Less is More in Kolmogorov-Arnold Networks: The Shared-Atom Architecture

Sanjam Wadhwa [*1] **Hrideya Sharma** [*1] **Shashank Singh** [1] **Saurabh Sharma** [1]

Abstract

Standard MLPs use fixed nonlinearities, which can limit interpretability despite strong universal approximation properties. Kolmogorov-Arnold Networks (KANs) address this issue by assigning learnable univariate functions to edges, but this induces a prohibitively large parameter tensor and corresponding scaling difficulties. We develop Shared-Atom KAN (SA-KAN) by enforcing a low-rank functional factorization in place of independent edge functions. SA-KAN learns a dictionary of K reusable atoms that are combined through sparse edge-wise coefficients, effectively separating the choice of basis from network architecture and compressing the layer into a symbolic vocabulary. Empirically, this formulation reduces parameter counts by 25–30% relative to Vanilla KAN while simultaneously improving performance on structured tasks.

1. Introduction

The emergence of Kolmogorov-Arnold Networks (KANs) presents an intriguing alternative to standard multilayer perceptrons, replacing learned scalar weights followed by fixed nonlinearities with learnable univariate functions parameterized on edges. Inspired by the Kolmogorov-Arnold representation theorem and its later constructive refinements, this architecture shifts the computational burden from static node activations to explicit modelling of individual input-output relations. The result is a more function-centered class of models with an inductive bias toward composing multivariate behaviour from simpler one-dimensional functions(Kolmogorov, 1956; Arnol'd, 1957; Kolmogorov, 1957; Liu et al., 2024b).

[*]Equal contribution [1]Thapar Institute of Engineering & Technology, Patiala, Punjab, India. Correspondence to: Sanjam wadhwa <swadhwa_be24@thapar.edu>, Hrideya Sharma <hsharma1_be24@thapar.edu>, Shashank Singh <shashank.singh@thapar.edu>, Saurabh Sharma <saurabh.sharma@thapar.edu>.

The primary strength of KANs, the ability to learn complex functions by representing edge relationships as parameterizable functions, is also the direct source of their central limitation. In a traditional KAN layer with 'nin' inputs and 'nout' outputs, and an edge represented by a d-dimensional basis expansion (e.g., Taylor series, Fourier series, etc.), the KAN requires the learning and storage of a $nin \times nout \times d$ third-order coefficient tensor. As the width of the network increases, assigning a separate basis expansion to each individual edge leads to a combinatorial explosion in the number of parameters and consequently a drastic increase in memory overhead and computational cost(Liu et al., 2024b;a; Hou & Zhang, 2024; Wang et al., 2024).

The explosive growth in parameter count highlights the central architectural trade-off in KANs: while moving expressive power to the edges is appealing, assigning fully independent function spaces to every edge also introduces substantial structural redundancy. Recent work on KAN architectures and parameter-efficient alternatives supports the view that the scaling challenge is not purely one of expressivity, but also one of how function space is allocated across edges(Liu et al., 2024b; Hou & Zhang, 2024; Wang et al., 2024; Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025).

We address this limitation by introducing Shared-Atom Kolmogorov-Arnold Networks (SA-KANs). SA-KAN achieves computational efficiency and structural redundancy reduction by avoiding the parameter proliferation characteristic of traditional KANs. Instead, it proposes a more structured, low-rank factorization of the KAN layer's functional transformations. SA-KAN explicitly decouples the dimensionality of the functional basis representation from the network's connectivity by projecting basic basis evaluations, such as those for Chebyshev polynomials or wavelets, into a compact dictionary of K globally shared, learnable "functional atoms." Each topological edge is then configured to consist of a learnable s-sparse combination of these atoms, rather than a redundant set of independent, isolated coefficients(Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025; Bozorgasl & Chen, 2024; Sidiropoulos et al., 2017).

The resulting SA-KAN, in imposing this structured dictionary-based prior, transforms the KAN from a sum

