

Article

Mathematical Generalization of Kolmogorov-Arnold Networks (KAN) and Their Variants

Fray L. Becerra-Suarez ^{1,*}, Ana G. Borrero-Ramírez ^{2,†}, Edwin Valencia-Castillo ³ and Manuel G. Forero ^{2,*}

¹ Grupo de Investigación en Inteligencia Artificial (UMA-AI), Facultad de Ingeniería y Negocios, Universidad Privada Norbert Wiener, Lima 15046, Peru

² Semillero Lún, Universidad de Ibagué, Ibagué 730001, Colombia; 2420201080@estudiantesunibague.edu.co

³ Departamento de Sistemas, Estadística e Informática, Universidad Nacional de Cajamarca, Cajamarca 06001, Peru; evalencia@unc.edu.pe

* Correspondence: fray.becerra@uwiener.edu.pe (F.L.B.-S.); manuel.forero@unibague.edu.co (M.G.F.)

† These authors contributed equally to this work.

Abstract

Neural networks have become a fundamental tool for solving complex problems, from image processing and speech recognition to time series prediction and large-scale data classification. However, traditional neural architectures suffer from interpretability problems due to their opaque representations and lack of explicit interaction between linear and nonlinear transformations. To address these limitations, Kolmogorov–Arnold Networks (KAN) have emerged as a mathematically grounded approach capable of efficiently representing complex nonlinear functions. Based on the principles established by Kolmogorov and Arnold, KAN offer an alternative to traditional architectures, mitigating issues such as overfitting and lack of interpretability. Despite their solid theoretical basis, practical implementations of KAN face challenges, such as optimal function selection and computational efficiency. This paper provides a systematic review that goes beyond previous surveys by consolidating the diverse structural variants of KAN (e.g., Wavelet-KAN, Rational-KAN, MonoKAN, Physics-KAN, Linear Spline KAN, and Orthogonal Polynomial KAN) into a unified framework. In addition, we emphasize their mathematical foundations, compare their advantages and limitations, and discuss their applicability across domains. From this review, three main conclusions can be drawn: (i) spline-based KAN remain the most widely used due to their stability and simplicity, (ii) rational and wavelet-based variants provide greater expressivity but introduce numerical challenges, and (iii) emerging approaches such as Physics-KAN and automatic basis selection open promising directions for scalability and interpretability. These insights provide a benchmark for future research and practical implementations of KAN.

Keywords: Kolmogorov-Arnold Networks; neural networks; interpretability; machine learning; deep learning; function approximation; computational efficiency; nonlinear functions

MSC: 68T07



Academic Editor: Matteo Cavaliere

Received: 22 August 2025

Revised: 24 September 2025

Accepted: 27 September 2025

Published: 30 September 2025

Citation: Becerra-Suarez, F.L.;

Borrero-Ramírez, A.G.;

Valencia-Castillo, E.; Forero, M.G.

Mathematical Generalization of Kolmogorov-Arnold Networks (KAN) and Their Variants. *Mathematics* **2025**, *13*, 3128. <https://doi.org/10.3390/math13193128>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent decades, neural networks have become a fundamental tool for solving complex problems, ranging from image processing and speech recognition to time series forecasting and large-scale data classification. Their ability to learn hierarchical representations and extract features from unstructured data has positioned neural networks at the

core of applications in computer vision, natural language processing, and applied artificial intelligence [1–5]. As architectures and algorithms continue to evolve, the applicability of neural networks has expanded significantly across multiple domains.

In their standard form, neural networks employ predefined activation functions (e.g., ReLU, sigmoid) in different layers [6]. While these nonlinear functions allow networks to model complex relationships, the resulting architectures often suffer from interpretability issues due to the opaque nature of their learned representations. In particular, linear and nonlinear transformations are treated independently, without explicit interactions, making the decision process difficult to trace and explain [7]. This limitation has motivated the development of alternative architectures that balance predictive performance with interpretability.

One such alternative is the family of Kolmogorov–Arnold Networks (KAN). Rather than being a single new architecture, KAN are inspired by the celebrated Kolmogorov–Arnold representation theorem, which states that any multivariate continuous function can be represented as a finite composition of univariate continuous functions. This theoretical result provides a mathematically grounded framework for constructing neural networks where nonlinearity is distributed along edges through functional activations, instead of being concentrated in fixed activation nodes [8–10]. Such a formulation has been explored in diverse contexts, from solving differential equations [11] and fraud detection [12,13] to data analysis, forecasting [14–17], and real-time autonomous control systems.

Given the growing interest and rapid development of KAN-based models, there is now a fragmented body of literature that introduces a variety of structural variants, each with its own mathematical formulation, strengths, and limitations. To the best of our knowledge, no systematic review has yet consolidated these contributions. Therefore, the purpose of this article is not to propose a novel network, but to provide a comprehensive review that unifies the main structural variants of KAN, including Wavelet-KAN, Rational-KAN, MonoKAN, Physics-KAN, Linear Spline KAN, and Orthogonal Polynomial KAN. Throughout the paper, we adopt a consistent mathematical notation: input vectors are denoted as $\mathbf{x} \in \mathbb{R}^d$, intermediate features as \mathbf{h} , edge functions as $f(\cdot)$, and outputs as $y \in \mathbb{R}^m$. This notation will be used to ensure clarity and uniformity in all subsequent sections.

In this context, the research aligns with the Sustainable Development Goals (SDG). It contributes to SDG 9 (Industry, Innovation and Infrastructure) by providing a rigorous, standardized synthesis of KAN variants that supports interoperable and reproducible digital infrastructures. In this way, the study not only organizes and clarifies the state of the art on KAN with unified notation and comparable criteria, but also narrows the implementation gap by proposing methodological guidelines for AI systems that are accurate, interpretable, and energy-aware.

The remainder of the paper is organized as follows: Section 2 introduces the mathematical background of the Kolmogorov–Arnold theorem and its neural interpretations. Section 3 reviews the main KAN variants in detail, highlighting their mathematical structures, advantages, and limitations. Sections 4–15 then develop, in progressive order, families and extensions based on splines, polynomials (Chebyshev, Gram, and other orthogonal ones), wavelets, monotonicity and physical constraints, rational functions, and piecewise linear activations, addressing their formulation, advantages, limitations, and applications, and including a comprehensive comparison. The Section 16 provides a comparative discussion of these approaches, and Section 17 concludes with perspectives and open challenges for future research.

2. Methodology and Model Architecture

The following section provides a description of the Kolmogorov–Arnold Network (KAN) architecture, intended to help the reader understand how the theoretical Kolmogorov–Arnold representation theorem is implemented in practice. This construction is typically realized using spline basis functions to approximate both the inner and outer univariate functions that appear in the decomposition [8–11]. The mathematical structure and its key components are described below.

2.1. Inner Functions as Spline Expansions

The inner univariate functions $\psi_{q,p}(x_p)$, which act on each coordinate x_p of the input vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, are parameterized by *univariate spline expansions*. Each inner function is expressed as a linear combination of local spline basis functions:

$$\psi_{q,p}(x_p) = \sum_{k=1}^K a_{q,p,k} B_{q,p,k}(x_p), \quad (1)$$

where:

- $B_{q,p,k}(x_p)$ are spline basis functions (commonly cubic B-splines) that guarantee local smoothness and piecewise continuity [12].
- $a_{q,p,k}$ are trainable coefficients controlling the contribution of each basis.
- K denotes the number of knots (partition points of the domain of x_p), which define local regions where different polynomial pieces apply.

Purpose: inner splines capture nonlinear relationships of each scalar input x_p with localized smooth adaptations, mitigating the global oscillations typical of high-order polynomial approximations [13].

2.2. Aggregation of Inner Functions

The outputs of the inner functions are aggregated linearly to form intermediate latent features z_q :

$$z_q = \sum_{p=1}^n \psi_{q,p}(x_p), \quad q = 0, 1, \dots, 2n, \quad (2)$$

where the index q ranges up to $2n$ in accordance with the Kolmogorov–Arnold theorem, which requires $2n + 1$ outer univariate functions for the representation of any n -variate continuous function [14]. These latent features z_q encode nonlinear interactions across the n input coordinates.

2.3. Outer Functions as Splines

The outer univariate functions $\Phi_q(z_q)$ transform the intermediate features into outputs. They are also represented via spline expansions:

$$\Phi_q(z_q) = \sum_{m=1}^M b_{q,m} C_{q,m}(z_q), \quad (3)$$

where:

- $C_{q,m}(z_q)$ are spline basis functions ensuring continuity and differentiability [15].
- $b_{q,m}$ are trainable coefficients.
- M is the number of basis functions selected for each outer spline.

Purpose: these functions combine nonlinear contributions from intermediate features z_q , yielding the hierarchical composition required by the Kolmogorov–Arnold decomposition [16,17].

2.4. Final Model Output

The prediction of the KAN is obtained by summing all transformed features:

$$f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q(z_q). \quad (4)$$

This output construction preserves consistency with the Kolmogorov–Arnold representation theorem while combining both localized inner contributions and global outer transformations.

2.5. Trainable Parameters and Optimization

The parameter set of the KAN consists of:

- Inner spline coefficients $a_{q,p,k}$: approximately $n \cdot (2n + 1) \cdot K$ parameters, accounting for n inputs, $2n + 1$ intermediate sums, and K spline basis functions per coordinate.
- Outer spline coefficients $b_{q,m}$: $(2n + 1) \cdot M$ parameters, with M basis functions per outer spline.

Optimization is typically performed via gradient-based algorithms (e.g., Adam), minimizing a loss such as the mean squared error over a dataset [18,19]. The inherent smoothness of splines acts as a form of implicit regularization. Explicit regularization techniques such as curvature penalization or L_2 -norm regularization can be applied to further prevent oscillations or overfitting [20,21]. Input normalization is recommended to stabilize training, since splines are defined over compact intervals [22].

3. Mathematical Foundations of Interpolation Methods

Understanding the theoretical underpinnings of Kolmogorov–Arnold Networks (KAN) requires a solid foundation in function approximation theory, particularly the Kolmogorov–Arnold representation theorem and its implications for neural architecture design. This section outlines the mathematical principles that justify the structure of KAN, including univariate function decomposition, spline-based function construction, and the transition from traditional neural activations to more expressive basis function models. By grounding KAN in formal mathematical constructs, we establish both the theoretical motivation and the functional versatility of this class of network

3.1. Lagrange Polynomials

The Lagrange polynomial interpolates n points (x_i, y_i) through a linear combination of basis polynomials:

$$P(x) = \sum_{i=1}^n y_i \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}. \quad (5)$$

Each term $\prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$ is a polynomial that equals 1 at $x = x_i$ and 0 at other nodes. This ensures $P(x_i) = y_i$. However, for large n , the Runge phenomenon causes extreme oscillations [23].

3.2. Bézier Curves

A Bézier curve of degree m is defined using Bernstein polynomials:

$$B(t) = \sum_{i=0}^m P_i \binom{m}{i} t^i (1-t)^{m-i}, \quad t \in [0, 1]. \quad (6)$$

The curve passes only through P_0 and P_m , while intermediate points (P_1, \dots, P_{m-1}) act as “shape controllers”. The derivative at the endpoints depends on adjacent points:

$$B'(0) = m(P_1 - P_0), \quad B'(1) = m(P_m - P_{m-1}).$$

Bézier curves are widely used in computer graphics and geometric modeling due to their intuitive control and smoothness properties [24].

3.3. Piecewise Linear Interpolation

Within each interval $[x_i, x_{i+1}]$, the function is a straight line:

$$f(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i). \tag{7}$$

The function is continuous (C^0) but not differentiable at nodes. The maximum error (for $f \in C^2$) is bounded by:

$$\max_{x \in [a,b]} |f(x) - S(x)| \leq \frac{h^2}{8} \max_{x \in [a,b]} |f''(x)|,$$

where h is the maximum interval size. This method is simple and fast, although limited in smoothness [25,26].

3.4. Cubic Splines

Each segment of the cubic spline $S(x)$ in $[x_i, x_{i+1}]$ is a third-degree polynomial:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \tag{8}$$

Continuity conditions C^2 (function, first, and second derivatives) are enforced at nodes. For a *natural spline*, the tridiagonal system is solved:

$$\begin{cases} 2c_1 + c_2 = \frac{3}{h_1}(y_2 - y_1), \\ c_{i-1} + 4c_i + c_{i+1} = \frac{3}{h_i}(y_{i+1} - y_i) - \frac{3}{h_{i-1}}(y_i - y_{i-1}), \\ c_{n-1} + 2c_n = 0. \end{cases} \tag{9}$$

Cubic splines are the foundation for many KAN variants due to their smoothness and local control properties [27–30].

