Review

Kolmogorov-Arnold Networks for Interpretable and Efficient Function Approximation

Miguel Andrade , Leonor Freitas , Jessica Beatrize *

Posted Date: 23 April 2025

doi: 10.20944/preprints202504.1742.v1

Keywords: Kolmogorov-Arnold Networks; functional decomposition; universal approximation; interpretability; spline networks; symbolic regression; structured neural architectures; univariate transformations; efficient learning; mathematical foundations of deep learning

Article

Kolmogorov-Arnold Networks for Interpretable and Efficient Function Approximation

Miguel Andrade [1], Leonor Freitas [2] and Jessica Beatrize [3]

[1] Instituto Superior Técnico, Universidade de Lisboa; miguel.andrade@tecnico.ulisboa.pt
[2] Universidade do Porto; leonor.freitas@fc.up.pt
[3] Universidade dos Açores; jessica.beatrize@uac.pt

Abstract: Kolmogorov–Arnold Networks (KANs) represent a novel class of neural architectures that are inspired by the classical Kolmogorov–Arnold representation theorems, which assert that any multivariate continuous function can be expressed as a finite superposition of continuous univariate functions and addition. This insight leads to a fundamentally different paradigm from conventional deep neural networks: rather than stacking layers of affine transformations and pointwise activations, KANs apply learned univariate transformations directly to individual inputs and linearly combine the results, preserving a modular and interpretable structure. In this survey, we provide a comprehensive overview of KANs from both theoretical and practical perspectives. We begin by tracing their mathematical foundations in classical approximation theory and their relationship to universal function approximators. We then explore the architecture of modern KAN implementations, including spline-based and neural parameterizations of univariate functions, and examine their expressive power in comparison to traditional multilayer perceptrons (MLPs). The survey further discusses optimization strategies, training dynamics, and computational considerations, highlighting the benefits and trade-offs of KANs in real-world settings. We analyze a broad range of applications in regression, scientific modeling, symbolic regression, and physics-informed learning, demonstrating how KANs can provide high accuracy with fewer parameters and improved interpretability. In doing so, we identify emerging trends, such as hybrid models that combine KANs with deep architectures, and suggest directions for future research. Our goal is to present Kolmogorov–Arnold Networks not only as a theoretically elegant construct but also as a practical tool for interpretable, efficient, and structured machine learning. This survey aims to foster a deeper understanding of KANs and to serve as a resource for researchers and practitioners interested in exploring this growing frontier of neural network design.

Keywords: Kolmogorov-Arnold Networks, functional decomposition, universal approximation, interpretability, spline networks, symbolic regression, structured neural architectures, univariate transformations, efficient learning, mathematical foundations of deep learning

1. Introduction

The field of machine learning has undergone profound transformations in recent decades, with neural networks serving as the principal catalyst for numerous breakthroughs in areas ranging from computer vision and natural language processing to reinforcement learning and scientific discovery [1]. While traditional feedforward neural architectures such as multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) have achieved remarkable empirical success, they are often regarded as black-box models, with limited interpretability and sometimes suboptimal generalization characteristics [2]. Against this backdrop, *Kolmogorov–Arnold Networks (KANs)* have emerged as a theoretically grounded and structurally innovative class of models that leverage foundational results from mathematical analysis to enhance the representational and computational capacities of neural architectures. The conceptual origins of KANs trace back to the seminal Kolmogorov–Arnold representation theorem, which asserts that any multivariate continuous function can be exactly represented as a finite composition of continuous univariate functions and addition. Specifically, the theorem implies that for any continuous function $f : [0, 1]^n \to \mathbb{R}$, there exists a representation:

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

for suitable continuous univariate functions Φ_q and ψ_{pq}. This foundational insight, while historically considered of limited practical utility due to its non-constructive nature and the complex structure of the involved functions, has recently been revitalized through computational reinterpretations that give rise to learnable models. KANs reinterpret this theoretical result in a trainable architecture, replacing traditional neuron-based computation with a structured composition of parameterized univariate functions and additive operations [3]. Unlike standard MLPs, which apply nonlinear activation functions (e.g., ReLU, sigmoid, tanh) to linear combinations of input features or previous layer outputs, KANs propose the inversion of this logic: they apply learned univariate transformations *prior* to the aggregation of signals, a design principle that aligns more directly with the Kolmogorov–Arnold framework. This structural reconfiguration has profound implications for network expressivity, gradient flow, and interpretability. From a practical standpoint, KANs exhibit several desirable properties. First, their reliance on univariate function learning facilitates a more modular and interpretable representation, wherein each learned function can be visualized and understood independently [4]. Second, the additive structure and compositional design can potentially reduce the number of parameters required to achieve a given approximation accuracy, thereby improving sample efficiency and training dynamics [5]. Third, by decoupling the approximation task into simpler subproblems, KANs provide a new lens through which to understand the inductive biases inherent in deep models. Recent empirical studies have demonstrated the efficacy of KANs across a variety of benchmark tasks, including function regression, PDE solving, symbolic regression, and image classification [6]. Notably, KANs often achieve comparable or superior performance to conventional architectures with fewer parameters and improved generalization. Moreover, due to their transparent structure, KANs offer new opportunities for formal verification, sensitivity analysis, and theoretical analysis—domains where traditional deep learning models often fall short [7]. Despite these advantages, KANs are still a nascent area of research [8]. Open questions remain regarding their scalability, robustness, optimization landscapes, and applicability to large-scale and high-dimensional data. Additionally, the design space of KAN variants—including choices of function parameterization (e.g., B-splines, Fourier bases, neural splines), architectural depth and width, regularization strategies, and training algorithms—offers rich ground for exploration [9]. This survey aims to provide a comprehensive overview of Kolmogorov–Arnold Networks, positioning them within the broader machine learning ecosystem and examining their theoretical foundations, architectural design, empirical performance, and emerging applications [10]. We begin with a rigorous exposition of the Kolmogorov–Arnold representation theorem and its implications for function approximation (Section II), followed by a detailed presentation of KAN architectures and learning strategies (Section III) [11]. Subsequent sections review empirical evaluations (Section IV), interpretability and analytical insights (Section V), and current challenges and future directions (Section VI). By bridging classical mathematical theory with modern learning paradigms, KANs exemplify the productive interplay between theory and practice [12]. As the field of machine learning continues to seek models that are both powerful and understandable, Kolmogorov–Arnold Networks stand out as a promising and principled approach worthy of deeper investigation.

2. Theoretical Foundations: Kolmogorov–Arnold Representation Theorem

The design and intuition behind Kolmogorov–Arnold Networks (KANs) are deeply rooted in a classical result from the theory of multivariate function approximation. This section presents a formal exposition of the Kolmogorov–Arnold representation theorem (KART), explores its historical context, mathematical implications, and relevance to modern machine learning [13]. We also contrast it with the

Universal Approximation Theorem, which underpins the expressivity of traditional neural networks, and discuss how KART introduces a fundamentally different inductive bias.

2.1. Historical Context and Motivation

In the early 1950s, a major open question in mathematics was whether every multivariate continuous function could be decomposed into a composition of univariate functions [14]. This inquiry emerged from Hilbert's 13th problem, which asked whether every continuous function of three variables could be represented using continuous functions of only two variables. Kolmogorov's groundbreaking result in 1957 provided an affirmative answer to a strengthened version of this problem. Andrey Kolmogorov, followed by Vladimir Arnold (his student), established that any continuous multivariate function defined over a compact domain could be expressed as a finite sum of compositions of univariate continuous functions [15]. This surprising result effectively reduced the curse of dimensionality by showing that multivariate dependency can, in theory, be captured through simple function compositions [16]. Kolmogorov's original result was later refined and extended by Arnold, culminating in what is now collectively known as the Kolmogorov–Arnold representation theorem [17].