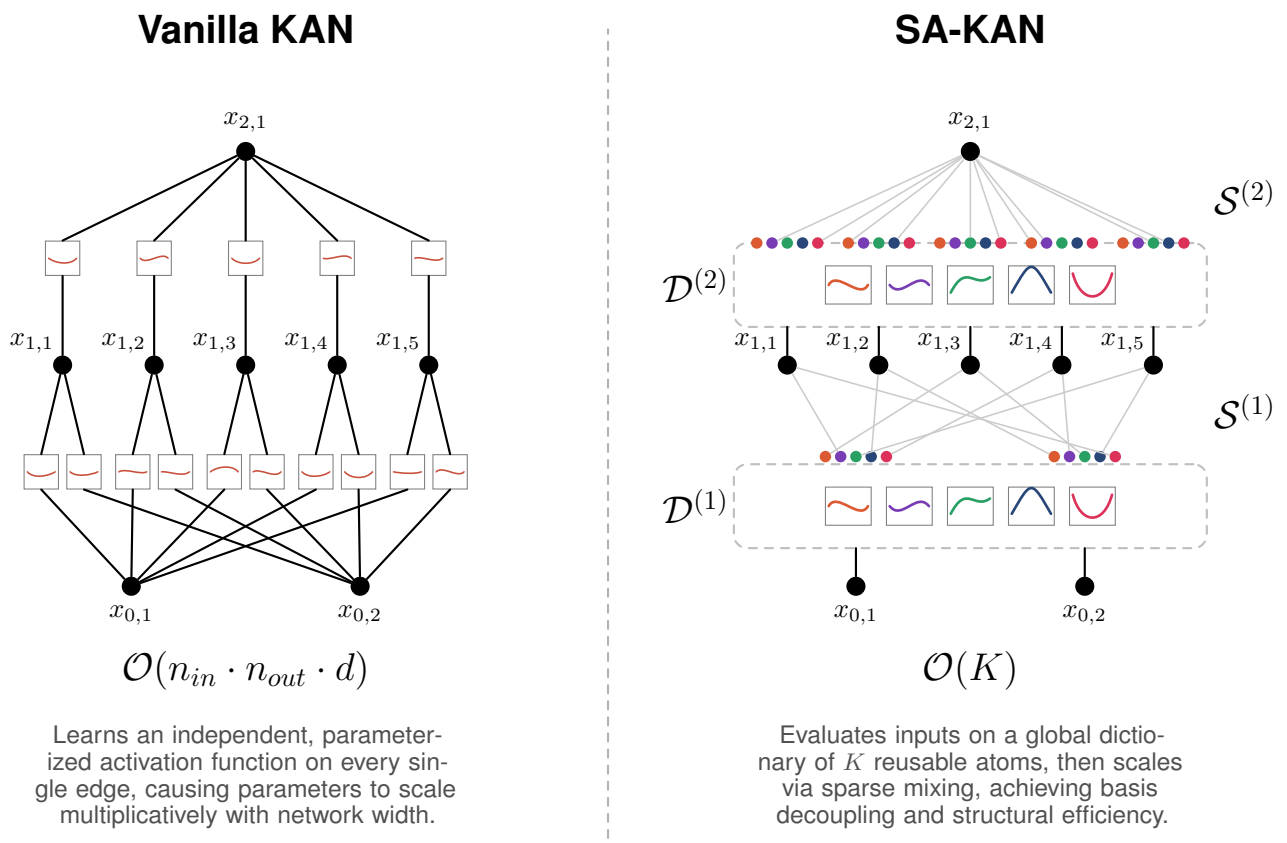

Figure 1. Overview of the Shared-Atom KAN architecture. The diagram illustrates how primitive basis evaluations are projected into a shared atom dictionary and then combined through sparse edge-wise routing to produce output features.

of weakly related edge functions into a sparsely routed mixture-of-experts functional system. This transformation, when considered in terms of the forward pass, enables each of the primitive bases to be evaluated a single time per input dimension and projected into a shared, compact atom space. From there, they can be selectively aggregated to specific output nodes via highly efficient sparse tensor contractions. Consequently, SA-KAN obviates the need to recover a fully dense coefficient tensor at each layer, directly mitigating the $O(nin \cdot nout \cdot d)$ computational bottleneck of standard KANs while preserving the learnable edge-wise functional transformations at its core. The explicit sparse dictionary structure of SA-KAN offers benefits beyond computational efficiency, namely by enabling enhanced interpretability. Unlike trying to decipher the interactions of thousands of partially overlapping edge expansions, the use of a small, discrete set of global atoms, combined with the enforceability of sparse router mechanisms, provides a directly analyzable internal representation of learned functions that can be extractable via symbolic regression(Gao et al., 2025; Ta et al., 2025; Sidiropoulos et al., 2017; Noorizadegan et al., 2025).

2. Related Work

Kolmogorov-Arnold Networks. Kolmogorov-Arnold Networks (KANs) are designed as an alternative to conventional MLPs by using trainable univariate functions on edges instead of fixed activations and scalar weights. The original KAN formulation takes the perspective of the Kolmogorov-Arnold representation and highlights the potential for improved interpretability and function approximation, particularly in scientific settings. A direct consequence of the edge-wise parameterization, however, is a large coefficient tensor whose size grows with connectivity, leading to significant memory and computational overhead(Liu et al., 2024b;a; Kolmogorov, 1956; Arnol'd, 1957; Kolmogorov, 1957).

Basis-oriented KAN variants. A significant portion of subsequent research aims to address KAN inefficiency by modifying the basis family inside each KAN layer while retaining the edge-wise learning paradigm. For instance, Wav-KAN replaces the usual polynomial-style basis with wavelets, arguing that multiresolution primitives are better suited for localized or oscillatory structure. Community surveys and repositories further document the rapid growth of such variants. In most of these approaches, however,

the basic independence of the edge functions is still preserved(Bozorgasl & Chen, 2024; Sidharth et al., 2024; Xu et al., 2024; Aghaei, 2024; 2025; Seydi, 2024; Chao et al., 2026; Chiu et al., 2026).

Parameter-efficient and low-rank KANs. A closely related area of work seeks to exploit redundancy in KAN-style parameterizations through architectural structure, parameter sharing, or lower-dimensional representations. This direction is particularly relevant to our work because it frames KAN inefficiency as a structural problem within the edge-function parameter tensor rather than merely a problem of basis evaluation(Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025; Delis, 2024).

