Deep-Koopman-KANDy: Symbolic Dictionary Coordinates for Deep-Koopman Operators with Kolmogorov–Arnold Networks for Dynamics

Kevin Slote
Clarkson Center for Complex Systems Science
Potsdam, NY 13676
kslote@clarkson.edu

Erik Bollt
Clarkson Center for Complex Systems Science
Potsdam, NY 13676
embollt@clarkson.edu

Jeremie Fish
Clarkson Center for Complex Systems Science
Potsdam, NY 13676
jafish@clarkson.edu

Abstract

Symbolic library—or Koopman dictionary—selection is a fundamental challenge in data-driven dynamical systems. Extended Dynamic Mode Decomposition (EDMD), Sparse Identification of Nonlinear Dynamics (SINDy), and Kolmogorov–Arnold Networks for Dynamics (KANDy) all require the practitioner to commit to a function library at training time; Deep-Koopman operators avoid this commitment but produce uninterpretable latent observables. We propose Deep-Koopman-KANDy, a structured approach to post-hoc symbolic dictionary readout that combines Deep-Koopman modeling with Kolmogorov–Arnold Networks for Dynamics. The encoder and decoder of a Deep-Koopman operator are replaced with two-layer Kolmogorov–Arnold Networks (KANs), and a level-set construction together with a chain-rule gradient identity exposes the compositional structure of the learned observables in a basis chosen after training. The recovered expressions are post-hoc symbolic descriptions of the learned coordinates and their foliations, not certified finite Koopman-invariant dictionaries. We evaluate the method on the Lorenz system, the Chirikov standard map, the Ikeda map, and the Arnold cat map. On Lorenz it recovers the target dictionary $\{x, y, z, xy, xz\}$ with perfect recall and Jaccard score 0.79 ± 0.06; on the standard map it recovers a low-order Fourier basis matching the analytical structure; on Ikeda—which has no sparse polynomial representation—a misspecified polynomial readout still recovers the correct foliation coordinate $g \approx x^2 + y^2$; and on the Arnold cat map—used as a negative control because finite-dimensional Koopman closure is provably impossible—the method fails to find a sparse closure, as expected.

1 Introduction

Data-driven discovery of governing equations offers a compelling route to forecasting and controlling nonlinear dynamical systems, but current approaches require an unsatisfying up-front choice of candidate libraries. Methods such as Extended Dynamic Mode Decomposition (EDMD) [67, 68], Sparse Identification of Nonlinear Dynamics (SINDy) [10], or Kolmogorov–Arnold Networks for Dynamics (KANDy) [60] depend on hand-crafted function libraries specified before fitting, while Deep-Koopman autoencoders avoid manual dictionary design at the cost of opaque latent observables [17, 41, 44, 48]. For scientific model discovery, this library choice is a central bottleneck. We address the timing of the library choice with Deep-Koopman-KANDy, which replaces standard encoder-decoder networks in Deep-Koopman operators with Kolmogorov–Arnold Networks

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

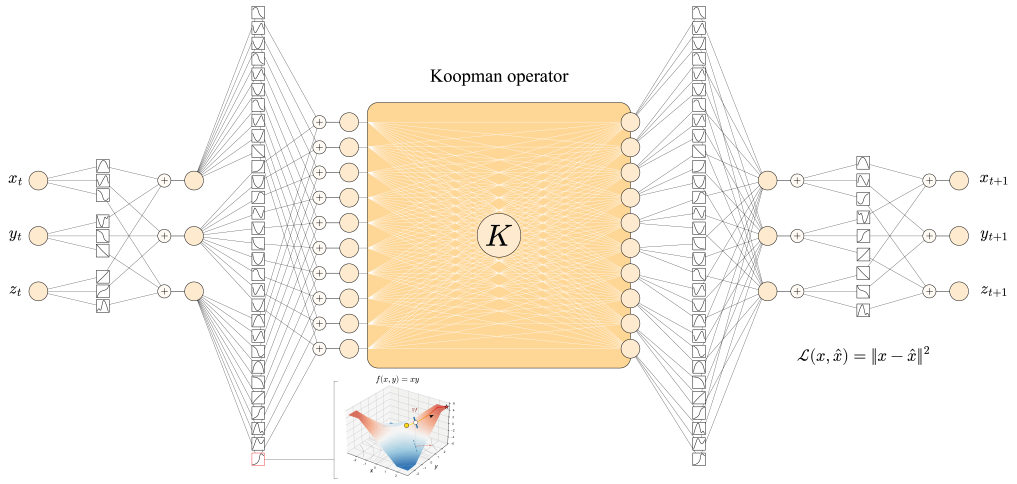

Figure 1: Deep-Koopman-KANDy architecture. A two-layer KAN encoder (**left**) lifts the state $\mathbf{x}$ into a structured latent observable $\mathbf{z}$. A stable linear generator K (**center**) advances the dynamics. A KAN decoder (**right**) inverts the lifting. Unlike a deep autoencoder, the KAN encoder exposes the compositional structure of each observable, allowing a level-set analysis of the higher-dimensional manifolds (**inset**) constructed from incident 1D splines in the hidden layer.

(KANs)—allowing for post-training pruning of edge activations, a concrete advantage over multi-layer perceptrons (MLPs)—and recovers candidate structured, interpretable Koopman observables from the learned lifting that can then be read out symbolically in a user-chosen basis.

Given a dynamical system $\mathbf{x}_{t+1} = F(\mathbf{x}_t)$ on $\mathbb{R}^n$, the Koopman operator $\mathcal{K}$ acts on scalar observables $\varphi : \mathbb{R}^n \to \mathbb{R}$ by composition: $(\mathcal{K}\varphi)(\mathbf{x}) = \varphi(F(\mathbf{x}))$. When a finite set of observables is (approximately) closed under $\mathcal{K}$, the dynamics reduce to a finite-dimensional linear system in lifted coordinates, enabling classical techniques in numerical linear algebra for model reduction, prediction, data fusion, and control [11, 12, 53], with applications in fluid dynamics [45, 57], energy modeling in buildings [23], oceanography [24], and molecular kinetics [70].

Prior Work & Literature Gap: Data-Driven Approximations and their Limitations. Dynamic Mode Decomposition (DMD) and its extension EDMD provide practical finite-dimensional approximations of the Koopman operator [11, 55, 67]. EDMD lifts the state via a dictionary of nonlinear observables, improving upon standard DMD in many contexts; recent work further advances this by coupling EDMD with a trainable neural network dictionary [41]. Discovery methods such as SINDy [10] and KANDy [60] identify governing equations from practitioner-specified libraries of candidate functions. Deep-Koopman methods [41, 44, 48] bypass dictionary selection by learning observables via neural-network autoencoders, but at the cost of interpretability—the latent features are unknown. A common limitation across these approaches is a commitment to a library at training time. Recent dictionary-learning methods differ mainly in when this commitment is made—a sub-dictionary selected a priori from a fixed analytic library [56], a dictionary learned implicitly during training as kernels [7] or as MLP dictionaries certified by spectral residuals [19], or a black-box KAN lifting [46, 47]—whereas we defer only the descriptive basis, to after training.

Kolmogorov–Arnold Networks. KANs [42] replace each weight with a learnable univariate spline, exposing the functional form of the learned mapping at every edge. They are competitive with MLPs on regression, PDE solving, and scientific benchmarks [32, 40, 42]. KANDy [60] combines a single KAN layer with a SINDy-style identification framework. Because a single KAN layer represents only additive functions $g(\mathbf{x}) = \sum_i \psi_i(x_i)$, KANDy must lift cross terms (e.g., xy) into the dictionary by hand. Other recent work applies KANs to discrete chaotic maps and ordinary differential equations [3, 33, 34, 49].

This work. Deep-Koopman-KANDy replaces the encoder and decoder of a Deep-Koopman architecture with two-layer KANs (Figure 1) and applies a level-set analysis on activation manifolds to recover compositional structure (Figure 1, inset). The architecture decouples representation from description—the encoder learns a representation as a basis-free foliation, and the level-set machinery describes that foliation in any chosen basis after training. A wrong basis is then a descriptive failure—a denser readout of the same learned object—rather than a representational failure that requires retraining.

Contributions.

- A two-layer KAN encoder–decoder architecture for Deep-Koopman models with a stably parameterized linear generator.
- A level-set analysis of activation manifolds that recovers the compositional decomposition $f = h \circ g$ of each learned observable—symbolizing KANs in totality rather than on isolated edges.
- Empirical demonstrations on continuous and discrete systems, including a negative-control case (Arnold cat map) where finite-dimensional spectral closure is provably impossible.
- Evidence that misspecified readout dictionaries (e.g., a polynomial readout on Ikeda) degrade gracefully—g becomes denser, h absorbs the residual—rather than failing catastrophically.

The aim is not to replace SINDy, KANDy, or EDMD downstream, but to learn the candidate observable structure that informs the library those methods use.

2 Background

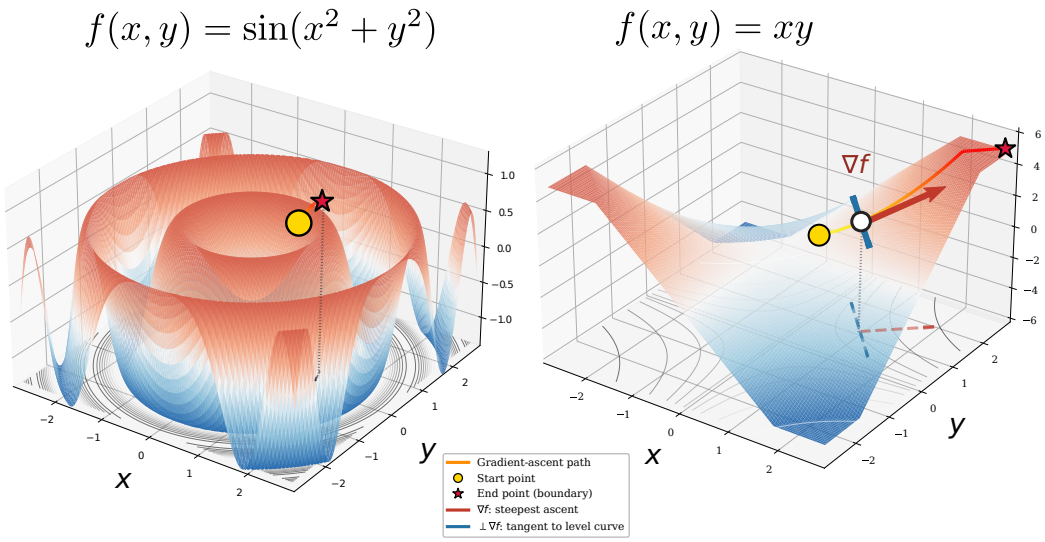

Figure 2: The hidden activation layer of the KAN defines a manifold in latent space. The level sets of the learned coordinate functions induce a foliation that we exploit to recover compositional structure.

Koopman operators. For a continuous-time system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, the Koopman generator $\mathcal{A}$ acts on observables by

$$\mathcal{A}\varphi = \nabla\varphi \cdot f(\boldsymbol{x}), \tag{1}$$

so eigenfunctions φ_λ with $\mathcal{A}\varphi_\lambda = \lambda\varphi_\lambda$ evolve as $\varphi_\lambda(\Phi^t(\boldsymbol{x})) = e^{\lambda t}\varphi_\lambda(\boldsymbol{x})$, where Φ^t is the flow of f. A Deep-Koopman model [44] learns an encoder $\mathcal{E}\colon \mathbb{R}^n \to \mathbb{R}^d$, a decoder $\mathcal{D}\colon \mathbb{R}^d \to \mathbb{R}^n$, and a matrix $K \in \mathbb{R}^{d\times d}$ such that

$$\boldsymbol{z}_t = \mathcal{E}(\boldsymbol{x}_t), \qquad \boldsymbol{z}_{t+\Delta t} \approx \exp(K\Delta t)\boldsymbol{z}_t, \qquad \boldsymbol{x}_{t+\Delta t} \approx \mathcal{D}(\boldsymbol{z}_{t+\Delta t}). \tag{2}$$

We parameterize the generator as $K = \Omega - L^\top L$ with Ω skew-symmetric and L lower triangular. This forces $\mathrm{Re}(\sigma(K)) \leq 0$ (the eigenvalues of K have non-positive real part) and hence

$|\lambda(\exp(K\Delta t))| \leq 1$ (equivalently, the one-step propagator $\exp(K\Delta t)$ has all eigenvalues in the closed unit disk, by the spectral mapping theorem), ensuring stable rollouts without spectral penalties or projection steps.

The encoder must represent a broad class of observables while exposing their compositional structure. Both properties follow from a classical representation theorem, whose inner–outer form our two-layer architecture and level-set readout directly exploit:

Theorem 1 (Kolmogorov–Arnold, 1957)**.** *Every continuous $f : [0,1]^n \to \mathbb{R}$ admits the representation*

$$f(\boldsymbol{x}) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \psi_{q,p}(x_p) \right), \tag{3}$$

with continuous univariate $\psi_{q,p}, \Phi_q$ [1, 35].

KANDy. KANDy [60] applies a single KAN layer to a hand-crafted SINDy-style library

$$\hat{f}_\theta(\Theta) = \mathrm{KAN}_\theta(\Theta) = \sum_i \psi_i(\theta_i) \approx d\boldsymbol{x}/dt$$

Additivity enables edgewise symbolic readout, but cross terms such as xy must already be present in Θ. Under the polynomial-edge assumptions of [60], depth alone does not remove this restriction at fixed width. KANs with sufficient width and depth can form cross terms internally but often lose edgewise interpretability: the terms lie in the composition of Equation (3), or arise via the polarization identity $2xy = (x+y)^2 - x^2 - y^2$ [42]. Recovering dictionary atoms therefore requires analyzing each composed observable as a whole, which motivates the level-set decomposition in Section 3.

3 Methods

The pipeline has three phases: train a stable Deep-Koopman model with a two-layer KAN autoencoder; decompose each learned latent coordinate by its level sets; and read out the union dictionary in a chosen basis. It is stated compactly as Algorithm 1 at the end of this section. The paragraphs below define each component, and Appendix G derives the level-set phase step by step.

Architecture. The encoder maps $\boldsymbol{x} \in \mathbb{R}^n$ to $\boldsymbol{z} \in \mathbb{R}^d$ in two KAN layers, with m intermediate units. We denote the architecture $[n, m, d]$. The first layer computes

$$u_j = \sum_{i=1}^{n} \phi_{j,i}^{(1)}(x_i), \qquad j = 1, \ldots, m, \tag{4}$$

where each edge activation is

$$\phi_{j,i}(x_i) = c_{j,i}\, \mathrm{SiLU}(x_i) + \sum_{k=1}^{G+S} w_{j,i,k}\, B_k(x_i), \tag{5}$$

with $\{B_k\}$ a set of order-S B-splines on a grid of size G, and $\mathrm{SiLU}(x) = x\sigma(x)$ a smooth residual. The B-spline basis is the default; Chebyshev polynomials, Fourier features, or radial basis functions are drop-in replacements. The second layer maps $\mathbb{R}^m \to \mathbb{R}^d$:

$$z_k = \sum_{j=1}^{m} \phi_{k,j}^{(2)}(u_j) = \sum_j \Phi_{k,j} \left(\sum_i \psi_{j,i}(x_i) \right), \qquad k = 1, \ldots, d. \tag{6}$$

By the Kolmogorov–Arnold theorem each z_k approximates a broad class of continuous functions of $(x_1, \ldots, x_n)$. The width m controls compositional richness; d controls dictionary size.

Stable propagator. The Koopman generator is parameterized as $K = \Omega - L^\top L$ with Ω skew-symmetric and $L^\top L \succeq 0$, guaranteeing $|\lambda| \leq 1$ for all discrete-time eigenvalues. One-step propagation uses

$$\boldsymbol{z}_{t+\Delta t} = \exp(K\, \Delta t)\, \boldsymbol{z}_t. \tag{7}$$

In practice, U and L are the trainable parameters (not K): $\Omega = U - U^\top$ is derived from the strictly upper triangular part of the free matrix $U \in \mathbb{R}^{d\times d}$, and L is a free lower-triangular matrix. At each forward pass, $K = \Omega - L^\top L$ is reconstructed from U and L, and the dense matrix exponential is used to compute $\exp(K\Delta t)$. Since any U, L give us $\frac{1}{2}(K + K^\top) = -L^\top L \preceq 0$, stability is ensured by construction. Thus, we learn U and L together with the encoder and decoder by minimizing (8).

Training objective. For consecutive pairs $(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ the loss is

$$\mathcal{L} = \gamma \, \|\hat{\boldsymbol{x}}_{t+1} - \boldsymbol{x}_{t+1}\|^2 + \alpha \, \|\hat{\boldsymbol{z}}_{t+1} - \mathcal{E}(\boldsymbol{x}_{t+1})\|^2 + \eta \, \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|^2, \tag{8}$$

with $\hat{\boldsymbol{z}}_{t+1} = \exp(K\Delta t)\boldsymbol{z}_t$, $\hat{\boldsymbol{x}}_{t+1} = \mathcal{D}(\hat{\boldsymbol{z}}_{t+1})$, and $\hat{\boldsymbol{x}}_t = \mathcal{D}(\mathcal{E}(\boldsymbol{x}_t))$. The three terms enforce prediction accuracy (γ), latent consistency (α), and autoencoder fidelity (η). We use $\gamma = \alpha = 1$, $\eta = 0.5$ throughout, which downweights reconstruction so the encoder prioritizes predictive coordinates.

Pruning. After training, edges with low edge-attribution importance $B_{i,j}$ below a threshold τ are removed from the encoder and decoder, and the model is retrained briefly at a reduced learning rate. This prunes spurious spline contributions and substantially improves dictionary parsimony in the level-set decomposition. We choose $\tau = 0.03$ from a sweep on Lorenz (Appendix Table 10); higher thresholds degrade accuracy, lower thresholds leave too many false positives. Appendix M gives exact definitions and computation.

