

Softly Symbolifying Kolmogorov–Arnold Networks

James Bagrow^{1,2,*} and Josh Bongard^{3,2}

¹Mathematics & Statistics, University of Vermont, Burlington, VT, United States

²Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States

³Computer Science, University of Vermont, Burlington, VT, United States

*Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

October 22, 2026

Abstract Kolmogorov–Arnold Networks (KANs) offer a promising path toward interpretable machine learning: their learnable activations can be studied individually while collectively fitting complex data accurately. In practice, however, trained activations often lack *symbolic fidelity*, learning pathological decompositions with no meaningful correspondence to interpretable forms. We propose Softly Symbolified Kolmogorov–Arnold Networks (S2KAN), which integrate symbolic primitives directly into training. Each activation draws from a dictionary of symbolic and dense terms, with learnable gates that sparsify the representation. Crucially, this sparsification is differentiable, enabling end-to-end optimization, and uses an optimization objective inspired by minimum description length. S2KAN can discover interpretable activation functions while retaining dense splines for flexible approximation. We introduce a quantitative measure of symbolic fidelity, assessing how closely learned activations can be approximated by short expressions from a specified symbolic language. Across real-world prediction tasks, dynamical-system forecasting, and synthetic symbolic benchmarks, S2KAN generally achieves higher symbolic fidelity and often far smaller models, while maintaining competitive predictive accuracy on real-world data and low prediction errors on most symbolic benchmark configurations. We also examine the models’ learned responses on scientific data, including the contributions of their symbolic components. Together, these results show that incorporating symbolic structure during training can yield compact, accurate KAN models whose learned responses can be examined through their symbolic components.

Keywords— differentiable sparsity, symbolic regression, minimum description length, dynamical systems, interpretable neural networks, scientific machine learning

1 Introduction

Neural networks and deep learning are powerful but opaque tools for modeling complex systems (Adadi & Berrada, 2018; Guidotti et al., 2018; Lipton, 2018; Rudin, 2019). In scientific problems, accurate predictions are often not enough, and practitioners seek interpretable models that reveal underlying mechanisms, suggest novel hypotheses, or confirm existing theories (Hempel, 1965; Shmueli, 2010; Woodward, 2004). Such interpretability can be supported by parsimony, seeking smaller or less parameterized models (Brunton et al., 2016; Hastie et al., 2015; Louizos et al., 2018; Schwarz, 1978), or through the use of interpretable building blocks, components such as mathematical functions or symbolic expressions that can be interrogated and understood individually (Cranmer, 2023; Cranmer et al., 2020; La Cava et al., 2021; Schmidt & Lipson, 2009; Udrescu & Tegmark, 2020). These avenues are not mutually exclusive, and in fact the most interpretable models often combine both: sparse combinations of symbolic primitives that are simultaneously compact and semantically meaningful.

Kolmogorov–Arnold Networks (KANs) (Liu, Wang, et al., 2025) have emerged as a promising alternative to traditional neural networks, replacing fixed activation functions with learnable univariate functions on each edge. Using flexible representations for the activation functions allows KANs to be highly performant (Bagrow & Bongard, 2025; Liu, Wang, et al., 2025; Panahi et al., 2025). But this design also lets practitioners examine the internal workings of the KAN by inspecting the learned activations, and possibly infer mathematical expressions (e.g., $\sin(x)$, e^x , x^2) for those activations (Liu, Tegmark, et al., 2025; Liu, Wang, et al., 2025), although interpreting many such functions becomes challenging in large KANs. This combination of accuracy and interpretability makes KANs attractive for scientific machine learning (Carleo et al., 2019; Karniadakis et al., 2021; Xu et al., 2021).

However, the standard KAN reliance on flexible dense representations is at odds with interpretability. Splines, the typical choice of representation, can fit arbitrary shapes, but the resulting activations may bear no resemblance to recognizable

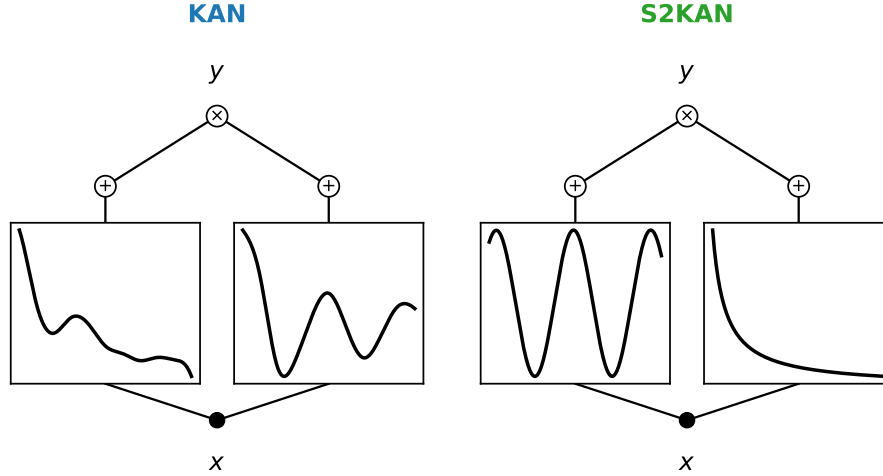


Figure 1: Learning $y = \text{sinc}(x) = \sin(x)/x$ with a multiplicative Kolmogorov–Arnold network (KAN). The standard KAN lacks symbolic fidelity, exhibiting a numerically accurate but otherwise pathological decomposition, whereas S2KAN captures the underlying sinc decomposition. The two models have empirical symbolic fidelities of 0% and 100%, respectively; see Methods (App. A) for evaluation details.

mathematical functions. The standard remedy—post-hoc symbolification, where each trained activation is independently fitted to a symbolic form (Liu, Wang, et al., 2025)—is problematic: splines may learn shapes difficult to fit symbolically, each activation is converted without considering network-level effects, and symbolic forms are not explored during training. The result: learned activations often lack *symbolic fidelity* (SF): accurate predictions but no meaningful correspondence to interpretable expressions (Fig. 1). What is needed is a method that treats symbolic and dense representations on equal footing, automatically selects among them while pruning unnecessary components, and does so continuously during training—yielding compact models with more symbolically expressible activation functions while retaining competitive accuracy.

We propose **Softly Symbolified KANs (S2KAN)**, which integrate symbolic primitives and sparse bases directly into the training process through a differentiable gating mechanism. Our key contributions are:

- An activation function dictionary combining dense representations, sparse basis functions, and symbolic primitives;
- A differentiable training process that jointly fits, symbolifies, and sparsifies the network;
- An objective motivated by minimum description length (MDL), balancing predictive accuracy and model complexity;
- A formal definition and quantitative measure of symbolic fidelity, distinguishing predictive accuracy from the simplicity of learned activation functions.

Our method combines symbolic and dense representations: S2KAN can select sparse symbolic forms when they suffice, while retaining dense terms to approximate functions outside the supplied symbolic dictionary. Rather than maximizing accuracy alone, S2KAN exposes the tradeoff between predictive performance and model complexity—achieving accuracy competitive with baseline KANs, often with far smaller models.

The rest of this paper is organized as follows. Section 2 describes Kolmogorov–Arnold networks and the standard symbolification method. Section 3 formalizes the notion of symbolic fidelity. Section 4 presents the S2KAN method: the activation function dictionary, differentiable gating mechanism, and MDL-inspired loss function. Section 5 evaluates S2KAN on real-world prediction tasks, chaotic dynamical systems, and symbolic benchmarks. Section 6 examines gate dynamics during training. We conclude with a discussion in Sec. 7.

2 Background

2.1 Kolmogorov–Arnold Networks

Motivated by the Kolmogorov–Arnold Representation Theorem (Arnold, 2009; Kolmogorov, 1957, 1961), a KAN with L layers and shape $[n_0, n_1, \dots, n_L]$ computes

$$x_j^{(\ell+1)} = \sum_{i=1}^{n_\ell} \phi_{\ell ij}(x_i^{(\ell)}) \quad (1)$$

where $\phi_{\ell ij} : \mathbb{R} \rightarrow \mathbb{R}$ is a learnable univariate activation function on the edge from neuron i in layer ℓ to neuron j in layer $\ell + 1$. The network output results from composing each layer of activation functions in sequence:

$$\text{KAN}(\mathbf{x}) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)(\mathbf{x}), \quad (2)$$

where $\mathbf{x}_{\ell+1} = \Phi_\ell(\mathbf{x}_\ell)$ is the matrix form of (1) and Φ_ℓ is the matrix containing the ϕ 's connecting layer ℓ and $\ell + 1$.

The $\phi_{\ell ij}$ can be parameterized in many ways, including radial basis functions (Li, 2024), Fourier series (GistNoesis, 2024) or sinusoidal functions (Reinhardt et al., 2025), wavelets (Bozorgasl & Chen, 2024), and Chebyshev polynomials (Sidharth SS et al., 2024). The original KAN formulation (Liu, Tegmark, et al., 2025; Liu, Wang, et al., 2025) used a combination of a B-spline and a fixed base function $b(x)$, typically the sigmoid linear unit (SiLU):

$$\phi(x) = w_b b(x) + w_s \text{spline}(x), \quad (3)$$

where w_b and w_s are learnable scale parameters, and

$$\text{spline}(x) = \sum_{m=1}^{G+K} c_m B_m(x), \quad (4)$$

where B_m are B-spline basis functions of order K over G grid intervals, and c_m are learnable coefficients. The nonpolynomial base function $b(x)$ supplements the piecewise-polynomial spline representation.

When using B-splines, KANs also implement grid updates, where the spline knots are periodically adjusted to better cover the activation function's input range, and grid refinement, where the number of knots is increased during training; see Liu, Wang, et al. (2025) for details. KAN 2.0 (Liu, Tegmark, et al., 2025) introduces multiplication nodes (Fig. 1), which compute products of sums of incoming activations, enabling more compact representations of multiplicative functions.

Standard KANs can be symbolified after training by fitting their learned activations to symbolic functions (Liu, Tegmark, et al., 2025; Liu, Wang, et al., 2025). Predictive accuracy alone does not ensure that these activations admit simple symbolic approximations. This motivates both our symbolic-fidelity measure and the inclusion of symbolic primitives during training.

3 Symbolic fidelity

We now formalize the notion of symbolic fidelity introduced informally above.

Let $G = (V, E)$ be the graph of a KAN of shape $[n_0, \dots, n_L]$: each node sums its incoming edges (a multiplication unit is a product of such summation subnodes), and each edge e carries one learned univariate Borel function ϕ_e ; $\phi := (\phi_e)_{e \in E}$ is the network's *decomposition*. Let μ be an input distribution on $\mathbb{R}^{n_0}$ and μ_e^ϕ the distribution of the preactivation of edge e (the pushforward of μ to the input of ϕ_e); assume $\phi_e \in L^2(\mu_e^\phi)$ (bounded on the support suffices). Let $f = \Phi(\phi) \in L^2(\mu)$ be the network output. Edge e is *functionally active* for the purposes of symbolic fidelity if

$$\text{Var}_{\mu_e^\phi}(\phi_e) > \eta \text{Var}_\mu(z_j) \quad \text{and} \quad \text{Var}_{\mu_e^\phi}(\phi_e) > \delta \text{Var}_\mu(f), \quad (5)$$

where z_j is the value of the summation (sub)node that e enters, $\eta > 0$ is a fixed fraction, and $\delta \geq 0$ is a numerical functional-activity floor. Write $E_{\eta, \delta}(\phi) \subseteq E$ for this set. Our computations use $\eta = 10^{-2}$ and $\delta = 10^{-8}$; setting $\delta = 0$ recovers the functional-activity rule without the numerical floor. This functional-activity criterion is distinct from gate activity: an edge with an open gate need not contribute to the SF denominator.

A *symbolic language* $\Sigma := (P, \mathcal{T}, J, \epsilon)$ consists of P , a finite set of Borel functions, each defined on a subset of $\mathbb{R}$ (e.g. x , $\sin$, $\exp$, $\log$, $1/x$), with $x \in P$; $\mathcal{T} := \{x \mapsto a g(bx + c) + d : a, b \neq 0\}$ acting on functions g ; a budget $J \in \mathbb{N}$; and a tolerance $\epsilon \in [0, 1)$. The *expressions* C form the smallest set containing the constants and $\mathcal{T}P$ and closed under $+$, $\cdot$, and $\circ$, each expression defined where its constituents are; $C_J \subseteq C$ are those using at most J occurrences of elements of P .

The evaluation primitives P need not coincide with the symbolic primitives supplied during S2KAN training (Sec. 4.1). Within each comparison, both methods are evaluated using the same language, budget, and tolerance; a spline activation can therefore qualify regardless of its parameterization. An activation consisting of a sum of at most J affinely transformed primitives from the evaluation language is expressible by construction. Overlap with the training dictionary can therefore favor representations built from those primitives; SF measures expressibility in the declared language rather than dictionary-independent interpretability.

Definition 1 (Expressible edge). *Given the decomposition ϕ , edge e is expressible with respect to Σ if some $s \in C_J$, defined μ_e^ϕ -a.e., satisfies*

$$\|\phi_e - s\|_{L^2(\mu_e^\phi)} \leq \epsilon \|\phi_e - \mathbb{E}_{\mu_e^\phi} \phi_e\|_{L^2(\mu_e^\phi)}. \quad (6)$$

The norm on the right is the standard deviation of ϕ_e , so for a non-constant edge this is $R_{\mu_e^\phi}^2(\phi_e, s) \geq 1 - \epsilon^2$; the other edges of ϕ enter only through μ_e^ϕ . Write $\Lambda(\phi) \subseteq E$ for the set of expressible edges.

Definition 2 (Symbolic fidelity). *The symbolic fidelity of a KAN with respect to μ, η, δ , and Σ is $|\Lambda(\phi) \cap E_{\eta, \delta}(\phi)| / |E_{\eta, \delta}(\phi)|$, the fraction of its functionally active edges that are, on the values that reach them, within tolerance of a short closed form; it is taken as 1 if $E_{\eta, \delta}(\phi) = \emptyset$ (a vacuous convention, not evidence of symbolic recovery; an empty functional-activity set does not by itself imply a constant network). The network is fully faithful if its symbolic fidelity is 1, i.e. $E_{\eta, \delta}(\phi) \subseteq \Lambda(\phi)$. Constants are included in the symbolic language, but functionally inactive edges do not contribute to the fidelity score; thus fully faithful means that every functionally active edge is expressible.*

SF measures expressibility of individual activation functions, not the readability of their composition: even a fully faithful network can produce a cumbersome overall expression.

To estimate symbolic fidelity empirically, we fit short sums of affinely transformed primitives to sampled responses of each functionally active edge and report the fraction meeting the prescribed approximation tolerance. This finite search considers additive expressions only and need not find every edge expressible in the broader formal language. The procedure builds on the affine symbolic fitting used in KAN post-hoc symbolification (Liu, Tegmark, et al., 2025; Liu, Wang, et al., 2025), extending it to short sums of primitives for evaluation without replacing the learned activations. The sampling and fitting procedures are described in Methods (App. A).

4 S2KAN: softly symbolified KANs

Instead of learning activations and converting them post hoc, we propose learning activation functions as combinations of the standard KAN components and other symbolic terms. Depending on the choice of terms, the activation functions can form an overcomplete dictionary of functions (Sec. 4.1) so sparsification will be necessary to avoid overfitting and stabilize training. Sparse basis selection can be approached with ℓ_1 regularization, as in the least absolute shrinkage and selection operator (LASSO). However, simply adding an ℓ_1 penalty to the loss optimized by Adam generally shrinks coefficients without producing exact zeros (Ziyin & Wang, 2023), whereas we seek explicit term selection during end-to-end training across the composed layers of a KAN. We therefore introduce binary gating variables that select which terms are active, made differentiable via a continuous relaxation of the ℓ_0 norm (Sec. 4.2), with selection guided by an objective inspired by minimum description length (Sec. 4.3).

4.1 Activation function dictionary

We organize the activation function using three categories of terms (or atoms):

$$\phi(x) = \sum_{s \in \mathcal{S}} z_s c_s \psi_s(x) + \sum_{f \in \mathcal{F}} z_f c_f \psi_f(x) + \sum_{r \in \mathcal{R}} z_r \phi_r(x; \mathbf{c}_r), \quad (7)$$

where the $z \in \{0, 1\}$ are binary gates selecting active terms (Sec. 4.2), $c \in \mathbb{R}$ are learnable coefficients, and each $\psi : D_\psi \rightarrow \mathbb{R}$, with $D_\psi \subseteq \mathbb{R}$, is a candidate univariate function. Here $\mathcal{S}$ indexes the symbolic library, $\mathcal{F}$ indexes sparse function bases, and $\mathcal{R}$ indexes dense parameterized representations. Equation (7) allows researchers to design problem-specific combinations as needed. Using the standard KAN B-spline dense representation for $\mathcal{R}$, setting $\mathcal{S} = \mathcal{F} = \emptyset$, and fixing $z_r = 1$ for all $r \in \mathcal{R}$, Eq. (7) recovers the original KAN formulation.

In this work, we consider the following dictionary:

Symbolic library ($\mathcal{S}$) A collection of elementary functions such as $1, x, \sin(x), 1/x, \log|x|, \dots$. Each function receives an independent gate and coefficient.

Sparse function bases ($\mathcal{F}$) Families of orthogonal or structured functions indexed by degree/frequency:

- *Chebyshev polynomials:* P terms $T_p(x)$, $p = 0, 1, \dots, P-1$, computed via recurrence.

- *Fourier basis*: $\{\sin(qx), \cos(qx)\}$ for mode $q = 1, \dots, Q$.

Setting $P = 0$ or $Q = 0$ removes the Chebyshev or Fourier family, respectively. Each term receives an independent gate and coefficient.

Dense representations ($\mathcal{R}$) A single gate controls an entire parameterized function class:

- *B-spline with SiLU residual*: $\phi_{\text{spline}}(x; \mathbf{c}) = w_b \text{SiLU}(x) + w_s \sum_{m=1}^B c_m B_m(x)$, where $B = G + K$, the B_m are B-spline basis functions, and $\mathbf{c} = (w_b, w_s, c_1, \dots, c_B)$ contains the two learnable scales and the spline coefficients.

This formulation allows for graceful degradation: when symbolic terms or their composition within the chosen architecture are insufficient, the model can fall back to the dense spline representation.

4.2 Differentiable sparsity via Hard Concrete Distribution

We follow the ℓ_0 regularization approach of Louizos et al. (2018), which leverages the reparameterization trick to provide a continuous, differentiable relaxation of binary gates while maintaining the ability to produce exact zeros and ones.

For each gate i , introduce a learnable parameter $\alpha_i \in \mathbb{R}$ and define a continuous relaxation of a binary variable via the Concrete (Gumbel-Softmax) distribution with temperature $\tau > 0$:

$$s_i = \sigma \left(\frac{\log u - \log(1 - u) + \alpha_i}{\tau} \right), \quad u \sim \text{Uniform}(0, 1), \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid function. This provides a continuous relaxation in $(0, 1)$, but cannot produce exact boundary values. To assign finite probability mass to exactly 0 and 1, apply a stretched and rectified transformation with parameters $\gamma < 0 < 1 < \zeta$ (typically $\gamma = -0.1, \zeta = 1.1$):

$$\bar{s}_i = s_i(\zeta - \gamma) + \gamma, \quad (9)$$

$$\tilde{z}_i = \min(1, \max(0, \bar{s}_i)). \quad (10)$$

The stretch extends the range to (γ, ζ) before clipping to $[0, 1]$, allowing the relaxed gate to reach the boundaries exactly during training.

The probability that this gate is nonzero, used for regularization (Sec. 4.3), has a closed form:

$$p_i := \Pr(\tilde{z}_i > 0) = \sigma \left(\alpha_i - \tau \log \frac{-\gamma}{\zeta} \right). \quad (11)$$

This is the expected active-gate indicator, $p_i = \mathbb{E}[\mathbb{1}_{\{\tilde{z}_i > 0\}}]$, rather than the expected amplitude $\mathbb{E}[\tilde{z}_i]$ of the relaxed gate. During training, sample $\{\tilde{z}_i\}$ and update their distribution parameters jointly with the other model parameters. At inference, deterministically threshold: $z_i = \mathbb{1}_{\{p_i > 1/2\}}$.

The gradient of the expected ℓ_0 penalty with respect to gate parameters is

$$\frac{\partial p_i}{\partial \alpha_i} = p_i(1 - p_i). \quad (12)$$

This gradient is maximal when $p_i = 1/2$ and vanishes as p_i approaches zero or one.

Remarks An activation function sparsifies as its gates close, and the entire function can be omitted when all its gates close, allowing for architectural sparsification. Beyond the mechanism of sparsity, these gates offer additional benefits. Initializing α_i acts as a prior, and one can tune the model towards or away from symbolic entries by, for instance, setting $\alpha_i \approx -1$ for dense terms and $\alpha_i \approx 0$ otherwise. Gate probabilities p_i also provide a structural convergence criterion: training can terminate, possibly with some patience, once gates become decisive (e.g., when most $p_i < 0.01$ or $p_i > 0.99$).

4.3 MDL-motivated loss function

To balance model accuracy while minimizing complexity, our training objective is motivated by the MDL principle, which seeks to minimize the total bits required to encode both the model and the data given the model:

$$\mathcal{L}_{\text{MDL}} = \mathcal{L}_{\text{model}} + \mathcal{L}_{\text{data}|\text{model}}. \quad (13)$$

The first term captures model complexity while the second encodes residuals. Assuming independent and identically distributed (iid) scalar residuals $\epsilon_t = y_t - \hat{y}_t \sim \mathcal{N}(0, \sigma^2)$, the data encoding term, $\mathcal{L}_{\text{data}|\text{model}}$, is proportional to the negative log-likelihood:

$$\mathcal{L}_{\text{data}|\text{model}} \propto \frac{n}{2} \log \sigma^2 + \frac{1}{2\sigma^2} \sum_{t=1}^n (y_t - \hat{y}_t)^2. \quad (14)$$

The maximum-likelihood estimate is the mean squared error (MSE), $\hat{\sigma}^2 = \text{MSE}(y, \hat{y})$. Substituting this estimate gives a per-sample data term $\frac{1}{2} \log \text{MSE}(y, \hat{y})$, up to additive constants. For $\mathcal{L}_{\text{model}}$, the model complexity term, let

$$k := \sum_{\ell ij} \sum_m p_{\ell ijm} \quad (15)$$

be the expected number of active terms over the activation functions in the network, where the nonzero-gate probabilities $p_{\ell ijm}$ are given by Eq. (11). Motivated by the Bayesian information criterion (BIC) (Schwarz, 1978), we assign the unweighted support penalty $\mathcal{L}_{\text{model}} = \frac{k}{2} \log n$. Combining these terms with the complexity weights defined below and a stabilized minibatch data term gives the implemented training objective

$$\mathcal{L} = \frac{1}{2} \log(\text{MSE}_{\mathcal{B}} + c_{\text{loss}}) + \beta \frac{k_w \log n}{2n}, \quad (16)$$

where hyperparameter $\beta \geq 0$ controls the sparsity-accuracy tradeoff, $\mathcal{B}$ is the current minibatch, n is the full fitting-set size, and k_w is the weighted expected support defined below. MSE is averaged over observations and output coordinates. The additive constant $c_{\text{loss}} = 10^{-12}$ stabilizes the natural logarithm and is expressed in squared units of the training target.

For deterministic, full-data scalar Gaussian regression with a discrete parameter count, the corresponding unstabilized objective at $\beta = 1$ equals BIC divided by $2n$, up to additive constants. Minibatch evaluation, stochastic gates, relaxed support counts, and pooled multioutput errors (for vector-valued models) make the implemented objective an MDL/BIC-motivated regularizer rather than exact BIC model selection or a complete description-length code. Accordingly, $\beta = 1$ serves as a motivated reference setting, while β remains tunable and may require problem-specific adjustment. The iid Gaussian assumption also need not hold for every dataset, particularly temporally dependent dynamical-system observations. More rigorous coding schemes, such as sequential or normalized maximum likelihood coding (Grünwald, 2007), are not implemented here.