Position of our work. SA-KAN most closely resembles parameter-efficient KAN variants in spirit. The key difference lies in the level at which structural constraints are imposed: instead of applying parameter sharing or compression after an edge-wise parameterization has already been specified, SA-KAN builds low-dimensional shared structure directly into the layer through a global dictionary of reusable atom functions and sparse edge-wise mixtures over those atoms. This also distinguishes it from basis-oriented variants such as Wav-KAN, which change the primitive family but still retain largely independent edge functions. In this sense, SA-KAN sits at the intersection of KAN architectures, low-dimensional factorization, and interpretable compression as a native reparameterization(Bozorgasl & Chen, 2024; Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025; Li, 2024; Noorizadegan et al., 2025).

3. Architecture

In this section, we introduce Shared-Atom Kolmogorov-Arnold Networks (SA-KAN) as a structured reconceptualization of basis-parameterized KAN layers. By shifting the conceptual framing of KANs from independent function approximators to a cohesive tensorial and polynomial architecture, we expose the fundamental redundancy in the original KAN formulation and derive an efficient multilinear low-rank prior that resolves the parametric scaling bottlenecks of deep KANs while maintaining expressivity(Liu et al., 2024b; Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025; Tucker, 1963; De Lathauwer et al., 2000).

3.1. Rethinking KANs as Tensorial and Polynomial Architectures

The classical Kolmogorov–Arnold representation theorem states that any continuous function $f : [0,1]^n \to \mathbb{R}$ can be written as a finite superposition of continuous univariate functions:

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

where the inner functions $\psi_{q,p}$ and outer functions Φ_q are continuous and univariate. KANs operationalize this theorem-inspired decomposition by replacing fixed scalar weights with learnable one-dimensional functions on network edges(Kolmogorov, 1956; Arnol'd, 1957; Kolmogorov, 1957; Liu et al., 2024b).

First, let us formalize a single KAN layer. Suppose a KAN layer takes inputs of dimension n_{in} and outputs of dimension n_{out}. We may parameterize the layer as a matrix $\Phi = \{\phi_{j,i}\}$ where $j = 1, \ldots, n_{\text{out}}$ and $i = 1, \ldots, n_{\text{in}}$. The layer can be expressed as

$$x_j^{\text{out}} = \sum_{i=1}^{n_{\text{in}}} \phi_{j,i}(x_i), \qquad j = 1, \ldots, n_{\text{out}}.$$

In Kolmogorov-Arnold Networks, each $\phi_{j,i}$ represents a shape function parameterized as a spline or polynomial curve. The generic form used in basis-parameterized KANs can be written as a linear combination of d basis functions, with $\beta(t) = [b_1(t), b_2(t), \ldots, b_d(t)]^\top$. Thus,

$$\phi_{j,i}(x_i) = \sum_{m=1}^{d} A_{j,i,m}\, b_m(x_i).$$

The coefficients $A_{j,i,m}$ are learnable scalars. KANs therefore operate as a polynomial architecture, since their edge functions are built from compositions of spline- or polynomial-like primitives. They also exhibit a structural deficit: the geometric layout of the network is separable from the evaluation of the underlying basis functions. The coefficients may be arranged in a third-order tensor

$$\mathcal{A} \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}} \times d},$$

which implies a very large and unconstrained parameter set growing quadratically with width, namely $O(N^2 d)$. If we stack L such layers, the total parameter count grows as $O(LN^2 d)$, which directly contributes to overfitting risk and high inference cost(Liu et al., 2024b; Hou & Zhang, 2024; Wang et al., 2024; Sidiropoulos et al., 2017).

3.2. Multilinear and Low-Rank Priors

A neural network must represent an input-output function, but it is not obvious that each individual connection should require a completely independent functional mapping. Low-dimensional manifolds and weight-sharing mechanisms are central to the feature abstraction found in conventional multilayer perceptrons. We suggest that the functions represented

by effective KANs also possess strong compositional structure and therefore substantial redundancy. To realize this idea, we impose multilinear and low-rank priors(Caruana, 1997; Zhang & Yang, 2021; Tucker, 1963; De Lathauwer et al., 2000).

Our hypothesis is that the target function often depends only on a small set of invariant latent features, such as physical laws or recurring motifs, so the many individual connections of a KAN need only span a much lower-dimensional functional space. In this sense, the underlying representation has low intrinsic rank. Standard KAN parameterization forces the model to conform to a high-rank tensor $\mathcal{A}$, whereas our design removes that extraneous structure and allows highly accurate inference with a much more compact representation(Tucker, 1963; De Lathauwer et al., 2000; Sidiropoulos et al., 2017; Caruana, 1997).

