HIGHER-ORDER KOLMOGOROV-ARNOLD NETWORKS FOR DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

The ability to derive the governing equations of a dynamical system from data is essential for prediction and control across many scientific fields. Two challenges, however, are the expansion of candidate terms when reducing higher-order equations to first-order form and noise from estimating derivatives. We propose Higher-Order-KANDy, which addresses both problems by using delay and derivative ("jet") embeddings of the original data with derivative-ordered Kolmogorov-Arnold Networks for Dynamics (KANDy) layers stacked into a machine learning architecture. Compared with SINDy and WSINDy on noisy data, Higher-Order-KANDy learns governing equations without a candidate library (e.g., recovering the full-period $-\sin(u)$ with nothing named in advance), natively for higher-order systems, and holds the correct term structure as noise grows, selecting the exact active set on every seed at noise levels up to 7.5% where an oracle weak-form regression admits a spurious term—though the oracle remains the more accurate coefficient estimator. We leverage the relationship between delay and differential embeddings, along with the Kolmogorov–Arnold representation theorem, to construct a delay-to-jet map and learn it as a Kolmogorov–Arnold network to preserve derivative structure and provide meaningful terms that appear in the governing equation. In our synthetic benchmarks, this construction recovers equations in the presence of coarse sampling and noise.

1 INTRODUCTION

Data-driven discovery of governing equations has reshaped forecasting and control grounded in physical law (Brunton et al., 2016; Rudy et al., 2017; Lai, 2021; Schmidt & Lipson, 2009; Bongard & Lipson, 2007; Cranmer, 2023). Sparse identification of nonlinear dynamics (SINDy) recovers interpretable equations by selecting a parsimonious set of terms from a user-specified candidate library (Brunton et al., 2016), and has been extended to PDEs (Rudy et al., 2017; Schaeffer, 2017; Li & Carvalho, 2024) and to discovery from partial measurements (Bakarji et al., 2023; Brunton et al., 2017; Champion et al., 2019). Two limitations persist: noise amplified when estimating derivatives, especially second- and third-order ones, and the dictionary growth incurred when higher-order ODEs and PDEs are rewritten as first-order systems for sparse regression. Weak Form SINDy (WSINDy) addresses the first by replacing pointwise differentiation with linear transformations and variance reduction (Messenger & Bortz, 2021a), with extensions to PDEs, interacting particle systems, and streaming data (Messenger & Bortz, 2021b; 2022; Messenger et al., 2022). The remaining bottleneck is the upfront library choice: too small a library cannot express the equations, while too large a one degrades conditioning and interpretability and presupposes knowledge of the governing structure—a difficulty compounded combinatorially by the extra variables introduced in first-order reduction.

Separately, Kolmogorov–Arnold Networks (KANs) (Liu et al., 2024; 2025), inspired by the Kolmogorov–Arnold representation theorem (Kolmogorov, 1957; Arnold, 2009), place learnable univariate functions on network edges instead of fixed scalar activations, so each edge is a one-dimensional curve that can be inspected, plotted, and symbolically fit. Their theoretical grounding remains debated (Girosi & Poggio, 1989; Lin & Unbehauen, 1993; Köppen, 2002; Sprecher & Draghici, 2002; Montanelli & Yang, 2020; He, 2023; Lai & Shen, 2021), yet KANs perform strongly across scientific domains (Panczyk et al., 2025; Cherednichenko & Poptsova, 2025; Li et al., 2025; Knottenbelt et al., 2025), with a growing set of architectural refinements (Bagrow & Bongard, 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

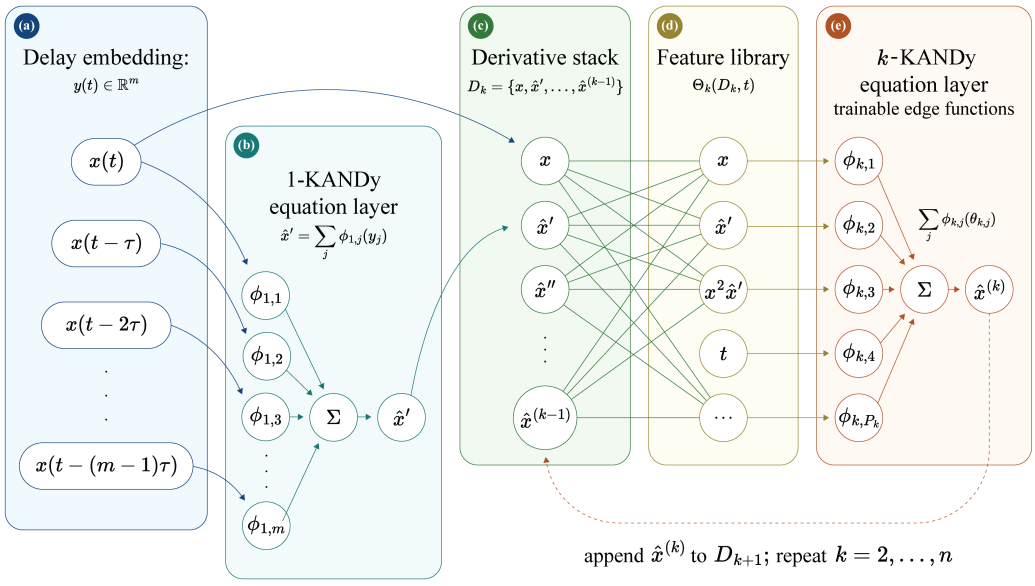

Figure 1: Overview of the Higher-Order-KANDy framework. **(a)** A delay embedding $y(t) = (x(t), x(t-\tau), \ldots, x(t-(m-1)\tau)) \in \mathbb{R}^m$ (m delays, step τ) is passed through **(b)** a 1-KANDy lifting layer that estimates the latent velocity $\hat{x}'$. **(c)** Repeated differentiation builds the derivative stack $\mathcal{D}_k = \{x, \hat{x}', \ldots, \hat{x}^{(k-1)}\}$, from which **(d)** the feature set $\Theta_k(\mathcal{D}_k, t)$ is assembled. **(e)** A k-KANDy equation layer learns univariate edge functions $\varphi_{k,j}$ whose sum predicts $\hat{x}^{(k)}$, so the edges read out directly as the additive terms of the governing equation.

Koenig et al., 2025; Shiraishi et al., 2025); in dynamical systems, KAN-based ODEs learn dynamics and hidden physics (Koenig et al., 2024).

The Kolmogorov–Arnold Network for Dynamics (KANDy) (Slote et al., 2026b) collapses the KAN to a single layer whose edges become the additive terms of the discovered equation (Slote et al., 2026b;a). Like most equation-discovery methods, however, KANDy targets first-order ODEs, predicting a single time derivative from the current state; many physical systems are better described at higher order, and converting them to first-order form expands the dictionary and the room for error.

Related work. Scientific machine learning blends mechanistic structure with data-driven modeling—neural ODE surrogates, physics-informed and operator-learning methods, structure-preserving models, and Koopman embeddings (Karniadakis et al., 2021; Chen et al., 2018; Norcliffe et al., 2020; Raissi et al., 2019; Lu et al., 2021; Greydanus et al., 2019; Williams et al., 2015)—but generally learns black-box vector fields. Symbolic regression and sparse equation discovery instead seek explicit laws by selecting or searching over candidate expressions (Brunton et al., 2016; Crutchfield & McNamara, 1987; Schmidt & Lipson, 2009; Udrescu & Tegmark, 2020; Cranmer, 2023). Complementary methods target derivative noise, higher-order equations, or partial observations through weak-form regression, differentiable symbolic networks, state inference, or surrogate differentiation (Kacprzyk et al., 2023; Martius & Lampert, 2016; Sahoo et al., 2018; Course & Nair, 2023; Kaheman et al., 2022; Hirsh et al., 2022; Messenger & Bortz, 2021a; Berg & Nyström, 2019). Within the KAN line, Panahi et al. (2025) fit KANs as interpretable surrogates for non-sparse dynamics, without recovering closed-form equations; KANDy (Slote et al., 2026b) then collapsed the network to a single layer whose univariate edges are the additive terms of the discovered equation; we extend the construction here to higher order. SINDy-KANs (Howard et al., 2026) (sparse regression per activation) and Symbolic–KAN (Faroughi et al., 2026) (discrete symbolic structure on the edges) pair KANs with discovery at first order; Note S6 runs both on the sine-Gordon benchmark. Higher-Order-KANDy differs in being end-to-end differentiable, with univariate edges that directly represent additive equation terms while stacking one layer per derivative order.

Our contribution. We introduce *Higher-Order-KANDy*, an architecture defined natively for higher-order ODEs and PDEs. We treat KANDy as a single layer of a deeper network whose n-th layer corresponds to the n-th derivative $x^{(n)}$ (n-KANDy), taking as input either a delay embedding (Takens, 1981; Sauer et al., 1991) or a *jet embedding*—the observable's derivatives up to the needed order, read off a smooth fit to the data (Section 2.2)—so unobserved velocity and higher derivatives are recovered from scalar data. The result is an interpretable chain of single-derivative KAN modules whose edges read out as the additive terms of an nth-order governing equation. We also characterize the map from a delay window to the derivatives it encodes: it is the transition map between the delay and differential embeddings of the attractor. We demonstrate the architecture on eleven systems and probes spanning second- and third-order ODEs, nonlinear PDEs, and a hysteretic oscillator with a hidden non-derivative state; Supplementary Table S1 lists them with what each establishes.

2 METHODS

We seek to discover an interpretable ordinary differential equation (ODE) directly from a scalar time series. Given samples $x(t_0), \ldots, x(t_{T-1})$ of a single observable, sampled at uniform spacing Δt, we posit that the observable obeys an autonomous n-th order equation

$$x^{(n)}(t) \;=\; F\big(x,\, x',\, \ldots,\, x^{(n-1)}\big), \tag{1}$$

and our goal is to recover the right-hand side F in a form whose individual additive terms can be read off and interpreted physically.

The central object is Higher-Order-KANDy (Kolmogorov–Arnold Networks for Dynamics), a chain of single-derivative modules, in which each KANDy layer (n-KANDy) learns one step of the derivative hierarchy, and whose learnable univariate edges map directly onto the additive terms in (1).

2.1 HIGHER-ORDER-KANDY ARCHITECTURE

Higher-Order-KANDy chains n KANDy layers (Figure 1), one per derivative order, so that the k-th layer predicts $x^{(k)}$ from the observable and the previously predicted lower derivatives.

Layer 1: lifting. The 1-KANDy layer is a black-box lifting map from the delay vector to the velocity,

$$\hat{x}'(t) \;=\; \mathcal{L}\big(\mathbf{y}(t)\big), \tag{2}$$

implemented as a KAN layer with m inputs and one output. By Section 2.2, x' is a function of the delay coordinates, so $\mathcal{L}$ recovers the unobserved velocity (in effect a learned, data-adapted finite-difference operator). For fully observed systems, where a directly measured velocity is available, this layer can be bypassed and the measured x' supplied instead; this avoids contaminating a near-marginal equation with the lift's approximation error.

Layers $2, \ldots, n$: equation layers. Each higher module consumes a feature library of candidate nonlinear terms built from the running derivative stack and predicts the next derivative. For equation order k, let $\mathcal{D}_k = \{x,\, \hat{x}',\, \ldots,\, \hat{x}^{(k-1)}\}$ be the available derivatives. A library $\Theta_k = \{\theta_{k,1}, \ldots, \theta_{k,P_k}\}$ is a list of named monomials up to order k in these quantities; each term is a product of factors $x^{(r)}$ (the r-th derivative) raised to integer powers (e.g., x, x', x^2, $x\,x'$, x^2x'), and may include nonlinear and transcendental functions. The equation layer (n-KANDy) learns the approximate symbolic form of the learned derivatives flowing through the network,

$$\hat{x}^{(k)}(t) \;=\; \sum_{j=1}^{P_k} \varphi_{k,j}\big(\theta_{k,j}\big), \tag{3}$$

each edge $\varphi_{k,j}$ being a learnable elementary function. In a canonical second-order case ($n = 2$) one may take the Duffing–van der Pol library $\Theta_2 = \{x,\, x',\, x^2x'\}$, giving the discovered equation

$$\hat{x}'' \;=\; \varphi_1(x) \;+\; \varphi_2(x') \;+\; \varphi_3(x^2x'), \tag{4}$$

in which φ_1 is the restoring force, φ_2 the (linear) damping, and φ_3 the cubic damping / nonlinear cross-coupling; the forced-Duffing experiment instead uses edges on x, x' and t, with the forcing

carried by the time edge. When the library is left empty, the equation layer reduces to free-form edges on the raw derivative coordinates $\mathcal{D}_k$, and a symbolic term such as x^2 is read off the shape of the corresponding edge rather than supplied as a column; this free-form additive discovery mode is what the canonical ODE and sine-Gordon experiments use.

Equation (3) is a sum of univariate functions, so a single-equation layer represents exactly those right-hand sides that are additively separable in their input coordinates. This is the condition under which the layer is free-form additively discoverable: when the governing law is a sum of terms each depending on one coordinate—as in $u_{tt} = u_{xx} - \sin(u)$, where the wave and restoring terms separate—each term is one edge, and no dictionary is needed. A term that couples coordinates multiplicatively, such as the transport $u\,u_x$ of Korteweg–de Vries or the cross terms $|\dot{x}|z^3$ of the Bouc–Wen memory law, is not of this form: it must be supplied as a composite feature θ_j, which is what a candidate library does.

2.2 Delay and differential embeddings: observability and noise

The p-jet of the observable at time t is simply the list of its derivatives up to order p, $(x, \dot{x}, \ldots, x^{(p)})$ (for a field, all partial derivatives up to order p); it is what (1) needs on its right-hand side and what the equation layer consumes. The map that sends a state of the system to this list is the differential embedding; its counterpart built from time-shifted copies of x is the delay embedding (Packard et al., 1980; Takens, 1981; Gibson et al., 1992). The two front ends of Higher-Order-KANDy differ only in which of these coordinates they build from data.

A scalar measurement does not expose the higher derivatives required by (1); the velocity x' in particular is unobserved. We resolve this by Takens' delay embedding (Takens, 1981), forming the m-dimensional delay vector

$$\mathbf{y}(t) \;=\; \big[\, x(t),\, x(t-\tau),\, x(t-2\tau),\, \ldots,\, x(t-(m-1)\tau) \,\big] \in \mathbb{R}^m, \tag{5}$$

with embedding dimension m and lag τ in sample units. For a generic observable of a smooth dynamical system, the delay map embeds the underlying attractor, so $\mathbf{y}(t)$ furnishes coordinates in which the time derivatives of x are functions of the delay coordinates—this is what makes the lifting step (Section 2.1) well-posed for scalar data.

Partial observation is not the only barrier. The lifting layer reconstructs the velocity but is trained against pointwise finite-difference targets, and the noise in those targets grows with derivative order. When that noise dominates, we stop estimating derivatives one at a time. Instead, we fit a single smooth surrogate to the data—an analytic spline for fields, a learned map for scalar series—and read the whole p-jet off the surrogate by differentiating it rather than the data. We call this front end the *jet embedding*: a differential embedding estimated from a smooth fit, whose output is the same set of derivative coordinates the delay lift produces, obtained without pointwise differencing (Whalen et al., 2015).

Analytic jet for fields (PDEs). For a field $u(x,t)$, the estimating spline must be two-dimensional, but every KAN edge is a one-dimensional spline: a single additive layer represents only sums $\varphi_1(x) + \varphi_2(t)$, never a general surface (Slote et al., 2026b); a learned per-line operator is also ill-posed, since different spatial locations have different amplitudes and shapes. We therefore use a single analytic surrogate: one tensor-product smoothing spline $S(x,t)$ of degree $(k_x, k_t) = (5,5)$ (Dierckx, 1981; 1993), fit to the noisy field with smoothing strength $\lambda = N\,\hat{\sigma}^2$ (N the number of samples, $\hat{\sigma}^2$ from second differences along t). Every jet entry is then an exact partial derivative of that one fit,

$$u \;=\; S, \qquad u_{xx} \;=\; \partial_x^2 S, \qquad u_{tt} \;=\; \partial_t^2 S, \tag{6}$$

so a single smoothing-plus-differentiation step supplies the complete jet, including spatial derivatives, with noise suppressed by the smoothing fit rather than by differencing neighboring samples. For a scalar field $u(x,t)$ the 2-jet at a grid point is $\big(u, u_x, u_t, u_{xx}, u_{xt}, u_{tt}\big)$. Computing it from the spline rather than learning it hands the equation layer a set of scalar coordinates at each sample point (u, u_{xx}, u_{tt}), and a scalar argument is exactly what a one-dimensional KAN edge requires, so the 2-KANDy layer consumes them natively, resolving the dimensional mismatch. For the undamped sine-Gordon collision, this yields $u_{tt} = c^2\,u_{xx} + \varphi(u)$, with the restoring force φ a single free-form edge fit by the linear read-out.

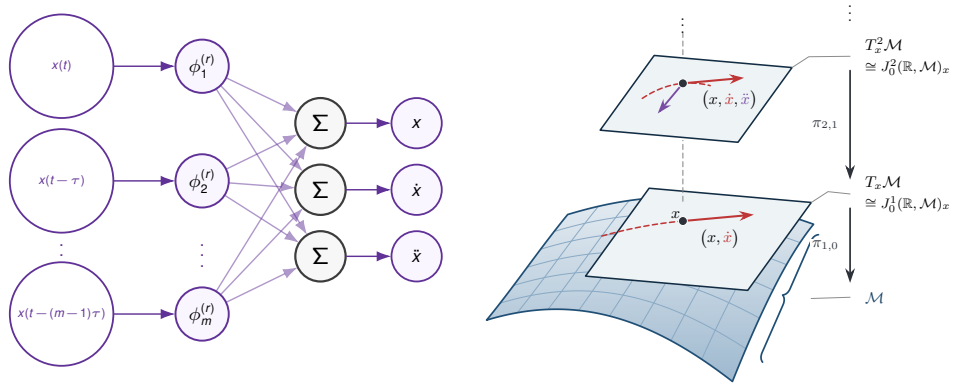

Figure 2: Learning mechanism for the KAN-based jet embedding. Left: the delay window $\mathbf{y}(t)$ flows through learned univariate KAN splines to the jet $(\hat{x}, \hat{x}', \ldots, \hat{x}^{(p)})$, so every derivative is read from one smooth map of the whole window rather than from pointwise differences. Right: the same map viewed as a change of chart on the attractor—from the delay embedding, which is what a scalar sensor measures, to the differential embedding, in which the governing equation takes the form of Eq. (1).

Kolmogorov–Arnold jet embeddings (scalar series). Two classical results make a learnable differential embedding principled (Fig. 2). For $m \geq 2D + 1$ (D the attractor dimension) the delay window determines the state, so the window-to-jet map exists and is continuous (Takens, 1981; Sauer et al., 1991), and the Kolmogorov–Arnold representation theorem (Kolmogorov, 1957) gives it the compositional (two-layer) form

$$F_k(\mathbf{y}) \;=\; \sum_{q=1}^{2m+1} \Phi_{k,q}\left(\sum_{i=1}^{m} \phi_{q,i}(y_i)\right), \tag{7}$$

exactly a two-layer KAN. Writing p for the highest order produced, the operator

$$\mathcal{J}:\; \mathbf{y}(t) \;\longmapsto\; \big(\hat{x}^{(0)}(t),\, \hat{x}^{(1)}(t),\, \ldots,\, \hat{x}^{(p)}(t)\big) \tag{8}$$

is a KAN with m input edges and $p + 1$ outputs, the k-th estimating $x^{(k)}$. With linear edges it reduces to a finite-difference formula (a fixed weighted sum of the window samples); with spline edges it is the data-adapted generalization, denoising in the manner of a Savitzky–Golay filter. At dense sampling one additive (one-layer) KAN suffices (Lemma S3.8); on undersampled series the window-to-jet map is the transition map between the delay and differential embeddings, which an additive layer measurably fails to represent, and the two-layer form of Eq. (7) is needed (Note S3: propositions and two supervised representation probes, on the Lorenz observable and on a coarse noisy sine-Gordon field).