4.3.1 Complexity-weighted sparsity

The unweighted count k assigns a common cost to each active term. However, when selecting among terms with varying structural complexity, we may sometimes wish to account for differential encoding costs, which we can do by incorporating *complexity weights*. For term ψ_m with complexity weight $w_m > 0$, the weighted number of active terms is

$$k_w := \sum_{\ell ij} \sum_m w_m p_{\ell ijm}, \quad (17)$$

as used in Eq. (16). The gradient $\partial k_w / \partial \alpha_m = w_m p_m (1 - p_m)$ shows that higher complexity weights will induce stronger regularization pressure on the corresponding gates. Appropriate choices for w_m include uniform weighting ($w_m = 1$, recovering the unweighted support penalty), degree-dependent complexity ($w_m = 1 + \log_2(d + 1)$ for degree d), or computational cost ($w_m \propto$ the number of floating-point operations, or FLOPs, required to evaluate ψ_m). Hand-tuning w_m is also common practice in symbolic regression. We assign weight one to each non-dense term, corresponding to its learned coefficient. A dense branch receives weight equal to its number of spline coefficients plus two scale parameters; see Methods (App. A). Thus k_w is a differentiable surrogate for retained parameter count, rather than a count of statistically identifiable parameters. We do not pursue weighting schemes other than this, but it may be beneficial to consider them in the future.

We discuss further setup and training details in Methods (App. A).

5 Results

We first compare predictive accuracy, model size, and symbolic fidelity on four real-world datasets. We then examine the scientific consistency of their predicted responses and connect selected responses to learned symbolic activations. Next, we evaluate forecasting and dynamical behavior in two chaotic systems, followed by the Nguyen symbolic regression benchmarks.

5.1 Real-world data

We apply S2KAN to four real-world datasets:

Airfoil self-noise Predict the scaled sound pressure level (in dB) generated by an airfoil from five inputs: frequency (Hz), angle of attack (degrees), chord length (m), free-stream velocity (m/s), and suction-side displacement thickness (m). The University of California, Irvine (UCI) dataset contains 1503 observations from National Aeronautics and Space Administration (NASA) wind-tunnel tests of National Advisory Committee for Aeronautics (NACA) 0012 airfoils of different sizes and at different flow conditions (Brooks et al., 1989). We use the five supplied features without additional derived features. For modeling, we allocated 1000 observations to fitting and validation (800 and 200, respectively), with the remaining 503 observations used for testing.

Concrete compressive strength Predict the compressive strength (in MPa) of concrete based on sample properties. The dataset contains 1030 samples with eight features: cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate (all in kg/m^3), and age (days). Compressive strength in concrete is well known to depend on the water-to-cement ratio (Neville, 2011), so we include this as a derived feature. We also compute total binder (cement plus slag and fly ash), total aggregate (coarse plus fine), and the water-to-binder ratio, which generalizes the water-to-cement relationship when supplementary cementitious materials are present. We include $\log(\text{age} + 1)$ as strength gain in concrete follows an approximately logarithmic relationship with curing time (Neville, 2011), motivating the inclusion of transformed age variables. For modeling, we allocated 824 observations to fitting and validation (659 and 165, respectively), with the remaining 206 observations used for testing. Data were collected by Yeh (1998, 2006).

Power plant Predict the net hourly electrical power output (in MW) of a combined-cycle power plant from four inputs: ambient temperature ($^{\circ}\text{C}$), exhaust vacuum (cm Hg), ambient pressure (mbar), and relative humidity (%). The UCI dataset contains 9568 observations collected during full-load operation over 2006–2011; the inputs are hourly averages of sensor measurements (Tüfekci, 2014). We use the four supplied features without additional derived features. For modeling, we sampled 1000 observations for fitting and validation (800 and 200, respectively) and 1000 separate observations for testing.

Superconductor critical temperature Predict critical temperature (in K) of superconductors based on their material properties. The original dataset includes many features and derived statistics; we focus on five that capture composition, electronic structure, and bonding: number of elements, weighted mean valence, valence entropy, weighted mean first ionization energy, and mean electron affinity. These same representative features were used in prior KAN modeling (Bagrow & Bongard, 2025). For modeling, we sampled 1000 observations for fitting and validation (800 and 200, respectively) and 1000 separate observations for testing. The data come from Japan’s National Institute for Materials Science superconductor database (Center for Basic Research on Materials, 2024; Hamdiah, 2018).

Details of data preprocessing, model fitting, and validation-based checkpoint selection are given in Methods (App. A). Baseline KAN retains the default MSE loss of the authors’ implementation (Liu, Wang, et al., 2025), whereas S2KAN uses the log-MSE-based objective in Eq. (16).

As shown in Table 1, across all four datasets, S2KAN found considerably more compact representations at $\beta = 1$, with no spline terms retained for concrete or power plant. For concrete, median test R^2 was 0.894 versus 0.915 for baseline KAN, with only 61 versus 448 active functions. For superconductivity, median test R^2 was 0.682 versus 0.696, with 61.3 versus 450 effective parameters. S2KAN achieved slightly higher median test R^2 with smaller models on airfoil and power plant.

S2KAN also achieved higher median symbolic fidelity on all four datasets at all six settings of expression budget and approximation tolerance. This fidelity advantage persists when baseline KAN is trained with log MSE (App. G.1).

Appendix G.3 examines how predictive accuracy and model size vary with β ; App. G.2 compares full S2KAN with no-dense and dense-only variants.

Table 1: Prediction, model size, and symbolic fidelity on four real-world datasets. S2KAN yields substantially smaller models and higher symbolic fidelity, with modest differences in predictive accuracy. Test accuracy is reported as R^2 and root mean squared error (RMSE). Values are medians over ten data splits with three initialization seeds per split; brackets are interquartile ranges (IQRs; omitted for constant values). S2KAN uses $\beta = 1$ after warmup. “Active” counts gated-on activation functions, and k denotes the effective parameter count k_w from Eq. (17). Symbolic fidelity (SF, percent; Section 3) is the fraction of functionally active edges approximated within relative tolerance ϵ using at most J terms.

Model	Test accuracy		Model size		SF (%), $\epsilon = 0.01$			SF (%), $\epsilon = 0.1$		
	R^2	RMSE	Active	k	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
Airfoil: shape [5, 5, 1]; RMSE in dB										
KAN	0.911 [0.902, 0.918]	2.05 [1.96, 2.13]	30	450	10.5 [7.2, 13.5]	26.8 [24.4, 28.6]	28.6 [28.6, 31.0]	14.3 [11.1, 17.7]	39.3 [36.0, 46.0]	59.6 [51.8, 62.1]
S2KAN	0.912 [0.909, 0.920]	2.02 [1.95, 2.05]	26	103.8 [94.7, 114.1]	22.4 [20.0, 25.7]	50.0 [46.3, 52.2]	64.0 [60.0, 69.2]	33.3 [29.4, 39.8]	84.3 [82.6, 87.5]	91.3 [87.5, 92.0]
Concrete: shape [13, 32, 1]; RMSE in MPa										
KAN	0.915 [0.898, 0.925]	4.90 [4.49, 5.23]	448	6720	0.0 [0.0, 0.3]	2.6 [1.9, 3.4]	12.9 [11.6, 13.9]	26.2 [19.3, 29.0]	61.5 [47.2, 65.1]	83.2 [71.5, 86.9]
S2KAN	0.894 [0.885, 0.906]	5.27 [5.00, 5.61]	61	76.2 [71.7, 80.4]	77.8 [73.7, 79.5]	85.4 [83.5, 88.6]	86.9 [85.1, 90.4]	82.1 [78.5, 85.6]	90.3 [89.1, 94.2]	94.4 [91.9, 96.1]
Power plant: shape [4, 5, 1]; RMSE in MW										
KAN	0.932 [0.930, 0.936]	4.41 [4.27, 4.50]	25	375	0.0	0.0	0.0 [0.0, 7.2]	12.8 [8.7, 20.0]	51.0 [43.6, 56.4]	74.5 [68.4, 80.3]
S2KAN	0.937 [0.931, 0.941]	4.19 [4.10, 4.47]	16	36.7 [34.3, 39.4]	35.5 [30.2, 39.6]	64.3 [59.1, 71.4]	69.9 [64.3, 79.6]	50.0 [46.2, 57.1]	92.6 [85.7, 93.9]	92.9 [92.3, 100.0]
Superconductor: shape [5, 5, 1]; RMSE in K										
KAN	0.696 [0.669, 0.727]	18.80 [18.22, 19.43]	30	450	0.0	0.0 [0.0, 3.7]	16.7 [15.3, 16.7]	3.3 [0.0, 3.6]	19.3 [16.7, 23.3]	33.9 [30.0, 36.7]
S2KAN	0.682 [0.637, 0.698]	19.51 [18.85, 20.27]	16	61.3 [51.8, 74.8]	19.4 [13.4, 27.5]	39.4 [30.1, 49.3]	53.3 [44.2, 60.2]	25.0 [15.7, 33.3]	55.6 [50.0, 65.6]	65.9 [57.1, 73.6]

5.2 Scientific response consistency and symbolic interpretation

We evaluated selected scientifically meaningful responses and physical admissibility across the four real-world datasets, using the same validation-selected checkpoints as Table 1: ten data splits and three initialization seeds per split. Unless stated otherwise, percentages below are medians of per-model fractions over the 30 fits per method, and split comparisons average the three initialization seeds. These exploratory analyses use frozen models without further training or symbolic refitting; the overlapping splits provide descriptive comparisons rather than independent experiments. Appendix D gives the diagnostic definitions, secondary comparisons, and sensitivity checks, including the Airfoil peak and speed analyses and the Power Plant pressure analysis.

Airfoil self-noise. For fixed airfoil geometry and flow conditions, the measured sound level varies with frequency and can decline after a spectral peak. Preserving these observed declines provides a response-consistency check: substantial upward reversals would contradict the measured trend. Because airfoil self-noise comprises several mechanisms with different spectral responses (Brooks et al., 1989), we restrict this check to recorded conditions exhibiting a clear decline. Among 106 recorded spectra, we identified 92 with a clear measured decline after their peak and evaluated the models over these observed frequency ranges. A pass required a predicted net decrease exceeding 1 dB and no upward reversal exceeding 1 dB, so flat predictions failed. S2KAN passed on a median 94.6% of spectra, compared with 70.1% for KAN, with all ten split averages favoring S2KAN. Held-out centered spectral errors were similar (1.40 versus 1.43 dB for S2KAN and KAN). The clearest advantage was therefore preservation of measured spectral decline; selection of the tails using observed responses makes this a descriptive diagnostic rather than a wholly held-out test.

Concrete compressive strength. As concrete cures, continuing cement hydration generally increases its compressive strength, with gains slowing at later ages. Substantial predicted strength declines at fixed composition are therefore inconsistent with this expected curing response. Across 181 fixed-composition recipes, S2KAN predicted strength declines exceeding 1 MPa for 8.8% of recipes, compared with 63.5% for KAN. Positive average growth with smaller late-age than early-age gains occurred for 93.4% versus 75.4% of recipes. Both comparisons favored S2KAN in all ten splits under either mean or median aggregation over seeds, despite higher test RMSE (5.27 versus 4.90 MPa). These related aging diagnostics sweep each recipe’s recorded age

range, changing age and the supplied $\log(\text{age} + 1)$ feature together; they assess response consistency rather than causal curing laws.

Power plant electrical output. For a combined-cycle power plant, warmer ambient air generally reduces the electrical power generated at full load (Ponce Arrieta & Silva Lora, 2005). Preserving this temperature penalty under small changes in ambient conditions provides a check of the model’s consistency with the expected plant response. We evaluated 325 local temperature sweeps spanning $\pm 2^\circ\text{C}$ around recorded conditions, holding the other inputs fixed and retaining sweeps within observed joint-input support. A pass required a net decrease exceeding 0.2 MW and no upward excursion exceeding 1 MW. The median passing fraction was 100.0% for S2KAN and 98.2% for KAN, with all ten split averages favoring S2KAN. Both methods usually captured the expected direction, making the absolute improvement modest. All 30 Power Plant S2KAN fits closed their dense branches, allowing their responses to be traced through explicit symbolic and finite-basis activations. In one fixed example, a decreasing polynomial temperature activation feeds an increasing linear-plus-sine output activation over the visited hidden-input range. This composition contributes a 0.92 MW decrease over $18.34\text{--}22.34^\circ\text{C}$, within the complete model’s 6.25 MW decrease. It illustrates how a learned expression implements the expected sign over a supported interval, without identifying a thermodynamic mechanism.

Superconductor critical temperature. The critical temperature T_c marks the onset of superconductivity and must be non-negative on the kelvin scale, so nonnegativity provides a basic check of physical admissibility. S2KAN produced nonphysical $T_c < 0$ predictions for 2.65% of held-out observations, versus 5.10% for KAN. After clipping negative predictions to zero for both methods, median test RMSE remained higher for S2KAN (19.44 versus 18.73 K).

The clearest response-consistency improvements are for concrete aging and airfoil spectral decline, with a smaller improvement for Power Plant temperature responses and fewer physically inadmissible Superconductor predictions. These diagnostics compare the complete training configurations, including their different losses and regularization, and do not isolate symbolic gating as the cause.

To illustrate how learned symbolic activations produce one of these responses, Fig. 2 traces the age response of one fitted S2KAN for a fixed concrete recipe. In the four leading branches, the raw-age edges are inactive and age dependence comes entirely from the supplied log-age feature, demonstrating selection of a domain-informed transformation available to both models. The dominant branch combines a linear function of standardized log-age with two Fourier terms. Over the hidden-input range visited here, this composition yields an increasing strength contribution with a progressively slower rate of gain; the trigonometric terms do not imply a periodic age response. This branch contributes 14.2 MPa of the complete model’s predicted 24.7 MPa increase from day 3 to day 365. This agreement holds over the evaluated age range; the Fourier terms do not enforce monotonic strength development under extrapolation.

5.3 Data-driven modeling of dynamical systems

We evaluate S2KAN on two dynamical systems that present challenging forecasting problems due to their chaotic attractors.

The first is the *Ikeda map* (Hammel et al., 1985; Ikeda, 1979), a discrete-time chaotic system arising from nonlinear optics that resists discovery by sparse regression methods like sparse identification of nonlinear dynamics (SINDy) (Brunton et al., 2016):

$$\begin{aligned} x_{n+1} &= 1 + \mu (x_n \cos(\phi_n) - y_n \sin(\phi_n)), \\ y_{n+1} &= \mu (x_n \sin(\phi_n) + y_n \cos(\phi_n)), \end{aligned} \quad (18)$$

where $\phi_n = 0.4 - 6(1 + x_n^2 + y_n^2)^{-1}$ and bifurcation parameter $\mu = 0.9$. KANs have been shown to model the Ikeda map effectively (Bagrow & Bongard, 2025; Panahi et al., 2025).

The second system is a continuous-time *three-species ecosystem*:

$$\begin{aligned} \frac{dN}{dt} &= N \left(1 - \frac{N}{K} \right) - x_p y_p \frac{NP}{N + N_0}, \\ \frac{dP}{dt} &= x_p P \left(y_p \frac{N}{N + N_0} - 1 \right) - x_q y_q \frac{PQ}{P + P_0}, \\ \frac{dQ}{dt} &= x_q Q \left(y_q \frac{P}{P + P_0} - 1 \right), \end{aligned} \quad (19)$$

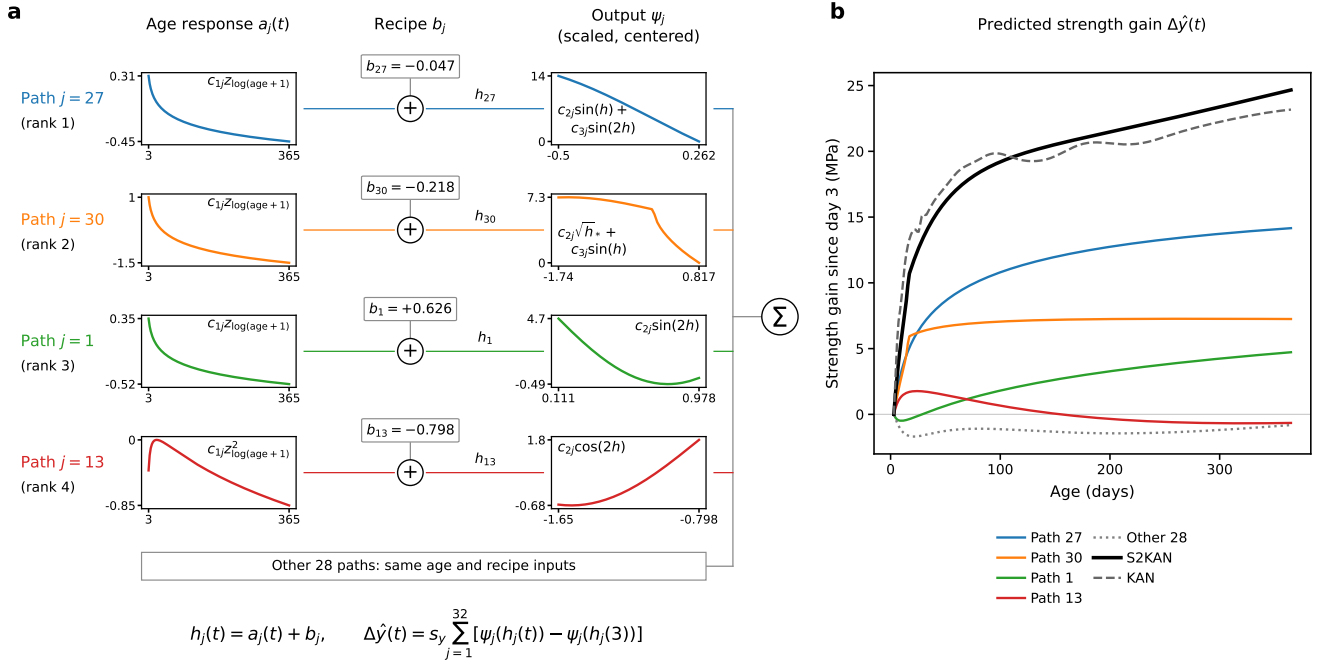


Figure 2: Illustrative model-response analysis of predicted concrete strength development with age. Consistent with expected curing behavior, the complete model predicts strength increasing with age at an overall slowing rate. The dominant branch exhibits a decreasing rate of gain throughout the evaluated interval. **a**, Four hidden branches for a fixed recipe, ranked by total variation of their output contributions over days 3–365. Each b_j sums the learned input activations evaluated at the recipe’s 11 fixed non-age features. Inset formulas show the learned functions, with branch-specific coefficients c_{kj} ; $z_{\log(\text{age}+1)}$ denotes standardized log-age, and $\sqrt{h_*}$ denotes the protected square root. These branches have no active raw-age or dense terms. Output curves are converted to MPa and centered at day 3. **b**, Branch contributions and complete S2KAN and KAN predictions relative to day 3. All 32 branches remain in the S2KAN total. Both models use the same data split and initialization seed; the recipe was selected by proximity to the median standardized ingredient composition among 181 eligible recipes, without using strengths or predictions. Here s_y is the population standard deviation of the fitting-set targets.

where N , P , and Q represent primary producer, herbivore, and carnivore populations, with carrying capacity K serving as the bifurcation parameter. We set $K = 0.98$, $x_p = 0.4$, $y_p = 2.009$, $x_q = 0.08$, $y_q = 2.876$, $N_0 = 0.16129$, and $P_0 = 0.5$ to produce chaotic dynamics (McCann & Yodzis, 1994).

Data for both systems were generated following Panahi et al. (2025); the partitions used here are specified in Methods (App. A). We compare KAN and S2KAN over ten initialization seeds per configuration. The S2KAN dictionaries combine generic primitives with rational terms motivated by the generating equations; these experiments therefore assess learning with a domain-informed expression language. Architectures, dictionaries, training settings, and Lyapunov-estimation procedures are given in Methods (App. A).

For the ecosystem only, the small changes between consecutive training observations motivate a “residual” formulation, $\hat{\mathbf{x}}_{n+1} = \mathbf{x}_n + f_\theta(\mathbf{x}_n)$, where f_θ learns the state increment. This parameterization requires no knowledge of the governing equations; the same fixed identity connection, which adds no trainable parameters, is applied to both KAN and S2KAN. Ikeda models predict the next state directly.

Table 2 summarizes the tradeoff between model size, predictive accuracy, and dynamical fidelity. S2KAN produces smaller models, with median symbolic fidelity meeting or exceeding KAN’s in 34 of 36 comparisons (94%). Preserving long-run dynamics is more sensitive to regularization for the ecosystem than for Ikeda. Appendix E reports full Lyapunov spectra, multi-step prediction errors and forecast horizons, state-distribution overlap, and autocorrelation and power spectral density comparisons.

At $\beta = 1$, S2KAN reduces the median effective parameter count from 720 to 287.5 for Ikeda and from 405 to 87.5 for the ecosystem, while matching or exceeding KAN’s median SF at all six settings. Median one-step RMSE decreases from 1.41×10^{-2} to 3.17×10^{-3} for Ikeda, but increases from 5.00×10^{-5} to 2.71×10^{-4} for the ecosystem. For Ikeda, this combines 60% fewer effective parameters with a median leading exponent close to the reference (0.5076 versus 0.5081), although the model from one of ten seeds converges to incorrect long-run behavior. For the ecosystem, the median leading Lyapunov exponent falls to approximately zero, substantially below the reference value of 0.00814.

Table 2: Model compression in chaotic dynamical systems. S2KAN yields smaller models with generally higher SF and lower one-step RMSE for Ikeda; stronger compression trades accuracy for size in the ecosystem. Entries are medians over ten initialization seeds; brackets are IQRs (omitted for constant values). The model-size measure k is defined in Eq. (17). λ_1 is the largest Lyapunov exponent (per iterate for Ikeda; per unit time for the ecosystem).

Model	β	Test accuracy	Dynamics	Model size		SF (%), $\epsilon = 0.01$			SF (%), $\epsilon = 0.1$		
		RMSE	λ_1	Active	k	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
Ikeda map: RMSE $\times 10^{-3}$; reference $\lambda_1 = 0.508$											
KAN	—	14.10 [10.74, 18.42]	0.508 [0.505, 0.511]	48	720.0	10.1 [4.9, 12.2]	27.7 [26.4, 40.3]	49.4 [44.0, 67.5]	56.7 [52.7, 62.7]	86.1 [81.8, 88.9]	97.6 [95.9, 99.4]
S2KAN	0	4.65 [3.83, 5.09]	0.508 [0.506, 0.512]	45	652.5	6.5 [5.1, 10.1]	25.7 [21.4, 30.3]	59.2 [55.9, 64.6]	60.3 [56.9, 62.6]	89.6 [82.9, 95.9]	100.0 [95.0, 100.0]
S2KAN	1	3.17 [2.60, 3.61]	0.508 [0.501, 0.512]	41.5	287.5	29.7 [25.8, 31.6]	66.6 [62.6, 68.6]	86.5 [86.1, 90.7]	67.9 [66.9, 70.1]	95.8 [94.6, 99.3]	100.0 [100.0, 100.0]
S2KAN	3	5.18 [3.33, 8.47]	0.507 [0.502, 0.508]	40	186.0	38.3 [36.9, 43.9]	78.8 [74.6, 82.4]	93.1 [91.2, 94.2]	81.6 [77.5, 83.7]	100.0 [97.1, 100.0]	100.0
Ecosystem: RMSE $\times 10^{-5}$; reference $\lambda_1 = 0.00814$											
KAN	—	5.00 [3.68, 6.68]	0.00666 [0.00092, 0.00861]	27	405.0	18.9 [11.7, 25.3]	66.7 [51.4, 81.7]	87.8 [85.2, 91.5]	83.3 [79.6, 84.6]	100.0 [95.9, 100.0]	100.0 [100.0, 100.0]
S2KAN	0	5.08 [4.87, 5.63]	0.00688 [0.00002, 0.00926]	25	288.5	51.2 [47.7, 61.5]	86.3 [85.0, 90.7]	95.5 [95.0, 100.0]	95.0 [90.9, 95.5]	100.0 [100.0, 100.0]	100.0
S2KAN	0.1	8.79 [7.91, 9.19]	0.00664 [-0.00002, 0.00909]	23	177.5	51.3 [50.0, 54.5]	88.6 [75.8, 93.2]	94.3 [90.2, 100.0]	92.5 [90.2, 94.7]	100.0	100.0
S2KAN	1	27.14 [23.93, 38.59]	0.00001 ^a [-0.00003, 0.00768]	22.5	87.5	68.2 [64.8, 71.7]	93.0 [90.9, 98.9]	100.0 [93.2, 100.0]	100.0 [95.5, 100.0]	100.0 [100.0, 100.0]	100.0

^a λ_1 uses 9/10 fits for the ecosystem at $\beta = 1$; all other entries use 10/10. Undefined Lyapunov estimates are excluded from medians and IQRs.

