Data-driven model discovery with Kolmogorov-Arnold networks

Mohammadamin Moradi,[1, *] Shirin Panahi,[1, *] Erik M. Bollt,[2] and Ying-Cheng Lai[1, 3, †]

[1] *School of Electrical, Computer, and Energy Engineering,*
Arizona State University, Tempe, AZ 85287, USA
[2] *Department of Electrical and Computer Engineering, Clarkson Center for Complex Systems Science,*
Clarkson University, Potsdam, New York 13699, USA
[3] *Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Dated: September 9, 2024)

Data-driven model discovery of complex dynamical systems is typically done using sparse optimization, but it has a fundamental limitation: sparsity in that the underlying governing equations of the system contain only a small number of elementary mathematical terms. Examples where sparse optimization fails abound, such as the classic Ikeda or optical-cavity map in nonlinear dynamics and a large variety of ecosystems. Exploiting the recently articulated Kolmogorov-Arnold networks, we develop a general model-discovery framework for any dynamical systems including those that do not satisfy the sparsity condition. In particular, we demonstrate non-uniqueness in that a large number of approximate models of the system can be found which generate the same invariant set with the correct statistics such as the Lyapunov exponents and Kullback–Leibler divergence. An analogy to shadowing of numerical trajectories in chaotic systems is pointed out.

Discovering the model of a system from observational or measurement data has been a fundamental problem since the beginning of science. For nonlinear dynamical systems, data-driven identification, and forecasting have attracted a great deal of research in the past four decades [1–37]. A diverse array of methodologies have been developed, e.g., calculating the information contained in sequential observations to deduce the deterministic equations [2], approximating a nonlinear system by a large collection of linear equations [1, 7, 12], fitting differential equations to chaotic data [9], exploiting chaotic synchronization [15] or genetic algorithms [17, 27], inverse Frobenius-Perron approach to designing a dynamical system "near" the original system [19], or using the least-squares best approximation for modeling [26]. An approach that has gained considerable interest is sparse optimization, where the system functions are assumed to have a sparse structure in that they can be represented by a small number of elementary mathematical functions, e.g., a few power- and/or Fourier-series terms. What is needed then is to estimate the coefficients associated with these terms. In a high-order series expansion, the coefficients with the vast majority of the terms are zero, except for a few. The problem of finding these nontrivial coefficients can then be naturally formulated [28, 38] as a compressive-sensing problem [39–43]. Under the same idea, a popular method was later developed [44, 45].

The sparse-optimization approach is effective for systems whose governing equations are sufficiently simple in the sense of sparsity, such as the chaotic Lorenz [46] and Rössler [47] oscillators whose velocity fields contain a small number of low-order power-series terms. However, sparsity can be self-sabotage because, while it is the reason that the approach is powerful, it also presents a fundamental limitation: it works only if the system equations do in fact have a sparse structure. Dynamical systems violating the sparsity condition arise in physical and biological situations. A known example is the Ikeda map that describes the propagation of an optical pulse in a cavity with a nonlinear medium [48, 49], whose functions contain an infinite number of series expansion terms. Many ecological systems and gene-regulatory circuits whose governing equations have a Holling-type of structure [50, 51] also violate the sparsity condition [52]. For these systems, the sparse-optimization approach to model discovery fails absolutely and completely.

In this Letter, we articulate an entirely different approach to discovering the models of any dynamical systems including those that do not meet the sparsity condition. The idea exploits Kolmogorov-Arnold networks (KANs), a recent computational framework for representing sophisticated mathematical functions [53] based on the classical Kolmogorov-Arnold theorem [54–56] that any multivariate mathematical function can be decomposed as a sum of single-variate functions, as illustrated in Fig. 1(a). KANs decompose complex high-dimensional problems into simpler, more manageable univariate functions, allowing for more efficient training and better interpretability of the machine-learning model, addressing some of the limitations in traditional neural networks such as the black-box nature and computational inefficiency. As a result, KANs are rapidly gaining attention as a promising alternative in machine learning.

In contrast to a standard neural network with thousands and perhaps millions of weights and biases but always fixed all the same activation functions say *atan* or *ReLu*, a KAN is a small network of say a dozen nodes but each different and carefully designed activation functions. We consider a dynamical system described by $d\mathbf{x}/dt = \mathbf{F}(\mathbf{x})$ or alternatively by $\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n)$, including where $\mathbf{F}(\mathbf{x})$ *does not* possess a sparse structure. Our goal is to find an approximation of $\mathbf{F}(\mathbf{x})$, de-

[*] The first two authors contributed equally to this work
[†] Ying-Cheng.Lai@asu.edu

noted as $\mathbf{G}(\mathbf{x})$, such that the system $d\mathbf{x}/dt = \mathbf{G}(\mathbf{x})$ or $\mathbf{x}_{n+1} = \mathbf{G}(\mathbf{x}_n)$ produces the identical dynamical behaviors as the original system (e.g., the same attractor with the same statistical and dynamical invariants to within certain numerical precision). We demonstrate, using the Ikeda map and a chaotic ecosystem as illustrative examples, that such a function $\mathbf{G}(\mathbf{x})$ in an implicit form can indeed be found by the KANs.

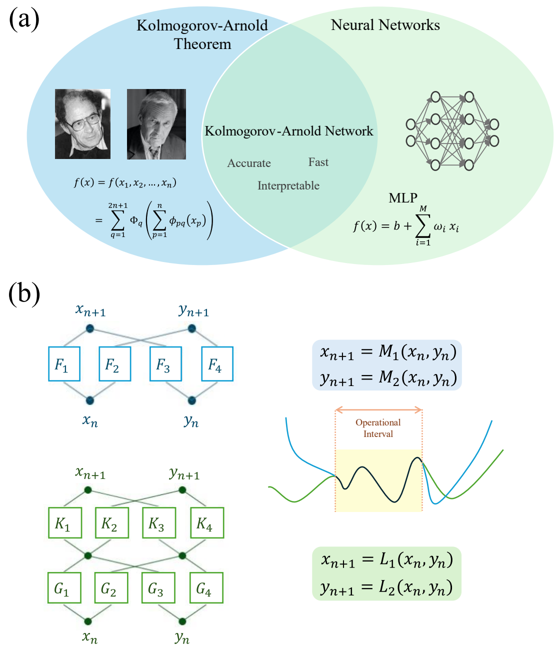
(a)
Kolmogorov-Arnold Theorem
Neural Networks
Kolmogorov-Arnold Network
Accurate
Fast
Interpretable
$f(x) = f(x_1, x_2, \dots, x_n)$
$= \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{pq}(x_p) \right)$
MLP
$f(x) = b + \sum_{i=1}^{M} \omega_i x_i$
(b)
x_{n+1}
y_{n+1}
F_1
F_2
F_3
F_4
x_n
y_n
$x_{n+1} = M_1(x_n, y_n)$
$y_{n+1} = M_2(x_n, y_n)$
Operational Interval
K_1
K_2
K_3
K_4
G_1
G_2
G_3
G_4
$x_{n+1} = L_1(x_n, y_n)$
$y_{n+1} = L_2(x_n, y_n)$

FIG. 1. Basics of KAN. (a) Kolmogorov-Arnold theorem and neural network. (b) Schematic illustration of two different structures (blue and green) leading to two different functions $\mathbf{M}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x})$ that generate the same dynamics as $\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n)$ in the relevant phase-space domain.

The interpretability of the KAN structure lies in its accessibility to the internal mechanisms of the model, such as the activation functions $F_i(\mathbf{x})$, $G_i(\mathbf{x})$, and $K_i(\mathbf{x})$ in Fig. 1(b). Unlike the conventional machine-learning methods, which often operate as "black boxes," KANs provide a more transparent view of how inputs are transformed into outputs. This transparency allows for a better understanding of the underlying dynamics and how each function influences the system's behavior, thereby making them more interpretable than conventional methods. Figure 1(b) presents two different KAN structures highlighted in blue and green. The blue KAN has two inputs and two outputs without any hidden nodes, where the functions $M_1 = F_1(\mathbf{x_n}) + F_3(\mathbf{y_n})$ and $M_2 = F_2(\mathbf{x_n}) + F_4(\mathbf{y_n})$ are linear combinations of the activation functions F_i for $i = 1, \cdots, 4$. The green structure has two extra hidden nodes where $L_1 = K_1\big(G_1(\mathbf{x_n}) + G_3(\mathbf{y_n})\big) + K_3\big(G_2(\mathbf{x_n}) + G_4(\mathbf{y_n})\big)$ and $L_2 = K_2\big(G_1(\mathbf{x_n}) + G_3(\mathbf{y_n})\big) + K_4\big(G_2(\mathbf{x_n}) + G_4(\mathbf{y_n})\big)$. Both structures produce the same dynamics in the relevant phase-space domain (yellow shaded area), where the dynamics outside of this domain can be different. This concept will be elucidated below with a concrete example.