In either realization, the jet serves as the lifting layer, and everything downstream is unchanged. The substitution is exact when the system is nth-order in the observable, as for Duffing, Genesio–Tesi, and sine-Gordon in u; a hidden coordinate that is not a derivative of the observable is recovered by the trained Takens lift but not by the jet (Section 3.5).

2.3 Training objective

The model is trained end-to-end on a weighted sum of the lifting and equation losses. With N aligned samples in the fitting window,

$$\mathcal{L} \;=\; w_{\text{lift}} \underbrace{\frac{1}{N}\sum_{t}\big(\hat{x}'(t) - x'(t)\big)^2}_{\text{lift loss}} \;+\; \underbrace{\frac{1}{N}\sum_{t}\frac{\big(\hat{x}^{(n)}(t) - x^{(n)}(t)\big)^2}{(\varsigma_n^{\text{out}})^2}}_{\text{equation loss}}, \tag{9}$$

where the targets x' and $x^{(n)}$ come from smoothed finite differencing or a Savitzky–Golay filter, the equation loss is normalized by the target variance, and w_{lift} balances velocity recovery against

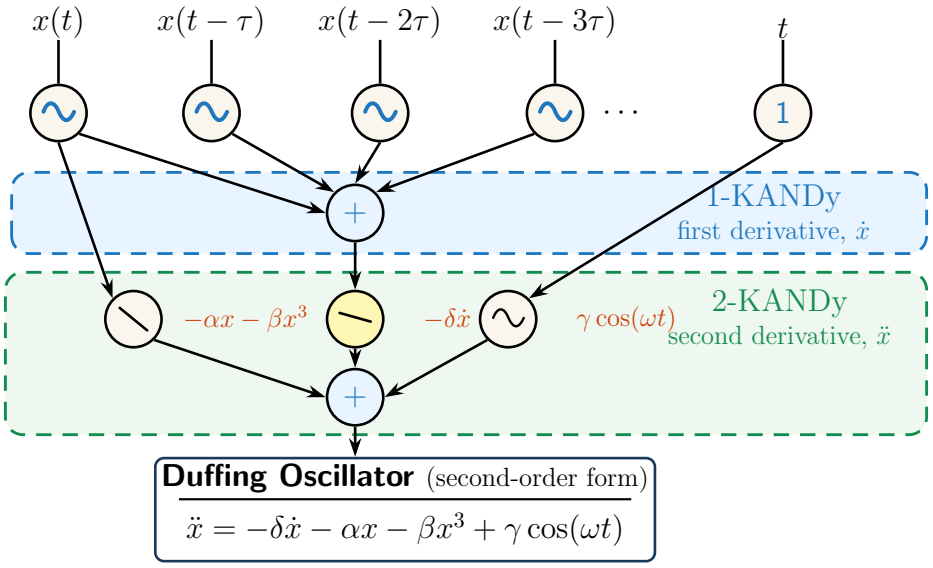

Figure 3: The Higher-Order-KANDy architecture applied to the Duffing oscillator. The inputs to the model are the states x and the delay embedding $x(t-\tau)$ (or the jet-embedding alternative for high-noise data), as shown, and t. Each layer denoted n-KANDy predicts $x^{(n)}$. Each layer feeds the predicted derivative to the next layer.

equation fit. When the lift is bypassed, only the equation loss is optimized, and the measured velocity is used in (3).

2.4 Baselines

On the PDE benchmark we compare pointwise SINDy/PDE-FIND (Brunton et al., 2016; Rudy et al., 2017), weak-form WSINDy (Messenger & Bortz, 2021a;b) with a naive polynomial library and with an oracle library containing $\sin(u)$, and D-CIPHER (Kacprzyk et al., 2023), all on identical noisy data; every method is scored on the recovered restoring force $R(u)$ against $-\sin(u)$ over the observed range, a design that separates derivative-noise error from library misspecification (configurations and D-CIPHER's model class in Note S5).

3 Results

3.1 Systems analyzed and calibration on canonical oscillators

Supplementary Table S1 lists every system, what it is meant to establish, and where its details appear. Three textbook second-order oscillators calibrate the read-out: the forced Duffing oscillator $\ddot{x} = -\alpha x - \beta x^3 - \delta\dot{x} + \gamma\cos(\omega t)$, the undamped pendulum $\ddot{\theta} = -(g/L)\sin\theta$, and the Rayleigh oscillator $\ddot{x} = -x + \mu(\dot{x} - \dot{x}^3)$. On each, the lifting layer recovers the velocity and the equation layer the second derivative, and each Layer-2 edge plotted against its argument overlays the corresponding analytic term, with absent terms collapsing to zero. Figure 3 shows the architecture as applied to Duffing; Supplementary Figs. S1–S4 and Table S2 give the training curves, edge read-outs, and recovered coefficients.

3.2 A native third-order system

The architecture is defined for arbitrary order n (Section 2.1); the oscillators above, like most equation-discovery methods, exercise only $n = 2$. The Genesio–Tesi system tests the native-order claim directly. From its scalar $x(t)$, with $\dddot{x} = -a\,\ddot{x} - b\,\dot{x} - c\,x + x^2$ ($a = 1.2$, $b = 2.92$, $c = 6.0$)—and with $\dot{x}$ and $\ddot{x}$ never supplied—a delay lift ($m = 7$, $\tau = 6$) recovers both intermediate derivatives

($R^2 = 1.000$ and 0.999) and a third-order equation layer predicts $\dddot{x}$ at $R^2 = 1.000$, reading the quadratic nonlinearity off the shape of the x-edge rather than from a supplied library column (Supplementary Fig. S7); the recovered coefficients sit within roughly 8% of truth, and free-running the discovered law reproduces the chaotic attractor. The same system coarsely sampled separates the front end from the equation representation (Note S4): at two samples per swing an oracle-tuned Savitzky–Golay estimate of the derivatives loses three of the four terms, while a delay lift calibrated on a densely sampled trajectory recovers all four to within 1% on a different, noise-free trajectory; oracle least squares and generic STLSQ on the lifted derivatives return the same equation, while the free-form additive discovery layer recovers the same terms with a seed spread of up to 14% on the stiffness from equation-layer collinearity; the one-layer and two-layer lifts tie, as Lemma S3.6 predicts for so weak a nonlinearity. Supporting evidence is in Supplementary Table S3 and Figs. S7 and S8.

3.3 Partial differential equations

We apply the method to nonlinear PDEs by treating each interior spatial location as a scalar time series and supplying the spatial derivatives as additional equation-layer inputs. On the damped sine-Gordon equation $u_{tt} = c^2 u_{xx} - \sin(u) - \gamma u_t$ the equation layer recovers all three right-hand-side terms: the wave-speed term ($c^2 = 0.99$, true 1.0) and the damping ($\gamma = 0.20$, true 0.2) accurately, and the periodic restoring nonlinearity $-\sin(u)$ (coefficient -1.00) across the full observed range $u \in [-1.8,\, 11.6]$ (Supplementary Fig. S5).

Robustness under noise. On the Korteweg–de Vries equation $u_t = -6\,u\,u_x - u_{xxx}$, whose dispersive third derivative strongly amplifies measurement noise, neither pipeline escapes coefficient bias, but they fail differently: as noise rises from 1% to 10% the PDE-FIND (STLSQ) coefficient on $u\,u_x$ collapses from -5.9 to -0.7, spurious cross terms enter the support, and forward integration blows up by $t = 0.08$, whereas the edge read-out degrades more slowly (-4.2 to -2.6), admits one spurious term, and forward-integrates a held-out two-soliton solution with the correct speed and shape (Supplementary Fig. S6).

3.4 Noise robustness versus SINDy and WSINDy

We quantify the restoring-force recovery against the sparse-regression baselines of Section 2.4, on the undamped sine-Gordon kink–antikink collision $u_{tt} = u_{xx} - \sin(u)$. This trajectory is chosen deliberately: at $x = 0$ it sweeps u over a $\pm 2\pi$ period, the only regime in which $-\sin(u)$ is genuinely non-polynomial (a degree-five polynomial fits $-\sin$ to ~ 0.004 RMS over $[-\pi, \pi]$ but degrades to 0.22 over a full period), and the collision breaks the travelling-wave collinearity $u_{tt} = v^2 u_{xx}$, so the wave-speed term $c\,u_{xx}$ and the restoring force $R(u)$ are separately identifiable. We fit $u_{tt} = c\,u_{xx} + R(u)$, the equation layer carrying five free-form edges—u, u_{xx}, and the null controls u_t, u_x, and u_{xt}—with no regularization on any edge (a sensitivity audit confirms the selection owes nothing to a sparsity prior; Note S7), and score two things separately: the support—whether a method selects exactly the true active set $\{u_{xx}, \sin u\}$—and the form, the RMS error of the recovered $R(u)$ against $-\sin(u)$ over the observed range. The learned edges are splines, not symbolic expressions: each term's shape is recovered nonparametrically, and naming it—crediting the restoring edge as $\sin(u)$, or reading a slope—is a post-hoc identification against elementary functions, scored by the shape-correlation gate. A term counts for any method only if its coefficient exceeds 0.01, one percent of the unit-scale true coefficients. Five methods are run at twelve noise levels from 0 to 10%, with an independent noise realization drawn per (level, repeat) over five repeats and the network initialization varied with the repeat; D-CIPHER joins as a reduced diagnostic (one seed at four levels, Supplementary Table S11) (Table 1; the full equation each method returned at every level is given in Supplementary Table S11).[1]

The comparison separates two axes. (i) Free-form discovery versus library: the free-form KAN edge recovers the full-period $-\sin(u)$, whereas both polynomial-library baselines are stuck at one true term out of two at every noise level—a dictionary deficit the weak form cannot fix. (ii) Pointwise versus weak/analytic derivatives. Pointwise derivative estimates bottleneck the second-derivative target, and moving the derivatives off the data—onto a smooth surrogate (the analytic two-dimensional spline jet, Section 2.2)—removes this bottleneck.

[1] All code and experiments are available on GitHub.

		Ground truth	SINDy poly.	WSINDy poly.	WSINDy oracle	**Higher-Order-KANDy**
Derivatives	u_t	○	–	○	○	○
	u_x	○	–	–	–	○
	u_{xx}	✓	✓	✓	✓	✓
	u_{xt}	○	–	–	–	○
Algebraic	1	○	○	○	○	○
	u	○	✗	✗	✗	○
	u^2	○	○	○	○	○
	u^3	○	○	✗	○	○
	u^4	○	○	○	○	○
	u^5	○	○	○	○	○
	$\sin(u)$	✓	⊘	⊘	✓	✓
TP / FP / FN			1 / 1 / 1	1 / 2 / 1	2 / 1 / 0	**2 / 0 / 0**
precision			0.50	0.33	0.67	**1.00**
F_1			0.50	0.40	0.80	**1.00**

Table 1: Discovered term supports for the sine-Gordon kink–antikink collision, $u_{tt} = u_{xx} - \sin(u)$, at 5% noise under the symmetric audit protocol: the Higher-Order-KANDy layer carries five free-form edges (u, u_{xx}, and the null controls u_t, u_x, u_{xt}) with no parsimony penalty on any edge, and a term counts for every method only if its coefficient exceeds 0.01. ✓: term selected; ✗: spurious selected; ⊘: true term never offered; ○: offered, not selected; –: outside the candidate space. Shaded cells: the branch a free-form additive discovery method searches with one learned $g(u)$. D-CIPHER appears as a reduced diagnostic in Supplementary Table S11. Precision separates all methods; recall separates only the polynomial libraries, whose missed $\sin(u)$ is a dictionary limitation. Restricted to its own candidate space each polynomial baseline's recall is 1.00: they fail by coverage (grey ⊘), not selection.

On form the WSINDy oracle—handed the exact $\sin(u)$ column—is the better estimator at every noise level (Supplementary Table S11; Supplementary Fig. S13). At 5% it returns $u_{tt} = -0.998\sin(u) + 1.007\,u_{xx}$ (plus a spurious $-0.012\,u$), within 1% of truth on the true terms, while the jet's coefficients attenuate to -0.790 and 0.879, and by 10% to -0.695 and 0.801.

On support, the ordering reverses. Higher-Order-KANDy selects exactly $\{u_{xx}, \sin u\}$ on all five seeds at every noise level through 7.5%, and on four of five at 10%. For every fit that holds the set, both criteria are met with wide margins: the restoring edge matches $-\sin(u)$ at $\rho \geq 0.952$ against an identification threshold of 0.90, and all three null edges fall to activation ≤ 0.12 against a pruning threshold of 0.30, while both live edges stay above 0.63. The oracle's support degrades instead: exact on all five seeds through 2%, it admits a small spurious linear u (coefficient ≈ 0.01) on some seeds from 3% upward, holding the exact set on two seeds in five at 5% and 10% (Supplementary Table S11 reports agreement with the displayed seed's set, $3/5$, which at those levels includes the spurious u) (Table 1; Supplementary Table S11 gives the level-by-level detail). The polynomial libraries never recover the support at any noise level, because $\sin(u)$ is not in their span.

Backends on identical derivatives. Feeding every backend the same spline-jet samples the equation layer consumes isolates the representation (Note S6). Handed the $\sin(u)$ column, STLSQ on the same derivatives matches or beats the free-form edge on form at every noisy level while retaining a spurious linear u at 5–10%, as the weak-form oracle does; without the column it fails by coverage even at zero noise. A re-implementation of SINDy-KAN (Howard et al., 2026) and the authors' Symbolic–KAN code (Faroughi et al., 2026) on the same derivatives recover no exact support, for structural reasons the note gives.

Learning the derivative part. Replacing the damped sine-Gordon's linear dissipation with a saturating law $u_{tt} = u_{xx} - \sin(u) - \kappa\tanh(\nu u_t)$ puts a genuinely nonlinear function on the derivative axis; with the equation layer unchanged, all three terms are recovered from one noisy field ($\tanh$ damping to $\sim 1\%$ relative error at zero noise, 3% at 2% noise), with nothing named in advance (Supplementary Note S1).

3.5 HIDDEN INTERNAL STATES BEYOND THE FIRST-ORDER JET

By the Takens embedding theorem, the delay lift reconstructs the full state from a scalar observable, including internal variables that are not derivatives of what we measure (Section 2.2). The Bouc–Wen hysteretic oscillator tests this directly: its restoring force depends on a hidden memory variable z with its own non-smooth evolution law, and traces a loop that is multivalued in x and in the instantaneous pair $(x, \dot{x})$—the state a first-order-reduced model sees (Fig. S9a). It is not multivalued in the full jet: by Eq. (13), z is an affine function of $(x, \dot{x}, \ddot{x})$. Reaching it that way, however, requires a clean second derivative and a third-order closure whose right-hand side ($|\dot{x}|\,|z|^2 z$ with z affine in the jet) is not additively separable, so a free-form additive discovery equation layer cannot represent it. The delay window supplies z without either. The two front ends therefore separate sharply: the same KANDy layer is fit from each. Fed the exact instantaneous coordinates, it recovers the hidden state only partially ($R^2 = 0.88$) and the restoring force and acceleration barely at all ($R^2 = 0.23$ and 0.24), whereas fed the delay window, it reconstructs z almost exactly ($R^2 = 1.00$) and predicts both at $R^2 = 0.99$. Reading the edges off the reconstructed state then recovers both laws; with the true z the motion law is within a percent of truth and the memory law within three, and in the observable-only pipeline the coupling holds to under a percent while the stiffness is 7% and the damping 13% high (Note S2).

4 DISCUSSION

Across the systems of Supplementary Table S1 the same stacked construction recovers governing laws at their native order: textbook oscillator coefficients within a few percent (Duffing, pendulum) and within 11% (Rayleigh), a genuinely third-order law from a scalar observable, PDE right-hand sides per spatial location, and—on sine-Gordon—the exact term support on every seed at noise levels up to 7.5%, learning $\sin(u)$ rather than being given it, where an oracle weak-form regression does not.

Parsimony and noise robustness. Sparse regression selects from a fixed library, and when high-order derivatives amplify noise, its coefficients drift, and forward integration diverges (Brunton et al., 2016; Rudy et al., 2017; Lai, 2021); Higher-Order-KANDy instead absorbs each right-hand-side contribution into one bounded edge and reads derivatives off a smooth surrogate, not the data. The two components pay on different axes: the edge recovers what no polynomial dictionary spans (the full-period $-\sin u$); the derivatives come from a smooth map rather than differenced data, and on coarsely sampled series only a learned delay-to-jet lift supplies them accurately (Note S3). A practical corollary (Note S3): delay-based discovery has an optimal sampling rate and the learned map's tangent condition number is U-shaped in the sampling interval—and that number is computable from network and data alone.

Delay-to-jet embeddings. Proposition S3.11 establishes sufficient conditions under which the delay-to-jet map is a change of coordinates between delay and differential embeddings and admits a two-layer Kolmogorov–Arnold representation (Figure 2). Lemma S3.6 supplies the complementary lower bound by showing that a single additive layer can be insufficient. Thus, two layers are sufficient in general and become necessary within the nonseparable regime identified by the lemma. Packard et al. (1980) proposed two ways to reconstruct dynamics from a single signal—time delays and successive derivatives—and the field settled on delays because derivatives amplify noise (Gibson et al., 1992; Krakovská et al., 2022), even though derivatives are the coordinates in which physical laws are written, carry physical units, and reduce the entire vector field to a single closure $x^{(n)} = F$. Higher-Order-KANDy recovers those coordinates by learning the change of representation between the two embeddings: a delay-to-jet map that admits an exact two-layer Kolmogorov–Arnold representation and, on coarsely sampled data, supplies derivatives where fixed differentiation formulas fail.

Limitations. First, where a term library Θ_k is supplied, the equation layer assumes it; though smaller than a sparse-regression dictionary, an inapt library still misattributes the dynamics. Second, under partial observability, a hidden coordinate that is not a derivative of the observable—the Bouc–Wen memory variable z—requires the delay lift, and that lift is a supervised probe: it recovers z at $R^2 = 0.998$, whereas an unsupervised latent matches z in shape only (Note S2); the coefficient read-outs therefore condition on the reconstructed state.

REFERENCES

V. Arnold. On the representation of functions of several variables as a superposition of functions of a smaller number of variables. In A. B. Givental, B. A. Khesin, J. E. Marsden, A. N. Varchenko, V. A. Vassiliev, O. Y. Viro, and V. M. Zakalyukin (eds.), *Collected Works: Representations of Functions, Celestial Mechanics and KAM Theory, 1957–1965*, pp. 25–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-01742-1_5.

James Bagrow and Josh Bongard. Multi-exit Kolmogorov–Arnold networks: enhancing accuracy and parsimony. *Machine Learning: Science and Technology*, 6(3):035037, aug 2025. doi: 10.1088/2632-2153/adf9bd.

Joseph Bakarji, Kathleen Champion, J. Nathan Kutz, and Steven L. Brunton. Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A*, 479(2276):20230422, August 2023. doi: 10.1098/rspa.2023.0422.

Jens Berg and K. Nyström. Data-driven discovery of PDEs in complex datasets. *Journal of Computational Physics*, 384:239–252, 2019. doi: 10.1016/j.jcp.2019.01.036.

Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 104(24): 9943–9948, 2007. doi: 10.1073/pnas.0609476104.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1):19, 2017. doi: 10.1038/s41467-017-00030-8.

Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences of the United States of America*, 116(45):22445–22451, 2019. doi: 10.1073/pnas.1906995116.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Neural Information Processing Systems*, 31, 2018.

Oleksandr Cherednichenko and Maria Poptsova. Kolmogorov–Arnold networks for genomic tasks. *Briefings in Bioinformatics*, 41(2):412–424, 2025. doi: 10.1093/bib/bbaf129.

Kevin Course and Prasanth B. Nair. State estimation of a physical system with unknown governing equations. *Nature*, 622(7982):261–267, 2023. doi: 10.1038/s41586-023-06574-8.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv.org*, 2023. doi: 10.48550/arXiv.2305.01582.

James P. Crutchfield and Bruce S. McNamara. Equations of motion from a data series. *Complex Systems*, 1(3):417–452, 1987.

Paul Dierckx. An algorithm for surface fitting with spline functions. *IMA Journal of Numerical Analysis*, 1(3):267–283, 1981.

Paul Dierckx. *Curve and Surface Fitting with Splines*. Monographs on Numerical Analysis. Oxford University Press, 1993.

Salah A. Faroughi, Farinaz Mostajeran, Amirhossein Arzani, and Shirko Faroughi. Symbolic–KAN: Kolmogorov-Arnold networks with discrete symbolic structure for interpretable learning. *Journal of Computational Physics*, 566:115223, 2026. doi: 10.1016/j.jcp.2026.115223.

John F. Gibson, J. Doyne Farmer, Martin Casdagli, and Stephen Eubank. An analytic approach to practical state space reconstruction. *Physica D: Nonlinear Phenomena*, 57(1–2):1–30, 1992. doi: 10.1016/0167-2789(92)90085-2.

Federico Girosi and Tomaso Poggio. Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation*, 1(4):465–469, 1989. doi: 10.1162/neco.1989.1.4.465.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Neural Information Processing Systems*, volume 32, pp. 15353–15363, 2019.

Juncai He. On the optimal expressive power of ReLU DNNs and its application in approximation with Kolmogorov superposition theorem. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. doi: 10.1109/TNNLS.2024.3514126.

Seth M. Hirsh, David A. Barajas-Solano, and J. Nathan Kutz. Sparsifying priors for Bayesian uncertainty quantification in model discovery. *Royal Society Open Science*, 9(2):211823, 2022. doi: 10.1098/rsos.211823.

Amanda A. Howard, Nicholas Zolman, Bruno Jacob, Steven L. Brunton, and Panos Stinis. SINDy-KANs: Sparse identification of non-linear dynamics through Kolmogorov-Arnold networks, 2026.

Krzysztof Kacprzyk, Zhaozhi Qian, and M. Schaar. D-CIPHER: Discovery of closed-form partial differential equations. volume 36, pp. 27609–27644, 2023. doi: 10.52202/075280-1201.

Kadierdan Kaheman, Steven L. Brunton, and J. Nathan Kutz. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *Machine Learning: Science and Technology*, 3(1):015031, 2022. doi: 10.1088/2632-2153/ac567a.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

William J. Knottenbelt, William McGough, R. Wray, Woody Z. Zhang, Jiashuai Liu, I. Machado, Ze-Yu Gao, and M. Crispin-Ortuzar. CoxKAN: Kolmogorov–Arnold networks for interpretable, high-performance survival analysis. *Bioinformatics*, 41(8):btaf413, 07 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf413.

Benjamin C. Koenig, Suyong Kim, and Sili Deng. KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics. *Computer Methods in Applied Mechanics and Engineering*, 432:117397, 2024. ISSN 0045-7825. doi: 10.1016/j.cma.2024.117397.

Benjamin C. Koenig, Suyong Kim, and Sili Deng. LeanKAN: a parameter-lean Kolmogorov-Arnold network layer with improved memory efficiency and convergence behavior. *Neural Networks*, 192: 107883, 2025. ISSN 0893-6080. doi: 10.1016/j.neunet.2025.107883.

A. N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114:953, 1957. doi: 10.1007/978-94-011-3030-1_56.

Mario Köppen. On the training of a Kolmogorov Network. In *Artificial Neural Networks*, pp. 474–479. Springer, 2002. doi: 10.1007/3-540-46084-5_77.

Anna Krakovská, Štefan Pócoš, Karolína Mojžišová, Iveta Bečková, and Jakub Xaver Gubáš. State space reconstruction techniques and the accuracy of prediction. *Communications in Nonlinear Science and Numerical Simulation*, 111:106422, 2022. ISSN 1007-5704. doi: https://doi.org/10.1016/j.cnsns.2022.106422.

Ming-Jun Lai and Zhaiming Shen. The Kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions. *arXiv.org*, 2021. doi: 10.48550/arXiv.2112.09963.

Ying-Cheng Lai. Finding nonlinear system equations and complex network structures from data: A sparse optimization approach. *Chaos*, 31(8):082101, August 2021. doi: 10.1063/5.0062042.

Longlong Li, Yipeng Zhang, Guanghui Wang, and Kelin Xia. Kolmogorov–Arnold graph neural networks for molecular property prediction. *Nature Machine Intelligence*, 7(8):1346–1354, 2025. doi: 10.1038/s42256-025-01087-7.

Weizhen Li and Rui Carvalho. Automating the discovery of partial differential equations in dynamical systems. *Machine Learning: Science and Technology*, 5(3):035046, 2024. doi: 10.1088/2632-2153/ad682f. arXiv:2404.16444.

Ji-Nan Lin and Rolf Unbehauen. On the realization of a Kolmogorov network. *Neural Computation*, 5(1):18–20, 1993. doi: 10.1162/neco.1993.5.1.18.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–Arnold networks. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2404.19756.

Ziming Liu, Max Tegmark, Pingchuan Ma, Wojciech Matusik, and Yixuan Wang. Kolmogorov–Arnold networks meet science. *Phys. Rev. X*, 15:041051, Dec 2025. doi: 10.1103/4t7t-v19l.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Georg Martius and Christoph H. Lampert. Extrapolation and learning equations. *International Conference on Learning Representations*, 2016.

D. Messenger, E. Dall'Anese, and D. Bortz. Online weak-form sparse identification of partial differential equations, 2022.

Daniel A. Messenger and David M. Bortz. Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, 2021a. doi: 10.1137/20M1343166.

Daniel A. Messenger and David M. Bortz. Weak SINDy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021b. doi: 10.1016/j.jcp.2021.110525.

Daniel A. Messenger and David M. Bortz. Learning mean-field equations from particle data using WSINDy. *Physica D: Nonlinear Phenomena*, 439:133406, 2022. doi: 10.1016/j.physd.2022.133406.

Hadrien Montanelli and Haizhao Yang. Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem. *Neural Networks*, 129:1–6, 2020. doi: 10.1016/j.neunet.2019.12.013.

A. Norcliffe, Cristian Bodnar, Ben Day, Nikola Simidjievski, and Pietro Lio'. On second order behaviour in augmented neural ODEs. In *Neural Information Processing Systems*, volume 33, 2020. arXiv:2006.07220.

N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical Review Letters*, 45(9):712–716, 1980.

Shirin Panahi, Mohammadamin Moradi, Erik M. Bollt, and Ying-Cheng Lai. Data-driven model discovery with Kolmogorov-Arnold networks. *Physical Review Research*, 7:023037, April 2025. doi: 10.1103/PhysRevResearch.7.023037.

Nataly R. Panczyk, Omer F. Erdem, and Majdi I. Radaideh. Opening the AI black-box: Symbolic regression with Kolmogorov–Arnold networks. *Energy and AI*, 22:100258, 2025. doi: 10.1016/j.egyai.2025.100595.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017. doi: 10.1126/sciadv.1602614.

Subham S. Sahoo, Christoph H. Lampert, and Georg Martius. Learning equations for extrapolation and control. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4442–4450, 2018.

Tim Sauer, James A. Yorke, and Martin Casdagli. Embedology. *Journal of Statistical Physics*, 65 (3–4):579–616, 1991. doi: 10.1007/BF01053745.

Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A*, 473(2197):20160446, 2017. doi: 10.1098/rspa.2016.0446.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893.

Hiroki Shiraishi, H. Ishibuchi, and Masaya Nakata. X-KAN: Optimizing local Kolmogorov–Arnold networks via evolutionary machine learning. In *International Joint Conference on Artificial Intelligence*, pp. 8930–8938, 2025. doi: 10.24963/ijcai.2025/993.

Kevin Slote, Erik Bollt, and Jeremie Fish. Deep-Koopman-KANDy: Dictionary discovery for deep-koopman operators with Kolmogorov-Arnold networks for dynamics, 2026a.

Kevin Slote, Jeremie Fish, and Erik Bollt. KANDy: Kolmogorov–arnold networks and dynamical system discovery. *arXiv preprint arXiv:2602.20413*, 2026b. doi: 10.48550/arXiv.2602.20413.

David A Sprecher and Sorin Draghici. Space-filling curves and Kolmogorov superposition-based neural networks. *Neural Networks*, 15(1):57–67, 2002. doi: 10.1016/S0893-6080(01)00107-1.

Floris Takens. Detecting strange attractors in turbulence. In David Rand and Lai-Sang Young (eds.), *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*, pp. 366–381. Springer Berlin Heidelberg, Berlin, Heidelberg, 1981. doi: 10.1007/BFb0091924.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. doi: 10.1126/sciadv.aay2631.

Andrew J. Whalen, Sean N. Brennan, Timothy D. Sauer, and Steven J. Schiff. Observability and controllability of nonlinear networks: The role of symmetry. *Physical Review X*, 5:011005, 2015. doi: 10.1103/PhysRevX.5.011005.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5.

USE OF LARGE LANGUAGE MODELS

We used a generative AI assistant to restructure this manuscript for readability and length, to draft several passages, to generate code that we validated and ran ourselves, and as an adversarial fact-checker of the text, figures, and tables. We reviewed all AI-assisted work and take responsibility for the paper's final content. No scientific content was devised by an LLM.

Table S1: Every system analyzed, the native order n at which it is treated, what the model observes, and what the experiment establishes. Rows above the rule are calibration, native-order, and representation checks reported in full in the supplement; the boldface rows below it carry the paper's headline claims and are detailed in the main text. Hyperparameters for the trained models are listed in Supplementary Table S5.

System	n	Obs.	What it establishes	Details
Duffing (forced)	2	x	all four coefficients within a few %	Tab. S2, Fig. S2
Pendulum	2	θ	full $-(g/L)\sin\theta$ edge; absent damping collapses to zero	Tab. S2, Fig. S3
Rayleigh	2	x	nonlinearity carried on the velocity edge, within 11%	Tab. S2, Fig. S4
Genesio–Tesi	3	x	native third order from a scalar; coefficients within 8%, attractor reproduced; survives two samples per swing	Tab. S3, Fig. S8, Note S4
Korteweg–de Vries	1	$u(x,t)$	degrades slowly where PDE-FIND diverges under noise	Fig. S6
Sine–Gordon (damped)	2	$u(x,t)$	all three right-hand-side terms recovered	Fig. S5
Sine–Gordon (tanh damping)	2	$u(x,t)$	nonlinear damping $-\kappa\tanh(\nu u_t)$ learned on the derivative axis	Note S1
Lorenz (probe)	—	x	delay-to-jet map: additive when dense, compositional when undersampled	Note S3
Sine–Gordon (coarse, noisy; probe)	2	$u(x,t)$	coefficients recovered from 80×33 samples at 1% noise by a compositional (two-layer) patch operator calibrated on exact solutions	Note S3
Sine–Gordon (collision)	2	$u(x,t)$	**exact term support to 7.5% noise, library-free**	Section 3.4
Bouc–Wen	2+1	x	**hidden non-derivative state supplied by the delay lift**	Section 3.5, Note S2

System	Term	Recovered	True
Duffing	α (restoring, x)	1.023	1.0
	β (cubic, x^3)	0.962	1.0
	γ (forcing)	0.492	0.5
	δ (damping)	0.294	0.3
Pendulum	g/L	9.800	9.810
	damping	0.0000	0.0
Rayleigh	stiffness	-0.900	-1.0
	μ_{lin}	0.910	1.0
	μ_{cub}	-0.890	-1.0

Table S2: Recovered versus true coefficients for the canonical ODEs.

Table S3: Recovered versus true coefficients for the third-order Genesio–Tesi system, from the scalar observable $x(t)$ alone.

Term	Role	Recovered	True
a	$\ddot{x}$ damping	1.29	1.20
b	$\dot{x}$ damping	2.90	2.92
c	x stiffness	6.39	6.00
q	x^2 nonlinearity	1.06	1.00

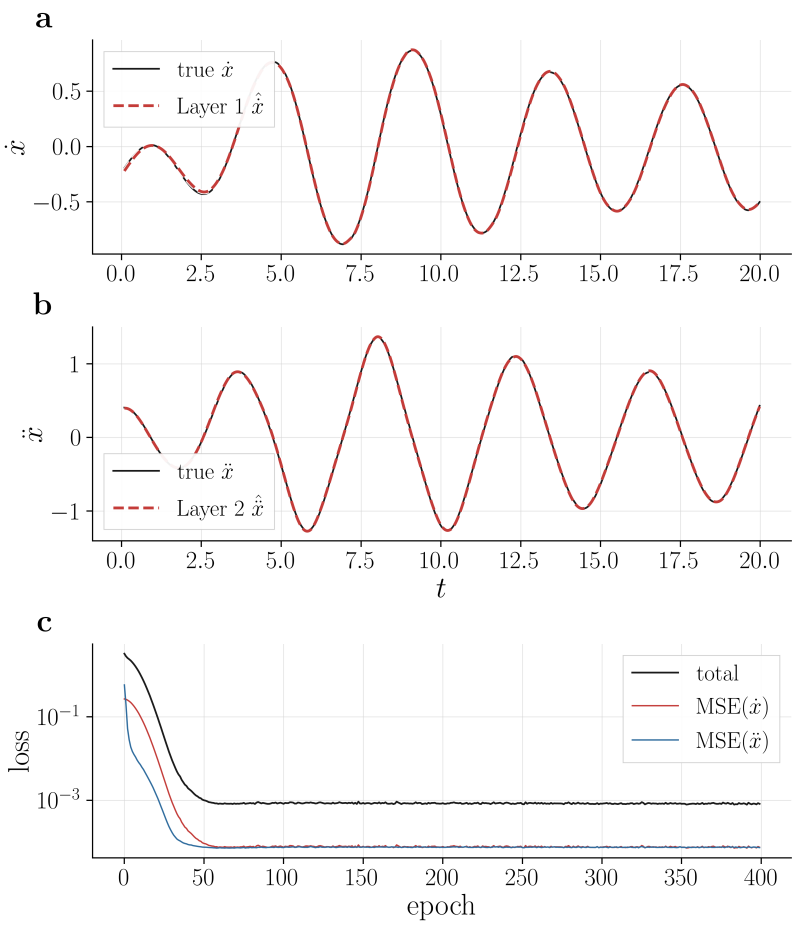

Figure S1: **Duffing oscillator: trajectory reconstruction.** Top: true velocity $\dot{x}$ versus the Layer-1 (lifting) prediction. Middle: true acceleration $\ddot{x}$ versus the Layer-2 (equation) prediction. Bottom: training losses (total, lift MSE, equation MSE) on a log scale. The two learned curves are visually indistinguishable from ground truth and the loss converges within $\sim$100 epochs.

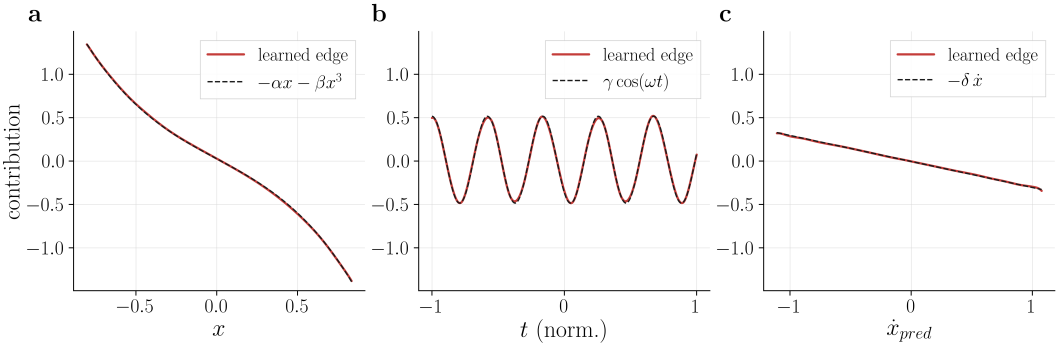

Figure S2: **Duffing: recovered equation terms.** Learned Layer-2 edges (red) versus analytic terms (black dashed), mean-removed. Recovered coefficients: $\alpha = 1.023$, $\beta = 0.962$ (true $1.0, 1.0$); $\gamma = 0.492$ (true 0.5); $\delta = 0.294$ (true 0.3).

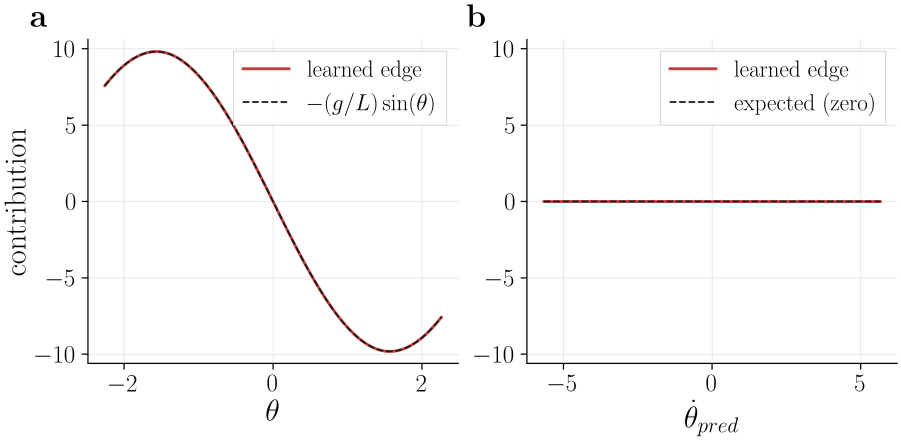

Figure S3: **Pendulum: recovered equation terms.** The state edge recovers $-(g/L)\sin\theta$ with $g/L = 9.800$ (true 9.810); the velocity edge is flat, giving a damping coefficient of 0.0 (true 0).

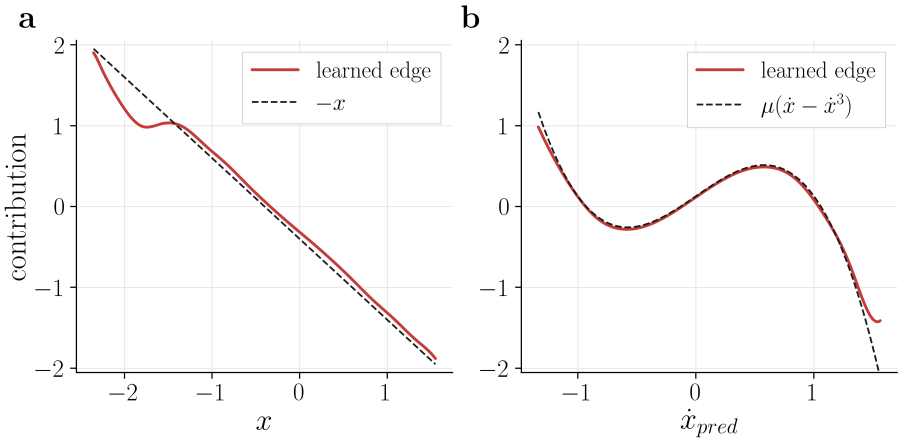

Figure S4: **Rayleigh oscillator: recovered equation terms.** Linear restoring force (stiffness -0.900, true -1.0) and cubic nonlinear damping on the velocity edge ($\mu_{\text{lin}} = 0.910$, $\mu_{\text{cub}} = -0.890$; true $1.0, -1.0$).