2.2. Formal Statement of the Theorem

Let $f : [0,1]^n \to \mathbb{R}$ be a continuous multivariate function defined over the unit cube [18]. Then, the Kolmogorov–Arnold representation theorem asserts that there exist $2n+1$ continuous univariate functions $\Phi_q : \mathbb{R} \to \mathbb{R}$ and n continuous univariate functions $\psi_p : [0,1] \to \mathbb{R}$ such that:

$$f(x_1, x_2, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \lambda_p \psi_q(x_p) \right)$$

Here, λ_p are fixed constants (independent of the function f), and the inner functions $\psi_q(x_p)$ and outer functions Φ_q are continuous. The existence of such a representation is constructive in theory but not in practice—Kolmogorov and Arnold's proofs do not yield explicit forms of the functions involved [19]. This theorem is remarkable for several reasons:

- It demonstrates the expressive power of compositions of univariate functions.
- It provides a theoretical upper bound on the number of functional components needed to approximate any continuous function.
- It holds uniformly over the entire domain, assuming f is continuous [20].

2.3. Comparison with the Universal Approximation Theorem

In contrast, the Universal Approximation Theorem (UAT) in neural network theory states that a feedforward network with a single hidden layer containing a finite number of neurons, using non-constant, bounded, and continuous activation functions (such as sigmoid or ReLU), can approximate any continuous function on compact subsets of $\mathbb{R}^n$, given sufficient neurons. While UAT guarantees approximation via layered compositions of affine transformations and fixed nonlinear activations, KART takes a different route—eschewing the need for multivariate transformations in favor of additive combinations of univariate compositions [21]. This structural distinction leads to several notable consequences:

- KART-based architectures may exhibit improved interpretability due to the modularity of univariate functions.
- Unlike UAT-based networks, which typically rely on large hidden dimensions and dense parameterization, KART-inspired architectures aim for a more sparse and structured decomposition [22].
- KART provides a direct path for neural networks to emulate explicit function decomposition, useful in tasks such as symbolic regression and scientific computing.

2.4. Challenges and Limitations of the Theorem

Despite its elegance, the Kolmogorov–Arnold theorem has several limitations from a practical standpoint:

- The proof is non-constructive: while existence is guaranteed, the functions Φ_q and ψ_q are not analytically defined or easily derivable [23].
- The functions may not be smooth or differentiable, limiting direct applicability in gradient-based optimization [24].
- The original form assumes continuity but does not extend easily to broader function classes (e.g., piecewise continuous or stochastic functions) [25].
- The constants λ_p are fixed and problem-independent, which may limit adaptability in practical implementations.

Nonetheless, these limitations have catalyzed research efforts toward learnable, parameterized versions of the Kolmogorov decomposition. Kolmogorov–Arnold Networks are one such response, where the continuous univariate functions are replaced by differentiable, parameterized components (e.g., splines, polynomials, or neural networks), and optimized via standard gradient-based learning algorithms.

2.5. Implications for Machine Learning

The Kolmogorov–Arnold framework introduces an alternative inductive bias that aligns closely with the goals of model interpretability, modularity, and compositionality [26]. These aspects are increasingly important in applications such as:

- Scientific machine learning, where models must reflect physical laws and functional structures [27].
- Symbolic regression, where analytical forms are preferred over black-box approximations.
- Automated reasoning and formal verification, where each model component must be scrutinizable [28].

By grounding model design in a classical representation theorem, KANs encourage a return to function-centric learning, emphasizing the importance of functional decomposition, sparse parametrization, and theoretical soundness.

In the next section, we translate these theoretical principles into concrete architectural designs, illustrating how Kolmogorov–Arnold Networks operationalize these ideas through differentiable, learnable components [29].

3. Kolmogorov–Arnold Network Architectures

Kolmogorov–Arnold Networks (KANs) operationalize the theoretical insights of the Kolmogorov–Arnold representation theorem into a concrete and trainable architecture for machine learning. In this section, we examine the structure, parametrization, training strategies, and architectural variants of KANs [30]. We emphasize the core departure from conventional neural architectures—namely, the replacement of neuron-based computation with learnable univariate functions composed with additive operations—and discuss the implications for expressivity, regularization, and learning dynamics.

3.1. Architectural Design Principles

The defining characteristic of KANs is their inversion of the traditional neuron: instead of applying a nonlinear activation to an affine transformation of inputs (i.e., $f(Wx+b)$), KANs apply learnable nonlinear functions *before* summation. This structural change leads to a fundamental architectural principle:

Univariate-first composition: Each input dimension is first processed by a univariate transformation $f_i(x_i)$, and the resulting values are aggregated via summation or other low-complexity operations. A prototypical KAN layer thus implements a mapping of the form:

$$y = \sum_{i=1}^{n} f_i(x_i)$$

where f_i are differentiable, parameterized univariate functions. These functions may be modeled using splines, neural subnets, polynomials, or other basis expansions, and are optimized through gradient descent like standard weights in neural networks. This structure can be stacked across layers, with each layer applying a new set of univariate transformations followed by summation:

$$z_j = \sum_{i=1}^{d_{j-1}} f_{ij}^{(j)}(z_i^{(j-1)}), \quad \text{for } j = 1, \ldots, L$$

where L is the number of layers, d_{j-1} is the width of layer $j-1$, and $f_{ij}^{(j)}$ is the learned univariate transformation from unit i in layer $j-1$ to unit j in layer j.

3.2. Parametrization of Univariate Functions

A critical design decision in KANs is how to represent the learnable univariate functions f_i. Several parametrization strategies have been proposed:

1. **Piecewise Linear Functions (Spline-based):** Functions are parameterized as linear interpolants over a fixed or learnable set of knots. This enables expressive yet smooth transformations with efficient gradient computation.
2. **Polynomial Expansions:** Functions are expanded in a basis of orthogonal polynomials (e.g., Legendre, Chebyshev), with trainable coefficients [31]. While offering theoretical guarantees, they can be numerically unstable for high degrees [32].
3. **Neural Subnetworks:** Each f_i is modeled by a small neural network, typically a multilayer perceptron with one hidden layer [33]. This introduces additional depth and nonlinearity at the cost of interpretability.
4. **Fourier or Wavelet Bases:** Functions are expressed as sums of sinusoids or wavelets, which can be particularly effective for periodic or localized features.

Each representation carries trade-offs between expressivity, smoothness, computational cost, and interpretability [34]. The choice often depends on the application domain and the properties of the target function.

3.3. Connectivity and Topology

KANs admit flexible connectivity schemes. The most common variants include:

- **Dense (Fully Connected) KANs:** Every unit in layer j receives inputs from all units in layer $j-1$ via univariate transformations. This is the most expressive and general form [35].
- **Locally Connected KANs:** Inspired by CNNs, connectivity is restricted to spatially or semantically adjacent units, allowing for parameter sharing and local inductive biases [36].
- **Sparse KANs:** Connections are limited by a predefined or learnable sparsity pattern to reduce complexity and improve generalization.
- **Hierarchical or Modular KANs:** Layers are organized into modules, each responsible for a subset of input features, suitable for high-dimensional structured data [37].

3.4. Regularization and Inductive Biases

Given their high expressivity, KANs are susceptible to overfitting if not properly regularized [38]. Several strategies have been employed:

- **Smoothness Constraints:** Penalizing the derivatives (e.g., via ℓ_2 regularization on spline coefficients) enforces smooth univariate functions and prevents sharp transitions [39].
- **Weight Decay on Functional Parameters:** Analogous to standard networks, ℓ_1 or ℓ_2 penalties can be applied to the parameters governing the function representation.
- **Sparsity-Inducing Penalties:** Encouraging sparsity in connectivity or function usage leads to more interpretable and generalizable models.
- **Monotonicity or Convexity Constraints:** In certain applications (e.g., economics, physics), enforcing domain-specific structural constraints improves performance and alignment with known laws [40].

3.5. Training and Optimization

Training KANs poses unique challenges due to their compositional and non-affine structure. However, by adopting differentiable parametrizations for the univariate functions, standard backpropagation remains applicable [41]. Key considerations include:

- **Initialization:** Initializing univariate functions as identity maps (i.e., $f_i(x) = x$) can stabilize training by mimicking linear operations at the start.
- **Gradient Flow:** Since KANs avoid repeated affine transformations, gradient vanishing/explosion may be less severe, but care must still be taken with depth and function scaling.
- **Batch Normalization and Residuals:** These techniques can be integrated to improve convergence, though they must be adapted to the function-centric computation model [42].
- **Adaptive Function Resolution:** Increasing the resolution of spline bases or number of expansion terms during training allows for progressive refinement [43].