Weaker regularization improves the ecosystem’s prediction accuracy and median leading-exponent estimate, although recovery varies across seeds. Even without an explicit sparsity penalty ($\beta = 0$), S2KAN’s median effective parameter count is 9% below KAN for Ikeda and 29% below KAN for the ecosystem, with lower one-step error for Ikeda and nearly unchanged error for the ecosystem. Appendix E.4 shows the corresponding reductions in parameter count during training.

5.4 Nguyen symbolic regression benchmark

Next, we turn to the Nguyen symbolic regression benchmark (Nguyen et al., 2011). This benchmark contains a sample of mathematical functions designed to capture a variety of symbolic regression challenges. We focus in Table 3 on the first 10, the last two of which are bivariate. For each problem we generate 1024 training and 256 testing points. We apply three different architectures: small (S), with no hidden units; large (L), with one hidden layer of 3 summation units; and large-mult (LM), with one hidden layer of 3 summation and 1 multiplication units. Models were trained for up to 10,000 epochs with a batch size of 128. S2KAN’s first 200 epochs were warmup ($\beta = 0$). We report medians over ten initialization seeds on a fixed training/test sample; Appendix F reports the corresponding quartiles. Table 3 reports test R^2 , RMSE, model size, and symbolic fidelity for both methods, using $\beta = 1$ for S2KAN.

Across 180 comparisons of median symbolic fidelity (10 problems $\times$ 3 architectures $\times$ 6 fidelity settings), S2KAN exceeds KAN in 114 (63.3%), ties in 66 (36.7%), and loses in none; excluding the small F10 pair, whose S2KAN fits have no functionally active edges and receive 100% fidelity by convention, gives 108 wins out of 174 comparisons (62.1%), with 66 ties and no losses. These counts are descriptive, since the six fidelity settings are not independent.

Baseline KAN meets the median-RMSE threshold of 1% of test-target SD in 29 of 30 problem–architecture configurations, compared with 23 for S2KAN at $\beta = 1$. Across the 30 configuration medians, median normalized RMSE is 0.040% of target SD for KAN and 0.44% for S2KAN. Both methods fail on F10 with the small architecture; S2KAN additionally exceeds the threshold on F5, F7 and F8 with L and LM. The largest departures occur on F5, where S2KAN’s median test R^2 is 0.9517 for L and 0.8041 for LM; the latter has an interquartile range of [0.2745, 0.9171], indicating substantial variability across initializations. Compression is architecture-dependent. Across the ten problem-level medians, S2KAN reduces the median effective parameter count from 90.0 to 46.5 for L and from 135.0 to 62.8 for LM, but increases it from 15.0 to 19.8 for S.

Appendix G.2 tests the contribution of the dense branch by disabling it throughout training and inference. Appendix G.3 reports the corresponding accuracy–size sensitivity to β across all three Nguyen architectures.

Table 3: Nguyen benchmarks across architectures. S2KAN generally retains high predictive accuracy, with the largest departures from baseline on F5 for the large and large-mult architectures. Values are medians over ten initialization seeds; quartiles are reported in Table 10. Y and N indicate whether RMSE is at most 1% of the test-target standard deviation (SD). Shapes: S= $[n_0, 1]$, L= $[n_0, 3, 1]$, LM= $[n_0, (3, 1), 1]$ with multiplicative interactions. The model-size measure k is defined in Eq. (17).

ID	Expression / domain	Shape	Model	Test accuracy		Model size		SF (%), $\epsilon = 0.01$			SF (%), $\epsilon = 0.1$			
				R^2	RMSE ($\leq 1\%$ SD)	Active	k	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$	
F1	$x^3 + x^2 + x$ $x \in [-1, 1]$	S	KAN	1.0000	5.41×10^{-6} (Y)	1	15.0	0.0	100.0	100.0	0.0	100.0	100.0	
			S2KAN	1.0000	8.06×10^{-4} (Y)	1	20.1	0.0	100.0	100.0	0.0	100.0	100.0	
		L	KAN	1.0000	5.50×10^{-4} (Y)	6	90.0	0.0	0.0	16.7	33.3	75.0	100.0	
			S2KAN	1.0000	5.46×10^{-3} (Y)	6	42.5	40.0	80.0	100.0	60.0	100.0	100.0	
		LM	KAN	1.0000	4.36×10^{-4} (Y)	9	135.0	0.0	0.0	12.5	35.4	76.4	88.2	
			S2KAN	1.0000	4.18×10^{-3} (Y)	9	65.9	38.1	100.0	100.0	57.1	100.0	100.0	
F2	$x^4 + x^3 + x^2 + x$ $x \in [-1, 1]$	S	KAN	1.0000	1.25×10^{-4} (Y)	1	15.0	0.0	100.0	100.0	0.0	100.0	100.0	
			S2KAN	1.0000	8.25×10^{-4} (Y)	1	19.2	0.0	100.0	100.0	0.0	100.0	100.0	
		L	KAN	1.0000	4.60×10^{-4} (Y)	6	90.0	0.0	16.7	50.0	41.7	91.7	100.0	
			S2KAN	1.0000	6.26×10^{-3} (Y)	6	41.4	60.0	100.0	100.0	100.0	100.0	100.0	
		LM	KAN	1.0000	4.50×10^{-4} (Y)	9	135.0	0.0	5.6	33.3	35.4	75.0	94.4	
			S2KAN	1.0000	5.25×10^{-3} (Y)	9	46.6	63.3	100.0	100.0	83.3	100.0	100.0	
F3	$x^5 + x^4 + x^3 + x^2 + x$ $x \in [-1, 1]$	S	KAN	1.0000	2.43×10^{-4} (Y)	1	15.0	0.0	0.0	100.0	0.0	100.0	100.0	
			S2KAN	1.0000	4.94×10^{-4} (Y)	1	20.1	0.0	0.0	100.0	0.0	100.0	100.0	
		L	KAN	1.0000	7.47×10^{-4} (Y)	6	90.0	0.0	0.0	25.0	33.3	75.0	100.0	
			S2KAN	1.0000	6.16×10^{-3} (Y)	6	55.9	40.0	80.0	80.0	60.0	100.0	100.0	
		LM	KAN	1.0000	6.24×10^{-4} (Y)	9	135.0	0.0	5.6	12.5	23.6	64.6	88.9	
			S2KAN	1.0000	4.97×10^{-3} (Y)	9	66.6	35.4	92.9	100.0	50.0	100.0	100.0	
F4	$x^6 + x^5 + x^4 + x^3 + x^2 + x$ $x \in [-1, 1]$	S	KAN	1.0000	5.61×10^{-4} (Y)	1	15.0	0.0	0.0	100.0	0.0	100.0	100.0	
			S2KAN	1.0000	7.55×10^{-4} (Y)	1	19.5	0.0	0.0	100.0	0.0	100.0	100.0	
		L	KAN	1.0000	7.91×10^{-4} (Y)	6	90.0	0.0	0.0	26.7	20.0	63.3	91.7	
			S2KAN	1.0000	7.81×10^{-3} (Y)	6	50.5	50.0	80.0	91.7	77.5	100.0	100.0	
		LM	KAN	1.0000	5.89×10^{-4} (Y)	9	135.0	0.0	12.5	25.0	37.5	64.6	87.5	
			S2KAN	1.0000	5.54×10^{-3} (Y)	9	62.8	28.6	100.0	100.0	57.1	100.0	100.0	
F5	$\sin(x^2) \cos(x) - 1$ $x \in [-1, 1]$	S	KAN	1.0000	5.11×10^{-5} (Y)	1	15.0	0.0	100.0	100.0	100.0	100.0	100.0	
			S2KAN	1.0000	9.01×10^{-4} (Y)	1	19.2	0.0	100.0	100.0	100.0	100.0	100.0	
		L	KAN	1.0000	1.77×10^{-4} (Y)	6	90.0	8.3	29.2	36.7	45.0	66.7	100.0	
			S2KAN	0.9517	3.52×10^{-2} (N)	6	58.2	75.0	100.0	100.0	100.0	100.0	100.0	
		LM	KAN	1.0000	1.26×10^{-4} (Y)	9	135.0	5.6	12.5	25.0	29.2	64.6	88.2	
			S2KAN	0.8041	7.34×10^{-2} (N)	8.5	73.2	70.8	100.0	100.0	91.7	100.0	100.0	
F6	$\sin(x) + \sin(x + x^2)$ $x \in [-1, 1]$	S	KAN	1.0000	8.28×10^{-5} (Y)	1	15.0	0.0	100.0	100.0	100.0	100.0	100.0	
			S2KAN	1.0000	9.55×10^{-4} (Y)	1	23.6	0.0	100.0	100.0	100.0	100.0	100.0	
		L	KAN	1.0000	2.85×10^{-4} (Y)	6	90.0	0.0	18.3	45.0	66.7	100.0	100.0	
			S2KAN	1.0000	2.95×10^{-3} (Y)	6	40.9	80.0	100.0	100.0	100.0	100.0	100.0	
		LM	KAN	1.0000	3.59×10^{-4} (Y)	9	135.0	0.0	11.8	33.3	37.5	77.8	94.4	
			S2KAN	1.0000	6.08×10^{-3} (Y)	9	62.7	71.4	100.0	100.0	85.7	100.0	100.0	
F7	$\log(x + 1) + \log(x^2 + 1)$ $x \in [0, 2]$	S	KAN	1.0000	1.03×10^{-5} (Y)	1	15.0	100.0	100.0	100.0	100.0	100.0	100.0	
			S2KAN	1.0000	2.30×10^{-3} (Y)	1	20.7	100.0	100.0	100.0	100.0	100.0	100.0	
		L	KAN	1.0000	3.07×10^{-4} (Y)	6	90.0	0.0	8.3	41.7	50.0	83.3	100.0	
			S2KAN	0.9994	1.96×10^{-2} (N)	6	56.5	55.0	100.0	100.0	80.0	100.0	100.0	
		LM	KAN	1.0000	3.60×10^{-4} (Y)	9	135.0	0.0	5.6	25.0	37.5	81.9	100.0	
			S2KAN	0.9963	5.02×10^{-2} (N)	9	72.7	58.6	100.0	100.0	80.4	100.0	100.0	
F8	$\sqrt{x}$ $x \in [0, 4]$	S	KAN	1.0000	1.83×10^{-3} (Y)	1	15.0	0.0	0.0	0.0	100.0	100.0	100.0	
			S2KAN	1.0000	2.05×10^{-3} (Y)	1	19.1	0.0	0.0	0.0	100.0	100.0	100.0	
		L	KAN	1.0000	3.51×10^{-4} (Y)	6	90.0	0.0	16.7	33.3	50.0	83.3	100.0	
			S2KAN	0.9991	1.41×10^{-2} (N)	6	51.8	41.7	77.5	100.0	63.3	100.0	100.0	
		LM	KAN	1.0000	1.90×10^{-4} (Y)	9	135.0	0.0	5.6	19.6	35.4	76.4	88.2	
			S2KAN	0.9993	1.25×10^{-2} (N)	9	61.5	34.3	80.0	100.0	66.7	100.0	100.0	
F9	$\sin(x) + \sin(y^2)$ $x, y \in [-1, 1]$	S	KAN	1.0000	6.75×10^{-5} (Y)	2	30.0	50.0	100.0	100.0	100.0	100.0	100.0	
			S2KAN	1.0000	1.03×10^{-3} (Y)	2	34.4	50.0	100.0	100.0	100.0	100.0	100.0	
		L	KAN	1.0000	2.13×10^{-4} (Y)	9	135.0	7.1	34.3	58.6	67.0	93.8	100.0	
			S2KAN	1.0000	3.73×10^{-3} (Y)	9	41.9	93.8	100.0	100.0	100.0	100.0	100.0	
		LM	KAN	1.0000	3.41×10^{-4} (Y)	14	210.0	0.0	8.7	45.8	53.8	96.2	100.0	
			S2KAN	1.0000	3.08×10^{-3} (Y)	13.5	50.2	80.0	100.0	100.0	89.4	100.0	100.0	
F10	$2 \sin(x) \cos(y)$ $x, y \in [-\pi, \pi]$	S	KAN	-0.0372	1.01×10^0 (N)	2	30.0	0.0	0.0	0.0	0.0	0.0	0.0	
			S2KAN ^a	-0.0006	9.90×10^{-1} (N)	0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	
		L	KAN	1.0000	5.51×10^{-3} (Y)	9	135.0	0.0	43.7	77.8	44.4	94.4	100.0	
			S2KAN	1.0000	3.73×10^{-3} (Y)	9	22.0	86.6	100.0	100.0	94.4	100.0	100.0	
		LM	KAN	1.0000	3.31×10^{-3} (Y)	14	210.0	0.0	14.3	50.0	37.1	78.6	96.4	
			S2KAN	1.0000	1.85×10^{-3} (Y)	10	27.2	85.7	100.0	100.0	100.0	100.0	100.0	
Median F1–F10		S	KAN	1.0000	1.04×10^{-4}	1	15.0	0.0	100.0	100.0	50.0	100.0	100.0	
			S2KAN	1.0000	9.28×10^{-4}	1	19.8	0.0	100.0	100.0	100.0	100.0	100.0	
		L	KAN	1.0000	4.05×10^{-4}	6	90.0	0.0	16.7	39.2	44.7	83.3	100.0	
			S2KAN	1.0000	6.21×10^{-3}	6	46.5	57.5	100.0	100.0	87.2	100.0	100.0	
		LM	KAN	1.0000	3.98×10^{-4}	9	135.0	0.0	7.1	25.0	36.3	76.4	91.7	
			S2KAN	1.0000	5.40×10^{-3}	9	62.8	61.0	100.0	100.0	81.8	100.0	100.0	

^a All ten fits have no functionally active edges; SF is 100% by convention. This does not imply an accurate model.

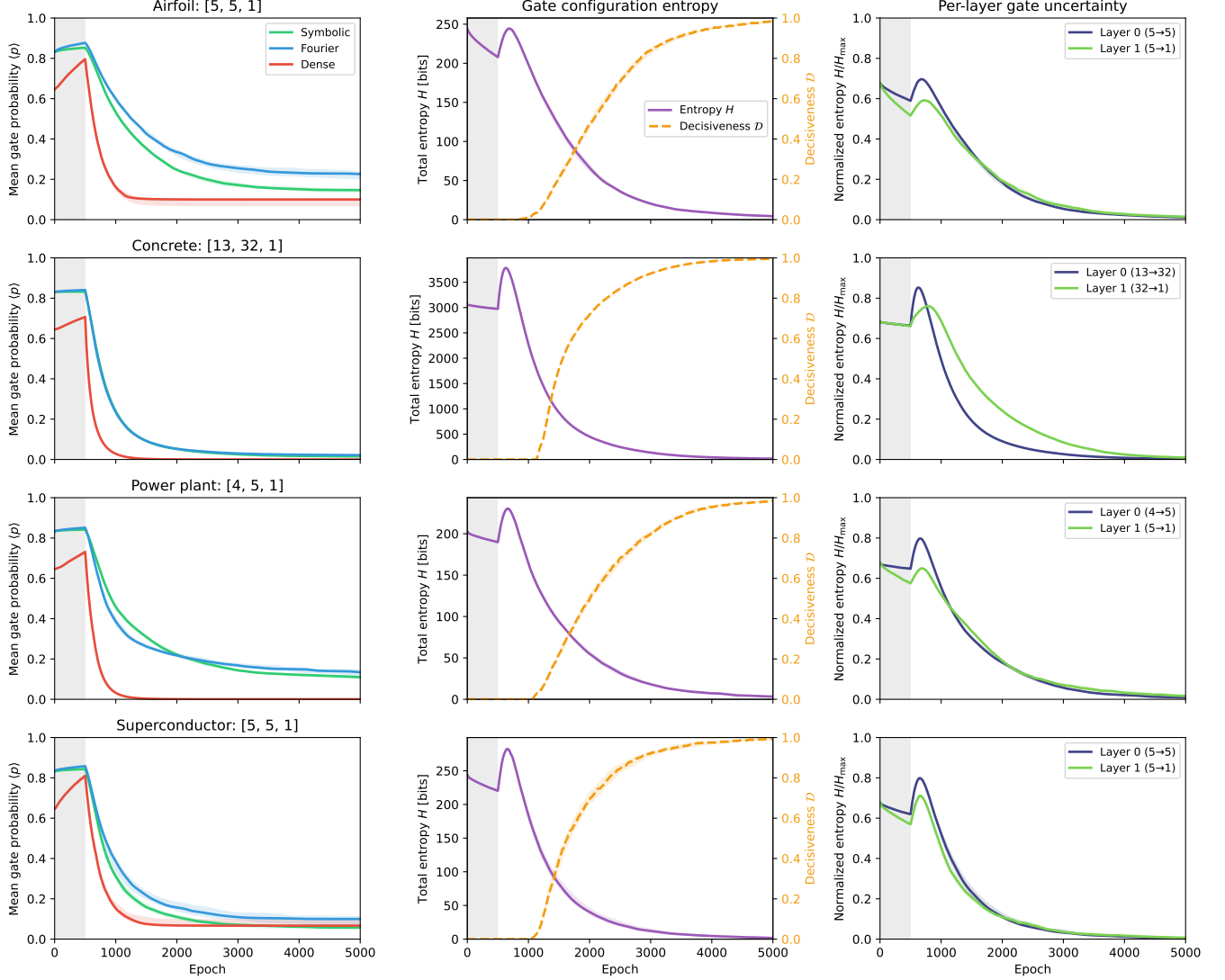


Figure 3: Learning dynamics for S2KAN on four real-world datasets, with $\beta = 1$ after 500 warmup epochs. Rows show Airfoil, Concrete, Power plant, and Superconductor. Columns show mean gate-opening probabilities by basis type; total gate entropy and decisiveness; and normalized per-layer entropy $H/H_{\max}$. Dense denotes the spline branch, including its SiLU residual. Curves show medians over ten data splits with three initialization seeds per split; bands show interquartile ranges. Gray shading marks warmup ($\beta = 0$). Full 5,000-epoch trajectories are shown.

6 Learning dynamics

To understand how S2KAN discovers sparse representations during training, we performed an experiment to track gate statistics throughout optimization. For each gate with learnable parameter α_i , p_i is the probability that the gate is nonzero (Eq. (11)). Using these, we compute the total binary entropy across all gates in the network:

$$H = - \sum_m [p_m \log_2 p_m + (1 - p_m) \log_2 (1 - p_m)], \quad (20)$$

which measures the total uncertainty in the gate configuration. When all gates are fully decided ($p_m \approx 0$ or $p_m \approx 1$), entropy is near zero; when gates are maximally uncertain ($p_m \approx 0.5$), entropy is maximized. We also report *decisiveness*, the fraction of gates with $p_m < 0.01$ or $p_m > 0.99$, indicating convergence to discrete selection.

Figure 3 shows these statistics for the four real-world datasets using the architectures and dictionaries described in Methods (App. A).

During warmup (when $\beta = 0$), mean dense-gate probabilities increase across all four datasets, while symbolic and Fourier gate probabilities remain high. After regularization begins ($\beta = 1$), gate entropy briefly increases before declining as gates become increasingly decisive. Dense-gate probabilities decrease more rapidly than those of the symbolic and Fourier terms, approaching zero for Concrete and Power Plant while remaining nonzero for Airfoil and Superconductor. The per-layer trajectories also show differences in timing: for Concrete, normalized entropy falls more rapidly in the input layer than in the output layer, whereas the two layers follow more similar trajectories on the other datasets. Together, these observations show a transition from broadly open gates toward sparse, increasingly discrete representations, with the retained activation families and timing of gate decisions varying across datasets.

Model size can also decrease without an explicit sparsity penalty: in the two dynamical systems trained with $\beta = 0$ throughout, we observe a decrease in expected effective parameter count during training. All ten fits per system finish below their first-epoch counts, and the median final count is below baseline KAN in both systems (App. E.4). These observations are specific to the tested training configurations; their mechanism and dependence on optimization settings remain unresolved.

To understand how sparsity distributes across network depth, we examined the expected fraction of active gates per layer, ρ_ℓ , for the multi-layer architectures in our experiments (Fig. 4):

$$\rho_\ell := \frac{1}{n_\ell n_{\ell+1} M} \sum_{ijm} p_{\ell ijm}, \quad (21)$$

where $M = |\mathcal{S}| + |\mathcal{F}| + |\mathcal{R}|$ is the number of dictionary terms per activation, n_ℓ and $n_{\ell+1}$ are the input and output widths of layer ℓ , and $p_{\ell ijm}$ is the probability that the gate is nonzero (Eq. (11)). Figure 4 compares the four-layer Ikeda, three-layer ecosystem, and two-layer real-world networks. Ikeda retains a smaller fraction of gates in its interior layers than at its boundaries; the ecosystem instead retains the smallest fraction in its output layer, while the real-world networks retain a larger fraction in the output layer. The resulting sparsity profiles appear architecture- and dataset-dependent, rather than a universal interior-layer pattern.

7 Discussion

Across real-world prediction tasks, dynamical-system forecasting, and synthetic symbolic benchmarks, S2KAN generally achieves higher symbolic fidelity and often far smaller models, while maintaining competitive predictive accuracy on real-world data and low prediction errors on most symbolic benchmark configurations. Differentiable gating and MDL-inspired regularization jointly select symbolic and dense activation components, allowing compact symbolic representations while preserving the option to retain dense terms. We now discuss limitations and directions for future work.

Like any symbolic regression method, S2KAN requires a set of candidate functions: PySR (Cranmer, 2023), SINDy (Brunton et al., 2016), and AI Feynman (Udrescu & Tegmark, 2020) likewise require the user to specify operators or candidate terms. In a KAN, the activation dictionary plays the role of the operator set. The Kolmogorov–Arnold representation shows that addition suffices with sufficiently expressive univariate functions and a suitable architecture; multiplication nodes can provide more compact representations (Liu, Tegmark, et al., 2025). When domain knowledge is available, it can guide this choice—physical systems, for instance, suggest trigonometric functions, rational functions, or exponentials depending on the phenomena involved; otherwise a generic dictionary can be used, as in our Nguyen experiments. Unlike approaches restricted to the supplied symbolic dictionary, S2KAN retains dense representations as a fallback for whatever the dictionary cannot express. These dense terms may or may not recover the missing symbolic structure: in the sinc ablation (App. G.4), a spline learned the omitted $1/x$ almost exactly in three of five seeds, whereas the splines of a baseline KAN never did. Whether they do can be checked by fitting the learned dense activations with candidate primitives outside the training dictionary, as we did for $1/x$; like symbolic fidelity generally, such a check is relative to the primitives tested. The fallback typically keeps predictive accuracy high even with a suboptimal library, though not always at baseline level, and strong regularization can suppress it (App. G.4).