3.5. Hermite Splines

Interpolates values y_i and derivatives d_i at each node:

$$H_i(x) = y_i + d_i(x - x_i) + \left(3\frac{y_{i+1} - y_i}{h_i^2} - \frac{d_{i+1} + 2d_i}{h_i}\right)(x - x_i)^2 + \left(\frac{d_{i+1} + d_i}{h_i^2} - 2\frac{y_{i+1} - y_i}{h_i^3}\right)(x - x_i)^3. \tag{10}$$

Continuity C^1 and explicit control of slopes at nodes are guaranteed, ideal for applications requiring known derivatives (e.g., trajectories with specified velocities) [30]. In order to contextualize the choice of functional bases used within Kolmogorov–Arnold Networks, it is useful to briefly review classical interpolation techniques. Table 1 summarizes the main methods, highlighting their advantages and disadvantages. This comparative overview provides the mathematical background against which spline- and polynomial-based variants of KANs can be better understood.

Table 1. Comparison of Interpolation Methods.

Method	Advantage	Disadvantage
Lagrange	Exact at points	Instability for large n
Bézier	Intuitive shape control	Does not interpolate all points
Linear	Simplicity	Lack of smoothness (C^0)
Cubic Splines	C^2 smoothness	High computational cost
Hermite	Derivative control	Requires extra data

4. Advantages of Splines in KAN

Splines, when implemented in Kolmogorov–Arnold Networks (KAN), provide mathematical and computational advantages that enhance model performance and stability. This section highlights their properties, followed by a detailed example of cubic B-splines [31–33].

4.1. Local Adaptability

Spline basis functions possess *compact support*, meaning they are non-zero only over a finite subinterval of the domain. As a result, splines can adjust their shape in specific regions without affecting distant areas. In a physiological signal modeling task, for example, a spline can adapt locally to a cardiac peak at $x \in [t_k, t_{k+4})$ while preserving stability in normal zones. Formally,

$$\psi_{q,p}(x_p) = \sum_{k=1}^K a_{q,p,k} B_{q,p,k}(x_p),$$

where $B_{q,p,k}(x_p)$ is non-zero only in the neighborhood of the knot t_k [34,35]. This property prevents overfitting at a global scale, since changes are localized.

4.2. Guaranteed Smoothness

Cubic splines ensure C^2 continuity, i.e., continuity of the function and its first two derivatives. This property is crucial to avoid physically unrealistic discontinuities and to guarantee smooth gradients in the network output. For instance, in fluid dynamics simulations one requires

$$\frac{\partial^2 S(x)}{\partial x^2} \text{ continuous on } [x_1, x_n],$$

which minimizes the curvature energy functional

$$E_{\text{curv}} = \int_{x_1}^{x_n} [S''(x)]^2 dx,$$

a standard regularization principle in physics and approximation theory [34,35].

4.3. Computational Efficiency

Efficient recursive algorithms such as De Boor’s algorithm allow evaluation of B-splines with complexity $\mathcal{O}(k)$ per evaluation point. For cubic splines ($k = 3$),

$$\text{Total cost} \propto N \cdot k, \quad N = \text{number of evaluation points.}$$

This is significantly more efficient than global interpolation methods such as Lagrange polynomials, which typically require $\mathcal{O}(n^2)$ operations [36,37].

4.4. Interpretability of Coefficients

The coefficients in spline expansions carry a direct geometric interpretation. For example, for an outer spline

$$\Phi_q(z_q) = \sum_{m=1}^M b_{q,m} C_{q,m}(z_q),$$

a large magnitude of $|b_{q,m}|$ indicates a dominant influence of the basis function $C_{q,m}(z_q)$ on the output. This property enables the inspection of fitted models: coefficients can highlight overfitting, emphasize relevant regions of the input space, or identify dominant features [38].

4.5. Detailed Example: Cubic B-Splines

A cubic B-spline basis function is defined piecewise over four consecutive knot intervals. Let $\{t_k\}$ be a non-decreasing sequence of knots, and $h = t_{k+1} - t_k$ the uniform knot spacing. Then

$$B_k(x) = \begin{cases} \frac{(x - t_k)^3}{6h^3}, & x \in [t_k, t_{k+1}), \\ \frac{-3(x - t_{k+1})^3 + 3h(x - t_{k+1})^2 + 3h^2(x - t_{k+1}) + h^3}{6h^3}, & x \in [t_{k+1}, t_{k+2}), \\ \frac{3(x - t_{k+3})^3 - 9h(x - t_{k+3})^2 + 9h^2(x - t_{k+3})}{6h^3}, & x \in [t_{k+2}, t_{k+3}), \\ \frac{(t_{k+4} - x)^3}{6h^3}, & x \in [t_{k+3}, t_{k+4}), \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Key properties:

- **Local support:** $B_k(x)$ is non-zero only on $[t_k, t_{k+4})$.
- **Normalization:** $\int_{-\infty}^{\infty} B_k(x) dx = 1$.
- **Smoothness:** $B_k(x)$ belongs to \mathcal{C}^2 , ensuring continuity of $S(x)$, $S'(x)$, and $S''(x)$.

Numerical example: For uniform knots $t = \{0, 1, 2, 3, 4\}$ ($h = 1$), evaluating $B_1(x)$ at $x = 1.5$ gives

$$B_1(1.5) = \frac{-3(0.5)^3 + 3(0.5)^2 + 3(0.5) + 1}{6} = \frac{-0.375 + 0.75 + 1.5 + 1}{6} \approx 0.479.$$

As shown in Figure 1, illustrates both the locality and the normalization property of the cubic B-spline.

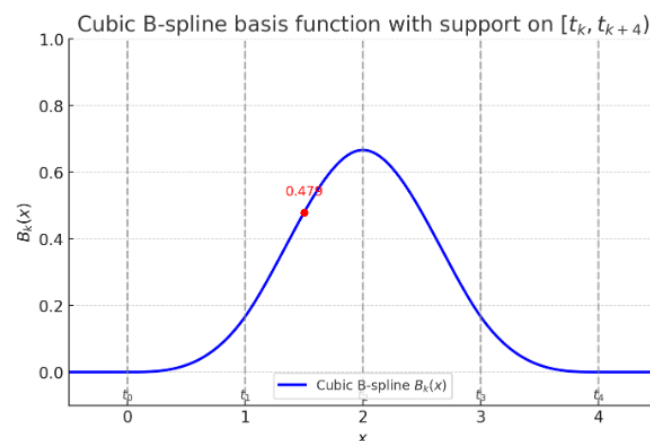


Figure 1. Cubic B-spline basis function $B_k(x)$ with support on four consecutive intervals $[t_k, t_{k+4})$.

5. New Architectures and Applications of Kolmogorov-Arnold Networks (KAN)

Kolmogorov-Arnold Networks (KAN) represent a significant evolution in the design of machine learning architectures, combining the classical Kolmogorov-Arnold representation theorem with modern techniques of interpolation through adaptive splines [37–39]. These networks stand out for their ability to process dynamic and sequential data, such as financial time series or biomedical signals [37,38,40], thanks to a structure that integrates internal ($\psi_{q,p}$) and external (Φ_q) functions in hierarchical layers. In finance, KAN are applied to market prediction, modeling nonlinearities in high-frequency data, and to option pricing, outperforming traditional methods like Black-Scholes in complex scenarios [32]. In robotics, their implementation in reinforcement learning enables autonomous control of dynamic systems, optimizing real-time decision-making [33,34], while in healthcare, they facilitate treatment personalization through the analysis of individual biological responses [35,36]. Among their key advantages are local adaptability, which adjusts parameters in specific regions without affecting the global model, and the interpretability of coefficients, which reveal contributions from critical variables [36,37]. However, they face challenges such as robustness against noise in volatile financial environments and latency in embedded hardware for real-time robotic applications [38]. Although promising, their industrial adoption requires advances in generalization through techniques like transfer learning [39], thereby solidifying their potential to revolutionize fields demanding mathematical precision and practical adaptability [41–43].

6. Classification of KAN Variants: Replacement of Splines with Chebyshev Polynomials

A key variant of Kolmogorov–Arnold Networks (KAN) replaces spline-based activation functions with Chebyshev polynomials, modifying both their mathematical structure and functional behavior [44,45]. In standard KAN, the inner functions $\psi_{q,p}(x_p)$ are defined using spline bases:

$$\psi_{q,p}(x_p) = \sum_{k=1}^K a_{q,p,k} B_{q,p,k}(x_p),$$

where $B_{q,p,k}(x_p)$ are local basis functions supported on intervals defined by knots. In the Chebyshev variant, splines are replaced with global orthogonal polynomials:

$$\psi_{q,p}(x) = \sum_{k=0}^K a_k T_k(x), \quad T_k(x) = \cos(k \arccos(x)), \quad x \in [-1, 1]. \tag{12}$$

Chebyshev polynomials $T_k(x)$ minimize the maximum error in the L_∞ norm, achieving near-optimal polynomial approximation with exponential convergence for smooth functions [44,46]. However, since $\arccos(x)$ is undefined outside $[-1, 1]$, direct use is unstable in unbounded domains.

To overcome this, a nonlinear mapping via the hyperbolic tangent compresses the domain:

$$T_k^{\text{mod}}(x) = \cos(k \cdot \frac{\pi}{2} \tanh(x)), \tag{13}$$

which ensures inputs $x \in \mathbb{R}$ are mapped into $(-1, 1)$, preserving stability [45]. For instance, for $x = 2$,

$$\tanh(2) \approx 0.96 \Rightarrow T_3^{\text{mod}}(2) = \cos(3 \cdot \frac{\pi}{2} \cdot 0.96) \approx -0.12,$$

whereas $T_3(2) = \cos(3 \arccos 2)$ would be undefined.

6.1. Structural Impact on KAN

Switching from local splines to global Chebyshev polynomials produces important architectural changes. Local adaptability is sacrificed, but extrapolation capabilities are enhanced [44]. Moreover, the model is simplified by removing explicit knot dependence, reducing the parameter count to the set of coefficients $\{a_k\}$.

6.2. Advantages and Disadvantages

The Chebyshev-based formulation provides several practical advantages. It offers numerical stability on bounded domains, ensures smooth extrapolation beyond the training interval, and typically requires fewer parameters compared to spline-based KAN. Furthermore, the reduced flexibility of global polynomials can help mitigate overfitting, particularly in noisy datasets [44,45]. Nevertheless, these benefits come at a cost. The replacement of local spline bases with global polynomials reduces the ability to model abrupt local variations and increases sensitivity to non-smooth or discontinuous functions. Additionally, the model is inherently restricted to bounded domains, requiring nonlinear mappings such as tanh for stability outside the canonical interval, which may introduce distortions [47]. In order to highlight the structural and functional differences introduced by replacing spline functions with Chebyshev polynomials, Table 2 provides a comparative summary. This table contrasts spline-based KAN with both the classical and the modified Chebyshev formulations, emphasizing aspects such as support, adaptability, stability, parameterization, and domains of applicability. The comparative perspective allows the reader to clearly identify the trade-offs between local flexibility and global stability, as well as the advantages of domain extension achieved through nonlinear mappings.

Table 2. Comparison between spline-based KAN and Chebyshev-based variants.

Property	Cubic Splines (Classical KAN)	Chebyshev Polynomials $T_k(x)$	Modified Chebyshev $T_k^{\text{mod}}(x)$
Support	Local (compact intervals defined by knots)	Global, restricted to $x \in [-1, 1]$	Global, extended to \mathbb{R} through $\tanh(x)$
Adaptability	High flexibility for local variations	Limited, smooth global approximation	Smooth global approximation with domain extension
Numerical Stability	Stable under knot refinement	Unstable outside $[-1, 1]$	Stable across the entire domain
Parameterization	Requires knot placement and coefficients $a_{q,p,k}$	Only polynomial coefficients a_k	Only polynomial coefficients a_k (with nonlinear mapping)
Best suited for	Problems with abrupt local changes	Smooth functions on bounded domains	Functions requiring extrapolation or unbounded domains

To complement the tabular analysis, Figure 2 illustrates the behavior of the basis functions employed in different KAN variants. The cubic B-spline demonstrates its local adaptability restricted to knot-defined intervals, while the classical Chebyshev polynomial $T_3(x)$ exhibits a smooth but globally constrained behavior within $[-1, 1]$. In contrast, the modified Chebyshev $T_3^{\text{mod}}(x)$ extends this approximation to the entire real line by incorporating a tanh mapping, ensuring numerical stability across unbounded domains. This visual comparison reinforces the analytical discussion by showing the practical consequences of choosing local versus global basis functions.