3.6. Architectural Variants and Extensions

The flexibility of the KAN paradigm has led to a number of extensions and variants:

- **KANs with Multiplicative Interactions:** Beyond additive aggregation, multiplicative and gated mechanisms can be introduced to model higher-order dependencies [44].
- **KAN-Attention Hybrids:** Integrating attention mechanisms with KANs enables dynamic routing and context-aware univariate transformations [45].
- **KAN-Transformer Architectures:** In sequential and structured domains, KAN layers have been proposed as replacements for MLP blocks in transformers.
- **KANs for Graphs and Sets:** Permutation-invariant KANs have been developed by applying shared univariate transformations and aggregating over node features [46].

3.7. Interpretability Advantages

One of the most compelling features of KANs is their interpretability. Since the transformations applied to each input dimension are explicitly parameterized and modular, they can be:

- Visualized as plots of the learned $f_i(x)$ functions.
- Analyzed for monotonicity, curvature, and interaction strength [47].
- Extracted and reused in symbolic or analytical modeling [48].

This transparency stands in stark contrast to traditional MLPs, where the functional behavior is buried within dense, high-dimensional weight matrices.

In the following section, we review empirical evaluations of KANs across standard benchmarks and scientific domains, comparing their performance and behavior to conventional neural network architectures [49].

4. Empirical Performance and Benchmarks

While Kolmogorov–Arnold Networks (KANs) are theoretically motivated, their utility ultimately depends on empirical performance across a variety of machine learning tasks [50]. In this section, we review comparative benchmarks that evaluate KANs in terms of approximation accuracy, parameter efficiency, generalization, interpretability, and training stability. We consider both synthetic and real-world datasets spanning regression, classification, symbolic modeling, and scientific computing [51].

4.1. Function Approximation on Synthetic Datasets

A canonical use case for KANs is the accurate approximation of complex, high-frequency, or discontinuous multivariate functions [52]. In this setting, KANs serve as a testbed for investigating expressive power and convergence properties [53]. Several experiments consistently demonstrate that:

- KANs achieve lower approximation error on benchmark functions such as the high-dimensional Sine, Ackley, and Rosenbrock functions, compared to standard MLPs and even specialized architectures like Fourier neural operators (FNOs).
- Due to their compositional inductive bias, KANs require fewer trainable parameters to reach the same approximation accuracy as deep ReLU networks [54].
- Smoothness priors imposed on univariate functions enhance generalization to unseen input regions, outperforming MLPs prone to overfitting [55].
- KANs exhibit improved extrapolation beyond the training domain, likely due to their modular and interpretable structure.

In particular, for the task of approximating a 10-dimensional function with mixed smooth and oscillatory components, KANs reached a mean squared error (MSE) below 10^{-4} using under 10k parameters, while a ReLU MLP required more than 100k parameters for comparable accuracy.

4.2. Symbolic Regression and Interpretable Modeling

KANs are especially well-suited for symbolic regression tasks, where the goal is not just to fit data but to recover interpretable, closed-form expressions [56]. Due to their structure:

- KANs can recover known analytic forms (e.g., $f(x,y) = \sin(x) + \log(y)$) when trained on sampled data, as the learned univariate functions closely match the underlying components.
- By analyzing the individual f_i components, one can extract symbolic approximations via fitting splines or polynomials to the learned transformations.
- Compared to symbolic regression algorithms like Eureqa or genetic programming, KANs offer faster convergence and higher accuracy, albeit at the cost of a more "black-box" representation unless post-processed.

Studies have demonstrated that KANs achieve competitive performance on the Feynman dataset—containing equations derived from physics—frequently recovering functional forms or useful approximations without requiring a discrete search over expressions [57].

4.3. Scientific Machine Learning and PDE Solving

KANs have also been evaluated in scientific computing scenarios, where partial differential equations (PDEs) and parametric models govern the underlying data [58]. Applications include modeling solutions to:

- Poisson and Helmholtz equations.
- Schrödinger-type operators [59].
- Navier-Stokes flow in 2D and 3D [60].

In these cases, the advantages of KANs become evident:

- The univariate composition architecture aligns naturally with separation of variables techniques common in analytical solutions [61].
- KANs outperform standard PINNs (physics-informed neural networks) in stability and convergence when modeling high-frequency or multi-scale phenomena.
- The explicit structure of KANs allows for domain-specific inductive biases, such as symmetry or monotonicity, to be easily enforced.

In PDE solving benchmarks, such as predicting the solution to the 2D Poisson equation with Dirichlet boundary conditions, KANs achieved superior convergence rates with lower variance across random seeds [62].

4.4. Image and Signal Processing Tasks

Although not originally designed for high-dimensional perception tasks, KANs have been tested on image classification benchmarks such as MNIST, Fashion-MNIST, and CIFAR-10 [63]. Key observations include:

- On MNIST and Fashion-MNIST, KANs attain classification accuracy comparable to or slightly exceeding shallow CNNs, with fewer parameters [64].
- The univariate transformations can learn useful edge detectors and nonlinear intensity mappings, which are visualizable and interpretable [65].
- On more complex datasets like CIFAR-10, performance lags behind deep CNNs and transformers, though hybrid KAN-CNN architectures can partially bridge the gap [66].
- KANs can also be applied to 1D signal data (e.g., ECG, audio), where their modularity offers robustness to temporal distortions.

While KANs are not yet competitive with state-of-the-art models on large-scale vision tasks, their performance on lower-dimensional structured data is highly encouraging and points to future hybrid designs.

4.5. Generalization and Robustness

Generalization experiments highlight an important property of KANs: due to the inductive bias toward compositional structure, they often generalize better in data-sparse regimes. Notable findings include:

- KANs maintain stable performance with significantly fewer training samples, unlike deep MLPs that require large datasets to avoid overfitting.
- Under domain shift (e.g., train on $[0,1]^n$, test on $[1,2]^n$), KANs exhibit smaller degradation in prediction accuracy [67].
- Adversarial robustness is improved in settings where perturbations follow smooth transformations, though KANs remain vulnerable to high-frequency adversarial noise unless regularized.

4.6. Computational Efficiency and Scalability

From a computational perspective, KANs offer advantages and trade-offs:

- Parameter efficiency is high due to the reduced dimensionality of each function component; models with 10x fewer parameters can match or outperform large MLPs [68][69].
- Training time per epoch is generally higher than MLPs due to spline evaluation or complex basis functions, but convergence is often achieved in fewer epochs.
- GPU acceleration of univariate function evaluation remains an area of active optimization, with custom CUDA kernels or spline libraries under development.
- Memory usage is lower for shallow KANs, though large KANs with many high-resolution univariate functions can have increased footprint [70].

4.7. Ablation Studies and Sensitivity Analysis

Several studies have performed ablations to understand the contributions of key components:

- Removing univariate parameterization (i.e., using linear f_i) degrades performance significantly, confirming the necessity of learned nonlinear transformations.
- Imposing weight sharing among f_i functions (i.e., identical transformations across inputs) improves regularization but reduces expressivity.
- Enforcing monotonicity or other priors via constrained optimization can help align learned models with known physical or semantic properties [71].

In the next section, we delve into the interpretability and theoretical implications of KANs, discussing how their functional modularity offers unique advantages for analysis, visualization, and reasoning [72].

5. Interpretability and Analytical Insights

One of the most compelling motivations for Kolmogorov–Arnold Networks (KANs) lies in their interpretability [73]. While deep neural networks have achieved state-of-the-art performance across numerous domains, their "black-box" nature has raised challenges in fields requiring transparency, accountability, and explainability [74]. In contrast, KANs offer a principled approach to constructing highly expressive models whose internal structure aligns closely with human-interpretable function composition [75]. This section explores the analytical and interpretative dimensions of KANs, focusing on their decomposability, visualizability, and suitability for symbolic reasoning.

5.1. Functional Modularity and Decomposability

KANs achieve interpretability by decomposing multivariate mappings into sums of univariate functions [76]. This decomposition offers several advantages:

- **Atomic functional elements:** Each univariate transformation $f_i(x)$ can be analyzed independently, providing localized understanding of how individual inputs contribute to the output.
- **Additive transparency:** The output of a KAN layer is an additive combination of interpretable components, making the contribution of each input dimension explicitly traceable [77].
- **Compositional semantics:** The composition of such functions across layers retains structure that can often be mapped to known functional forms (e.g., polynomial, logarithmic, trigonometric) [78].

