2024. Generalization bounds and model complexity for Kolmogorov-Arnold Networks. *arXiv preprint arXiv:2410.08026*.
[133] H. Shuai and F. Li. 2025. Physics-informed kolmogorov-arnold networks for power system dynamics. *IEEE Open Access Journal of Power and Energy*.
[134] Boris A. Galitsky. 2024. Kolmogorov-arnold network for word-level explainable meaning representation.
[135] S. Shen and R. Younes. 2024. Reimagining linear probing: Kolmogorov-arnold networks in transfer learning. *arXiv preprint arXiv:2409.07763*.
[136] M. H. Sulaiman, Z. Mustaffa, A. I. Mohamed, A. S. Samsudin, and M. I. M. Rashid. 2024. Battery state of charge estimation for electric vehicle using kolmogorov-arnold networks. *Energy* 311 (2024), 133417. https://doi.org/10.1016/j.energy.2024.133417
[137] Mohd Herwan Sulaiman, Zuriani Mustaffa, Muhammad Salihin Saealal, Mohd Mawardi Saari, and Abu Zaharin Ahmad. 2024. Utilizing the Kolmogorov–Arnold Networks for chiller energy consumption prediction in commercial building. *Journal of Building Engineering* 96 (2024), 110475.

[138] T. Yu, J. Qiu, J. Yang, and I. Oseledets. 2024. Sinc kolmogorov-arnold network and Iits applications on physics-informed neural networks. *arXiv preprint arXiv:2410.04096*.
[139] M. Poluektov and A. Polar. 2023. Construction of the kolmogorov-arnold representation using the newton-kaczmarz method. *arXiv preprint arXiv:2305.08194*.

Received 16 November 2024; revised 8 April 2025; accepted 31 May 2025