From the standpoint of data-driven model discovery, the Ikeda map represents perhaps the most difficult kind of system - so far there has been no success with any sparse optimization method. The two-dimensional map is given by [48, 49] $x_{n+1} = 1 + \mu\left(x_n \cos(\phi_n) - y_n \sin(\phi_n)\right)$ and $y_{n+1} = \mu\left(x_n \sin(\phi_n) + y_n \cos(\phi_n)\right)$, where $\phi_n = 0.4 - 6(1 + x_n^2 + y_n^2)^{-1}$ and μ is a bifurcation parameter. (We fix $\mu = 0.9$, so that the map generates a chaotic attractor in the phase-space domain $(x \in [-1, 2], y \in [-2.5, 1])$. Sparse optimization fails spectacularly for this system because in either the power- or the Fourier-series expansions or a combination of both, an infinite number of terms are required to represent each map function - see Supplementary Information (SI) for more details [57].

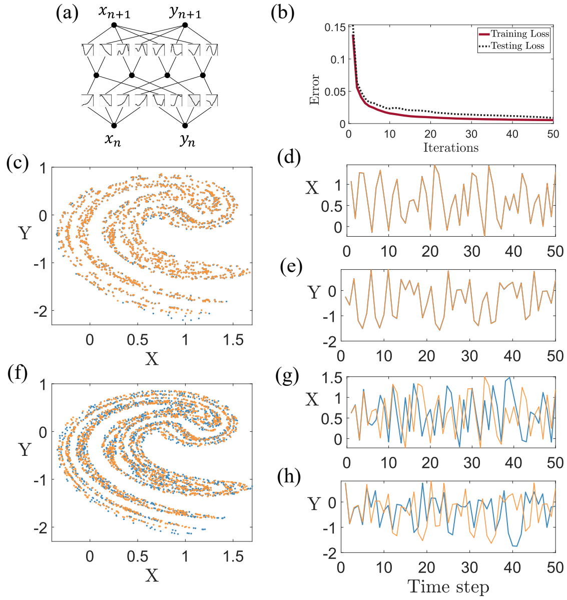

FIG. 2. KANs applied to the Ikeda map. (a) A KAN structure with 2 input, 4 hidden, and 2 output nodes. (b) Training (red) and testing (black dashed) loss curves. (c) Chaotic attractor during the training phase (blue - ground truth; orange - KAN produced). (d,e) Time series during the training. The blue and orange traces overlap well, signifying a high training accuracy. (f) Chaotic attractor during testing (blue - ground truth; orange - KAN produced). (g,h) The corresponding time series. While the predicted time series diverges from the ground truth after a few iterations due to chaos, the KAN generates the correct attractor in the pertinent phase-space domain. The true Lyapunov exponents of the chaotic attractor are $[0.5025, -0.7263]$. The KAN predicted model gives the values of the two exponents as $[0.5075, -0.7182]$, agreeing with the ground truth.

We first use a $[2, 4, 2]$ KAN structure, as shown in Fig. 2(a), which has 2 input, 4 hidden, and 2 output nodes. The time-series data contain 10^4 points, with 80% allocated for training and the remaining 20% for testing. The training process contains 50 iterations with the following hyperparameter values: $k = 3$ (cubic B-splines), grid size $G = 10$ for the splines, regularization parameters $\lambda = 0$ and $\lambda_{\text{entropy}} = 10$, learning rate 0.1, and a zero initial random seed. (see SI [57] for a detailed description of these hyperparameters). Training is administered in a feedforward process in which the KAN is trained to minimize the difference between the input and output so as to predict the evolution of the Ikeda map into the future with the input of the dynamical variables from the past. The training loss as a function of time is shown as the red curve in Fig. 2(b), and the KAN-produced attractor and time series during the training phase in comparison with the ground truth are shown in Figs. 2(c-e), respectively. The training loss decreases rapidly to zero, indicating high training accuracy and efficiency with skill. For the testing phase, we use the same set of parameter values but replace the original input data point with the output of the KAN at each iteration. The testing loss is shown in Fig. 2(b) as the black dashed curve, and the KAN predicted attractor and time series are shown in Figs. 2(f-h), respectively. While the KAN-predicted time series diverges from the ground truth after a few iterations due to chaos, the predicted attractor agrees with the ground truth well, indicating that the KAN has generated the correct model of the Ikeda map.

To demonstrate that a KAN can be readily modified to generate a different representation of the Ikeda map but with the same chaotic attractor, we construct a more sophisticated architecture than the one in Fig. 2(a), as shown in Fig. 3(a). The training and prediction results are shown in Figs. 3(b-h).

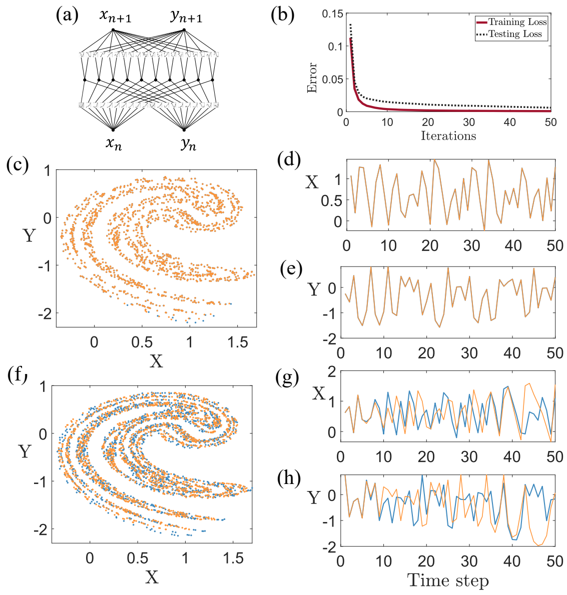

FIG. 3. A KAN configuration generating a different representation of the Ikeda map but with the same chaotic attractor. The KAN has 2 input, 10 hidden, and 2 output nodes. Legends are the same as those in Fig. 2. The two Lyapunov exponents of the KAN predicted model are $[0.5033, -0.7311]$, which again agrees with the true exponents.

For generality, we now present results from a continuous-time system, a chaotic ecosystem [58] of three dynamical variables: $\dot{N} = N\,(1 - N/K) - x_p y_p NP/(N + N_0)$, $\dot{P} = x_p P\,(y_p N/(N + N_0) - 1) - x_q y_q PQ/(P + P_0)$, and $\dot{Q} = x_q Q\,(y_q P/(P + P_0) - 1)$, where N, P, and Q are the populations of the primary producer, the herbivore, and the carnivore, respectively, and the bifurcation parameter K is the carrying capacity. For $K = 0.98$ and other parameters set as $x_p = 0.4$, $y_p = 2.009$, $x_q = 0.08$, $y_q = 2.876$, $N_0 = 0.16129$, and $P_0 = 0.5$, the system exhibits a chaotic attractor [58]. A power-series expansion of the velocity field contains an infinite number of terms, violating the sparsity condition - see SI for more details [57].

Our KAN architecture has a $[3, 3]$ structure (3 input and 3 output nodes, no hidden nodes), as illustrated in Fig. 4(a). The neural network was trained using 10,000 data points of sampling interval $\delta t = 0.5$ (corresponding to about 1,155 cycles of oscillation), with 90% of the data allocated for training and the remaining 10% for testing. The training process involved 100 iterations for the following hyperparameter values: cubic B-spline ($K = 3$), grid size $G = 3$, $\lambda = 0$, $\lambda_{\text{entropy}} = 10$, learning rate 0.5, and a zero initial random seed. Figure 4(b) shows the rapid decrease in the training and testing loss with increasing epochs. The KAN generated attractor and the corresponding time series during the training phase are shown in Figs. 4(c-f), where a comparison with the ground truth indicates successful training. The KAN attractor and the time series generated during the testing phase are shown in Figs. 4(g-j), demonstrating the KAN's forecasting power. The Lyapunov exponents of the attractor are consistent with the true values. (Detailed comparative results for the power spectra, correlation dimension and three types of distance divergences are provided in SI [57].)

To gain insights into the meaning of the interpretability of the KAN-discovered models, we offer a mathematical scheme to interpret machine-learning modeling errors of as representing the true underlying system. The issue of considering models that produce realistic data, even with orbital errors, is general. In our case, the KAN model $\mathbf{G}$ is said to produce identical behavior as the true system $\mathbf{F}$ if numerically computed orbits of $\mathbf{G}$ shadow some true orbits of $\mathbf{F}$, at least for the observed finite time of the data set. For maps, if a true orbit of $\mathbf{F}$ is a sequence $\text{Orbit}_{\mathbf{F}}(\mathbf{x}_0) = \{\mathbf{x}_0, \mathbf{F}(\mathbf{x}_0), \mathbf{F}^2(\mathbf{x}_0)...\} \equiv \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, ...\}$, it is unreasonable to expect that a good but imperfect model $\mathbf{G}$ will produce an orbit, denoted as $\text{Orbit}_{\mathbf{G}}(\mathbf{x}_0) = \{\mathbf{x}_0, \mathbf{G}(\mathbf{x}_0), \mathbf{G}^2(\mathbf{x}_0)...\} \equiv \{\mathbf{x}_0, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ...\}$, that stays close to $\text{Orbit}_F(\mathbf{x}_0)$. If the model is good in the sense that a pointwise error $e(x) = |\mathbf{G}(x) - \mathbf{F}(x)|$ on the domain $\mathbf{x} \in \mathcal{D}$ satisfies in terms of the sup-norm, $\|e\|_\infty := \sup_{\mathbf{x}\in\mathcal{D}} |e(\mathbf{x})| < \epsilon$ for some small $\epsilon > 0$, then at each step of the model the error is small: $\tilde{\mathbf{x}}_{i+1} = \mathbf{G}(\tilde{\mathbf{x}}_i) = F(\tilde{x}_i) + \epsilon_i$ and with each step error, $0 \leq |\epsilon_i| < \epsilon$. Nonetheless a small normed error of the function difference be-