The symbolic library presents its own practical challenges. Terms such as $1/x$, $\log(x)$, or $\exp(x)$ can diverge for certain inputs, destabilizing training. Likewise, $\log(x)$ and $\sqrt{x}$ have restricted domains that may be violated during training. In this light, the standard KAN’s reliance on dense representations is a practical strength: its spline and SiLU components avoid the restricted domains and singularities of primitives such as logarithms and reciprocals. Yet these challenges are not unique to S2KAN: the problem is endemic to symbolic regression methods including neural network approaches like equation learner (EQL) networks (Martius & Lampert, 2016; Sahoo et al., 2018; Werner et al., 2021). Practical mitigations include input normalization, domain restriction, careful initialization, and replacing problematic terms with protected variants such as $\sqrt{|x|}$ or $\log(|x| + \epsilon)$. The equation learner with division (EQL-div) (Sahoo et al., 2018) and informed equation learning (iEQL) (Werner et al., 2021)

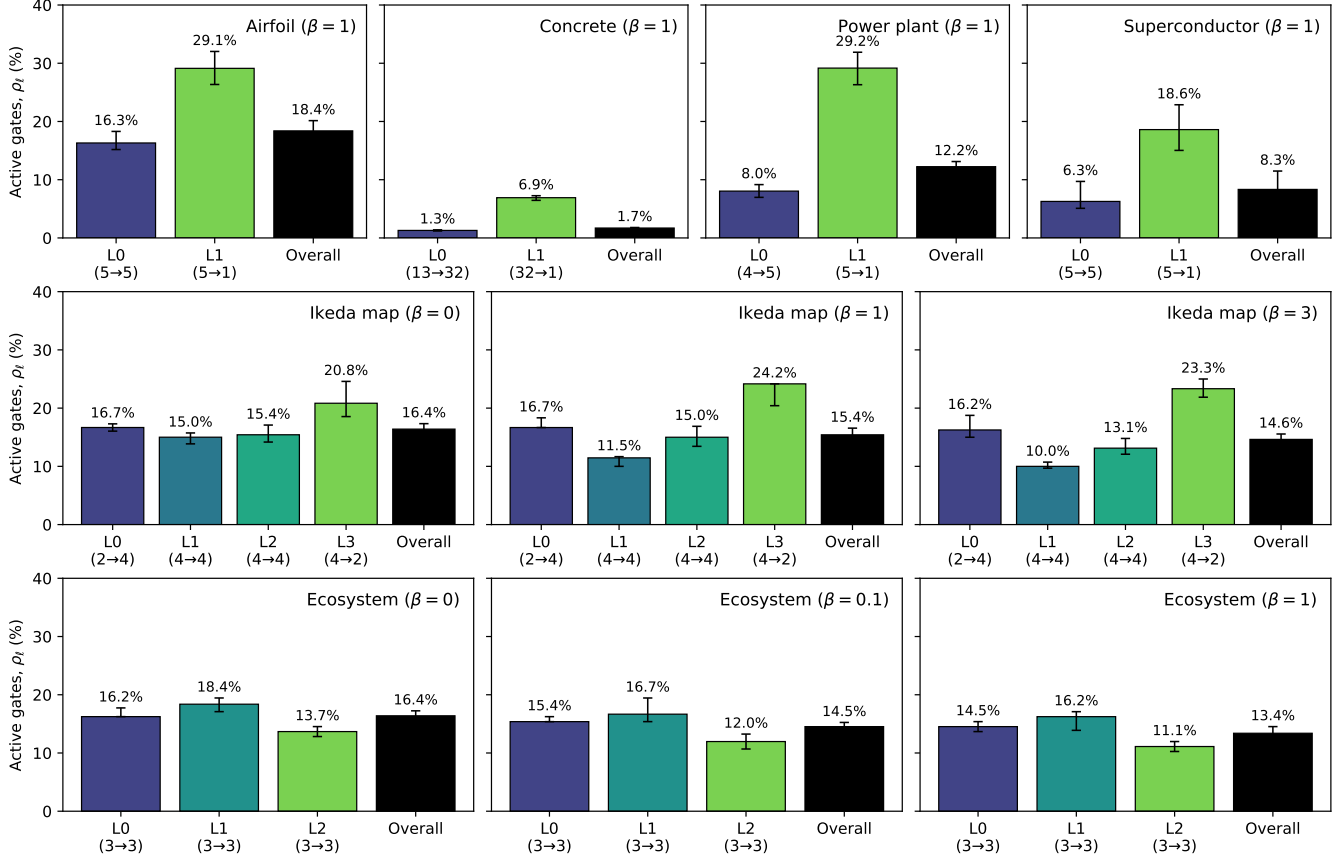


Figure 4: Expected fraction of active gates per layer, ρ_ℓ (Eq. (21)), for the real-world and dynamical-systems S2KAN models. Bars show medians; whiskers indicate IQRs across ten initialization seeds per dynamical-system configuration and 30 fits per real-world dataset (ten data splits and three initialization seeds per split). Overall fractions are computed within each model before aggregation.

variants offer a more principled solution, introducing symbolic primitives with learnable cutoffs that prevent divergences and provide smooth out-of-domain behavior; similar modifications could be incorporated into S2KAN. Beyond content, dictionary size also matters: larger dictionaries increase training overhead. When properly implemented to omit inactive terms, inference evaluates only the retained representation; replacing spline expansions with simple symbolic primitives can therefore reduce evaluation cost. Appendix C reports training and inference times for KAN and S2KAN across the four real-world datasets, with inactive terms omitted during inference.

S2KAN takes an explicit approach to symbolic fidelity, incorporating symbolic primitives directly into the activation function dictionary. However, the training instabilities discussed above—divergences, domain violations, the need for protected operators—motivate an alternative: biasing dense representations implicitly toward forms with high symbolic fidelity. Rather than embedding symbolic functions explicitly, implicit approaches modify the training process or architecture to encourage learned activations that, while remaining dense, are more amenable to post-hoc symbolic fitting, effectively a symbolic inductive bias. Projective KAN (Poole et al., 2026) exemplifies this direction, using projection-based regularization to bias activations toward higher symbolic fidelity. Confining ill-behaved terms to the regularization rather than the main computational graph generally makes them easier to isolate and control. Combining explicit symbolic terms with implicit biases toward symbolic fidelity is a promising direction: such hybrid approaches could encourage symbolic representations while improving stability when explicit symbolic terms are ill-suited.

How does S2KAN fit into the broader context of symbolic regression and related methods? Table 4 compares S2KAN qualitatively with related methods. Since our contribution focuses on symbolifying KANs, our empirical evaluation centers on KAN versus S2KAN, although one of our ablations (App. G.2) yields a model resembling EQL. To our knowledge, S2KAN is the first method to combine gradient-based optimization, compositionality, symbolic learning, and graceful degradation through a trainable spline fallback. The table also highlights that compositionality is shared by nearly all of these methods, a feature that interacts with depth and readability in ways worth examining.

Table 4: Qualitative comparison of S2KAN with selected symbolic regression and KAN methods.

Method	Gradient-based	Compositional	Symbolic	Graceful degradation
SINDy (Brunton et al., 2016)	✗	✗	✓	✗
PySR (Cranmer, 2023)	✗	✓	✓	✗
AI Feynman (Udrescu et al., 2020; Udrescu & Tegmark, 2020)	✗ ^a	✓	✓	✗
EQL / iEQL (Sahoo et al., 2018; Werner et al., 2021)	✓	✓	✓	✗
KAN (Liu, Wang, et al., 2025)	✓	✓	✗ ^b	n/a
Projective KAN (Poole et al., 2026)	✓	✓	✗ ^b	n/a
S2KAN (ours), $\mathcal{S}$ or $\mathcal{S}+\mathcal{F}$	✓	✓	✓	✗
S2KAN (ours), $\mathcal{S}+\mathcal{F}+\mathcal{R}$	✓	✓	✓	✓

^a Brute-force search over expressions; uses neural network surrogates internally for decomposition.

^b Post-hoc symbolification available but not part of the training objective.

Compositionality is essential for discovering complex expressions, but the number of active paths can grow combinatorially as depth and breadth increase, producing large expressions that may hinder readability. Sparsity mitigates this to an extent by reducing the number of active paths, but depth (more function compositions) and breadth (more terms per function) interact with sparsity and must be considered together. S2KAN promotes sparsity, while its distribution across layers appears to depend on both the dataset and architecture (Fig. 4). The observed profiles describe how retained capacity is distributed through each model, but do not by themselves establish whether its depth or breadth is appropriate for the task. These observations raise a broader question for future work: when do interpretable components produce an interpretable model? Understanding how depth, breadth, and sparsity affect whole-model readability and predictive accuracy could help guide architecture selection.

We have presented S2KAN, a method that integrates symbolic primitives directly into Kolmogorov–Arnold networks through differentiable gating and MDL-inspired regularization. By placing symbolic and dense representations on equal footing, S2KAN can learn interpretable activation functions while retaining dense components for approximation. Exploring whether this differentiable sparse selection mechanism can support learned symbolic structure in other architectures remains a direction for future work. These results show how incorporating symbolic structure during training can produce compact KAN models whose learned responses can be examined through the contributions of their symbolic and dense components.

A Methods

S2KAN was implemented in PyTorch v2.8.0 (Paszke et al., 2019). Optimization was performed using Adam (Kingma & Ba, 2014) with default parameters and a constant learning rate of 10^{-3} for both coefficients and gates. Predictive-benchmark R^2 and RMSE values are computed on held-out test sets; the response diagnostics use the evaluation subsets specified in App. D. Non-spline coefficients were initialized from $U(-0.05, 0.05)$; spline coefficients were initialized as in Liu, Wang, et al. (2025). For ϕ_{spline} we used SiLU residuals and B-splines with $G = 10$ grid intervals and order $K = 3$ for all experiments.

We apply gates to individual activation terms before summing edge outputs into subnodes. Multiplication nodes multiply their two designated subnode sums: a closed term contributes zero, and a zero subnode makes the product zero. No factor is dropped from the product. A single dense gate jointly controls the spline and SiLU contributions.

Unless otherwise noted, spline grids were adapted using training activations before optimization and at the start of epochs 0, 5, ..., 45, then held fixed; grid refinement was not used. Chebyshev polynomials are defined on $[-1, 1]$, so activation function domains are tracked per activation and updated alongside spline grid updates; inputs are rescaled to this domain before computing the Chebyshev basis. Fourier terms use natural frequencies $\sin(qx)$, $\cos(qx)$ without domain rescaling. We use the following implementations for selected symbolic terms: $\sqrt{x} \rightarrow \sqrt{\max(x, \epsilon)}$, $1/x \rightarrow 1/(x + \epsilon)$, and $\log(x) \rightarrow \log(\max(x, \epsilon))$, with $\epsilon = 10^{-8}$; $\exp(x)$ is clamped to $\exp(\text{clamp}(x, -10, 10))$ to prevent overflow.

For regularization term $k = \sum_{\ell ij} \sum_m w_m p_{\ell ijm}$, we used $w_m = 1$ for all non-dense terms, and $w_m = K + G + 2$ for the dense term (the +2 accounts for the learnable scale parameters; Eq. (3)).

For the Hard Concrete distribution, we used temperature $\tau = 2/3$ and stretch parameters $\gamma = -0.1$, $\zeta = 1.1$ throughout. Unless otherwise noted, gates were initialized with $\alpha_i \sim \mathcal{N}(0, 0.1)$ except for spline gates which used $\alpha_i = -1$, providing a slight bias toward symbolic representations at initialization. No temperature annealing was used.

Unless otherwise noted, models were trained with a batch size of 128. For S2KAN, we set $\beta = 0$ for the first 200 epochs, then switch to the stated value for the remainder of training; experiment-specific warmup durations are given below. Unless otherwise noted, baseline KAN used ϕ_{spline} only, with gates fixed open, and minimized raw MSE without regularization. S2KAN used the log-MSE objective in Eq. (16). Appendix G.1 compares raw and log MSE for baseline KAN under otherwise identical settings.

The real-world datasets contain measurement noise, and we observed overfitting in baseline KANs. To limit overfitting, we split each training sample 80/20 into fitting and validation subsets for both KAN and S2KAN and selected the checkpoint with the lowest validation MSE, evaluated every ten epochs. These runs completed the full training budget; validation selected the reported checkpoint without stopping training. The Nguyen and dynamical-system datasets were generated without added observation noise; for these experiments, we report final training checkpoints rather than selecting checkpoints by validation error.

Real-world data (Table 1, Fig. 3)— For concrete compressive strength, we used UCI dataset 165 (Yeh, 1998, 2006) with 8 raw features augmented by 5 derived features (water-cement ratio, water-binder ratio, total binder, total aggregate, $\log(\text{age} + 1)$), for 13 total. We used an 80/20 train-test split and architecture [13, 32, 1]. For superconductor critical temperature prediction, we used 5 features from UCI dataset 464 (Center for Basic Research on Materials, 2024; Hamidieh, 2018): number of elements, weighted mean valence, weighted mean first ionization energy, mean electron affinity, and valence entropy. We sampled 1000 observations for fitting and validation and 1000 separate test observations, and used architecture [5, 5, 1] (previously used in Bagrow and Bongard (2025)). For Airfoil (UCI dataset 291) and Power Plant (UCI dataset 294), we used the five and four supplied inputs, respectively, with architectures [5, 5, 1] and [4, 5, 1] for both methods. Airfoil used the remaining 503 observations for testing; Power Plant used 1000 separate test observations. For all four datasets and both methods, each input feature and the target were standardized as $(v - \mu_{\text{fit}})/(\sigma_{\text{fit}} + 10^{-8})$, using means and population standard deviations computed only on the fitting subset. Concrete used 659 fitting and 165 validation observations; Airfoil, Power Plant, and Superconductor each used 800 fitting and 200 validation observations. The same transformation was applied to the fitting, validation, and test sets. Predictions were transformed back to the dataset’s target units before reporting RMSE: dB for Airfoil’s scaled sound pressure level, MPa for Concrete, MW for Power Plant, and K for Superconductor. For S2KAN, Concrete used symbolic library $\mathcal{S} = \{1, x, x^2, \sqrt{x}, \log(x + 1)\}$; Airfoil, Power Plant, and Superconductor additionally included x^3 and $\exp(x)$. S2KAN used Fourier modes $Q = 2$ on all four datasets. Baseline KAN used dense spline and SiLU activations with gates fixed open and no symbolic or Fourier terms; neither method used a Chebyshev basis. All models were trained for 5000 epochs with batch size 64; S2KAN used 500 warmup epochs before sparsity regularization was activated. Early stopping was not used. Appendix B provides a full per-layer breakdown of gate counts by term type.

Scientific consistency checks (Sec. 5.2)— We evaluate the same validation-selected real-world checkpoints deterministically, using fitting-set preprocessing and reporting responses in physical units. Native activation decompositions retain all active terms, protected operations, and fixed-input contributions; no further coefficient fitting is performed. The response protocols, input-support restrictions, illustrative-example selection, and secondary results are given together in App. D.

Dynamical systems (Table 2)— Training/test/unseen partitions contain 6,400/1,600/2,000 observations for Ikeda and 3,000/750/1,250 for the ecosystem; the unseen block is reserved for rollout evaluation. The partitions cover consecutive time periods at the end of each simulated trajectory, in the order listed, and are shared across all configurations and initialization seeds. Both systems were fitted in their original state coordinates without standardization. Ecosystem observations are separated by $\Delta t = 0.5$ time units, as in Panahi et al. (2025). For the Ikeda map, we used architecture [2, 4, 4, 4, 2] with symbolic library $\mathcal{S} = \{\sqrt{x}, 1/(1 + x^2)\}$, $P = 4$ Chebyshev terms, and Fourier modes $Q = 4$. For the ecosystem, we used architecture [3, 3, 3, 3] with symbolic library $\mathcal{S} = \{1, x, x^2, 1/(1 + \max(x, 0))\}$, no Chebyshev basis ($P = 0$), and Fourier modes $Q = 4$. Both methods used the same architectures, following Bagrow and Bongard (2025). In the ecosystem’s residual formulation, the fixed identity connection is excluded from parameter counts and symbolic fidelity. Early stopping was not used. Training used batch size 128 for 20,000 epochs (Ikeda) or 45,000 epochs (ecosystem), with ten initialization seeds per configuration. We evaluated $\beta \in \{0, 1, 3\}$ for Ikeda and $\beta \in \{0, 0.1, 1\}$ for the ecosystem, including $\beta = 0.1$ to examine weaker ecosystem regularization. One-step RMSE pools errors over observations and output coordinates before taking the square root.

To assess dynamical behavior beyond one-step prediction accuracy, we iterate each trained model in original state coordinates, starting from the first unseen state. For long-orbit diagnostics, we discard 200 iterates and retain 20,000 iterates for Ikeda or 40,000 for the ecosystem, analyzing the second half. Lyapunov spectra are estimated by the standard QR method (Skokos, 2010), using automatic-differentiation Jacobians along model trajectories; ecosystem rates are divided by Δt . Reference spectra use generating maps and the final 100,000 points of 200,000-point trajectories following a 1,000-step transient. Additional dynamical diagnostic procedures and results are described in App. E.

Nguyen benchmark (Table 3)— We evaluated the first 10 Nguyen problems (Nguyen et al., 2011), generating 1024 training and 256 test points per problem. Training and test inputs were sampled independently and uniformly over the tabulated domains, with independent coordinates for the bivariate problems. Inputs and targets were used without standardization. Across all Nguyen β values, S2KAN training stopped when gate decisiveness remained ≥ 0.99 for 500 consecutive epochs, or at 10,000 epochs; baseline KAN always completed 10,000 epochs. This stopping rule uses only gate probabilities, with no validation or test loss involved (see also App. G.3.2). We report, for both baseline and S2KAN, medians over ten initialization seeds. In Table 3, this stopping criterion was reached only by the ten small-architecture S2KAN fits for Nguyen-10, which closed all gates

and stopped after 3,337–4,059 epochs; all other fits completed 10,000 epochs. Three architectures were tested: small (no hidden layer), large (one hidden layer with 3 summation units), and large-mult (3 summation plus 1 multiplication unit). S2KAN used symbolic library $\mathcal{S} = \{1, x, x^2, \sin(x), \cos(x)\}$, $P = 4$, and $Q = 4$.

Sinc function (Fig. 1)— We generated 1024 training and 256 test points on uniform grids over $x \in [1, 15]$, with target $y = \sin(x)/x$ and no standardization. Both models used a single multiplication unit with no hidden layer, shape $[1, (0, 1)]$, trained for 8,000 epochs with batch size 32 and constant learning rate 10^{-3} for coefficients, scale parameters, and gates. S2KAN used symbolic library $\mathcal{S} = \{x, 1/(x + 10^{-8})\}$, no Chebyshev terms ($P = 0$), Fourier modes $Q = 2$, regularization $\beta = 1$, and 100 warmup epochs. KAN used raw MSE, while S2KAN used the log-MSE objective. In this experiment, baseline gates were initialized near-open and remained trainable. Figure 1 shows the final checkpoints for seed 636139, the first seed in the experiment protocol; no coefficient refitting or term removal was applied. Early stopping was not used. Appendix G.4 examines the effect of removing the reciprocal term from this dictionary.

A.1 Measuring symbolic fidelity

We evaluate frozen models deterministically on all fitting inputs, excluding validation and test inputs, and apply the functional-activity criteria of Sec. 3. All functionally active edges count equally across layers; the ecosystem’s fixed identity additions are excluded. For vector outputs, the numerical functional-activity floor uses the variance of all predicted values pooled across observations and output coordinates, about a single common mean. For each functionally active edge, we fit

$$s(u) = d + \sum_{m=1}^j a_m p_m(b_m u + c_m), \quad j \leq J, \quad (22)$$

with a free intercept and repeated primitives allowed. We use $P = \{x, x^2, \sin\}$ for Nguyen and real-world data, adding $1/x$ for sinc, $\sqrt{\max(x, 10^{-8})}$ and $1/(1 + x^2)$ for Ikeda, or $1/(1 + \max(x, 0))$ for the ecosystem; both methods use the same evaluation dictionary.

The initialization grid uses mean-centered edge inputs and their 99th–1st percentile range r (set to one if zero). Sines use 40 geometrically spaced frequencies from 0.25 to 64 cycles per r and eight equally spaced phases in $[0, \pi)$. Polynomial primitives use unit input scale and nine equally spaced shifts over $[-r, r]$; other primitives use scales $\{\pm \frac{1}{2}, \pm 1, \pm 2\}$ with the same shifts. These values initialize the search rather than constrain subsequent fitting.

After standardizing the edge response, greedy forward selection considers the three grid-initialized atoms with largest absolute normalized residual correlations and the best atom of each primitive. For each candidate addition, SciPy’s nonlinear least-squares solver jointly refits all coefficients, scales, shifts, and the intercept, with at most 2,000 function evaluations. Parameter step scales adapt to the column norms of the residual Jacobian. We retain the best fit only if it improves the residual sum of squares. Fitting and scoring use the same input sample. We report the fraction of functionally active edges with approximation $R^2 \geq 1 - \epsilon^2$ for $J = 1, 2, 3$ and $\epsilon = 0.01, 0.1$. This searches additive expressions only; products and nested compositions are not searched. For Fig. 1, $J = 1$ and $\epsilon = 0.01$ give 0/2 qualifying edges for KAN and 2/2 for S2KAN.

Large, mutually cancelling terms could produce accurate symbolic approximations with unstable coefficients. As a sensitivity check (App. G.5), we limit each fitted term’s standard deviation to three times that of the edge response; 94.7% of 6,132 SF scores remain unchanged.

AI assistance. The authors used OpenAI Codex and Anthropic Claude to assist with code development, data analysis, and manuscript preparation. The authors reviewed and verified the outputs used in this work and take responsibility for the final content.

B Sparsity breakdown by term type

Tables 5 and 6 report per-layer counts of active activation functions and open gates for the dynamical-systems and real-world S2KAN models, respectively, including all four real-world datasets. We summarize all ten initialization seeds for each dynamical-system configuration and all thirty fits (ten data splits, three initialization seeds per split) for each real-world dataset. Entries give medians and first and third quartiles. Active counts are defined by gate state; symbolic fidelity uses separate functional-activity and approximation criteria.

Table 5: Sparsity breakdown for the dynamical-systems S2KAN models. Entries are medians over ten initialization seeds, with first and third quartiles below (omitted for constant counts). Ikeda uses the direct map at 20,000 epochs with $P = Q = 4$; Ecosystem uses the residual map at 45,000 epochs with $P = 0, Q = 4$. Spline grids are updated early in training. Active means at least one open gate; $\mathcal{S}$ -only excludes both $\mathcal{F}$ and $\mathcal{R}$, while $\mathcal{S}+\mathcal{F}$ excludes $\mathcal{R}$. Gates are counted at $p_{\text{open}} > 0.5$: $\mathcal{S}$ is the symbolic library, $\mathcal{F}$ is Chebyshev/Fourier, and $\mathcal{R}$ is the spline branch (one gate per edge, not one per spline coefficient). Ecosystem’s three fixed identity additions are excluded. Totals are computed within each model before aggregation, so their medians need not equal sums of layer medians.