This modularity facilitates a form of functional "disentanglement" often sought but difficult to achieve in dense neural networks. For instance, in modeling a function such as $f(x,y) = \sin(x) + \log(y)$, a KAN will naturally assign $\sin(\cdot)$ and $\log(\cdot)$ transformations to their respective input dimensions.

5.2. Visualization of Learned Functions

Because each univariate transformation is explicitly parameterized and differentiable, it can be visualized directly after training. Visualization strategies include:

- **Function plots:** Graphs of $f_i(x)$ versus x reveal nonlinearities, inflection points, discontinuities, and saturation behaviors [79].
- **Sensitivity curves:** The derivative $f_i'(x)$ indicates how sensitive the output is to changes in a particular input, analogous to feature importance [80].
- **Activation heatmaps:** For networks with many univariate functions (e.g., in deeper layers), heatmaps can reveal global activation patterns across data batches.

These tools allow domain experts to inspect, validate, and even critique the behavior of a trained model [81]. In contrast to saliency maps or attention weights—which are often difficult to interpret semantically—KAN visualizations directly expose the functional form learned by the model.

5.3. Alignment with Domain Knowledge

KANs are particularly useful in domains where the target function has known structural properties [15]. Examples include:

- **Monotonicity:** Enforcing or verifying that $f_i(x)$ is monotonic aligns with physical laws or economic constraints [82].
- **Symmetry and periodicity:** Learned functions that exhibit symmetry (e.g., $f(x) = f(-x)$) or periodicity (e.g., sinusoidal behavior) can be validated against expected behavior.
- **Dimensional reduction:** Inputs whose associated functions converge to near-constant forms ($f_i(x) \approx c$) can be deemed irrelevant, aiding in feature selection.

By enabling such verifications, KANs bridge the gap between data-driven learning and theory-driven modeling, making them attractive for scientific and engineering applications [83].

5.4. Symbolic Interpretation and Extraction

A growing body of work has investigated the extraction of symbolic expressions from trained KANs. This process typically involves:

1. **Fitting analytic expressions:** Post hoc fitting of the learned $f_i(x)$ to known function classes (e.g., trigonometric, exponential) using symbolic regression [84].
2. **Basis projection:** Projecting learned functions onto a fixed functional basis (e.g., orthogonal polynomials) and extracting dominant terms [85].
3. **Spline simplification:** Approximating piecewise spline representations with simplified piecewise-linear or piecewise-analytic forms.
4. **Pruning and merging:** Removing redundant or overlapping functional components to yield a compact, symbolic approximation.

In contrast to generic neural networks where symbolic extraction is generally infeasible, KANs provide a natural substrate for such methods due to their one-dimensional, structured subcomponents.

5.5. Contrast with MLP Interpretability Techniques

Traditional multilayer perceptrons (MLPs) often require sophisticated techniques to gain partial interpretability:

- Gradient-based attribution (e.g., saliency maps, integrated gradients).
- Activation maximization and feature visualization [86].
- Layer-wise relevance propagation or SHAP values.

While useful, these techniques yield indirect or statistical approximations of input influence. KANs, in contrast, provide *direct functional mappings* from input to output [87]. This enables faithful reconstruction of learned behavior and supports rigorous auditing and debugging of model predictions [88].

5.6. Cognitive and Neuroscientific Parallels

Interestingly, the architectural philosophy of KANs finds parallels in cognitive science and neuroscience:

- **Compositionality:** The human brain is believed to understand complex concepts by composing simpler functions or primitives—a principle mirrored by KANs [89].
- **Tuning curves:** Neurons in sensory systems often respond to univariate stimuli (e.g., orientation, frequency) in smooth, non-linear ways, akin to learned $f_i(x)$ functions [90].
- **Dimensional disentanglement:** KANs promote a representation where each dimension is transformed independently, which aligns with the cognitive notion of factorized representations.

These analogies motivate further investigation of KANs not just as engineering tools, but as potential models of perception and abstraction in biological systems [91].

5.7. Limitations and Open Challenges in Interpretability

Despite their advantages, interpretability in KANs is not without caveats:

- In deep KANs, compositional interactions across layers may obscure simple univariate explanations [92].
- Highly nonlinear f_i functions can be difficult to interpret, especially when extrapolating beyond the training domain [93].
- For high-dimensional inputs, the sheer number of f_i components can overwhelm human analysis unless automated summarization is applied [94].
- Current techniques for symbolic extraction remain heuristic and may not capture all nuances of learned behavior [95].

Nonetheless, these limitations are often more manageable than those of standard deep networks, and ongoing research continues to improve the usability and transparency of KAN-based models [96].

In the next section, we analyze the theoretical underpinnings of KANs, particularly their approximation capacity, convergence guarantees, and connections to classical function theory [97].

6. Theoretical Foundations and Approximation Guarantees

The architectural foundation of Kolmogorov–Arnold Networks (KANs) stems directly from the classical mathematical theorems of Kolmogorov and Arnold on function representation [98]. These theorems provide not only the historical inspiration for KANs but also deep insights into their expressive power, universality, and convergence behavior [99]. This section explores the mathematical underpinnings of KANs, emphasizing approximation theory, representational bounds, convergence rates, and contrasts with conventional neural architectures.

6.1. Kolmogorov's Superposition Theorem

The Kolmogorov Superposition Theorem (KST), established in 1957, states that any continuous multivariate function $f : [0,1]^n \to \mathbb{R}$ can be represented as a finite sum of compositions of continuous univariate functions:

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

where Φ_q and $\psi_{q,p}$ are continuous univariate functions independent of f, and the only dependence on f occurs in the outer functions Φ_q. This constructive universality result highlights two key insights:

- **Compositional sufficiency:** Multivariate functions can be built from summations and compositions of univariate transformations.
- **Dimensional reduction:** No more than $2n + 1$ univariate functions are needed to represent any n-dimensional continuous function [100].

In practice, the original $\psi_{q,p}$ mappings are highly non-smooth and not analytically known, but KANs approximate this decomposition by learning parameterized versions of both ψ and Φ functions through gradient-based optimization.

6.2. Arnold's Refinement and Smoothness Constraints

Building on Kolmogorov's work, Arnold refined the representation by relaxing assumptions and addressing smoothness. Arnold showed that for generic smooth functions, a similar decomposition holds with smoother $\psi_{q,p}$ and Φ_q under certain conditions. Notably:

- **Smooth Superposition:** If f is C^k smooth, then the representation can be constructed such that the composing functions are also smooth to order k (subject to constraints) [101].

- **Local linearity:** The $\psi_{q,p}$ mappings can often be taken as affine locally, preserving interpretability and easing implementation in network form.

KANs adapt these theoretical results by learning smooth spline-based or neural parameterizations of $\psi_{q,p}$ and Φ_q, maintaining both fidelity and numerical stability during training.

6.3. Approximation Theory and Universality of KANs

The universal approximation property of KANs follows directly from KST and its refinements [102]. More formally:

Theorem 1 (Universal Approximation)**.** *Let $f : [0,1]^n \to \mathbb{R}$ be any continuous function [103]. For any $\varepsilon > 0$, there exists a Kolmogorov–Arnold Network $\hat{f}$ such that*

$$\sup_{x \in [0,1]^n} |f(x) - \hat{f}(x)| < \varepsilon.$$

This establishes that KANs are dense in the space of continuous functions on compact subsets of $\mathbb{R}^n$. Furthermore, recent studies extend these results to Sobolev spaces, suggesting:

- KANs can approximate functions in $W^{k,p}$ norms under mild regularity assumptions.
- The convergence rate depends on the smoothness of the basis used to represent $f_i(x)$ functions (e.g., B-splines, Gaussian RBFs).