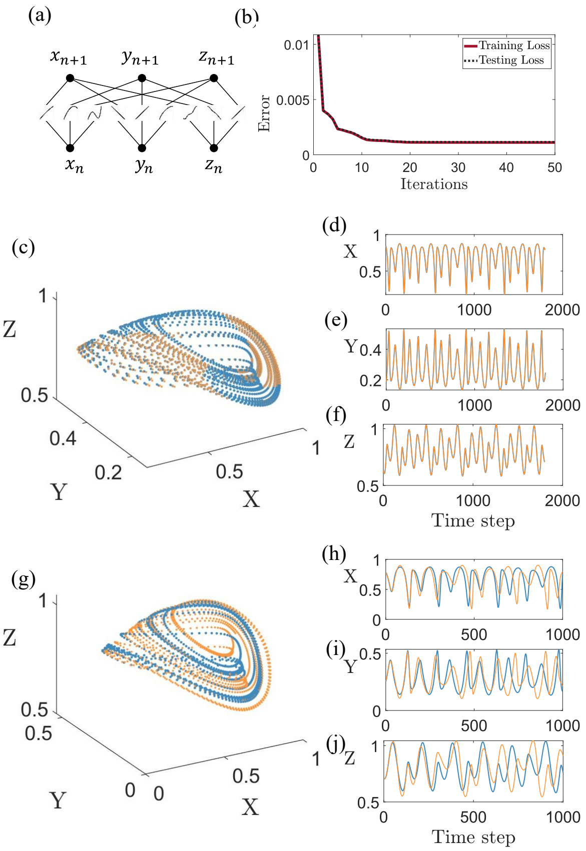

FIG. 4. KAN applied to a chaotic ecosystem. (a) KAN structure with 3 input and 3 output nodes. (b) Training and testing loss curves. (c) KAN generated attractor during the training phase (orange), which agrees completely with the ground truth (blue). (d-f) KAN generated time series (orange) in agreement with the true time series (blue). (g-j) Similar to (c-f) but for the testing phase. Due to chaos, the KAN generated time series diverges from the true ones from the same initial condition, but the KAN attractor agrees with the true one. The true Lyapunov exponents are $[0.0053, 0, -0.2288]$. The exponents of the KAN-generated attractor are consistent: $[0.0095, -5.8 \times 10^{-6}, -0.3932]$. The errors arise from the implicit numerical evaluation of the Jacobian matrix.

tween the system and model alone does not prevent the model from producing an unrealistic orbit $\mathrm{Orbit}_G(i\mathbf{x}_0)$ that behaves quite differently from any orbit of $\mathbf{F}$, e.g., a model orbit that diverges to infinity even if the true orbit produces bounded attractor. Furthermore, it is even more difficult to consider a model orbit that has statistical properties such as the invariant measure of a chaotic attractor analogous to the attractor of the true system.

The KAN was represented as an efficient way to replace a standard multi-layer perceptron (MLP) [53] and, in so doing, the weights of edges are in principle eliminated, but in practice they are absorbed into representing the various activation functions at the vertices of the network. That is, in stating the basic form of a KAN as $G(x) = \sum_{q=1}^{2n+1} \Phi_q \circ \sum_{p=1}^{n} \phi_{q,p}(x_p)$, in practice each activation function $\phi_{q,p}$ was represented as a cubic spline numerically [53], and therefore each has many internal fitted parameters of the scalar piecewise cubics. Collecting all these as the set of parameters $\Theta_{q,p}$ for each $\phi_{q,p}$, and Θ_q for each Φ_q, we can state the complete collection of parameters $\Theta = \cup_q(\Theta_q) \cup (\cup_{qp}\Theta_{q,p})$ and rewrite the function to emphasize the internal parameters: $G_\Theta(x) = \sum_{q=1}^{2n+1} \Phi_{q,\Theta_q} \circ \sum_{p=1}^{n} \phi_{q,p,\Theta+q,p}(x_p)$, and for a multivariate argument $x = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$. It is shown [53] that a regularized fit to the data by a loss function $\mathcal{L}(\mathcal{D}; \Theta)$ (over a data set $\mathcal{D}$ with respect to the fitting parameters Θ), with an objective of data fidelity as least squares fit across the data set balanced against L_2 norm on the parameters to prevent overfitting.

While excellent fit when optimizing $\mathcal{L}(\mathcal{D}; \Theta)$ was observed, it is possible to emphasize sparsification. That is, one or some of the activation functions may be set to zero, a procedure that was called "pruning" [53]. This procedure is possible when the representation of the activation functions by splines is sufficiently fine so that there are more parameters than data points. In such case, $\mathcal{L}(\mathcal{D}; \Theta)$ will generally have nontrivial level sets. The sparsification concept speaks to one of the many reasons to exploit these level sets, generally in terms of machine-learning interpretability, where the fitted KAN model is pushed toward just a few physics recognizable activation functions and the residual in a few terms is collected. The mathematical reason this kind of procedure is possible hinges on the implicit function theorem [59]. In brief, the KAN model function $G_\Theta(x)$ can be varied smoothly with respect to the fitting parameters so that $\mathcal{L}(\mathcal{D}; \Theta) = c$ is constant for a given parameter c. Therefore even following numerical optimization to a small value c, there will generally be smooth level sets with respect to the Θ parameters to emphasize other goals of explainability. A smooth implicit function $\Theta = h(s)$ exists under the conditions of a nonsingular Jacobian derivative $D_\Theta\mathcal{L}$ that continues a c-level set, and in principle this level of constancy $\mathcal{L}(\mathcal{D}; h(s)) = c$ set may intersect the other useful or desirable interpretabile states, including sparsification.

To summarize, we exploited KANs to solve the problem of data-driven model discovery for any dynamical systems including those for which the popular sparsity-optimization approach to finding the governing equations fails. Our result may be understood as realizing shadowing in the functional space where KANs find certain functions that produce the same dynamics. These functions may or may not have the same mathematical forms as the governing equations of the system and may even be implicit with a numerical representation. In the space of all functions, an infinite number of such "shadowing" functions may exist. We demonstrated that KAN-based machine learning can indeed find many of them, depending on the neural-network architecture.

This work was supported by AFOSR under Grant No. FA9550-21-1-0438. E.B. was supported by the ONR, ARO, DARPA RSDN, and NIH-NSF CRCNS.

[1] J. D. Farmer and J. J. Sidorowich, Predicting chaotic time series, Phys. Rev. Lett. **59**, 845 (1987).

[2] J. P. Crutchfield and B. McNamara, Equations of motion from a data series, Complex Sys. **1**, 417 (1987).

[3] M. Casdagli, Nonlinear prediction of chaotic time series, Physica D **35**, 335 (1989).

[4] G. Sugihara, B. Grenfell, R. M. May, P. Chesson, H. M. Platt, and M. Williamson, Distinguishing error from chaos in ecological time series, Phil. Trans. Roy. Soc. London B **330**, 235 (1990).

[5] J. Kurths and A. A. Ruzmaikin, On forecasting the sunspot numbers, Solar Phys. **126**, 407 (1990).

[6] P. Grassberger and T. Schreiber, Nonlinear time sequence analysis, Int. J. Bif. Chaos **1**, 521 (1990).

[7] G. Gouesbet, Reconstruction of standard and inverse vector fields equivalent to a Rössler system, Phys. Rev. A **44**, 6264 (1991).

[8] A. A. Tsonis and J. B. Elsner, Nonlinear prediction as a way of distinguishing chaos from random fractal sequences, Nature (London) **358**, 217 (1992).

[9] E. Baake, M. Baake, H. G. Bock, and K. M. Briggs, Fitting ordinary differential equations to chaotic data, Phys. Rev. A **45**, 5524 (1992).

[10] A. Longtin, Nonlinear forecasting of spike trains from sensory neurons, Int. J. Bif. Chaos **3**, 651 (1993).

[11] D. B. Murray, Forecasting a chaotic time series using an improved metric for embedding space, Physica D **68**, 318 (1993).

[12] T. Sauer, Reconstruction of dynamical systems from interspike intervals, Phys. Rev. Lett. **72**, 3811 (1994).

[13] G. Sugihara, Nonlinear forecasting for the classification of natural time series, Philos. T. Roy. Soc. A. **348**, 477 (1994).

[14] B. Finkenstädt and P. Kuhbier, Forecasting nonlinear economic time series: A simple test to accompany the nearest neighbor approach, Empiri. Econ. **20**, 243 (1995).

[15] U. Parlitz, Estimating model parameters from time series by autosynchronization, Phys. Rev. Lett. **76**, 1232 (1996).