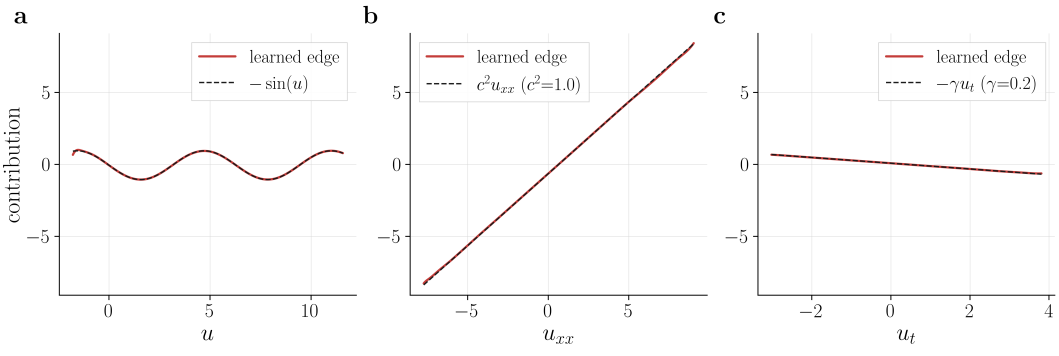

Figure S5: **Sine-Gordon PDE: recovered equation terms.** Learned edges (red) versus analytic terms (black dashed) for the u, u_{xx}, and u_t inputs. All three terms are recovered accurately ($c^2 = 0.99$, $\gamma = 0.20$, sin coefficient -1.00), the restoring nonlinearity tracking $-\sin(u)$ over the full observed range $u \in [-1.8,\ 11.6]$.

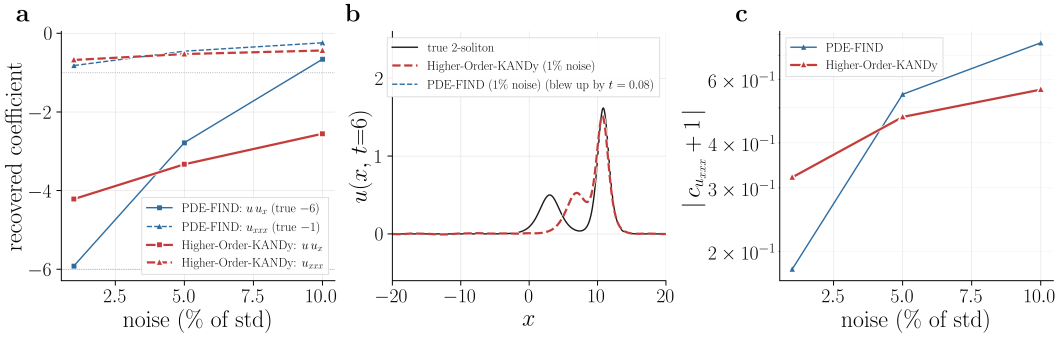

Figure S6: **KdV: robustness to noise in the third spatial derivative.** (A) Recovered coefficients versus noise: both methods drift from truth, the baseline faster and with spurious support entering at every level. (B) Forward integration of a held-out two-soliton solution: the baseline blows up by $t = 0.08$, the proposed recovery matches speed and shape (RMS 0.165 at $t = 6$, 1% noise). (C) u_{xxx} coefficient error versus noise (log scale).

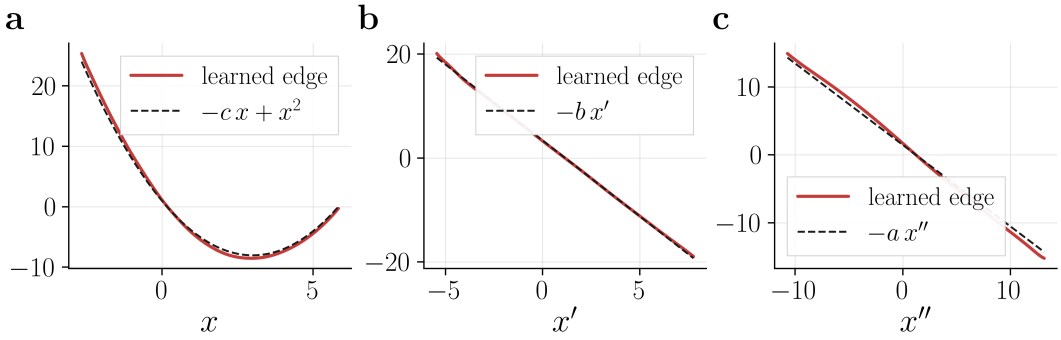

Figure S7: **Third-order discovery (Genesio–Tesi): recovered equation terms.** Learned equation-layer edges (red) against their analytic terms (black dashed). **a:** the x edge carries the stiffness and the quadratic nonlinearity, $-c\,x + x^2$, read off the shape of the edge rather than supplied as a library column. **b:** the x' edge carries the linear damping $-b\,x'$. **c:** the x'' edge carries $-a\,x''$.

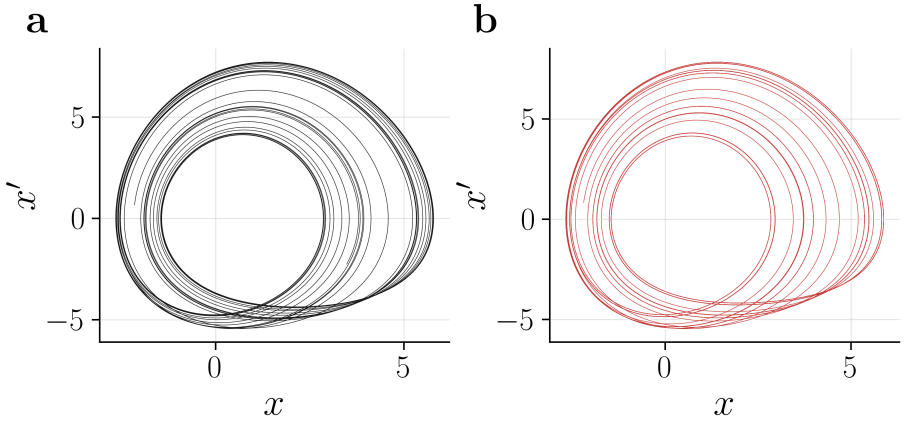

Figure S8: **Third-order discovery (Genesio–Tesi): free-run attractor. a:** the true Genesio–Tesi chaotic attractor in the (x, x') plane. **b:** the attractor produced by freely integrating the discovered third-order equation from a single initial condition.

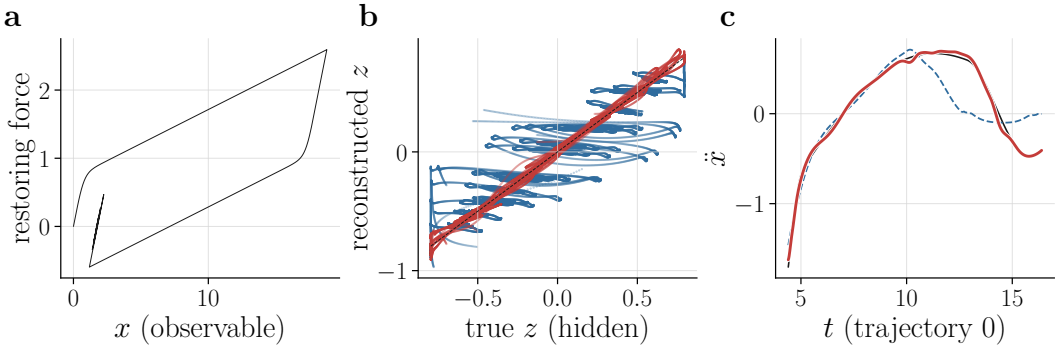

Figure S9: **Bouc–Wen hysteresis and hidden memory variables. a:** the restoring force traces a hysteresis loop—it is multivalued in the observable x and in $(x, \dot{x})$, so no function of the instantaneous pair can represent it. **b:** reconstruction of the hidden memory variable z: from the instantaneous coordinates $(x, \dot{x})$ (blue) the fit is spread and biased at the load reversals, whereas from the delay window (red) it collapses onto the diagonal ($R^2 = 1.00$). **c:** the acceleration $\ddot{x}$ over one trajectory—the Takens delay lift (red) tracks the truth (black) while the instantaneous predictor (blue) fails.

Law	Term	Recovered	True
Motion (13)	αk (restoring, x)	0.100	0.100
	c (damping, $\dot{x}$)	0.151	0.150
	$(1-\alpha)k$ (coupling, z)	0.900	0.900
Memory (14)	A ($\dot{x}$)	0.99	1.0
	β ($\lvert\dot{x}\rvert z^3$)	0.97	1.0
	γ ($\dot{x}\lvert z\rvert^3$)	1.01	1.0

Table S4: Recovered versus true coefficients for the Bouc–Wen oscillator. The motion-law coefficients shown are read with the reconstructed state (true z); the observable-only pipeline (delays $\to \hat{z} \to$ equation layer) gives -0.107, -0.170, -0.906 (Note S2). The memory law is identified given the lifted state, which the delays recover at $R^2 = 0.998$.

Experiment	Basis	n	Grid	m/τ	Epochs	LR	Batch
Canonical second-order ODEs							
Duffing	B-spline	2	25	4/3	400	2×10^{-3}	128
Pendulum	B-spline	2	25	3/5	400	2×10^{-3}	128
Rayleigh	B-spline	2	25	4/3	400	2×10^{-3}	128
Higher-order systems							
Genesio–Tesi	B-spline	3	25	7/6	1200	2×10^{-3}	512
Nonlinear PDEs							
Sine–Gordon (damped)	B-spline	2	35	4/3	400	2×10^{-3}	256
Sine–Gordon (tanh damping)	B-spline	2	25	–	400	2×10^{-3}	256
Sine–Gordon (WSINDy benchmark)	B-spline	2	25	–	400	2×10^{-3}	256
Korteweg–de Vries	B-spline	1	25	–	400	10^{-3}	256
Additional systems							
Bouc–Wen hysteresis	B-spline	2	12 (8 mem.)	12/40	350–500	5×10^{-3}	512

Table S5: Hyperparameters for each experiment. n is the ODE/PDE order, "Grid" the KAN grid size, and m/τ the number of delays and delay step (in samples); all KAN edges use spline order 3 and the Adam optimizer. Dashes mark inapplicable fields: the sine-Gordon WSINDy and tanh-damping benchmarks and Korteweg–de Vries use the analytic jet, so none has a delay embedding. For Bouc–Wen, the state and force probes use grid 12 at 350 epochs, the motion-law read-out 400 epochs, and the memory-law read-out grid 8 at 500 epochs.

S1 LEARNING A NONLINEAR DERIVATIVE TERM

The damped sine-Gordon of Section 3.3 of the main text has linear dissipation, so its u_t edge is recovered as a slope and is indistinguishable in kind from a scalar coefficient. To test the derivative axis itself, we replace it with a saturating law,

$$u_{tt} \;=\; u_{xx} \;-\; \sin(u) \;-\; \kappa\,\tanh(\nu\,u_t), \qquad \kappa = 0.15,\ \nu = 1.5, \tag{10}$$

evaluated on a breather, whose oscillation sweeps u_t through both signs so that an odd damping law is identifiable. The equation layer is unchanged: free-form edges over $\{u,\, u_{xx},\, u_t\}$ (a mixed feature set where u_{xx} is a spatial derivative), giving

$$\hat{u}_{tt} \;=\; R(u) \;+\; \varphi_2(u_{xx}) \;+\; \varphi_3(u_t). \tag{11}$$

All three terms are recovered simultaneously from one noisy field: $R(u)$ matches $-\sin(u)$ to 0.0020 RMS at zero noise and 0.0068 at 2%; φ_2 is linear with slope 1.001 (true $+1$); and φ_3 matches $-\kappa\tanh(\nu u_t)$ to 0.0011 and 0.0036 respectively, against a term whose own RMS amplitude is 0.1115—roughly 1% relative error at zero noise and 3% at 2% noise. Nothing was named in advance.

D-CIPHER (Kacprzyk et al., 2023) searches for a relation

$$\sum_j \eta_j\,\mathcal{O}_j u \;=\; g(\mathbf{x}, u), \qquad \eta_j \in \mathbb{R}, \tag{12}$$

in which the $\mathcal{O}_j$ are differential operators and only the source g is discovered in closed form. The derivative dependence is therefore a linear combination with scalar weights, and g may not depend on u_t; no choice of $\{\eta_j, g\}$ reproduces $\kappa\tanh(\nu u_t)$, so (10) lies outside the model class. The same holds for SINDy and WSINDy unless powers of u_t are placed in Θ by hand, and a truncated polynomial column set cannot represent a saturating function. To quantify the gap without disadvantaging that family we compute its ceiling—the best constant-coefficient damping $\eta\,u_t$ fitted to the same learned edge—which is 0.0238. Higher-Order-KANDy is thus $21\times$ closer to the true damping law at zero noise and still $6.6\times$ closer at 2%, and the ceiling is flat in the noise level because it is a structural limit of the model class rather than a statistical one.

S2 HYSTERESIS AND HIDDEN MEMORY VARIABLES (BOUC–WEN)

By the Takens embedding theorem the delay lift reconstructs the full state from a scalar observable, including internal variables that are not derivatives of what we measure (Section 2.2). We test this directly on the Bouc–Wen hysteretic oscillator, a standard model in structural dynamics whose restoring force depends on a hidden memory variable z with its own nonlinear, non-smooth evolution law:

$$\ddot{x} = -\alpha k\,x - c\,\dot{x} - (1-\alpha)k\,z, \tag{13}$$

$$\dot{z} = A\,\dot{x} - \beta\,|\dot{x}|\,|z|^{n-1}z - \gamma\,\dot{x}\,|z|^{n}, \tag{14}$$

with unit mass, $c = 0.15$, $k = 1$, $\alpha = 0.1$ (hysteresis-dominated, so 90% of the restoring force is carried by z), and $A = \beta = \gamma = 1$, $n = 3$. We observe only $x(t)$, from an ensemble of free-decay trajectories of differing amplitude. Because the restoring force traces a hysteresis loop, it is multivalued in x—and in the instantaneous pair $(x, \dot{x})$, the state a first-order-reduced model sees (Fig. S9a). It is not multivalued in the full jet: by Eq. (13), $z = -(\ddot{x} + \alpha k x + c\dot{x})/((1-\alpha)k)$ is affine in $(x, \dot{x}, \ddot{x})$. Reaching it that way requires a clean second derivative and a third-order closure $\dddot{x} = F(x, \dot{x}, \ddot{x})$ whose right-hand side, obtained by substituting this z into Eq. (14), contains $|\dot{x}|\,|z|^2 z$ and is not additively separable; a library-free equation layer cannot represent it. The delay window supplies z without either.

This is precisely the regime the delay lift is built for, and it sharply separates the two front ends; to isolate the coordinates rather than the model, the same KAN layer is fit from each. Handed the exact instantaneous coordinates $(x, \dot{x})$—the idealized jet front end, and the best case for SINDy—it recovers the hidden state only partially ($R^2 = 0.88$) and the restoring force and acceleration barely at all ($R^2 = 0.23$ and 0.24), because both are dominated by the load reversals where z is genuinely multivalued in $(x, \dot{x})$. The delay window resolves the ambiguity: it reconstructs z almost exactly

($R^2 = 1.00$; Fig. S9b), and predicts the restoring force and acceleration at $R^2 = 0.99$ (Supplementary Fig. S9c).

The motion law (13) is linear in $(x, \dot{x}, z)$. With the reconstructed state the edges recover $\ddot{x} = -0.100\,x - 0.151\,\dot{x} - 0.900\,z$—i.e. $\alpha k = 0.10$, $c = 0.15$, and coupling $(1-\alpha)k = 0.90$, hence $k = 1.00$ and $\alpha = 0.10$, every term within a percent of truth. Replacing the true z by the delay-reconstructed $\hat{z}$ (an observable-only pipeline: delays $\to \hat{z} \to$ equation layer) gives $-0.107\,x - 0.170\,\dot{x} - 0.906\,z$: the coupling stays within a percent, while the stiffness reads 7% and the damping 13% high, the cost of the lift's residual error concentrating on the smallest term. The memory law (14) is nonlinear and non-smooth; because a KAN layer sums univariate edges, its two cross terms enter as equation-layer features $|\dot{x}|z^3$ and $\dot{x}|z|^3$, and—given the lifted state, which the delays recover at $R^2 = 0.998$—the edges recover $\dot{z} = 0.99\,\dot{x} - 0.97\,|\dot{x}|z^3 - 1.01\,\dot{x}|z|^3$, i.e. $A, \beta, \gamma \approx 1$ (Supplementary Table S4). Recovering this coupled, non-smooth hysteretic law from a single scalar channel is possible only because the delay lift first supplies the hidden state that the instantaneous pair $(x, \dot{x})$ cannot, and that the 2-jet reaches only through a non-separable third-order closure.

The delay lift used above is a supervised probe: it is trained against the true z, and on held-out data recovers it at $R^2 = 0.998$. Recovering the same state without supervision is harder. A latent trained only to predict the acceleration improves the held-out acceleration fit from $R^2 = 0.40$ (a comparable instantaneous-coordinate baseline on that experiment's held-out split) to 0.82 ± 0.12 across seeds, yet aligns with the true z in shape only (Pearson 0.79 ± 0.11; aligned $R^2 \leq 0$). The coefficient read-outs reported here therefore condition on the reconstructed state, and closing the gap between supervised probes and unsupervised latent recovery remains open.

S3 Undersampling and the compositional delay-to-jet map

The lifting layer maps a delay window to derivative estimates, and the question of how much architecture that map requires has a sharp answer set by the sampling rate rather than by noise. When the observable is densely sampled, differentiation is a local, linear operation: an optimal linear map on the window—a fixed weighted sum of neighboring samples, the class containing finite differences and Savitzky–Golay—is exact to the truncation order of Lemma S3.8, and nonlinearity gains nothing. When the data are undersampled, no Taylor-based finite-difference formula approximates the derivative—for nonlinear f, no fixed linear combination of window samples does (for linear f a flow-adapted linear map can remain exact, Remark S3.4); the window determines the state only through the inverse of the delay map Φ, and the derivative is the Lie derivative of the observable evaluated at that state,

$$x'(t) \;=\; (\mathcal{L}_f h)\big(\Phi^{-1}(\mathbf{y}_t)\big), \qquad \mathbf{y}_t = \big(x(t),\, x(t-\tau),\, \ldots,\, x(t-(m-1)\tau)\big), \tag{15}$$

a genuine composition. For nonlinear f this map is generically not additively separable, so a single KAN layer—a sum of univariate edges $\sum_i \phi_i(x(t-i\tau))$—is structurally too small a class, while the two-layer Kolmogorov–Arnold composition contains it.

Additionally, combining results from the Takens' delay embedding theorem and the Kolmogorov–Arnold representation theorem, we show the existence of a map that embeds the attractor in the differential (when the map is injective) or in a jet embedding. This section first makes these arguments precise, then measures the gap directly.