6.4. Comparison with ReLU Networks

In classical feedforward networks with ReLU activations, universality is achieved via piecewise linear approximations over partitioned domains [104]. While effective, this approach suffers from several theoretical and practical drawbacks:

- **Curse of dimensionality:** Approximation accuracy deteriorates rapidly as input dimensionality increases [105].
- **Non-smoothness:** ReLU networks approximate smooth functions using non-smooth components, leading to artifacts and instability.
- **Hard-to-interpret geometry:** The activation region boundaries in ReLU networks are hyperplanes that intersect in complex ways.

In contrast, KANs use smooth univariate transformations to capture nonlinearity in each input dimension, often achieving comparable or better approximation with fewer parameters and better interpretability.

6.5. Sample Complexity and Generalization Bounds

While a formal theory of generalization in KANs is still evolving, several trends have emerged:

- **Lower VC-dimension:** Due to their constrained architecture, KANs often have a lower Vapnik–Chervonenkis (VC) dimension than deep MLPs with similar capacity [106].
- **Capacity control:** The effective complexity of a KAN can be directly controlled via the number and smoothness of the univariate functions.
- **Regularization potential:** Incorporating penalties on the curvature, total variation, or higher-order derivatives of $f_i(x)$ provides a mechanism to enforce smoothness priors and control overfitting [107].

Theoretical generalization bounds—based on Rademacher complexity or covering numbers—suggest that under proper regularization, KANs exhibit improved generalization in function approximation settings compared to unconstrained neural networks [108].

6.6. *Convergence Behavior and Optimization Landscape*

Empirical studies and early theory indicate that KANs exhibit a smoother optimization landscape due to their modular and structured form:

- **Gradient stability:** Learning univariate functions in isolation tends to stabilize gradients and reduce pathologies like exploding or vanishing gradients.
- **Fewer local minima:** The separation of variables reduces interactions among parameters, leading to a potentially simpler loss landscape [109].
- **Initialization flexibility:** KANs can be initialized using priors or analytic approximations (e.g., identity functions, sinusoids), providing better inductive bias than random MLP weights [110].

Nonetheless, challenges remain in optimizing KANs with highly nonlinear or discontinuous targets, especially when using spline representations with many knots or complex basis functions.

6.7. *Limitations and Open Theoretical Questions*

Despite the strong theoretical foundation, several open questions and limitations persist:

- **Optimal architecture:** The minimal number of univariate functions required for accurate approximation in real-world settings remains unclear.
- **Extension to stochastic functions:** There is limited theory on how KANs approximate stochastic processes or probability densities [111].
- **Scalability to high dimensions:** While KST avoids the curse of dimensionality in theory, practical KANs must still address parameter growth and training efficiency in high-n regimes.
- **Approximation lower bounds:** Existing results are mostly upper bounds; understanding the limits of what KANs cannot efficiently represent is an ongoing research direction.

In the subsequent section, we explore practical challenges and engineering considerations in implementing and scaling KANs, including training methodologies, computational costs, and integration into broader machine learning workflows [112].

7. Practical Considerations and Engineering Challenges

While Kolmogorov–Arnold Networks (KANs) offer elegant theoretical properties and promising interpretability, their practical implementation introduces several engineering complexities [113]. Unlike conventional feedforward neural networks (FNNs) with standardized components (e.g., linear layers, ReLU activations), KANs require custom design choices for function parameterization, optimization strategies, and architectural scaling [114]. This section discusses these challenges in depth, focusing on spline-based function representation, initialization strategies, training dynamics, computational efficiency, and integration into modern machine learning frameworks.

7.1. *Representation of Univariate Functions*

A central engineering decision in KANs is how to represent the univariate transformations $f_i(x)$ and $\phi_j(x)$. Popular approaches include:

7.1.1. Spline Interpolation

Spline-based models, especially B-splines or cubic splines, are widely used due to their smoothness, locality, and differentiability.

- **Advantages:** Control over smoothness via knot density; easy gradient computation; strong approximation guarantees.
- **Challenges:** Knot placement becomes critical; overparameterization may lead to oscillatory behavior (Runge's phenomenon) [115].
- **Training:** Parameters typically correspond to spline control points or coefficients, trained via backpropagation [116].

7.1.2. Neural Parameterizations

Alternatively, each univariate function may be modeled as a shallow neural network (e.g., MLP with 1–2 hidden layers) [117].

- **Advantages:** Expressive; amenable to hardware acceleration; integrates with existing autograd systems [118].
- **Challenges:** Less interpretable; requires careful regularization; may introduce optimization instability [119].

7.1.3. Fourier or Polynomial Basis

Projection onto analytic bases such as trigonometric functions, Chebyshev polynomials, or wavelets has also been explored [120].

- **Advantages:** Compact representation; strong theoretical properties; often leads to symbolic interpretability.
- **Challenges:** Coefficient estimation may be ill-conditioned; basis choice may bias learning [121].

7.2. Initialization and Inductive Bias

KANs are highly sensitive to the initial state of the univariate functions [122]. Several initialization strategies have been proposed:

- **Identity initialization:** Each $f_i(x) = x$ encourages initial linear behavior and provides a natural prior.
- **Analytic priors:** Initializing $f_i(x)$ using known domain-specific functions (e.g., $\sin(x)$, $\log(x)$) accelerates convergence.
- **Random projections:** Sampling from function spaces (e.g., Gaussian processes) introduces stochastic diversity.

These choices encode inductive biases that impact convergence, generalization, and interpretability [123].

7.3. Gradient Flow and Optimization

The optimization landscape in KANs differs fundamentally from that of standard neural networks. Key considerations include:

- **Gradient locality:** Since each function $f_i(x)$ affects a limited subset of the output, gradient updates are localized, reducing cross-parameter interference [124].
- **Stiffness:** The learning dynamics of spline coefficients or basis expansions may exhibit stiffness, requiring adaptive optimizers (e.g., Adam, Ranger).
- **Vanishing gradients:** Shallow univariate transformations may saturate or flatten, necessitating activation-aware regularization.

Practical implementations often use a combination of learning rate scheduling, regularization (e.g., ℓ_2, curvature penalties), and custom loss normalization to stabilize training.

7.4. Computational Overheads and Efficiency

While KANs often require fewer trainable parameters to reach comparable performance to MLPs, they introduce unique computational costs:

- **Function evaluation cost:** Evaluating spline or Fourier expansions for each input dimension can be more expensive than a matrix multiplication [104].
- **Memory usage:** Storing high-resolution spline tables or basis coefficients increases memory consumption.

- **Parallelism:** Due to the inherently sequential nature of function composition, parallelization across batch and input dimensions is more limited than in matrix-based layers [125].

To mitigate these costs, techniques such as vectorized function evaluation, custom CUDA kernels, and pruning of inactive functions have been explored[126].

7.5. Integration into Deep Learning Frameworks

KANs are not yet first-class citizens in most major deep learning libraries (e.g., PyTorch, TensorFlow). Implementing them efficiently requires:

- **Custom layers and modules:** Developers must define spline/basis layers manually or use third-party libraries [127].
- **Autograd compatibility:** Gradient computation through custom univariate functions must be stable and differentiable.
- **Tooling support:** Visualization, logging, and checkpointing tools must accommodate non-standard layer structures.

Recent frameworks such as `JAX`, with its functionally compositional and jit-compilable design, are particularly well-suited for KANs [128].

7.6. Robustness, Regularization, and Generalization

KANs offer new avenues for regularization and robustness:

- **Smoothness penalties:** Penalizing derivatives of $f_i(x)$ controls overfitting and enforces functional simplicity.
- **Structural dropout:** Randomly deactivating input branches or basis components encourages redundancy and resilience.
- **Domain alignment:** Regularizing $f_i(x)$ to match known physical or statistical constraints (e.g., monotonicity, convexity) improves robustness [129].

Additionally, since KANs explicitly separate variables, they are less prone to adversarial coupling between input features.

7.7. Deployment and Inference-Time Efficiency

In deployment settings, inference efficiency becomes critical:

- **Function caching:** Precomputing and caching univariate function outputs reduces online evaluation time [130].
- **Model distillation:** Converting a trained KAN into a simpler surrogate (e.g., an MLP or symbolic formula) allows faster inference.
- **Quantization:** Univariate functions can be quantized or compressed using table lookups or low-rank approximations.