3.3. The Shared-Atom Factorization

We realize this structural prior by decomposing the parameterization of individual edges. Instead of assigning every edge its own basis expansion, we introduce a dictionary of global functional atoms. Let

$$D \in \mathbb{R}^{K \times d}$$

be the dictionary matrix, where $K \ll d$. Each row of D defines a shared basis function:

$$\psi_k(t) = \sum_{m=1}^{d} D_{k,m}\, b_m(t), \qquad k = 1, \ldots, K.$$

The individual edges no longer define independent polynomial bases. Instead, they possess a mixing vector $s_{j,i} \in \mathbb{R}^K$ that combines these global basis functions:

$$\phi_{j,i}(x_i) = \sum_{k=1}^{K} s_{j,i,k}\, \psi_k(x_i) = s_{j,i}^{\top} D\, \beta(x_i).$$

The topological coefficients $s_{j,i,k}$ form a routing tensor

$$S \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}} \times K},$$

where S acts as the topological component. Under this construction, the standard KAN coefficient tensor is factorized as

$$A^{\flat} = S^{\flat} D,$$

turning the original edge-wise coefficient matrix into a lightweight, low-rank atom-routing network. The task of representing complicated shape information is handled by the atom dictionary D, while the task of mixing those atoms according to network topology falls to S(Tucker, 1963; De Lathauwer et al., 2000; Sidiropoulos et al., 2017; Gao et al., 2025).

3.4. Fused Tensorial Forward Computation

If we first compute the full dense tensor $\mathcal{A} = S \times_3 D$ and then apply the standard KAN computation, we would obtain the same mathematical function as the low-rank model, but without any computational advantage: the full $O(N^2 d)$ memory footprint would reappear during execution. SA-KAN is designed to avoid this inefficiency.

For a batch input $X \in \mathbb{R}^{B \times n_{\text{in}}}$, the network output can be computed by exploiting associativity so that the full tensor $\mathcal{A}$ is never instantiated.

First, the primitive basis functions are evaluated:

$$\mathcal{B}_{b,i,m} = b_m(X_{b,i}) \in \mathbb{R}^{B \times n_{\text{in}} \times d}.$$

Second, these primitive responses are projected into the shared atom space:

$$\mathcal{G}_{b,i,k} = \sum_{m=1}^{d} \mathcal{B}_{b,i,m} D_{k,m} \in \mathbb{R}^{B \times n_{\text{in}} \times K}.$$

Finally, the topological mixture and summation are applied:

$$X_{b,j}^{\text{out}} = \sum_{i=1}^{n_{\text{in}}} \sum_{k=1}^{K} S_{j,i,k}\, \mathcal{G}_{b,i,k}.$$

By performing the computation in this order, we never form the dense tensor $\mathcal{A}$. The forward pass is instead dominated by multiplication with the low-dimensional dictionary D, making the architecture substantially more efficient while preserving the desired factorized structure(De Lathauwer et al., 2000; Sidiropoulos et al., 2017; Kilmer & Martin, 2011).

4. Optimization and Routing

While the Shared-Atom factorization mathematically compresses the parameter space, optimizing this bilinear formulation introduces new challenges. A fully dense routing tensor S would allow every edge to utilize every atom, risking homogenization of the dictionary and hindering interpretability. To circumvent this, we formalize the optimization landscape using sparse geometrical priors(Sidiropoulos et al., 2017; Noorizadegan et al., 2025; Gao et al., 2025).

4.1. Sparse Routing and Union-of-Subspaces Geometry

Beyond learning a low-rank multilinear representation, SA-KAN imposes a hard sparsity constraint on the topological routing. We mandate that each edge function can only activate a highly restricted subset of the available atoms. Mathematically, the mixing vector $s_{j,i}$ for any given edge is constrained by an L_0 pseudo-norm:

$$\|s_{j,i}\|_0 \le s, \qquad \text{where } s \ll K.$$

Under this constraint, the edge function $\phi_{j,i}(x_i)$ is constructed using at most s non-zero atoms. This is not merely an exercise in weight pruning; it fundamentally alters the geometric inductive bias of the model.

For a fixed shared dictionary D, the admissible set of edge coefficient vectors is defined as:

$$\mathcal{A}_s(D) = \left\{ D^\top v \mid v \in \mathbb{R}^K,\ \|v\|_0 \leq s \right\}.$$

Each unique support pattern of size s corresponds to a linear subspace spanned by exactly s atoms. Consequently, the set $\mathcal{A}_s(D)$ is a union of subspaces rather than a single uninterrupted vector space. While vanilla KANs assume that edge functions roam freely in an unconstrained d-dimensional space, SA-KAN optimization assumes that the underlying mathematical operators of the dataset can be represented within a finite set of intersecting low-dimensional hyperplanes(Tucker, 1963; De Lathauwer et al., 2000; Sidiropoulos et al., 2017).

4.2. Differentiable Top-s Masking via STE

Optimizing the discrete L_0 sparsity constraint during standard gradient descent is notoriously difficult. L_1 regularization often induces excessive shrinking of weights, while iterative post-training pruning requires expensive retraining cycles. SA-KAN resolves this by integrating a differentiable Top-s masking mechanism directly into the forward pass(Gao et al., 2025; Ta et al., 2025).