Level-set decomposition. Each latent coordinate is a scalar function $z_k = f(\boldsymbol{x})$ whose level sets foliate the learned activation manifold (Figure 2). We decompose it as $f \approx h \circ g$, where g is a sparse inner function in a chosen post-hoc basis and h is a univariate outer function.[1] This decomposition mirrors the structure the encoder already computes: by the Kolmogorov–Arnold form, each latent coordinate is an outer function of additive inner features (6). Expressing $h \circ g$ clarifies the distinction between the variables on which the observable depends, specifically the sparse support of the inner function g that identifies the dictionary atoms, and the manner in which these variables are combined into a single scalar by the univariate outer function h. Equivalently, $f = h \circ g$ holds exactly when f is constant on the level sets of g, so the observable is organized by a single foliation coordinate.

The inner g is recovered by Lasso on a design matrix $\Theta \in \mathbb{R}^{N\times P}$ with columns drawn from any chosen basis (monomials of total degree $\leq D$ in our experiments, but trigonometric or RBF columns are equally admissible):

$$\min_{\boldsymbol{a}} \; \frac{1}{2N}\|\Theta_s \boldsymbol{a} - f_s\|_2^2 + \lambda \|\boldsymbol{a}\|_1. \tag{9}$$

Standardization (subscript s) ensures the regularizer treats columns uniformly.

Proposition 1 (Outer derivative formula)**.** *If $f(\boldsymbol{x}) = h(g(\boldsymbol{x}))$ and $\nabla g(\boldsymbol{x}) \neq \mathbf{0}$, then*

$$h'\big(g(\boldsymbol{x})\big) = \frac{\nabla f(\boldsymbol{x}) \cdot \nabla g(\boldsymbol{x})}{\|\nabla g(\boldsymbol{x})\|^2}. \tag{10}$$

Proof. See Appendix H. □

The right-hand side of (10) is computed pointwise from samples and binned over g to recover h up to integration constants (Algorithm 1, Phase 2; derivation in Appendix G).

Proposition 1 recovers h assuming $f = h \circ g$, but does this factorization hold for the latent observables produced by training? The relevant condition is that ∇f and ∇g be parallel at each point—equivalent to f being constant on level sets of g. In the ambient space $\mathbb{R}^n$ this is restrictive: the polynomial identity $xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2$ shows that $f = xy$ and $g = (x+y)^2$ have transverse level sets, so no ambient h exists.

The dynamics, however, live on a d-dimensional attractor $\mathcal{A} \subset \mathbb{R}^n$, and only the intrinsic gradients $\nabla^{\mathcal{A}} f := P_x \nabla f$ and $\nabla^{\mathcal{A}} g := P_x \nabla g$—the ambient gradients projected onto the tangent space $T_x\mathcal{A}$ via P_x—are dynamically meaningful. Parallelism on $\mathcal{A}$ is much weaker than parallelism in $\mathbb{R}^n$.

[1] Here "inner" and "outer" refer to the composition order in $f = h \circ g$ (g inner, h outer); this is unrelated to the inner and outer functions of Hardy-space factorization in complex analysis.

Algorithm 1 Deep-Koopman-KANDy.

Require: Trajectory data $\{\boldsymbol{x}_t\}$, time step Δt, latent dimension d
Ensure: Koopman generator K, observable decompositions $\{g_k, h_k\}_{k=1}^{d}$, discovered dictionary $\mathcal{T}$

Phase 1: Train the Deep-Koopman model
Construct two-layer KAN autoencoder $(\mathcal{E}, \mathcal{D})$ as in (4)–(6), with stable Koopman generator $K = \Omega - L^\top L \in \mathbb{R}^{d\times d}$
Form pair dataset $\{(\boldsymbol{x}_t, \boldsymbol{x}_{t+\Delta t})\}$
Minimize $\mathcal{L}$ in (8) via AdamW; prune at threshold τ and retrain briefly

Phase 2: Level-set manifold analysis
for each latent coordinate $k = 1, \ldots, d$ **do**
    Define the scalar observable $f_k(\boldsymbol{x}) = [\mathcal{E}(\boldsymbol{x})]_k$
    Compute ∇f_k via autograd on training samples
    Recover the inner function g_k via the standardized Lasso (9) on a candidate library
    Compute ∇g_k analytically from the recovered library
    Recover the outer function h_k via the gradient identity (10) and term-by-term integration
end for

Phase 3: Interpretation
Forming the union of active supports across all d coordinates gives a discovered Koopman dictionary $\mathcal{T}$. Read the learned dynamics $\boldsymbol{z}_{t+\Delta t} = \exp(K\Delta t)\, \boldsymbol{z}_t$ through $\mathcal{T}$: the atoms of $\mathcal{T}$ describe the coordinates on which the learned generator acts. **Note that** K is not re-fit on $\mathcal{T}$; reported predictions use $\mathcal{D} \circ \exp(K\Delta t) \circ \mathcal{E}$ (see Discussion).

Theorem 2 (Codimension reduction; informal)**.** *On a d-dimensional attractor in $\mathbb{R}^n$, the pointwise condition that $\nabla^{\mathcal{A}} f$ and $\nabla^{\mathcal{A}} g$ be parallel has codimension $d-1$, versus codimension $n-1$ in the ambient space. When $d = 1$ it is automatic; in general, the obstruction is reduced by $n - d = \operatorname{codim} \mathcal{A}$.*

A formal statement and proof of Theorem 2 appear as Theorem 5 in Appendix I; Example 1 below illustrates the resolution for cross terms. A quantitative companion (Theorem 6) bounds the residual of the ambient chain-rule estimator by $\beta = \|\nabla^{\perp} g\|^2 / \|\nabla g\|^2$, the squared sine of the angle between ∇g and $T_x\mathcal{A}$. Under a uniform-orientation model, the expected residual $\mathbb{E}[\beta] = (n-d)/n$ predicts false-positive coefficient magnitudes consistent with observations across all four systems (Appendix Table 8).

Example 1 (The cross-term obstruction on a curve)**.** *Take $f(x, y) = xy$ and $g(x, y) = (x + y)^2$ on $\mathbb{R}^2$. Ambient gradients are $\nabla f = (y, x)$, $\nabla g = 2(x + y)(1, 1)$. For $x \neq y$, these are not parallel, so no ambient h exists with $f = h \circ g$. Now restrict to a 1-dimensional attractor $\mathcal{A} \subset \mathbb{R}^2$ with tangent $\tau(s) = (\cos\theta(s), \sin\theta(s))$. The intrinsic gradients are scalars $df_x(\tau) = y\cos\theta + x\sin\theta$ and $dg_x(\tau) = 2(x + y)(\cos\theta + \sin\theta)$. By Theorem 5(1), the intrinsic factorization holds whenever both are nonzero, and*

$$h'\big(g(x, y)\big) = \frac{y\cos\theta + x\sin\theta}{2(x + y)(\cos\theta + \sin\theta)}.$$

The right-hand side is constant on level sets of g restricted to $\mathcal{A}$ (as required for h to exist). The would-be obstruction $\nabla f - h'(g)\nabla g$ is nonzero in $\mathbb{R}^2$ but lies in $N_x\mathcal{A}$ pointwise.

4 Results

We evaluate four systems spanning Koopman structural regimes—sparse-polynomial (Lorenz), sparse non-polynomial (standard map), finite non-sparse (Ikeda), and no finite closure (Arnold cat map). In the appendix, we test scalability with a higher-dimensional case, the 64-D Kuramoto–Sivashinsky PDE (Appendix E), and Appendix Table 7 gives a decision procedure for diagnosing a poor or ambiguous readout. Appendix K quantifies the readout's sensitivity to the Lasso weight, polynomial degree, basis choice, and pruning threshold on the non-polynomial systems.

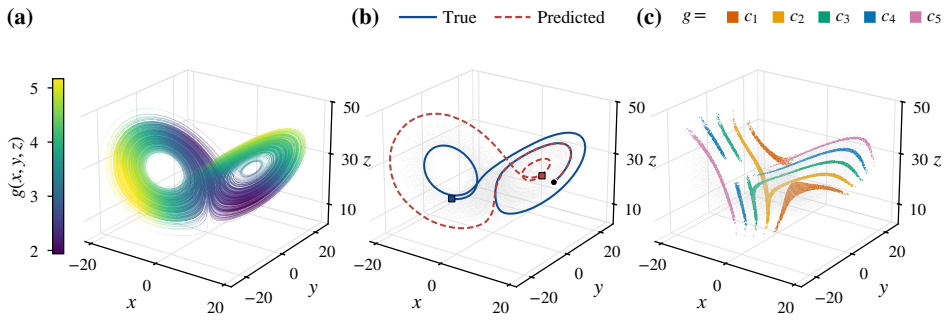

Figure 3: **Lorenz attractor and learned Koopman observable. (a)** Attractor colored by the learned observable $g(x, y, z)$. **(b)** Ground-truth and predicted trajectory over $\approx 2\tau_L$ where $\tau_L = 1/\lambda_{\max} \approx 1.1$ ($\lambda_{\max}$ the largest Lyapunov exponent). **(c)** Level-set bands $g(x, y, z) \approx c_i$ induce a structured foliation of the attractor.

4.1 Lorenz system

$$\dot{x} = \sigma(y - x), \quad \dot{y} = x(\rho - z) - y, \quad \dot{z} = xy - \beta z, \tag{11}$$

with $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. The vector field is supported by the target dictionary $\mathcal{T} = \{x, y, z, xy, xz\}$.

Headline result. Deep-Koopman-KANDy with architecture $[3, 5, 5]$ recovers $\mathcal{T}$ at the union level: every term in $\mathcal{T}$ appears in at least one latent coordinate across all five seeds, with per-seed Jaccard 0.79 ± 0.06 and union recall 1.0 (Table 1). Each z_k contains a subset of $\mathcal{T}$ together with residual false positives $\{x^2, y^2\}$ at coefficients below 10^{-2}, attributable to polynomial-approximation residuals of the learned splines. Three of five latent coordinates achieve $R^2(h \circ g) \geq 0.92$ (Appendix Table 3). The learned observable and its level-set foliation of the attractor are shown in Figure 3.

Spectral interpretation. The latent z_k-coordinate aligns with the dominant spiral pair at ± 1.43 Hz ($r = 0.94$, $R^2 = 0.99$). The cross terms xy and xz are distributed across all five Koopman modes ($R^2 = 0.85$ and 0.77), indicating that cross-term dynamics arise from collective interference rather than a single mode—a reading of mode mixing that opaque latent models cannot provide.

We train on 60 trajectories, 6000 RK4 steps each, AdamW with learning rate 10^{-3}, 200 epochs followed by 100 epochs at 5×10^{-4} after pruning at $\tau = 0.03$ (threshold sweep in Appendix Table 10). The pruning step reduces the manifold dimensionality (the spline $\psi_{5,3}(z)$ vanishes) and lowers false-positive rates from ~ 4 to 2.2 per dimension while leaving one-step MSE essentially unchanged (2.6 to 2.5×10^{-3}). A degree-3 Lasso readout with $\lambda = 10^{-6}$ then yields $\mathcal{T}$.[2]

4.2 Standard map

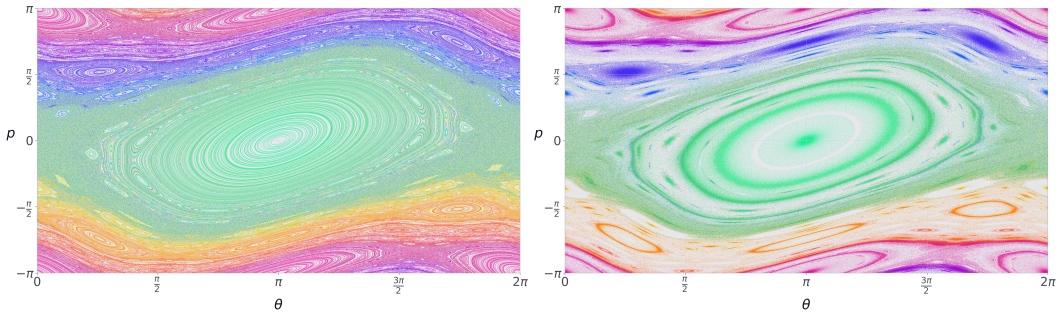

Figure 4: Deep-Koopman-KANDy on the standard map at $\kappa = 0.97$. **Left:** ground-truth phase portrait. **Right:** predicted phase portrait as the encoder width m increases. Larger m reproduces both the KAM tori (invariant curves of regular motion) and the surrounding chaotic sea.

[2]GitHub: https://github.com/Center-For-Complex-Systems-Science/Deep-Koopman-KANDy

We evaluate the standard map $(\theta_{n+1}, p_{n+1}) = (\theta_n + p_n + \kappa \sin\theta_n,\ p_n + \kappa \sin\theta_n) \bmod 2\pi$ at $\kappa = 0.97$, the mixed regime where regular tori coexist with chaos.

The encoder recovers the analytical Fourier basis. The dominant learned observables span

$$\left\{ \begin{array}{l} \cos\theta,\ \sin\theta,\ \cos p,\ \sin p, \\ \cos\theta\cos p,\ \sin\theta\sin p, \cos\theta\sin p,\ \sin\theta\cos p, \\ \cos\theta\sin\theta,\ \cos p\sin p \end{array} \right\}, \tag{12}$$

which in Fourier language is $e^{i(m\theta+np)}$ with $|m| + |n| \leq 2$, the analytical observable structure for the standard map. The recovery is independent of the propagator parameterization: the encoder learns a candidate basis, and the propagator separately learns its evolution.

The stable parameterization preserves geometry. Of 128 latent dimensions, ~ 77 have $\sigma_i \approx 1$ and the rest lie in $[0.84, 0.95]$, allocating one subset to nearly conservative dynamics and the other to mildly contractive dynamics consistent with mixing in the chaotic sea. Long-horizon rollouts preserve KAM island geometry while populating the chaotic sea (Figure 4).

4.3 Ikeda map (misspecified readout)

The Ikeda map depends on $r^2 = x^2+y^2$ through trigonometric functions of $t(r^2) = 0.4-6/(1+r^2)$. A degree-3 polynomial Lasso readout is therefore misspecified by construction. Lacking a sparse polynomial representation, Ikeda tests readout under a wrong basis.

Misspecification yields a dense but accurate decomposition. Across all four latent coordinates of a $[2, 4, 4]$ encoder, the level-set decomposition holds with $R^2(h \circ g) \in [0.92, 0.99]$ (Figure 6, Table 4). The recovered g is not sparse: each dimension picks up several higher-order monomials as the polynomial basis strains to approximate the non-polynomial composition. The level sets nonetheless form coherent bands across the attractor, and the gradient identity recovers a nontrivial $h'(g)$ rather than collapsing to an affine correction on z_0 and z_2.

The compositional structure is recovered. The level-set Lasso identifies $g \approx x^2 + y^2$ from matched quadratic coefficients, and the gradient identity reveals an outer function well-described by $\{\cos(t(g)), \sin(t(g))\}$ with $t(g) = 0.4 - 6/(1 + g)$. The foliation coordinate is correct; the outer map must still be expressed in the trigonometric basis to obtain a closed form. This is the central decoupling claim: when the readout dictionary is wrong, the error appears as a denser g and a more complex h, not as a representational failure.

4.4 Arnold cat map (continuous-spectrum control)

The Arnold cat map on the two-torus,

$$T : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \bmod 1, \qquad (x, y) \in [0, 1)^2, \tag{13}$$

is the canonical uniformly hyperbolic, area-preserving map with purely continuous Koopman spectrum: the constant function is the only eigenfunction. The Lyapunov exponent $\lambda_{\max} = \log((3 + \sqrt{5})/2) \approx 0.962$ sets the predictability horizon at $\tau_L \approx 1.04$ iterates. We use this system as a negative control where a finite-dimensional Koopman model should fail, and we ask whether our method fails.

We embed (x, y) as $(\cos 2\pi x, \sin 2\pi x, \cos 2\pi y, \sin 2\pi y) \in \mathbb{R}^4$ to obtain a smooth torus representation, train a $[4, 24, 1024]$ encoder for 200 epochs on 200 trajectories of length 500, and reach train and validation losses of 6×10^{-5}.

The spectral failure. The learned 1024×1024 propagator has $|\lambda| \in [0, 0.958]$ with mean ≈ 0.88; no eigenvalue satisfies $|\lambda| > 0.96$. This contrasts with Lorenz, where many modes lie near $|\lambda| = 1$, and is the spectral signature of the absent eigenfunction structure. As model size grows (Table 6), one-step accuracy improves by two orders of magnitude, but the prediction horizon extends by only ~ 4 steps, consistent with the Lyapunov barrier. One-step prediction is sharp ($\mathrm{MSE}_1 \approx 2.4 \times 10^{-6}$, roughly $35{,}000\times$ below persistence), but error grows exponentially and reaches the persistence scale by step ~ 7 (Appendix Table 5). Figure 5 shows the ground-truth, full-model, and ablated rollouts and their phase portraits.