[16] S. J. Schiff, P. So, T. Chang, R. E. Burke, and T. Sauer, Detecting dynamical interdependence and generalized synchrony through mutual prediction in a neural ensemble, Phys. Rev. E **54**, 6708 (1996).

[17] G. G. Szpiro, Forecasting chaotic time series with genetic algorithms, Phys. Rev. E **55**, 2557 (1997).

[18] R. Hegger, H. Kantz, and T. Schreiber, Practical implementation of nonlinear time series methods: The tisean package, Chaos **9**, 413 (1999).

[19] E. M. Bollt, Controlling chaos and the inverse frobenius-perron problem: global stabilization of arbitrary invariant measures, Int. J. Bif. Chaos **10**, 1033 (2000).

[20] R. Hegger, H. Kantz, L. Matassini, and T. Schreiber, Coping with nonstationarity by overembedding, Phys. Rev. Lett. **84**, 4092 (2000).

[21] S. Sello, Solar cycle forecasting: a nonlinear dynamics approach, Astron. Astrophys. **377**, 312 (2001).

[22] T. Matsumoto, Y. Nakajima, M. Saito, J. Sugi, and H. Hamagishi, Reconstructions and predictions of nonlinear dynamical systems: a hierarchical bayesian approach, IEEE Trans. Signal Proc. **49**, 2138 (2001).

[23] L. A. Smith, What might we learn from climate forecasts?, Proc. Nat. Acad. Sci. (USA) **19**, 2487 (2002).

[24] K. Judd, Nonlinear state estimation, indistinguishable states, and the extended kalman filter, Physica D **183**, 273 (2003).

[25] T. D. Sauer, Reconstruction of shared nonlinear dynamics in a network, Phys. Rev. Lett. **93**, 198701 (2004).

[26] 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).

[27] C. Tao, Y. Zhang, and J. J. Jiang, Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm, Phys. Rev. E **76**, 016209 (2007).

[28] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, Phys. Rev. Lett. **106**, 154101 (2011).

[29] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J.-P. Ye, Network reconstruction based on evolutionary-game data via compressive sensing, Phys. Rev. X **1**, 021021 (2011).

[30] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and M. A. F. Harrison, Time-series-based prediction of complex oscillator networks via compressive sensing, EPL (Europhys. Lett.) **94**, 48006 (2011).

[31] R.-Q. Su, X. Ni, W.-X. Wang, and Y.-C. Lai, Forecasting synchronizability of complex networks from data, Phys. Rev. E **85**, 056220 (2012).

[32] R.-Q. Su, W.-X. Wang, and Y.-C. Lai, Detecting hidden nodes in complex networks from time series, Phys. Rev. E **85**, 065201 (2012).

[33] R.-Q. Su, Y.-C. Lai, and X. Wang, Identifying chaotic fitzhugh-nagumo neurons using compressive sensing, Entropy **16**, 3889 (2014).

[34] R.-Q. Su, Y.-C. Lai, X. Wang, and Y.-H. Do, Uncovering hidden nodes in complex networks in the presence of noise, Sci. Rep. **4**, 3944 (2014).

[35] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, Reconstructing propagation networks with natural diversity and identifying hidden sources, Nat. Commun. **5**, 4323 (2014).

[36] R.-Q. Su, W.-W. Wang, X. Wang, and Y.-C. Lai, Data based reconstruction of complex geospatial networks, nodal positioning, and detection of hidden node, R. Soc. Open Sci. **3**, 150577 (2016).

[37] A. A. R. AlMomani, S. Jie, and E. M. Bollt, How entropic regression beats the outliers problem in nonlinear system identification, Chaos **30**, 013107 (2020).

[38] R. Yang, Y.-C. Lai, and C. Grebogi, Forecasting the future: is it possible for time-varying nonlinear dynamical systems?, Chaos **22**, 033119 (2012).

[39] E. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Info. Theory **52**, 489 (2006).

[40] E. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure Appl. Math. **59**, 1207 (2006).

[41] D. Donoho, Compressed sensing, IEEE Trans. Info. Theory **52**, 1289 (2006).

[42] R. G. Baraniuk, Compressed sensing, IEEE Signal Pro-

cess. Mag. **24**, 118 (2007).

[43] E. Candès and M. Wakin, An introduction to compressive sampling, IEEE Signal Process. Mag. **25**, 21 (2008).

[44] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Nat. Acad. Sci. (USA) **113**, 3932 (2016).

[45] Y.-C. Lai, Finding nonlinear system equations and complex network structures from data: A sparse optimization approach, Chaos **31**, 082101 (2021).

[46] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. **20**, 130 (1963).

[47] O. E. Rössler, Equation for continuous chaos, Phys. Lett. A **57**, 397 (1976).

[48] K. Ikeda, Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system, Opt. Commun. **30**, 257 (1979).

[49] S. M. Hammel, C. K. R. T. Jones, and J. V. Moloney, Global dynamical behavior of the optical field in a ring cavity, J. Opt. Soc. Am. B **2**, 552 (1985).

[50] C. S. Holling, The components of predation as revealed by a study of small-mammal predation of the european pine sawfly, Canad. Entomol. **91**, 293–320 (1959).

[51] C. S. Holling, Some characteristics of simple types of predation and parasitism, Canad. Entomol. **91**, 385 (1959).

[52] 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, Proc. Natl. Acad. Sci. (USA) **115**, E639 (2018).

[53] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, KAN: Kolmogorov-Arnold networks, arXiv:2404.19756 (2024).

[54] A. N. Kolmogorov, On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, Rus. Acad. Sci. **114**, 953 (1957).

[55] A. B. Givental, B. A. Khesin, J. E. Marsden, A. N. Varchenko, V. A. Vassiliev, O. Y. Viro, and V. M. Zakalyukin, eds., On the representation of functions of several variables as a superposition of functions of a smaller number of variables, in *Collected Works: Representations of Functions, Celestial Mechanics and KAM Theory, 1957–1965* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009) pp. 25–46.

[56] J. Braun and M. Griebel, On a constructive proof of kolmogorov's superposition theorem, Const. Appro. **30**, 653 (2009).

[57] Supplementary Information contains a detailed description of the KAN framework for model discovery of nonlinear dynamical systems, statistical analysis of the KAN generated attractors, and additional examples.

[58] K. McCann and P. Yodzis, Nonlinear dynamics and population disappearances, Am. Nat **144**, 873 (1994).

[59] K. G. Binmore, *Mathematical Analysis: a straightforward approach* (Cambridge University Press, 1982).

Supplementary Information for "Data-driven model discovery with Kolmogorov-Arnold networks"

Mohammadamin Moradi,[1, *] Shirin Panahi,[1, *] Erik M. Bollt,[2] and Ying-Cheng Lai[1, 3, †]

[1]*School of Electrical, Computer, and Energy Engineering,*
Arizona State University, Tempe, AZ 85287, USA
[2]*Department of Electrical and Computer Engineering,*
Clarkson Center for Complex Systems Science,
Clarkson University, Potsdam, New York 13699, USA
[3]*Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Dated: September 9, 2024)

Abstract

The Supplementary Information provides additional details to support the results in the main text. It is helpful but not essential for understanding the main results of the paper.

CONTENTS

[*] The first two authors contributed equally to this work
[†] Ying-Cheng.Lai@asu.edu

I. KOLMOGOROV-ARNOLD NETWORKS

Kolmogorov-Arnold networks (KANs) are based on the Kolmogorov-Arnold representation theorem, which states that any continuous multivariate function can be decomposed into a finite set of univariate functions and their combinations [1]. Conventional neural networks have limitations such as their “black-box” nature and high computational costs, but KANs can potentially alleviate these limitations as a promising alternative in machine learning. Furthermore, a conventional deep neural network may have thousands or millions of nodes, and hence weights and biases but all the same activation functions standards such as *atnh*, *ReLu*, or sigmoid. A KAN, however, may be dramatically smaller with dozens of nodes, but all empirically fitted with different threshold functions that give meaning and interpretability to the results. This follows their power and ability in machine learning for dynamical systems.

A. Activation functions

In the numeric phase of KAN training, each one-dimensional (1D) function is parameterized as a B-spline curve, where B-splines, or basis splines, are piecewise-defined polynomials that offer a flexible and efficient way to represent functions. The B-spline is defined by its degree, a set of control points, and a knot vector that determines where and how the polynomial pieces connect. In a KAN, the 1D function $f_i(x)$ can be expressed as a linear combination of B-spline basis functions $B_j(x)$:

$$f_i(x) = \sum_j c_j B_j(x), \tag{S1.1}$$

where c_js are the learnable coefficients that are optimized during the training process to best fit the data. However, the activation functions in KANs are not limited just to B-splines; they can incorporate a combination of a basis function $b(x)$ (often a residual function) and the B-spline function. Each final activation function can then be written as

$$\phi_i(x) = b(x) + f_i(x), \tag{S1.2}$$

where $b(x) = x/(1 + e^{-x})$.