Theoretical basis. Let φ^t be the flow of $\dot{s} = f(s)$ on a compact invariant set $\mathcal{A}$ of box-counting dimension d, let h be the scalar observable, $x(t) = h(s(t))$, and let

$$\Phi(s) = \big(h(s),\, h(\varphi^{-\tau}s),\, \ldots,\, h(\varphi^{-(m-1)\tau}s)\big)$$

be the delay map, so that $\mathbf{y}_t = \Phi(s(t))$. For generic (h, τ) and $m \geq 2d+1$, Φ is an embedding of $\mathcal{A}$, and a diffeomorphism onto its image (Takens, 1981; Sauer et al., 1991).

Proposition S3.1 (The jet identifiability at any sampling rate)**.** *Assume $f, h \in C^{k+1}$ and that Φ is an embedding. Then there is a continuous (and smooth when $\mathcal{A}$ is a smooth invariant manifold) function $G_k : \Phi(\mathcal{A}) \to \mathbb{R}$ with $x^{(k)}(t) = G_k(\mathbf{y}_t)$ for all t, namely $G_k = (\mathcal{L}_f^k h) \circ \Phi^{-1}$, where $\mathcal{L}_f$ is the Lie derivative along f. No smallness condition on τ enters: the delay step is constrained only by the genericity of the embedding.*

Proof. $x(t+\sigma) = h\big(\varphi^\sigma(s(t))\big)$; differentiating k times at $\sigma = 0$ gives $x^{(k)}(t) = (\mathcal{L}_f^k h)(s(t))$, a C^1 function of the state. Since Φ is an embedding, $s(t) = \Phi^{-1}(\mathbf{y}_t)$ with Φ^{-1} smooth on the compact image; compose. □

Corollary S3.2 (The jet embedding is a change of chart)**.** *For generic (h, f) and $p \geq 2d$, the derivative map*

$$\Theta(s) = \big(h(s),\, (\mathcal{L}_f h)(s),\, \ldots,\, (\mathcal{L}_f^p h)(s)\big)$$

is itself an embedding of $\mathcal{A}$ and is the classical differential embedding, and the observability map of the system (Takens, 1981; Packard et al., 1980; Whalen et al., 2015). The full lift $G = (G_0, \ldots, G_p) = \Theta \circ \Phi^{-1}$ of Proposition S3.1 is therefore the transition map between two generic charts of the same attractor: the delay chart, which is what a scalar sensor measures, and the jet chart, in which a system that is nth-order in the observable takes the graph form $x^{(n)} = F(x, \ldots, x^{(n-1)})$ of Eq. (1)*. The lifting layer thus constructs a new embedding of the data, and the map it must represent is a diffeomorphism between embedded copies of $\mathcal{A}$—a generic composition, with no reason to be additively separable in the delay coordinates.*

Proposition S3.3 (Diffeomorphism, conditioning, and stability)**.** *Assume, in addition to the hypotheses of Proposition S3.1, that $\mathcal{A}$ is a compact smooth invariant manifold with the ambient-induced metric and that $\Theta = (h, \mathcal{L}_f h, \ldots, \mathcal{L}_f^p h)$ is an embedding of $\mathcal{A}$. Write $F = \Theta \circ \Phi^{-1}$ and $\mathbf{z} = (z_0, \ldots, z_p)$ for the jet coordinates. Then:*

- *(i) $F : \Phi(\mathcal{A}) \to \Theta(\mathcal{A})$ is a diffeomorphism; its differential between attractor tangent spaces is $DF_{\Phi(s)} = D\Theta_s \circ (D\Phi_s)^{-1}$, and F conjugates the two coordinate flows, $F \circ \big(\Phi\, \varphi^t\, \Phi^{-1}\big) = \big(\Theta\, \varphi^t\, \Theta^{-1}\big) \circ F$.*
- *(ii) In the jet chart the dynamics take the companion form $\dot z_0 = z_1,\ \ldots,\ \dot z_{p-1} = z_p,\ \dot z_p = G(\mathbf{z})$ with the single closure $G = (\mathcal{L}_f^{p+1} h) \circ \Theta^{-1}$—the graph form of Eq.* (1) *that the equation layer reads: in this chart the entire vector field reduces to one unknown function.*
- *(iii) Let $\alpha_\Delta = \inf_{s \in \mathcal{A}} \sigma_{\min}\big(D\Phi_s|_{T_s\mathcal{A}}\big)$ and $M = \sup_{s \in \mathcal{A}} \|D\Theta_s\|$. Then $\alpha_\Delta > 0$ and $\sup_{\mathbf{y} \in \Phi(\mathcal{A})} \big\|DF_{\mathbf{y}}|_{T_{\mathbf{y}}\Phi(\mathcal{A})}\big\| \leq M/\alpha_\Delta$.*
- *(iv) If a learned map $\widehat F$ satisfies $\|\widehat F - F \circ \pi\|_\infty \leq \varepsilon$ on a neighborhood of $\Phi(\mathcal{A})$ of radius below its reach, with π the nearest-point projection, then for a measured delay vector $\tilde{\mathbf{y}} = \Phi(s) + \eta$ with $\|\eta\|$ below that radius,*

$$\|\widehat F(\tilde{\mathbf{y}}) - \Theta(s)\| \;\leq\; \varepsilon + c\,\frac{M}{\alpha_\Delta}\,\|\eta\|, \tag{16}$$

with c a curvature constant of the image ($c = 1$ for perturbations tangent to $\Phi(\mathcal{A})$).

Proof. We prove the claims in turn.

(i) Compositions and inverses of diffeomorphisms onto images; the conjugacy is immediate from $F = \Theta \circ \Phi^{-1}$.

(ii) $\dot z_i = \frac{d}{dt}\,\mathcal{L}_f^i h = \mathcal{L}_f^{i+1} h = z_{i+1}$ for $i < p$, and $\dot z_p = (\mathcal{L}_f^{p+1} h)(s) = G(\mathbf{z})$ since Θ inverts.

(iii) $\sigma_{\min}\big(D\Phi|_{T\mathcal{A}}\big)$ is continuous and positive on compact $\mathcal{A}$ because Φ immerses; the bound is the chain rule with $\|(D\Phi)^{-1}\| = 1/\sigma_{\min}(D\Phi)$.

(iv) Triangle inequality, then the mean-value bound of (iii) along a path in $\Phi(\mathcal{A})$ from $\Phi(s)$ to $\pi(\tilde{\mathbf{y}})$, whose length is at most $c\,\|\eta\|$ within the reach.

□

The sampling interval enters Eq. (16) only through the conditioning α_Δ of the delay embedding—there is no finite-difference truncation factor Δ^{-k} anywhere in the bound. Coarse sampling therefore degrades reconstruction only insofar as it degrades the observability of the delay map, and the failure criterion is $\alpha_\Delta \to 0$ (or loss of injectivity), not large Δ per se.

Remark S3.4 (An exact delay-to-derivative map: the harmonic oscillator). For $\ddot{x} = -\omega^2 x$ observed through x, two delays suffice: from $x(t-\Delta) = x(t)\cos\omega\Delta - (\dot{x}(t)/\omega)\sin\omega\Delta$,

$$\dot{x}(t) \;=\; \omega\,\frac{x(t)\cos\omega\Delta - x(t-\Delta)}{\sin\omega\Delta}, \tag{17}$$

exactly, at every Δ with $\sin\omega\Delta \neq 0$. This is not a finite difference: its weights are flow-adapted, its error does not grow with Δ, and it fails precisely at the resonances $\omega\Delta = k\pi$, where the delay pair ceases to identify the state and $\alpha_\Delta \to 0$. Undersampling harms reconstruction through observability, not truncation. (Here $m = 2$ sits below the generic Takens bound; the theorem's conditions are sufficient, not necessary.)

Proposition S3.5 (Representation classes)**.** *Let G_k be as in Proposition S3.1. Then:*

(i) By the Kolmogorov superposition theorem (Kolmogorov, 1957; Arnold, 2009), every continuous function on a compact $K \subset \mathbb{R}^m$ can be written $\sum_{q=1}^{2m+1} \Phi_q\big(\sum_{p=1}^{m} \phi_{qp}(y_p)\big)$—the two-layer architecture with inner width $2m+1$. The classical inner functions are only continuous; smooth spline parameterizations of both layers are instead dense in $C(K)$ (Lai & Shen, 2021), so the inner width is a capacity hyperparameter rather than the theorem's $2m+1$ (the probes below use width 12 for $m = 7$). Hence every G_k of Proposition S3.1 lies in the closure of the compositional (two-layer) lift class.

(ii) The one-layer class is the additive family $\{\sum_p \phi_p(y_p)\}$: every C^2 member has identically vanishing mixed partials $\partial^2 F/\partial y_p\,\partial y_q$ $(p \neq q)$. The composition $G_k = (\mathcal{L}_f^k h)\circ\Phi^{-1}$ lies in this class only when all its mixed partials vanish identically on $\Phi(\mathcal{A})$, which for nonlinear f fails generically (it can hold in special cases: a scalar $\dot{x} = f(x)$ with $h = x$ gives $G_1(\mathbf{y}) = f(y_1)$, and for linear f Remark S3.4 gives an exactly linear G_k); Lemma S3.6 below computes the mixed partial in closed form for $\ddot{x} = g(x)$. The additive class is therefore a proper subclass; how far a given G_k lies from it is a per-system quantity, measured exactly in Table S6 (Remark S3.10).

(iii) For the compositional (two-layer) form $F_k(\mathbf{y}) = \sum_q \Psi_{kq}\big(\sum_i \phi_{qi}(y_i)\big)$, the Jacobian factorizes symbolically,

$$DF(\mathbf{y}) = A(\mathbf{y})\,B(\mathbf{y}), \qquad A_{kq} = \Psi'_{kq}\big(\textstyle\sum_i \phi_{qi}(y_i)\big), \qquad B_{qi} = \phi'_{qi}(y_i), \tag{18}$$

so the differential of the learned multivariate chart change is expressed entirely through derivatives of its univariate edges. With $U_{\mathbf{y}}$ an orthonormal basis of the tangent space $T_{\mathbf{y}}\Phi(\mathcal{A})$—estimable from data alone by local principal components—local invertibility of the learned map on the attractor is certified by $\sigma_{\min}\big(A(\mathbf{y})B(\mathbf{y})U_{\mathbf{y}}\big) > 0$, and its noise amplification by the tangent condition number $\kappa_F(\mathbf{y}) = \sigma_{\max}/\sigma_{\min}$: both are computable from the trained network with no ground truth.

The genericity in (ii) can be made exact in the simplest higher-order case, where the departure from additivity is computable in closed form and is proportional to the product of the nonlinearity and the sampling interval.

Lemma S3.6 (Non-separability of the window-to-velocity map)**.** *Let $\ddot{x} = g(x)$ with $g \in C^5(\mathbb{R})$, write $\varphi^s(x,v) = \big(X(s;x,v),\, V(s;x,v)\big)$ for its phase-space flow, and let $K \subset \mathbb{R}^2$ be a compact set of states (x,v) with nonempty interior. For $\Delta > 0$ define the two-sample delay map*

$$H_\Delta(x,v) = \big(x,\; X(-\Delta;x,v)\big).$$

Then there is $\Delta_0 > 0$ such that for every $\Delta \in (0,\Delta_0]$:

(i) H_Δ is a C^2 diffeomorphism from a neighborhood of K onto its image, so the window-to-velocity map $G_{1,\Delta} = \pi_v \circ H_\Delta^{-1}$ is well defined and C^2 on the open set $H_\Delta(\operatorname{int} K)$;

(ii) uniformly for $\mathbf{y} = (y_1, y_2) \in H_\Delta(K)$,

$$\frac{\partial^2 G_{1,\Delta}}{\partial y_1\,\partial y_2}(\mathbf{y}) = \frac{\Delta}{12}\,g''(y_1) + O(\Delta^2); \tag{19}$$

(iii) if $U \subset \operatorname{int} K$ is open and g'' has no zero on the x-projection of U, then, after shrinking Δ_0, $G_{1,\Delta}$ is not additively separable on $H_\Delta(U)$: no functions ϕ_1, ϕ_2 satisfy $G_{1,\Delta}(\mathbf{y}) = \phi_1(y_1) + \phi_2(y_2)$ there;

(iv) if g is affine, then $G_{1,\Delta}$ is exactly affine, hence additively separable, on $H_\Delta(K)$;

(v) $G_{1,\Delta}$ admits an additive approximation on $H_\Delta(K)$ with sup-norm error $O(\Delta^2)$, namely $\mathbf{y} \mapsto (y_1 - y_2)/\Delta + \frac{\Delta}{2} g(y_1)$.

Proof. Step 1: expansion of the delay map. Along a solution, $\ddot{x} = g$, $\dddot{x} = g'\dot{x}$, $x^{(4)} = g''\dot{x}^2 + g'g$, and $x^{(5)} = g'''\dot{x}^3 + 3g''g\,\dot{x} + (g')^2\dot{x}$. Taylor's theorem in s with integral remainder gives

$$\begin{aligned} X(-\Delta; x, v) &= x - \Delta\, Q(x, v, \Delta), \\ Q &= v - \frac{\Delta}{2} g(x) + \frac{\Delta^2}{6} g'(x)\, v \\ &\quad - \frac{\Delta^3}{24}\big(g''(x)v^2 + g'(x)g(x)\big) + \Delta^4 S(x, v, \Delta) \end{aligned} \tag{20}$$

where $S(x, v, \Delta) = \int_0^1 \frac{(1-\theta)^4}{4!}\, P\big(\varphi^{-\theta\Delta}(x, v)\big)\, d\theta$ with $P(x, v) = g'''(x)v^3 + 3g''(x)g(x)v + g'(x)^2 v$. Since $g \in C^5$, $P \in C^2(\mathbb{R}^2)$ and the flow is C^5, so S and its first two derivatives in (x, v) are bounded uniformly on $K' \times [0, \Delta_0]$ for any compact neighborhood K' of K.

Step 2: H_Δ is a diffeomorphism onto its image. From (20), $\partial_v Q = 1 + \frac{\Delta^2}{6} g'(x) - \frac{\Delta^3}{12} g''(x)v + \Delta^4 \partial_v S = 1 + O(\Delta^2)$ uniformly on K', so for Δ_0 small $\partial_v Q \geq \frac{1}{2}$ on $K' \times (0, \Delta_0]$. Hence for each fixed x the map $v \mapsto X(-\Delta; x, v) = x - \Delta Q$ is strictly decreasing on the v-slice of K', which makes H_Δ injective on K', and $\det DH_\Delta = -\Delta\, \partial_v Q \neq 0$. Thus H_Δ is a C^2 diffeomorphism of K' onto its image, and $H_\Delta(\operatorname{int} K)$ is open. This proves (i).

Step 3: inversion. For $\mathbf{y} \in H_\Delta(K')$ put $q = (y_1 - y_2)/\Delta$; then $q = Q(y_1, v, \Delta)$ with $v = G_{1,\Delta}(\mathbf{y})$. Because $\partial_v Q \geq \frac{1}{2}$, the implicit function theorem (with Δ as a parameter) yields $v = \mathcal{V}(y_1, q, \Delta)$, C^2 in (y_1, q) with derivatives bounded uniformly in $\Delta \in (0, \Delta_0]$ on the compact set $\mathcal{K} = \{(y_1, q)\}$ induced by K' (note $q = v + O(\Delta)$, so $\mathcal{K}$ is bounded). Let

$$v_3(y_1, q, \Delta) = q + \frac{\Delta}{2} g(y_1) - \frac{\Delta^2}{6} g'(y_1)\, q + \frac{\Delta^3}{24} g''(y_1)\, q^2 - \frac{\Delta^3}{24} g'(y_1) g(y_1)$$

be the third fixed-point iterate of $v = q + \frac{\Delta}{2} g - \frac{\Delta^2}{6} g' v + \frac{\Delta^3}{24}(g''v^2 + g'g) - \Delta^4 S(y_1, v, \Delta)$, and set $w = \mathcal{V} - v_3$. Substituting $\mathcal{V} = v_3 + w$ into this equation and collecting terms gives

$$\Big(1 + \frac{\Delta^2}{6} g'(y_1) - \frac{\Delta^3}{24} g''(y_1)(\mathcal{V} + v_3)\Big)\, w = \Delta^4\, T(y_1, q, \Delta),$$

where T collects $S(y_1, \mathcal{V}, \Delta)$ together with the $O(\Delta^4)$ terms produced by the truncation of v_3; T is a composition of C^2 functions with uniform bounds, so T and its first two (y_1, q)-derivatives are uniformly bounded. Dividing by the bracket, which is $1 + O(\Delta^2)$, yields

$$G_{1,\Delta}(\mathbf{y}) = q + \frac{\Delta}{2} g(y_1) - \frac{\Delta^2}{6} g'(y_1)\, q + \frac{\Delta^3}{24} g''(y_1)\, q^2 - \frac{\Delta^3}{24} g'(y_1) g(y_1) + \Delta^4 R(y_1, q, \Delta), \tag{21}$$

with $R \in C^2$ in (y_1, q) and R, ∂R, $\partial^2 R$ uniformly bounded on $\mathcal{K} \times (0, \Delta_0]$. Rewriting in (y_1, y_2),

$$\begin{aligned} G_{1,\Delta}(\mathbf{y}) &= \frac{y_1 - y_2}{\Delta} + \frac{\Delta}{2} g(y_1) - \frac{\Delta}{6} g'(y_1)(y_1 - y_2) + \frac{\Delta}{24} g''(y_1)(y_1 - y_2)^2 \\ &\quad - \frac{\Delta^3}{24} g'(y_1) g(y_1) + \Delta^4 R\Big(y_1, \frac{y_1 - y_2}{\Delta}, \Delta\Big). \end{aligned} \tag{22}$$

Step 4: the mixed partial. The first, second, and fifth terms of (22) are additive and contribute nothing to $\partial^2_{y_1 y_2}$. The third and fourth contribute

$$\frac{\Delta}{6} g''(y_1) \quad \text{and} \quad -\frac{\Delta}{12} g''(y_1) - \frac{\Delta}{12} g'''(y_1)(y_1 - y_2),$$

respectively. Since K has bounded velocity, $y_1 - y_2 = \Delta q = O(\Delta)$ uniformly on $H_\Delta(K)$, so the last piece is $O(\Delta^2)$. For the remainder, the chain rule gives

$$\partial^2_{y_1 y_2}\Big[\Delta^4 R\Big(y_1, \tfrac{y_1-y_2}{\Delta}, \Delta\Big)\Big] = -\Delta^3\,\partial_{qy_1}R - \Delta^2\,\partial_{qq}R = O(\Delta^2),$$

using the uniform bounds on $\partial^2 R$. Summing, $\partial^2_{y_1 y_2} G_{1,\Delta} = \frac{\Delta}{6} g'' - \frac{\Delta}{12} g'' + O(\Delta^2) = \frac{\Delta}{12} g''(y_1) + O(\Delta^2)$ uniformly on $H_\Delta(K)$, which is (ii).

Step 5: consequences. (iii) If $|g''| \geq c > 0$ on the x-projection of an open $U \subset \operatorname{int} K$, then by (ii) there is Δ_0' such that $|\partial^2_{y_1 y_2} G_{1,\Delta}| \geq c\Delta/24 > 0$ on the open set $H_\Delta(U)$ for $\Delta \leq \Delta_0'$. Any C^2 function of the form $\phi_1(y_1) + \phi_2(y_2)$ has identically vanishing mixed partial, and if $G_{1,\Delta}$ agreed with such a sum on an open set the sum would be C^2 there; hence no additive representation exists on $H_\Delta(U)$. (iv) If $g(x) = ax + b$, the vector field is affine, so φ^s, H_Δ, and H_Δ^{-1} (on the image, where it exists by (i)) are affine, and $G_{1,\Delta} = \pi_v \circ H_\Delta^{-1}$ is affine. (v) In (21) the first two terms are additive in (y_1, y_2), while q, g and its derivatives, and R are uniformly bounded on $\mathcal{K}$, so the remaining terms are uniformly $O(\Delta^2)$ on $H_\Delta(K)$. □

Remark S3.7. The requirement $\operatorname{int} K \neq \emptyset$ in (iii) is not technical. On a curve $\{y_2 = h(y_1)\}$ every function is trivially additive ($\phi_1 = G \circ (\mathrm{id}, h)$, $\phi_2 = 0$), so on the delay image of a single closed orbit of the conservative oscillator $\ddot{x} = g(x)$ the additive class always suffices, whatever g is. The obstruction (19) is visible only when the training data cover an open region of phase space—an ensemble of energies or transients—which is the regime in which the lifts of this paper are trained.