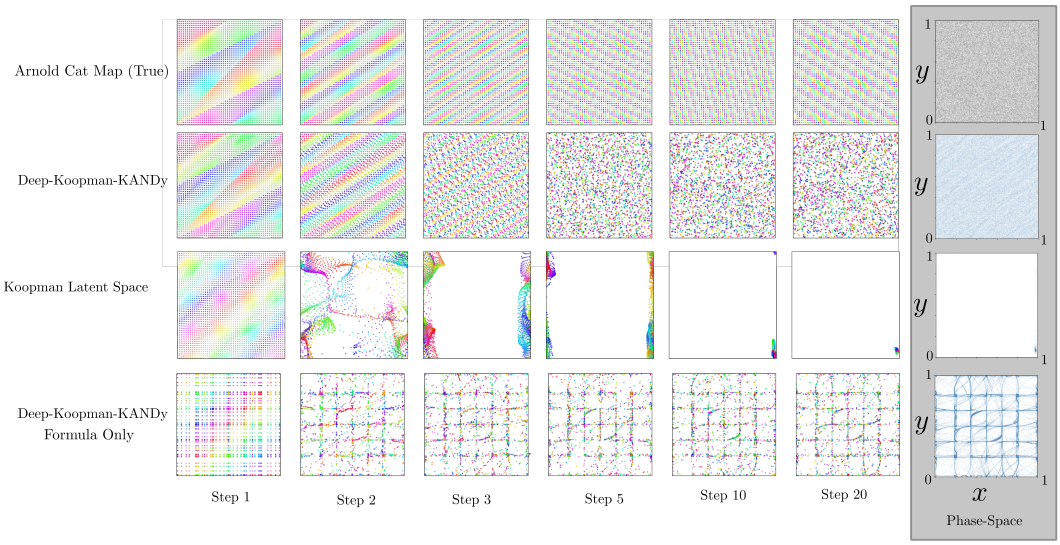

Figure 5: Arnold cat map: ground truth, full model, and two ablations. **Row 1:** ground-truth iterations and phase portrait. **Row 2:** autoregressive rollout of the full Deep-Koopman-KANDy model and its phase portrait. **Row 3:** latent-only rollout (M^n after a single encode where $M = \exp(K\Delta t)$); amplitudes decay and the rollout converges to the ergodic mean. **Row 4:** symbolic-only rollout from the recovered Fourier formula ($\mathcal{D} \circ M \circ \mathcal{E}$ with no re-encoding); a lattice phase portrait emerges from Gibbs artifacts at the sawtooth discontinuity. The full model rolls out beyond the Lyapunov time only when the latent generator and the KAN encoder/decoder act together.

Three rollout regimes expose the architecture's roles. (i) Pure latent rollout (M^n after a single encode) collapses to a point as latent amplitudes decay like 0.958^n, with information effectively gone after $\sim$ 62 steps. (ii) Symbolic rollout of the recovered Fourier formula $f_x \approx 1/2 - \sum_{k=1}^{6} \sin(2\pi k(2x+y))/(\pi k)$ and $f_y \approx 1/2 - \sum_{k=1}^{6} \sin(2\pi k(x+y))/(\pi k)$ matches the analytical $1/(\pi k)$ coefficients to within 0.4%, but iteration produces a lattice phase portrait from Gibbs artifacts at the sawtooth discontinuity. (iii) Full autoregressive rollout ($\mathcal{D} \circ M \circ \mathcal{E}$, re-encoding at every step) broadly fills the torus and remains mixing-like over long horizons—the encoder rebuilds nonlinear information at each step.

4.5 Ablation study

We compare Deep-Koopman-KANDy on Lorenz against external baselines and controlled architectural ablations: SINDy-Autoencoder (SINDy-AE), EDMD with dictionary learning (EDMD-DL), an MLP autoencoder with stable propagator (MLP+stable), and ablations isolating KAN depth, encoder/decoder placement, and the stability constraint. Dictionary recovery is measured against $\mathcal{T} = \{x, y, z, xy, xz\}$ via Jaccard similarity and recall; for SINDy-AE and EDMD-DL we apply symbolic regression to approximate a symbolic readout.

The pruned Deep-Koopman-KANDy obtains the best Jaccard (0.79 ± 0.06) at perfect recall. EDMD-DL and SINDy-AE recover only $\{x, y, z\}$ and miss both cross terms (Jaccard 0.60): EDMD-DL includes states by construction, and SINDy-AE places nonlinear structure in the coefficient matrix rather than the lifting map. Architectural ablations cut recall to 0.84–0.96, showing that two stacked KAN layers materially improve cross-term representation. Removing the stability constraint preserves recall but increases false positives (Jaccard 0.63 vs. 0.70, both unpruned); spectral constraints improve parsimony. The pruning threshold $\tau = 0.03$ removes about 18% of the 40 encoder edges, restores MSE within 2% of baseline after 100 epochs of retraining, and avoids the degradation seen at $\tau = 0.05$ ($1.07\times$ MSE) (Appendix Table 10). Across 12 runs, $\tau = 0.03$ sits near the 14th percentile of edge scores, indicating that pruning targets consistently low-importance connections rather than seed-specific artifacts.

The pipeline scales to a 64-D PDE. A higher-dimensional system tests the method beyond the low-dimensional core (full results in Appendix E). On the Kuramoto–Sivashinsky equation, a

Variant	Params ↓	Wall ↓	MSE ↓	NRMSE$_{2\tau}$ ↓	Jaccard ↑	Rec. ↑
External baselines						
EDMD-DL	2,501	3.9h	2.78×10^{-1}	$\mathbf{0.91 \pm 0.07}$	0.60 ± 0.00	0.60
SINDy-AE	5,280	11.1h	$\mathbf{1.93 \times 10^{-3}}$	1.01 ± 0.07	0.60 ± 0.00	0.60
Architectural ablations						
1L-KAN + stable	300	32m	3.28×10^{-3}	1.16 ± 0.22	0.61 ± 0.13	0.84
KAN-enc + MLP-dec	448	33m	7.63×10^{-3}	1.27 ± 0.21	0.70 ± 0.12	0.96
MLP-enc + KAN-dec	450	36m	1.25×10^{-2}	1.39 ± 0.20	0.66 ± 0.07	0.92
MLP + stable	98	24m	3.21×10^{-2}	1.18 ± 0.25	0.62 ± 0.13	0.88
2L-KAN + unconstrained	800	45m	3.63×10^{-3}	1.35 ± 0.29	0.63 ± 0.05	**1.00**
Ours						
Deep-Koopman-KANDy (ℓ_1)	800	46m	4.61×10^{-3}	1.10 ± 0.35	0.70 ± 0.08	**1.00**
Deep-Koopman-KANDy (pruned)	800	3.7h	2.29×10^{-3}	1.13 ± 0.21	$\mathbf{0.79 \pm 0.06}$	**1.00**

Table 1: Dictionary recovery and one-step prediction on Lorenz over 5 seeds. Best results in **bold**.

64-dimensional chaotic PDE lifted to 128 latent coordinates, the autoregressive rollout reaches $\mathrm{NRMSE} = 1.0$ at $1.45\,\tau_L$ (Figure 7 and Figure 8).

Prediction against prior art. To compare with recent Koopman-learning methods on the axis they target, we benchmark one-step and autoregressive prediction against three open-source baselines across all five systems: the ResDMD trainable dictionary [19], a reimplementation of the KAN Deep-Koopman model of Nehma and Tiwari [47], and EDMD with dictionary learning [41] (Appendix L, Table 17). Across three seeds, including the KAN reduced the one-step nMSE by one to three orders of magnitude relative to EDMD dictionary learning on Lorenz, the cat map, and Kuramoto–Sivashinsky (6×10^{-4} versus 3.8×10^{-1} on the KS field), and achieved errors within a factor of two on the standard and Ikeda maps; and the stable generator yields the longest or tied-longest reliable forecast horizon on all five systems.

5 Discussion

Deep-Koopman-KANDy decouples representation from description—the encoder represents the dynamics by learning a foliation of state space, and the readout describes that foliation in a basis chosen after training. The decoupling shifts dictionary dependence from the lifting, where miscommitment is representational, to the symbolic readout, where any flexible basis can be tried without retraining.

Two failure modes, two diagnostics. Two distinct failure modes arise. When the readout dictionary is misspecified but the dynamics admit some finite-dimensional structure (Ikeda), the decomposition degrades gracefully: g becomes denser and h absorbs the compositional residual. When no finite-dimensional basis suffices at all (Arnold cat map), no readout choice recovers sparse structure, and the flat eigenvalue profile becomes the diagnostic. Two measured quantities distinguish these regimes. The first is the profile of the inner Lasso coefficients: sparse and hierarchical for Lorenz and the standard map, dense but structured for Ikeda and KS, and flat for the cat map. The second is the gain $\Delta R^2 = R^2(h \circ g) - R^2(g)$, which is near zero when g alone names the coordinate (h affine) and large when the level sets are right, but reconstruction requires a nonlinear h (Appendix F).

Position in the pipeline. Deep-Koopman-KANDy is a way to learn the candidate observable structure that methods such as SINDy, KANDy, or EDMD need as input. The generator K is learned once, jointly with the encoder and decoder; the post-hoc readout does not re-fit K in the recovered basis, and every reported full-model prediction applies $\mathcal{D} \circ \exp(K\Delta t) \circ \mathcal{E}$ at each step. The architecture and $h'(g) = (\nabla f \cdot \nabla g)/\|\nabla g\|^2$ recovery procedure form a principled route to KAN symbolic function recovery, and the pruning step sharpens dictionary parsimony.

Limitations. First, the method still requires a basis for the symbolic readout, even though the choice can be deferred until after training. Removing all candidate-library dependence—learning the readout basis itself—is left to future work. Second, all experiments use noise-free data; robustness to measurement noise is untested.

Acknowledgments and Disclosure of Funding

This work was supported by the Army Research Office under Grant No. W911NF2310393.

References

[1] V. I. 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, editors, *Collected Works: Representations of Functions, Celestial Mechanics and KAM Theory, 1957–1965*, pages 25–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-01742-1_5.

[2] E. Baake, M. Baake, H.-G. Bock, and K. M. Briggs. Fitting ordinary differential equations to chaotic data. *Physical Review A*, 45:5524, 1992. doi: 10.1103/PhysRevA.45.5524.

[3] J. Bagrow and J. 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.

[4] J. Bakarji, K. Champion, J. N. Kutz, and S. L. Brunton. Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A*, 479 (2276):20230422, Aug. 2023. doi: 10.1098/rspa.2023.0422.

[5] R. G. Baraniuk. Compressed sensing. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007. doi: 10.1109/MSP.2007.4286571.

[6] M. A. Basiri and S. Khanmohammadi. SINDyG: sparse identification of nonlinear dynamical systems from graph-structured data, with applications to Stuart–Landau oscillator networks. *Journal of Complex Networks*, 13(5):cnaf029, Sept. 2025. doi: 10.1093/comnet/cnaf029.

[7] E. L. Bolager, B. Hamzi, H. Owhadi, I. G. Kevrekidis, and F. Dietrich. Dictionary learning for Kernel EDMD. *arXiv preprint arXiv:2604.25572*, 2026.

[8] E. M. Bollt. Controlling chaos and the inverse Frobenius-Perron problem: Global stabilization of arbitrary invariant measures. *International Journal of Bifurcation and Chaos*, 10:1033, 2000.

[9] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representation of nonlinear dynamical systems for control. *PLOS ONE*, 11: e0150171, 2016. doi: 10.1371/journal.pone.0150171.

[10] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

[11] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz. Modern Koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, May 2022. doi: 10.1137/21M1401243.

[12] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos*, 22:047510, 2012. doi: 10.1063/1.4772195.

[13] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008. doi: 10.1109/MSP.2007.914731.

[14] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. doi: 10.1109/TIT.2005.862083.

[15] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006. doi: 10.1002/cpa.20124.

[16] M. Casdagli. Nonlinear prediction of chaotic time series. *Physica D*, 35:335, 1989.

[17] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. doi: 10.1073/pnas.1906995116.

[18] O. Cherednichenko and M. Poptsova. Kolmogorov–Arnold networks for genomic tasks. *Bioinformatics*, 41(2):412–424, 2025.

[19] G. Coote and M. J. Colbrook. Residual-guided dictionary learning for spectrally accurate Koopman approximation. *arXiv preprint arXiv:2606.29083*, 2026.

[20] J. P. Crutchfield and B. McNamara. Equations of motion from a data series. *Complex Systems*, 1:417, 1987.

[21] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. doi: 10.1109/TIT.2006.871582.

[22] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. *Physical Review Letters*, 59: 845, 1987. doi: 10.1103/PhysRevLett.59.845.

[23] M. Georgescu and I. Mezić. Building energy modeling: A systematic approach to zoning and model reduction using Koopman mode analysis. *Energy and Buildings*, 86:794–802, 2015. doi: 10.1016/j.enbuild.2014.10.046.

[24] D. Giannakis, J. Slawinska, and Z. Zhao. Spatiotemporal feature extraction with data-driven Koopman operators. In D. Storcheus, A. Rostamizadeh, and S. Kumar, editors, *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 103–115, Montreal, Canada, 11 Dec 2015. PMLR. URL `https://proceedings.mlr.press/v44/giannakis15.html`.

[25] G. Gouesbet. Reconstruction of standard and inverse vector fields equivalent to a Rössler system. *Physical Review A*, 44:6264, 1991. doi: 10.1103/PhysRevA.44.6264.

[26] P. Grassberger and T. Schreiber. Nonlinear time sequence analysis. *International Journal of Bifurcation and Chaos*, 1:521, 1990.

[27] S. M. Hammel, C. K. R. T. Jones, and J. V. Moloney. Global dynamical behavior of the optical field in a ring cavity. *Journal of the Optical Society of America B*, 2:552, 1985.

[28] C. S. Holling. The components of predation as revealed by a study of small-mammal predation of the European pine sawfly. *The Canadian Entomologist*, 91:293–320, 1959.

[29] C. S. Holling. Some characteristics of simple types of predation and parasitism. *The Canadian Entomologist*, 91:385, 1959.

[30] K. Ikeda. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications*, 30(2):257–261, 1979. doi: 10.1016/0030-4018(79)90090-7.

[31] J. Jiang, Z.-G. Huang, T. P. Seager, W. Lin, C. Grebogi, A. Hastings, and Y.-C. Lai. Predicting tipping points in mutualistic networks through dimension reduction. *Proceedings of the National Academy of Sciences of the USA*, 115:E639, 2018.

[32] W. Knottenbelt, W. McGough, R. Wray, W. Z. Zhang, J. Liu, I. P. Machado, Z. 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.

[33] B. C. Koenig, S. Kim, and S. 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.

[34] B. C. Koenig, S. Kim, and S. 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.

[35] 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.

[36] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315–318, 1931. doi: 10.1073/pnas.17.5.315.

[37] B. O. Koopman and J. von Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):255–263, 1932. doi: 10.1073/pnas.18.3.255.

[38] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93(C):149–160, July 2018. ISSN 0005-1098. doi: 10.1016/j.automatica.2018.03.046.

[39] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In H. Araki, editor, *International Symposium on Mathematical Problems in Theoretical Physics*, volume 39 of *Lecture Notes in Physics*. Springer, Berlin, Heidelberg, 1975. doi: 10.1007/BFb0013365.

[40] L. Li, Y. Zhang, G. Wang, and K. 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. URL `https://doi.org/10.1038/s42256-025-01087-7`.

[41] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos*, 27(10):103111, 2017. doi: 10.1063/1.4993854.

[42] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou, and M. Tegmark. KAN: Kolmogorov–Arnold networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=Ozo7qJ5vZi`.

[43] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141, 1963. doi: 10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2.

[44] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

[45] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013. doi: 10.1146/annurev-fluid-011212-140652.

[46] M. Moradi, S. Panahi, E. Bollt, and Y.-C. Lai. Kolmogorov-Arnold network autoencoders. *arXiv preprint arXiv:2410.02077*, 2024. doi: 10.48550/arXiv.2410.02077.

[47] G. Nehma and M. Tiwari. Leveraging KANs for enhanced deep Koopman operator discovery. *arXiv preprint arXiv:2406.02875*, 2024.

[48] S. E. Otto and C. W. Rowley. Linearly-recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846.

[49] S. Panahi, M. Moradi, E. M. Bollt, and Y.-C. Lai. Data-driven model discovery with Kolmogorov-Arnold networks. *Physical Review Research*, 7:023037, April 2025. doi: 10.1103/PhysRevResearch.7.023037.

[50] N. R. Panczyk, O. F. Erdem, and M. I. Radaideh. Opening the AI black-box: Symbolic regression with Kolmogorov–Arnold networks. *Energy and AI*, 22:100595, 2025. ISSN 2666–5468. doi: 10.1016/j.egyai.2025.100595.

[51] U. Parlitz. Estimating model parameters from time series by autosynchronization. *Physical Review Letters*, 76:1232, 1996.

[52] O. E. Rössler. Equation for continuous chaos. *Physics Letters A*, 57:397, 1976.

[53] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009. doi: 10.1017/S0022112009992059.

[54] T. Sauer. Reconstruction of dynamical systems from interspike intervals. *Physical Review Letters*, 72:3811, 1994. doi: 10.1103/PhysRevLett.72.3811.

[55] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217.

[56] R. Schurig, P. van Goor, K. Worthmann, and R. Findeisen. Shaping the Koopman dictionary by learning on the Grassmannian. *arXiv preprint arXiv:2511.07234*, 2025.

[57] A. S. Sharma, I. Mezić, and B. J. McKeon. Correspondence between Koopman mode decomposition, resolvent mode decomposition, and invariant solutions of the Navier–Stokes equations. *Physical Review Fluids*, 1:032402, 2016. doi: 10.1103/PhysRevFluids.1.032402.

[58] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai. Reconstructing propagation networks with natural diversity and identifying hidden sources. *Nature Communications*, 5:4323, 2014.

[59] K. Slote and J. Fish. Higher-Order Kolmogorov–Arnold Networks for Dynamics. Oct. 2026. doi: 10.48550/arXiv.2610.02637.

[60] K. Slote, J. Fish, and E. Bollt. KANDy: Kolmogorov–Arnold networks and dynamical system discovery. *arXiv preprint arXiv:2602.20413*, 2026. doi: 10.48550/arXiv.2602.20413.