B. Hyperparameters

Hyperparameter tuning in KANs plays a crucial role not only in optimizing machine-learning performance but also in enhancing its interpretability by promoting a sparser structure. A primary hyperparameter is the overall penalty strength λ that controls the overall regularization magnitude. The penalty strength of entropy, denoted as λ_{ent}, is specifically designed to control sparsity and reduce the number of active activation functions. A larger λ_{ent} value encourages the machine-learning model to utilize fewer functions, potentially leading to a simpler and more interpretable model. Another important set of hyperparameters is those associated with the B-spline activation functions, such as the order K and the number

G of control points of such a spline. More specifically, in a B-spline, each control point corresponds to a basis function, a polynomial of order K. These control points play a role in the interpretability of the model: a smaller number G of control points can make the model more challenging to interpret as it restricts the complexity of the basis functions. In addition, the structure of KANs, which includes the number of hidden nodes and hidden layers, provides another set of key hyperparameters impacting the model capacity and accuracy. The learning rate, the number of iterations, and the batch size are also crucial, as they can affect the convergence speed and stability of the training process.

C. Number of KAN parameters

In a KAN, the total number of trainable parameters is determined by the number of activation functions in close relation to the architecture of the network defined by the numbers of the input nodes (N_i), of the hidden nodes in each hidden layer (N_{h1}, N_{h2}, $\ldots$, N_{hj}), and of the output nodes (N_o). The structural complexity of the KAN is then determined by the number of activation functions (N_a), expressed as

$$N_a = (N_i \times N_{h1}) + (N_{h1} \times N_{h2}) + \cdots + (N_{hj} \times N_o). \tag{S1.3}$$

Consider the numeric training phase of KAN. Each activation function within the KAN is parameterized by a B-spline curve represented as a linear combination of the basis functions, as outlined in Eq. (S1.1). Each B-spline curve is characterized by $(G + K)$ trainable coefficients. The total number of trainable parameters in a KAN is then given by:

$$(G + K) \times N_a, \tag{S1.4}$$

which gives a direct relationship between the network's architecture and its trainable parameters. Increasing the number of hidden layers or nodes can significantly impact the total number of parameters, influencing the network's capacity and complexity of the learned representations. Table S1 presents the number of trainable parameters for the KAN structures used to generate the Ikeda and food-chain dynamics.

Parameters System	Structure	G	K	Na
Ikeda 1	[2,4,2]	10	3	208
Ikeda 2	[2,10,2]	10	3	520
Food-Chain	[3,3]	3	3	54

TABLE S1. Number of trainable parameters in KAN

II. DYNAMICAL INVARIANTS AND STATISTICAL ANALYSIS

Statistical analysis offers quantitative metrics to assess how accurately a KAN model replicates the underlying behavior of a system, facilitating performance comparison. Quantities such as the Lyapunov exponents, power spectra, correlation dimensions, and statistical distances allow us to assess the accuracy and fidelity of the KAN-produced models in capturing the complex dynamics of the ground truth. Here we provide a brief description of each of the above quantities.

A. Lyapunov exponents

Lyapunov exponents are critical indicators of the dynamical behavior of a system, particularly in identifying chaos. They measure the average rate at which trajectories in the system diverge or converge along different local directions in the phase space. To compare the dynamics of KAN-produced models with the ground truth, we calculate the Lyapunov exponents for both. A positive Lyapunov exponent signifies a sensitive dependence on initial conditions - the hallmark of chaos [2]. For the KAN model, we compute the Lyapunov exponents using the standard numerical approach that involves the Jacobian matrix and QR decomposition [3]. First, we initialize a set of orthonormal vectors that evolve according to the dynamics of the system. At each time step, we numerically calculate the Jacobian matrix $J(x)$ of the KAN model, which represents the local linearization of the system. A commonly used method is Newton-based, e.g., the standard finite difference method, the central finite-difference method, and the complex-step derivative approximation method [4, 5].

The standard finite difference (SFD) method approximates the derivative by evaluating the function at slightly perturbed values around a point. The forward finite difference formula is typically used, which is given by:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

where h is a small step size. This method is straightforward to implement but suffers from truncation errors and can be sensitive to the choice of h, potentially leading to inaccuracies, especially when h is too large or too small. The central finite difference (CFD) method improves on SFD by taking the average of the forward and backward differences:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

This method generally yields better accuracy than SFD because it reduces the truncation error, making it second-order accurate. However, it requires two function evaluations for each derivative calculation, which can be computationally expensive. Additionally, there are situations where only forward or backward evaluation is feasible, limiting the use of the central method. The complex-step derivative approximation (CDA) method offers even greater accuracy by leveraging complex arithmetic:

$$f'(x) \approx \frac{\mathrm{Im}(f(x+ih))}{h}$$

where i is the imaginary unit. This method is known for its precision, as it avoids the subtraction errors that can plague a finite difference method and offers an effective means to achieve highly accurate derivatives with smaller step sizes [6]. However, it is only applicable when the function $f(x)$ supports complex numbers and does not involve non-analytic functions such as those containing absolute values or conditional statements. While the

CDA method is highly accurate and efficient, its use is restricted by the requirement that the function support complex numbers. Since KANs cannot be evaluated using complex numbers, we opt for the CFD method, which offers a reliable balance between accuracy and practical applicability.

The numerically calculated Jacobian matrix allows us to update the orthonormal vectors, whose dynamical evolution is tracked over time and their growth rates are calculated. To maintain the orthonormality and prevent numerical errors, we perform QR decomposition on the product of the Jacobian matrices at each step. The logarithms of the diagonal elements of the resulting upper triangular matrix R give the local Lyapunov exponents. By averaging these values over a long integration time, we obtain values of the exponents. The calculated Lyapunov exponents for the KAN-produced Ikeda and food-chain systems are listed in Tabs. S2 and Tabs. S3.

TABLE S2. Lyapunov exponents of the Ikeda dynamics

LEs Sys	L_1	L_2
Ground Truth	0.502494	-0.726278
Ikeda 1	0.507518	-0.718239
Ikeda 2	0.503313	-0.731115

TABLE S3. Lyapunov exponents of the food-chain dynamics

LEs Sys	L_1	L_2	L_3
Ground Truth	0.009495	-0.00058	-0.393213
Food chain	0.005354	-0.000005	-0.228759

Despite different models, the resulting Lyapunov exponents are essentially the same, revealing the same attractor.

B. Power spectrum

Power spectra provide another way to compare the KAN-produced model with the ground truth, as shown in Fig. S1. It can be seen that the KAN model captures the underlying periodicities and complex oscillations present in the ground truth dynamics for both the Ikeda and food-chain systems.

C. Correlation dimension

The correlation dimension is a measure of the fractal structure of a system's attractor in phase space. It quantifies the complexity of the dynamical system by describing how the number of points within a given distance scales with the distance. To compare the KAN-produced model with the ground truth, we calculate the correlation dimension for both sets of time series data. This involves reconstructing a phase space from the time series and then using methods such as the Grassberger-Procaccia algorithm to estimate the correlation dimension [7]. The results listed in Tab. S4 show a close match in the correlation dimensions, indicating that the KAN model is fully capable of replicating the true attractor of the target system.