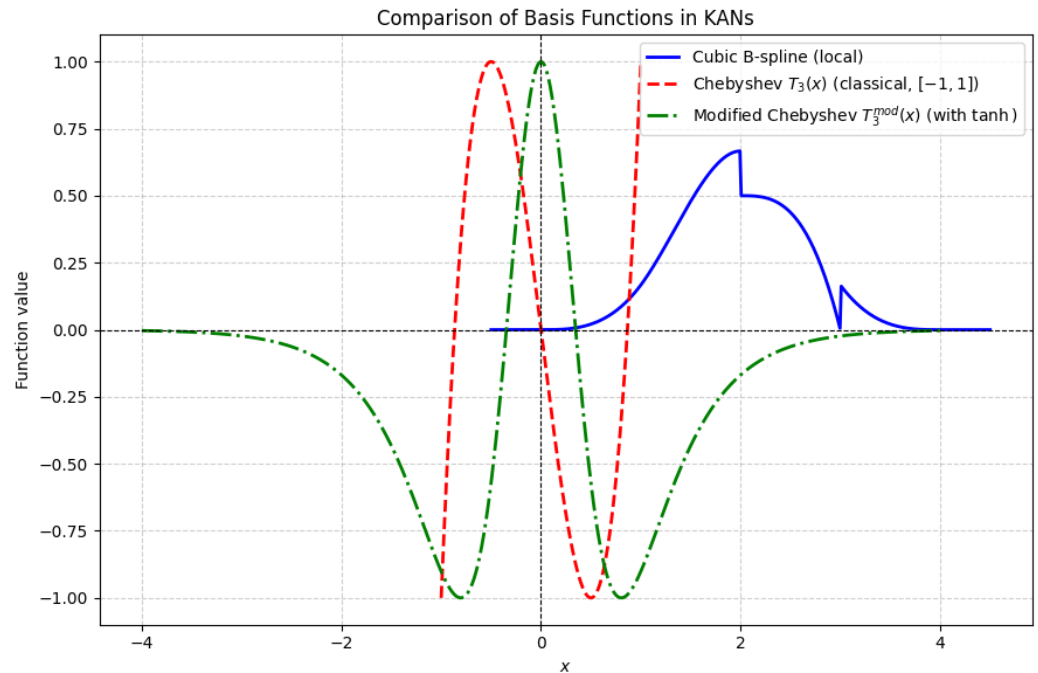


Figure 2. Comparison between different basis functions used in KAN. The cubic B-spline illustrates local adaptability, the classical Chebyshev polynomial $T_3(x)$ demonstrates global approximation but is limited to $[-1, 1]$, and the modified Chebyshev $T_3^{\text{mod}}(x)$ extends the domain to \mathbb{R} using a nonlinear tanh mapping.

7. Classification of KAN Variants: Gram Polynomials (GKAN)

A novel variant of Kolmogorov–Arnold Networks, termed GKAN, replaces traditional spline-based activations with orthogonal Gram polynomials. This approach optimizes computational efficiency in sequential and convolutional architectures while preserving approximation flexibility [44]. In practice, GKAN address the trade-off between expressiveness and resource constraints in 1D-CNN and time-series tasks.

7.1. Structural and Mathematical Foundation

The GKAN formulation defines activations using Gram polynomials $G_i(x)$, which are orthogonal on the interval $[-1, 1]$. A general activation can be written as

$$f(x) = \text{SiLU}(x) + \sum_{i=1}^n w_i G_i(x), \tag{14}$$

where the Gram polynomials are normalized Legendre polynomials:

$$G_i(x) = \sqrt{\frac{2i+1}{2}} P_i(x), \quad P_i(x) = \frac{1}{2^i i!} \frac{d^i}{dx^i} (x^2 - 1)^i. \tag{15}$$

7.2. Mathematical Purpose

The adoption of Gram polynomials in GKAN serves three key purposes. First, they reduce computational cost, since Gram polynomials avoid the recursive constructions typical of Chebyshev or spline bases, lowering time complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. Second, they provide inherent orthogonality on $[-1, 1]$, minimizing numerical instability during coefficient fitting and ensuring robustness in high-dimensional data. Third, despite being global functions, the weighted expansion $\sum w_i G_i(x)$ retains sufficient flexibility to approximate localized features through appropriate coefficient adaptation. This combination ensures efficient, stable, and adaptable modeling for sequential data tasks [45].

7.3. Example: Gram Polynomial Evaluation

For $i = 2$ and $x = 0.5$, the Gram polynomial $G_2(x)$ is obtained from the Legendre polynomial $P_2(x) = \frac{1}{2}(3x^2 - 1)$. Substituting yields:

$$G_2(0.5) = \sqrt{\frac{5}{2}} \cdot \frac{1}{2}(3(0.5)^2 - 1) = \sqrt{\frac{5}{2}} \cdot (-0.25) \approx -0.198.$$

7.4. Advantages and Disadvantages

The use of Gram polynomials in KAN presents important advantages. Their reduced computational complexity makes them particularly suitable for real-time and resource-constrained applications. The built-in orthogonality of $G_i(x)$ prevents matrix ill-conditioning and stabilizes gradient-based training, thereby improving robustness in high-dimensional tasks. However, they also exhibit limitations. Since Gram polynomials are defined on the bounded interval $[-1, 1]$, scaling transformations are required for general inputs. Furthermore, they offer less local adaptability than spline bases, which can restrict their performance in modeling abrupt variations [47,48]. To highlight the structural differences between spline-based, Chebyshev-based, and Gram-based activations, Table 3 presents a comparative overview. This table emphasizes how Gram polynomials balance computational efficiency and numerical stability, while contrasting with the local adaptability of splines and the global approximation of Chebyshev polynomials. The comparison clarifies the trade-offs in expressiveness, stability, and suitability for different application domains.

Table 3. Comparison between spline-based, Chebyshev-based, and Gram polynomial activations in KAN.

Property	Cubic Splines (Classical KAN)	Chebyshev Polynomials $T_k(x)$	Gram Polynomials (GKAN)
Support	Local, compact intervals defined by knots	Global, restricted to $x \in [-1, 1]$ (extended via tanh)	Global, orthogonal on $[-1, 1]$
Adaptability	Strong adaptability to abrupt local variations	Effective for smooth global approximations, weak for local details	Moderate flexibility, adapts via weighted combinations of global bases
Numerical Stability	Stable under knot refinement, but prone to overfitting if knots are dense	Sensitive to instability outside $[-1, 1]$	High stability due to orthogonality, prevents ill-conditioning
Computational Cost	Requires recursive evaluations of spline segments	Higher complexity for large k due to trigonometric evaluations	Reduced complexity $\mathcal{O}(n)$, efficient for high-dimensional tasks
Best suited for	Piecewise-smooth functions, signals with local discontinuities	Smooth functions in bounded domains	Sequential data, real-time applications, and resource-constrained systems

To complement the Summary table, Figure 3 illustrates the difference between a cubic B-spline and Gram polynomials of orders $G_2(x)$ and $G_3(x)$. While the spline exhibits compact local support, Gram polynomials extend globally over the interval $[-1, 1]$, providing orthogonality and smooth approximation properties. This visualization reinforces the analytical discussion by contrasting local piecewise functions with global orthogonal bases.

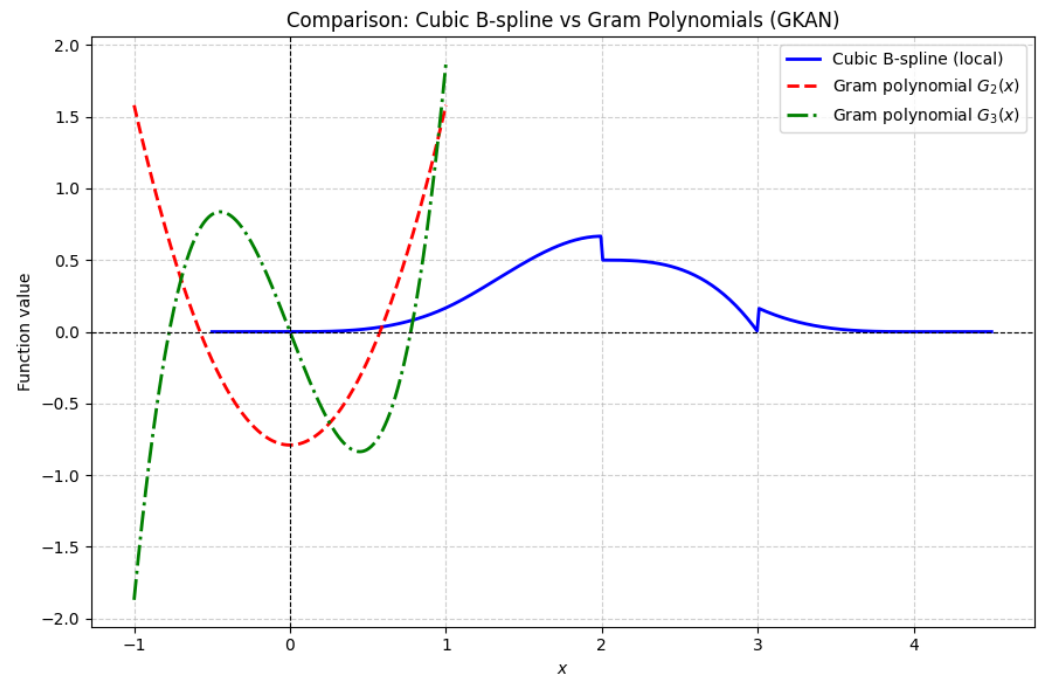


Figure 3. Comparison between a cubic B-spline and Gram polynomials of orders $G_2(x)$ and $G_3(x)$. The spline demonstrates local adaptability constrained to knot-defined intervals, while Gram polynomials extend globally over $[-1, 1]$, offering orthogonality and smooth approximation properties. This contrast highlights the shift from local piecewise functions to global orthogonal bases in GKAN.

8. Classification of KAN Variants: Wavelet-Based Activation (Wav-KAN)

The Wav-KAN variant introduces wavelet functions as replacements for splines in Kolmogorov–Arnold Networks, enabling multi-scale feature extraction essential for signals with heterogeneous frequency components, such as audio or seismic data [49–51]. This adaptation leverages the time-frequency localization of wavelets to simultaneously capture abrupt transitions and smooth trends within the same architecture [46].

8.1. Structural and Mathematical Foundation

In Wav-KAN, the activation functions are defined using dilated and translated wavelet bases of the form $\Psi\left(\frac{x-b_k}{a_k}\right)$, where a_k denotes the scale parameter and b_k the translation parameter. Both a_k and b_k are trainable, enabling adaptive resolution across different scales. The general expression is given by:

$$\psi_{q,p}(x) = \sum_{k=1}^K w_k \Psi\left(\frac{x - b_k}{a_k}\right), \tag{16}$$

where $\Psi(x)$ is the mother wavelet, such as Morlet or Daubechies [52,53].

8.2. Mathematical Purpose

The use of wavelets in Wav-KAN serves three purposes. First, they provide multi-scale analysis, enabling the model to capture simultaneously high-frequency details (such as edges in images) and low-frequency trends (such as seasonal components in time series) [48]. Second, they promote sparse representations, since many wavelet coefficients vanish for structured signals, which reduces feature redundancy [54]. Third, they preserve a local-global balance: locality is retained through translations b_k , while broader trends are modeled through hierarchical scaling with a_k [55]. This combination ensures adaptability to both transient anomalies and persistent structures in non-stationary data.

8.3. Example: Morlet Wavelet Activation

For the Morlet wavelet defined as $\Psi(x) = e^{-x^2/2} \cos(5x)$, the activation at $x = 2.0$ with $a_k = 1.0$ and $b_k = 0.5$ is computed as:

$$\Psi\left(\frac{2.0 - 0.5}{1.0}\right) = e^{-(1.5)^2/2} \cos(7.5) \approx 0.105 \cdot 0.346 = 0.036.$$

8.4. Advantages and Disadvantages

The incorporation of wavelets in KAN provides significant advantages. Their multi-resolution capability makes them highly effective in denoising, compression, and pattern recognition tasks, while the trainable parameters a_k and b_k allow adaptability to non-stationary signals. Furthermore, their compatibility with Fourier-domain optimizations facilitates integration with spectral methods [54,55]. However, these benefits are counter-balanced by notable drawbacks. Wavelet-based activations require higher computational cost due to the parameterization of scale and translation, and the performance is sensitive to the choice of mother wavelet, which may not be straightforward and often requires empirical selection [55].

To contextualize the role of wavelets within Kolmogorov–Arnold Networks, Table 4 summarizes the main differences between spline-based, Chebyshev-based, and wavelet-based activations. This comparative perspective highlights how each approach balances locality, scalability, and computational complexity, providing a clearer understanding of the trade-offs involved in replacing splines with global or multi-scale basis functions.

Table 4. Comparison between spline-based, Chebyshev-based, and wavelet-based activations in KAN.