Layer	Edges				Open gates		
	Edges	Active	$\mathcal{S}$ -only	$\mathcal{S}+\mathcal{F}$	$\mathcal{S}$	$\mathcal{F}$	$\mathcal{R}$
Ikeda, [2, 4, 4, 4, 2], $\beta = 0$							
L0 (2 $\rightarrow$ 4)	8	8	0	0	0	12.5	8
				[0, 0]		[11.25, 13]	[8, 8]
L1 (4 $\rightarrow$ 4)	16	15.5	0	0	0.5	20.5	15
		[15, 16]		[0, 0]	[0, 1]	[17.25, 21.75]	[14.25, 16]
L2 (4 $\rightarrow$ 4)	16	14	0	3	2.5	24	10.5
		[13.25, 14.75]	[0, 0]	[2, 4.75]	[1, 3.75]	[22, 26.75]	[9, 12]
L3 (4 $\rightarrow$ 2)	8	8	0	2	4	15.5	6
		[8, 8]	[0, 0]	[2, 2]	[3.25, 4.75]	[13, 16.75]	[6, 6]
Total	48	45	0	5.5	7	72.5	38
		[43, 46]	[0, 0]	[4, 8.75]	[6, 8.75]	[67, 77.25]	[37, 41]
Ikeda, [2, 4, 4, 4, 2], $\beta = 1$							
L0 (2 $\rightarrow$ 4)	8	8	0	2	0	13.5	6
				[1, 2]	[0, 0]	[13, 16]	[6, 7]
L1 (4 $\rightarrow$ 4)	16	12	0	8.5	1	23	3
		[10, 13]	[0, 0]	[6.25, 10]	[0.25, 2]	[19.25, 25]	[2, 4]
L2 (4 $\rightarrow$ 4)	16	13.5	0	11.5	2	31	2
		[13, 15.75]		[10, 14]	[1, 3.75]	[28.25, 38.5]	[1, 2]
L3 (4 $\rightarrow$ 2)	8	8	0	5.5	4.5	21	2.5
		[8, 8]	[0, 0]	[4.25, 6]	[4, 5]	[18, 21.75]	[2, 3]
Total	48	41.5	0	27.5	8.5	90.5	13
		[39.5, 42]	[0, 1]	[25.5, 30]	[7, 10.5]	[83.5, 96]	[12.25, 14.5]
Ikeda, [2, 4, 4, 4, 2], $\beta = 3$							
L0 (2 $\rightarrow$ 4)	8	8	0	3	0	15	5
				[2.25, 3.75]	[0, 0.75]	[13, 16]	[4.25, 5.75]
L1 (4 $\rightarrow$ 4)	16	11.5	0	11.5	2	21	0.5
		[10.25, 12]	[0, 0]	[9.25, 12]	[0.25, 3]	[21, 23]	[0, 1]
L2 (4 $\rightarrow$ 4)	16	14	0	13.5	2.5	28	0
		[12.25, 14]	[0, 0]	[12.25, 14]	[1, 4.5]	[26.25, 32.5]	[0, 0]
L3 (4 $\rightarrow$ 2)	8	8	0	7	6.5	21.5	0.5
		[8, 8]	[0, 0]	[7, 8]	[5.25, 7]	[19, 24.75]	[0, 1]
Total	48	40	0	33	11	88	6
		[38.25, 40.75]	[0, 0.75]	[32.25, 36.25]	[10.25, 12.75]	[83.5, 93]	[5, 7.75]
Ecosystem, [3, 3, 3, 3], $\beta = 0$							
L0 (3 $\rightarrow$ 3)	9	9	0	1	2	10	8
		[8.25, 9]		[0.25, 1]	[2, 2]	[9.25, 10]	[7.25, 8.75]
L1 (3 $\rightarrow$ 3)	9	7	0	3	8.5	8.5	4.5
		[7, 8]	[0, 1]	[2.25, 3]	[7.25, 10]	[6.25, 9.75]	[4, 5]
L2 (3 $\rightarrow$ 3)	9	8.5	3	4.5	8	4	4
		[8, 9]	[2, 4]	[4, 5]	[7.25, 9]	[3, 4]	[3.25, 5]
Total	27	25	4	8	18.5	21.5	16.5
		[24, 25.75]	[2.25, 4]	[7.25, 8.75]	[16.25, 20.5]	[20, 23]	[15.25, 18]
Ecosystem, [3, 3, 3, 3], $\beta = 0.1$							
L0 (3 $\rightarrow$ 3)	9	8	0	1	2	10	6.5
		[8, 8.75]		[1, 2]	[2, 2]	[10, 10]	[6, 7.75]
L1 (3 $\rightarrow$ 3)	9	7	1	6	9.5	10	1
		[7, 8]	[0.25, 2]	[6, 7]	[8.25, 10]	[8.25, 11]	[1, 1]
L2 (3 $\rightarrow$ 3)	9	8	3	6	8	4	2
		[7, 8]	[3, 4]	[5.25, 6.75]	[7.25, 8]	[3, 4.75]	[2, 2]
Total	27	23	4	13	19.5	23	9
		[22, 24.75]	[4, 5.75]	[12.25, 13.75]	[17, 20]	[22.25, 24.75]	[9, 10]
Ecosystem, [3, 3, 3, 3], $\beta = 1$							
L0 (3 $\rightarrow$ 3)	9	7	0	4.5	2.5	12	3
		[7, 7.75]		[4, 5]	[2, 3]	[11, 12.75]	[2, 3]
L1 (3 $\rightarrow$ 3)	9	7	1	7	9	9.5	0
		[7, 8]	[1, 2]	[7, 8]	[8.25, 10]	[8.25, 10.75]	
L2 (3 $\rightarrow$ 3)	9	8	5	8	8.5	4.5	0
		[8, 8.75]	[3.25, 5]	[8, 8.75]	[8, 9.75]	[3, 5]	
Total	27	22.5	6	20	19.5	25.5	3
		[22, 23.75]	[5, 6.75]	[19, 21]	[19, 20.75]	[24.25, 27.5]	[2, 3]

Table 6: Sparsity breakdown for the real-world S2KAN models. All use $\beta = 1$, $P = 0$, $Q = 2$. Entries are medians over ten data splits with three initialization seeds per split (30 fits per dataset); brackets below give the first and third quartiles, omitted for constant counts. These are descriptive intervals. Gate definitions and within-model totals follow the dynamical-systems table (Table 5).

Layer	Edges				Open gates		
	Edges	Active	$\mathcal{S}$ -only	$\mathcal{S}+\mathcal{F}$	$\mathcal{S}$	$\mathcal{F}$	$\mathcal{R}$
Airfoil , $[5, 5, 1]$, $\beta = 1$							
L0 ($5 \rightarrow 5$)	25	21 [19.25, 21]	6 [5, 7]	18 [17.25, 19]	27 [25, 31]	20 [17, 21]	2 [2, 2]
L1 ($5 \rightarrow 1$)	5	5	0	4 [4, 5]	7 [5, 7.75]	10 [9, 11]	1 [0, 1]
Total	30	26 [24.25, 26]	6 [5, 7]	23 [22, 23.75]	33.5 [31, 36]	30.5 [27, 32]	3 [2, 3]
Concrete , $[13, 32, 1]$, $\beta = 1$							
L0 ($13 \rightarrow 32$)	416	45 [43.25, 48]	23 [19.25, 25.75]	45 [43.25, 48]	27 [24.25, 30]	25 [23, 28.75]	0
L1 ($32 \rightarrow 1$)	32	16 [14.25, 16]	3 [2, 4]	16 [14.25, 16]	7 [6.25, 8.75]	14 [14, 15]	0
Total	448	61 [58, 63.75]	25 [22.25, 28.75]	61 [58, 63.75]	35 [32, 37]	40 [37, 43]	0
Power Plant , $[4, 5, 1]$, $\beta = 1$							
L0 ($4 \rightarrow 5$)	20	11 [10, 12]	6 [4.25, 7]	11 [10, 12]	14 [12, 15.75]	5 [4.25, 6]	0
L1 ($5 \rightarrow 1$)	5	5	0 [0, 0]	5	8 [7, 9]	10 [9, 10]	0
Total	25	16 [15, 17]	6 [5, 7]	16 [15, 17]	22 [19.25, 23]	15 [13.25, 16]	0
Superconductor , $[5, 5, 1]$, $\beta = 1$							
L0 ($5 \rightarrow 5$)	25	12 [10, 14]	5 [4, 6]	10 [8.25, 12]	9.5 [7, 12]	7 [5, 10.75]	2 [1.25, 3]
L1 ($5 \rightarrow 1$)	5	4 [4, 5]	0 [0, 1]	4 [4, 5]	4 [4, 5]	6 [5, 8]	0 [0, 0]
Total	30	16 [14.25, 19]	6 [4, 7]	14 [12.25, 16.75]	14 [11, 19.25]	13 [9.5, 18.75]	2 [2, 3]

C Training and inference times

Table 7 compares training and inference times for the four real-world datasets on an AMD EPYC 9654 (single-threaded). Entries are medians and first and third quartiles over thirty fits per dataset and method (ten data splits and three initialization seeds). Training times cover the complete 5,000-epoch fitting pipeline, including validation checks and checkpoint saving. Prior to inference, both models are exported to an evaluator that omits computations for inactive terms wherever possible. Times are per entire batch of one or 256 observations, excluding preprocessing, model loading, and export. Each model’s latency is the median of eight warmed timing blocks, with paired methods timed sequentially on the same pinned core.

S2KAN’s additional training cost is accompanied by lower inference latency in seven of eight dataset–batch-size settings: batch-256 inference is approximately 1.5–30 times faster, while single-observation inference is faster except on Airfoil.

Table 7: Training and inference wall times for the models in Table 1, on an AMD EPYC 9654 (single-threaded). Entries are medians [first quartile, third quartile] over 30 fits per row. S2KAN takes about 1.6 times as long to train but generally provides faster inference.

Model	Training (min)	Inference	
		Batch 1 (μ s)	Batch 256 (μ s)
Airfoil			
KAN	2.77 [2.73, 2.80]	452.7 [442.0, 472.1]	1128.0 [1106.5, 1172.0]
S2KAN	4.34 [4.24, 4.42]	514.0 [318.4, 531.0]	731.6 [558.3, 842.8]
Concrete			
KAN	4.71 [3.93, 4.94]	484.7 [473.0, 490.2]	3427.9 [3352.8, 3806.8]
S2KAN	7.49 [6.10, 7.75]	66.4 [63.0, 70.2]	113.1 [106.2, 116.8]
Power plant			
KAN	2.74 [2.70, 2.80]	439.7 [423.4, 454.8]	1039.7 [992.0, 1083.3]
S2KAN	4.31 [4.26, 4.34]	58.2 [52.3, 60.5]	83.8 [78.7, 88.4]
Superconductor			
KAN	2.73 [2.47, 2.76]	440.9 [424.2, 452.9]	1089.6 [1050.0, 1101.2]
S2KAN	4.24 [3.89, 4.29]	300.5 [288.4, 376.2]	521.7 [404.5, 556.7]

D Scientific consistency checks

This appendix gives the response definitions, secondary comparisons, and sensitivity checks supporting Sec. 5.2. All analyses use the validation-selected models in Table 1: 30 fits per method and dataset, comprising ten data splits and three initialization seeds per split. Both methods receive the same physical-input queries, transformed using each fit’s preprocessing, and outputs are returned to the original units. Checkpoints remain frozen, with no further training, checkpoint selection, or symbolic refitting. We summarize each fit separately before reporting medians and, where given, first and third quartiles in brackets; split comparisons average the three seeds within each method. These are descriptive summaries across overlapping splits, not confidence intervals from independent experiments. The diagnostics are exploratory, evaluate selected response patterns and physical admissibility, and compare the complete training configurations. Response thresholds specify the diagnostic tolerances and are not universal physical or engineering limits.

D.1 Airfoil self-noise

Post-peak decline and sensitivity. The 1,503 observations form 106 spectra defined by identical angle, chord, speed, and displacement thickness. A spectrum is eligible for the decline diagnostic when it has at least four measurements from its observed maximum onward, a terminal decrease exceeding 3 dB, and no later-minus-earlier increase exceeding 1 dB. These criteria select 92 spectra; the other spectra are excluded from this diagnostic without being classified as physically invalid. Models are evaluated at 257 logarithmically spaced frequencies between the observed peak and the largest recorded frequency, holding the four other inputs fixed. A pass requires an endpoint decrease exceeding 1 dB and a maximum later-minus-earlier increase no greater than 1 dB, with 10^{-6} dB numerical slack for the reversal comparison. Flat responses fail the endpoint requirement. The passing fractions are 70.1% [41.0, 81.0] for KAN and 94.6% [92.4, 95.7] for S2KAN. Replacing both the required decrease and permitted reversal by 0.5 dB gives 44.6% versus 88.0%; using 2 dB gives 91.3% versus 98.9%, respectively. All ten split averages favor S2KAN at each threshold. Increasing the grid to 513 points changes no classifications at the primary 1 dB threshold or at 2 dB, and changes three of 5,520 model–spectrum classifications at 0.5 dB. These checks establish numerical stability at the tested resolutions, not monotonicity between samples. Eligibility uses observed target values across the corpus, including training observations.

Spectral shape and peak localization. For held-out spectral shape, we retain spectra with at least three test frequencies (89–99 spectra per split), subtract each predicted and observed spectrum’s own test-row mean, and compute RMSE with equal weight per spectrum. Median centered errors are 1.427 [1.380, 1.505] dB for KAN and 1.404 [1.361, 1.480] dB for S2KAN, with S2KAN lower in six splits. The peak comparison uses 86 spectra whose observed maximum is interior and at least 3 dB above both endpoints. Predicted peaks are located on a 257-point logarithmic grid spanning each spectrum’s lowest to highest recorded frequency; exact ties select the lowest frequency. Median mean absolute peak-frequency errors are 0.250 [0.238, 0.270] and 0.250 [0.230, 0.263] octaves for KAN and S2KAN. The fractions within one-third octave of the measured peak are 72.7% [68.0, 76.7] and 74.4% [70.9, 77.9], respectively, favoring S2KAN in six splits. Thus peak localization is similar. Peak eligibility uses the full measured spectra; even the held-out shape errors assess interpolation among recorded conditions, since other frequencies at the same condition can occur in fitting data.

Contrasts between recorded speeds. Within identical angle/chord families, 73 adjacent-speed pairs have at least four shared measured frequency bands. Each speed uses the displacement thickness supplied with the dataset, which is computed from the source’s empirical relations for chord, speed, and angle of attack (Brooks et al., 1989). The contrast is the arithmetic mean difference in the dataset’s scaled sound pressure level (dB) across common frequency bands. It evaluates prediction of the supplied response across speeds; it does not directly quantify the change in unscaled sound pressure level or identify a velocity-scaling exponent. For held-out evaluation, at least two common bands must have both speed endpoints in the test set, leaving 29–34 eligible pairs per split. Median contrast mean absolute error (MAE) is 1.208 [1.128, 1.331] dB for KAN and 1.023 [0.940, 1.292] dB for S2KAN, with S2KAN lower in nine splits. Direction agreement on held-out contrasts exceeding 1 dB in observed absolute magnitude is 89.7% [84.6, 91.5] versus 94.7% [84.4, 98.9], favoring S2KAN in seven splits. Across the full corpus, however, direction agreement at the same magnitude threshold favors KAN: 95.7% [91.3, 95.7] versus 91.3% [89.7, 91.3], with KAN higher in eight splits, S2KAN higher in one, and one tie. The speed comparisons therefore provide mixed evidence and do not establish generally better velocity dependence.

D.2 Concrete compressive strength

The aging diagnostics use the 181 exact-composition recipes with at least two observed ages. Each recipe receives equal weight and is evaluated on 257 equally spaced ages within its recorded range, varying raw age and the supplied $\log(\text{age} + 1)$ feature together. The decline score is the largest earlier-to-later decrease anywhere on that grid; a recipe violates the criterion when this exceeds 1 MPa. The slowing-growth criterion compares endpoint secant slopes over the first and last thirds of the grid: indices 0–85 and 171–256, respectively, with zero-based indexing. A pass requires $0 < s_{\text{late}} < s_{\text{early}}$. This excludes a flat late interval and does not require the complete curve to be everywhere increasing or concave; a physical plateau can therefore fail this particular criterion. The two scores describe related properties of the same model curves and do not constitute independent discoveries or held-out aging-accuracy tests.

On a one-day grid, the median fraction of recipes with declines exceeding 1 MPa is 8.3% for S2KAN versus 62.7% for KAN, favoring S2KAN in all ten splits under both mean and median aggregation. Grids finer than the dataset’s one-day age-recording resolution detect additional declines in both methods, including sub-hour dips from protected logarithmic activations in some S2KAN fits; the overall median decline frequency remains lower for S2KAN.

The concrete illustration in Fig. 2 uses split and initialization seed 42. The recipe was selected by proximity to the median standardized ingredient composition among the 181 eligible recipes, without strengths or model predictions. All native activations, fixed-input offsets, target scaling, and protected operations are retained; the complete prediction includes all 32 branches. The four displayed branches were ranked by total variation of their output contributions after extraction. For the dominant branch, the increasing and concave contribution was verified throughout the displayed 3–365-day interval from its native coefficients. This property of one contribution does not imply concavity of every branch or every fitted model. Contributions from other branches introduce local increases in the rate of gain, including a corner near day 16 associated with a protected square root; the complete prediction nevertheless remains increasing on the evaluated grid. This corner is an implementation feature, not an identified curing transition.

D.3 Power plant electrical output

Local support and directional criteria. We sampled 512 anchors from distinct recorded input vectors using inputs alone. Temperature sweeps vary ambient temperature by $\pm 2^\circ\text{C}$, and pressure sweeps vary atmospheric pressure by ± 4 mbar, with the other coordinates fixed. Each sweep has 65 equally spaced points. We reject a sweep if any point leaves the corpus marginal ranges or its nearest-neighbor distance to a recorded input, in corpus-standardized four-dimensional input space, exceeds the 95th percentile of corpus leave-one-out nearest-neighbor distances. This retains 325 temperature and 148 pressure sweeps. A stricter 90th-percentile screen retains 212 and 71 sweeps, respectively. Support uses all corpus inputs and is a geometric restriction, not proof of training support or an achievable controlled intervention. In particular, holding relative humidity fixed while changing temperature does not hold absolute humidity fixed.

The expected directions are decreasing power with temperature and increasing power with pressure. For either direction, a pass requires a net change exceeding 0.2 MW in that direction and no earlier-to-later excursion exceeding 1 MW in the opposite direction. Temperature passing fractions are 98.2% [93.8, 99.4] for KAN and 100.0% [99.7, 100.0] for S2KAN. The corresponding fractions with an upward excursion exceeding 1 MW are 1.23% [0.00, 5.15] and 0.00% [0.00, 0.00]. For pressure, passing fractions are 35.8% [25.5, 45.6] and 54.1% [47.8, 57.8], while wrong-way excursions exceeding 1 MW occur in 61.5% [51.5, 71.6] and 24.7% [5.4, 30.4] of sweeps, respectively. Both primary endpoints favor S2KAN in all ten split averages for each variable, but substantial pressure-response failures remain.

Observed-data checks and sensitivity. To assess the directional evidence in recorded observations, we match each input to its nearest eligible partner in the other three standardized coordinates, with each coordinate difference at most 0.2 standard deviations and a focal difference of 1–4 $^\circ\text{C}$ or 2–8 mbar. Duplicate input vectors are averaged, and a 0.1-standard-deviation matching box provides a sensitivity check. Temperature pairs show the expected decrease in 76.5–78.2% of comparisons, whereas pressure pairs show the expected increase in only 51.3–53.6%. Pairs share observations and describe associations, not independent interventions. Under the stricter support screen, temperature passing fractions are 98.35% for KAN and 100.00% for S2KAN; pressure passing fractions are 30.99% and 46.48%. All ten split averages still favor S2KAN, but its median desired pressure net change in this subset is only 0.051 MW over the 8 mbar interval. Fewer reversals therefore do not establish recovery of a nontrivial pressure law. Doubling the evaluation grid to 129 points changes three KAN temperature classifications for excursions exceeding 1 MW, and none for S2KAN or pressure. Across excursion thresholds of 0.1, 0.5, 1, and 2 MW and net-change thresholds of 0, 0.2, and 0.5 MW, favorable-split counts for the directional pass criteria range from eight to ten; the advantage is not universal for every sensitivity and split.

The symbolic temperature example uses split and initialization seed 42 at fixed exhaust vacuum 46.63 cm Hg, atmospheric pressure 1013.16 mbar, and relative humidity 79.08%. The context was selected using inputs before inspecting the native activations; the polynomial-to-sine branch was subsequently chosen for readability. All five branches remain in the prediction, including a small opposing contribution. The reported branch and full-model decreases are exact endpoint contributions in MW, not measured intervention effects or evidence that a larger predicted decrease is more accurate.

D.4 Superconductor critical temperature

The physical-admissibility check measures the fraction of negative predicted critical temperatures among each fit’s 1,000 untouched test observations, without clipping. The clipping control replaces predictions by $\max(\hat{T}_c, 0)$ without retraining, and computes test RMSE on the same observations.

E Dynamical-system diagnostics

These diagnostics assess recovery of dynamical behavior beyond one-step prediction accuracy for the models in Table 2. Lyapunov estimation is described in Methods (App. A); additional evaluation procedures are given below.

E.1 Lyapunov spectra

Table 8 shows similar median leading Lyapunov exponents for all Ikeda configurations, close to the reference value 0.5081. All spectra are defined, but one S2KAN seed at $\beta = 1$ has a negative leading exponent and incorrect long-run behavior. For the ecosystem, KAN and unregularized S2KAN have median leading exponents 0.00666 and 0.00688, respectively, versus the reference 0.00814; each has three nonpositive estimates. At $\beta = 1$, S2KAN’s median falls near zero and one trajectory diverges. Agreement of medians therefore does not imply recovery in every seed.

E.2 Temporal structure

To assess whether the learned models reproduce the reference systems’ temporal dependence and oscillatory behavior, we compare autocorrelation functions (ACFs) and power spectral densities (PSDs) on the model and reference orbits described in Methods (App. A).

Coordinate-wise PSDs use Welch’s method with segment lengths 1,024 (Ikeda) or 4,096 (ecosystem), segment-mean subtraction, a periodic Hann window, 50% overlap, and one-sided density scaling. ACFs use the demeaned sample covariance with denominator N , the number of samples in the analysis window (10,000 for Ikeda and 20,000 for the ecosystem), normalized

Table 8: Full Lyapunov spectra of the models in Table 2. Entries are medians [first quartile, third quartile] over defined spectra; “Defined” gives their count out of ten fits. The Ikeda map has two exponents, so λ_3 is inapplicable. Reference entries are finite-orbit estimates for the generating map. $|\Delta\lambda_1|$ is the absolute difference between leading-exponent estimates from the first and second halves of the evaluated orbit.

Model	β	Defined	λ_1	λ_2	λ_3	$ \Delta\lambda_1 $
Ikeda map: exponents per iterate						
Reference	—	—	0.508	-0.719	—	—
			0.508	-0.727	—	0.0059
KAN	—	10/10	[0.505, 0.511]	[-0.731, -0.722]	—	[0.0040, 0.0094]
			0.508	-0.725	—	0.0074
S2KAN	0	10/10	[0.506, 0.512]	[-0.731, -0.723]	—	[0.0046, 0.0108]
			0.508	-0.727	—	0.0068
S2KAN	1	10/10	[0.501, 0.512]	[-0.729, -0.725]	—	[0.0037, 0.0143]
			0.507	-0.742	—	0.0044
S2KAN	3	10/10	[0.502, 0.508]	[-0.756, -0.730]	—	[0.0009, 0.0115]
Ecosystem: exponents per unit time						
Reference	—	—	0.00814	0.00007	-0.39132	—
			0.00666	-0.00013	-0.29512	0.00115
KAN	—	10/10	[0.00092, 0.00861]	[-0.00189, 0.00002]	[-0.33723, -0.27225]	[0.00045, 0.00262]
			0.00688	-0.00019	-0.37177	0.00081
S2KAN	0	10/10	[0.00002, 0.00926]	[-0.00099, 0.00002]	[-0.38882, -0.33182]	[0.00052, 0.00193]
			0.00664	-0.00022	-0.36529	0.00065
S2KAN	0.1	10/10	[-0.00002, 0.00909]	[-0.00316, -0.00001]	[-0.38974, -0.35080]	[0.00046, 0.00072]
			0.00001	-0.00392	-0.31852	0.00041
S2KAN	1	9/10	[-0.00003, 0.00768]	[-0.01525, -0.00031]	[-0.35507, -0.10222]	[0.00023, 0.00050]

Table 9: Rollout diagnostics for the models in Table 2. Entries are medians [first quartile, third quartile] over ten fits, with undefined multi-step RMSEs excluded as described in the text. Horizon is in discrete steps and late overlap lies in $[0, 1]$.

Model	β	Multi-step RMSE	Horizon	Late overlap
Ikeda map				
KAN	—	0.858 [0.851, 0.871]	9.0 [8.5, 9.0]	0.913 [0.910, 0.914]
S2KAN	0	0.866 [0.856, 0.876]	9.0 [8.0, 10.0]	0.913 [0.909, 0.914]
S2KAN	1	0.861 [0.855, 0.871]	8.5 [7.6, 9.0]	0.909 [0.907, 0.912]
S2KAN	3	0.873 [0.863, 0.875]	7.2 [6.5, 7.5]	0.904 [0.897, 0.910]
Ecosystem				
KAN	—	0.130 [0.120, 0.147]	270.8 [261.9, 408.5]	0.831 [0.784, 0.911]
S2KAN	0	0.093 [0.060, 0.122]	396.2 [308.6, 464.4]	0.818 [0.745, 0.869]
S2KAN	0.1	0.080 [0.075, 0.134]	313.8 [294.6, 385.9]	0.841 [0.784, 0.871]
S2KAN	1	0.138 [0.131, 0.171]	177.5 [94.5, 222.5]	0.674 [0.309, 0.729]

at lag zero, through 100 iterates or 1,000 time units, respectively. Each diagnostic uses the second half of a completed model orbit; pointwise medians and IQRs summarize seeds, with PSD quantiles computed in linear power. Constant-coordinate ACFs are undefined and excluded from ACF summaries; this affects one ecosystem S2KAN fit at $\beta = 1$, whose PSD remains included. Reference curves average diagnostics over ten 10,000-point Ikeda blocks or five 20,000-point ecosystem blocks from the reference orbit’s final 100,000 points, matching each model’s analysis-window length. Frequencies use sampling intervals 1 and $\Delta t = 0.5$, respectively; only the zero-frequency bin is omitted from the logarithmic PSD display.