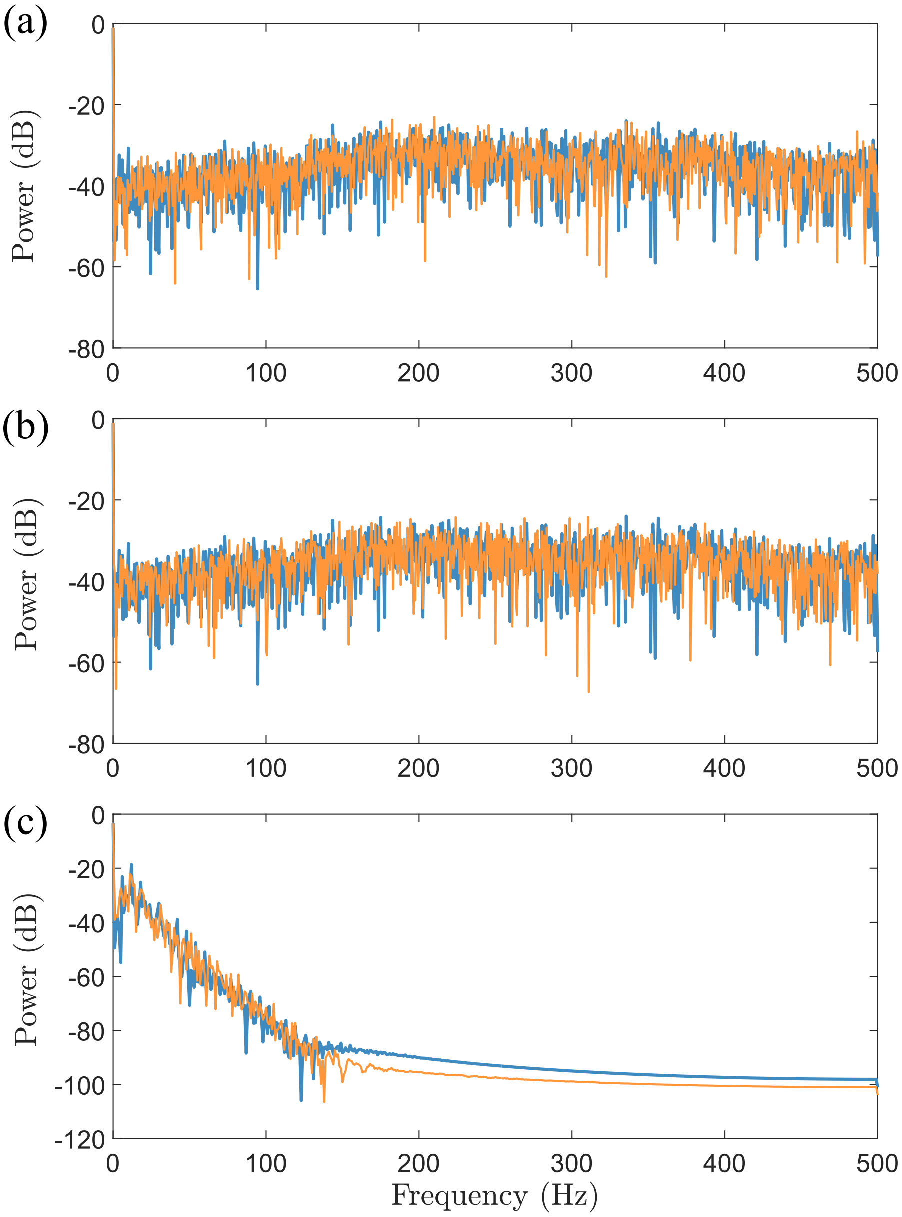

FIG. S1. Power spectrum Comparison. Shown are the power spectra of the KAN-produced dynamics (orange) compared with the ground truth (blue): (a,b) the two KAN models generating the same Ikeda dynamics as described in the text, and (c) the food-chain dynamics.

Model System	Ground Truth	KAN
Ikeda 1	1.6296	1.7062
Ikeda 2	1.62968	1.6134
Food chain	2.5732 8	2.3649

TABLE S4. Comparison of correlation dimension between and KAN model and ground truth

D. Statistical distance: three measures of divergence

Statistical distances are another measure of comparing the similarities of the two attractors in the phase space. Commonly used measures include the Kullback-Leibler divergence and Hellinger or total-variation divergence [8, 9]. The measures characterize the differences between the distributions on the attractors from different angles. More specifically, The Kullback-Leibler divergence measures how one probability distribution Q diverges from a second, reference probability distribution P:

$$\mathrm{KL}(P||Q) = \sum_i P(i) \log\big(\frac{P(i)}{Q(i)}\big),$$

which is asymmetric and indicative of how much information is lost when Q is used to approximate P. The Hellinger distance, derived from the Bhattacharyya coefficient, measures the similarity between two distributions:

$$\mathrm{H}(P,Q) = \sqrt{\sum_i \big(\sqrt{P(i)} - \sqrt{P(i)}\big)^2},$$

which is symmetric and gives the maximum possible difference between P and Q. The total-variation divergence measures the maximum difference between the probabilities assigned to the same event by two distributions:

$$\delta(P,Q) = \frac{1}{2}\sum_i |P(i) - Q(i)|,$$

which is symmetric and gives the overall difference between the two distributions in terms of their probability masses.

For all three divergence measures, a smaller value indicates greater similarity between the distributions, while a larger value suggests a more significant difference. To compare the KAN model with the ground truth, we first estimate the probability density functions of both attractors using the kernel density estimation method. We then compute the Kullback-Leibler, Hellinger, and total-variation divergences. The results for both the Ikeda and food-chain systems are presented in Tab. S5, where distribution p corresponds to the ground truth model and distribution q is from the KAN-produced model. To ensure the reliability and fairness of the comparison, we also compare the probability density function of the ground truth Ikeda attractor p with that of a random attractor q with the following results: $KL(P||Q) = 1.0532$, $H(P,Q) = 0.6024$, and $\delta(P,Q) = 0.5999$. These results demonstrate that the KAN-produced attractors are essentially statically identical to the ground truth (within numerical errors), highlighting the effectiveness of the KAN model in capturing the true system's behavior.

III. SPECTACULAR FAILURES OF SPARSE OPTIMIZATION APPROACH TO FINDING EQUATIONS FOR THE IKEDA MAP AND CHAOTIC FOOD-CHAIN SYSTEM

The sparse-optimization approach to finding the governing equations of nonlinear dynamical systems from data was first introduced in 2011 [10]. The idea is that power-series or

Method System	Kullback-Leibler	Hellinger	Total-variation
Ikeda 1	0.0124	0.0556	0.0595
Ikeda 2	0.0180	0.0652	0.0669
Food chain	0.0002	0.0034	0.0063

TABLE S5. Attractor distribution comparison between ground truth and KAN-produced models. The small divergence values indicate that the model and the true systems produce essentially the same attractor.

Fourier-series expansions can be used to approximate smooth but nonlinear dynamical functions, converting the problem to that of estimating the coefficients of the series-expansion terms. If the series contain many high-order terms, the number of coefficients to be estimated is large, making the problem unsolvable. However, the equations of many classical dynamical systems are relatively simple in terms of series expansion in the sense that a vast majority of the coefficients are zero, resulting in a sparse coefficient vector. The sparsity allows the use of sparse optimization methods such as compressive sensing to solve the coefficients. An advantage of the sparse-optimization methods lies in the need to of only limited observational data.

In the main text, it is emphasized that the Ikeda system violates the sparsity condition as the map functions contain an infinite number of power-series or Fourier-series terms. When applying sparse optimization to such a system, every term, no matter how many are initially assumed, exists. As a result, any such algorithm would fail spectacularly. Here we present an example of such a spectacular failure when attempting to estimate the dynamical equation of the Ikeda map using a commonly used sparse-optimization algorithm [11] that employs library of base functions including polynomials, inverse functions, products, exponential, and sinusoidal functions, etc. The estimated map functions are

$$\begin{aligned} x^+ =& 14.413e^x + 21.543e^y - 10.137x - 18.639y \\ & + 5.308\sin(x) + 15.552\sin(y) - 41.853\cos(x) + 6.222\cos(y) \\ & - 0.218\sin(x+y) - 10.137x - 18.638y - 28.135x^2 - 8.364y^2 + 0.152xy, \\ y^+ =& - 3.604e^x + 0.170e^y + 5.990x - 0.282y \\ & - 9.407\sin(x) + 3.029\cos(x) + 0.903\cos(y) + 0.246\sin(x+y) \\ & + 5.990x - 0.283y + 2.439x^2 + 0.493y^2 + 0.552xy. \end{aligned} \tag{S3.5}$$

The food-chain system described in the main text is another example where sparse optimization fails spectacularly. The estimated governing equations are

$$\begin{aligned} x^+ =& 3.445e^x + 6.290\sin(x) + 0.869\sin(y) \\ & - 3.424\cos(x) + -0.391\sin(x+y) - 8.528x \\ & - 0.710y + -3.967x^2 + -1.363\frac{xy}{1+x}, \\ y^+ =& 0.572y^2 + 2.050xy - 3.879\frac{xy}{1+x} \\ z^+ =& 0. \end{aligned} \tag{S3.6}$$

The time series produced by these equations (orange) versus the ground truth (blue) are shown in Figs. S2(a) and S2(b) for the Ikeda map and the food-chain system, respectively. It can be seen that the discovered equations fail to produce the true time series from the respective system.

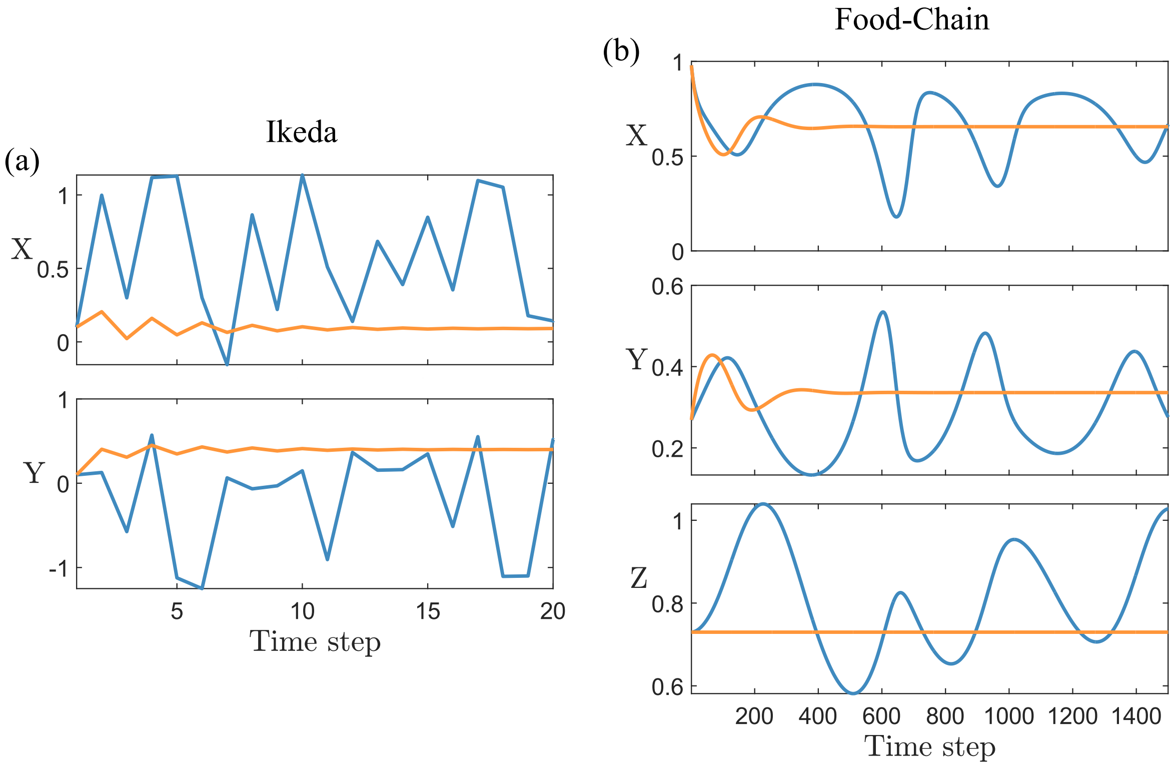

FIG. S2. Comparison of time series from the equations found by sparse optimization and the ground truth. Shown are two sets of time series (orange: equations from sparse optimization; blue: ground truth) for (a) Ikeda map and (b) chaotic food-chain system described in the main text. The sparse-optimization method fails to find the correct equations.