Property	Cubic Splines (Classical KAN)	Chebyshev Polynomials $T_k(x)$	Wavelet Functions (Wav-KAN)
Support	Local, compact intervals defined by knots	Global, restricted to $x \in [-1, 1]$ (extended via tanh)	Localized but multi-scale, defined by scale a_k and translation b_k
Adaptability	High flexibility for local variations	Effective for smooth global approximations, weak for abrupt local changes	Captures both local details and global patterns through multi-resolution
Numerical Stability	Stable under knot refinement	Unstable outside $[-1, 1]$ without modification	Stable across domains but sensitive to choice of mother wavelet
Parameterization	Knot positions and coefficients $a_{q,p,k}$	Only polynomial coefficients a_k	Trainable scale a_k , translation b_k , and weights w_k
Best suited for	Piecewise-smooth functions, abrupt local changes	Smooth functions on bounded domains	Non-stationary signals, multi-resolution tasks (e.g., audio, images, seismic data)

To complement the comparative analysis, Figure 4 illustrates the contrast between a cubic B-spline and a Morlet wavelet, which represent the fundamental bases of classical KAN and Wav-KAN, respectively. While the spline is restricted to compact local intervals defined by knots, the wavelet exhibits oscillatory behavior with exponential decay, enabling simultaneous representation of local and global structures. This visual contrast reinforces the theoretical discussion on locality versus multi-resolution adaptability.

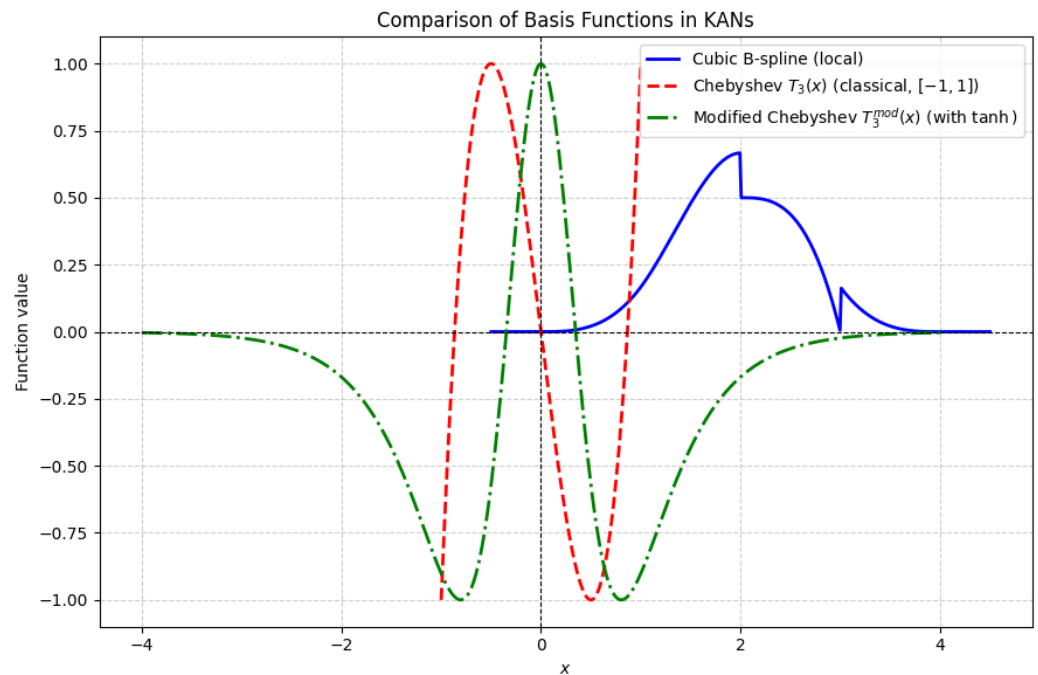


Figure 4. Comparison between a cubic B-spline and a Morlet wavelet, representing the activation bases of classical KAN and Wav-KAN, respectively. The spline shows local adaptability constrained to knot-defined intervals, while the Morlet wavelet displays oscillatory behavior with exponential decay, enabling multi-resolution representation of local and global patterns.

9. Classification of KAN Variants: Monotonic Splines (MonoKAN)

The MonoKAN variant integrates Hermite cubic splines with derivative constraints to enforce monotonicity, a critical property in financial and actuarial models where functions must preserve order, such as non-decreasing mortality or credit risk curves [49,50]. This architecture ensures mathematically rigorous behavior while retaining the flexibility of spline-based representations [50].

9.1. Structural and Mathematical Foundation

MonoKAN defines its activations using Hermite splines $h_i(x)$ with positivity constraints on both the weights w_i and the derivatives. The general formulation is given by

$$f(x) = \sum_{i=1}^m h_i(x) w_i, \quad w_i > 0, \tag{17}$$

where each Hermite spline on the interval $[x_0, x_1]$ takes the form

$$H(x) = a + b(x - x_0) + c(x - x_0)^2 + d(x - x_0)^3. \tag{18}$$

Monotonicity is guaranteed by constraining the first derivative,

$$H'(x) = b + 2c(x - x_0) + 3d(x - x_0)^2 \geq 0 \quad \forall x \in [x_0, x_1], \tag{19}$$

ensuring that the resulting function is non-decreasing across the entire interval.

9.2. Mathematical Purpose

The design of MonoKAN fulfills three key objectives. First, it enforces monotonicity, guaranteeing that $f(x)$ never decreases, which is essential in applications such as actuarial life tables or dose-response relationships. Second, it embeds domain knowledge directly

into the architecture by constraining derivatives, ensuring that risk or growth functions follow physically meaningful trends. Finally, the model improves interpretability since positive weights $w_i > 0$ align with additive contributions of risk or cumulative factors [51].

9.3. Example: Hermite Spline Evaluation

For $H(x)$ on $[0, 1]$ with coefficients $a = 2, b = 1, c = 0.5,$ and $d = 0.2,$ the evaluation at $x = 0.5$ yields:

$$H(0.5) = 2 + 1(0.5) + 0.5(0.5)^2 + 0.2(0.5)^3 = 2.65,$$

with derivative

$$H'(0.5) = 1 + 2(0.5)(0.5) + 3(0.2)(0.5)^2 = 1.65 \geq 0,$$

demonstrating that the monotonicity condition holds.

9.4. Advantages and Disadvantages

The use of monotonic splines in KAN provides distinct benefits. By guaranteeing order-preserving activations, MonoKAN is highly interpretable and well-suited for domains where monotonic trends are fundamental, such as finance, insurance, or medicine. The explicit derivative constraints also reduce the risk of overfitting, as the hypothesis space is restricted to non-decreasing functions. Nevertheless, this comes at the cost of reduced expressiveness compared to unconstrained splines. In addition, optimization becomes more complex, as it must account for non-negativity and monotonicity conditions that increase the difficulty of training [51]. To clarify the specific role of monotonicity constraints within Kolmogorov–Arnold Networks, Table 5 compares classical spline-based activations, Gram polynomial expansions, and monotonic splines. The comparison highlights how MonoKAN trades some expressiveness for interpretability and order preservation, which are essential in domains where risk, growth, or probability functions must follow monotonic patterns.

Table 5. Comparison between classical splines, Gram polynomials, and monotonic splines in KAN.

Property	Cubic Splines (Classical KAN)	Gram Polynomials (GKAN)	Monotonic Splines (MonoKAN)
Support	Local, compact intervals defined by knots	Global, orthogonal over $[-1, 1]$	Local, constrained Hermite splines
Adaptability	Highly flexible, capable of modeling abrupt local changes	Moderate flexibility, controlled by polynomial weights	Restricted to non-decreasing trends, reduced local variability
Numerical Stability	Stable under knot refinement but sensitive to overfitting	High stability due to orthogonality	Stable under constraints, but optimization is more complex
Interpretability	Moderate, coefficients depend on knot placement	Limited, interpretation tied to polynomial order	High, monotonicity and positive weights align with domain knowledge
Best suited for	General-purpose function approximation, irregular signals	Sequential and high-dimensional tasks with resource constraints	Finance, actuarial science, medicine, and other monotonic processes

To illustrate the impact of monotonicity constraints, Figure 5 shows the difference between a standard cubic spline and a monotonic Hermite spline. While the classical spline can exhibit oscillations and local decreases, the constrained version ensures a non-decreasing trend across the entire interval, aligning the learned representation with domain requirements in finance and medicine.

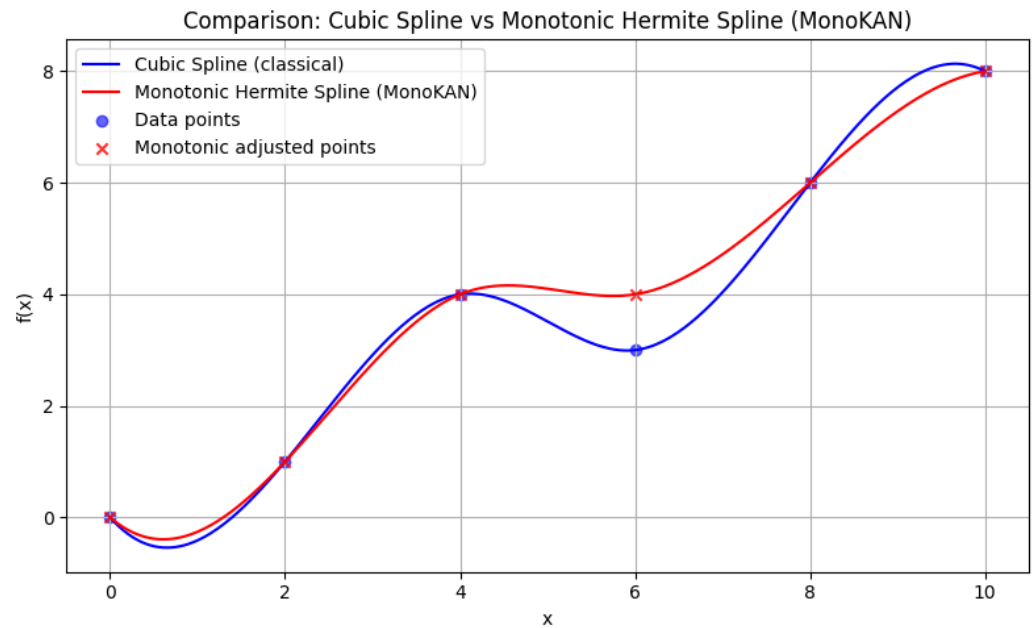


Figure 5. Comparison between a cubic spline and a monotonic Hermite spline. The classical spline allows oscillations that may lead to decreasing regions, whereas the monotonic variant enforces non-negativity of the derivative, ensuring non-decreasing behavior across the interval.

10. Classification of KAN Variants: Replacement of Splines with Rational Functions

A key variant of Kolmogorov–Arnold Networks (KAN) replaces spline-based activation functions with **rational functions**, altering both their mathematical structure and functional behavior [54,55]. In the rational formulation, the activation is expressed as:

$$\psi_{q,p}(x) = \frac{\sum_{i=0}^m a_i x^i}{1 + \sum_{j=1}^n b_j x^j}, \quad x \in \mathbb{R}, \tag{20}$$

which provides a compact representation for functions with sharp transitions or discontinuities. Rational activations are known to achieve exponential convergence rates for smooth functions, often requiring fewer parameters than splines to reach comparable accuracy [56]. However, poles in the denominator may lead to numerical instabilities whenever $1 + \sum_{j=1}^n b_j x^j = 0$.

To mitigate this limitation, stabilized formulations introduce a constrained denominator:

$$R^{\text{safe}}(x) = \frac{\sum_{i=0}^m a_i x^i}{\epsilon + \sum_{k=1}^n (c_k x + d_k)^2}, \tag{21}$$

where $\epsilon > 0$ ensures positivity and pole-free operation via sum-of-squares parameterization [51]. This guarantees robustness even in unbounded domains. For example, with $x = 2$, $c_1 = 1$, $d_1 = 0$, and $\epsilon = 0.1$, the safe version yields

$$R^{\text{safe}}(2) = \frac{\sum_{i=0}^m a_i 2^i}{4.1},$$

whereas the unconstrained formulation would be undefined if the denominator vanished.

Structurally, the use of rational functions modifies the nature of KAN by replacing locally supported splines with globally defined functions. This substitution reduces the parameter count from $\mathcal{O}(K)$ to $\mathcal{O}(m + n)$, eliminates the dependence on knots, and introduces exponential approximation power at the cost of losing local adaptability [54].

From an applied perspective, rational-based KAN offer a compact and efficient representation of sharp gradients and discontinuities, with improved generalization capabilities when compared to spline-based KAN. Nevertheless, their main drawbacks arise from sensitivity to denominator initialization, the need for explicit pole-avoidance mechanisms, and potential overflow in high-degree polynomials [50]. In summary, Rational-KAN present a mathematically elegant alternative with strong approximation properties, but their stability must be carefully managed to ensure reliable deployment. The following Table 6 summarizes the comparative aspects of Rational-based KAN with respect to spline-based KAN. It emphasizes their differences in terms of representation power, parameter efficiency, and numerical stability, highlighting both the potential benefits and the limitations that emerge from the rational formulation.