[61] R.-Q. Su, W.-X. Wang, and Y.-C. Lai. Detecting hidden nodes in complex networks from time series. *Phys. Rev. E*, 85:065201, Jun 2012. doi: 10.1103/PhysRevE.85.065201.

[62] G. Sugihara, B. Grenfell, R. M. May, P. Chesson, H. M. Platt, and M. Williamson. Distinguishing error from chaos in ecological time series. *Philosophical Transactions of the Royal Society of London. Series B*, 330:235, 1990.

[63] G. G. Szpiro. Forecasting chaotic time series with genetic algorithms. *Physical Review E*, 55: 2557, 1997. doi: 10.1103/PhysRevE.55.2557.

[64] C. Tao, Y. Zhang, and J. J. Jiang. Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm. *Physical Review E*, 76:016209, 2007. doi: 10.1103/PhysRevE.76.016209.

[65] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J. Ye. Network reconstruction based on evolutionary-game data via compressive sensing. *Physical Review X*, 1:021021, Dec 2011. doi: 10.1103/PhysRevX.1.021021.

[66] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*, 106:154101, Apr 2011. doi: 10.1103/PhysRevLett.106.154101.

[67] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015. doi: 10.1007/s00332-015-9258-5.

[68] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015. doi: 10.3934/jcd.2015005.

[69] M. O. Williams, C. W. Rowley, I. Mezić, and I. G. Kevrekidis. Data fusion via intrinsic dynamic variables: An application of data-driven Koopman spectral analysis. *Europhysics Letters*, 109:40007, 2015. doi: 10.1209/0295-5075/109/40007.

[70] H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé. Variational Koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of Chemical Physics*, 146:154104, 2017. doi: 10.1063/1.4979344.

[71] R. Yang, Y.-C. Lai, and C. Grebogi. Forecasting the future: Is it possible for adiabatically time-varying nonlinear dynamical systems? *Chaos*, 22(3):033119, 2012. doi: 10.1063/1.4740057.

[72] C. Yao and E. M. Bollt. Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems. *Physica D*, 227:78, 2007.

A Related Work

Koopman Operator Theory and Data-Driven Approximations. The Koopman operator [36, 37] provides a linear, infinite-dimensional description of nonlinear dynamical systems and has been applied to model reduction, prediction, data fusion, and control across fluid dynamics [45, 53, 57], energy systems [23], oceanography [24], and molecular kinetics [70]. Dynamic Mode Decomposition (DMD) [55] offers a practical finite-dimensional Koopman approximation, with theoretical foundations surveyed in [11, 12]. Extended DMD (EDMD) [67] improves accuracy by lifting the state through a hand-crafted dictionary of nonlinear observables; replacing this fixed dictionary with a trainable neural-network dictionary achieves strong reconstruction without manual feature engineering [41]. Linear Koopman predictors are studied in [38], and data fusion applications are explored in [9, 69]. Deep-Koopman methods [41, 44, 48] learn observable functions via neural-network autoencoders, achieving flexible lifting at the cost of interpretability; delay-coordinate autoencoders [4] partially address identifiability in this setting.

Data-Driven System Identification. Identifying nonlinear dynamical systems from data has a long history [8, 16, 20, 22, 26, 58, 61, 62, 65, 66, 72], encompassing linear approximations [22, 25, 54], fitting of differential equations [2], chaotic synchronization [51], genetic algorithms [63, 64], the inverse Frobenius–Perron method [8], and least-squares approaches [72]. Sparse optimization methods assume that governing equations have a sparse representation in a chosen basis [66, 71], connecting naturally to compressive sensing [5, 13–15, 21]. SINDy [10] operationalizes sparse regression for continuous systems and performs well on the Lorenz [43] and Rössler [52] systems. Sparsity-based methods nevertheless fail when governing equations lack a sparse polynomial structure, as in the Ikeda map [27, 30], Holling-type ecological models [28, 29, 31], and the Kuramoto model [6, 39].

Kolmogorov–Arnold Networks. KANs [42] replace fixed nonlinear activations with learnable univariate spline functions on each edge, grounded in the Kolmogorov–Arnold representation theorem [1, 35]. This design directly exposes the functional form of the learned mapping. KANs have demonstrated competitive or superior performance relative to MLPs on regression, PDE solving, genomics, and scientific machine learning tasks, often with improved interpretability and symbolic extractability [18, 32, 40, 42, 50].

KANs for Dynamical Systems. Several recent works apply KANs to ODE and dynamical system identification. Panahi et al. [49] demonstrate that KANs can reconstruct attractor statistics for discrete dynamical systems, though without a principled method for recovering governing equations. Bagrow and Bongard [3] introduced multi-exit KANs, in which per-layer prediction branches identify the shallowest network depth that suffices for a given task. Koenig et al. [33] apply symbolic regression to KAN activation weights to recover the Lotka–Volterra predator–prey model (KAN-ODE); subsequent work [34] refines equation estimation via an improved multiplication layer. KANDy [60] replaces sparse regression in SINDy with a single KAN layer, enabling symbolic extraction of univariate observables, and Higher-Order-KANDy [59] extends this to higher-order dynamical systems. Most closely related to our architecture, Nehma and Tiwari [47] replace the MLP encoder/decoder of a Deep-Koopman operator with KANs, and Moradi et al. [46] study KAN autoencoders; in both, the KAN is a black-box lifting with no symbolic readout.

Koopman Dictionary Learning. Among recent dictionary-learning methods, Schurig et al. [56] select an approximately invariant sub-dictionary from a fixed analytic basis by optimizing on the

Grassmannian, Bolager et al. [7] learn an implicit kernel dictionary with relevance-based pruning, and Coote and Colbrook [19] train MLP dictionaries against ResDMD residuals to obtain spectrally certified eigenpairs. Across these the descriptive basis is fixed a priori, learned implicitly, or never made symbolic; Deep-Koopman-KANDy instead recovers a post-hoc symbolic description in any basis chosen after training, under a stability-constrained generator, with a provable-failure negative control. Certifying the recovered observables—e.g., via ResDMD residuals [19]—is natural future work.

B Lorenz

The Lorenz system in Equation (11) is a canonical three-dimensional nonlinear system exhibiting sensitive dependence on initial conditions and a strange attractor. The Lorenz system is a common benchmark in data-driven dynamical systems [10, 17, 60]. It provides a stringent test for evaluating whether the Deep-Koopman-KANDy representation can recover the sparse nonlinear structure of the governing equations.

Table 2 summarizes the performance of the best learned model. The stable Koopman parameterization yields accurate one-step prediction and reconstruction while maintaining all learned eigenvalues inside the unit circle. Despite the intrinsic chaotic divergence of nearby trajectories, the model achieves prediction horizons on the order of one Lyapunov time, indicating that the learned latent dynamics capture the dominant short-time evolution before chaotic error growth dominates.

Beyond forecasting accuracy, the learned representation also recovers the algebraic structure of the Lorenz vector field. Using level-set decomposition (Proposition 1) of the encoder coordinates followed by sparse polynomial regression, every target term in the Lorenz dictionary, $\{x, y, z, xy, xz\}$, is identified in at least one latent coordinate. Table 3 gives the corresponding per-coordinate decomposition, showing that the dominant recovered terms are linear, while the nonlinear interaction terms appear with smaller but consistent coefficients.

Metric	Value
Architecture	KAN $[3, 5, 5]$ + $K = \Omega - L^\top L$
Pruning	Edge threshold 0.03, retrain 100 epochs
One-step MSE	2.55×10^{-3}
Val loss (3-term)	0.013
Reconstruction MSE	5.42×10^{-3}
Horizon (NRMSE > 0.5)	$0.97 \pm 0.47\ \tau_L$
Horizon (NRMSE > 1.0)	$1.37 \pm 0.81\ \tau_L$
$\lvert\lambda\rvert$ range	$[0.854,\ 0.997]$
$\max\ \mathrm{Re}\,\sigma(K)$	-0.26
Eigenmodes	± 1.43 Hz (spiral), 3 real damping
Dictionary (Lasso $\lambda = 1\times10^{-6}$, degree 3)	$\{x,\ y,\ z,\ xy,\ xz\}$
Union recall	1.0
Per-seed Jaccard (mean $\pm$ std)	0.79 ± 0.06
Mean FP / dim	2.2

Table 2: Performance summary of the best Lorenz model: KAN $[3, 5, 5]$ with stable Koopman propagator, after edge pruning (threshold 0.03) and retraining. Every target term in $\{x,\ y,\ z,\ xy,\ xz\}$ is recovered by at least one latent coordinate (union recall $= 1.0$); per-seed Jaccard is 0.79 ± 0.06 across 5 seeds, with residual false positives at coefficients $\lesssim 10^{-2}$ consistent with the codimension residual predicted in Appendix I. Recovery uses level-set decomposition with degree-3 polynomial inner function and Lasso $\lambda = 1\times10^{-6}$. Prediction horizon averaged over 12 validation trajectories; $\tau_L \approx 1.1$.

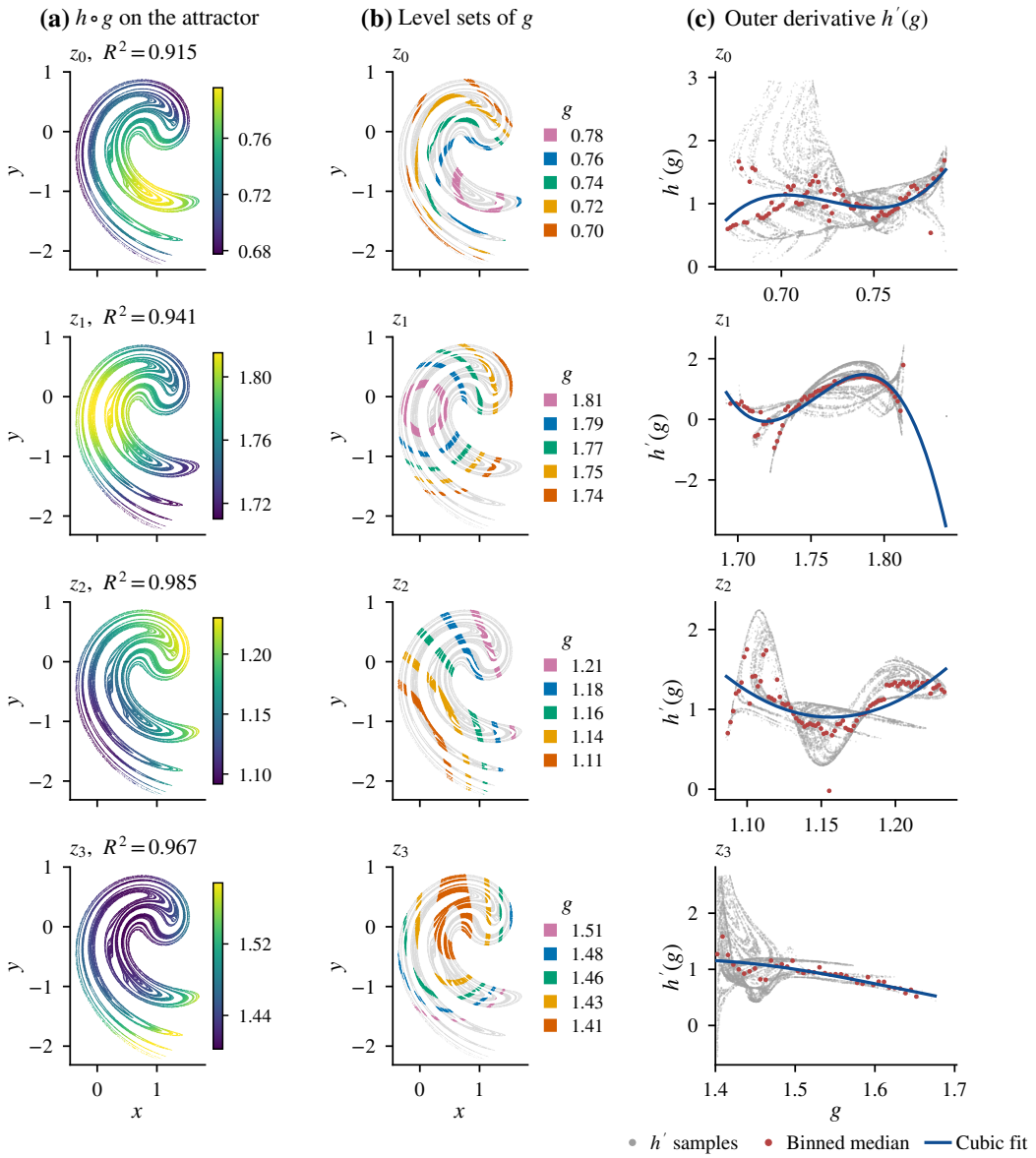

Figure 6: **Ikeda map: level-set decomposition with a misspecified polynomial readout dictionary.** The Ikeda map admits no sparse polynomial representation, so the degree-3 polynomial Lasso used to recover the inner function g is misspecified by construction. We use it anyway, treating the polynomial as a flexible interpolant rather than a structural prior. **(a)** Attractor colored by the reconstruction $h(g(\mathbf{x}))$ for each of four latent dimensions $z_0, \ldots, z_3$ of the trained $[2, 4, 4]$ encoder; $R^2 \in [0.92, 0.99]$ across all four dimensions. **(b)** Level sets of the recovered g trace coherent bands across the attractor, indicating that the polynomial captures the encoder's foliation structure despite its misspecification for the underlying dynamics. **(c)** Outer derivatives $h'(g)$ recovered via the gradient identity (10). All four dimensions yield nontrivial curvature structure on h'—the gradient identity recovers nonlinear outer functions from the polynomial residual rather than collapsing to affine fallback on z_0 and z_2. The z_2 panel is the clearest case: the binned medians (red) show two-peak structure that the cubic fit (blue) tracks tightly. Together, the four dimensions demonstrate the central decoupling claim: a misspecified readout dictionary degrades parsimony, not expressivity, because compositional structure recovered through h compensates for what g alone cannot represent.

Dim	$R^2(h \circ g)$	x	y	z	xy	xz	FP
z_0	0.924	$+0.524$	-0.121	-0.018	-0.008	-0.025	—
z_1	0.584	-0.393	$+0.322$	—	$+0.033$	—	y^2, x^2
z_2	0.962	$+0.446$	-0.082	$+0.187$	—	-0.028	x^2
z_3	0.947	$+0.208$	$+0.065$	$+0.007$	—	-0.009	x^2
z_4	0.826	-0.234	$+0.032$	$+0.246$	$+0.028$	—	x^2

Table 3: Level-set decomposition $z_k \approx h \circ g$ for the Deep-Koopman-KANDy encoder (Lasso $\lambda = 10^{-6}$, degree-3 polynomial basis). Columns show the coefficients of the target terms; dashes indicate terms outside the top 5 or below 10^{-3}. Every target term $\{x, y, z, xy, xz\}$ appears in at least one dimension. Cross-term coefficients are 1–2 orders of magnitude below the linear terms but consistently recovered. FP: false-positive monomials in the top 5.

Dim	$R^2(h \circ g)$	x	y	x^2	y^2	xy	h affine?	FP
z_0	0.915	$+0.110$	-0.033	-0.081	-0.011	-0.038	no	y^3, x^3, x^2y
z_1	0.941	$+0.012$	-0.020	-0.076	-0.050	$+0.014$	yes	x^3, xy^2, y^3, x^2y
z_2	0.986	$+0.039$	$+0.040$	$+0.012$	—	-0.012	no	y^3
z_3	0.968	-0.096	—	$+0.085$	$+0.020$	—	yes	y^3, x^2y

Table 4: Level-set decomposition $z_k \approx h \circ g$ for the Ikeda $[2, 4, 4]$ Deep-Koopman-KANDy encoder (Lasso $\lambda = 10^{-3}$, degree-3 polynomial basis). Columns show the coefficients of the inner function g for the target monomials $\{x, y, x^2, y^2, xy\}$; dashes indicate coefficients below 10^{-3}. The true Ikeda dictionary is compositional: $\{x, y, \cos(t(r^2)), \sin(t(r^2))\}$ where $t(g) = 0.4 - 6/(1 + g)$ and $g = x^2 + y^2$. The degree-3 polynomial Lasso Taylor-expands the trig terms, yielding 7–10 active monomials per dimension. z_1 shows matched y^2/x^2 coefficients (ratio 0.66), consistent with the radial inner function $g = x^2 + y^2$. FP: monomials outside $\{x, y, x^2, y^2, xy\}$ active above threshold.

C Ikeda

The Ikeda map provides a deliberately more challenging test of the proposed post-hoc readout procedure because its natural dictionary is not sparse in low-degree polynomials. In particular, the map depends compositionally on the radial quantity $r^2 = x^2 + y^2$ through trigonometric functions of $t(r^2) = 0.4 - 6/(1 + r^2)$, so a degree-3 polynomial Lasso readout is misspecified by construction. Nevertheless, Figure 6 and Table 4 show that the learned KAN observables still admit accurate level-set decompositions $z_k \approx h(g(\mathbf{x}))$, with $R^2(h \circ g)$ ranging from 0.915 to 0.986 across the four latent dimensions. The recovered inner functions g are not sparse structural dictionaries in the usual SINDy sense: each dimension contains several higher-order false-positive monomials, reflecting the polynomial basis's attempt to approximate the non-polynomial Ikeda composition. However, the level sets of g form coherent bands across the attractor, and the recovered outer derivatives $h'(g)$ are nontrivial rather than collapsing to an affine correction. This is clearest for z_2, where the binned derivative medians exhibit a two-peak structure that is closely tracked by the fitted outer function, while the polynomial inner readout remains relatively parsimonious. These results support the central decoupling claim: when the readout dictionary is misspecified, the error appears primarily as reduced parsimony and extra active monomials, not as a failure of expressivity, because the nonlinear outer map h can absorb compositional structure that the polynomial inner function g cannot represent directly.