Figure 5 shows broadly similar median ACFs and PSDs for Ikeda. Ecosystem KAN and unregularized S2KAN retain the broad reference oscillatory patterns, whereas S2KAN at $\beta = 1$ has more sharply peaked spectra and less broadband low-frequency power. Of the 60 trajectories represented in Fig. 5, one ecosystem trajectory at $\beta = 1$ diverges and another converges to a constant state. Completed trajectories remain included wherever the diagnostic is defined, so these median curves should be read alongside the seed variation and exceptions above.

E.3 Forecasts and long-run state distributions

Multi-step RMSE averages coordinate-wise RMSEs over 2,000 Ikeda steps or 1,250 ecosystem steps from the first unseen state; divergent rollouts yield undefined RMSE. This differs from pooling coordinates before taking the square root for one-step RMSE. Forecast horizons use 20 evenly spaced initial states with a full evaluation window in the unseen trajectory. State errors are divided coordinate-wise by reference standard deviations over the same 2,000- or 1,250-step unseen window (plus 10^{-12}). Each horizon counts the steps before the root-mean-square standardized error first exceeds 0.2 or becomes nonfinite, capped at 200 steps (Ikeda) or 500 (ecosystem); the fit’s horizon is the median over the 20 starts.

Late density overlap pools samples from the final third of 20 rollouts started across the unseen block, each lasting 6,000 steps for Ikeda or 3,750 for the ecosystem. The reference “seen block” comprises the training and test partitions. Overlap is $\sum_b \min(p_b, q_b)$ for model and seen-block reference joint histogram masses, using 25 or 10 bins per coordinate, respectively, on seen-block bounds padded by 5%. Normalization retains all requested samples; nonfinite or out-of-box mass is lost. Nonfinite states or magnitudes above 10^6 terminate trajectories. Initial states are not independent model replicates.

Table 9 complements pointwise errors with forecast horizons and late density overlap. Reference-map overlaps are 0.916 for Ikeda and 0.903 for the ecosystem, illustrating finite-sample differences even between reference trajectories. Multi-step RMSE is defined for nine ecosystem S2KAN fits at $\beta = 1$ and all ten fits otherwise; all fits contribute to horizon and overlap. Nine of the 200 late-evaluation rollouts diverge for ecosystem S2KAN at $\beta = 1$, with none elsewhere. These counts differ from the spectral evaluation because the initial states and evaluation horizons differ.

For Ikeda, S2KAN at $\beta = 1$ has a slightly shorter median forecast horizon than KAN (8.5 versus 9 steps) and similar late overlap (0.909 versus 0.913). For the ecosystem, unregularized S2KAN lowers multi-step RMSE in nine of ten paired seeds and lengthens the horizon in seven, although its median late overlap is slightly lower (0.818 versus 0.831). At $\beta = 1$, ecosystem overlap falls to 0.674. Thus smaller models can retain useful dynamical behavior, but the appropriate regularization strength depends on the system and the property being evaluated.

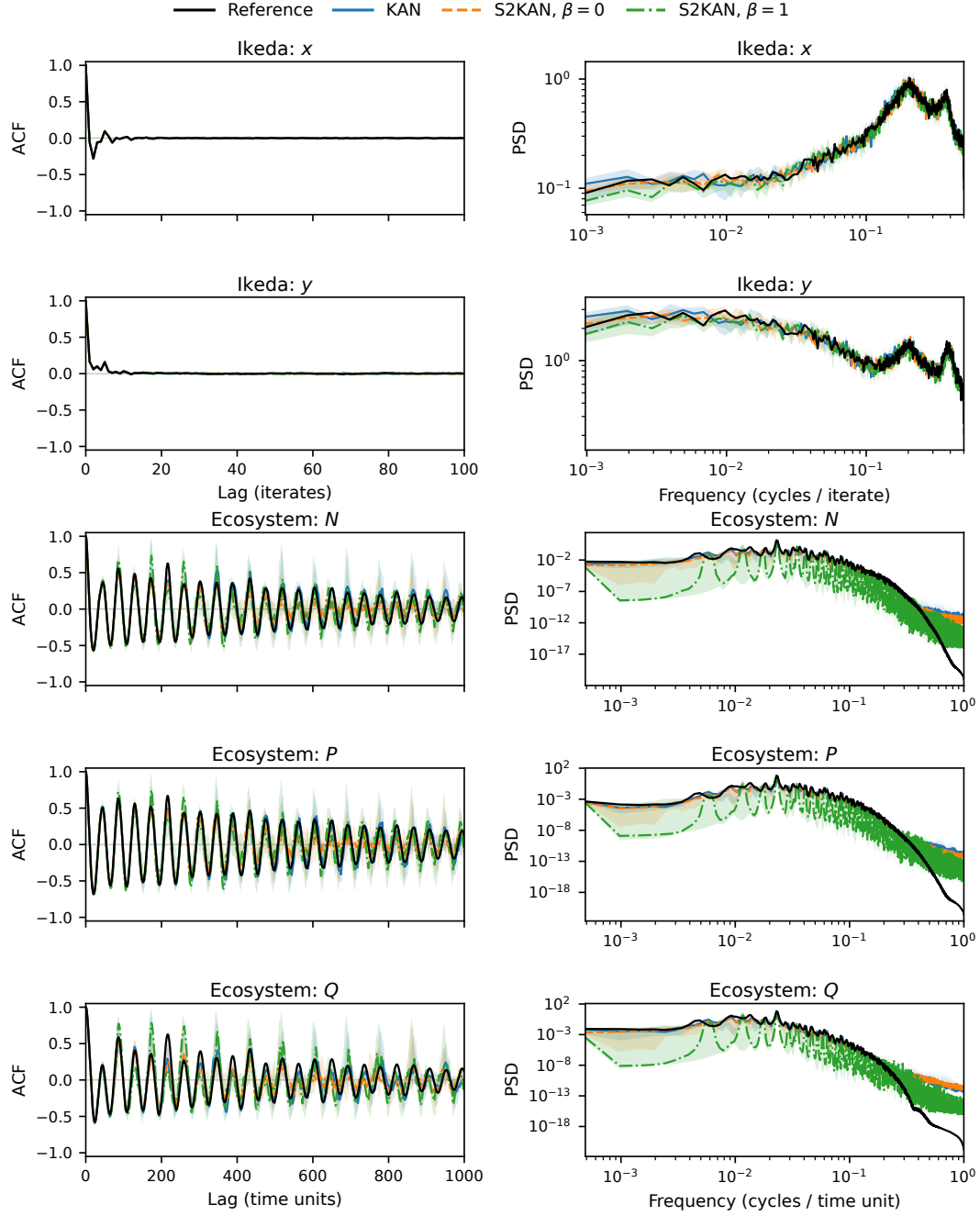


Figure 5: Temporal structure of the learned dynamical systems measured using autocorrelation functions (left) and power spectral densities (right), for each state coordinate. Colored curves and shading show pointwise medians and IQRs across completed fits; black curves are reference block means. All shown Ikeda, KAN, and $\beta = 0$ configurations contribute 10/10 fits. Ecosystem S2KAN at $\beta = 1$ contributes nine PSDs and eight ACFs: one trajectory diverges and another is constant, making its ACF undefined. Other completed trajectories remain included.

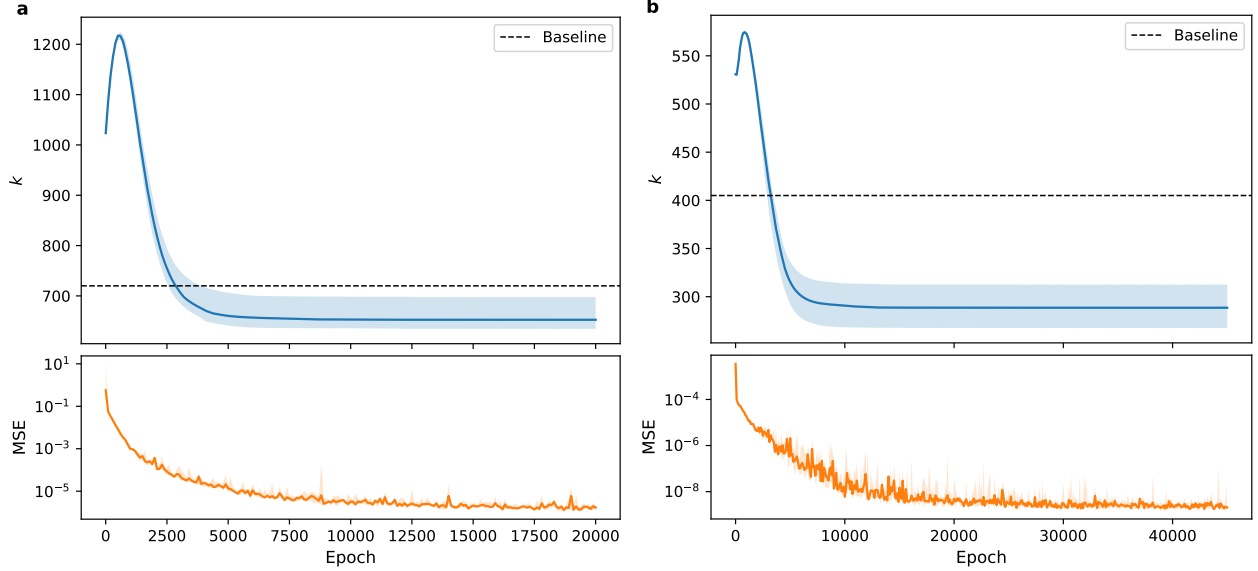


Figure 6: Self-sparsification of the Ikeda map (a) and Ecosystem networks (b), both at $\beta = 0$. Solid curves show medians across ten seeds; shading indicates the interquartile range. Horizontal dashed lines indicate the effective parameter count k of baseline KAN. Counts are expected effective parameter counts, rather than hard gate or active-edge counts. The first point follows the first completed epoch. Training MSEs are epoch-averaged stochastic minibatch values; shading shows descriptive interquartile ranges, not confidence intervals.

E.4 Self-sparsification under the tested training configurations

For the $\beta = 0$ fits in Table 2, we tracked the expected effective parameter count k throughout training. In both systems, its median rises early and then falls below the baseline KAN count (Fig. 6). Median counts decrease from 1,023.5 after the first epoch to 652.5 at the fixed 20,000-epoch endpoint for Ikeda, compared with 720 for baseline KAN. For the ecosystem, the corresponding medians are 530.7 and 288.5 at the fixed 45,000-epoch endpoint, compared with 405 for baseline KAN. The first-epoch count follows one completed epoch and is not the untrained initialization. All ten fits per system finish below their first-epoch counts; nine of ten Ikeda fits and all ten ecosystem fits finish below baseline KAN. The remaining Ikeda fit ends at $k = 720.049$, slightly above the baseline count. Thus shrinkage during training and a final size below baseline KAN are distinct observations; the latter holds at the median for both systems. Training continued to fixed endpoints without early stopping. These results document self-sparsification under the tested configurations, without establishing its mechanism or its robustness to changes in optimizer, gate parameterization, dictionary, architecture, or data.

F Nguyen benchmark variability

For completeness, Table 10 reports seed variability for the full Nguyen comparison in Table 3, using the same fits for KAN and S2KAN across architectures S, L and LM.

G Ablations and controls

G.1 Baseline KAN loss control

To separate the choice of training loss from the model comparison, we retrained baseline KAN using either raw MSE or $\frac{1}{2} \log(\text{MSE} + c_{\text{loss}})$ on all four real-world datasets. Each loss uses the ten data splits and three initialization seeds of Table 1, giving 120 paired comparisons. Within each pair, initialization, data, preprocessing, architecture, optimizer, and training schedule are identical; both fits run on the same AMD EPYC 9654 host with one thread each. All gates remain fixed at one, with no complexity penalty. Both arms train for 5,000 epochs and select checkpoints by minimum validation MSE, evaluated every ten epochs. SF uses the same procedure and dictionaries as the main results.

Table 10: Seed variability for Table 3. Entries are [first, third quartiles] over ten initialization seeds; S2KAN uses $\beta = 1$.

ID	Shape	Model	Test accuracy		Model size		SF (%), $\epsilon = 0.01$			SF (%), $\epsilon = 0.1$		
			R^2	RMSE	Active	k	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
F1	S	KAN	[1.0000, 1.0000]	$[0.20, 1.99] \times 10^{-5}$	[1, 1]	[15, 15]	[0, 0]	[100, 100]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[0.58, 1.24] \times 10^{-3}$	[1, 1]	[20.1, 21.1]	[0, 0]	[100, 100]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[4.27, 7.10] \times 10^{-4}$	[6, 6]	[90, 90]	[0, 12.5]	[0, 12.5]	[4.2, 31.2]	[27.1, 33.3]	[61.7, 95.8]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[3.30, 8.20] \times 10^{-3}$	[6, 6]	[38.4, 51.3]	[21.3, 40]	[76.2, 95]	[100, 100]	[50, 60]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[3.36, 5.99] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 0]	[0, 0]	[11.5, 24.3]	[25, 44.4]	[65.6, 77.8]	[87.5, 100]
		S2KAN	[1.0000, 1.0000]	$[2.96, 5.78] \times 10^{-3}$	[9, 9]	[57.2, 67.9]	[28.6, 42.9]	[100, 100]	[100, 100]	[57.1, 69.2]	[100, 100]	[100, 100]
F2	S	KAN	[1.0000, 1.0000]	$[1.00, 1.59] \times 10^{-4}$	[1, 1]	[15, 15]	[0, 0]	[100, 100]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[0.44, 1.63] \times 10^{-3}$	[1, 1]	[18.5, 19.5]	[0, 0]	[100, 100]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[3.63, 6.54] \times 10^{-4}$	[6, 6]	[90, 90]	[0, 12.5]	[16.7, 33.3]	[33.3, 50]	[33.3, 50]	[70.8, 100]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[4.50, 9.98] \times 10^{-3}$	[6, 6]	[40, 45.4]	[52.5, 72.9]	[80, 100]	[100, 100]	[76.2, 100]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[2.96, 6.12] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 0]	[0, 11.1]	[16.5, 33.3]	[29.8, 42.7]	[67.9, 77.8]	[78.1, 100]
		S2KAN	[1.0000, 1.0000]	$[4.73, 5.59] \times 10^{-3}$	[6.2, 9]	[39.5, 58.3]	[42.5, 66.7]	[100, 100]	[100, 100]	[80.8, 100]	[100, 100]	[100, 100]
F3	S	KAN	[1.0000, 1.0000]	$[2.23, 2.74] \times 10^{-4}$	[1, 1]	[15, 15]	[0, 0]	[0, 0]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[4.35, 7.44] \times 10^{-4}$	[1, 1]	[19.3, 20.1]	[0, 0]	[0, 0]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[0.47, 1.21] \times 10^{-3}$	[6, 6]	[90, 90]	[0, 0]	[0, 16.7]	[16.7, 33.3]	[33.3, 33.3]	[66.7, 83.3]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[0.43, 1.20] \times 10^{-2}$	[6, 6]	[53.2, 59.5]	[28.7, 40]	[75, 80]	[80, 95.8]	[42.5, 60]	[85, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[2.35, 7.37] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 0]	[0, 12.2]	[2.8, 24.3]	[12.5, 31.2]	[57.3, 75]	[78.1, 100]
		S2KAN	[1.0000, 1.0000]	$[3.19, 8.40] \times 10^{-3}$	[9, 9]	[63.9, 70]	[20.8, 42.9]	[83.9, 100]	[100, 100]	[42.9, 55.4]	[100, 100]	[100, 100]
F4	S	KAN	[1.0000, 1.0000]	$[5.46, 6.05] \times 10^{-4}$	[1, 1]	[15, 15]	[0, 0]	[0, 0]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[6.66, 9.18] \times 10^{-4}$	[1, 1]	[19.1, 20]	[0, 0]	[0, 0]	[100, 100]	[0, 0]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[0.74, 1.78] \times 10^{-3}$	[6, 6]	[90, 90]	[0, 0]	[0, 12.5]	[4.2, 33.3]	[16.7, 33.3]	[52.5, 79.2]	[70, 100]
		S2KAN	[1.0000, 1.0000]	$[4.35, 9.11] \times 10^{-3}$	[5.2, 6]	[46.6, 59]	[50, 60]	[76.2, 80]	[80, 100]	[75, 80]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[4.88, 7.92] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 0]	[0, 12.5]	[15.6, 31.2]	[22.9, 37.5]	[62.5, 77.1]	[75.7, 96.9]
		S2KAN	[0.9999, 1.0000]	$[0.41, 1.24] \times 10^{-2}$	[9, 9]	[55.1, 64.4]	[7.1, 53.6]	[89.3, 100]	[100, 100]	[57.1, 68.6]	[100, 100]	[100, 100]
F5	S	KAN	[1.0000, 1.0000]	$[4.50, 7.09] \times 10^{-5}$	[1, 1]	[15, 15]	[0, 0]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
		S2KAN	[0.9998, 1.0000]	$[0.54, 2.35] \times 10^{-3}$	[1, 1]	[19.1, 19.9]	[0, 0]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[0.92, 2.43] \times 10^{-4}$	[6, 6]	[90, 90]	[0, 19.2]	[5, 38.3]	[17.5, 50]	[23.3, 50]	[61.7, 82.5]	[100, 100]
		S2KAN	[0.8682, 0.9856]	$[1.99, 6.02] \times 10^{-2}$	[6, 6]	[57.5, 58.7]	[52.5, 75]	[100, 100]	[100, 100]	[75, 100]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[0.72, 1.87] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 12.5]	[11.5, 21.9]	[14.9, 34.4]	[15.6, 37.5]	[53.1, 66.7]	[78.1, 100]
		S2KAN	[0.2745, 0.9171]	$[0.47, 1.37] \times 10^{-1}$	[7, 9]	[65.9, 77.3]	[66.7, 84.3]	[100, 100]	[100, 100]	[83.3, 100]	[100, 100]	[100, 100]
F6	S	KAN	[1.0000, 1.0000]	$[6.37, 9.51] \times 10^{-5}$	[1, 1]	[15, 15]	[0, 0]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[0.81, 1.97] \times 10^{-3}$	[1, 1]	[23.2, 24.1]	[0, 0]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[2.16, 5.42] \times 10^{-4}$	[6, 6]	[90, 90]	[0, 0]	[16.7, 33.3]	[35, 50]	[52.5, 76.7]	[87.5, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[2.06, 4.10] \times 10^{-3}$	[6, 6]	[37.1, 44.2]	[65, 95]	[100, 100]	[100, 100]	[85, 100]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[2.13, 6.04] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 0]	[11.1, 12.5]	[22.9, 41.7]	[34.4, 42.7]	[75, 85.1]	[87.8, 100]
		S2KAN	[0.9999, 1.0000]	$[5.15, 9.13] \times 10^{-3}$	[9, 9]	[55.2, 63.7]	[58.5, 74.1]	[100, 100]	[100, 100]	[85.7, 87.5]	[100, 100]	[100, 100]
F7	S	KAN	[1.0000, 1.0000]	$[0.91, 3.49] \times 10^{-5}$	[1, 1]	[15, 15]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[1.15, 6.63] \times 10^{-3}$	[1, 1]	[11.3, 21.2]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[2.59, 9.16] \times 10^{-4}$	[6, 6]	[90, 90]	[0, 0]	[0, 16.7]	[23.3, 57.5]	[42.5, 65]	[80, 83.3]	[87.5, 100]
		S2KAN	[0.9988, 0.9999]	$[0.88, 2.87] \times 10^{-2}$	[6, 6]	[55.5, 57.4]	[50, 60]	[100, 100]	[100, 100]	[63.7, 100]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[2.57, 5.04] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 0]	[0, 12.2]	[22.2, 46.9]	[27.1, 58]	[75, 88.9]	[100, 100]
		S2KAN	[0.9947, 0.9994]	$[1.69, 5.99] \times 10^{-2}$	[7.5, 9]	[69.3, 76.1]	[51.8, 71.4]	[100, 100]	[100, 100]	[71.4, 85.7]	[100, 100]	[100, 100]
F8	S	KAN	[1.0000, 1.0000]	$[1.82, 1.84] \times 10^{-3}$	[1, 1]	[15, 15]	[0, 0]	[0, 0]	[0, 0]	[100, 100]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[1.87, 2.32] \times 10^{-3}$	[1, 1]	[19.1, 19.9]	[0, 0]	[0, 0]	[0, 0]	[100, 100]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[2.39, 5.70] \times 10^{-4}$	[6, 6]	[90, 90]	[0, 0]	[0, 22.9]	[17.5, 40]	[42.5, 57.5]	[80, 95.8]	[87.5, 100]
		S2KAN	[0.9981, 0.9997]	$[0.74, 2.03] \times 10^{-2}$	[6, 6]	[48.6, 54.2]	[20, 57.5]	[63.7, 80]	[100, 100]	[52.5, 95]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[1.68, 2.85] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 8.3]	[0, 12.2]	[12.9, 25]	[29.8, 48.6]	[75, 88.5]	[87.5, 100]
		S2KAN	[0.9985, 0.9997]	$[0.87, 1.79] \times 10^{-2}$	[7.2, 9]	[52.7, 63.1]	[21.3, 48.2]	[72.3, 95.8]	[100, 100]	[61.7, 71.4]	[100, 100]	[100, 100]
F9	S	KAN	[1.0000, 1.0000]	$[5.20, 9.67] \times 10^{-5}$	[2, 2]	[30, 30]	[50, 50]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
		S2KAN	[1.0000, 1.0000]	$[0.59, 1.84] \times 10^{-3}$	[2, 2]	[22.2, 36.2]	[50, 50]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
	L	KAN	[1.0000, 1.0000]	$[1.46, 4.07] \times 10^{-4}$	[9, 9]	[135, 135]	[0, 27.7]	[28.6, 62.5]	[57.1, 90.6]	[57.9, 96.4]	[85.7, 100]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[2.89, 4.43] \times 10^{-3}$	[9, 9]	[39.8, 43.7]	[70.8, 100]	[100, 100]	[100, 100]	[87.5, 100]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[2.67, 6.28] \times 10^{-4}$	[14, 14]	[210, 210]	[0, 5.8]	[7.9, 16.3]	[40, 49]	[44.7, 73.6]	[91.7, 100]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[2.23, 5.18] \times 10^{-3}$	[13, 14]	[49.5, 53.3]	[68.8, 88.1]	[100, 100]	[100, 100]	[80.2, 97.5]	[100, 100]	[100, 100]
F10	S	KAN	$[-0.0374, -0.0372]$	[1.01, 1.01]	[2, 2]	[30, 30]	[0, 0]	[0, 0]	[0, 0]	[0, 0]	[0, 0]	[0, 0]
		S2KAN ^a	$[-0.0006, -0.0006]$	$[9.90, 9.90] \times 10^{-1}$	[0, 0]	[0, 0]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]
	L	KAN	[0.9999, 1.0000]	$[0.40, 1.01] \times 10^{-2}$	[9, 9]	[135, 135]	[0, 0]	[25, 52.8]	[68.8, 88.1]	[25, 62.5]	[80.6, 100]	[100, 100]
		S2KAN	[0.9999, 1.0000]	$[3.12, 6.96] \times 10^{-3}$	[9, 9]	[20.1, 24.1]	[55.6, 96.9]	[100, 100]	[100, 100]	[87.5, 100]	[100, 100]	[100, 100]
	LM	KAN	[1.0000, 1.0000]	$[3.07, 5.05] \times 10^{-3}$	[14, 14]	[210, 210]	[0, 0]	[7.3, 21.4]	[37.5, 68]	[21.4, 48.2]	[73.2, 90.7]	[92.9, 100]
		S2KAN	[1.0000, 1.0000]	$[1.39, 2.40] \times 10^{-3}$	[8, 10]	[24.5, 29.3]	[68.8, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]	[100, 100]

^a All ten S2KAN fits for F10 with shape S have no functionally active edges; SF is 100% by convention. Metric and shape definitions follow Table 3.

Table 11: Baseline KAN trained with raw or log MSE. Values are medians over thirty fits; smaller brackets give IQRs, omitted for constant values. All checkpoints are validation-selected. Parameter counts are unchanged by the loss choice.