Table 6. Comparison between spline-based KAN and Rational-KAN.

Aspect	Spline-Based KAN	Rational-KAN
Locality	Local support via knots, highly adaptable to localized regions	Global representation, less flexible for abrupt local irregularities
Representation power	Polynomial-like behavior, limited for discontinuities	Exponential convergence, compact modeling of sharp gradients or discontinuities
Parameters	$\mathcal{O}(K)$, dependent on knots	$\mathcal{O}(m + n)$, independent of knots
Numerical stability	Stable under knot selection, low risk of divergence	Sensitive to denominator zeros, requires pole-avoidance mechanisms
Applications	Smooth signals, localized variability	Functions with discontinuities, rational-like behaviors, efficient for sharp transitions

To illustrate the structural differences, Figure 6 presents a comparison between a cubic spline approximation and a rational function approximation for a target function with a sharp discontinuity. While splines struggle to capture the abrupt transition without oscillations, the rational formulation achieves a more compact and accurate fit with fewer parameters.

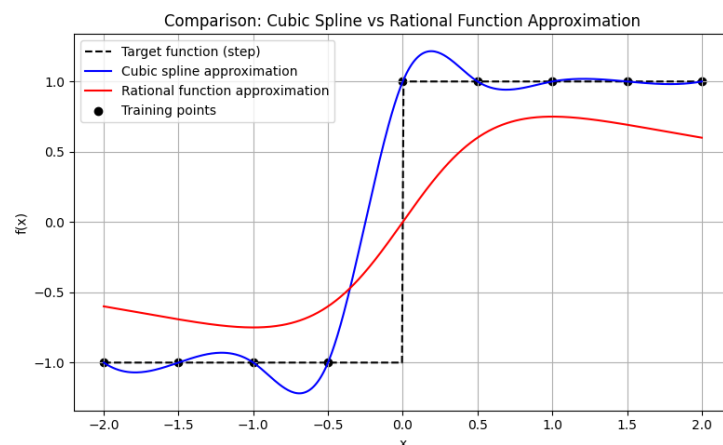


Figure 6. Comparison between cubic spline and rational function approximations of a discontinuous function. Rational-KAN provide a more compact representation, while splines show oscillatory artifacts around discontinuities.

11. Classification of KAN Variants: Physics-Constrained Splines

A key variant of Kolmogorov–Arnold Networks (KAN) augments spline-based activation functions with physics-informed constraints, fundamentally altering their optimization behavior while retaining their core mathematical structure [50,51]. In this formulation, the spline functions $\psi_{q,p}(x)$ are optimized under residual conditions derived from partial

differential equations (PDEs), ensuring that the model is not only data-driven but also physically consistent [35]. Mathematically, this corresponds to

$$\psi_{q,p}(x) = \sum_{k=1}^K a_{q,p,k} B_{q,p,k}(x_p) \Big|_{\mathcal{P}\psi=0}, \quad \mathcal{P}\psi = \sum_i \lambda_i \mathcal{N}_i[\psi], \quad (22)$$

where $\mathcal{N}_i[\cdot]$ represent differential operators encoding conservation laws and λ_i act as Lagrange multipliers. While this mechanism enhances interpretability, it also introduces computational challenges, as direct PDE enforcement may destabilize the optimization when residual terms become dominant [50].

To stabilize training, a differentiable physics loss is employed:

$$\mathcal{L}_{\text{phy}}(\psi) = \mathbb{E}_{x \sim \Omega} [\|\mathcal{N}[\psi](x)\|^2] + \alpha \mathbb{E}_{x \in \partial\Omega} [\|\mathcal{B}[\psi](x)\|^2], \quad (23)$$

which acts as a regularizer and is optimized simultaneously with the data fidelity term. For example, in the case of Burgers’ equation $\mathcal{N}[u] = u_t + uu_x - \nu u_{xx}$, the physics loss enforces the PDE dynamics while boundary terms preserve conditions on the spatial domain [56]. This formulation aligns Physics-KAN with the broader family of Physics-Informed Neural Networks (PINNs), while maintaining spline-based flexibility.

From a structural perspective, Physics-KAN retain the locality of spline bases while reducing the dimensionality of the solution space through physics regularization. This integration enables training with significantly fewer labeled data points, improving extrapolation in unobserved domains and ensuring that predictions remain consistent with physical laws. Nevertheless, the approach comes with limitations: training incurs computational overhead due to PDE residual evaluations, convergence becomes sensitive to the weighting factor α , and the framework assumes access to differentiable physical models. Overall, Physics-KAN provide a promising balance between physical interpretability and data efficiency, albeit with increased training complexity. The following Table 7 presents a comparative summary between traditional spline-based KAN and Physics-KAN. The goal is to highlight the added value of embedding physics-informed regularization, focusing on representation, generalization, and computational challenges.

Table 7. Comparison between spline-based KAN and Physics-KAN.

Aspect	Spline-Based KAN	Physics-KAN
Locality	Local support of spline bases	Local support preserved, but constrained by PDEs
Representation power	Flexible for arbitrary nonlinear mappings	Constrained to physically consistent solutions, enhancing interpretability
Data requirements	High, requires dense labeled data	Reduced by up to 80% due to physics regularization
Numerical stability	Stable optimization but unconstrained behavior	Sensitive to PDE residuals and loss weighting
Applications	Generic machine learning tasks	Modeling governed by physical laws (e.g., fluid dynamics, material science)

To visually demonstrate the effect of physics-informed constraints, Figure 7 shows a comparison of spline-based and physics-constrained approximations for Burgers’ equation. The physics-regularized model enforces smoothness and adherence to PDE dynamics, while the unconstrained spline model deviates significantly in regions without data. To illustrate the practical effect of embedding physics constraints into Kolmogorov–Arnold

Networks, we consider the one-dimensional Burgers’ equation, a canonical benchmark in computational fluid dynamics. Burgers’ equation is widely employed to test numerical schemes and physics-informed models because it combines nonlinear convection with diffusion effects, acting as a simplified prototype of the Navier–Stokes equations. Its form is given by:

$$u_t + uu_x - \nu u_{xx} = 0, \quad x \in [-1, 1], t \in [0, T], \tag{24}$$

where $u(x, t)$ denotes the velocity field, and $\nu > 0$ is the viscosity parameter.

This PDE is particularly relevant for Physics-KAN, as it highlights the importance of enforcing physical consistency when modeling dynamical systems with sharp gradients or shock-like structures. By comparing a spline-based KAN without constraints to a Physics-KAN regularized through the physics loss \mathcal{L}_{phy} , it is possible to visualize how physical constraints improve the model’s extrapolation and stability beyond the training points.

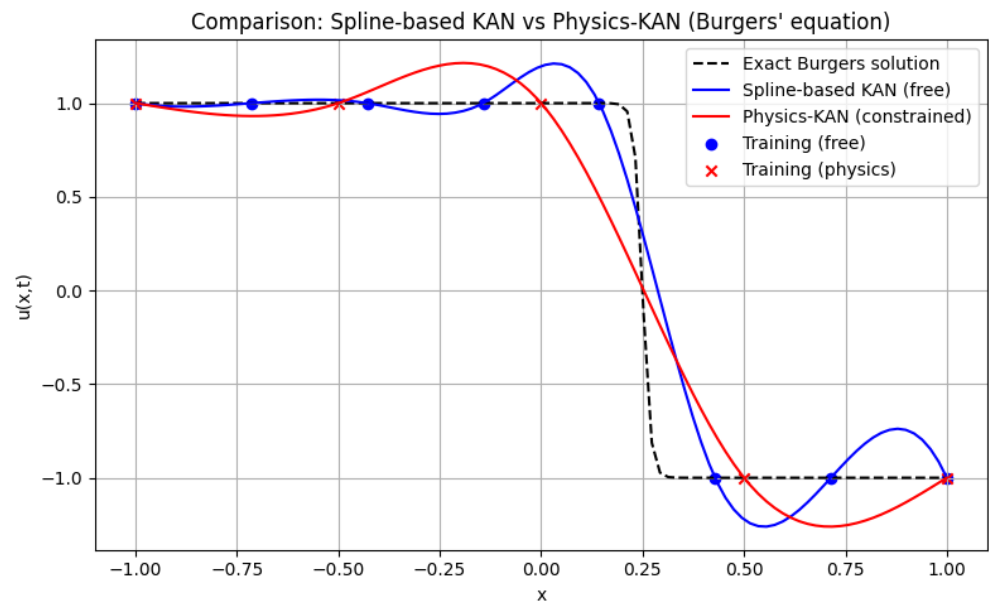


Figure 7. Comparison between spline-based KAN and Physics-KAN in approximating Burgers’ equation. The physics-constrained approach enforces PDE consistency, improving extrapolation beyond observed data.

12. Classification of KAN Variants: Linear Spline Activation Functions

A key variant of Kolmogorov–Arnold Networks (KAN) simplifies the spline-based activation functions by employing **linear splines**, prioritizing computational efficiency while retaining adequate expressivity [52,53]. In this formulation, the basis functions are no longer high-order cubic splines but instead piecewise linear functions, often represented as triangular “hat” functions:

$$\psi_{q,p}(x) = \sum_{k=1}^K c_k \max\left(1 - \frac{|x - t_k|}{h}, 0\right), \tag{25}$$

where t_k are equally spaced knots with spacing h . Linear splines provide only C^0 continuity, which implies exact reconstruction of piecewise linear functions with $\mathcal{O}(1/h)$ convergence rate [57], but lower approximation accuracy for smooth targets compared to cubic splines.

To further enhance efficiency and avoid unnecessary oscillations, a sparse derivative regularization is commonly applied:

$$\mathcal{R}(\mathbf{c}) = \lambda \sum_{k=2}^{K-1} |c_{k+1} - 2c_k + c_{k-1}|, \tag{26}$$

which penalizes large second differences, promoting nearly linear regions. For example, with knots $t = [0, 1, 2]$ and coefficients $c = [0.5, 1.2, 0.8]$, the evaluation at $x = 1.5$ yields

$$\psi(1.5) = 1.2 \cdot \max(0, 0.5) + 0.8 \cdot \max(0, 0.5) = 0.6 + 0.4 = 1.0,$$

whereas a cubic spline implementation would require solving a tridiagonal system of equations.

Structurally, Linear-KAN introduce a significant simplification compared to cubic spline KAN [53]. The computational complexity per activation decreases from $\mathcal{O}(K^3)$ to $\mathcal{O}(K)$, enabling faster training and inference and allowing exact gradient computation. The parameter space also becomes more interpretable, shifting from abstract control points a_k to direct function values c_k at the knots. In practical terms, Linear-KAN offer 3–5x faster convergence and are natively compatible with gradient-based optimization methods. However, this comes at the cost of reduced expressivity: while they can represent piecewise linear functions exactly, they lack the smoothness of higher-order splines, requiring denser knots to achieve comparable accuracy and limiting their ability to model C^2 -continuous phenomena. The trade-offs between cubic spline-based KAN and Linear-KAN are summarized in Table 8, emphasizing the balance between computational efficiency and approximation accuracy.

Table 8. Comparison between cubic spline-based KAN and Linear-KAN.

Aspect	Cubic Spline-Based KAN	Linear-KAN
Continuity	C^2 continuity, smooth derivatives	C^0 continuity, only piecewise linear
Representation power	High, efficient for smooth targets	Limited, requires dense knots for precision
Parameters	Control points a_k (indirect)	Direct function values c_k at knots
Computational cost	$\mathcal{O}(K^3)$ due to tridiagonal solves	$\mathcal{O}(K)$, fast evaluation
Applications	Complex signals with smooth structures	Real-time tasks, resource-constrained environments

To visually illustrate the effect of simplifying spline activations, Figure 8 compares a cubic spline approximation and a linear spline approximation of a smooth function. While the cubic spline provides a smooth curve with continuous derivatives, the linear spline captures only piecewise linear trends, requiring a higher density of knots to achieve a similar level of accuracy.