During execution, the active routing tensor S^{act} is computed by masking the raw underlying parameters S:

$$M_{j,i,k} = \begin{cases} 1 & \text{if } |S_{j,i,k}| \text{ is among the top } s \text{ values of } |S_{j,i,:}|, \\ 0 & \text{otherwise,} \end{cases}$$

$$S^{\text{act}} = S \odot M.$$

Because the masking operation is discontinuous, we approximate its derivative using the Straight-Through Estimator (STE). During the backward pass, gradients flow directly through the non-differentiable mask into the raw dense tensor S:

$$\frac{\partial \mathcal{L}}{\partial S} \approx \frac{\partial \mathcal{L}}{\partial S^{\text{act}}}.$$

This mechanism is crucial for optimization stability. It allows inactive routing parameters, namely those currently zeroed out by the mask, to continue accumulating gradients. If the loss landscape shifts such that a previously inactive atom becomes highly correlated with the residual error, its raw weight in S can grow until it surpasses the Top-s threshold and dynamically re-enters the active support set. This ensures that the sparse topology remains adaptive throughout training without suffering from permanent dead-neuron collapse(Ta et al., 2025; Gao et al., 2025).

4.3. Breaking Symmetry: Orthogonal Initialization

The bilinear factorization $A^\flat = S^\flat D$ is inherently subject to gauge freedom: for any invertible matrix $R \in \mathbb{R}^{K \times K}$, there exists an equivalent factorization $(S^\flat R^{-1})(RD)$ that yields the exact same forward pass. Without explicit conditioning, gradient descent can easily traverse this flat parameter space, leading to highly correlated atoms, where multiple rows in D learn nearly identical polynomial shapes, and thereby destabilizing routing sparsity.

To break this symmetry and enforce representation diversity, the polynomial dictionary D is initialized as a strictly orthogonal matrix. This guarantees that at the onset of training, every functional atom spans a mutually exclusive direction within the primitive basis space. Consequently, the routing tensor S is encouraged to learn distinct, non-overlapping assignments, improving the conditioning of the optimization landscape and preventing early-stage atom collapse(De Lathauwer et al., 2000; Sidiropoulos et al., 2017).

4.4. Analytical Parameter Scaling Laws

The theoretical advantage of SA-KAN optimization is most starkly realized in its parameter scaling laws. Consider a deep KAN comprised of layers with uniform width N. For a vanilla KAN, the number of parameters per layer scales quadratically with network width and multiplicatively with the basis resolution d:

$$P_{\text{Vanilla}} = \mathcal{O}(N^2 d).$$

In contrast, SA-KAN partitions this space into the topological routing tensor and the shared dictionary:

$$P_{\text{SA-KAN}} = \mathcal{O}(N^2 K + K d).$$

Because $K \ll d$ and $K \ll N^2$, SA-KAN decouples network width from basis resolution. For deep, highly complex polynomial architectures where high-frequency resolution is required, SA-KAN avoids the full quadratic explosion and maintains a substantially smaller parameter footprint while preserving the functional expressivity of a Kolmogorov-Arnold representation(Liu et al., 2024b; Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025).

5. Experiments

SA-KANs are tested on four sets of benchmarks which assess quality of approximation, robustness across the functional space, and capability in high dimension. The benchmark families consist of symbolic regression, analytic regression in a physics context, PDE surrogate modeling, and image classification. All splits, random seeds, and training parameters are kept the same across runs(Raissi et al., 2019; Wang et al., 2025; Takamoto et al., 2022; LeCun et al., 1998).

Table 1. Symbolic regression tasks used to evaluate approximation quality on low-dimensional analytic functions. The listed properties indicate the main structural pattern each task is intended to test.

Task	Expression	Property tested
S1	$\sin(x)\cos(y)$	Smooth bivariate oscillation
S2	$\sin(3x)\cos(3y)$	Higher-frequency interference
S3	$x^2 + y^2$	Radial polynomial
S4	$\exp(-(x^2 + y^2))$	Exponential radial decay
S5	$\sin(\pi x) + \cos(\pi y)$	Additive trigonometric structure
S6	$xy + \sin(x^2)$	Mixed polynomial-trigonometric composition

5.1. Benchmark design

5.1.1. Symbolic regression

The first benchmark consists of six synthetic regression tasks on $[-1, 1]^2$. These tasks cover smooth oscillatory structure, higher-frequency interference, radial structure, localized decay, additive composition, and mixed polynomial-trigonometric behavior. (Liu et al., 2024b; Bozorgasl & Chen, 2024).

5.1.2. Physics-inspired regression

The second benchmark contains three low-dimensional analytic regression tasks with physically meaningful domains: kinetic energy, inverse-square interaction, and damped oscillation(Raissi et al., 2019; Wang et al., 2025).

Table 2. Physics-inspired regression tasks used to test behavior on interpretable scientific formulas with physically meaningful input domains.

Task	Expression	Inputs	Domain
P1	$\frac{1}{2}mv^2$	m, v	$[0.1, 2.0]^2$
P2	$\frac{m_1 m_2}{r^2+\epsilon},\ \epsilon = 0.01$	m_1, m_2, r	$[0.5, 3.0]^3$
P3	$e^{-\gamma t}\cos(\omega t)$	γ, ω, t	$[0.1, 3.0]^3$

5.1.3. PDE surrogate modeling

The third benchmark evaluates the ability to solve standard scientific problems with PDEBench, using five representative tasks: Diffusion-Sorption, Diffusion-Reaction, Burgers, Allen-Cahn, and Wave Interference, reporting both RMSE and PSNR values(Takamoto et al., 2022).