		SF (%), $\epsilon = 0.01$			SF (%), $\epsilon = 0.1$		
KAN loss	RMSE	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
Airfoil: RMSE in dB							
Raw MSE	2.047 [1.963, 2.132]	10.5 [7.2, 13.5]	26.8 [24.4, 28.6]	28.6 [28.6, 31.0]	14.3 [11.1, 17.7]	39.3 [36.0, 46.0]	59.6 [51.8, 62.1]
Log MSE	2.021 [1.972, 2.110]	11.1 [10.3, 14.2]	25.9 [25.0, 28.6]	28.6 [28.6, 31.0]	14.8 [13.8, 17.9]	43.7 [37.6, 44.8]	58.2 [54.0, 65.2]
Concrete: RMSE in MPa							
Raw MSE	4.900 [4.488, 5.234]	0.0 [0.0, 0.3]	2.6 [1.9, 3.4]	12.9 [11.6, 13.9]	26.2 [19.3, 29.0]	61.5 [47.2, 65.1]	83.2 [71.5, 86.9]
Log MSE	4.686 [4.495, 5.067]	0.3 [0.0, 0.6]	2.7 [1.9, 3.6]	13.3 [12.3, 14.6]	23.1 [19.5, 28.9]	56.1 [50.0, 64.2]	78.9 [73.7, 87.7]
Power plant: RMSE in MW							
Raw MSE	4.408 [4.269, 4.496]	0.0	0.0	0.0 [0.0, 7.2]	12.8 [8.7, 20.0]	51.0 [43.6, 56.4]	74.5 [68.4, 80.3]
Log MSE	4.330 [4.232, 4.514]	0.0	0.0	4.1 [0.0, 4.5]	13.3 [8.7, 23.4]	54.4 [47.9, 59.6]	82.2 [72.7, 86.8]
Superconductor: RMSE in K							
Raw MSE	18.803 [18.222, 19.431]	0.0	0.0 [0.0, 3.7]	16.7 [15.3, 16.7]	3.3 [0.0, 3.6]	19.3 [16.7, 23.3]	33.9 [30.0, 36.7]
Log MSE	18.874 [18.009, 19.381]	0.0	3.3 [0.0, 3.5]	16.7 [15.3, 16.7]	3.3 [0.0, 3.6]	18.9 [14.5, 24.6]	33.3 [26.7, 39.8]

We restrict this loss control to unregularized KAN. For S2KAN, replacing log MSE with raw MSE makes the balance with the complexity penalty depend on the scale of the squared error, so retaining $\beta = 1$ would not preserve the regularization strength; a raw-loss comparison would require separate calibration of β .

Log loss reduces median test RMSE from 2.047 to 2.021 dB on airfoil, 4.900 to 4.686 MPa on concrete, and 4.408 to 4.330 MW on power plant, while increasing it from 18.803 to 18.874 K on superconductors (Table 11). Log loss changes KAN’s SF in both directions; S2KAN’s median SF remains higher in all 24 dataset–setting comparisons. The unchanged S2KAN models in Table 1 also retain substantially smaller parameter counts. Thus the SF and size advantages persist when both methods use a log-MSE data term; predictive accuracy remains dataset dependent.

G.2 Activation-family ablations

We test two restrictions of the activation dictionary. No-dense S2KAN disables both spline and SiLU contributions throughout training and inference. Dense-only S2KAN retains only this jointly gated branch, with an empty symbolic library and $P = Q = 0$; its dense gates remain trainable. The no-dense variant yields a model related to equation-learner architectures (Martius & Lampert, 2016; Sahoo et al., 2018), although this is an internal ablation of S2KAN rather than a reproduction of EQL.

G.2.1 Real-world datasets

We compare both controls with the full S2KAN models in Table 1, using all four datasets, ten data splits and three initialization seeds per split (120 models per method). All three methods use the same architecture, log-MSE objective, $\beta = 1$ after 500 warmup epochs, constant learning rate 10^{-3} , batch size 64, and 5,000 training epochs. Checkpoints minimize validation MSE evaluated every ten epochs; fitting, validation and test samples and preprocessing are identical between methods. The controls use the same EPYC 9654 central processing unit (CPU) class as the archived full models but were trained separately. Table 12 includes all fits and uses the same SF procedure and evaluation dictionary for every method.

Full S2KAN retains dense terms in all 30 Airfoil and all 30 Superconductor models, and achieves lower median test RMSE than the no-dense control on both datasets (2.022 versus 2.087 and 19.514 versus 20.068, respectively). Its RMSE is lower in nine of ten splits for each dataset after averaging over the three initialization seeds. These gains accompany larger models: median k is 103.8 versus 64.4 for Airfoil and 61.3 versus 41.6 for Superconductor. On Concrete and Power Plant, all full-S2KAN models close their dense gates; the control has lower median error on Concrete, with larger k , while Power Plant differs little in either error or size. Across the six correlated SF settings and four datasets, 18 median comparisons favor the control, four favor full S2KAN, and two tie. Across all 24 comparisons, including ties, the median SF difference (full S2KAN minus no-dense S2KAN) is -1.9 percentage points. This comparison shows a dataset-dependent tradeoff among predictive accuracy, representation size and symbolic fidelity.

Table 12: Activation-family ablations on four real-world datasets. Entries are medians over 30 fits per dataset and method, with IQRs in brackets (omitted for constant values).

(a) Prediction and model size							
Dataset	Model	RMSE	R^2	Active	k		
Airfoil	Full S2KAN	2.022 [1.954, 2.048]	0.912 [0.909, 0.920]	26 [24.2, 26.0]	103.8 [94.7, 114.1]		
	No-dense S2KAN	2.087 [2.021, 2.148]	0.908 [0.902, 0.915]	26 [25.0, 26.0]	64.4 [62.1, 69.8]		
	Dense-only S2KAN	2.334 [2.168, 2.414]	0.886 [0.875, 0.898]	14 [13.0, 16.0]	211.9 [197.3, 240.0]		
Concrete	Full S2KAN	5.268 [4.996, 5.612]	0.894 [0.885, 0.906]	61 [58.0, 63.8]	76.2 [71.7, 80.4]		
	No-dense S2KAN	5.165 [4.876, 5.404]	0.898 [0.891, 0.909]	71 [63.5, 75.8]	88.1 [79.5, 94.6]		
	Dense-only S2KAN	7.366 [7.016, 7.878]	0.793 [0.781, 0.809]	4 [4.0, 5.0]	64.4 [60.9, 71.4]		
Power plant	Full S2KAN	4.186 [4.104, 4.468]	0.937 [0.931, 0.941]	16 [15.0, 17.0]	36.7 [34.3, 39.4]		
	No-dense S2KAN	4.178 [4.114, 4.466]	0.937 [0.931, 0.942]	15 [15.0, 16.0]	36.8 [33.8, 39.0]		
	Dense-only S2KAN	4.622 [4.496, 4.844]	0.924 [0.920, 0.930]	7 [6.0, 8.0]	107.2 [95.1, 122.6]		
Superconductor	Full S2KAN	19.514 [18.853, 20.273]	0.682 [0.637, 0.698]	16 [14.2, 19.0]	61.3 [51.8, 74.8]		
	No-dense S2KAN	20.068 [19.541, 20.639]	0.665 [0.642, 0.672]	20.5 [18.0, 22.0]	41.6 [33.2, 45.6]		
	Dense-only S2KAN	19.952 [18.987, 20.521]	0.658 [0.641, 0.687]	6.5 [5.2, 8.0]	100.8 [82.8, 130.8]		
(b) Symbolic fidelity (%)							
Dataset	Model	$\epsilon = 0.01$			$\epsilon = 0.1$		
		$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
Airfoil	Full S2KAN	22.4 [20.0, 25.7]	50.0 [46.3, 52.2]	64.0 [60.0, 69.2]	33.3 [29.4, 39.8]	84.3 [82.6, 87.5]	91.3 [87.5, 92.0]
	No-dense S2KAN	20.9 [17.6, 26.7]	54.2 [48.9, 59.1]	71.4 [65.2, 73.6]	38.6 [34.8, 41.5]	88.2 [85.1, 91.8]	96.0 [95.7, 100.0]
	Dense-only S2KAN	6.5 [0.0, 9.6]	25.0 [19.1, 27.1]	25.8 [25.0, 30.8]	8.0 [6.2, 13.8]	33.3 [31.2, 41.7]	50.0 [40.4, 55.8]
Concrete	Full S2KAN	77.8 [73.7, 79.5]	85.4 [83.5, 88.6]	86.9 [85.1, 90.4]	82.1 [78.5, 85.6]	90.3 [89.1, 94.2]	94.4 [91.9, 96.1]
	No-dense S2KAN	76.4 [73.0, 81.7]	86.5 [84.7, 91.5]	88.0 [85.1, 92.3]	82.2 [78.5, 85.4]	92.0 [89.9, 95.0]	94.3 [92.7, 95.9]
	Dense-only S2KAN	0.0	0.0	0.0 [0.0, 23.8]	0.0	25.0 [21.2, 50.0]	75.0 [52.5, 78.8]
Power plant	Full S2KAN	35.5 [30.2, 39.6]	64.3 [59.1, 71.4]	69.9 [64.3, 79.6]	50.0 [46.2, 57.1]	92.6 [85.7, 93.9]	92.9 [92.3, 100.0]
	No-dense S2KAN	27.6 [22.9, 40.9]	65.5 [60.4, 74.1]	73.2 [64.9, 80.9]	50.0 [39.1, 56.5]	92.6 [87.5, 98.6]	93.8 [92.3, 100.0]
	Dense-only S2KAN	0.0	0.0	0.0	0.0	29.2 [16.7, 42.1]	63.3 [50.0, 78.8]
Superconductor	Full S2KAN	19.4 [13.4, 27.5]	39.4 [30.1, 49.3]	53.3 [44.2, 60.2]	25.0 [15.7, 33.3]	55.6 [50.0, 65.6]	65.9 [57.1, 73.6]
	No-dense S2KAN	21.5 [15.8, 29.4]	46.4 [35.7, 57.7]	58.6 [47.4, 66.7]	27.6 [19.0, 33.3]	63.2 [55.6, 68.0]	71.4 [63.5, 77.4]
	Dense-only S2KAN	0.0	0.0	0.0	0.0	0.0 [0.0, 13.3]	16.7 [12.9, 20.0]

Full S2KAN has lower median test RMSE than dense-only S2KAN on all four datasets. After averaging over the three seeds within each split, its RMSE is lower in all ten splits for Airfoil, Concrete and Power Plant, and six of ten for Superconductor. Its median k is also smaller on three datasets; on Concrete, dense-only is smaller (64.4 versus 76.2) but less accurate (RMSE 7.366 versus 5.268 MPa). Full S2KAN has higher median SF in all 24 dataset–setting comparisons. Thus sparsifying dense branches alone does not reproduce the accuracy and symbolic fidelity of the full dictionary under this protocol. One dense-only Concrete fit selects an epoch-10 checkpoint during warmup; it remains included, following the common validation rule.

Table 13: Dense-branch ablation on Nguyen benchmarks, using architecture L and $\beta = 1$ for both methods. Entries are medians over ten initialization seeds on the same training/test sample; brackets give the first and third quartiles (omitted for constant values). “No-dense S2KAN” disables both spline and SiLU contributions while retaining the symbolic, Chebyshev and Fourier terms. RMSE is test error; Active counts gate-active edges; k is weighted expected support. SF is the percentage of functionally active edges approximated within relative RMSE tolerance ϵ by at most J gauged terms from $\{x, x^2, \sin x\}$. All fits are included; none has an empty functional-activity set.

Problem	Model	RMSE	Model size		SF (%), $\epsilon = 0.01$			SF (%), $\epsilon = 0.1$		
			Active	k	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
F1	S2KAN	0.0055 [0.0033, 0.0082]	6 [6, 6]	42.5 [38.4, 51.3]	40.0 [21.3, 40.0]	80.0 [76.2, 95.0]	100.0	60.0 [50.0, 60.0]	100.0 [100.0, 100.0]	100.0
	No-dense S2KAN	0.0041 [0.0030, 0.0083]	6	40.3 [39.0, 42.2]	40.0 [40.0, 47.5]	100.0 [100.0, 100.0]	100.0	60.0 [60.0, 60.0]	100.0	100.0
F2	S2KAN	0.0063 [0.0045, 0.0100]	6 [6, 6]	41.4 [40.0, 45.4]	60.0 [52.5, 72.9]	100.0 [80.0, 100.0]	100.0 [100.0, 100.0]	100.0 [76.2, 100.0]	100.0 [100.0, 100.0]	100.0
	No-dense S2KAN	0.0065 [0.0035, 0.0076]	6	41.3 [38.6, 45.3]	55.0 [40.0, 60.0]	100.0	100.0	90.0 [80.0, 100.0]	100.0	100.0
F3	S2KAN	0.0062 [0.0043, 0.0120]	6 [6, 6]	55.9 [53.2, 59.5]	40.0 [28.7, 40.0]	80.0 [75.0, 80.0]	80.0 [80.0, 95.8]	60.0 [42.5, 60.0]	100.0 [85.0, 100.0]	100.0 [100.0, 100.0]
	No-dense S2KAN	0.0089 [0.0041, 0.0098]	6	40.1 [39.2, 42.0]	45.0 [28.7, 50.0]	87.5 [75.0, 100.0]	100.0	60.0 [50.0, 60.0]	100.0	100.0
F4	S2KAN	0.0078 [0.0043, 0.0091]	6 [5.25, 6]	50.5 [46.6, 59.0]	50.0 [50.0, 60.0]	80.0 [76.2, 80.0]	91.7 [80.0, 100.0]	77.5 [75.0, 80.0]	100.0	100.0
	No-dense S2KAN	0.0050 [0.0039, 0.0074]	6	40.4 [37.8, 43.2]	40.0 [40.0, 55.0]	100.0 [100.0, 100.0]	100.0	60.0 [60.0, 80.0]	100.0	100.0
F5	S2KAN	0.0352 [0.0199, 0.0602]	6	58.2 [57.5, 58.7]	75.0 [52.5, 75.0]	100.0	100.0	100.0 [75.0, 100.0]	100.0	100.0
	No-dense S2KAN	0.0627 [0.0322, 0.0944]	6	57.4 [55.5, 58.4]	75.0 [52.5, 75.0]	100.0	100.0	87.5 [75.0, 100.0]	100.0	100.0
F6	S2KAN	0.0030 [0.0021, 0.0041]	6	40.9 [37.1, 44.2]	80.0 [65.0, 95.0]	100.0 [100.0, 100.0]	100.0	100.0 [85.0, 100.0]	100.0	100.0
	No-dense S2KAN	0.0048 [0.0029, 0.0070]	6	44.8 [44.3, 45.7]	63.3 [45.0, 82.5]	100.0	100.0	100.0 [80.8, 100.0]	100.0	100.0
F7	S2KAN	0.0196 [0.0088, 0.0287]	6	56.5 [55.5, 57.4]	55.0 [50.0, 60.0]	100.0 [100.0, 100.0]	100.0	80.0 [63.7, 100.0]	100.0	100.0
	No-dense S2KAN	0.0176 [0.0153, 0.0225]	6	59.5 [57.0, 60.2]	60.0 [52.5, 71.2]	100.0	100.0	100.0 [100.0, 100.0]	100.0	100.0
F8	S2KAN	0.0141 [0.0074, 0.0203]	6	51.8 [48.6, 54.2]	41.7 [20.0, 57.5]	77.5 [63.7, 80.0]	100.0 [100.0, 100.0]	63.3 [52.5, 95.0]	100.0	100.0
	No-dense S2KAN	0.0079 [0.0070, 0.0082]	6	56.8 [54.5, 58.1]	10.0 [0.0, 25.0]	75.0 [75.0, 100.0]	100.0 [100.0, 100.0]	50.0 [31.2, 57.5]	100.0 [81.2, 100.0]	100.0
F9	S2KAN	0.0037 [0.0029, 0.0044]	9	41.9 [39.8, 43.7]	93.8 [70.8, 100.0]	100.0	100.0	100.0 [87.5, 100.0]	100.0	100.0
	No-dense S2KAN	0.0037 [0.0020, 0.0046]	9 [8.25, 9]	38.2 [32.9, 42.4]	85.7 [80.8, 96.4]	100.0	100.0	85.7 [83.9, 100.0]	100.0	100.0
F10	S2KAN	0.0037 [0.0031, 0.0070]	9 [9, 9]	22.0 [20.1, 24.1]	86.6 [55.6, 96.9]	100.0 [100.0, 100.0]	100.0	94.4 [87.5, 100.0]	100.0	100.0
	No-dense S2KAN	0.0041 [0.0028, 0.0074]	9 [8.25, 9]	22.6 [17.8, 25.2]	93.8 [85.7, 100.0]	100.0 [100.0, 100.0]	100.0	100.0 [91.7, 100.0]	100.0	100.0

G.2.2 Nguyen benchmarks

We compare full and no-dense S2KAN, retaining the symbolic, Chebyshev and Fourier terms in both models. Both methods use the L architecture $[n_0, 3, 1]$, $\beta = 1$, the same dictionary, training/test sample and ten initialization seeds, a log-MSE data term, constant learning rate 10^{-3} , batch size 128, and 200 warmup epochs. Their grid-update and stopping rules match; all 200 fits reached 10,000 epochs. Table 13 retains every fit and uses the same symbolic-fidelity procedure as Table 3.

The no-dense control achieves strong predictive performance on the Nguyen benchmarks, demonstrating the approximation capacity of the symbolic and finite-basis dictionary alone. Full S2KAN often selects a representation with no dense contribution: 71 of its 100 fitted models close all dense gates at the final checkpoint, while keeping these components available during training. Full S2KAN also achieves lower median test RMSE on six of ten problems, with the control achieving lower error on the remaining four. Across the ten problems and six correlated fidelity settings, 39 median comparisons tie, 11 favor full S2KAN and 10 favor the control; these descriptive counts indicate no consistent fidelity advantage for either configuration. Across all 60 comparisons, including ties, the median SF difference (full S2KAN minus no-dense S2KAN) is zero percentage points. Together with the library-misspecification experiment, these results support the flexibility of joint symbolic and dense selection: the model can arrive at a restricted representation when the dictionary suffices while retaining access to a dense fallback.

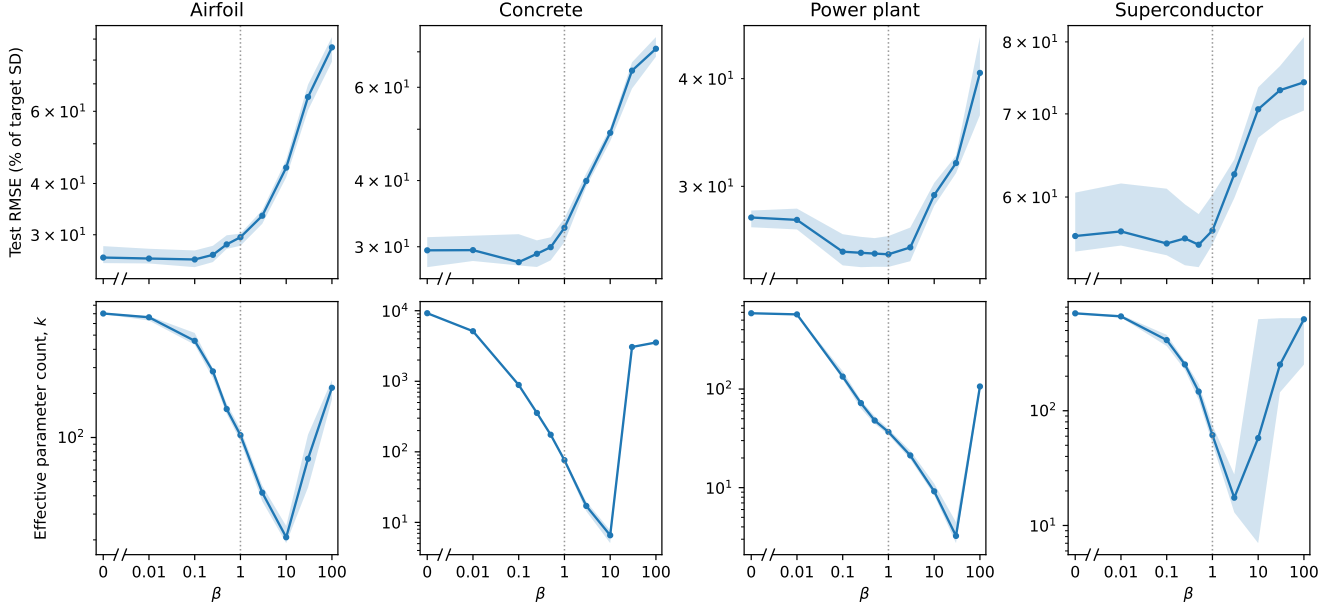


Figure 7: Regularization trades model size against predictive accuracy on real-world datasets. Curves and shading show medians and IQRs over 30 validation-selected fits at each β . RMSE is expressed as a percentage of the test-target standard deviation (SD), calculated separately for each fit before aggregation. The dotted line marks $\beta = 1$.

G.3 Regularization-strength sensitivity

G.3.1 Real-world datasets

With the remaining real-world training settings held fixed, we vary β over $\{0, 0.01, 0.1, 0.25, 0.5, 1, 3, 10, 30, 100\}$ on all ten data splits and three initialization seeds per split (1,200 fits). Each split–seed block uses identical initialization and warmup on the same CPU host. Figure 7 uses the main validation-selection rule. Relative to $\beta = 0$, $\beta = 1$ reduces median k by 85–99%; median test RMSE improves for Power Plant, changes little for Superconductor, and increases for Airfoil and Concrete. Stronger regularization eventually degrades accuracy on every dataset. This sensitivity analysis does not select a new β using test performance.

The increase in selected model size at large β reflects selection of earlier checkpoints, before further sparsification worsens validation error. Selection occurs within the 500-epoch unregularized warmup for 5/30 Airfoil fits at $\beta = 100$ and 9/30, 14/30 and 16/30 Superconductor fits at $\beta = 10, 30, 100$, respectively; no other settings select warmup checkpoints. For Concrete at $\beta = 30$ and 100, every selected checkpoint is between epochs 580 and 610. At the common 5,000-epoch endpoint (Fig. 8), median k instead decreases monotonically with β , approaching zero at $\beta = 100$ for every dataset, with poor predictive accuracy.

G.3.2 Nguyen benchmarks

Figure 9 reports a sweep over $\beta \in \{0, 0.01, 0.1, 1, 3, 10, 30, 100\}$ for all ten Nguyen problems and architectures S, L and LM, retaining all 2,400 fits through $\beta = 100$. Unlike the real-world sweep, each problem uses a fixed training/test sample with ten initialization seeds per configuration. Except for the small F10 models at $\beta = 1$, gate-based early stopping occurs only at $\beta \geq 3$ and becomes widespread at $\beta = 30$ and 100 (bottom row).

G.4 Behavior under library misspecification

We examine how S2KAN approximates targets whose generating primitives are absent from the symbolic library. The Nguyen benchmarks illustrate approximation using the remaining dictionary, while a controlled sinc ablation examines recovery through dense components.