On $\ddot{x} = -x^3$ ($g'' = -6y_1$) the formula predicts $-\Delta\, y_1/2$; numerically inverting the flow and fitting G_1 locally gives -0.049 at $(y_1, \Delta) = (1, 0.1)$ and -0.197 at $(1, 0.4)$ against -0.05 and -0.2. Lemma S3.6 is the local form of the regime map: the same nonlinearity that closes the equation forces the lift out of the additive class, at a rate set by Δ, and Lemma S3.8 is the $\Delta \to 0$ limit in which that departure disappears.

Lemma S3.8 (The finite-difference regime)**.** *If $x \in C^{p+1}$ with $|x^{(p+1)}| \leq M$ on the window span, degree-p polynomial interpolation on the window yields finite-difference weights $c_i^{(k)}$ with $\big|x^{(k)}(t) - \sum_i c_i^{(k)}\, x(t - i\tau)\big| \leq C_{p,m}\, M\, \tau^{\,p+1-k}$. Hence the distance from G_k to the linear class—and a fortiori to the additive and compositional classes—is $O(\tau^{\,p+1-k})$ as $\tau \to 0$: at dense sampling all three classes are near-optimal, and nothing is gained from nonlinearity or depth.*

Definition S3.9 (Undersampled time series)**.** A time series sampled at interval τ is undersampled for order k when the truncation bound of Lemma S3.8 is vacuous over any usable window—τ a non-negligible fraction of the dynamics' characteristic time—while the delay map remains an embedding with conditioning α_Δ bounded away from zero (Proposition S3.3). In this regime G_k exists and is stably invertible on the attractor, but only as a genuine composition: it is far from the linear and additive classes, and only the compositional class is guaranteed to contain it (Proposition S3.5). Reconstruction fails not when τ is large per se, but when $\alpha_\Delta \to 0$ or the delay map loses injectivity (Remark S3.4). The *aliasing wall* is the sampling interval at which the second condition fails: $\alpha_\Delta \to 0$ or Φ ceases to be injective (Remark S3.4).

Remark S3.10 (Certificates and scope)*.* A general lower bound on the additive distance is delicate—$\Phi(\mathcal{A})$ is a d-dimensional submanifold of $\mathbb{R}^m$, so mixed-partial arguments do not apply off the image—but it is measurable per system: with fixed spline knots the additive model is linear in its parameters, its squared loss is convex, and the fitted additive error in Table S6 is therefore (up to knot resolution and optimization tolerance) the class distance rather than a training artifact. The linear rows are exact class optima by construction (ridge regression). Three conventions and caveats attach to the stability statements. Norms and singular values are taken with per-order-normalized outputs and the ambient-induced metric on $\mathcal{A}$; with heterogeneous derivative scales the conditioning claims hold after this normalization, which the experiments apply throughout. The perturbation bound (16) concerns the learned map on a tubular neighborhood: ambient noise moves the delay vector off $\Phi(\mathcal{A})$, so the off-manifold extension of $\widehat{F}$—shaped in practice by training on noisy inputs, as in the field experiment below—is part of the estimator, not an afterthought. Finally, the propositions assume a smooth autonomous flow on a compact invariant manifold: for fractal attractors the smooth-manifold hypotheses should be replaced by the box-dimension embedding theory of Sauer et al. (1991), under which Φ^{-1} is guaranteed continuous rather than smooth; non-smooth dynamics (the Bouc–Wen memory law) and forced systems (extend the state by the forcing phase) require the correspondingly extended hypotheses.

Existence and Kolmogorov–Arnold structure of the delay-to-jet map. Proposition S3.1 and Corollary S3.2 establish that the lift exists and is a change of chart, and Lemma S3.6 shows that a one-layer (additive) KAN is generically too small to represent it. The result below closes the argument from the other side, sharpening Proposition S3.5(i). It shows that the delay-to-jet map exists under injectivity of the delay map alone, that it is the transition map between the delay and differential embeddings, that the Kolmogorov–Arnold theorem makes this map, for all derivative orders at once, exactly a two-layer KAN with a shared inner layer, and that two-layer spline KANs approximate it uniformly. The density statement is derived directly from the Kolmogorov–Arnold representation rather than assumed.

Setting. Let φ^t be the flow of $\dot{s} = f(s)$ on a manifold M with $f \in C^p$, let $\mathcal{A} \subset M$ be a compact invariant set of box-counting dimension d, and let $h \in C^p(M, \mathbb{R})$ be the observable, $x(t) = h(s(t))$. With the delay map Φ of Proposition S3.1 and the differential map $\Theta(s) = \big(h(s), (\mathcal{L}_f h)(s), \ldots, (\mathcal{L}_f^p h)(s)\big)$ of Corollary S3.2, we have $x^{(k)}(t) = (\mathcal{L}_f^k h)(s(t))$, since $\frac{d^k}{d\sigma^k} h(\varphi^\sigma s)\big|_{\sigma=0} = (\mathcal{L}_f^k h)(s)$. Write $\mathbf{y}_t = \Phi(s(t))$. We use two hypotheses:

- (H1) Φ is injective on $\mathcal{A}$. This holds for generic (prevalent) (h, τ) when $m \geq 2d+1$ for a smooth $\mathcal{A}$ (Takens, 1981), and when $m > 2d$ for a fractal $\mathcal{A}$ (Sauer et al., 1991).
- (H2) Θ is injective on $\mathcal{A}$ (generic for $p \geq 2d$, Corollary S3.2). Used only in part (ii).

Unlike Proposition S3.1, part (i) below requires only (H1), not that Φ be an embedding, and so covers fractal attractors without modification (cf. Remark S3.10).

Proposition S3.11 (Kolmogorov–Arnold structure of the delay-to-jet map)**.**

(i) Existence. *Under (H1) there is a unique continuous map $G = (G_0, \ldots, G_p) : \Phi(\mathcal{A}) \to \mathbb{R}^{p+1}$ with*

$$G \circ \Phi = \Theta, \qquad \textit{i.e.} \qquad x^{(k)}(t) = G_k(\mathbf{y}_t) \quad \textit{for all } t,\ k = 0, \ldots, p. \tag{23}$$

(ii) Chart transition. *Under (H1) and (H2), $G : \Phi(\mathcal{A}) \to \Theta(\mathcal{A})$ is a homeomorphism that conjugates the flow in delay coordinates to the flow in jet coordinates,*

$$G \circ \big(\Phi\, \varphi^t\, \Phi^{-1}\big) = \big(\Theta\, \varphi^t\, \Theta^{-1}\big) \circ G. \tag{24}$$

If $\mathcal{A}$ is a smooth manifold and Φ, Θ are embeddings, G is a diffeomorphism, as in Proposition S3.3(i).

(iii) Exact two-layer representation with a shared inner layer. *There exist a cube $K = [a, b]^m \supset \Phi(\mathcal{A})$ and continuous functions $\psi_{q,i} : [a, b] \to \mathbb{R}$ and $\Psi_{k,q} : \mathbb{R} \to \mathbb{R}$ such that for all $\mathbf{y} \in \Phi(\mathcal{A})$,*

$$G_k(\mathbf{y}) = \sum_{q=0}^{2m} \Psi_{k,q}\Big(\sum_{i=1}^{m} \psi_{q,i}(y_i)\Big), \qquad k = 0, \ldots, p. \tag{25}$$

The inner functions $\psi_{q,i}$ do not depend on k; only the outer layer depends on the derivative order. The whole jet map is therefore a single two-layer KAN of shape $m \to 2m+1 \to p+1$.

(iv) Approximation by spline KANs. *For every $\varepsilon > 0$ there is a two-layer KAN with spline edges and a shared inner layer, $\hat{G}_k(\mathbf{y}) = \sum_q \hat{\Psi}_{k,q}\big(\sum_i \hat{\psi}_{q,i}(y_i)\big)$, such that*

$$\sup_{\mathbf{y} \in \Phi(\mathcal{A})} \max_{0 \leq k \leq p} \big|\hat{G}_k(\mathbf{y}) - G_k(\mathbf{y})\big| < \varepsilon. \tag{26}$$

Proof. We prove each part in turn.

(i) By (H1), $\Phi : \mathcal{A} \to \Phi(\mathcal{A})$ is a continuous bijection from a compact space onto a Hausdorff space, hence a homeomorphism, and $\Phi^{-1} : \Phi(\mathcal{A}) \to \mathcal{A}$ is continuous. Set $G := \Theta \circ \Phi^{-1}$. Each component $G_k = (\mathcal{L}_f^k h) \circ \Phi^{-1}$ is continuous because $\mathcal{L}_f^k h$ is continuous for $f, h \in C^p$ and $k \leq p$, and

$$x^{(k)}(t) = (\mathcal{L}_f^k h)(s(t)) = (\mathcal{L}_f^k h)\big(\Phi^{-1}(\mathbf{y}_t)\big) = G_k(\mathbf{y}_t).$$

Any G' with $G' \circ \Phi = \Theta$ equals $\Theta \circ \Phi^{-1}$ on $\Phi(\mathcal{A})$, which gives uniqueness. No condition on τ enters beyond (H1).

(ii) Under (H2), Θ is likewise a homeomorphism onto $\Theta(\mathcal{A})$, so $G = \Theta \circ \Phi^{-1}$ is a homeomorphism with inverse $\Phi \circ \Theta^{-1}$. The conjugacy follows by substitution:

$$G \, \Phi \, \varphi^t \, \Phi^{-1} = \Theta \, \varphi^t \, \Phi^{-1} = \big(\Theta \, \varphi^t \, \Theta^{-1}\big)\big(\Theta \, \Phi^{-1}\big) = \big(\Theta \, \varphi^t \, \Theta^{-1}\big) \, G.$$

In the smooth case, compositions and inverses of embeddings onto their images are diffeomorphisms.

(iii) *Step 1: extension to a cube.* $\Phi(\mathcal{A})$ is compact, so it lies in some cube $K = [a, b]^m$ and is closed in K. By the Tietze extension theorem each G_k extends to a continuous $\tilde{G}_k : K \to \mathbb{R}$ agreeing with G_k on $\Phi(\mathcal{A})$. This step is needed because the Kolmogorov–Arnold theorem applies to continuous functions on a cube, whereas G_k is defined only on the (typically lower-dimensional, possibly fractal) set $\Phi(\mathcal{A})$. The extension is not unique, but every extension agrees with G_k wherever data can occur.

Step 2: Kolmogorov's theorem. After rescaling K to $[0, 1]^m$, the Kolmogorov superposition theorem (Kolmogorov, 1957; Arnold, 2009), in its form with universal inner functions, provides continuous $\psi_{q,i}$ ($q = 0, \ldots, 2m$, $i = 1, \ldots, m$), depending only on m, such that every continuous g on $[0, 1]^m$ admits $g(\mathbf{y}) = \sum_{q=0}^{2m} \chi_q\big(\sum_i \psi_{q,i}(y_i)\big)$ with continuous outer functions χ_q depending on g. Applying this to each $\tilde{G}_k$ with the same $\psi_{q,i}$ yields outer functions $\Psi_{k,q}$; restricting to $\Phi(\mathcal{A})$ gives (25).

(iv) Let $n_h = 2m + 1$ and $s_q(\mathbf{y}) = \sum_i \psi_{q,i}(y_i)$, whose range over K is a compact interval I_q; let I_q^1 denote I_q enlarged by 1 on each side.

Tolerance for the inner layer. Each $\Psi_{k,q}$ is uniformly continuous on I_q^1, so there is $\delta \in (0, 1)$ such that for all k, q,

$$|u - v| < \delta \implies |\Psi_{k,q}(u) - \Psi_{k,q}(v)| < \frac{\varepsilon}{2n_h}.$$

Inner layer. Piecewise-linear interpolants, and hence splines of degree d are dense in $C[a, b]$. Choose splines $\hat{\psi}_{q,i}$ with $\|\hat{\psi}_{q,i} - \psi_{q,i}\|_\infty < \delta/m$. Then $\hat{s}_q = \sum_i \hat{\psi}_{q,i}$ satisfies $|\hat{s}_q - s_q| < \delta$, so $\hat{s}_q(\mathbf{y}) \in I_q^1$.

Outer layer. Choose splines $\hat{\Psi}_{k,q}$ on I_q^1 with $\|\hat{\Psi}_{k,q} - \Psi_{k,q}\|_{\infty, I_q^1} < \varepsilon/(2n_h)$.

Combination. For $\mathbf{y} \in \Phi(\mathcal{A})$,

$$\big|\hat{\Psi}_{k,q}(\hat{s}_q) - \Psi_{k,q}(s_q)\big| \le \underbrace{\big|\hat{\Psi}_{k,q}(\hat{s}_q) - \Psi_{k,q}(\hat{s}_q)\big|}_{< \varepsilon/2n_h} + \underbrace{\big|\Psi_{k,q}(\hat{s}_q) - \Psi_{k,q}(s_q)\big|}_{< \varepsilon/2n_h}.$$

Summing over the n_h hidden units gives $|\hat{G}_k - G_k| < \varepsilon$. The inner splines were chosen independently of k, so the inner layer remains shared.

□

Corollary S3.12 (Lift error propagates linearly to the discovered law)**.** *Suppose the system is nth-order in the observable, $n \le p$, so that Eq.* (1) *holds, $x^{(n)} = F\big(x, x', \ldots, x^{(n-1)}\big)$, with a continuous closure F (namely $F = (\mathcal{L}_f^n h) \circ \Theta_{n-1}^{-1}$ when the $(n-1)$-jet map Θ_{n-1} is injective on $\mathcal{A}$; this is the closure of the companion form in Proposition S3.3(ii)). If F is L-Lipschitz and the lift satisfies $\|\hat{G} - G\|_\infty < \varepsilon$ on $\Phi(\mathcal{A})$, then along every trajectory*

$$\Big|\hat{G}_n(\mathbf{y}_t) - F\big(\hat{G}_0(\mathbf{y}_t), \ldots, \hat{G}_{n-1}(\mathbf{y}_t)\big)\Big| \le \big(1 + L\sqrt{n}\big)\, \varepsilon. \tag{27}$$

Proof. Insert $G_n(\mathbf{y}_t) = F\big(G_0(\mathbf{y}_t), \ldots, G_{n-1}(\mathbf{y}_t)\big)$ and apply the triangle inequality. The first term is bounded by ε; the second by $L \, \big\|(\hat{G}_k - G_k)_{k<n}\big\|_2 \le L\sqrt{n}\, \varepsilon$. □

Remark S3.13 (Interpretation). Proposition S3.11 supplies the logical chain behind the lifting layer. *Existence* (i) makes the lift well-posed at any τ for which (H1) holds; where (H1) fails, no architecture can succeed, which is the content of the aliasing wall of Definition S3.9 (Table S6 at $\Delta t = 0.8$) and of the Bouc–Wen memory variable seen through $(x, \dot{x})$ (Note S2). *Chart transition* (ii) shows that the flow read in jet coordinates is conjugate to the observed flow, and in those coordinates the vector field reduces to the single closure of Proposition S3.3(ii); this is what licenses reading the equation layer's edges as the governing law. *Representation* (iii) shows that depth two is sufficient for every system satisfying (H1), whatever the nonlinearity or sampling rate, and that one inner layer can serve all $p + 1$ derivative orders, which is the jet operator of Eq. (8) in its compositional (two-layer) form. Lemma S3.6 supplies the complementary lower bound by showing that a single additive layer can be insufficient. Thus, two layers are sufficient in general and become necessary within the nonseparable regime identified by the lemma. Corollary S3.12 connects lift accuracy to the residual of the discovered law.

Remark S3.14 (Scope). The inner functions of the Kolmogorov–Arnold theorem are continuous in general, but far from smooth and often "Cantor-like" (Girosi & Poggio, 1989). Part (iii) is therefore an existence statement about architecture; part (iv) is what a trained spline KAN realizes, and it inherits no convergence rate from the theorem. Rates must come from the smoothness of G (Proposition S3.3) and from spline approximation theory (Lai & Shen, 2021). In particular, the width used in the experiments (12 for $m = 7$) is a capacity choice, not the theorem's $2m + 1$ with its pathological inner functions, as noted in Proposition S3.5(i). Corollary S3.12 assumes a Lipschitz closure and exact targets; measurement noise enters separately through Proposition S3.3(iv).

From representation classes to measurement. The results above establish which model classes can represent the lift, but not how large the gaps between them are in practice. Proposition S3.11 is an existence statement with no rate (Remark S3.14), Lemma S3.6 is asymptotic in Δ, and the distance to the additive class can only be measured system by system (Remark S3.10). Taken together, however, they make three predictions that can be checked directly as the sampling interval grows:

(i) at dense sampling, the linear class is already near-exact and nothing is gained from nonlinearity or depth (Lemma S3.8);

(ii) in the undersampled regime, the linear and additive classes fall away while the compositional class remains accurate (Lemma S3.6, Proposition S3.11(iii));

(iii) beyond the aliasing wall, every class fails, because the window no longer determines the state (Definition S3.9).

We test these on the Lorenz system observed through $x(t)$. Its strong nonlinearity makes the departure from additivity large, and its hidden states supply exact derivative targets. Along the same sweep, we evaluate the conditioning certificate of Proposition S3.5(iii). We then repeat the comparison on a coarsely sampled, noisy sine-Gordon field, where the delay window becomes a space–time patch.

Protocol. The Lorenz system is observed through the scalar $x(t)$ alone (hidden y, z), noise-free, so the probe isolates representation capacity from estimation error. Targets are exact, $x' = \sigma(y - x)$ and $x'' = \sigma(\dot{y} - x')$, evaluated from the simulator's hidden states at the center of each window of $m = 7$ consecutive samples at spacing Δt. Three model classes fit the same windows: an optimal linear map (ridge regression on the window), a one-layer KAN (the additive class of the jet operator, Eq. (8)), and a two-layer KAN of hidden width 12. Training uses one trajectory; evaluation uses an independent trajectory from a different initial condition, both burned onto the attractor. KAN fits are repeated over three seeds and reported as the mean; the linear map is deterministic. Sweeping Δt from 0.01 to 0.80—from dense sampling, through roughly three samples per orbit turn at 0.25, to less than one at 0.80—moves the task from the finite-difference regime, through the flow-inversion regime, to the aliasing wall of Definition S3.9.