5.1.4. High-dimensional classification

To test scalability beyond the few-variable setting, the final benchmark uses MNIST. Each 28×28 grayscale image is flattened into a 784-dimensional vector and classified into 10 classes using the standard 60k/10k train-test split(LeCun et al., 1998).

5.2. Baselines and setup

Four models are compared throughout: SA-KAN, SA-Wav-KAN, Vanilla KAN, and Wav-KAN. For symbolic and physics-inspired regression, all models use the same width pattern $[n_{\text{in}}, 32, 32, 1]$, with $n_{\text{in}} \in \{2, 3\}$. For MNIST, a compact $[784, 64, 10]$ architecture is used(Liu et al., 2024b; Bozorgasl & Chen, 2024).

For symbolic and physics-inspired regression, models are trained for 200 epochs with Adam, a learning rate of 10^{-3}, batch size 256, cosine annealing, and gradient clipping. MNIST uses the same optimizer with batch size 128 for 20 epochs. Regression performance is measured by RMSE, classification performance by top-1 accuracy, and parameter counts are reported for all models(Liu et al., 2024b; Bozorgasl & Chen, 2024; Raissi et al., 2019; Wang et al., 2025).

Table 3. Architectures and baselines used in the experiments.

Model	Basis	K	Notes
SA-KAN	Chebyshev polynomials	6	Shared dictionary
SA-Wav-KAN	Mexican-Hat wavelets	6	Shared wavelet atoms
Vanilla KAN	Chebyshev polynomials	—	Independent edge coefficients
Wav-KAN	Mexican-Hat wavelets	—	Independent wavelet parameters

5.3. Low-dimensional regression

5.3.1. Symbolic and physics-inspired tasks

Table 4 reports RMSE on the symbolic and physics-inspired benchmarks. On all six symbolic tasks, either SA-KAN or SA-Wav-KAN achieves the lowest error. This shows that the shared-atom parameterization preserves, and often improves, approximation quality in structured low-dimensional settings.

The gains are most visible on tasks such as S4, S5, and S6, where the shared-atom models substantially outperform the dense baselines. These functions contain reusable local structure and moderate compositional regularity, which are well matched to a shared dictionary and sparse routing scheme(Caruana, 1997; Zhang & Yang, 2021; Gao et al., 2025).

The trend differs on the physics-inspired tasks. Vanilla KAN performs best on all three tasks, with the largest gap appearing on the inverse-square interaction in P2. This suggests that highly localized or near-singular structure can still benefit from fully independent edge parameterization. Even so, SA-KAN remains competitive on P1 and P3, while Wav-KAN degrades sharply on the hardest case(Liu et al., 2024b; Bozorgasl & Chen, 2024; Wang et al., 2025).

Table 4. RMSE on symbolic and physics-inspired tasks. Lower values indicate better regression accuracy across both synthetic symbolic functions and physics-motivated targets.

Task	SA-KAN	SA-Wav-KAN	Vanilla KAN	Wav-KAN
S1	0.0053	0.0056	0.0133	0.0103
S2	0.0072	0.0071	0.0142	0.0130
S3	0.0079	0.0063	0.0129	0.0076
S4	0.0035	0.0038	0.0135	0.0059
S5	0.0079	0.0080	0.0147	0.0181
S6	0.0100	0.0078	0.0143	0.0164
P1	0.0102	0.0138	0.0117	0.0228
P2	0.1306	0.1814	0.0811	0.6978
P3	0.0264	0.0415	0.0184	0.0245

5.3.2. Parameter efficiency

These results are obtained with a smaller parameter budget than Vanilla KAN. Since tasks sharing the same input dimension use the same widths, parameter counts are constant within each group(Liu et al., 2024b; Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025).

Table 5. Parameter count by input dimension for the low-dimensional regression setting. Shared-atom models reduce parameter usage relative to Vanilla KAN while keeping the same layer widths within each dimension group.

Input dimension	SA / SA-Wav	Vanilla KAN	Wav-KAN
2	6,929	9,025	3,553
3	7,121	9,281	3,649

For both 2D and 3D tasks, SA-KAN and SA-Wav-KAN decrease parameter counts by approximately 23% compared to Vanilla KAN. Wav-KAN is smaller but sacrifices regression accuracy for this compactness. The shared-atom formulation strikes a good trade-off between compressibility and approximation accuracy(Gao et al., 2025; Koenig et al., 2025; Ta et al., 2025; Bozorgasl & Chen, 2024).

5.4. PDE surrogate modeling

Tables 6 and 7 report the PDEBench results in terms of RMSE and PSNR. The shared-atom models, particularly SA-Wav-KAN, perform strongly on most tasks.

SA-Wav-KAN performs best on four of the five PDE tasks, and on the same four in PSNR. The largest improvement occurs in Allen-Cahn with an RMSE of 0.00224 and PSNR of 53.01 dB. This shows the shared-atom formulation does not break down and generalizes beyond analytic regression. The formulation also performs quite well on structured scientific modeling tasks(Takamoto et al., 2022; Raissi et al., 2019; Wang et al., 2025).