The Nguyen benchmark (Table 3) provides several examples of misspecification: F7 requires $\log(\cdot)$ and F8 requires $\sqrt{\cdot}$, neither of which appear in the symbolic library (see Methods (App. A)). Despite this, S2KAN achieved median $R^2 > 0.99$ on both problems for every architecture. Accuracy was nonetheless lower than for baseline KAN: on the two larger architectures,

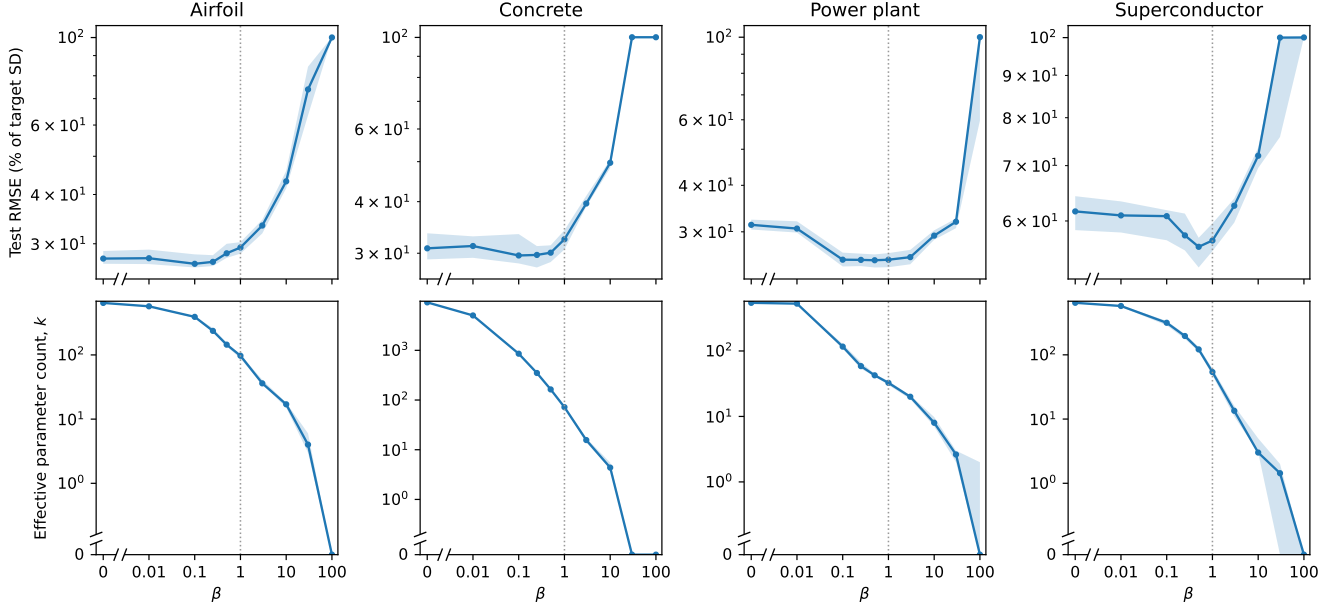


Figure 8: The same real-world regularization sweep shown in Fig. 7 but for fits evaluated at the end of training. Unlike the validation-selected fits, median size decreases throughout the sweep; excessive regularization produces small but inaccurate models.

the median S2KAN RMSE exceeds 1% of the target standard deviation for both problems, whereas KAN stays within it (Table 3). In fact, S2KAN retains high symbolic fidelity on both problems: on the larger architectures, $J = 2$, $\epsilon = 0.01$ fidelity is 78–100% for S2KAN versus 6–17% for KAN, indicating that the available library terms serve as effective approximations even without the ground-truth functions. The no-dense control also approximates F7 and F8, achieving lower median RMSE than full S2KAN for architecture L (Table 13). Thus, these examples demonstrate approximation using available symbolic and finite-basis terms despite missing generating primitives, rather than isolating a benefit from dense fallback.

Given the tradeoff observed in these Nguyen problems, we next test whether the dense branch can recover a symbolic component deliberately omitted from the training dictionary. We therefore repeated the sinc experiment (Fig. 1) with the reciprocal removed from the symbolic library, $\mathcal{S} = \{x\}$, for five seeds; all other settings, including $\beta = 1$, match the figure (see Methods (App. A)). Without access to this term, S2KAN cannot symbolically represent the factor $1/x$ in $\text{sinc}(x) = \sin(x) \cdot (1/x)$. All five ablated models still fit the data (test MSE $< 10^{-5}$; median 2.1×10^{-6} , versus 2.9×10^{-8} for baseline KAN). In three of the five, one activation is a pure $\sin(x)$ and the other falls back to a spline (in one seed with a small additional Fourier term). That activation is numerically the missing primitive: a single affinely gauged $1/x$ fits it with $R^2 > 0.9999$, whereas the best alternative primitive we tested, $1/x^2$, reaches only $R^2 \approx 0.998$. The two remaining ablated S2KAN fits retain more complex decompositions, with splines on both activations. The spline-only baseline KAN, trained with the same data, architecture, and training schedule, never learns such a decomposition: its five fits are accurate (test MSE $\leq 1.2 \times 10^{-6}$), but none of their ten spline activations is well approximated by a single gauged $\sin(x)$ or $1/x$ ($R^2 \leq 0.89$ in every case). The same three seeds keep this sine-times-reciprocal structure at $\beta = 2, 10$ and 30 , but at $\beta = 30$ the other two close both spline branches and fail ($R^2 \approx 0.31$). In this controlled product architecture, with sine terms supplied by the Fourier basis, the dense branch recovers the omitted reciprocal over the tested interval $[1, 15]$ in three of five seeds. This demonstrates recovery of a missing symbolic component when part of the compositional structure is supplied; recovery is not guaranteed, and strong regularization can suppress the fallback.

G.5 Sensitivity of symbolic fidelity to an amplitude constraint

All reported SF values use unconstrained symbolic approximation. This sensitivity analysis covers the main results, the Nguyen no-dense control, and the real-world dense-only control. Here we test whether the conclusions depend on allowing large, mutually cancelling fitted terms. For an edge response y_e approximated by $d + \sum_{m=1}^J a_m p_m(b_m x + c_m)$, the constrained extractor additionally requires

$$\text{sd}[a_m p_m(b_m x + c_m)] \leq C \text{sd}(y_e), \quad C = 3, \quad (23)$$

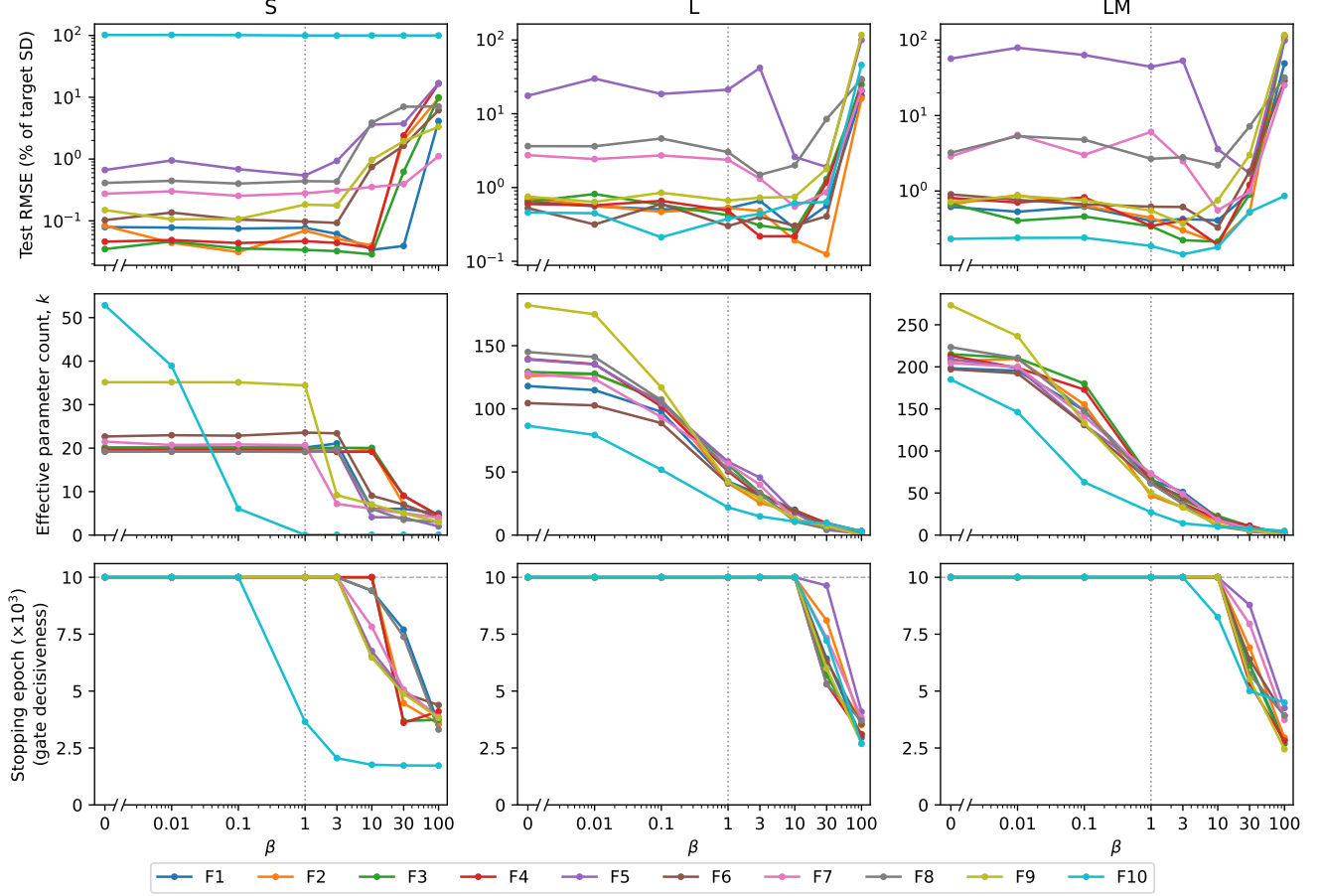


Figure 9: Accuracy, model size, and stopping epoch versus regularization strength on Nguyen benchmarks. Each curve shows one problem’s median over ten initialization seeds; IQRs are omitted for readability. Test RMSE is expressed as a percentage of the test-target standard deviation, calculated per fit before aggregation; k follows Table 3. The bottom row shows median stopping epochs under the gate-decisiveness rule, including runs reaching the 10,000-epoch cap (dashed line). The dotted line marks $\beta = 1$. Positive β values are log spaced, with zero shown separately; the RMSE axis is logarithmic and the other axes linear.

for each fitted term, with standard deviations evaluated on the same empirical input sample used for SF. The additive offset is unrestricted. This is a post-hoc extraction control: model weights, predictions, functional-activity criteria, primitive dictionaries, input samples, and the six combinations of $J \in \{1, 2, 3\}$ and $\epsilon \in \{0.01, 0.1\}$ are held fixed. The one-term search is unchanged. Multi-term fits use a penalty for violating Eq. (23), followed by further optimization and rejection of candidates exceeding the bound by more than a relative tolerance of 10^{-4} . Since the constraint also changes the nonlinear search path, we report the actual score from each search; occasional increases in constrained SF are retained.

For the main results and the Nguyen no-dense control, the inventory contains 1,020 distinct models contributing numerical SF results: 600 Nguyen models, 240 validation-selected real-world models, 80 dynamical-system models, and 100 additional no-dense Nguyen models. The full S2KAN models reused in the no-dense comparison are counted only once. We also evaluate the two illustrative sinc checkpoints in Fig. 1. Each pair uses the same archived checkpoint and extractor settings. Recent score records identify checkpoints by hash; historical Nguyen S2KAN records are matched through their checkpoint directories and verified against their functional-activity sets, edge variances, and one-term fits. All six SF scores are independently reconstructed from the saved edge fits.

All 1,022 pairs passed the matching checks. Across the six settings, 5,804 of 6,132 scores (94.7%) are identical; restricting the comparison to $J = 2, 3$, where the constraint can affect the search, 3,760 of 4,088 scores (92.0%) are identical. Figure 10 shows all pairs, and Table 14 summarizes the changes. Both illustrated sinc models have identical scores at all six settings. Models with empty functional-activity sets retain the same SF convention in both versions.

For the additional 120 real-world dense-only models, 712 of 720 scores (98.9%) are unchanged. The largest change in a dataset median is 3.33 percentage points. Full S2KAN retains higher median SF than dense-only S2KAN in all 24 comparisons

Table 14: Amplitude-constraint sensitivity of SF. Changes are constrained minus unconstrained SF, in percentage points. “Equal scores” counts identical values across all six settings for the paired models; settings are correlated. “Largest median shift” compares medians within the same dataset/problem, architecture, method and regularization group at each setting, matching the main tables’ aggregation. All expected fits are included in each group. The sinc illustration contains one checkpoint per method.

Cohort	Paired / total	Equal scores	Median $ \Delta $	Largest $ \Delta $	Largest median shift
Dynamical systems	80/80	440/480	0.00	5.88	2.27
Nguyen	600/600	3472/3600	0.00	50.00	28.57
No-dense control	100/100	547/600	0.00	50.00	27.50
Real world	240/240	1333/1440	0.00	33.33	17.88
Sinc illustration	2/2	12/12	0.00	0.00	0.00

Table 15: Method comparisons using median SF at each of the six settings. Entries are wins/ties/losses for S2KAN versus baseline KAN, except the final row, which compares full S2KAN with its no-dense control. Nguyen comparisons span ten problems and three architectures; real-world comparisons span four datasets; dynamical-system comparisons span six S2KAN configurations, each compared with its system’s baseline. Each median uses all fits in the corresponding paper table.

Comparison	Unconstrained	$C = 3$
Nguyen	114/66/0	113/67/0
Real world	24/0/0	24/0/0
Dynamical systems	28/6/2	28/6/2
Full vs no dense	11/39/10	12/36/12

with or without the constraint.

The main comparative findings persist (Table 15). For Nguyen, unconstrained SF favors S2KAN in 114 of 180 median comparisons, with 66 ties; $C = 3$ gives 113 wins and 67 ties, with no losses in either case. S2KAN retains higher median SF than baseline KAN in all 24 real-world comparisons. The dynamical-system comparison gives 28 wins, six ties and two losses with or without the amplitude constraint. The full-versus-no-dense comparison remains mixed: 11 wins, 39 ties and ten losses without the bound, versus 12 wins, 36 ties and 12 losses with it. These are descriptive counts across correlated settings, not independent statistical tests.

The constraint does affect some numerical SF estimates appreciably. The largest change in a Nguyen table median is 28.6 percentage points (F4, architecture LM, $\epsilon = 0.01$, $J = 2$); the largest real-world change is 17.9 points for power-plant S2KAN at the same setting, from 64.3% to 46.4%. The largest changes in configuration medians are 2.27 points for dynamical systems and 27.5 points for the no-dense control. Thus the broad comparisons are robust to this amplitude constraint, while individual fidelity values need not be invariant. Unconstrained SF measures approximation by a short symbolic expression; it does not itself guarantee well-conditioned coefficients.

Statements and Declarations

Funding

No funding was received for this work.

Competing interests

The authors declare no competing interests.

Data availability

The real-world datasets used in this study are publicly available from the UCI Machine Learning Repository (<https://archive.ics.uci.edu/>). Dataset identifiers and descriptions are provided in Methods (App. A).

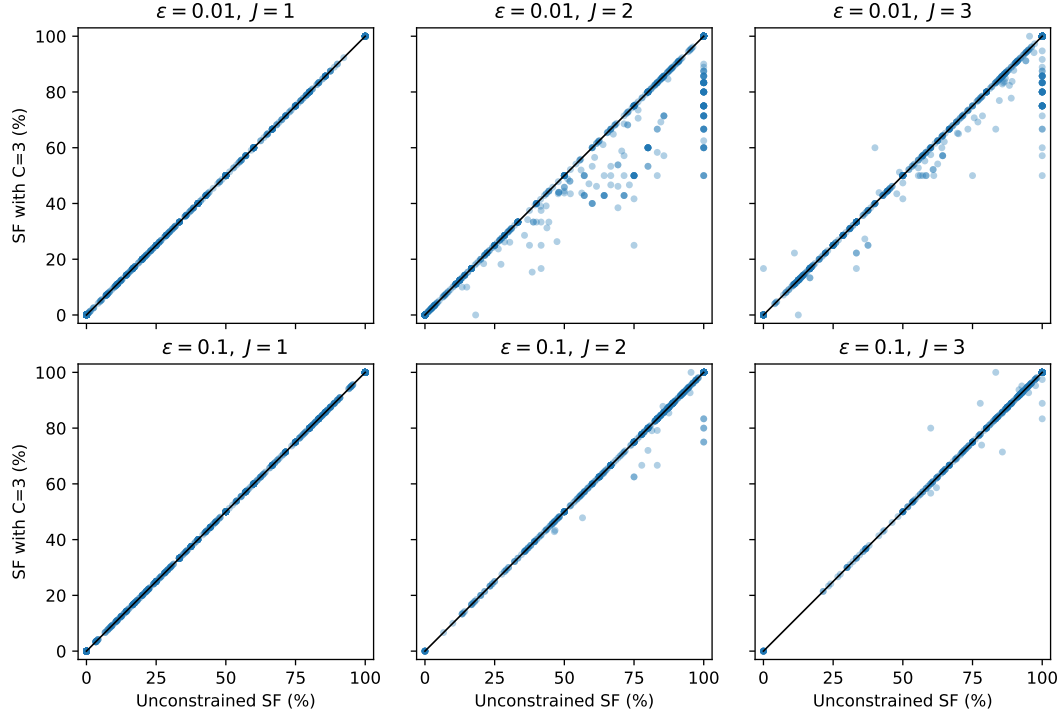


Figure 10: Unconstrained versus amplitude-constrained ($C = 3$) symbolic fidelity for the 1,022 models in the main-results and Nguyen no-dense inventory, including both sinc illustrations. Each panel contains one point per model; coincident points overlap. The line is equality. The $J = 1$ search is unchanged by construction. Occasional points above the line reflect differences in the nonlinear search path rather than an improvement in the globally optimal constrained approximation.

Code availability

Code supporting this study will be made publicly available upon acceptance. The repository URL and version identifier will be included in the final manuscript.

Author contributions

J. Bagrow led the research and manuscript preparation, with contributions from J. Bongard.

References

- Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6, 52138–52160. doi: 10.1109/ACCESS.2018.2870052
- Arnold, V. I. (2009). On functions of three variables. In A. B. Givental et al. (Eds.), *Collected works: Representations of functions, celestial mechanics and KAM theory, 1957–1965* (Vol. 1, pp. 5–8). Springer. doi: 10.1007/978-3-642-01742-1_2
- Bagrow, J., & Bongard, J. (2025, August). Multi-exit Kolmogorov–Arnold networks: enhancing accuracy and parsimony. *Machine Learning: Science and Technology*, 6(3), 035037.
- Bozorgasl, Z., & Chen, H. (2024). Wav-KAN: Wavelet Kolmogorov–Arnold networks. *arXiv preprint arXiv:2405.12832*.
- Brooks, T. F., Pope, D. S., & Marcolini, M. A. (1989, July). *Airfoil self-noise and prediction* (Tech. Rep. No. NASA-RP-1218). NASA Langley Research Center. Retrieved from <https://ntrs.nasa.gov/citations/19890016302> (NASA Reference Publication 1218)

7, 8, 22

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15), 3932–3937.

1, 9, 14, 16

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), 045002.

1

Center for Basic Research on Materials. (2024). *MDR SuperCon datasheet ver.240322*. National Institute for Materials Science.

7, 17

Cranmer, M. (2023). Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*.

1, 14, 16

Cranmer, M., Sanchez Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., & Ho, S. (2020). Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 17429–17442).

1

GistNoesis. (2024). *FourierKAN*. <https://github.com/GistNoesis/FourierKAN>. (Accessed: 2025-07-07)

3

Grünwald, P. D. (2007). *The minimum description length principle*. Cambridge, MA: MIT Press.

6

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5), 93:1–93:42.

1

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154, 346–354.

7, 17

Hammel, S. M., Jones, C. K. R. T., & Moloney, J. V. (1985). Global dynamical behavior of the optical field in a ring cavity. *Journal of the Optical Society of America B*, 2(4), 552–564. doi: 10.1364/JOSAB.2.000552

9

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: The lasso and generalizations* (Vol. 143). CRC Press.

1

Hempel, C. G. (1965). *Aspects of scientific explanation and other essays in the philosophy of science*. New York: Free Press.

1

Ikeda, K. (1979). Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications*, 30(2), 257–261.

9

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.

1

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

16

Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSSR*, 114(5), 953–956.

2

Kolmogorov, A. N. (1961). On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. In *Twelve papers on algebra and real functions* (Vol. 17, pp. 369–373). American Mathematical Society. doi: 10.1090/trans2/017/12

2

La Cava, W., Orzechowski, P., Burlacu, B., de França, F. O., Virgolin, M., Jin, Y., Kommenda, M., & Moore, J. H. (2021). Contemporary symbolic regression methods and their relative performance. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*.

1

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

- 3
Lipton, Z. C. (2018, September). The mythos of model interpretability. *Communications of the ACM*, 61(10), 36–43.
- 1
Liu, Z., Tegmark, M., Ma, P., Matusik, W., & Wang, Y. (2025). Kolmogorov–Arnold Networks meet science. *Physical Review X*, 15, 041051.
- 1, 3, 4, 14
Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2025). KAN: Kolmogorov–Arnold networks. In *The thirteenth international conference on learning representations*.
- 1, 2, 3, 4, 7, 16
Louizos, C., Welling, M., & Kingma, D. P. (2018). Learning sparse neural networks through l_0 regularization. In *International conference on learning representations*.
- 1, 5
Martius, G., & Lampert, C. H. (2016). Extrapolation and learning equations. *arXiv preprint arXiv:1610.02995*.
- 14, 29
McCann, K., & Yodzis, P. (1994). Nonlinear dynamics and population disappearances. *The American Naturalist*, 144(5), 873–879.
- 10
Neville, A. M. (2011). *Properties of concrete* (5th ed.). Pearson.
- 7
Nguyen, Q. U., Nguyen, X. H., O’Neill, M., McKay, R. I., & Galván-López, E. (2011). Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2), 91–119.
- 11, 17
Panahi, S., Moradi, M., Bollt, E. M., & Lai, Y.-C. (2025, April). Data-driven model discovery with Kolmogorov–Arnold networks. *Physical Review Research*, 7, 023037.
- 1, 9, 10, 17
Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., . . . Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
- 16
Ponce Arrieta, F. R., & Silva Lora, E. E. (2005). Influence of ambient temperature on combined-cycle power-plant performance. *Applied Energy*, 80(3), 261–272. doi: 10.1016/j.apenergy.2004.04.007
- 9
Poole, A., McArthur, S., & Kumar, S. (2026). Entropy-guided projective Kolmogorov–Arnold networks for compact and robust edge-function learning. *Frontiers in Artificial Intelligence*. (Accepted for publication; final formatted article pending) doi: 10.3389/frai.2026.1880092
- 15, 16
Reinhardt, E., Ramakrishnan, D., & Gleyzer, S. (2025). SineKAN: Kolmogorov–Arnold networks using sinusoidal activation functions. *Frontiers in Artificial Intelligence*, 7, 1462952. doi: 10.3389/frai.2024.1462952
- 3
Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215.
- 1
Sahoo, S., Lampert, C., & Martius, G. (2018, 10–15 Jul). Learning equations for extrapolation and control. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 4442–4450). PMLR.
- 14, 16, 29
Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, 324(5923), 81–85.
- 1
Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 461–464.
- 1, 6
Shmueli, G. (2010). To explain or to predict? *Statistical Science*, 25(3), 289–310. doi: 10.1214/10-STS330
- 1

- Sidharth SS, Gokul R, Anas K P, & Keerthana AR. (2024). Chebyshev polynomial-based Kolmogorov–Arnold networks: An efficient architecture for nonlinear function approximation. *arXiv preprint arXiv:2405.07200*.
3
- Skokos, C. (2010). The Lyapunov characteristic exponents and their computation. In J. J. Souchay & R. Dvorak (Eds.), *Dynamics of small solar system bodies and exoplanets* (Vol. 790, pp. 63–135). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-04458-8_2
17
- Tüfekci, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 60, 126–140.
7
- Udrescu, S.-M., Tan, A., Feng, J., Neto, O., Wu, T., & Tegmark, M. (2020). AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 4860–4871).
16
- Udrescu, S.-M., & Tegmark, M. (2020). AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16), eaay2631.
1, 14, 16
- Werner, M., Junginger, A., Hennig, P., & Martius, G. (2021). Informed equation learning. *arXiv preprint arXiv:2105.06331*.
14, 16
- Woodward, J. (2004). *Making things happen: A theory of causal explanation*. Oxford University Press.
1
- Xu, Y., Liu, X., Cao, X., Huang, C., Liu, E., Qian, S., Liu, X., Wu, Y., Dong, F., Qiu, C.-W., Qiu, J., Hua, K., Su, W., Wu, J., Xu, H., Han, Y., Fu, C., Yin, Z., Liu, M., . . . Zhang, J. (2021). Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2(4), 100179. doi: 10.1016/j.xinn.2021.100179
1
- Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12), 1797–1808.
7, 17
- Yeh, I.-C. (2006). Analysis of strength of concrete using design of experiments and neural networks. *Journal of Materials in Civil Engineering*, 18(4), 597–604.
7, 17
- Ziyin, L., & Wang, Z. (2023). spread: Solving L1 penalty with SGD. In *Proceedings of the 40th international conference on machine learning* (Vol. 202, pp. 43407–43422). PMLR. Retrieved from <https://proceedings.mlr.press/v202/ziyin23a.html>
4