IV. KAN MODEL DISCOVERY OF THREE CLASSICAL NONLINEAR DYNAMICAL SYSTEMS

A. Logistic map

The map is given by [12]

$$X(n+1) = rX(n)\big(1 - X(n)\big) \tag{S4.7}$$

where $X(n)$ represents the population at generation n, and r is a parameter that controls the growth rate. For values of r between 0 and 4, the map displays a range of behaviors from stable fixed points to periodic and chaotic attractors. For $r = 4$, the map generates a chaotic attractor in the unit interval $X \in [0, 1]$.

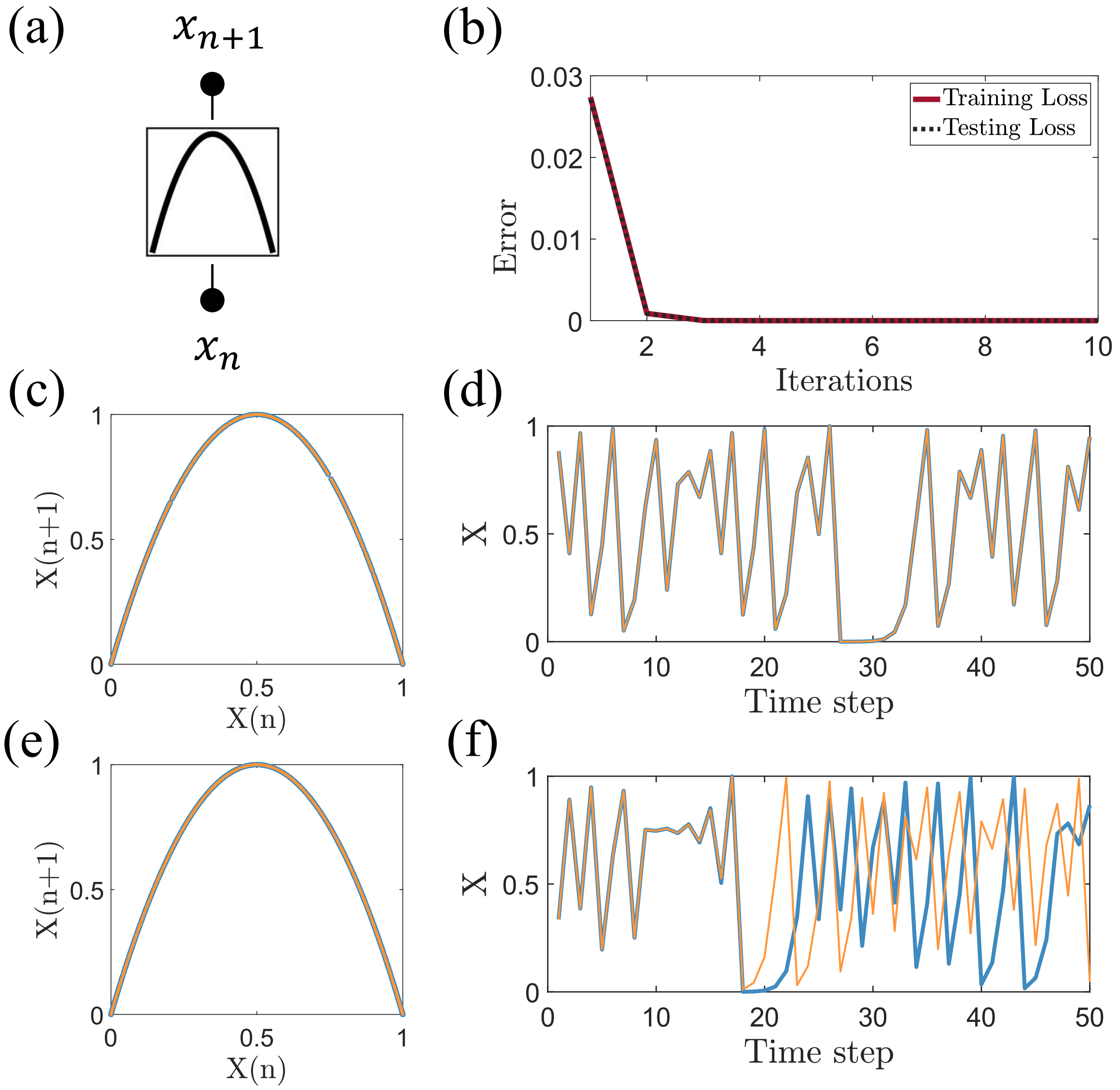

FIG. S3. KAN Model applied to the logistic map. (a) KAN model structure with one input and one output node, without hidden nodes. (b) Training and testing loss curves over 10 iterations. (c) Attractor and (d) time series during the training phase (orange) in comparison with the ground truth (blue), indicating high accuracy in first-step predictions. (e) Attractor and (f) time series during the testing phase. The KAN model faithfully replicates the logistic map's dynamics, in spite of the inevitable divergence due to the fundamental sensitive dependence on initial conditions of chaotic systems.

We utilize a simple KAN structure, as depicted in Fig. S3(a), which consists of a single input and a single output node without any hidden nodes. We use 10^4 time-series data points, where 80% are for training and the remaining 20% for testing. The training process spans 10 iterations, with the following hyperparameter values: $K = 3$ (cubic B-splines), grid size of 5 for the splines, loss-function parameters $\lambda = 0$ and $\lambda_{\text{entropy}} = 10$, learning rate 0.1, and a random seed initialized to zero. Similar to the examples in the main text, training is administered in a feedforward process, where the KAN is trained to minimize the difference between the input and output, predicting the future evolution of the target system based on the past dynamical variables. The red curve in Fig. S3(b) shows the training loss over time, while Figs. S3(c) and S3(d) display the KAN-produced attractor and time series during the training phase in comparison with the ground truth. The rapid decrease in the training loss to zero signifies high training accuracy and efficiency.

During the testing phase, we maintain the same set of training parameter values but replace the original input data point with the output of the KAN at each iteration. The black dashed curve in Fig. S3(b) represents the testing loss, and Figs. S3(e) and S3(f) show the KAN-predicted attractor and time series, respectively. While the KAN-predicted time series diverge from the ground truth after several iterations due to the fundamental sensitivity to initial conditions, the predicted attractor closely aligns with the ground truth, demonstrating that the KAN has successfully learned the chaotic dynamics of the logistic map.

B. Circle map

The map is given by

$$X(n+1) = X(n) + \Omega - \frac{K}{2\pi}\sin(2\pi X(n)) \mod 1, \tag{S4.8}$$

where $X(n)$ represents the phase at iteration n, Ω is a frequency parameter, and K is a nonlinearity parameter. The map's behavior varies from periodic and quasiperiodic motions to chaos, depending on the values of K and Ω. For $K > 1$, chaos can arise. We fix $K = 1$ and $\omega = 0.3$, for which the map exhibits quasiperiodic behavior with the trajectories that do not repeat exactly but densely cover a region of the phase space without ever closing.

Figure S4(a) illustrates the KAN structure, which consists of a single input and a single output node, with two hidden layers (three nodes in the first and two nodes in the second layer). The dataset consists of 10^4 points, with 90% allocated for training and 10% for testing. Training involves 200 iterations. The hyperparameter values are: $K = 3$ (cubic B-splines), grid size of 5 for the splines, loss-function parameters $\lambda = 0$ and $\lambda_{ent} = 10$, learning rate 0.1, and a random seed set to zero. The training (testing) is carried out in a feedforward (recurrent) process. The red (black dashed) curve in Fig.S4(b) illustrates the training (testing) loss over time. Figures S4(c,e) and S4(d,f) show the KAN-generated attractor and time series during the training and testing phases, respectively, in comparison with the ground truth. The rapid convergence of the training loss to zero highlights a high accuracy and efficiency in training. The time series during the testing phase diverges from the ground truth after more than 700 iterations, indicating an effectively zero Lyapunov exponent and good agreement of predicted attractor with the ground truth. This example then demonstrates that KAN is a faithful representation of a dynamical system generating quasiperiodic dynamics.