The only task that Vanilla KAN continues to outperform is 2D Wave Interference. This implies that deeply oscillatory structures may still benefit from a completely unshared edge parameterization. The performance of SA-Wav-KAN on this task remains significantly better than that of Wav-KAN(Liu et al., 2024b; Bozorgasl & Chen, 2024; Takamoto et al., 2022).

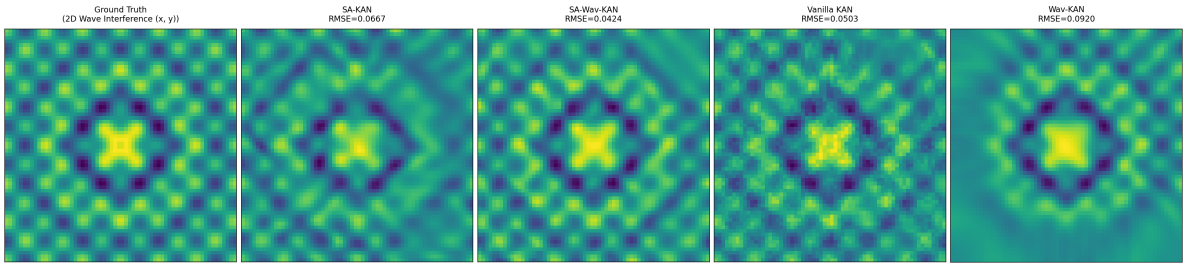

Figure 2. Qualitative comparison on the 2D wave interference task across representative x-y slices. The image highlights differences in spatial prediction quality for the competing models on the most oscillatory PDE benchmark.

Table 6. PDE surrogate modeling RMSE on five PDEBench tasks. Lower values indicate more accurate surrogate predictions for the underlying spatiotemporal dynamics.

PDE task	SA-KAN	SA-Wav-KAN	Vanilla KAN	Wav-KAN
1D Diffusion-Sorption	0.01067	**0.00981**	0.02281	0.02372
2D Diffusion-Reaction	0.03834	**0.02846**	0.03467	0.07216
1D Burgers Equation	0.01183	**0.01117**	0.03214	0.01705
2D Allen-Cahn Equation	0.00528	**0.00224**	0.01755	0.01243
2D Wave Interference	0.0667	**0.0424**	0.0503	0.0920

Table 7. PDE surrogate modeling PSNR on five PDEBench tasks. Higher values indicate better reconstruction fidelity for the predicted PDE solution fields.

PDE task	SA-KAN	SA-Wav-KAN	Vanilla KAN	Wav-KAN
1D Diffusion-Sorption	39.44	**40.17**	32.84	32.50
2D Diffusion-Reaction	28.33	**30.92**	29.20	22.83
1D Burgers Equation	38.54	**39.04**	29.86	35.36
2D Allen-Cahn Equation	45.55	**53.01**	35.12	38.11
2D Wave Interference	23.95	26.68	**28.61**	20.71

5.5. High-dimensional learning

5.5.1. MNIST CLASSIFICATION

Table 8 shows the MNIST results for the compact $[784, 64, 10]$ models(LeCun et al., 1998).

Table 8. MNIST classification results for the compact $[784, 64, 10]$ models. Accuracy measures test performance, while the parameter count shows the cost of each architecture at this width.

Model	Accuracy (%)	Parameters
SA-KAN	96.65	305,066
SA-Wav-KAN	97.22	305,066
Vanilla KAN	91.09	406,602

This shows that the shared dictionary is likely not an efficient parameter compression, but also an effective structural regularizer for high dimensions. Dense edge-wise parameterization appears difficult to scale for the $784 \to 64$ setting, but the shared-atom approach remains both efficient and able to generalize well.

6. Interpretability

SA-KAN derives interpretability directly from the structure's parameterization rather than via an analytical component external to it. While typical KANs represent each edge independently with their own univariate functions and do not give insight into whether general mathematical motifs or simply isolated transformations are learned, SA-KAN solves this by forcing all edge functions into a common dictionary governed by sparse routing coefficients. This factorized architecture separates interpretation into two orthogonal components: the dictionary describes the space of functions learned by the layer (the basic operations it has discovered), and the coefficients tell the reader where those specific motifs are used within the layer (and how they are combined or reused). Compared to the diffuse, unstructured parameterization of Vanilla KANs (Liu et al., 2024b; Noorizadegan et al., 2025; Caruana, 1997; Zhang & Yang, 2021), SA-KAN therefore provides a significantly more interpretable and coherent representation of information. Our experiments demonstrate the value of this architectural property: in systems with obvious compositional structure SA-KAN leverages functional reuse to produce better error rates at lower cost. Efficiency and interpretability, in this case, are co-optimization criteria. But on the singular task of the gravitational force SA-KAN actually shows increased error over Vanilla KAN because a low-rank dictionary compromises fidelity over global structure. However this limitation means SA-KAN also acts as a good "structural" probe for measuring the existence of an underlying compositional structure, especially when contrasted against other methods which don't directly account for it.