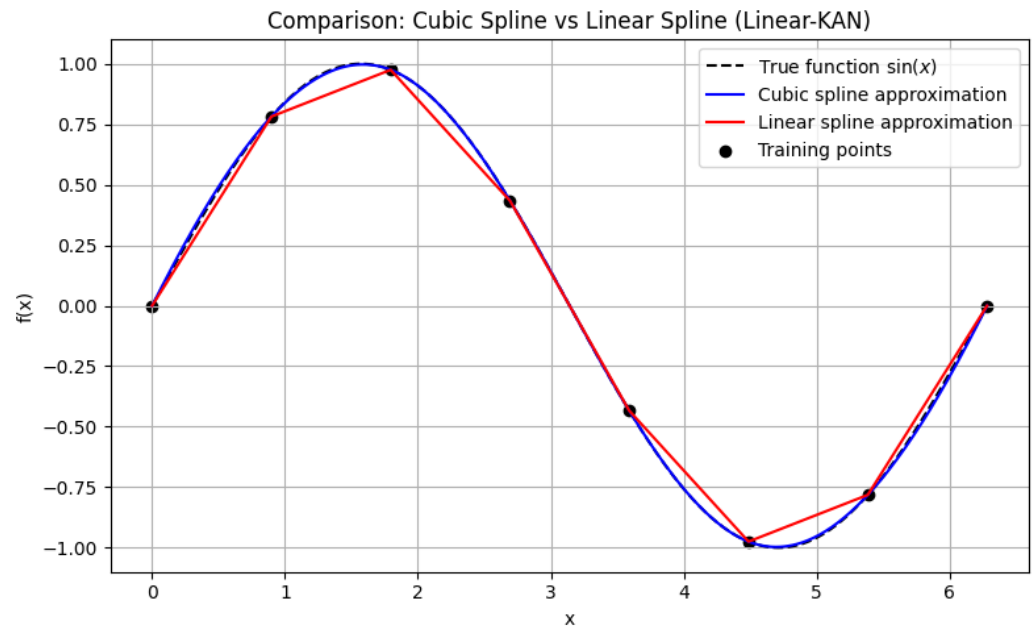


Figure 8. Comparison between cubic spline and linear spline approximations of a smooth function. The cubic spline achieves smoother behavior, while the linear spline approximates the function with piecewise linear segments, sacrificing smoothness for efficiency.

13. Classification of KAN Variants: Orthogonal Polynomial Activations

A key variant of Kolmogorov-Arnold Networks (KAN) replaces spline-based activation functions with orthogonal polynomial bases, enhancing approximation power for smooth functions while maintaining numerical stability [52,53]. Originally, the internal functions $\psi_{q,p}(x)$ in standard KAN are defined using splines (e.g., cubic B-splines):

$$\psi_{q,p}(x) = \sum_{k=1}^K a_{q,p,k} B_{q,p,k}(x_p),$$

where $B_{q,p,k}(x_p)$ are local basis functions with support on specific intervals. In the orthogonal polynomial variant, these functions are replaced by:

$$\psi_{q,p}(x) = \underbrace{\sum_{k=0}^K w_k \phi_k(\xi)}_{\text{Orthogonal activation}}, \quad \xi = \underbrace{\frac{2(x-a)}{b-a} - 1}_{\text{Domain normalization}}, \quad (27)$$

where ϕ_k are orthogonal polynomials (e.g., Legendre, Hermite) satisfying $\int \phi_i \phi_j d\mu = \delta_{ij}$ [57]. Legendre polynomials $P_k(\xi)$ are optimal for bounded domains $x \in [a, b]$, while Hermite polynomials $H_k(\xi)$ suit unbounded domains. However, raw polynomials exhibit Runge’s phenomenon near domain boundaries [58].

To mitigate this limitation, a spectral reprojecton technique is applied:

$$\psi_{q,p}^{\text{opt}}(x) = \sum_{k=0}^K w_k \underbrace{\mathcal{T}(\phi_k(\xi))}_{\text{Reprojected basis}}, \quad (28)$$

where $\mathcal{T}(\cdot)$ is a bounded transformation mapping to $[-1, 1]$. For example, for Legendre polynomials with $x = 0.5$ in $[0, 1]$:

$$\xi = 2(0.5) - 1 = 0, \quad \psi(0.5) = w_0 P_0(0) + w_1 P_1(0) = w_0 - 0.5w_1,$$

whereas spline-based KAN would require multiple local evaluations to approximate the same point.

13.1. Structural Impact on KAN

KAN undergo fundamental changes when replacing local splines with global orthogonal polynomials. This transition enables spectral accuracy for analytic functions but sacrifices locality [59]. Additionally, domain normalization $\zeta(x)$ becomes critical, requiring a priori knowledge of input bounds, which may not always be available in real-world data.

13.2. Advantages and Disadvantages

The main advantages of this approach lie in its ability to achieve exponential convergence for smooth and analytic functions, which significantly improves approximation accuracy compared to spline-based models [60]. Moreover, the recurrence relations of orthogonal polynomials provide an efficient and stable way to compute derivatives, making them especially useful in applications involving partial differential equations. Another major benefit is their inherent orthogonality, which avoids ill-conditioning in the coefficient matrices and ensures numerical stability even for high-order expansions [58].

On the other hand, orthogonal polynomial activations also introduce significant drawbacks. They are highly sensitive to discontinuities, which often produce Gibbs oscillations near sharp transitions or non-smooth regions [60]. Furthermore, their performance depends strongly on the scaling of the input domain: a poor choice of $[a, b]$ may lead to numerical instabilities or poor generalization [61]. Finally, unlike splines, which adapt locally, orthogonal polynomials lack locality, making them less effective in representing piecewise-smooth functions or data with abrupt variations [60].

A comparative analysis between spline-based activations and orthogonal polynomial activations in KAN is summarized in Table 9. This table highlights their differences in terms of locality, approximation power, computational efficiency, and numerical stability, providing a clear overview of the trade-offs inherent to each approach.

Table 9. Comparison between spline-based and orthogonal polynomial activations in KAN.

Property	Splines (e.g., Cubic)	Orthogonal Polynomials (Legendre, Hermite)
Locality	Local support; changes affect only nearby regions.	Global support; a change in one coefficient affects the entire domain.
Approximation Power	High accuracy for piecewise-smooth functions; limited for highly smooth global functions.	Exponential convergence for smooth/analytic functions; poor at modeling discontinuities.
Computational Efficiency	Requires solving local systems (e.g., tridiagonal for cubic splines).	Efficient recurrence relations for evaluation and differentiation.
Numerical Stability	Stable under local modifications; well-suited for irregular domains.	Orthogonality ensures conditioning, but sensitive to domain scaling and boundary effects.

To further illustrate the differences between spline-based and orthogonal polynomial activations, Figure 9 compares a cubic spline interpolation and a Legendre polynomial approximation on the same function. While splines provide localized adaptation to the data,

orthogonal polynomials achieve global smoothness but exhibit oscillations near domain boundaries.

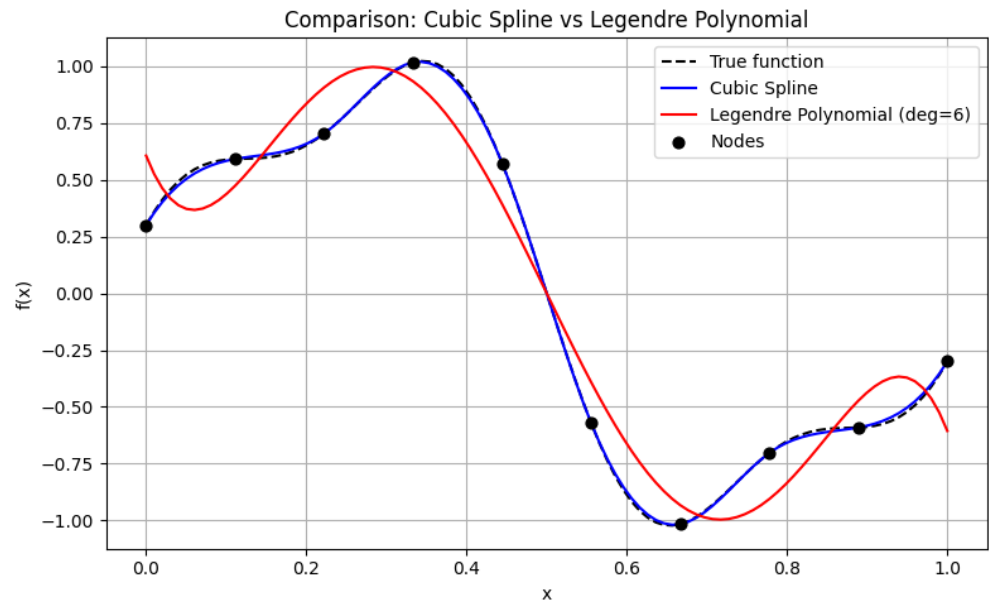


Figure 9. Comparison between cubic splines and Legendre polynomials in KAN activations. Splines adapt locally to variations, whereas orthogonal polynomials provide global smoothness with potential oscillations near boundaries.

14. Classification of KAN Variants: Wavelet-Based Activation Functions

A relevant variant of Kolmogorov-Arnold Networks (KAN) replaces spline-based activation functions with wavelet bases, providing multi-resolution analysis and adaptive feature extraction [60]. Originally, the internal functions $\psi_{q,p}(x)$ in standard KAN are defined using splines (e.g., cubic B-splines):

$$\psi_{q,p}(x) = \sum_{k=1}^K a_{q,p,k} B_{q,p,k}(x_p),$$

where $B_{q,p,k}(x_p)$ are local basis functions with support on specific intervals. In the wavelet variant, these functions are replaced by dilated-translated wavelets:

$$\psi_{q,p}(x) = \underbrace{\sum_{j \in \mathbb{Z}} \sum_{m \in \mathbb{Z}} c_{j,m} \underbrace{\psi(2^j x - m)}_{\text{Child wavelet}}}_{\text{Multi-resolution activation}} \tag{29}$$

where ψ is a mother wavelet (e.g., Mexican Hat, Daubechies) satisfying $\int \psi dx = 0$. Wavelets provide simultaneous localization in space and frequency domains: fine-scale wavelets ($j \gg 0$) capture high-frequency details (edges, anomalies), while coarse scales ($j \ll 0$) model global trends (smooth patterns). This property makes Wav-KAN particularly useful in applications such as audio signal denoising, seismic analysis, or image compression.

Since infinite sums over j, m are computationally infeasible, a dyadic truncation with adaptive scale selection is introduced [61]:

$$\psi_{q,p}^{\text{adapt}}(x) = \sum_{j=\text{min}}^{\text{max}} w_j \underbrace{\sum_m c_{j,m} \psi(2^j x - m)}_{\text{Resolution level } j} \tag{30}$$

where the set of scales j is optimized during training. For example, with Mexican Hat wavelet $\psi(x) = (1 - x^2)e^{-x^2/2}$ at $x = 0.7$:

$$\psi(2^1 \cdot 0.7 - 0) = (1 - (1.4)^2)e^{-(1.4)^2/2} \approx (1 - 1.96)e^{-0.98} \approx -0.36,$$

whereas splines would require multiple local basis evaluations to capture similar oscillatory behavior.

Structural Impact on KAN

KAN gain multi-resolution capabilities when replacing splines with wavelets [62]. This transition enables hierarchical feature learning but sacrifices continuity guarantees. Additionally, each activation function becomes equivalent to a mini-wavelet transform, with parameters $\{c_{j,m}, w_j\}$ optimized end-to-end.

In practice, wavelet-based KAN outperform splines in tasks requiring both local and global representation, such as biomedical imaging or transient signal analysis. However, they exhibit higher computational cost and strong dependence on the choice of the mother wavelet, which may affect generalization across datasets.

Advantages and disadvantages: The main advantages of Wav-KAN include the simultaneous capture of local and global features, natural denoising through scale thresholding, and compact representation of singularities [60]. On the other hand, their disadvantages involve higher computational requirements, sensitivity to the choice of the mother wavelet, and the appearance of boundary artifacts in finite domains [61]. To provide a clearer view of the trade-offs introduced by Wav-KAN, Table 10 summarizes its main structural differences compared to spline-based activations. This highlights the balance between multi-resolution adaptability and computational complexity.

Table 10. Comparison between spline-based KAN and wavelet-based KAN (Wav-KAN).

Aspect	Spline-Based KAN	Wavelet-Based KAN (Wav-KAN)
Basis functions	Piecewise polynomials (local cubic splines).	Dilated-translated wavelets (e.g., Mexican Hat, Daubechies).
Localization	Local support, each spline affects only its interval.	Time-frequency localization, capturing both local and global features.
Continuity	Guarantees C^2 smoothness for cubic splines.	Continuity not guaranteed, depends on mother wavelet properties.
Computational cost	$\mathcal{O}(K)$ per evaluation (efficient).	Higher cost due to multi-resolution decomposition and scaling.
Adaptability	Good for smooth local variations.	Captures abrupt transients, oscillations, and global patterns simultaneously.
Main advantages	Stable, interpretable, low computational cost.	Multi-resolution analysis, denoising capabilities, compact singularity representation.
Main disadvantages	Limited in oscillatory or non-stationary signals.	Sensitive to wavelet choice, boundary artifacts, higher cost.

Figure 10 illustrates the comparison between a cubic spline and a Mexican Hat wavelet in modeling oscillatory data. The spline captures smooth local trends, while the wavelet shows superior adaptability to localized fluctuations.