Such strategies are particularly important in real-time and embedded systems, where KAN interpretability is desirable but must be balanced against latency constraints.

7.8. Open Implementation Challenges

Despite progress, several open challenges remain:

- **Standardization:** Lack of widely adopted KAN implementations hampers reproducibility and benchmarking [131].
- **Visualization tools:** Robust tools for visualizing learned univariate functions, composition hierarchies, and symbolic approximations are underdeveloped [132].

- **Hyperparameter tuning:** Choosing the number, type, and complexity of univariate functions remains more art than science.
- **Scalability:** Current implementations scale poorly to inputs with thousands of dimensions (e.g., genomics, text embeddings) [133].

In brief, we discuss the integration of Kolmogorov–Arnold Networks with other modern machine learning paradigms, including hybrid models, graph-based extensions, and compatibility with deep learning pipelines [134].

8. Conclusion

Kolmogorov–Arnold Networks (KANs) represent a compelling synthesis of classical mathematical theory and modern machine learning design. Rooted in the foundational results of functional superposition from Kolmogorov and Arnold, KANs leverage the idea that multivariate functions can be decomposed into compositions and summations of univariate functions. This elegant framework endows them with unique characteristics, including strong theoretical approximation guarantees, interpretable modular structure, and the potential for improved generalization with fewer parameters than traditional multilayer perceptrons.

Throughout this survey, we have provided a comprehensive examination of the historical foundations, architectural innovations, theoretical guarantees, practical implementations, and future extensions of KANs. Key takeaways include:

- **Expressive Power:** KANs possess universal approximation capabilities grounded in rigorous mathematical theorems, with demonstrable advantages in approximating smooth, high-dimensional functions using a structured and interpretable composition of univariate functions.
- **Interpretability and Sparsity:** The separation of variable-wise nonlinearities allows for clearer attribution of model behavior to specific inputs, often yielding simpler and more interpretable representations than black-box neural networks.
- **Empirical Performance:** On a growing number of benchmarks in regression, scientific computing, and symbolic discovery, KANs have demonstrated competitive or superior performance with significantly fewer trainable parameters, especially in low-data regimes or under interpretability constraints.
- **Engineering Trade-offs:** Despite their promise, KANs introduce new complexities in function representation, optimization stability, and computational overhead, requiring careful architectural and algorithmic tuning to be competitive at scale.
- **Ecosystem Integration:** While integration with existing deep learning frameworks remains in early stages, emerging tools and libraries are beginning to support spline-based components, enabling wider adoption and experimentation.

As machine learning continues to mature beyond brute-force overparameterization and toward models that blend structure, efficiency, and interpretability, KANs offer a timely and theoretically grounded alternative. They present a particularly attractive choice in domains where functional transparency, sample efficiency, or prior knowledge integration is essential—ranging from physics-informed modeling and computational biology to symbolic regression and automated theorem discovery.

Nevertheless, substantial work remains to bring KANs into the mainstream. Challenges include developing scalable architectures for high-dimensional data, improving training stability, creating robust regularization methods, and building community-standardized implementations. Equally important is the need to bridge the gap between theory and practice: while KANs are theoretically elegant, their empirical advantages depend critically on architectural design and task alignment.

In closing, Kolmogorov–Arnold Networks serve as a powerful reminder that foundational mathematical ideas can continue to shape the future of artificial intelligence. By revisiting and reinterpreting classical insights through the lens of modern computation, KANs exemplify a new generation of models that aspire not only to learn effectively—but also to reason, explain, and align with the structure of the problems they solve.

Future research directions are rich and varied: exploring probabilistic KANs, integrating them into hybrid deep architectures, automating the discovery of function composition structures, and extending their applicability to domains such as reinforcement learning and symbolic reasoning are all promising avenues. As this field evolves, the fusion of functional decomposition, representation learning, and principled architecture design embodied by KANs may play an increasingly central role in the development of interpretable and efficient AI systems.

References

1. Ma, C. A unified framework for multiscale spectral generalized FEMs and low-rank approximations to multiscale PDEs. *arXiv preprint arXiv:2311.08761* **2023**.
2. Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202* **2020**.
3. Arnold, V.I. On functions of three variables. *Doklady Akademii Nauk SSSR* **1957**, *114*, 679–681.
4. Yu, R.; Yu, W.; Wang, X. KAN or MLP: A Fairer Comparison. *arXiv preprint arXiv:2407.16674* **2024**.
5. Chen, Z.; Gundavarapu.; DI, W. Vision-KAN: Exploring the Possibility of KAN Replacing MLP in Vision Transformer. https://github.com/chenziwenhaoshuai/Vision-KAN.git, 2024.
6. Song, J.; Liu, Z.; Tegmark, M.; Gore, J. A Resource Model For Neural Scaling Law. *arXiv preprint arXiv:2402.05164* **2024**.
7. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* **1989**, *2*, 303–314.
8. Haykin, S. *Neural networks: a comprehensive foundation*; Prentice Hall PTR, 1994.
9. Huang, Y. Improving Robustness of Deep Neural Networks with KAN-based Adversarial Training (KAT). *IEEE Transactions on Artificial Intelligence* **2024**, *9*, 130–145.
10. Guo, H.; Li, F.; Li, J.; Liu, H. KAN v.s. MLP for Offline Reinforcement Learning, 2024, [arXiv:cs.LG/2409.09653].
11. 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, 2024, [arXiv:cs.LG/2410.07446].
12. Jamali, A.; others. Advances in Kolmogorov-Arnold Networks for Data Fitting and PDE Solving. *Journal of Computational Physics* **2024**, *423*, 145–160.
13. Yu, B.; others. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics* **2018**, *6*, 1–12.
14. Xu, L.; others. Time-Kolmogorov-Arnold Networks and Multi-Task Kolmogorov-Arnold Networks for Time Series Prediction. *Journal of Time Series Analysis* **2024**, *45*, 200–220.
15. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural networks* **1989**, *2*, 359–366.
16. Agarwal, R.; Melnick, L.; Frosst, N.; Zhang, X.; Lengerich, B.; Caruana, R.; Hinton, G.E. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems* **2021**, *34*, 4699–4711.
17. Lahini, Y.; Pugatch, R.; Pozzi, F.; Sorel, M.; Morandotti, R.; Davidson, N.; Silberberg, Y. Observation of a localization transition in quasiperiodic photonic lattices. *Physical review letters* **2009**, *103*, 013901.
18. Kiamari, M.; others. Graph Kolmogorov-Arnold Networks: A Novel Approach for Node Classification. *IEEE Transactions on Neural Networks and Learning Systems* **2024**, *35*, 1450–1465.
19. Hestness, J.; Narang, S.; Ardalani, N.; Diamos, G.; Jun, H.; Kianinejad, H.; Patwary, M.M.A.; Yang, Y.; Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409* **2017**.
20. Xu, K.; Chen, L.; Wang, S. Kolmogorov-Arnold Networks for Time Series: Bridging Predictive Power and Interpretability, 2024, [arXiv:cs.LG/2406.02496].
21. Yang, Z.; Zhang, J.; Luo, X.; Lu, Z.; Shen, L. Activation Space Selectable Kolmogorov-Arnold Networks, 2024, [arXiv:cs.LG/2408.08338].
22. Aziznejad, S.; Unser, M. Deep spline networks with control of Lipschitz regularity. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 3242–3246.