7. Conclusion

We presented Shared-Atom Kolmogorov-Arnold Networks (SA-KAN), a factorized architecture that resolves the severe parameter scaling bottlenecks of standard KANs. By replacing dense, edge-wise functions with a globally shared dictionary and sparse multilinear routing, SA-KAN cleanly decouples basis expressivity from network connectivity. Our evaluations confirm that this structural low-rank prior reduces parameter budgets by up to 25% while simultaneously improving approximation fidelity across symbolic regression, complex PDE surrogate modeling, and high-dimensional image classification. Ultimately, SA-KAN provides a highly scalable, parameter-efficient, and inherently interpretable foundation, advancing the practical viability of Kolmogorov-Arnold Networks for rigorous scientific machine learning.

References

Aghaei, A. A. rkan: Rational Kolmogorov-Arnold networks. *arXiv preprint arXiv:2406.14495*, 2024.

Aghaei, A. A. fkan: Fractional Kolmogorov-Arnold networks with trainable jacobi basis functions. *Neurocomputing*, pp. 129414, 2025.

Arnol'd, V. I. On functions of three variables. *Doklady Akademii Nauk SSSR*, 114:679–681, 1957.

Bozorgasl, Z. and Chen, H. Wav-kan: Wavelet Kolmogorov-Arnold networks. *arXiv preprint arXiv:2405.12832*, 2024.

Caruana, R. Multitask learning. *Machine Learning*, 28: 41–75, 1997.

Chao, Z., Liu, X., Wu, Z., and Li, X. Rbf-kan: Radial basis function-Kolmogorov-Arnold networks. *IEEE Internet of Things Journal*, 2026.

Chiu, S. T., Cheung, S. W., Braga-Neto, U., Lee, C. S., and Li, R. P. Free-rbf-kan: Kolmogorov-Arnold networks with adaptive radial basis functions for efficient function learning. *arXiv preprint arXiv:2601.07760*, 2026.

De Lathauwer, L., De Moor, B., and Vandewalle, J. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Delis, A. Fasterkan. https://github.com/AthanasiosDelis/faster-kan, 2024. GitHub repository.

Gao, Y., Ng, M. K., and Tan, V. Y. F. Low tensor-rank adaptation of Kolmogorov-Arnold networks. *arXiv preprint arXiv:2502.06153*, 2025.

Hou, Y. and Zhang, D. A comprehensive survey on Kolmogorov-Arnold networks (kan). *arXiv preprint arXiv:2407.11075*, 2024.

Howard, A. A., Jacob, B., Murphy, S. H., Heinlein, A., and Stinis, P. Finite basis Kolmogorov-Arnold networks: domain decomposition for data-driven and physics-informed problems. *arXiv preprint arXiv:2406.19662*, 2024.

Kilmer, M. E. and Martin, C. D. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

Koenig, B. C., Kim, S., and Deng, S. Leankan: A parameter-lean Kolmogorov-Arnold network layer with improved memory efficiency and convergence behavior. *arXiv preprint arXiv:2502.17844*, 2025.

Kolmogorov, A. N. On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables. *Doklady Akademii Nauk SSSR*, 108(2):179–182, 1956.

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 SSSR*, 114:953–956, 1957.

LeCun, Y., Cortes, C., and Burges, C. J. C. The mnist database of handwritten digits, 1998. Online dataset.

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

Liu, Z., Ma, P., Wang, Y., Matusik, W., and Tegmark, M. Kan 2.0: Kolmogorov-Arnold networks meet science. *arXiv preprint arXiv:2408.10205*, 2024a.

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

Noorizadegan, A., Wang, S., Ling, L., and Dominguez-Morales, J. P. A practitioner's guide to Kolmogorov-Arnold networks. *arXiv preprint arXiv:2510.25781*, 2025.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Seydi, S. T. Exploring the potential of polynomial basis functions in Kolmogorov-Arnold networks: A comparative study of different groups of polynomials. *arXiv preprint arXiv:2406.02583*, 2024.

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

Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

Ta, H. T., Thai, D.-Q., Tran, A. N. T., Sidorov, G., and Gelbukh, A. F. Prkan: Parameter-reduced Kolmogorov-Arnold networks. *arXiv preprint arXiv:2501.07032*, 2025.

Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pfl"uger, D., and Niepert, M. PDEBench: An Extensive Benchmark for Scientific Machine Learning. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks*, 2022. URL https://arxiv.org/abs/2210.07182.

Tucker, L. R. Implications of factor analysis of three-way matrices for measurement of change. *Problems in Measuring Change*, 15(122-137):3, 1963.

Wang, Y., Siegel, J. W., Liu, Z., and Hou, T. Y. On the expressiveness and spectral bias of kans. *arXiv preprint arXiv:2410.01803*, 2024.

Wang, Y., Sun, J., Bai, J., Anitescu, C., Eshaghi, M. S., Zhuang, X., Rabczuk, T., and Liu, Y. A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov-Arnold networks. *Computer Methods in Applied Mechanics and Engineering*, 433:117518, 2025.

Xu, J., Chen, Z., Li, J., Yang, S., Wang, W., Hu, X., and Ngai, E. C. H. Fourierkan-gcf: Fourier Kolmogorov-Arnold network—an effective and efficient feature transformation for graph collaborative filtering. *arXiv preprint arXiv:2406.01034*, 2024.

Zhang, Y. and Yang, Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.