D Arnold Cat Map: Extended Analysis

The Arnold cat map is a canonical example of a simple deterministic system with nontrivial global structure. On the two-dimensional torus $\mathbb{T}^2 = [0, 1)^2$, it is defined by the linear map

$$T(x, y) = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \mod 1,$$

or equivalently

$$T(x, y) = (2x + y,\ x + y) \mod 1.$$

Although the transformation is linear before reduction modulo one, the modulo operation introduces discontinuities on the unit square. These discontinuities are responsible for the characteristic folding behavior of the map and make the cat map a useful probe of whether a learned model has captured the true torus dynamics rather than only a local linear approximation.

In our setting, the composed map

$$f = \mathcal{D} \circ M \circ \mathcal{E}$$

represents one full learned step: encoding into the latent space, applying the learned latent dynamics, and decoding back to the torus. Since the exact cat map consists of a linear integer transformation followed by coordinatewise modulo reduction, its nonlinear component has an explicit Fourier description. In particular, the modulo-induced sawtooth terms have Fourier coefficients proportional to $1/(\pi k)$. This gives a direct spectral diagnostic: if the learned map has recovered the true torus folding mechanism, its Fourier coefficients should match the theoretical sawtooth spectrum.

Evaluating the composed map $f = \mathcal{D} \circ M \circ \mathcal{E}$ on a 256×256 torus grid and decomposing in a Fourier basis with maximum frequency 6, we recover the sawtooth representation of the modulo map. The learned coefficients match the theoretical values $1/(\pi k)$ to within 0.4%:

k	Learned	Theory $1/(\pi k)$	Ratio
1	0.318302	0.318310	1.000
2	0.159137	0.159155	1.000
3	0.105647	0.106103	0.996
4	0.079498	0.079577	0.999
5	0.063575	0.063662	0.999
6	0.052957	0.053052	0.998

Since $\max|\lambda| = 0.958$, latent amplitudes decay as 0.958^n. After approximately

$$n \approx \frac{\log(0.01)}{\log(0.958)} \approx 107$$

steps, only 1% of the original amplitude remains. Empirically, the mean latent standard deviation across 500 trajectories drops from 8.5×10^{-3} at step 0 to 8.5×10^{-5} by step 62, and to 3.8×10^{-7} by step 200. After this step, all trajectories decode to essentially the same point.

The two-dimensional Fourier power spectrum concentrates along $(k_x, k_y) \propto (2, 1)$ for f_x and $(k_x, k_y) \propto (1, 1)$ for f_y—precisely the dynamically relevant directions of the cat-map matrix.

The recovered Fourier series is highly accurate for a single step but fails under repeated iteration due to the Gibbs phenomenon: truncating the sawtooth series introduces overshoot near discontinuity lines, and these small one-step errors are re-injected and amplified geometrically under iteration, producing the lattice-like phase portrait shown in Figure 5 (Row 4).

Steps	Model MSE	Persistence	Random
1	2.36×10^{-6}	0.082	0.085
3	5.53×10^{-5}	0.082	0.082
5	2.54×10^{-3}	0.084	0.084
7+	8.22×10^{-2}	0.084	0.083

Table 5: Angular MSE vs. prediction horizon for the Arnold cat map.

Dictionary Recovery: Flat Coefficients Signal Continuous Spectrum. Applying level-set Lasso with a degree-3 trigonometric basis yields $R^2(h \circ g) \approx 0.98$ for all latent dimensions, with h nearly affine. Unlike the Lorenz case, the coefficient structure is flat: each dimension uses nearly all 35 basis terms with similarly small magnitudes ($\sim 4 \times 10^{-3}$), and no sparse basis emerges. This is the signature of continuous spectrum—smooth representability in trigonometric coordinates does not imply the existence of persistent Koopman eigenfunctions.

Architecture	Parameters	Val. loss	$\max\|\lambda\|$	Horizon
$[4, 16, 32]$	12,544	7×10^{-3}	0.985	~ 2
$[4, 32, 64]$	47,616	7×10^{-4}	0.990	~ 3
$[4, 24, 1024]$	1,542,016	6×10^{-5}	0.958	~ 6

Table 6: Effect of model size on cat-map prediction. One-step accuracy improves dramatically with width, but the forecast horizon grows only marginally, consistent with the Lyapunov limit.

E Kuramoto–Sivashinsky: A Chaotic PDE

The Kuramoto–Sivashinsky equation

$$u_t \;=\; -u\,u_x \;-\; u_{xx} \;-\; u_{xxxx}, \qquad x \in [0, L],\; L = 22,\; \text{periodic}, \tag{14}$$

exhibits spatiotemporal chaos ($\lambda_{\max} \approx 0.043$, $\tau_L \approx 23$ time units). We discretize on $N = 64$ grid points (ETDRK4, $\Delta t_{\text{save}} = 0.25$; 30 trajectories of 250 time units, split $24/6$) and train a $[64, 96, 128]$ KAN autoencoder—a $2\times$ lift of the full PDE state—with the stable generator and the plain one-step loss for 500 epochs (final validation loss 7.2×10^{-4}).

Prediction and spectrum. The fully autoregressive rollout ($\mathcal{D} \circ M \circ \mathcal{E}$, re-encoding every step) tracks the cellular dynamics well past a Lyapunov time: NRMSE reaches 0.5 near $0.7\,\tau_L$ and 1.0 at $1.45\,\tau_L$ (33.8 time units, 135 Koopman steps). The learned spectrum is stable by construction ($|\lambda| \in [0.253, 1.000]$, generator $\max \mathrm{Re}\,\sigma(K) = -2 \times 10^{-4}$), with a large cluster near $|\lambda| = 1$. The spectrum also reflects the PDE's linear physics: the attractor-averaged linearized model Jacobian reproduces the KS dispersion relation $\sigma(k) = k^2 - k^4$ across the resolved band—the anti-diffusive growth of every unstable mode ($k < 1$) and the hyperviscous roll-off—and is near-diagonal in Fourier space. The linear operators u_{xx} and u_{xxxx} therefore live in the learned spectrum, not in the symbolic dictionary.

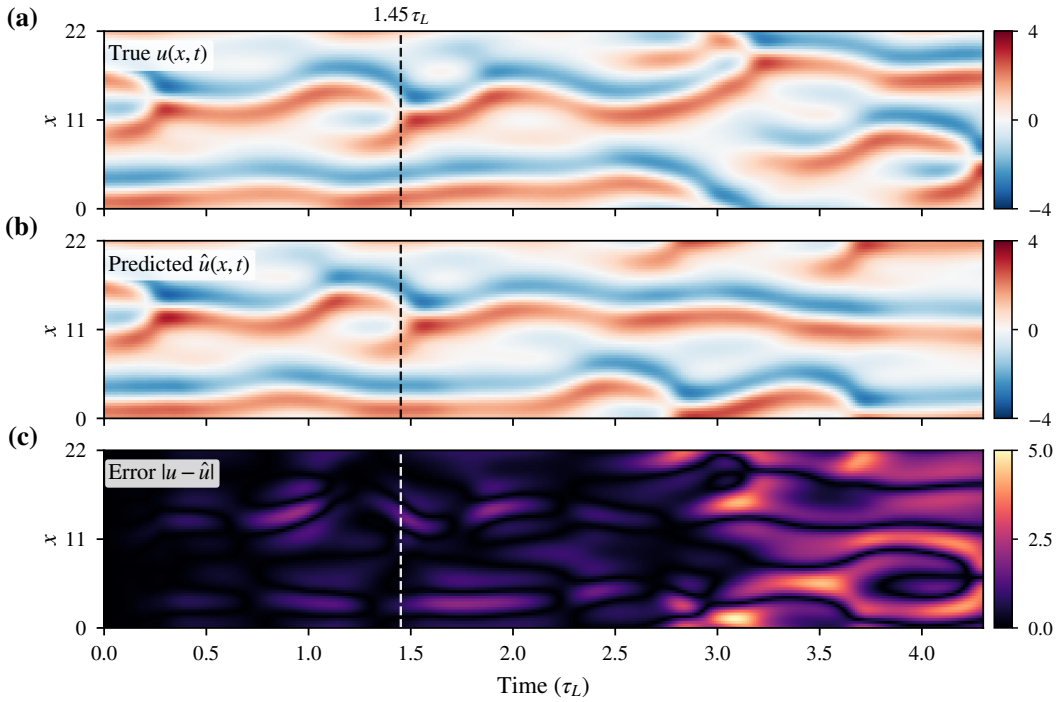

Figure 7: KS autoregressive rollout: **(a)** true field, **(b)** prediction, and **(c)** error in the x–t plane. The model tracks the cellular structure beyond one Lyapunov time and remains bounded and KS-like thereafter.

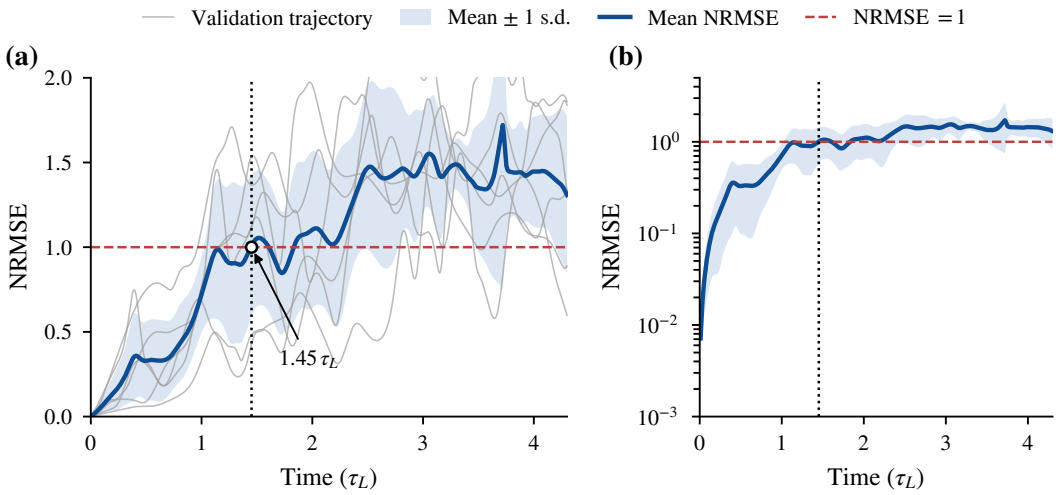

Figure 8: KS prediction horizon: **(a)** NRMSE on a linear scale for each validation trajectory (grey). The mean NRMSE (blue) crosses 1.0 at $1.45\,\tau_L$ (33.8 time units, 135 autoregressive Koopman steps). **(b)** Mean NRMSE on a log scale. The dotted line marks $1.45\,\tau_L$.

Level-set readout. Every latent coordinate admits a degree-2 decomposition with $R^2(h \circ g) > 0.9$ on all 128 dims (median ≈ 0.998), h affine throughout. The recovered observables are smooth collective coordinates whose level sets foliate the activation manifold into coherent spatial-profile classes. They are descriptive coordinates rather than Koopman eigenfunctions: their lagged self-predictability decays to $R^2 \approx 0.04$–0.23 within one Lyapunov time, at or below a leading-POD-mode baseline, whereas an eigenfunction would remain coherent at all lags.

Dictionary: dense but structured, under an exact solver. On the collinear 2145-column degree-2 design, a converged coordinate-descent Lasso returns a median of 160 active atoms per coordinate, while the exact LARS path returns 81 (67–91) at the same regularization and fit; the dictionary does not collapse to a sparse readout. The surviving atoms are quadratic-dominated (median 10 linear $+11$ squares $+60$ cross terms) and the linear-only regression caps at median $R^2 = 0.936$, while degree-2 reaches 0.998, a gain on every one of the 128 coordinates. The nearest-neighbor advection stencils $u_i u_{i+1}$ appear with a pair-normalized enrichment of $2.4\times$ over uniform, the signature of $u\,u_x$, but the same stencil repeats at every grid site, and the operator's eigenmodes are spatially global, so no local-monomial readout is sparse. This is a misspecified-basis case (Table 7) because the structure is present but not sparse in local monomials.

F Diagnosing the Readout: A Decision Procedure

A practitioner facing a poor or surprising readout needs to know which of four situations they are in: a genuine sparse dictionary, a misspecified basis, a benign codimension residual, or the absence of any finite-dimensional closure. We do not claim to certify the underlying truth from data: for a fractal attractor no finite set of candidate bases can prove that a sparse dictionary is absent. Instead we give a set of one-directional tests, each measurable and each with a concrete action (Table 7). Two of the outcomes are actionable and self-validating: a misspecified basis is confirmed by finding a sparse fit in another basis, and a failed factorization by a low $R^2(h \circ g)$. One is a predicted, benign residual. The last, the absence of a finite-dimensional closure, can be suggested by a flat eigenvalue spectrum but not certified, because finite-rank truncation keeps $|\lambda| < 1$. The cat map is the one system whose diagnosis rests on an independent theorem (its Koopman spectrum is provably continuous), against which the flat-spectrum signal is calibrated.

Diagnostic statistic. An additional advantage of the two-stage approach is the diagnostic statistic $\Delta R^2 = R^2(h \circ g) - R^2(g)$, which is near zero when g alone names the coordinate (h affine) and large when the level sets are right, but reconstruction requires a nonlinear h. On Ikeda, the gain is largest on the two coordinates with a nonlinear h, z_2 (from 0.980 to 0.985) and z_0 (from 0.913 to

Observation	Diagnosis	Action
Low $R^2(h \circ g)$ (below ≈ 0.9)	Failed factorization: the coordinate does not factor as $h \circ g$	Do not read a dictionary off it
High $R^2(h \circ g)$ but dense g; a basis swap sparsifies at comparable R^2	Misspecified basis (Ikeda: polynomial vs. radial–trigonometric, Table 13; suspected for Kuramoto–Sivashinsky, Appendix E)	Swap the readout basis post hoc (no retraining)
Small residual coefficients at the scale $\mathbb{E}[\beta] = (n-d)/n$	Benign codimension residual (Appendix I)	Ignore; predicted, not error
Flat coefficient profile across every basis and no eigenvalue near $\lvert\lambda\rvert = 1$	Likely no finite-dimensional Koopman closure (Arnold cat map)	Accept; no basis helps. Suggestive, not certifiable

Table 7: Diagnostic decision procedure for the post-hoc symbolic readout. Each row pairs a measurable observation with a diagnosis and an action. The tests are one-directional: they confirm a misspecified basis or a failed factorization and flag a benign codimension residual, but can only suggest the absence of a finite-dimensional closure.

0.915), and at most 10^{-3} on the affine coordinates z_1 and z_3, as seen in Table 4. The standard-map and KS encoders show essentially no gain. Because g is fit directly to the observable, a near-zero gain indicates only that a nonlinear h is unnecessary over the sampled range, not that the observable is globally affine in g or that compositional structure is absent.

G Complete Algorithm: Step-by-Step Derivation

This appendix derives each step of Algorithm 1.

Once the encoder $\mathcal{E}$—the two-layer KAN of (4)–(6)—is trained, we extract the Koopman dictionary by decomposing each latent coordinate as a scalar composition

$$z_k \;=\; f_k(\boldsymbol{x}) \;=\; h_k\big(g_k(\boldsymbol{x})\big), \qquad k = 1, \ldots, d, \tag{15}$$

where $g_k : \mathbb{R}^n \to \mathbb{R}$ is a sparse inner function in a chosen post-hoc basis (monomials of total degree $\leq D$ in our experiments, with trigonometric or RBF columns equally admissible) and $h_k : \mathbb{R} \to \mathbb{R}$ is a smooth univariate outer function. The atoms surviving the sparse recovery of g_k are the dictionary entries the encoder discovered for coordinate k; their union across coordinates gives the full dictionary $\mathcal{T}$.

The decomposition proceeds in three steps.

Step 1: Inner function via Lasso. Let $\{\boldsymbol{x}_j\}_{j=1}^N$ be sample points on the attractor and let $f_j = f_k(\boldsymbol{x}_j) = [\mathcal{E}(\boldsymbol{x}_j)]_k$ be the encoder output computed by forward evaluation. We construct a polynomial design matrix $\Theta \in \mathbb{R}^{N \times P}$ whose P columns are all monomials $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ with $|\boldsymbol{\alpha}| \leq D$ (in practice $D = 3$). The inner function is recovered by solving the standardized ℓ_1-penalized regression of (9),

$$\hat{\boldsymbol{a}} \;=\; \arg\min_{\boldsymbol{a}} \; \frac{1}{2N} \big\| \Theta_s \, \boldsymbol{a} - \mathbf{f}_s \big\|_2^2 \;+\; \lambda \, \|\boldsymbol{a}\|_1,$$

on column-standardized data (subscript s), then unscaling to obtain

$$g_k(\boldsymbol{x}) \;=\; \sum_{|\boldsymbol{\alpha}| \leq D} \hat{a}_{\boldsymbol{\alpha}} \, \boldsymbol{x}^{\boldsymbol{\alpha}}.$$

Active terms $\mathcal{S}_k = \{\boldsymbol{\alpha} : |\hat{a}_{\boldsymbol{\alpha}}| > \delta\}$ (with sparsity threshold $\delta = 10^{-4}$, distinct from the encoder pruning threshold τ) form the dictionary support for coordinate k. When h_k is approximately affine—as occurs for most Lorenz latent coordinates—g_k already approximates f_k, and the surviving monomials directly identify the Koopman observables.