Results. Table S6 shows three regimes. At $\Delta t \leq 0.05$ the linear map is exact ($R^2 = 1.000$) and every class ties, confirming that with dense data the lift gains nothing from nonlinearity—differentiation is linear there, and the one-layer jet of Eq. (8) is the appropriate model. At $\Delta t = 0.25$ the ordering is graded and large: the linear map collapses, the additive class improves on it but plateaus, and the compositional map stays near-exact on both derivative orders. In residual terms, at $\Delta t = 0.25$ the two-layer map reduces $1 - R^2$ on x'' from 0.365 (linear) and 0.099 (additive) to

Δt	Target	linear	additive (1-layer)	compositional (2-layer)
0.01	x'	1.0000	0.9849 [0.9561, 0.9996]	1.0000 [1.0000, 1.0000]
	x''	1.0000	0.9721 [0.9692, 0.9737]	0.9992 [0.9991, 0.9993]
0.05	x'	1.0000	0.9998 [0.9997, 0.9999]	1.0000 [1.0000, 1.0000]
	x''	1.0000	0.9986 [0.9982, 0.9988]	1.0000 [0.9999, 1.0000]
0.15	x'	0.8967	0.9712 [0.9545, 0.9851]	0.9997 [0.9997, 0.9997]
	x''	0.9279	0.9756 [0.9690, 0.9877]	0.9994 [0.9992, 0.9996]
0.25	x'	0.4436	0.8986 [0.8463, 0.9326]	**0.9940** [0.9931, 0.9949]
	x''	0.6351	0.9008 [0.8512, 0.9867]	**0.9963** [0.9953, 0.9969]
0.40	x'	0.0642	0.5718 [0.5695, 0.5741]	**0.8810** [0.8761, 0.8852]
	x''	0.3428	0.9056 [0.8682, 0.9671]	**0.9699** [0.9675, 0.9722]
0.60	x'	0.0026	0.2062 [0.1970, 0.2126]	**0.6097** [0.5363, 0.7012]
	x''	0.2538	0.9216 [0.9004, 0.9593]	0.9315 [0.9227, 0.9362]
0.80	x'	−0.0266	−0.2900 [−0.3131, −0.2621]	0.2132 [0.1019, 0.3844]
	x''	0.2958	0.8852 [0.8632, 0.9273]	0.8675 [0.8496, 0.8864]

Table S6: Held-out R^2 for the window-to-jet map on the Lorenz observable versus sampling interval Δt ($m = 7$, noiseless, independent test trajectory). KAN entries are seed means with $[\min, \max]$ over three seeds; the linear map is deterministic. Dense sampling: the linear map is exact and all classes tie. Coarse sampling: the linear map collapses, the additive (one-layer) class plateaus, and only the compositional (two-layer) map remains near-exact—and is also the most seed-stable.

Δt	med. $\sigma_{\min}$	p5 $\sigma_{\min}$	med. κ_F	compositional R^2 (x')
0.01	0.62	0.38	7.3	1.000
0.05	0.62	0.36	1.6	1.000
0.15	0.41	0.08	2.5	1.000
0.25	0.39	0.12	3.2	0.994
0.40	0.51	0.12	4.0	0.881
0.60	0.63	0.16	4.5	0.610
0.80	0.73	0.16	4.2	0.213

Table S7: Tangent-space conditioning of the learned chart change (Proposition S3.5(iii)): median and 5th-percentile $\sigma_{\min}(J_F U_{\mathbf{y}})$ and median κ_F over 300 held-out windows, seed means, per-order-normalized units, tangent basis from local PCA ($k = 40$, $d = 2$). κ_F is U-shaped in Δt (delay redundancy at dense sampling, observability loss at coarse), while the invertibility certificate stays positive even where reconstruction fails.

0.0037—a factor of 27 over the additive class and nearly 100 over the linear map. The additive class beating the optimal linear map shows that nonlinearity already helps; only composition closes the gap, as Eq. (15) predicts. Extending the sweep locates the lower boundary: the compositional advantage persists through $\Delta t = 0.6$ (x': 0.61 versus 0.21 additive and 0.00 linear), and by $\Delta t = 0.8$—below one sample per orbit turn—every class fails on x' (best $R^2 = 0.21$): the aliasing wall of Definition S3.9, where no representation can recover information the samples no longer carry.

Conditioning of the learned chart change. Proposition S3.5(iii) makes the trained lift auditable. At 300 held-out windows we assemble the joint Jacobian of the fitted chart change—the exact order-0 projection row stacked with the analytic gradients of the fitted x' and x'' maps, in per-order-normalized units—restrict it to a tangent basis $U_{\mathbf{y}}$ estimated by local principal components, and report $\sigma_{\min}(J_F U_{\mathbf{y}})$ and κ_F (Table S7). Two findings. First, κ_F is U-shaped in Δt with its minimum near $\Delta t = 0.05$: at dense sampling the delay coordinates are nearly redundant—the classical too-short-delay degeneracy, α_Δ small—while at coarse sampling observability degrades, so noise amplification is governed at both extremes by the conditioning of the delay embedding, the mechanism of Proposition S3.3(iii). Second, the invertibility certificate stays bounded away from

Regime	Front end	NRMSE u_{xx}	NRMSE u_{tt}	$\hat{c}^2$	$\hat{s}$
coarse, 1% noise	analytic spline (oracle)	0.280	0.342	0.691	-0.514
	additive (1-layer)	0.099	0.283	0.954	-0.989
	compositional (2-layer)	$\mathbf{0.047}$	$\mathbf{0.087}$	$\mathbf{0.985}$	$\mathbf{-0.970}$
dense, 1% noise	analytic spline (oracle)	0.072	0.147	0.977	-0.959
	additive (1-layer)	0.304	0.693	0.636	-0.547
	compositional (2-layer)	0.135	0.274	0.981	-0.973

Table S8: Sine-Gordon derivative accuracy and discovered coefficients (least squares on $[u_{xx}, \sin u]$; true $(c^2, s) = (1, -1)$) on the held-out velocity, means over three seeds (noise realization and network initialization vary jointly). Coarse: 80×33 samples; dense: 400×257. The spline's smoothing is oracle-tuned per realization.

zero even at $\Delta t = 0.8$, where reconstruction of x' nonetheless fails. Local nonsingularity of the learned map is therefore necessary but not sufficient: at the aliasing wall the failure is global—the true chart change loses injectivity and the approximation term of Eq. (16) dominates—and a pointwise Jacobian audit cannot see it.

Scope. This is a supervised representation probe, in the same sense as the Bouc–Wen state probe of Note S2: targets use the simulator's hidden states, so the experiment measures what each class can represent. It identifies the two regimes in which a compositional lift is warranted—undersampled observations, where the window must invert the flow, and hidden non-derivative states, where the window must reconstruct a coordinate (Note S2)—and the regime in which the single-layer jet is already the right tool, namely densely sampled observables, which covers the calibration systems examined.

Field-side discovery under joint undersampling and noise (sine-Gordon). The same class separation governs field data, where the delay window becomes a space–time patch. Undersampling alone does not produce it: on the noiseless kink–antikink family, an interpolating tensor spline recovers u_{tt} to NRMSE 0.015 even from 33 frames, because piecewise-quintic interpolation converges rapidly on an analytic field. Noise closes that escape: it forbids interpolation, and the smoothing bias–variance tradeoff then binds every linear front end. We therefore test discovery of $u_{tt} = c^2 u_{xx} + s \sin(u)$ (true $c^2 = 1$, $s = -1$) on a coarse, noisy field: 80 spatial points ($\Delta x = 0.46$ against a kink width of ≈ 0.75), 33 frames ($\Delta t = 0.5$), and 1% Gaussian noise. Patch operators (5×5 coarse samples $\to (u, u_{xx}, u_{tt})$ at the patch center; one- versus two-layer KAN) are calibrated on exact solutions at pooled velocities $v \in \{0.40, 0.45, 0.50\}$ observed through the same coarse noisy grid, and deployed on the unseen velocity $v = 0.55$ with fresh noise; exact solution values are used for scoring only. The classical comparator is the paper's own analytic-spline jet (Eq. (6)) fit to the coarse noisy field, with its smoothing parameter swept against ground truth on an oracle-tuned linear baseline.

Table S8 and Fig. S10 show three things. First, on the coarse noisy field the oracle-tuned spline fails discovery: errors-in-variables attenuation from its derivative estimates (NRMSE 0.28–0.34) biases the recovered law by 31% on c^2 and 49% on s, and no smoothing choice repairs this. Both learned patch operators restore discovery; the compositional map has the cleanest derivatives (u_{tt} NRMSE 0.087, three times below the additive map and roughly four times below the oracle spline) and the tightest, most seed-stable coefficients. With the library-free equation layer the compositional front end recovers, on its best seed, $\hat{c}^2 = 1.00$ and $\hat{s} = -0.95$ with the u-edge matching $-\sin(u)$ at correlation 0.999 (seed-mean correlation 0.980; the spread comes from the equation-layer fit, whose collinear-edge variance is a back-end property independent of the front end). Second, the additive map survives on coefficients ($\hat{c}^2 = 0.95$) despite its three-fold worse derivative error: with dense-enough spatial context, u_{tt} can be written as a fixed linear combination of the patch samples plus a univariate function of u, which is additive, and its residual errors decorrelate from the regressors. The separation between the classes is therefore carried by derivative fidelity and stability rather than by outright failure—weaker than the scalar case, as the additive route predicts. Third, the dense control flips the ordering: with 257 frames the global spline is the best front end (u_{tt} NRMSE 0.147) and the small-patch operators are handicapped by their receptive field under noise (the additive map fails outright). The regime map runs in both directions—global linear smoothing is the right tool for dense

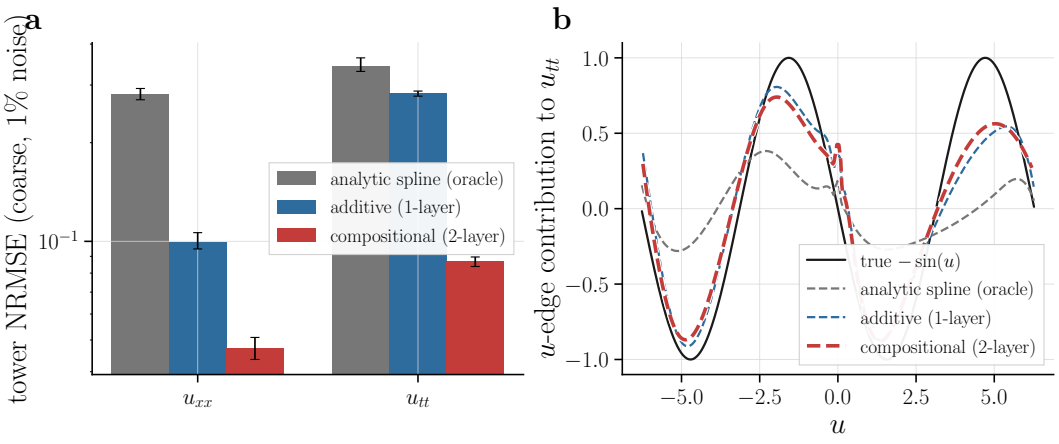

Figure S10: Coarse, noisy sine-Gordon. (a) Derivative NRMSE per front end (bars: seed mean; whiskers: min–max). (b) The library-free u-edge of the equation layer against the true $-\sin(u)$, per front end (first seed).

noisy fields, and the compositional patch operator is the only front end that keeps discovery accurate on coarse noisy ones.

S4 Controlled comparison on the Genesio–Tesi system: front ends across sampling interval

The main-text benchmarks vary the derivative estimate and the equation representation together. This note separates them on the third-order Genesio–Tesi system $\dddot{x} = -a\ddot{x} - b\dot{x} - cx + qx^2$ ($a = 1.2$, $b = 2.92$, $c = 6$, $q = 1$), observed through x alone, as the sampling interval Δ is coarsened from 0.05 to 0.9 (the attractor's half-period is ≈ 1.9, so $\Delta = 0.9$ is about two samples per swing). The protocol has a dense calibration trajectory (one initial condition, $\Delta t = 0.01$, 600 time units) whose derivative targets are Savitzky–Golay estimates of the observable itself—no simulator state enters—and a coarse deployment trajectory from a different initial condition on which only samples at spacing Δ exist and discovery is run; exact derivatives are used for scoring only. Five front ends produce $(\hat{x}', \hat{x}'', \hat{x}''')$ on the deployment trajectory from windows of $m = 7$ coarse samples: Savitzky–Golay on the coarse samples with window and polynomial order tuned per derivative against ground truth (an oracle-tuned fixed formula); the optimal linear map (ridge regression on the window); the additive class solved exactly (least squares on a per-coordinate cubic B-spline basis, no optimizer); an additive (one-layer) KAN trained by Adam; and a compositional (two-layer) KAN of hidden width 12. The deployment trajectory is noise-free; the comparison isolates sampling interval, not noise. Three backends consume each front end's derivatives unchanged: least squares on the oracle library $[x, \dot{x}, \ddot{x}, x^2]$, STLSQ on the generic degree-two polynomial library in $(x, \dot{x}, \ddot{x})$ (ten terms, threshold 0.3), and the library-free KANDy equation layer over $(x, \dot{x}, \ddot{x})$. KAN fits use three seeds; the linear and exact-additive maps are deterministic.

Table S9 and Fig. S11 give three results. First, the front end dominates: at $\Delta = 0.9$ every learned lift, including the exactly solved additive class, recovers all four coefficients to within 1% from the same data on which the oracle-tuned fixed formula is wrong by 82% and sparse regression loses three terms. Transferring the delay-to-derivative map from a densely sampled calibration trajectory to a coarsely sampled deployment trajectory is exactly what a learned lift can do and a fixed formula cannot. Second, with derivatives held fixed, the sparse-regression backends agree: on a polynomial, correctly specified law, the oracle least squares and generic STLSQ recover the same equation from the same lifted derivatives. The library-free KANDy layer recovers the same four terms but with a seed spread of up to 14% on the stiffness a and 11% on c at $\Delta = 0.9$ (Fig. S11b shows the first seed), from collinearity among the three derivative edges—a back-end property independent of the front end, and the reason the coefficient rows of Table S9 use the oracle least squares. Third, depth is not needed here: the exact-additive lift ties the compositional one at every Δ. This is the prediction of

front end	sampling interval Δ 0.05	0.1	0.2	0.4	0.6	0.9
held-out R^2 of $\hat{x}'''$						
Savitzky–Golay (oracle-tuned)	1.000	1.000	1.000	0.996	0.959	0.788
linear (ridge on window)	1.000	1.000	1.000	1.000	0.994	0.879
additive, exact LS	1.000	1.000	1.000	1.000	1.000	1.000
additive, one-layer KAN	0.987	0.996	1.000	1.000	1.000	1.000
compositional, two-layer KAN	0.999	1.000	1.000	1.000	1.000	1.000
largest relative coefficient error over (a, b, c, q), oracle-library least squares						
Savitzky–Golay (oracle-tuned)	0.000	0.001	0.011	0.131	0.412	0.819
linear (ridge on window)	0.000	0.000	0.000	0.002	0.034	0.509
additive, exact LS	0.000	0.000	0.000	0.000	0.000	0.000
additive, one-layer KAN	0.064	0.016	0.005	0.001	0.002	0.009
compositional, two-layer KAN	0.002	0.001	0.000	0.000	0.000	0.000
STLSQ on the generic degree-two library: terms missed / spurious						
Savitzky–Golay (oracle-tuned)	0/0	0/0	0/0	0/0	0/0	3/0
all learned front ends	0/0	0/0	0/0	0/0	0/0	0/0

Table S9: Genesio–Tesi discovery from a coarsely sampled scalar: the front end is what fails. Seed means over three seeds for the KAN maps; the exact-additive and linear maps are deterministic. At $\Delta = 0.9$ the oracle-tuned Savitzky–Golay tower misses three of the four terms under STLSQ and its least-squares coefficients are wrong by up to 82%; every learned lift recovers all four terms, the exact-additive and compositional lifts to within 0.1%.

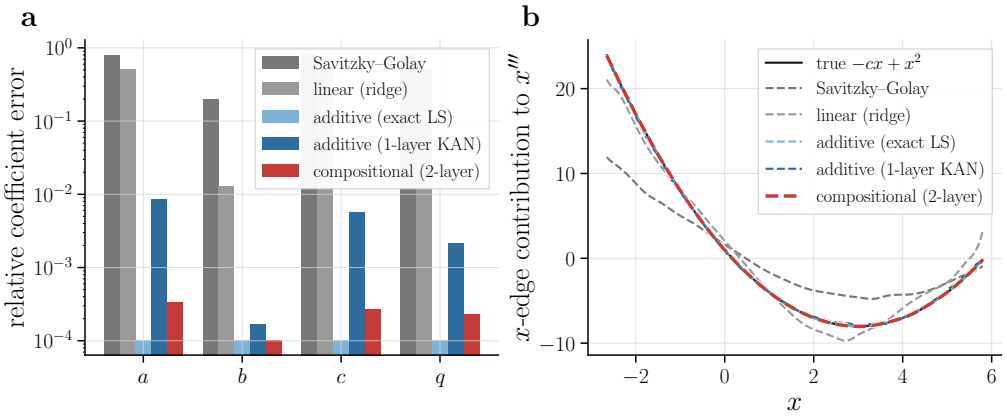

Figure S11: Genesio–Tesi at $\Delta = 0.9$. **a:** relative coefficient error (oracle-library least squares) per front end; bars floored at 10^{-4}. **b:** the library-free x-edge of the KANDy equation layer read off each front end's derivatives against the true $-cx + qx^2$; the learned lifts reproduce the quadratic, the fixed Savitzky–Golay formula does not.

Lemma S3.6 for a weakly nonlinear vector field—the only nonlinearity is qx^2, so $g'' = 2q$ is small, and the window-to-derivative map stays additively separable to well past the sampling interval at which fixed formulas fail (we find the exact-additive R^2 on x''' first drops below 0.999 at $\Delta \approx 2$, near the aliasing wall). The compositional lift earns its keep on the strongly nonlinear Lorenz observable of Table S6, where the same exact-additive solve reaches only $R^2 = 0.62$ on x' at $\Delta t = 0.40$ against 0.88 for the two-layer map. On x'', however, the exact-additive class is on par with the compositional one there (0.968 versus 0.970 at $\Delta t = 0.40$), so the depth requirement of Section 2.2 is a statement about the velocity map, which is what Lemma S3.6 addresses; Table S6's one-layer column is an Adam-trained KAN and sits 0.02–0.06 below the exactly solved class.

S5 Baseline configuration for the sine-Gordon benchmark

On the PDE benchmark of Section 3.4 we compare pointwise SINDy/PDE-FIND (Brunton et al., 2016; Rudy et al., 2017) and weak-form WSINDy (Messenger & Bortz, 2021a;b) on identical noisy data, both through their PySINDy implementations. SINDy/PDE-FIND estimates derivatives with a Savitzky–Golay filter and solves $u_{tt} = \Theta\boldsymbol{\xi}$ by sequential thresholded least squares (threshold 0.05 on normalized columns) over polynomial powers of u up to degree five together with u_{xx}; WSINDy avoids pointwise differentiation by integrating the equation against compactly supported test functions $\Psi(x,t)$ ($K = 2000$ test-function centres) and transferring derivatives onto Ψ. WSINDy is run with both a naive polynomial library and an oracle library containing $\sin(u)$. We also compare D-CIPHER (Kacprzyk et al., 2023), which searches for $\sum_j \eta_j \mathcal{O}_j u = g(\mathbf{x}, u)$ with scalar weights on the differential operators and only g discovered in closed form; since g may not depend on u_t, a saturating damping $\kappa \tanh(\nu u_t)$ lies outside its model class at any noise level, a structural ceiling measured in Note S1. All methods are scored on the recovered restoring force $R(u)$ against $-\sin(u)$ over the observed range, which separates derivative-noise error from library misspecification.