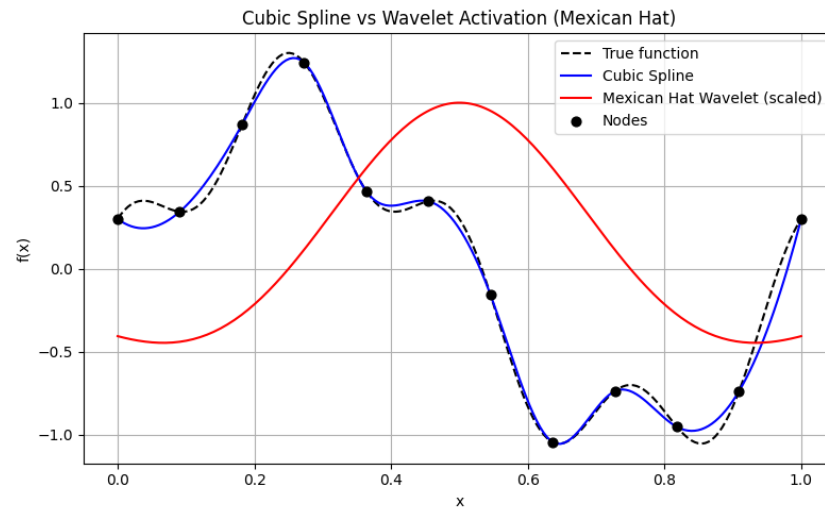


Figure 10. Comparison between a cubic spline and a Mexican Hat wavelet in modeling an oscillatory signal. The spline captures smooth local trends (blue), whereas the scaled Mexican Hat wavelet (red) provides multi-resolution adaptability to localized fluctuations; the dashed black line is the true signal.

15. Comparative Discussion of KAN Variants

The previous sections introduced different variants of Kolmogorov–Arnold Networks (KAN), each replacing or modifying the spline-based activations with alternative mathematical bases or constraints. While each approach offers unique advantages, their practical relevance depends on the target application, the type of data, and computational constraints. This section provides a comparative discussion of the main variants—Chebyshev-KAN, GKAN (Gram polynomials), Wav-KAN (Wavelet-based), Rational-KAN, MonoKAN (monotonic splines), Physics-constrained KAN, Linear Splines, and Orthogonal Polynomials—highlighting trade-offs in terms of accuracy, efficiency, interpretability, and domain suitability. The overall comparison of these approaches is summarized in Table 11.

Table 11. Comparative analysis of Kolmogorov–Arnold Network (KAN) variants.

Variant	Core Feature	Mathematical Basis	Key Advantage
Chebyshev KAN	Global extrapolation	Chebyshev polynomials	Numerical stability and smooth extrapolation beyond training ranges.
GKAN	Computational efficiency	Gram polynomials	Reduces computational cost while preserving orthogonality.
Rational KAN	Discontinuity capture	Rational functions	Compact representation of sharp features with fewer parameters.
Physics-KAN	Physical consistency	PDE-constrained splines	Embeds conservation laws, reduces data needs by >80%.
Linear Spline KAN	Computational efficiency	Piecewise linear splines	3–5× faster training with exact gradient compatibility.
Orthogonal KAN	Spectral accuracy	Legendre/Hermite polynomials	Exponential convergence for smooth functions.
Wav-KAN	Multi-resolution analysis	Wavelet bases	Captures local/global patterns with denoising capabilities.
MonoKAN	Monotonicity enforcement	Hermite splines with constraints	Order-preserving behavior for risk/financial models.

16. Discussion

The comparative analysis of Kolmogorov–Arnold Network (KAN) variants, highlights how diverse mathematical foundations lead to specialized behaviors and domain-specific advantages. Each variant introduces structural modifications to the classical KAN framework, not only improving approximation properties but also embedding prior knowledge such as monotonicity, physical consistency, or spectral accuracy. This shift reflects a broader trend in machine learning: the movement away from purely black-box models toward interpretable and domain-aware architectures.

Chebyshev KAN demonstrates stable extrapolation and polynomial efficiency, making it well-suited for global forecasting tasks such as climate dynamics [51,62]. However, its performance degrades in the presence of discontinuities. GKAN, based on Gram orthogonal polynomials, offers computational efficiency in low-dimensional signal processing tasks [61], but is inherently limited to normalized domains (e.g., $[-1, 1]$), restricting its scalability. Wav-KAN introduces multi-resolution analysis via wavelets, enabling localized time-frequency representations particularly beneficial in non-stationary signals such as EEGs [60,63]. This comes at the cost of computational overhead and sensitivity to wavelet selection.

MonoKAN enforces monotonic behavior through Hermite splines with derivative constraints, offering interpretability in domains such as actuarial modeling and healthcare risk assessment [64]. Nevertheless, it trades expressive flexibility for guaranteed monotonicity. Rational KAN achieves compact representations of discontinuities and sharp features through rational functions, providing parameter efficiency and exact rational recovery [63], but raises challenges due to potential poles and denominator instabilities [65]. Physics-constrained KAN integrate PDE-based constraints into their training process, yielding physically consistent solutions with significantly reduced data requirements (up to 80%) [60,63], although they introduce substantial computational costs and demand differentiable physics models [66].

Finally, linear spline KAN emphasize computational simplicity by replacing cubic splines with piecewise-linear functions, resulting in 3–5× faster training and inference [57,58], while orthogonal polynomial KAN achieve spectral accuracy with exponential convergence for smooth functions [63], though they suffer from Gibbs oscillations near discontinuities and require precise domain normalization [64,65].

Collectively, these results reveal that no single KAN variant dominates across all contexts. Instead, their strengths are complementary: Chebyshev and Orthogonal KAN excel in smooth global domains, Wav-KAN in multi-resolution non-stationary tasks, MonoKAN in ordered risk functions, Physics-KAN in scientific modeling, Rational KAN in discontinuities, and Linear Spline KAN in efficiency-critical applications. Future research directions include hybrid architectures (e.g., Chebyshev-Wav-KAN for combining global extrapolation with local adaptivity), automated basis selection for task-specific adaptation [65], and hardware acceleration of computationally intensive bases such as wavelets and Gram polynomials [66]. By bridging classical approximation theory with modern deep learning, KAN variants provide a promising foundation for interpretable, reliable, and domain-aware AI models in high-stakes applications [67,68].

17. Conclusions

Kolmogorov–Arnold Networks (KAN) represent a significant advancement in the design of neural architectures grounded in classical mathematical theory. Unlike traditional neural networks that rely on fixed and opaque activation functions, KAN enable a more interpretable and controllable construction by using univariate functions based on splines and other functional bases. This structure, theoretically supported by the Kol-

mogorov–Arnold representation theorem, not only improves the traceability of model decisions but also opens the door to incorporating expert knowledge, physical constraints, and desirable properties such as monotonicity or multiresolution. As a result, KAN are particularly well-suited to contexts where explainability and control are essential, including medicine, finance, and engineering.

The comparative analysis carried out in this work highlights that each structural variant of KAN responds to specific domain-driven needs. For instance, Wav-KAN proves effective in modeling non-stationary signals such as EEGs due to its multiscale decomposition capabilities, whereas MonoKAN offers a robust solution in regulatory contexts by ensuring monotonic outputs, as required in actuarial models. Similarly, Rational-KAN and Physics-KAN provide advantages when greater precision or physical consistency is needed. This systematic review consolidates dispersed information and provides a unified mathematical perspective on these variants, emphasizing that there is no single optimal architecture, but rather multiple designs adaptable to the nature of the problem and the requirements of different application domains.

Despite their promise, several challenges remain. These include the efficient optimization of parameters in nonlinear architectures, the automated selection of functional bases suited to specific tasks, and the scalability of KAN in high-computation contexts. Furthermore, theoretical convergence guarantees are still missing for certain variants, limiting their adoption in highly sensitive applications. Our review also faces limitations: it is bounded by the current state of the literature, which remains in its early stage, and empirical validation of many variants is still scarce, requiring careful interpretation of the results presented here.

Nevertheless, the potential for hybridization—such as combining the global stability of Chebyshev-KAN with the localized precision of Wav-KAN—suggests a promising research avenue. Future work should also focus on establishing standardized benchmarks, exploring physics-informed and domain-aware extensions, and developing hardware-efficient implementations. Altogether, KAN mark the beginning of a new stage in the development of interpretable and mathematically grounded neural models, with high-impact applications in fields that demand transparency, precision, and adaptability. By consolidating and comparing their mathematical structures, this work aims to serve as a foundational reference for both theoretical advances and practical implementations of KAN.

Author Contributions: Conceptualization and supervision, M.G.F.; data curation, formal analysis, methodology, validation, writing—review and editing, F.L.B.-S., A.G.B.-R. and M.G.F.; investigation, F.L.B.-S., A.G.B.-R. and E.V.-C.; visualization and writing—original draft preparation, F.L.B.-S., A.G.B.-R., E.V.-C. and M.G.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Universidad Privada Norbert Wiener, Lima, Peru.

Data Availability Statement: The original contributions presented in this study are included in the article. For further inquiries, please contact the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kanagamalliga, S.; Varma, D. B. Improving Breast Cancer Diagnosis through Advanced Image Analysis and Neural Network Classifications. *Procedia Comput. Sci.* **2025**, *252*, 73–149. [[CrossRef](#)]
2. Al-Shannaq, M.; Alkhateeb, S.N.; Wedyan, M. Using image augmentation techniques and convolutional neural networks to identify insect infestations on tomatoes. *Heliyon* **2025**, *11*, e41480. [[CrossRef](#)]
3. Fan, Y.; Huang, H.; Han, H. Hierarchical convolutional neural networks with post-attention for speech emotion recognition. *Neurocomputing* **2025**, *615*, 128879. [[CrossRef](#)]