Step 2: Outer function via gradient identity. When h_k is genuinely nonlinear, the Lasso fit g_k alone has poor R^2 (e.g., $g \approx x^2 + y^2$ captures only the inner polynomial of $z = \sin(x^2 + y^2)$). To recover h_k we apply Proposition 1: if $f(\boldsymbol{x}) = h(g(\boldsymbol{x}))$ and $\nabla g(\boldsymbol{x}) \neq \mathbf{0}$, then $\nabla f = h'(g)\,\nabla g$, and projecting both sides onto ∇g gives the gradient identity (10),

$$h'\big(g(\boldsymbol{x})\big) \;=\; \frac{\nabla f(\boldsymbol{x}) \cdot \nabla g(\boldsymbol{x})}{\|\nabla g(\boldsymbol{x})\|^2}.$$

This identity arises by parameterizing points by arc length s along the gradient curves of g ($d\gamma/ds = \nabla g/\|\nabla g\|$), applying the multivariate chain rule to both $f(\gamma(s))$ and $g(\gamma(s))$, and eliminating s. The arc-length parameter cancels: the formula requires only pointwise evaluation of ∇f and ∇g, with no numerical integration of trajectories.

In practice we evaluate (10) at the N sample points:

1. $\nabla f_k(\boldsymbol{x}_j)$ is computed via automatic differentiation through the trained encoder $\mathcal{E}$;
2. $\nabla g_k(\boldsymbol{x}_j)$ is computed analytically from the polynomial coefficients $\hat{\boldsymbol{a}}$;
3. points with $\|\nabla g_k(\boldsymbol{x}_j)\|^2$ below the 5th percentile are discarded (ill-conditioned denominators), and outliers in h_k' beyond $3 \times \mathrm{IQR}$ are removed.

This yields a one-dimensional dataset $\{(g_j,\, q_j)\}_{j=1}^{N}$ where $g_j = g_k(\boldsymbol{x}_j)$ and $q_j \approx h_k'(g_j)$.

Step 3: Integration and reconstruction. We fit a polynomial $\hat{h}_k'(\zeta) = \sum_{l=0}^{p} b_l\,(\zeta - \bar{g}_k)^l$ to the binned medians of $\{(g_j, q_j)\}$ (80 bins; median per bin for robustness to residual outliers, where $\bar{g}_k$ denotes the sample mean of g_k), then integrate term-by-term:

$$\hat{h}_k(\zeta) \;=\; C + \sum_{l=0}^{p} \frac{b_l}{l+1}\,(\zeta - \bar{g}_k)^{l+1}, \tag{16}$$

where the constant C is fixed by $\hat{h}_k\big(g_k(\boldsymbol{x}_0)\big) = f_k(\boldsymbol{x}_0)$ at a reference point $\boldsymbol{x}_0$ near the median of g_k. If the polynomial reconstruction $\hat{h}_k \circ g_k$ achieves higher residual than a simple affine fit $f_k \approx a\,g_k + b$, we fall back to the affine model (indicating h_k is linear and g_k alone suffices).

The full reconstruction $\hat{f}_k(\boldsymbol{x}) = \hat{h}_k\big(g_k(\boldsymbol{x})\big)$ is evaluated on the sample set and scored by

$$R^2(h_k \circ g_k) \;=\; 1 \;-\; \frac{\sum_j \big(f_j - \hat{h}_k(g_j)\big)^2}{\sum_j \big(f_j - \bar{f}_k\big)^2},$$

where $\bar{f}_k$ is the sample mean of f_k. This metric separates the quality of the inner polynomial ($R^2(g_k)$, how well g_k alone fits f_k) from the gain contributed by the nonlinear outer function ($R^2(h_k \circ g_k) - R^2(g_k)$).

H Proof of Propositions

Proof. Consider curves $\gamma(s)$ parameterized by arc length satisfying $\dot{\gamma}(s) = \nabla g(\gamma(s))/\|\nabla g(\gamma(s))\|$. Along such a curve, let $G(s) = g(\gamma(s))$ and $F(s) = f(\gamma(s))$. Then:

$$\frac{dG}{ds} = \nabla g \cdot \frac{\nabla g}{\|\nabla g\|} = \|\nabla g\|, \tag{17}$$

$$\frac{dF}{ds} = \nabla f \cdot \frac{\nabla g}{\|\nabla g\|}. \tag{18}$$

Since $F(s) = h(G(s))$, the chain rule gives $dF/ds = h'(G(s))\,dG/ds = h'(G(s))\,\|\nabla g\|$. Equating with (18):

$$h'(g)\,\|\nabla g\| = \frac{\nabla f \cdot \nabla g}{\|\nabla g\|} \implies h'(g) = \frac{\nabla f \cdot \nabla g}{\|\nabla g\|^2}. \tag{19}$$

□

Proposition 2 (Single-pair regime for KAN level-set recovery)**.** *Let*

$$z_k(x) \;=\; \sum_{j=1}^{m} \Phi_{k,j}\big(u_j(x)\big), \qquad u_j(x) \;=\; \sum_{i=1}^{n} \psi_{j,i}(x_i),$$

be a latent coordinate of the trained two-layer KAN encoder, and let $\mathcal{A}_k := \{\, j : \Phi_{k,j} \not\equiv 0 \,\}$ denote its set of active channels. Suppose there exist a scalar field $g_k : \mathbb{R}^n \to \mathbb{R}$ with $\nabla g_k(x) \neq 0$ on the support of the data, and univariate functions $\alpha_{k,j} : \mathbb{R} \to \mathbb{R}$ for $j \in \mathcal{A}_k$, such that

$$u_j(x) \;=\; \alpha_{k,j}\big(g_k(x)\big) \qquad \text{for every } j \in \mathcal{A}_k. \tag{20}$$

Define

$$h_k(\xi) \;:=\; \sum_{j \in \mathcal{A}_k} \Phi_{k,j}\big(\alpha_{k,j}(\xi)\big).$$

Then $z_k(x) = h_k\big(g_k(x)\big)$, and the gradient identity

$$h_k'\big(g_k(x)\big) \;=\; \frac{\nabla z_k(x) \cdot \nabla g_k(x)}{\|\nabla g_k(x)\|^2} \tag{21}$$

recovers h_k exactly, up to an additive constant, along any integral curve of ∇g_k.

Proof. Substituting (20) into the definition of z_k and using that $\Phi_{k,j} \equiv 0$ for $j \notin \mathcal{A}_k$,

$$z_k(x) \;=\; \sum_{j \in \mathcal{A}_k} \Phi_{k,j}\big(u_j(x)\big) \;=\; \sum_{j \in \mathcal{A}_k} \Phi_{k,j}\big(\alpha_{k,j}(g_k(x))\big) \;=\; h_k\big(g_k(x)\big),$$

which establishes the decomposition. Applying the chain rule,

$$\nabla z_k(x) \;=\; h_k'\big(g_k(x)\big)\, \nabla g_k(x).$$

Taking the inner product of both sides with $\nabla g_k(x)$ and dividing by $\|\nabla g_k(x)\|^2 > 0$ yields (21). Integration of h_k' along any integral curve of ∇g_k recovers h_k up to the constant of integration. □

Corollary 1 (Sufficient conditions on the trained network)**.** *The hypothesis* (20) *of Proposition 2 holds in either of the following regimes:*

1. ***Channel sparsity.*** *The active set is a singleton, $\mathcal{A}_k = \{j^*\}$. Then $g_k = u_{j^*}$, $\alpha_{k,j^*} = \mathrm{id}$, and $h_k = \Phi_{k,j^*}$.*
2. ***Co-foliation.*** *The active inner functions $\{u_j\}_{j \in \mathcal{A}_k}$ share level sets: each u_j is constant on $\{g_k = c\}$ for every c in the data range. Then each u_j admits a univariate reparameterization $u_j = \alpha_{k,j} \circ g_k$.*

Remark 1 (Misspecification outside the single-pair regime)**.** *When no regime of Corollary 1 holds, no exact factorization $z_k = h_k \circ g_k$ exists, and the level-set Lasso in Equation* (9) *fits a misspecified model with positive residual*

$$\inf_{g,\, h} \; \mathbb{E}_x\Big[\big(z_k(x) - h(g(x))\big)^2\Big] \;>\; 0.$$

Empirically, this residual manifests as the small spurious monomial coefficients (e.g., x^2, y^2, z^2 at magnitudes $\lesssim 10^{-2}$) reported in Appendix Table 3, which provide a direct measure of the deviation from exact channel sparsity in the trained encoder.

Remark 2 (Cross-terms and empirical co-foliation)**.** *Hypothesis* (20) *can fail when z_k contains multiplicative cross-terms: for instance, the level sets of xy (hyperbolas) and $(x+y)^2$ (lines of slope -1) intersect transversally rather than co-foliating, so a network that represented $z_k = xy$ via the polynomial identity $xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2$ would fall outside the hypothesis of Corollary 1. This appears, in principle, to leave a theoretical gap for cross-term discovery. Empirically, however, the recovered observables in our experiments are dominated by linear coordinates whose level sets foliate the Lorenz attractor in parallel sheets (Figure 3), placing the recovery in the co-foliation regime of Corollary 1. The small cross-term coefficients in Table 3 ($\lesssim 10^{-2}$, 1–2 orders of magnitude below the linear terms) quantify the deviation from exact co-foliation and behave as the misspecification residuals discussed in Remark 1, rather than as a primary representational mechanism.*

I Attractor-restricted level-set theory

The chain-rule identity (Proposition 1) is exact whenever an exact factorization $f = h \circ g$ exists. The harder question is when such a factorization exists for the latent observables produced by the trained encoder. Proposition 2 gives a sufficient condition (co-foliation of active KAN channels), but Remark 2 notes that this condition fails for ambient cross-terms of the form xy vs. $(x+y)^2$. We resolve the apparent gap here by showing that the relevant geometry is not ambient: the dynamics live on a low-dimensional attractor $\mathcal{A} \subset \mathbb{R}^n$, and the factorization condition, correctly stated, involves only the intrinsic gradients $\nabla^{\mathcal{A}} f$ and $\nabla^{\mathcal{A}} g$ on $\mathcal{A}$. Generic ambient non-parallelism becomes generic intrinsic parallelism whenever the attractor has codimension at least one.

Let $\mathcal{A} \subset \mathbb{R}^n$ be a compact, connected, C^1-embedded d-dimensional submanifold ($1 \le d \le n-1$). For $x \in \mathcal{A}$, denote the tangent space $T_x\mathcal{A} \subseteq \mathbb{R}^n$ and normal space $N_x\mathcal{A} = (T_x\mathcal{A})^\perp$, with orthogonal projections $P_x : \mathbb{R}^n \to T_x\mathcal{A}$ and $P_x^\perp = I - P_x$. For $f \in C^1(\mathbb{R}^n)$ define the tangential and normal components of the ambient gradient:

$$\nabla^{\mathcal{A}} f(x) := P_x \nabla f(x), \qquad \nabla^{\perp} f(x) := P_x^{\perp} \nabla f(x), \qquad \nabla f(x) = \nabla^{\mathcal{A}} f(x) + \nabla^{\perp} f(x).$$

Equip $\mathcal{A}$ with a probability measure μ absolutely continuous with respect to the d-dimensional Hausdorff measure on $\mathcal{A}$. All L^2 norms below are with respect to μ. Throughout this appendix $f, g \in C^1(\mathbb{R}^n)$ with $\nabla^{\mathcal{A}} g(x) \neq 0$ for all $x \in \mathcal{A}$ (intrinsic regularity of g).

I.1 Intrinsic factorization

We separate the local condition (when does f admit a factorization in a neighborhood of each point of $\mathcal{A}$) from the global compatibility condition (when do these local factorizations glue to a global h).

Theorem 3 (Local intrinsic factorization)**.** *The following are equivalent at each $x \in \mathcal{A}$:*

(i) There exist a neighborhood $U \subseteq \mathcal{A}$ of x and $h \in C^1\big(g(U)\big)$ such that $f|_U = h \circ g|_U$.

(ii) $\ker\big(dg_x|_{T_x\mathcal{A}}\big) \subseteq \ker\big(df_x|_{T_x\mathcal{A}}\big)$.

(iii) $\nabla^{\mathcal{A}} f(x)$ is parallel to $\nabla^{\mathcal{A}} g(x)$.

When these hold, the local outer derivative is given by the intrinsic chain rule

$$h'\big(g(x)\big) \;=\; \frac{\langle \nabla^{\mathcal{A}} f(x),\, \nabla^{\mathcal{A}} g(x)\rangle}{\|\nabla^{\mathcal{A}} g(x)\|^2}. \tag{22}$$

Proof. (ii)$\Leftrightarrow$(iii). As d-dimensional 1-forms on $T_x\mathcal{A}$, $df_x|_{T_x\mathcal{A}}$ and $dg_x|_{T_x\mathcal{A}}$ are linearly dependent iff their kernels are nested. Since dg_x has kernel of dimension $d-1$ on $T_x\mathcal{A}$ by the assumption $\nabla^{\mathcal{A}} g \neq 0$, the inclusion in (ii) is equality, and linear dependence is equivalent (under the metric identification) to parallelism of the gradient vectors.

(i)$\Rightarrow$(ii). Differentiating $f|_U = h \circ g|_U$ along any $v \in T_x\mathcal{A}$ gives $df_x(v) = h'(g(x))\, dg_x(v)$, so $dg_x(v) = 0$ implies $df_x(v) = 0$.

(ii)$\Rightarrow$(i). Since $\nabla^{\mathcal{A}} g(x) \neq 0$, the implicit function theorem on $\mathcal{A}$ provides a chart $(g, \xi_2, \dots, \xi_d)$ on a neighborhood U of x. In these coordinates the kernel of dg is spanned by $\partial_{\xi_2}, \dots, \partial_{\xi_d}$; condition (ii) then reads $\partial f/\partial \xi_i \equiv 0$ for $i \ge 2$ on U. Hence $f|_U = h(g)$ with h a C^1 function of one variable.

Formula (22). Take $v = \nabla^{\mathcal{A}} g(x)$ in $df_x(v) = h'(g(x))\, dg_x(v)$. Since $\nabla^{\mathcal{A}} g$ is tangential, $df_x(v) = \langle \nabla f, \nabla^{\mathcal{A}} g\rangle = \langle \nabla^{\mathcal{A}} f, \nabla^{\mathcal{A}} g\rangle$, and $dg_x(v) = \|\nabla^{\mathcal{A}} g\|^2$. Dividing yields (22). □

Proposition 3 (Global compatibility)**.** *Suppose $\mathcal{A}$ is connected and condition (ii) of Theorem 3 holds at every $x \in \mathcal{A}$. Then a global $h \in C^1(g(\mathcal{A}))$ with $f|_{\mathcal{A}} = h \circ g|_{\mathcal{A}}$ exists if and only if*

$$f(x) = f(y) \quad \textit{whenever } x, y \in g^{-1}(c) \cap \mathcal{A} \textit{ for any } c \in g(\mathcal{A}). \tag{23}$$

The local pieces from Theorem 3 agree on overlapping charts whose level sets are connected; condition (23) *additionally enforces consistency across disconnected components of fibers.*

Proof. The local pieces h_U from Theorem 3 agree on chart overlaps within a single connected component of each fiber, by the local-constancy argument and continuity. A global h exists iff the values across distinct components of each fiber match, which is (23). □

In the empirical regime relevant to our method, condition (23) is enforced statistically by the $L^2(\mu)$-best approximation. The h recovered by the chain-rule estimator is implicitly the conditional expectation $h^*(c) = \mathbb{E}_\mu[f \mid g = c]$, and the residual variance $\mathbb{E}_\mu[\mathrm{Var}(f \mid g)]$ measures the joint failure of the local condition (ii) and the global condition (23). The reported $R^2(h \circ g)$ is precisely $1 - \mathbb{E}_\mu[\mathrm{Var}(f \mid g)]/\mathrm{Var}_\mu(f)$.

I.2 The ambient identity is the intrinsic identity plus a normal correction

The chain-rule formula in Proposition 1 uses the ambient gradient. We now show the discrepancy with the intrinsic identity (22) is a normal-bundle term that vanishes in well-understood regimes.

Theorem 4 (Ambient = intrinsic + normal correction)**.** *For any $f, g \in C^1(\mathbb{R}^n)$ and $x \in \mathcal{A}$ with $\nabla g(x) \neq 0$, define*

$$\alpha(x) := \frac{\|\nabla^{\mathcal{A}} g(x)\|^2}{\|\nabla g(x)\|^2} \in [0,1], \qquad \beta(x) := \frac{\|\nabla^{\perp} g(x)\|^2}{\|\nabla g(x)\|^2} = 1 - \alpha(x).$$

Let

$$R^{\mathcal{A}}(x) := \frac{\langle \nabla^{\mathcal{A}} f, \nabla^{\mathcal{A}} g\rangle}{\|\nabla^{\mathcal{A}} g\|^2}, \qquad R^{\perp}(x) := \frac{\langle \nabla^{\perp} f, \nabla^{\perp} g\rangle}{\|\nabla^{\perp} g\|^2}$$

be the intrinsic and normal projection ratios. Then the ambient identity satisfies

$$\frac{\langle \nabla f(x), \nabla g(x)\rangle}{\|\nabla g(x)\|^2} \;=\; \alpha(x)\, R^{\mathcal{A}}(x) + \beta(x)\, R^{\perp}(x). \tag{24}$$

If the intrinsic factorization $f|_{\mathcal{A}} = h \circ g|_{\mathcal{A}}$ holds (Theorem 3), then $R^{\mathcal{A}}(x) = h'(g(x))$ and

$$\underbrace{\frac{\langle \nabla f, \nabla g\rangle}{\|\nabla g\|^2}}_{\hat{h}'(x)\ \textit{(ambient identity)}} \;-\; h'\big(g(x)\big) \;=\; \beta(x)\,\big[R^{\perp}(x) - h'\big(g(x)\big)\big]. \tag{25}$$

Proof. The orthogonal decomposition $\nabla u = \nabla^{\mathcal{A}} u + \nabla^{\perp} u$ for $u \in \{f, g\}$ gives $\langle \nabla f, \nabla g\rangle = \langle \nabla^{\mathcal{A}} f, \nabla^{\mathcal{A}} g\rangle + \langle \nabla^{\perp} f, \nabla^{\perp} g\rangle$ (cross terms vanish since $\nabla^{\mathcal{A}} u \in T_x\mathcal{A} \perp N_x\mathcal{A} \ni \nabla^{\perp} u$), and $\|\nabla g\|^2 = \|\nabla^{\mathcal{A}} g\|^2 + \|\nabla^{\perp} g\|^2$. Dividing yields (24). Equation (25) follows by substituting $R^{\mathcal{A}} = h'(g)$ and using $\alpha + \beta = 1$. □

The residual (25) vanishes in three concrete situations: $\beta(x) \equiv 0$ (the inner function g has no normal gradient on $\mathcal{A}$); $R^{\perp}(x) \equiv h'(g(x))$ (the normal-projected ratio happens to coincide with the intrinsic one, e.g., because f admits a global ambient factorization); or $\nabla^{\perp} f \equiv 0$ on $\mathcal{A}$ (the encoder's extension off $\mathcal{A}$ is locally constant in directions normal to $\mathcal{A}$).