23. Cheon, M. Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing. *arXiv preprint arXiv:2406.00600* **2024**.
24. Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems* **2020**, *33*, 7462–7473.
25. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* **2016**.
26. Gao, Y.; Tan, V.Y.F. On the Convergence of (Stochastic) Gradient Descent for Kolmogorov–Arnold Networks, 2024, [arXiv:cs.LG/2410.08041].
27. Hadash, G.; Kermany, E.; Carmeli, B.; Lavi, O.; Kour, G.; Jacovi, A. Estimate and Replace: A Novel Approach to Integrating Deep Neural Networks with Existing Applications. *arXiv preprint arXiv:1804.09028* **2018**.
28. Kundu, A.; Sarkar, A.; Sadhu, A. Kanqas: Kolmogorov-Arnold Network for Quantum Architecture Search. *arXiv preprint arXiv:2406.17630* **2024**.
29. Liu, Z.; Ma, P.; Wang, Y.; Matusik, W.; Tegmark, M. KAN 2.0: Kolmogorov-Arnold Networks Meet Science. *arXiv preprint arXiv:2408.10205* **2024**.
30. Cunningham, H.; Ewart, A.; Riggs, L.; Huben, R.; Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600* **2023**.
31. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* **2022**, *35*, 17359–17372.
32. Gukov, S.; Halverson, J.; Ruehle, F. Rigor with machine learning from field theory to the Poincaréconjecture. *Nature Reviews Physics* **2024**. doi:10.1038/s42254-024-00709-0.
33. Ismayilova, A.; Ismailov, V.E. On the Kolmogorov neural networks. *Neural Networks* **2024**, p. 106333.
34. Yang, X.; Wang, X. Kolmogorov-Arnold Transformer, 2024, [arXiv:cs.LG/2409.10594].
35. Wang, Y.; Xia, X.; Zhang, L.; Yao, H.; Chen, S.; You, J.; Zhou, Q.; Liu, X.J. One-dimensional quasiperiodic mosaic lattice with exact mobility edges. *Physical Review Letters* **2020**, *125*, 196604.
36. Lai, M.J.; Shen, Z. The kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions. *arXiv preprint arXiv:2112.09963* **2021**.
37. Cranmer, K.; others. Discovering Symbolic Models from Deep Learning with Inductive Biases. *Advances in Neural Information Processing Systems* **2020**, *33*, 17429–17442.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
39. Fukushima, K. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics* **1969**, *5*, 322–333.
40. Kohler, M.; Langer, S. On the rate of convergence of fully connected deep neural network regression estimates. *The Annals of Statistics* **2021**, *49*, 2231–2249.
41. SS, S.; AR, K.; R, G.; KP, A. Chebyshev Polynomial-Based Kolmogorov-Arnold Networks: An Efficient Architecture for Nonlinear Function Approximation, 2024, [arXiv:cs.LG/2405.07200].
42. Sprecher, D.A.; Draghici, S. Space-filling curves and Kolmogorov superposition-based neural networks. *Neural Networks* **2002**, *15*, 57–67.
43. Kashefi, A. PointNet with KAN versus PointNet with MLP for 3D Classification and Segmentation of Point Sets, 2024, [arXiv:cs.CV/2410.10084].
44. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 702–703.
45. Telgarsky, M. Neural networks and rational functions. International Conference on Machine Learning. PMLR, 2017, pp. 3387–3393.
46. He, J.; Li, L.; Xu, J.; Zheng, C. ReLU deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973* **2018**.
47. Gukov, S.; Halverson, J.; Manolescu, C.; Ruehle, F. Searching for ribbons with machine learning, 2023, [arXiv:math.GT/2304.09304].
48. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
49. Girosi, F.; Poggio, T. Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation* **1989**, *1*, 465–469.

50. Mahara, A.; Rishe, N.D.; Deng, L. The Dawn of KAN in Image-to-Image (I2I) Translation: Integrating Kolmogorov-Arnold Networks with GANs for Unpaired I2I Translation, 2024, [arXiv:cs.CV/2408.08216].
51. Yu, R.C.; Wu, S.; Gui, J. Residual Kolmogorov-Arnold Network for Enhanced Deep Learning, 2024, [arXiv:cs.CV/2410.05500].
52. LeCun, Y.; Bengio, Y.; others. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* **1995**, *3361*, 1995.
53. Li, Y.; Mao, H.; Girshick, R.; He, K. Exploring plain vision transformer backbones for object detection. European conference on computer vision. Springer, 2022, pp. 280–296.
54. Elhage, N.; Hume, T.; Olsson, C.; Nanda, N.; Henighan, T.; Johnston, S.; ElShowk, S.; Joseph, N.; DasSarma, N.; Mann, B.; Hernandez, D.; Askell, A.; Ndousse, K.; Jones, A.; Drain, D.; Chen, A.; Bai, Y.; Ganguli, D.; Lovitt, L.; Hatfield-Dodds, Z.; Kernion, J.; Conerly, T.; Kravec, S.; Fort, S.; Kadavath, S.; Jacobson, J.; Tran-Johnson, E.; Kaplan, J.; Clark, J.; Brown, T.; McCandlish, S.; Amodei, D.; Olah, C. Softmax Linear Units. *Transformer Circuits Thread* **2022**. https://transformer-circuits.pub/2022/solu/index.html.
55. Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the spectral bias of neural networks. International conference on machine learning. PMLR, 2019, pp. 5301–5310.
56. Yang, X.; Zhou, D.; Liu, S.; Ye, J.; Wang, X. Deep model reassembly. *Advances in neural information processing systems* **2022**, *35*, 25739–25753.
57. Schmidt-Hieber, J. The Kolmogorov–Arnold representation theorem revisited. *Neural networks* **2021**, *137*, 119–126.
58. Kour, G.; Saabne, R. Real-time segmentation of on-line handwritten arabic script. Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. IEEE, 2014, pp. 417–422.
59. Liu, J.; others. Kolmogorov-Arnold Networks for Symbolic Regression and Time Series Prediction. *Journal of Machine Learning Research* **2024**, *25*, 95–110.
60. Kovachki, N.; Li, Z.; Liu, B.; Azizzadenesheli, K.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research* **2023**, *24*, 1–97.
61. Poeta, E.; Giobergia, F.; Pastor, E.; Cerquitelli, T.; Baralis, E. A Benchmarking Study of Kolmogorov-Arnold Networks on Tabular Data, 2024, [arXiv:cs.LG/2406.14529].
62. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; others. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **2017**, *114*, 3521–3526.
63. Aghaomidi, P.; Wang, G. ECG-SleepNet: Deep Learning-Based Comprehensive Sleep Stage Classification Using ECG Signals, 2024, [arXiv:cs.AI/2412.01929].
64. Hecht-Nielsen, R. Kolmogorov's mapping neural network existence theorem. Proceedings of the international conference on Neural Networks. IEEE press New York, NY, USA, 1987, Vol. 3, pp. 11–14.
65. Meunier, D.; Lambiotte, R.; Bullmore, E.T. Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience* **2010**, *4*, 7572.
66. Inzirillo, H.; Genet, R. A Gated Residual Kolmogorov-Arnold Networks for Mixtures of Experts, 2024, [arXiv:cs.LG/2409.15161].
67. Qiu, R.; Miao, Y.; Wang, S.; Yu, L.; Zhu, Y.; Gao, X.S. PowerMLP: An Efficient Version of KAN, 2024, [arXiv:cs.LG/2412.13571].
68. Zniyed, Y.; Nguyen, T.P.; others. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
69. Walsh, J.L. *Interpolation and approximation by rational functions in the complex domain*; Vol. 20, American Mathematical Soc., 1935.
70. Courant, R.; Friedrichs, K.; Lewy, H. On the partial difference equations of mathematical physics. *Mathematische Annalen* **1928**, *100*, 32–74.
71. Jin, J.; Li, X.; Huang, H.; Liu, L.; Sun, Y. PEP-GS: Perceptually-Enhanced Precise Structured 3D Gaussians for View-Adaptive Rendering, 2024, [arXiv:cs.CV/2411.05731].
72. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6023–6032.

73. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895* **2020**.
74. Poggio, T.; Banburski, A.; Liao, Q. Theoretical issues in deep networks. *Proceedings of the National Academy of Sciences* **2020**, *117*, 30039–30045.
75. 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, 2024, [arXiv:cs.LG/2406.02917].
76. Wang, Y.; Siegel, J.W.; Liu, Z.; Hou, T.Y. On the expressiveness and spectral bias of KANs. *arXiv preprint arXiv:2410.01803* **2024**.
77. Schmidt-Hieber, J. Nonparametric regression using deep neural networks with ReLU activation function **2020**.
78. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; others. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **2019**, *32*.
79. Elhage, N.; Hume, T.; Olsson, C.; Schiefer, N.; Henighan, T.; Kravec, S.; Hatfield-Dodds, Z.; Lasenby, R.; Drain, D.; Chen, C.; others. Toy models of superposition. *arXiv preprint arXiv:2209.10652* **2022**.
80. Ronen, B.; Jacobs, D.; Kasten, Y.; Kritchman, S. The convergence rate of neural networks for learned functions of different frequencies. *Advances in Neural Information Processing Systems* **2019**, *32*.
81. Xu, H.; Sin, F.; Zhu, Y.; Barbič, J. Nonlinear material design using principal stretches. *ACM Transactions on Graphics (TOG)* **2015**, *34*, 1–11.
82. He, Y. *Machine Learning in Pure Mathematics and Theoretical Physics*; G - Reference,Information and Interdisciplinary Subjects Series, World Scientific, 2023.
83. Polo-Molina, A.; Alfaya, D.; Portela, J. MonoKAN: Certified Monotonic Kolmogorov-Arnold Network, 2024, [arXiv:cs.LG/2409.11078].
84. Braun, J.; Griebel, M. On a constructive proof of Kolmogorov's superposition theorem. *Constructive Approximation* **2009**, *30*, 653–675.
85. Zhang, S.; Zhang, P.; Hou, T.Y. Multiscale invertible generative networks for high-dimensional Bayesian inference. International Conference on Machine Learning. PMLR, 2021, pp. 12632–12641.
86. Li, Z. Kolmogorov-Arnold Networks are Radial Basis Function Networks. *ArXiv* **2024**, *abs/2405.06721*.
87. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756* **2024**.
88. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022.
89. Jacob, B.; Howard, A.A.; Stinis, P. SPIKANs: Separable Physics-Informed Kolmogorov-Arnold Networks, 2024, [arXiv:cs.LG/2411.06286].
90. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature machine intelligence* **2021**, *3*, 218–229.
91. Wang, S.; Li, B.; Chen, Y.; Perdikaris, P. PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks. *arXiv preprint arXiv:2402.00326* **2024**.
92. Sun, Y.; Zhu, H.; Qin, C.; Zhuang, F.; He, Q.; Xiong, H. Discerning decision-making process of deep neural networks with hierarchical voting transformation. *Advances in Neural Information Processing Systems* **2021**, *34*, 17221–17234.
93. Michaud, E.J.; Liu, Z.; Tegmark, M. Precision machine learning. *Entropy* **2023**, *25*, 175.
94. Boullé, N.; Nakatsukasa, Y.; Townsend, A. Rational neural networks. *Advances in neural information processing systems* **2020**, *33*, 14243–14253.
95. Lin, H.W.; Tegmark, M.; Rolnick, D. Why does deep and cheap learning work so well? *Journal of Statistical Physics* **2017**, *168*, 1223–1247.
96. Wang, S.; Sankaran, S.; Perdikaris, P. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404* **2022**.
97. Kolmogorov, A.N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. Doklady Akademii Nauk. Russian Academy of Sciences, 1957, Vol. 114, pp. 953–956.

98. Wang, H.; others. SpectralKAN: Spatial-Spectral Kolmogorov-Arnold Networks for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* **2024**, *62*, 500–515.

99. Goyal, M.; Goyal, R.; Lall, B. Learning activation functions: A new paradigm for understanding neural networks. *arXiv preprint arXiv:1906.09529* **2019**.

100. Udrescu, S.M.; Tan, A.; Feng, J.; Neto, O.; Wu, T.; Tegmark, M. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems* **2020**, *33*, 4860–4871.

101. Galitsky, B.A. Kolmogorov-Arnold Network for Word-Level Explainable Meaning Representation. *Preprints* **2024**. Retrieved from https://www.preprints.org/manuscript/202405.1981.

102. Braun, J.; Griebel, M. On a constructive proof of Kolmogorov's superposition theorem. *Constructive approximation* **2009**, *30*, 653–675.

103. Kauffman, L.H.; Russkikh, N.E.; Taimanov, I.A. Rectangular knot diagrams classification with deep learning, 2020, [arXiv:math.GT/2011.03498].

104. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv preprint arXiv:1710.05941* **2017**.

105. Deng, X.; He, X.; Bao, J.; Zhou, Y.; Cai, S.; Cai, C.; Chen, Z. MvKeTR: Chest CT Report Generation with Multi-View Perception and Knowledge Enhancement, 2025, [arXiv:cs.CV/2411.18309].

106. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA; Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H.M.; Fergus, R.; Vishwanathan, S.V.N.; Garnett, R., Eds., 2017, pp. 5998–6008.

107. Gordon, M.A.; Duh, K.; Kaplan, J. Data and Parameter Scaling Laws for Neural Machine Translation. ACL Rolling Review - May 2021, 2021.

108. Bingham, G.; Miikkulainen, R. Discovering parametric activation functions. *Neural Networks* **2022**, *148*, 48–65.

109. Lagendijk, A.; Tiggelen, B.v.; Wiersma, D.S. Fifty years of Anderson localization. *Physics today* **2009**, *62*, 24–29.

110. Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks* **2017**, *94*, 103–114.

111. Gukov, S.; Halverson, J.; Ruehle, F.; Sułkowski, P. Learning to Unknot. *Mach. Learn. Sci. Tech.* **2021**, *2*, 025035, [arXiv:math.GT/2010.16263]. doi:10.1088/2632-2153/abe91f.

112. Zhang, S.; Shen, Z.; Yang, H. Neural network architecture beyond width and depth. *Advances in Neural Information Processing Systems* **2022**, *35*, 5669–5681.

113. Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified perceptual parsing for scene understanding. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 418–434.

114. Yu, W.; Si, C.; Zhou, P.; Luo, M.; Zhou, Y.; Feng, J.; Yan, S.; Wang, X. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**.

115. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; others. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155* **2019**.

116. Unser, M.; Aldroubi, A.; Eden, M. B-spline signal processing. I. Theory. *IEEE transactions on signal processing* **1993**, *41*, 821–833.

117. Wu, Y.; He, K. Group normalization. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.

118. Nanda, N.; Chan, L.; Lieberum, T.; Smith, J.; Steinhardt, J. Progress measures for grokking via mechanistic interpretability. The Eleventh International Conference on Learning Representations, 2023.

119. SpringerLink. On functions of three variables, 2023. Retrieved from https://link.springer.com/article/10.1007/BF01213206.

120. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* **2020**.

121. Horner, W. A New Method of Solving Numerical Equations of all Orders, by Continuous Approximation. Abstracts of the Papers Printed in the Philosophical Transactions of the Royal Society of London. JSTOR, 1815, Vol. 2, pp. 117–117.

122. Siegel, J.W. Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces. *Journal of Machine Learning Research* **2023**, *24*, 1–52.

123. Ruehle, F. Data science applications to string theory. *Phys. Rept.* **2020**, *839*, 1–117. doi:10.1016/j.physrep.2019.09.005.

124. Lin, J.N.; Unbehauen, R. On the realization of a Kolmogorov network. *Neural Computation* **1993**, *5*, 18–20.

125. Lu, A.; Feng, T.; Yuan, H.; Song, X.; Sun, Y. Revisiting Neural Networks for Continual Learning: An Architectural Perspective, 2024, [arXiv:cs.LG/2404.14829].

126. Zniyed, Y.; Nguyen, T.P.; others. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
127. Zhou, X.C.; Wang, Y.; Poon, T.F.J.; Zhou, Q.; Liu, X.J. Exact new mobility edges between critical and localized states. *Physical Review Letters* **2023**, *131*, 176401.
128. Kolb, B.; Whishaw, I.Q. Brain plasticity and behavior. *Annual review of psychology* **1998**, *49*, 43–64.
129. Dubcáková, R. Eureqa: software review. *Genetic Programming and Evolvable Machines* **2011**, *12*, 173–178.
130. Cheon, M. Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing, 2024, [arXiv:cs.CV/2406.00600].
131. Ruijters, D.; ter Haar Romeny, B.M.; Suetens, P. Efficient GPU-based texture interpolation using uniform B-splines. *Journal of Graphics Tools* **2008**, *13*, 61–69.
132. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
133. Li, X.; Ganeshan, S.; Pixley, J.; Sarma, S.D. Many-body localization and quantum nonergodicity in a model with a single-particle mobility edge. *Physical review letters* **2015**, *115*, 186601.
134. 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, 2024, [arXiv:cs.LG/2406.11045].

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.