S6 Equation backends on identical derivatives: what the free-form edges adds

The sine-Gordon comparison of Section 3.4 pairs each method with its own derivative estimator. To isolate the equation representation, this note feeds every back end the same spline-jet samples (u, u_{xx}, u_t, u_{tt}) that the Higher-Order-KANDy equation layer consumes—the surrogate of Eq. (6) fitted once per noise realization—and changes only the read-out. Five back ends: the free-form additive KAN layer (the paper's method, $\lambda = 0$); PySINDy's STLSQ optimizer with the settings of the paper's baselines (threshold 0.05 on normalized columns) on $[1, u, u^2, u^3, \sin u, u_{xx}, u_t]$, i.e. sparse regression handed the $\sin(u)$ column; the same optimizer on the polynomial library of Table 1 without it; a SINDy-KAN (Howard et al., 2026); and a Symbolic–KAN (Faroughi et al., 2026). SINDy-KAN has no released code; the Symbolic–KAN authors released code after their preprint, which we also ran (below). The SINDy-KAN row re-implements the mechanism of their Section 3: a B-spline KAN is trained while each edge is simultaneously regressed by STLSQ onto the univariate library $[x, x^2, x^3, \sin x, \cos x]$ of its own input (their Eq. 13), with a trainable shadow coefficient matrix carrying an ℓ_1 penalty and the composite loss of their Eq. 16; the equation is the sparse per-edge representation read off the trained edges. The Symbolic–KAN row re-implements their Eqs. 5–13 and 22–29: units that apply one analytic primitive from $\{0, 1, x, x^2, x^3, \sin, \cos, \tanh, e^x, \log(1 + |x|)\}$, chosen by an annealed Gumbel-softmax gate with entropy regularization, to a learned scalar projection of the inputs, summed with sigmoid unit gates under a budgeted sparsity penalty and hardened to one-hot at the end. Support is scored as in Table 1: each method's own selection step (the STLSQ threshold for the library methods, activation pruning for the free-form layer, the per-edge sparse regression for the SINDy-KAN) followed by the 0.01 coefficient tolerance; the Symbolic–KAN's hardened network is not additive in the inputs, so it is scored on the restoring curve $u \mapsto f(u, \bar{u}_{xx}, \bar{u}_t)$ and the finite-difference slope in u_{xx} only. Noise $\{0, 1, 5, 10\}\%$, five independent seeds per level, $12{,}000$ interior collocation points, an independent noise field per (seed, level), as in the audit; the free-form layer here carries the three edges (u, u_{xx}, u_t) of Section 3.4's fit, without the two extra null controls of the audit.

Table S10 and Fig. S12 support two readings. First, the free-form edges add nothing to coefficient accuracy once sparse regression is handed the right column: the STLSQ oracle on the same derivatives is the better form estimator at every noisy level, as the WSINDy oracle is in Section 3.4. Its advantage is that it does not require the column—the full-period $-\sin(u)$ is recovered from a spline edge with nothing named in advance, whereas the polynomial library fails by coverage at zero noise. Second, the two KAN-based discovery architectures fail for reasons that follow from their design rather than their tuning. SINDy-KAN discovers the equation by snapping each edge to a univariate library; on a coordinate whose observed range is narrow, the library is collinear, and the snap is ill-conditioned, the same range argument that limits polynomial libraries on the u axis. Symbolic–KAN is a general symbolic regressor whose units act on learned projections of all inputs; nothing in it enforces that a term be a function of one derivative coordinate, so the additive-by-order structure that Higher-Order-KANDy imposes must instead be discovered, and on this problem is not. The comparison therefore does not show that one KAN parameterization is better than another; it shows that imposing one

noise	back end (all on the same spline jet)	form RMS	$\hat{c}$ (u_{xx})	exact support
0%	Higher-Order-KANDy (free-form edges)	0.002 ± 0.000	1.000	5/5
	STLSQ, $\sin(u)$ column supplied	0.000 ± 0.000	1.000	5/5
	STLSQ, polynomial library	0.707 ± 0.000	0.376	0/5
	SINDy-KAN (re-impl.)	0.029 ± 0.046	0.91 ± 0.26	0/5
	Symbolic–KAN (re-impl.)	0.723 ± 0.009	0.208	—
1%	Higher-Order-KANDy (free-form edges)	0.030 ± 0.013	0.993	5/5
	STLSQ, $\sin(u)$ column supplied	0.012 ± 0.006	0.990	5/5
	STLSQ, polynomial library	0.707 ± 0.000	0.377	0/5
	SINDy-KAN (re-impl.)	0.049 ± 0.040	0.32 ± 0.24	0/5
	Symbolic–KAN (re-impl.)	0.725 ± 0.007	0.201	—
5%	Higher-Order-KANDy (free-form edges)	0.095 ± 0.039	0.964	5/5
	STLSQ, $\sin(u)$ column supplied	0.089 ± 0.041	0.917	2/5
	STLSQ, polynomial library	0.706 ± 0.001	0.377	0/5
	SINDy-KAN (re-impl.)	0.105 ± 0.042	-0.44 ± 0.34	0/5
	Symbolic–KAN (re-impl.)	0.726 ± 0.013	0.203	—
10%	Higher-Order-KANDy (free-form edges)	0.230 ± 0.037	0.796	5/5
	STLSQ, $\sin(u)$ column supplied	0.170 ± 0.041	0.845	0/5
	STLSQ, polynomial library	0.706 ± 0.002	0.378	0/5
	SINDy-KAN (re-impl.)	0.243 ± 0.021	-0.73 ± 0.31	0/5
	Symbolic–KAN (re-impl.)	0.728 ± 0.015	0.165	—

Table S10: Equation back ends on identical spline-jet derivatives (sine-Gordon collision; mean $\pm$ std over five seeds; form RMS is the error of the recovered restoring curve against $-\sin(u)$ over the observed range; true $\hat{c} = 1$; support: each method's own selection step, then the 0.01 tolerance). Handed the $\sin(u)$ column, sparse regression matches or beats the free-form edge on form at every level, while its own selection retains a spurious linear u (-0.013 to -0.034) on three seeds at 5% and on every seed at 10%, as the weak-form oracle does in Table S11; without the column it fails by coverage at every level, including zero noise. The SINDy-KAN's per-edge symbolic library recovers $\sin(u)$ on the u edge (shape correlation $\geq$0.997 at every level) but its read-out of the u_{xx} edge is ill-conditioned—on the narrow u_{xx} range the columns x, x^3, $\sin x$, $\cos x$ are nearly collinear—so $\hat{c}$ changes sign across seeds and the support is never exact. The Symbolic–KAN fits the data (R^2 0.92–0.99) but, with dense projections mixing all three inputs, isolates no term.

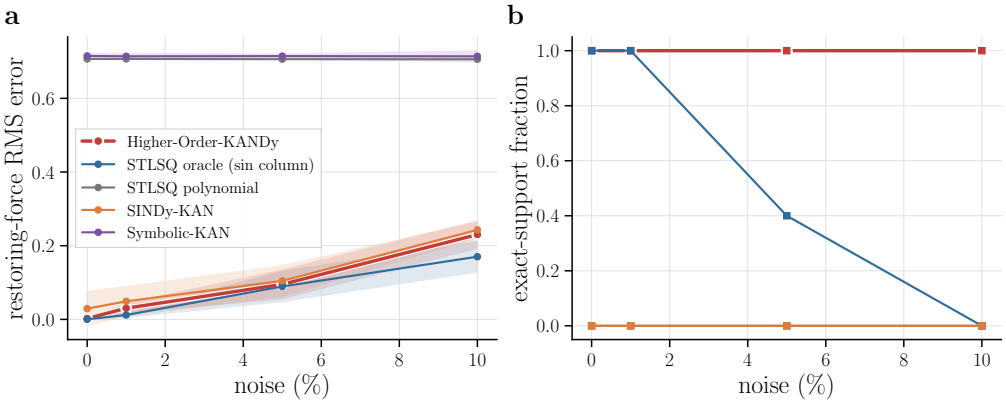

Figure S12: Back ends on identical derivatives. **a:** restoring-force RMS error versus noise (bands: $\pm$1 std over seeds). **b:** fraction of seeds with exact support (Symbolic–KAN omitted: non-additive).

univariate edge per derivative coordinate is the right prior for the class of equations this paper targets. Both re-implementations are ours and are labeled as such. We also ran the released Symbolic–KAN code[2] with only its dataset generator replaced by the same jet samples, at the function defaults (two blocks of six units, 2,000 Adam epochs followed by L-BFGS, default gate and unit-sparsity weights);

[2] `github.com/sfaroughi3/Pub_Symbolic_KANs`, `Exp_reaction_diffusion`

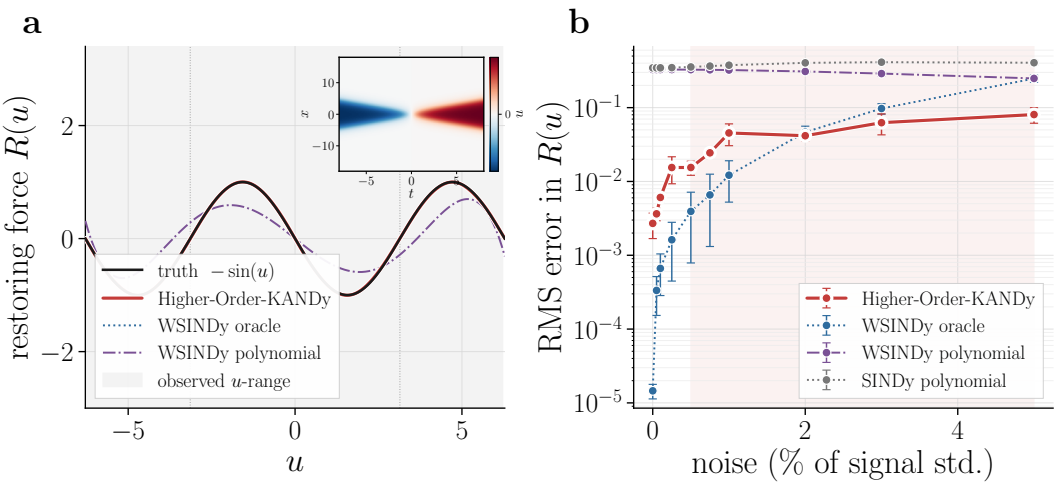

Figure S13: **Sine-Gordon kink–antikink collision: form recovery under noise. a:** the recovered restoring force $R(u)$ against $-\sin(u)$ over the observed u-range (shaded; inset: the collision field). The free-form Higher-Order-KANDy edge tracks the full-period $-\sin(u)$, while the polynomial-library recovery flattens outside a single period—the range limitation of Section 3.4, which no noise level fixes. **b:** RMS error of $R(u)$ versus noise: the WSINDy oracle, handed the exact $\sin(u)$ column, is the better form estimator at every noise level, the gap narrowing toward 5%.

its unit gates killed every unit at 1% and 5% noise (a constant output) and left one linear unit at 0% and 10% (R^2 0.25 and 0.40).

S7 Discovered equations, sine-Gordon benchmark

Table S11 provides the complete equations generated by each method applied to the kink-antikink collision equation, $u_{tt} = u_{xx} - \sin(u)$, at each of the twelve noise levels in the sweep found in Section 3.4. The equation shown is the representative repeat (the first seed). The support column counts how many of the five repeats selected that exact term set. Every selected term is shown, including spurious terms; a term counts for every method only if its coefficient in the u_{tt}- normalized equation exceeds 0.01—one percent of the unit-scale true coefficients—and smaller selected terms are omitted as numerically inactive. The correct column indicates the number of the two true terms $u_{xx}, \sin u$ that were recovered. All rows come from the symmetric audit protocol: the Higher-Order-KANDy equation layer carries five free-form edges—u, u_{xx}, and the null controls u_t, u_x, u_{xt}—with no parsimony penalty on any edge ($\lambda = 0$), an independent noise realization is drawn per (repeat, level), and the network initialization varies with the repeat seed. A λ-sensitivity audit confirms the support selection owes nothing to a sparsity prior—indeed the opposite: adding a uniform L1 penalty on all five edges destroys recovery rather than aiding it (at $\lambda = 0.1$ the discovered equation collapses to $u_{tt} = 0.23\,u_{xx}$ at every noise level tested, and at $\lambda = 0.3$ to $u_{tt} = 0.10\,u_{xx}$), so the active-set selection is a property of the learned free-form edges and the audit gates, not of a parsimony penalty. D-CIPHER is a reduced diagnostic (one seed, four noise levels, original protocol) re-scored at the same coefficient tolerance.

Table S11: Equations discovered for the sine-Gordon kink–antikink collision, $u_{tt} = u_{xx} - \sin(u)$, at every noise level of the symmetric audit protocol (noise to 10%; support = repeats out of five selecting exactly the displayed term set, which is the first seed's and coincides with the true set only where the spurious column is empty). Higher-Order-KANDy rows are set in bold; the grey D-CIPHER rows are the reduced one-seed diagnostic and carry no support count.

noise	method	discovered equation	correct	spurious	support
0%	**Higher-Order-KANDy**	$u_{tt} = -1.000\ \sin(u) + 1.001\ u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.000\ \sin(u) + 1.000\ u_{xx}$	2/2	–	5/5
	D-CIPHER (GP)	$u_{tt} = -0.891\ \sin(u) + 0.794\ u_{xx}$	2/2	–	–
	WSINDy polynomial	$u_{tt} = -0.520\,u + 0.053\,u^3 - 0.024\,u_t + 0.775\,u_{xx}$	1/2	u, u^3, u_t	3/5

continued on next page

Table S11, continued

noise	method	discovered equation	correct	spurious	support
	SINDy polynomial	$u_{tt} = -0.437\,u + 0.044\,u^3 + 0.692\,u_{xx}$	1/2	u, u^3	5/5
0.05%	**Higher-Order-KANDy**	$u_{tt} = -1.000\,\sin(u) + 1.003\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.000\,\sin(u) + 1.000\,u_{xx}$	2/2	–	5/5
	WSINDy polynomial	$u_{tt} = -0.570\,u + 0.058\,u^3 + 0.015\,u_t + 0.805\,u_{xx}$	1/2	u, u^3, u_t	3/5
	SINDy polynomial	$u_{tt} = -0.426\,u + 0.043\,u^3 + 0.677\,u_{xx}$	1/2	u, u^3	5/5
0.10%	**Higher-Order-KANDy**	$u_{tt} = -1.002\,\sin(u) + 1.005\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.000\,\sin(u) + 1.000\,u_{xx}$	2/2	–	5/5
	WSINDy polynomial	$u_{tt} = -0.536\,u + 0.055\,u^3 - 0.016\,u_t + 0.783\,u_{xx}$	1/2	u, u^3, u_t	3/5
	SINDy polynomial	$u_{tt} = -0.396\,u + 0.039\,u^3 + 0.635\,u_{xx}$	1/2	u, u^3	5/5
0.25%	**Higher-Order-KANDy**	$u_{tt} = -0.981\,\sin(u) + 0.988\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.002\,\sin(u) + 1.000\,u_{xx}$	2/2	–	5/5
	WSINDy polynomial	$u_{tt} = -0.537\,u + 0.055\,u^3 + 0.016\,u_t + 0.787\,u_{xx}$	1/2	u, u^3, u_t	2/5
	SINDy polynomial	$u_{tt} = -0.261\,u + 0.023\,u^3 + 0.445\,u_{xx}$	1/2	u, u^3	5/5
0.50%	**Higher-Order-KANDy**	$u_{tt} = -0.966\,\sin(u) + 0.991\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.000\,\sin(u) + 1.000\,u_{xx}$	2/2	–	5/5
	D-CIPHER (GP)	$u_{tt} = -0.843\,\sin(u) + 0.764\,u_{xx}$	2/2	–	–
	WSINDy polynomial	$u_{tt} = -0.487\,u + 0.049\,u^3 + 0.735\,u_{xx}$	1/2	u, u^3	1/5
	SINDy polynomial	$u_{tt} = -0.102\,u + 0.222\,u_{xx}$	1/2	u	5/5
0.75%	**Higher-Order-KANDy**	$u_{tt} = -0.982\,\sin(u) + 0.998\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.001\,\sin(u) + 1.000\,u_{xx}$	2/2	–	5/5
	WSINDy polynomial	$u_{tt} = -0.562\,u + 0.057\,u^3 - 0.014\,u_t + 0.818\,u_{xx}$	1/2	u, u^3, u_t	2/5
	SINDy polynomial	$u_{tt} = -0.032\,u + 0.127\,u_{xx}$	1/2	u	5/5
1%	**Higher-Order-KANDy**	$u_{tt} = -0.969\,\sin(u) + 0.984\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -0.994\,\sin(u) + 1.000\,u_{xx}$	2/2	–	5/5
	D-CIPHER (GP)	$u_{tt} = -0.857\,\sin(u) - 0.011\,u_{xt} + 0.770\,u_{xx}$	2/2	u_{xt}	–
	WSINDy polynomial	$u_{tt} = -0.500\,u + 0.050\,u^3 + 0.749\,u_{xx}$	1/2	u, u^3	4/5
	SINDy polynomial	$u_{tt} = 0.083\,u_{xx}$	1/2	–	4/5
2%	**Higher-Order-KANDy**	$u_{tt} = -0.967\,\sin(u) + 1.003\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.002\,\sin(u) + 0.999\,u_{xx}$	2/2	–	5/5
	WSINDy polynomial	$u_{tt} = -0.520\,u + 0.053\,u^3 - 0.019\,u_t + 0.768\,u_{xx}$	1/2	u, u^3, u_t	4/5
	SINDy polynomial	$u_{tt} = 0.036\,u_{xx}$	1/2	–	4/5
3%	**Higher-Order-KANDy**	$u_{tt} = -0.984\,\sin(u) + 1.023\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.009\,\sin(u) + 1.001\,u_{xx}$	2/2	–	4/5
	WSINDy polynomial	$u_{tt} = -0.531\,u + 0.054\,u^3 + 0.792\,u_{xx}$	1/2	u, u^3	2/5
	SINDy polynomial	$u_{tt} = -0.047\,u + 0.025\,u_{xx}$	1/2	u	5/5
5%	**Higher-Order-KANDy**	$u_{tt} = -0.790\,\sin(u) + 0.879\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -0.012\,u - 0.998\,\sin(u) + 1.007\,u_{xx}$	2/2	u	3/5
	D-CIPHER (GP)	$u_{tt} = -1.085\,\sin(u) + 0.021\,u_{xt} + 0.842\,u_{xx}$	2/2	u_{xt}	–
	WSINDy polynomial	$u_{tt} = -0.548\,u + 0.056\,u^3 + 0.793\,u_{xx}$	1/2	u, u^3	2/5
	SINDy polynomial	$u_{tt} = -0.172\,u + 0.019\,u_{xx}$	1/2	u	5/5
7.5%	**Higher-Order-KANDy**	$u_{tt} = -0.670\,\sin(u) + 0.774\,u_{xx}$	2/2	–	5/5
	WSINDy oracle	$u_{tt} = -1.018\,\sin(u) + 1.013\,u_{xx}$	2/2	–	4/5
	WSINDy polynomial	$u_{tt} = -0.546\,u + 0.055\,u^3 - 0.020\,u_t + 0.788\,u_{xx}$	1/2	u, u^3, u_t	2/5
	SINDy polynomial	$u_{tt} = -0.413\,u + 0.028\,u^3 + 0.018\,u_{xx}$	1/2	u, u^3	3/5
10%	**Higher-Order-KANDy**	$u_{tt} = -0.695\,\sin(u) + 0.801\,u_{xx}$	2/2	–	4/5
	WSINDy oracle	$u_{tt} = -0.015\,u - 0.988\,\sin(u) + 0.989\,u_{xx}$	2/2	u	3/5
	WSINDy polynomial	$u_{tt} = -0.545\,u + 0.055\,u^3 + 0.787\,u_{xx}$	1/2	u, u^3	1/5
	SINDy polynomial	$u_{tt} = -0.694\,u + 0.054\,u^3 + 0.016\,u_{xx}$	1/2	u, u^3	3/5