I.3 Cross-term obstruction is ambient, not intrinsic

We can now precisely diagnose the apparent failure of co-foliation for multiplicative cross-terms. Remark 2 observed that $\ker(d(xy))$ and $\ker(d((x+y)^2))$ intersect transversally in $\mathbb{R}^2$, so the polynomial identity $xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2$ does not give a co-foliation in the sense of Corollary 1(2). The next theorem shows that this is an obstruction in $\mathbb{R}^n$, not on the attractor.

Theorem 5 (Codimension reduction of the obstruction)**.** *Let $\mathcal{A} \subset \mathbb{R}^n$ be a compact, connected C^1-embedded d-submanifold. The pointwise intrinsic parallelism condition (Theorem 3, item (iii)) is strictly weaker than its ambient counterpart whenever $d < n$.*

1. *$d = 1$ **case (automatic local parallelism).** For any $f, g \in C^1(\mathbb{R}^n)$ with $\nabla^{\mathcal{A}} f \neq 0$ and $\nabla^{\mathcal{A}} g \neq 0$ on $\mathcal{A}$, the pointwise intrinsic parallelism condition holds automatically. Local factorization (Theorem 3(i)) therefore exists at every point. Global factorization additionally requires the compatibility condition* (23) *across disconnected fibers; this is non-trivial but is the only remaining obstruction.*

2. $d \geq 2$ ***case (codimension reduction).*** *The pointwise condition that $\nabla^{\mathcal{A}} f(x)$ and $\nabla^{\mathcal{A}} g(x)$ be parallel cuts out a set of codimension $d-1$ in $T_x\mathcal{A}\setminus\{0\} \times T_x\mathcal{A}\setminus\{0\}$. The corresponding ambient condition (parallelism in $\mathbb{R}^n$) has codimension $n-1$. The reduction by $n-d$ is exactly the codimension of $\mathcal{A}$ in $\mathbb{R}^n$.*

Proof. (1) For $d = 1$, $T_x\mathcal{A} = \operatorname{span}\{\tau(x)\}$ for a unit tangent τ. Then $\nabla^{\mathcal{A}} f = \langle \nabla f, \tau\rangle \tau$ and $\nabla^{\mathcal{A}} g = \langle \nabla g, \tau\rangle \tau$ are scalar multiples of the same vector; they are automatically parallel whenever both are nonzero, with proportionality constant $\lambda(x) = \langle \nabla f, \tau\rangle / \langle \nabla g, \tau\rangle$.

(2) Pointwise, $\nabla^{\mathcal{A}} f$ and $\nabla^{\mathcal{A}} g$ are vectors in the d-dimensional space $T_x\mathcal{A}$. The locus where they are parallel is the rank-≤ 1 subset of $(T_x\mathcal{A} \setminus \{0\})^2$, which has codimension $d-1$ (it requires the ratios of the $d-1$ component pairs in any basis of $T_x\mathcal{A}$ to coincide). The analogous ambient locus has codimension $n-1$, and the difference $n-d = \operatorname{codim}(\mathcal{A})$ is the extent to which intrinsic parallelism is more generic than ambient. □

I.4 Quantitative residual and predictions for the experiments

The Lasso-fit inner function g is determined by minimizing $\|f - g\|_{L^2(\mu)}$ on attractor samples, with L^1 sparsity penalty, over a finite-dimensional polynomial subspace. We use this to bound the residual (25) in terms of geometric quantities of the attractor and the polynomial approximation power.

Theorem 6 (Residual bound)**.** *Suppose $f|_{\mathcal{A}} = h \circ g|_{\mathcal{A}}$ holds intrinsically. Let $\hat{h}'(x) := \langle \nabla f(x), \nabla g(x)\rangle / \|\nabla g(x)\|^2$ denote the ambient chain-rule estimator. Then*

$$\left\|\hat{h}' - h'\circ g\right\|_{L^2(\mu)} \;\leq\; \overline{\beta}\left(\|R^{\perp}\|_{L^2(\mu)} + \|h'\circ g\|_{L^2(\mu)}\right), \tag{26}$$

where $\overline{\beta} := \sup_{x\in\mathcal{A}} \beta(x)$ is the maximum squared sine of the angle between ∇g and the tangent space.

Proof. Apply (25) pointwise, take L^2 norms, and use the triangle inequality together with $|\beta| \leq \overline{\beta}$. □

For a generic polynomial g on a d-dimensional attractor in $\mathbb{R}^n$, the expected value of $\beta(x)$ over $\mathcal{A}$ under uniform orientation of ∇g is $\mathbb{E}[\beta] = (n-d)/n$. We use this to predict the residual scale in each experiment.

System	n	d (Hausdorff)	$\mathbb{E}[\beta]$	Predicted residual	Observed FP scale
Lorenz	3	~ 2.06	0.31	$\lesssim 10^{-1.5}$	$\sim 10^{-2}$
Ikeda	2	~ 1.7	0.15	$\lesssim 10^{-2}$	10^{-2} to 10^{-1}
Standard map	2	~ 2	~ 0	$\to 0$	near machine ϵ
Arnold cat	2	2	0	no factorization	flat $\sim 4\times10^{-3}$

Table 8: Predicted vs. observed residual scales. The residual bound from Theorem 6 is multiplied by an empirical $h' \sim O(1)$. The Arnold cat case is special: $d = n = 2$, so $\mathcal{A}$ has zero codimension and the manifold framework does not apply; the flat coefficient profile is the diagnostic.

The predictions in Table 8 match the empirical scales reported in the main text. In particular: the Lorenz cross-term coefficients xy, xz at magnitude $\sim 10^{-2}$ in Table 3 sit at the predicted residual scale, which clarifies the role of these small coefficients. They are not noise to be discarded, nor are they the primary signal; they are the codimension-$(n-d)$ residual of the intrinsic factorization, which Theorem 5(2) predicts must be present whenever the attractor has codimension less than 1 (here, codim ≈ 0.94).

I.5 Training produces nearly co-foliating channels (informal)

Theorem 3 is a statement about whether a factorization exists for a given (f, g); it does not address why training yields encoders whose latent coordinates admit such factorizations with small residual. We sketch the connection without claiming a theorem.

The KAN encoder is trained to minimize a one-step prediction loss in the linear latent dynamics $z_{t+\Delta t} = \exp(K\Delta t)z_t$. The optimum aligns latent coordinates with eigendirections of the Koopman operator $\mathcal{K}$ on $L^2(\mu)$: any z_k minimizing the prediction residual is, in the limit, a finite-dimensional projection of $\mathcal{K}$-eigenfunctions. Eigenfunctions of $\mathcal{K}$ are Φ_t-equivariant, so their level sets form a flow-invariant foliation of $\mathcal{A}$. Two distinct eigenfunctions either share this foliation (degenerate eigenvalue) or define transverse foliations (distinct eigenvalues). The active channels of a single encoder coordinate z_k are coupled through the spline activation $\Phi_{k,j}$, and the dominant mode within z_k aligns with one Koopman eigendirection. This is the dynamical reason intrinsic co-foliation holds approximately in trained models.

J Ablations

Lasso λ	Jaccard ↑	Precision ↑	Recall ↑	FP
10^{-6}	**0.83**	**0.83**	**1.00**	1
10^{-5}	**0.83**	**0.83**	**1.00**	1
5×10^{-5}	0.71	0.71	**1.00**	2
10^{-4}	0.50	0.57	0.80	3
5×10^{-4}	0.50	0.57	0.80	3
10^{-3}	0.63	0.63	**1.00**	3
5×10^{-3}	0.50	0.57	0.80	3
10^{-2}	0.30	0.38	0.60	5

Table 9: Lasso regularization sweep for the pruned Deep-Koopman-KANDy model (seed 0, prune threshold 0.03). The Lasso parameter λ controls sparsity of the level-set polynomial decomposition $z_k \approx h \circ g$. Tight regularization ($\lambda \leq 10^{-5}$) recovers all five target terms with one false positive; stronger regularization suppresses the cross-terms xy and xz before eliminating spurious monomials.

Prune threshold	Edges pruned	MSE (retrained)	MSE ratio	Edges below τ
none	$0/40$	2.60×10^{-3}	$1.00\times$	—
0.01	$\sim 2/40$ (5%)	1.27×10^{-3}	$0.49\times$	2%
0.03	$\sim 7/40$ (18%)	2.55×10^{-3}	$0.98\times$	14%
0.05	$\sim 10/40$ (25%)	2.80×10^{-3}	$1.07\times$	24%

Table 10: Prune threshold sweep for the KAN encoder (architecture $[3, 5, 5]$, 40 total edges). Edges with KAN attribution score below the threshold are zeroed, then the model is retrained for 100 epochs. MSE ratio is relative to the unpruned baseline. The rightmost column shows the fraction of edges below τ across 12 independent models (3 data seeds $\times$ 4 model seeds).

K Readout Robustness on the Non-Polynomial Systems

The symbolic readout has four hyperparameters: the Lasso weight λ, the polynomial degree D, the readout basis, and the encoder pruning threshold τ. The Lorenz sweeps (Tables 9 and 10) cover the system with a sparse polynomial ground truth; here we sweep all four axes on the two systems where the readout is structurally misspecified: the Ikeda map (polynomial readout of a rational–trigonometric composition) and the standard map (readout in the embedded coordinates $(\cos\theta, \sin\theta, \cos p, \sin p)$). The λ, D, and basis axes are post-hoc: each setting re-runs the level-set decomposition on the same trained encoder, with no retraining, in seconds per setting (wall times below). The τ axis follows the Lorenz protocol (prune, retrain briefly at reduced learning rate, re-decompose). Before sweeping, we confirm that the base models reproduce the decompositions of Section 4: the retrained Ikeda $[2, 4, 4]$ encoder closely matches Table 4 (matched-quadratic ratio 0.656 vs. the reported 0.66), and the standard-map encoder recovers every atom of (12). The sweep tests decoupling quantitatively, as the recovered foliation coordinate should be stable across settings, with only the density of g and the complexity of h varying.

Ikeda: Lasso weight. Table 11 sweeps λ over four decades at $D = 3$ and $P = 10$ candidate monomials. Fit and density trade off smoothly and monotonically—$R^2(h \circ g)$ median from 0.976 to 0.908, active terms from 10 to 6, false positives from 4 to 1. The structural result is the invariance of the partition of latent coordinates: at every λ, the same two coordinates (z_1, z_3) carry the radial foliation coordinate (R^2 of the recovered g against a function of $r^2 = x^2 + y^2$ between 0.91 and 0.97), and the same two (z_0, z_2) do not (0.08–0.27). The matched-quadratic coefficient ratio on z_1 is not itself λ-invariant—it sweeps from 0.50 to 0.96 monotonically with λ—so the invariant is the identity of the radial coordinate, not the ratio's numerical value.

λ	$R^2(h \circ g)$ med / min	Active	FP	R^2_{radial}	ratio(z_1)	Wall (s)
10^{-6}	0.976 / 0.919	10	4.0	0.947	0.50	5.1
10^{-5}	0.976 / 0.919	10	4.0	0.947	0.50	3.7
5×10^{-5}	0.975 / 0.919	10	4.0	0.948	0.50	2.9
10^{-4}	0.975 / 0.919	10	4.0	0.949	0.51	2.7
5×10^{-4}	0.970 / 0.917	8.5	2.5	0.954	0.55	1.4
10^{-3} (default)	0.954 / 0.912	8.0	2.5	0.957	0.66	1.2
5×10^{-3}	0.931 / 0.871	7.0	1.5	0.965	0.85	0.2
10^{-2}	0.908 / 0.822	6.0	1.0	0.967	0.96	0.1

Table 11: Ikeda Lasso-λ sweep ($D = 3$, polynomial basis, $P = 10$). Active and FP are medians per latent coordinate; R^2_{radial} is the best-coordinate R^2 of the recovered g against a degree-4 polynomial in $r^2 = x^2 + y^2$ (basis-independent); ratio(z_1) is the matched $y^2 : x^2$ coefficient ratio of Table 4; Wall is the total decomposition time over all four coordinates.

Ikeda: polynomial degree. Table 12 sweeps D at $\lambda = 10^{-3}$. The decomposition breaks below the correct degree ($D = 1$: $R^2(h \circ g)$ median 0.44; $D = 2$: 0.79) and is flat above it in median ($0.954/0.965/0.948$ for $D = 3/4/5$), with the radial signature stable at 0.954–0.957. Overshooting D costs candidates (P doubles from 10 to 21) and wall time ($\sim 5\times$), but not accuracy, with the false-positive count saturating at 2.5 per coordinate. $D = 3$ is the elbow, and the practical rule is to increase D until $R^2(h \circ g)$ plateaus—the outer function h absorbs residual univariate nonlinearity, so a modest degree suffices even when the true dependence is non-polynomial.

D	P	$R^2(h \circ g)$ med / min	Active	FP	R^2_{radial}	s / coord
1	3	0.441 / 0.261	3.0	0.0	0.845	0.002
2	6	0.791 / 0.328	6.0	0.0	0.924	0.011
3	10	0.954 / 0.912	8.0	2.5	0.957	0.30
4	15	0.965 / 0.946	8.5	2.5	0.954	0.97
5	21	0.948 / 0.617	8.5	2.5	0.954	1.6

Table 12: Ikeda degree sweep ($\lambda = 10^{-3}$). P is the number of candidate monomials, $\binom{n+D}{D}$ for $n = 2$ (including the constant). The readout fails below the correct degree and is insensitive above it in median R^2; extra candidates raise cost, not false positives.

Ikeda: readout basis. Table 13 compares the misspecified polynomial basis against a correctly specified radial–trigonometric family (atoms in x, y, and $\cos t(r^2), \sin t(r^2)$ with $t(g) = 0.4 - 6/(1 + g)$) and generic Gaussian RBFs, all on the same trained encoder. Richer bases fit better and describe more densely (R^2 median $0.954/0.991/0.995$ at $8/12/17$ active atoms). Under the correct basis, the outer function collapses to affine on all four coordinates and g absorbs the composition, selecting exactly the Ikeda bilinear atoms (e.g., $x \cos t$, $x \sin t$, $y \cos t$, $y \sin t$). Two readings are important. First, the basis swap is post-hoc, and the misspecified basis loses only parsimony, supporting the decoupling. Second, among these four axes, the basis axis is the only one that changes the meaning of g with respect to the correct family; g is the full observable, not the foliation coordinate r^2. Therefore, the score based on radial recovery is not the correct metric there (0.931 vs. 0.957 for the polynomial basis) and, counter-intuitively, the polynomial basis in fact demonstrates the foliation coordinate more clearly, since Ikeda's misspecification is contained in the outer rational function, which is h, not in the inner coordinate.

Basis	P	$R^2(h \circ g)$ med / min	Active	R^2_{radial}	h affine
Polynomial D=3 (default, misspecified)	10	0.954 / 0.912	8.0	0.957	3/4
Radial–trigonometric (correct)	13	0.991 / 0.929	12.0	0.931	4/4
Gaussian RBF	37	0.995 / 0.993	17.0	0.931	3/4

Table 13: Ikeda readout-basis sweep ($\lambda = 10^{-3}$), post-hoc on the same encoder. A wrong basis costs parsimony; the correctly specified basis makes h affine on every coordinate.

Ikeda: pruning threshold. We compute the edge-attribution score of Appendix M for the Ikeda encoder and sweep $\tau \in \{0, 0.01, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5\}$, retraining at each setting (Table 14). The trained $[2, 4, 4]$ encoder has no edge with attribution below 0.19, so every threshold in the Lorenz-calibrated range ($\tau \leq 0.1$) prunes nothing and the recovered dictionary is identical. Pruning begins to degrade performance only at $\tau \geq 0.2$, where it costs 3–10$\times$ in one-step MSE even after retraining and permutes which latent index carries the radial coordinate, although the radial character itself survives (at $\tau = 0.2$ all four coordinates become radial, R^2_{radial} 0.93–0.98). The contrast with Lorenz, where $\tau = 0.03$ removes 18% of edges, is itself informative: pruning targets slack capacity, and the minimal Ikeda encoder has none.

τ	Edges pruned	MSE ratio	$R^2(h \circ g)$ med	Active	R^2_{radial}
0–0.1	0/24	1.00$\times$	0.954	8.0	0.957
0.2	2/24	6.4$\times$	0.994	7.5	0.979
0.3	3/24	3.1$\times$	0.976	8.0	0.998
0.5	9/24	9.7$\times$	0.964	7.0	0.948

Table 14: Ikeda prune-τ sweep (prune, retrain 100 epochs, re-decompose; $\lambda = 10^{-3}$, $D = 3$). All thresholds in the Lorenz-calibrated range $\tau \leq 0.1$ are below the encoder's minimum edge score (0.19) and leave the readout bit-identical. MSE ratio is one-step MSE relative to the unpruned model.