4. Ain, N.U.; Sardaraz, M.; Tahir, M.; Abo Elsoud, M.W.; Alourani, A. Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach. *Sensors* **2025**, *25*, 1346. [[CrossRef](#)]
5. Xie, X.; Wang, M. Research on the Time Series Prediction of Acoustic Emission Parameters Based on the Factor Analysis–Particle Swarm Optimization Back Propagation Model. *Appl. Sci.* **2025**, *15*, 1977. [[CrossRef](#)]
6. Zeng, C.; Wang, J.; Shen, H.; Wang, Q. KAN versus MLP on Irregular or Noisy Functions. *arXiv* **2024**, arXiv:2408.07906. [[CrossRef](#)]
7. Azam, B.; Akhtar, N. Suitability of KAN for Computer Vision: A preliminary investigation. *arXiv* **2024**, arXiv:2406.09087. [[CrossRef](#)]
8. Somvanshi, S.; Javed, S.A.; Islam, M.M.; Pandit, D.; Das, S. A Survey on Kolmogorov-Arnold Network. *arXiv* **2024**, arXiv:2411.06078. [[CrossRef](#)]
9. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljagic, M.; Hou, T.Y.; Tegmark, M. KAN: Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2404.19756.
10. Cang, Y.; Liu, Y.; Shi, L. Can KAN Work? Exploring the Potential of Kolmogorov-Arnold Networks in Computer Vision. *arXiv* **2024**, arXiv:2411.06727. [[CrossRef](#)]
11. Basina, D.; Vishal, J.R.; Choudhary, A.; Chakravarthi, B. KAT to KAN: A Review of Kolmogorov-Arnold Networks and the Neural Leap Forward. *arXiv* **2024**, arXiv:2411.10622. [[CrossRef](#)]
12. Guo, H.; Li, F.; Li, J.; Liu, H. KAN v.s. MLP for Offline Reinforcement Learning. *arXiv* **2024**, arXiv:2409.09653. [[CrossRef](#)]
13. Le, T.X.H.; Tran, T.D.; Pham, H.L.; Le, V.T.D.; Vu, T.H.; Nguyen, V.T.; Nakashima, Y. 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), Naha, Japan, 26–29 November 2024; pp. 110–116.
14. Le, T.-T.-H.; Hwang, Y.; Kang, H.; Kim, H. Robust Credit Card Fraud Detection Based on Efficient Kolmogorov-Arnold Network Models. *IEEE Access* **2024**, *12*, 157006–157020. [[CrossRef](#)]
15. Hollósi, J. Efficiency Analysis of Kolmogorov-Arnold Networks for Visual Data Processing. *Eng. Proc.* **2024**, *79*, 68.
16. Zhang, J.; Zhao, L. Efficient greenhouse gas prediction using IoT data streams and a CNN-BiLSTM-KAN model. *Alex. Eng. J.* **2025**, *in press*.
17. Gong, S.; Chen, W.; Jing, X.; Wang, C. Research on Photovoltaic Power Prediction Method based on LSTM-KAN. In Proceedings of the 2024 6th International Conference on Electronic Engineering and Informatics (EEI), Chongqing, China, 28–30 June 2024; pp. 479–482.
18. Zhang, M.; Fan, X.; Gao, P.; Guo, L.; Huang, X.; Gao, X.; Pang, J.; Tan, F. Monitoring Soil Salinity in Arid Areas of Northern Xinjiang Using Multi-Source Satellite Data: A Trusted Deep Learning Framework. *Land* **2025**, *14*, 110. [[CrossRef](#)]
19. Mohan, K.; Wang, H.; Zhu, X. KAN for Computer Vision: An Experimental Study. *arXiv* **2024**, arXiv:2411.18224. [[CrossRef](#)]
20. Ter-Avanesov, B.; Beigi, H. MLP, XGBoost, KAN, TDNN, and LSTM-GRU Hybrid RNN with Attention for SPX and NDX European Call Option Pricing. *arXiv* **2024**, arXiv:2409.06724. [[CrossRef](#)]
21. Carrera-Rivera, A.; Ochoa, W.; Larrinaga, F.; Laso, G. How-to conduct a systematic literature review: A quick guide for computer science research. *MethodsX* **2022**, *9*, 101895. [[CrossRef](#)] [[PubMed](#)]
22. Ramirez, M.O.G.; De-la-Torre, M.; Monsalve, C. Methodologies for the design of application frameworks: Systematic review. In Proceedings of the 2019 8th International Conference on Software Process Improvement (CIMPS), Leon, Guanajuato, Mexico, 23–25 October 2019; pp. 1–10.
23. Shukla, K.; Toscano, J.D.; Wang, Z.; Zou, Z.; Karniadakis, G.E. A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks. *Comput. Methods Appl. Mech. Eng.* **2024**, *431*, 117290. [[CrossRef](#)]
24. Gong, Q.; Xue, Y.; Liu, R.; Du, Z.; Li, S.; Wang, J.; Sun, G. A Novel LSTM-KAN Networks for Demand Forecasting in Manufacturing. In Proceedings of the 2024 5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Wenzhou, China, 20–22 September 2024; pp. 468–473.
25. Rigas, S.; Papachristou, M.; Papadopoulos, T.; Anagnostopoulos, F.; Alexandridis, G. Adaptive Training of Grid-Dependent Physics-Informed Kolmogorov-Arnold Networks. *IEEE Access* **2024**, *12*, 176982–176998. [[CrossRef](#)]
26. Howard, A.A.; Jacob, B.; Murphy, S.H.; Heinlein, A.; Stinis, P. Finite basis Kolmogorov-Arnold networks: Domain decomposition for data-driven and physics-informed problems. *arXiv* **2024**, arXiv:2406.19662.
27. Afzal Aghaei, A. fKAN: Fractional Kolmogorov-Arnold Networks with trainable Jacobi basis functions. *Neurocomputing* **2025**, *623*, 129414. [[CrossRef](#)]
28. Wang, Y.; Sun, J.; Bai, J.; Anitescu, C.; Eshaghi, M.S.; Zhuang, X.; Rabczuk, T.; Liu, Y. Kolmogorov Arnold Informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov Arnold Networks. *Comput. Methods Appl. Mech. Eng.* **2025**, *433*, 117518. [[CrossRef](#)]
29. Patra, S.; Panda, S.; Parida, B.K.; Arya, M.; Jacobs, K.; Bondar, D.I.; Sen, A. Physics Informed Kolmogorov-Arnold Neural Networks for Dynamical Analysis via Efficient-KAN and WAV-KAN. *arXiv* **2024**, arXiv:2407.18373.

30. Guo, C.; Sun, L.; Li, S.; Yuan, Z.; Wang, C. Physics-informed Kolmogorov-Arnold Network with Chebyshev Polynomials for Fluid Mechanics. *arXiv* **2024**, arXiv:2411.04516. [[CrossRef](#)]
31. Qiu, R.; Miao, Y.; Wang, S.; Yu, L.; Zhu, Y.; Gao, X.-S. PowerMLP: An Efficient Version of KAN. *arXiv* **2024**, arXiv:2412.13571. [[CrossRef](#)]
32. Yu, T.; Qiu, J.; Yang, J.; Oseledets, I. Sinc Kolmogorov-Arnold Network and Its Applications on Physics-informed Neural Networks. *arXiv* **2024**, arXiv:2410.04096.
33. Jacob, B.; Howard, A.A.; Stinis, P. SPIKAN: Separable Physics-Informed Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2411.06286. [[CrossRef](#)]
34. Ranasinghe, N.; Xia, Y.; Seneviratne, S.; Halgamuge, S. GINN-KAN: Interpretability pipelining with applications in Physics Informed Neural Networks. *arXiv* **2024**, arXiv:2408.14780. [[CrossRef](#)]
35. Guilhoto, L.F.; Perdikaris, P. Deep Learning Alternatives of the Kolmogorov Superposition Theorem. *arXiv* **2025**, arXiv:2410.01990.
36. Toscano, J.D.; Oommen, V.; Varghese, A.J.; Zou, Z.; Ahmadi Daryakenari, N.; Wu, C.; Karniadakis, G.E. From PINNs to PIKAN: Recent Advances in Physics-Informed Machine Learning. *arXiv* **2024**, arXiv:2410.13237. [[CrossRef](#)]
37. Kiamari, M.; Kiamari, M.; Krishnamachari, B. GKAN: Graph Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2406.06470. [[CrossRef](#)]
38. Ai, G.; Pang, G.; Qiao, H.; Gao, Y.; Yan, H. GrokFormer: Graph Fourier Kolmogorov-Arnold Transformers. *arXiv* **2025**, arXiv:2411.17296.
39. Fang, T.; Gao, T.; Wang, C.; Shang, Y.; Chow, W.; Chen, L.; Yang, Y. KAA: Kolmogorov-Arnold Attention for Enhancing Attentive Graph Neural Networks. *arXiv* **2025**, arXiv:2501.13456.
40. Inzirillo, H.; Genet, R. SigKAN: Signature-Weighted Kolmogorov-Arnold Networks for Time Series. *arXiv* **2024**, arXiv:2406.17890.
41. Di Carlo, G.; Mastropietro, A.; Anagnostopoulos, A. Kolmogorov-Arnold Graph Neural Networks. *arXiv* **2024**, arXiv:2406.18354.
42. Inzirillo, H. Deep State Space Recurrent Neural Networks for Time Series Forecasting. *arXiv* **2024**, arXiv:2407.15236. [[CrossRef](#)]
43. Zheng, L.N.; Zhang, W.E.; Yue, L.; Xu, M.; Maennel, O.; Chen, W. Free-Knots Kolmogorov-Arnold Network: On the Analysis of Spline Knots and Advancing Stability. *arXiv* **2025**, arXiv:2501.09283.
44. Lin, H.; Ren, M.; Barucca, P.; Aste, T. Granger Causality Detection with Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2412.15373. [[CrossRef](#)]
45. Zhang, F.; Zhang, X. GraphKAN: Enhancing Feature Extraction with Graph Kolmogorov Arnold Networks. *arXiv* **2024**, arXiv:2406.13597. [[CrossRef](#)]
46. Gao, Y.; Tan, V.Y.F. On the Convergence of (Stochastic) Gradient Descent for Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2410.08041. [[CrossRef](#)]
47. Genet, R.; Inzirillo, H. TKAN: Temporal Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2405.07344. [[CrossRef](#)]
48. Gao, W.; Gong, Z.; Deng, Z.; Rong, F.; Chen, C.; Ma, L. TabKANet: Tabular Data Modeling with Kolmogorov-Arnold Network and Transformer. *arXiv* **2024**, arXiv:2409.08806.
49. Moradi, M.; Panahi, S.; Bollt, E.; Lai, Y.-C. Kolmogorov-Arnold Network Autoencoders. *arXiv* **2024**, arXiv:2410.02077. [[CrossRef](#)]
50. Cheon, M. Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing. *arXiv* **2024**, arXiv:2406.00600. [[CrossRef](#)]
51. Yang, X.; Wang, X. Kolmogorov-Arnold Transformer. *arXiv* **2024**, arXiv:2409.10594. [[PubMed](#)]
52. Li, C.; Liu, X.; Li, W.; Wang, C.; Liu, H.; Liu, Y.; Chen, Z.; Yuan, Y. U-KAN Makes Strong Backbone for Medical Image Segmentation and Generation. *arXiv* **2024**, arXiv:2406.02918. [[CrossRef](#)]
53. SS, S.; AR, K.; R, G.; KP, A. Chebyshev Polynomial-Based Kolmogorov-Arnold Networks: An Efficient Architecture for Nonlinear Function Approximation. *arXiv* **2024**, arXiv:2405.07200.
54. Ferdaus, M.M.; Abdelguerfi, M.; Ioup, E.; Dobson, D.; Niles, K.N.; Pathak, K.; Sloan, S. KANICE: Kolmogorov-Arnold Networks with Interactive Convolutional Elements. *arXiv* **2024**, arXiv:2410.17172. [[CrossRef](#)]
55. Ta, H.-T.; Thai, D.-Q.; Tran, A.; Sidorov, G.; Gelbukh, A. PRKAN: Parameter-Reduced Kolmogorov-Arnold Networks. *arXiv* **2025**, arXiv:2501.07032.
56. Patel, S.B.; Goh, V.; FitzGerald, J.F.; Antoniadis, C.A. 2D and 3D Deep Learning Models for MRI-based Parkinson's Disease Classification: A Comparative Analysis of Convolutional Kolmogorov-Arnold Networks, Convolutional Neural Networks, and Graph Convolutional Networks. *arXiv* **2024**, arXiv:2407.17380.
57. Bodner, A.D.; Tepsich, A.S.; Spolski, J.N.; Pourceau, S. Convolutional Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2406.13155. [[PubMed](#)]
58. Reinhardt, E.A.F.; Dinesh, P.R.; Gleyzer, S. SineKAN: Kolmogorov-Arnold Networks Using Sinusoidal Activation Functions. *Front. Artif. Intell.* **2025**, *7*, 1462952. [[CrossRef](#)] [[PubMed](#)]
59. Bozorgasl, Z.; Chen, H. Wav-KAN: Wavelet Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2405.12832. [[CrossRef](#)]
60. Wang, Y.; Yu, X.; Gao, Y.; Sha, J.; Wang, J.; Gao, L.; Zhang, Y.; Rong, X. SpectralKAN: Kolmogorov-Arnold Network for Hyperspectral Images Change Detection. *arXiv* **2024**, arXiv:2407.00949.

61. Al-qaness, M.A.A.; Ni, S. TCNN-KAN: Optimized CNN by Kolmogorov-Arnold Network and Pruning Techniques for sEMG Gesture Recognition. *IEEE J. Biomed. Health Inform.* **2025**, *29*, 188–197. [[CrossRef](#)] [[PubMed](#)]
62. Jiang, C.; Li, Y.; Luo, H.; Zhang, C.; Du, H. KANNet: Kolmogorov–Arnold Networks and multi slice partition channel priority attention in convolutional neural network for lung nodule detection. *Biomed. Signal Process. Control* **2025**, *103*, 107358. [[CrossRef](#)]
63. Jahin, M.A.; Masud, M.A.; Mridha, M.F.; Aung, Z.; Dey, N. KACQ-DCNN: Uncertainty-Aware Interpretable Kolmogorov-Arnold Classical-Quantum Dual-Channel Neural Network for Heart Disease Detection. *arXiv* **2024**, arXiv:2410.07446. [[CrossRef](#)]
64. Liu, T.; Xiao, H.; Sun, Y.; Zuo, K.; Li, Z.; Yang, Z.; Liu, S. PhysKANNet: A KAN-based model for multiscale feature extraction and contextual fusion in remote physiological measurement. *Biomed. Signal Process. Control* **2025**, *100*, 107111. [[CrossRef](#)]
65. Abd Elaziz, M.; Fares, I.A.; Aseeri, A.O. CKAN: Convolutional Kolmogorov–Arnold Networks Model for Intrusion Detection in IoT Environment. *IEEE Access* **2024**, *12*, 134837–134851. [[CrossRef](#)]
66. Batreddy, S.; Pamisetty, G.; Kothuri, M.; Verma, P.; Babu, C.S. Adaptive Neighborhood Sampling and KAN-Based Edge Features for Enhanced Fraud Detection. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 15–18 December 2024; pp. 3375–3381.
67. Danish, M.U.; Grolinger, K. Kolmogorov–Arnold recurrent network for short term load forecasting across diverse consumers. *Energy Rep.* **2025**, *13*, 713–727. [[CrossRef](#)]
68. Shuai, H.; Li, F. Physics-Informed Kolmogorov-Arnold Networks for Power System Dynamics. *IEEE Open Access J. Power Energy* **2025**, *12*, 46–58. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.