Standard map: Lasso weight and degree. On the standard map, the recovered dictionary is scored against the complete $|m| + |n| \leq 2$ Fourier family (the 14 monomials of degree ≤ 2 in the embedded coordinates, spanning (12)). The λ axis is stable across all eight values from 10^{-6} to 10^{-2} at the default degree $D = 2$, recall $= 1.000$, precision $= 1.000$, zero extra atoms, and $R^2(h \circ g)$ median 0.976–0.977. The degree axis is where candidate-set size matters (Table 15): $D = 1$ underfits (recall 0.29), $D = 2$ is exact, and for $D \geq 3$ recall stays at 1.000 while precision collapses from 1.000 to 0.126 as extra atoms grow roughly linearly in the candidate count (≈ 0.8 extra atoms per added candidate), yielding only $+0.023$ in $R^2(h \circ g)$. Over-provisioning the candidate set therefore costs parsimony, not accuracy; the true atoms are always found. A basis-independent check separates the two, where the span of the recovered g stays inside the $|m| + |n| \leq 2$ family ($R^2 \geq 0.978$ against that span) at every degree, so the extras are descriptive redundancy, not structural drift.

D	P	$R^2(h \circ g)$ med / p10	Active	Recall	Precision	Extra	$R^2_{\mathcal{F}_2}$
1	5	0.481 / 0.153	5.0	0.286	1.000	0	1.000
2 (default)	15	0.977 / 0.871	14.0	1.000	1.000	0	1.000
3	35	0.997 / 0.990	27.0	1.000	0.412	20	0.981
4	70	0.999 / 0.995	45.5	1.000	0.203	55	0.978
5	126	1.000 / 0.998	74.0	1.000	0.126	97	0.978

Table 15: Standard-map degree sweep ($\lambda = 10^{-5}$, 128 latent coordinates). Recall/precision/extras are scored against the complete $|m| + |n| \leq 2$ family; $R^2_{\mathcal{F}_2}$ is the (basis-independent) R^2 of the recovered g against the span of that family. Extra candidates never cost recall—only precision.

Standard map: readout basis. Table 16 repeats the decomposition in a direct Fourier dictionary ($e^{i(m\theta+np)}$, $|m| + |n| \leq 3$) and in periodic radial basis functions on the angles. The recovered coordinate stays inside the $|m|+|n| \leq 2$ span in all three bases ($R^2_{\mathcal{F}_2} \geq 0.978$)—including the radial basis function basis, which is not sparse at all (63 of 65 atoms active). Sparsity of the description

varies by 4.5-fold across bases while the described object does not move. The pruning axis is not applicable to this system's encoder because the attribution score is defined on the spline activation graph, and the standard-map model was not pruned; the τ axis is covered on Lorenz (Table 10) and Ikeda (Table 14).

Basis	P	$R^2(h \circ g)$ med / p10	Active	Recall	Precision	$R^2_{\mathcal{F}_2}$
Poly-in-trig D=2 (default)	15	0.977 / 0.871	14.0	1.000	1.000	1.000
Fourier $\lvert m\rvert + \lvert n\rvert \leq 3$	25	0.997 / 0.986	22.0	1.000	0.500	0.981
Periodic (von Mises) RBF	65	0.999 / 0.991	63.0	—	—	0.979

Table 16: Standard-map readout-basis sweep ($\lambda = 10^{-5}$), post-hoc on the same encoder. The recovered coordinate stays inside the $|m| + |n| \leq 2$ Fourier span regardless of which dictionary describes it.

Choosing the candidate set in practice. Because the readout is post-hoc, the inner-function candidate set is not a training-time commitment, and the sweeps above quantify what its selection costs. The practical procedure they support: (i) pick the basis family from the domain structure—polynomials for smooth vector fields, Fourier atoms for angular variables, radial atoms where a radial symmetry is plausible; (ii) increase D until $R^2(h \circ g)$ plateaus (the elbow sits at the correct degree on both systems; below it the factorization fails, above it only precision decays); (iii) set λ at the recall-saturation elbow as in Table 9; (iv) take τ from the edge-score distribution, where it either finds slack capacity (Lorenz) or verifiably nothing to prune (Ikeda). Each trial is a sparse regression costing well under a second per latent coordinate at $D \leq 3$ on these systems (Tables above; the design matrix has $P = \binom{n+D}{D}$ columns), so trying several bases is cheaper than a single training epoch.

L Prediction Benchmark against Prior-Art Methods

The main-text evaluation compares dictionary recovery, which requires a known finite target dictionary and is therefore meaningful only on Lorenz. To position the method against recent work on the axis those methods are designed for—prediction—we run a controlled forecasting benchmark against the prior-art Koopman methods that provide open-source code, across all five systems studied.

Prior art. We compare four models. **Ours** (Deep-Koopman-KANDy): a two-layer KAN autoencoder with the stable generator $K = \Omega - L^\top L$, rolled out autoregressively as $\mathcal{D} \circ M \circ \mathcal{E}$ (re-encoding every step). Nehma & Tiwari [47]: a faithful reimplementation of their core architecture—a KAN encoder/decoder with an unconstrained linear Koopman operator (no stable propagator)—rolled out in their native pure-latent linear mode $\mathcal{D}(M^n \mathcal{E}(\boldsymbol{x}_0))$; architecturally this coincides with our *2L-KAN + unconstrained* ablation (Table 1). Coote & Colbrook [19]: the ResDMD trainable dictionary (their PyResDMD package, used directly), a ReLU dictionary optimized against spectral residuals, with an EDMD operator and least-squares state read-out. EDMD-DL [41]: dictionary $[\,\boldsymbol{x},\, \mathrm{MLP}(\boldsymbol{x})\,]$ with a closed-form EDMD operator. Each baseline uses its own native roll-out; we do not grant the baselines our re-encoding scheme. The two remaining recent methods we discuss (Bolager et al. [7]; Schurig et al. [56]) do not provide public code and are omitted.

Why prediction, not dictionary recovery. None of the baselines produces a symbolic dictionary, so a Jaccard-style recovery score is undefined for them (the same reason the main text restricts recovery comparisons to Lorenz). We therefore report: the one-step normalized MSE (squared error of $\hat{\boldsymbol{x}}_{t+1}$, divided by per-coordinate variance); and the valid-prediction time $\mathrm{VPT}_{0.5}$, the first roll-out step at which the mean NRMSE exceeds 0.5, in steps and—where a clean largest Lyapunov exponent exists—in Lyapunov times τ_L. We deliberately report $\mathrm{VPT}_{0.5}$ rather than a divergence-threshold horizon: a bounded-but-inaccurate roll-out (the EDMD family on chaotic data) can hover just below $\mathrm{NRMSE} = 1$ indefinitely, so a $\mathrm{VPT}_{1.0}$ would credit it with an "infinite" horizon despite an order-of-magnitude worse per-step error. All numbers are medians over three seeds; the two torus maps are embedded as $(\cos, \sin) \in \mathbb{R}^4$.

System	Method	Params	1-step nMSE ↓	$\mathrm{VPT}_{0.5}$ (steps) ↑	(τ_L)
Lorenz	Ours	3,296	3.3×10^{-5}	**127**	1.15
	Nehma & Tiwari	3,296	$\mathbf{3.0 \times 10^{-5}}$	25	0.23
	Coote & Colbrook	19,472	1.3×10^{-3}	18	0.16
	EDMD-DL	4,741	4.4×10^{-2}	13	0.12
Cat map†	Ours	9,536	1.1×10^{-3}	**5**	4.8
	Nehma & Tiwari	9,536	$\mathbf{1.8 \times 10^{-4}}$	2	1.9
	Coote & Colbrook	29,976	7.4×10^{-2}	2	1.9
	EDMD-DL	5,000	8.3×10^{-1}	1	1.0
Standard map	Ours	12,544	2.1×10^{-4}	**8**	—
	Nehma & Tiwari	12,544	$\mathbf{7.0 \times 10^{-5}}$	3	—
	Coote & Colbrook	39,968	3.3×10^{-2}	3	—
	EDMD-DL	5,000	1.4×10^{-4}	**8**	—
Ikeda	Ours	3,136	5.7×10^{-3}	**5**	—
	Nehma & Tiwari	3,136	$\mathbf{1.6 \times 10^{-3}}$	3	—
	Coote & Colbrook	18,960	5.7×10^{-1}	1	—
	EDMD-DL	4,677	3.3×10^{-3}	**5**	—
KS (64-D)	Ours	126,976	$\mathbf{5.9 \times 10^{-4}}$	**41**	0.44
	Nehma & Tiwari	126,976	6.0×10^{-4}	20	0.21
	Coote & Colbrook	202,816	3.8×10^{-1}	14	0.15
	EDMD-DL	27,928	3.8×10^{-1}	14	0.15

Table 17: Prediction benchmark against prior-art dictionary learning Koopman methods (medians over 3 seeds). One-step nMSE is the variance-normalized one-step error; $\mathrm{VPT}_{0.5}$ is the first roll-out step at which mean NRMSE exceeds 0.5. Each method uses its native roll-out. Best per system in **bold**. †The cat map is a negative control (no finite Koopman closure); the τ_L column is a within-system reference only, since τ_L differs across systems (e.g., ≈ 1 iterate on the cat map). Lorenz models here are larger than the $[3, 5, 5]$ ablation model of Table 1.

Comparison to prior art. Table 17 supports three conclusions. First, the KAN autoencoder lifting outperforms classical EDMD dictionaries on per-step accuracy: the two KAN models (ours and the Nehma–Tiwari reimplementation) are one to three orders of magnitude more accurate one-step than the ResDMD and EDMD-DL dictionaries on several systems—by $\sim 600\times$ on the 64-dimensional KS field (6×10^{-4} vs. 3.8×10^{-1}). Second, ours attains the best or tied-best reliable horizon $\mathrm{VPT}_{0.5}$ on all five systems, with the largest margins on dissipative chaotic systems (Lorenz 127 vs. 25 steps; KS 41 vs. 20): the stable generator together with autoregressive re-encoding yields the most durable forecast. Third, on one-step accuracy the stable constraint is not universally optimal: the unconstrained KAN (Nehma–Tiwari) is more accurate one-step on four of five systems, most clearly on the near-measure-preserving maps (standard map 7.0×10^{-5} vs. 2.1×10^{-4}; Ikeda 1.6×10^{-3} vs. 5.7×10^{-3}), as expected: the negative-semidefinite constraint on $\mathrm{sym}(K)$ is an inductive bias suited to dissipative attractor dynamics, and is neutral-to-mildly costly on volume-preserving maps (it can still represent the conservative boundary $|\lambda| = 1$ at $L = 0$, but training need not drive it there). Even there, Deep-Koopman-KANDy retains the longer reliable horizon. On the cat-map negative control, every method is limited in absolute terms, consistent with the absence of a finite-dimensional Koopman closure, though ours still leads via re-encoding.

These benchmarks show that the KAN encoder/decoder increases per-step accuracy over classical dictionary learning, and the stable generator buys the most reliable forecast horizon across dynamical regimes—while the unconstrained variant remains a competitive one-step predictor on conservative maps, exactly where a dissipative bias is not called for.

M Pruning criterion: the edge-attribution score

The encoder/decoder pruning step of Section 3 removes low-importance spline edges using the edge-attribution score of the KAN implementation [42]. Because this criterion is easily mistaken for an ℓ_1 penalty on spline coefficients, we give its exact definition and how it is computed in practice. In

a sense, it is a data-dependent, variance-normalized relevance propagated backwards from the latent outputs.

Two data-dependent primitives. Index the encoder layers by l and write an edge as $(j \to i)$, from lower node j to upper node i. During a forward pass over a data batch $\{\boldsymbol{x}^{(b)}\}_{b=1}^{B}$ the implementation caches, per layer, two non-negative statistics:

$$E_l[i,j] = \mathrm{std}_b\Big(\phi_{i,j}^{(l)}\big(x_j^{(b)}\big)\Big) \qquad \text{(edge activation scale)}, \tag{27}$$

$$S_l[i] = \mathrm{std}_b\Big(u_i^{(l)}\big(\boldsymbol{x}^{(b)}\big)\Big) \qquad \text{(node activation scale)}, \tag{28}$$

where $\phi_{i,j}^{(l)}$ is the spline (plus residual) activation on edge $(j \to i)$ and $u_i^{(l)} = \sum_j \phi_{i,j}^{(l)}$ is the aggregated value at node i before its affine transform. Both are standard deviations over the data batch, not functions of the parameters alone.

Backward relevance propagation. Relevance is propagated from the d latent coordinates downward. Seed the output layer L with the identity, $a_{k,i}^{(L)} = \delta_{k,i}$ (each latent output k assigns unit relevance to itself). For $l = L, \ldots, 1$,

$$r_{k,i,j}^{(l)} = \frac{E_l[i,j]}{S_l[i] + \varepsilon}\, a_{k,i}^{(l)}, \qquad a_{k,j}^{(l-1)} = \sum_i r_{k,i,j}^{(l)}, \qquad \varepsilon = 10^{-4}. \tag{29}$$

The ratio $E_l[i,j]/(S_l[i] + \varepsilon)$ measures the activation scale of edge $(j \to i)$ relative to that of its target node. Multiplying this ratio by $a_{k,i}^{(l)}$ weights the relevance passed backward through the edge. Summing over the target nodes i gives the relevance assigned to node j in the preceding layer. These weights compare activation scales and need not sum to one.

Edge importance and the prune rule. The reported importance of edge $(j \to i)$ averages its relevance over the d latent coordinates,

$$B_{i,j}^{(l)} = \frac{1}{d} \sum_{k=1}^{d} r_{k,i,j}^{(l)}, \tag{30}$$

and the edge is removed—its activation frozen to zero—when $B_{i,j}^{(l)} \le \tau$, retained otherwise; edges already pruned in a prior pass stay pruned. After masking, the model is retrained briefly with the mask held fixed. In the main text we drop the layer index and write $B_{i,j}$ for the encoder's first layer.

Interpretation and practice. Equations (29) and (30) define a layerwise relevance propagation attribution on the variance of activations. During propagation, latent outputs diffuse relevance backward to the various edges in the layer, splitting the relevance proportionally to the edge's influence $E_l[i,j]/S_l[i]$ on the node it feeds; an edge is removed if, averaged over the latent coordinates, it explains little of the variability of the node it feeds. Two consequences are important. (i) The score depends on the data: it is computed on the most recent batch, so pruning reflects the sampled attractor. (ii) It is distinct from an ℓ_1 penalty on spline coefficients during training, a different mechanism that the same networks support (see Table 1 ℓ_1 variant). Across the 12 encoders in Table 10 (3 data sets and 4 model seeds), $\tau = 0.03$ is near the 14th percentile of edge scores and consistently removes the same low-importance edges in all runs, so it behaves as a low-percentile constant. The target dictionary is recovered at unit recall without pruning (Table 1, ℓ_1 row); pruning, however, removes false positives.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract claims we make progress in estimating required dictionaries, but it does not totally solve the problem. Additionally, the KAN encoder heads combined with the Koopman model do, in fact, help learn the dynamics, as seen in the Standard Map and Cat Map examples.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We acknowledge the key limitation that you still need a dictionary for the Lasso step.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main proof for our gradient identity and when the method is successful are in the Appendix H and I.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Between the main text and the supplement, we provide all the details needed for reproduction. Our experiments use two classes of KAN models written in two frameworks, both of which are available at https://github.com/Center-For-Complex-Systems-Science/Deep-Koopman-KANDy.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A GitHub repository containing all experimental code, training scripts, and configuration files is referenced via a footnote in the text 4. The repository includes scripts to reproduce every figure and table in the paper, along with a README that documents the environment setup and the exact commands. Our paper includes a footnote with a link to a repository containing all experiments, end-to-end. The repository is publicly available at https://github.com/Center-For-Complex-Systems-Science/Deep-Koopman-KANDy.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and the appendices provide hyperparameters, data generation, splits, and training details; the KAN implementation is available in our repository: https://github.com/Center-For-Complex-Systems-Science/Deep-Koopman-KANDy.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the mean ± one standard deviation over five random seeds for the main Lorenz ablation metrics in Table 1, including NRMSE and dictionary-recovery Jaccard scores. The reported variation reflects seed-to-seed variability across independent training runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All experiments were run on a single NVIDIA Quadro RTX 4000 GPU (8 GB VRAM, CUDA 12.2). Wall-clock times for the main Lorenz comparison are reported in Table 1 (ranging from 24 minutes for the MLP baseline to 11.1 hours for SINDy-AE). Total compute for all reported experiments, including five-seed ablations, was approximately 22 GPU-hours. Preliminary and discarded experiments during development consumed an additional 60 GPU-hours.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: There are no ethical considerations required for physics problems.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work develops a methodology for discovering interpretable observables of physical dynamical systems (Lorenz, Ikeda, area-preserving maps). It is foundational research in scientific machine learning with no direct path to deployed applications, and we do not foresee specific societal harms or benefits beyond those generic to improvements in scientific computing tools.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: All experiments were performed on physics simulations and no safeguards are required.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use PyTorch (BSD-3), NumPy and scikit-learn (BSD-3), and PyKAN (MIT). The ResDMD baseline uses the PyResDMD package of Coote & Colbrook [19] (`https://github.com/GeorgeCoote/pyresdmd`), which has no license; we used it unmodified for benchmarking only and do not redistribute it with our code. All simulations were generated by the authors.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code under the MIT license, with a README documenting installation, data generation, and the commands to reproduce each experiment.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects in this study.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No IRB approval required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not core to the method of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.