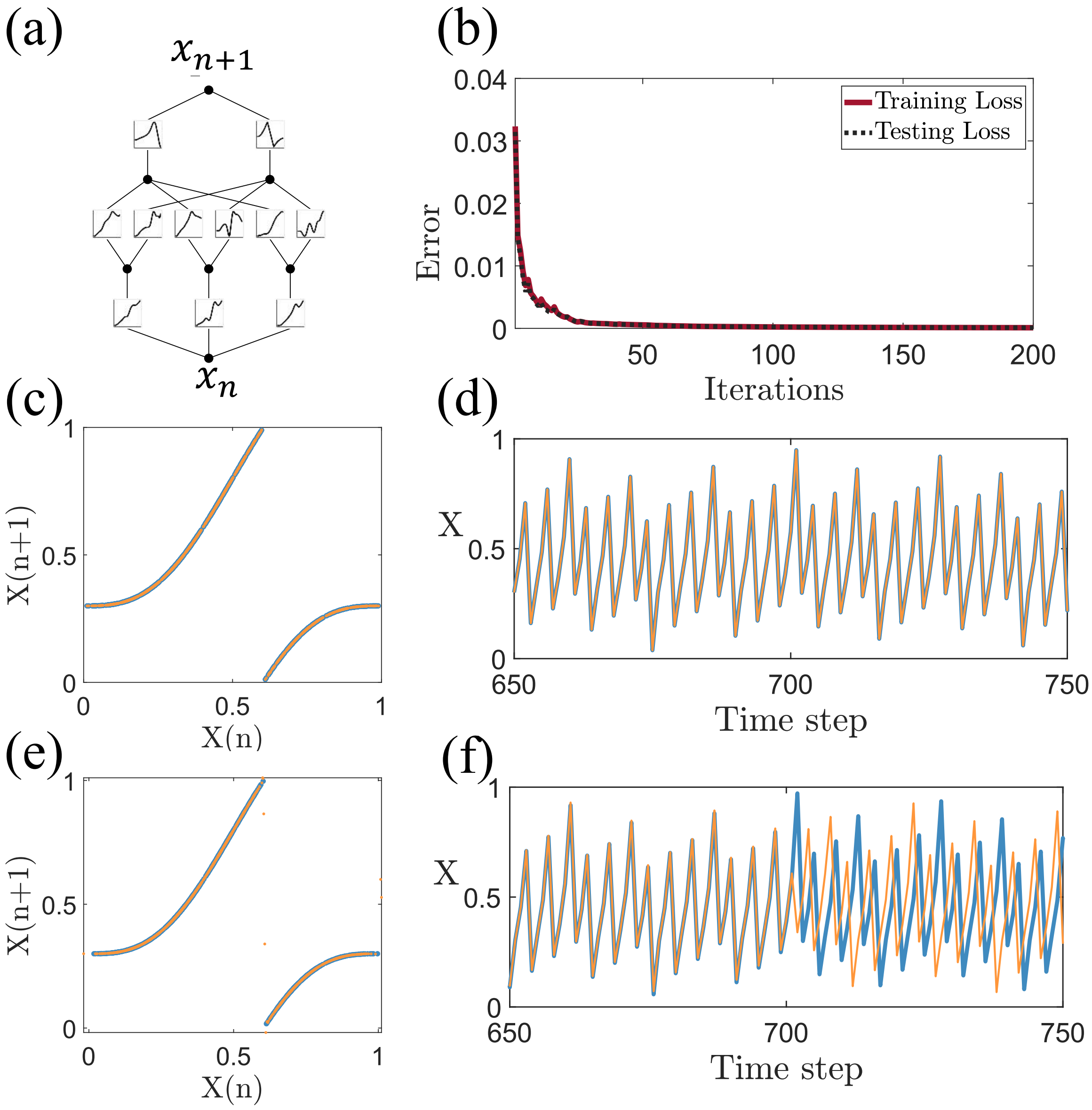

FIG. S4. KAN applied to the circle Map. (a) Structure of the KAN, which includes a single input and output node with two hidden layers. (b) Training and testing loss curves over 200 iterations. (c) Attractor and (d) time series during the training phase (orange) in comparison with the ground truth (blue), demonstrating high accuracy of first-step prediction. (e) Attractor and (f) time series during the testing phase, demonstrating that the KAN model effectively replicates the map dynamics, following the true time series for more than 700 steps before diverging. Such a long prediction time is indicative of the null Lyapunov exponent characteristic of quasiperiodic motion.

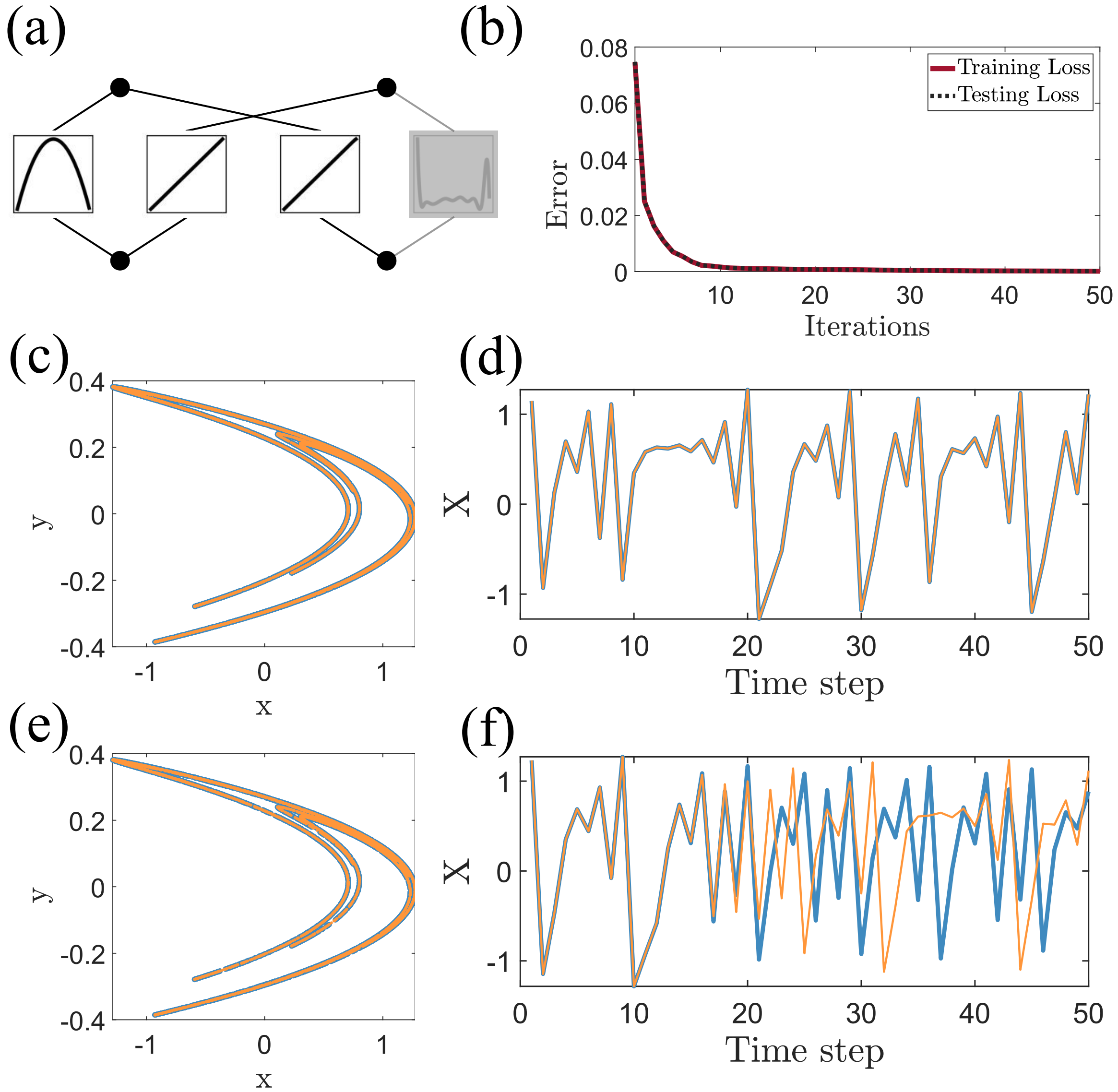

FIG. S5. KAN applied to the Hénon map. (a) KAN structure: two input and two output nodes, no hidden layers. (b) Training and testing loss curves over 50 iterations. (c) Attractor and (d) time series during the training phase (orange) in comparison with the ground truth. (e) Attractor and (f) time series during the testing phase, demonstrating that the KAN model effectively replicates the chaotic dynamics of the Hénon map.

C. Hénon map

The two-dimensional map is given by [13]

$$\begin{aligned} x(n+1) =& 1 - ax(n)^2 + y(n), \\ y(n+1) =& bx(n), \end{aligned} \tag{S4.9}$$

where $x(n)$ and $y(n)$ are the dynamical variables at the n^{th} iteration, a and b are parameters. The standard Hénon attractor is for $a = 1.4$ and $b = 0.3$. The KAN has a $[2, 2]$ structure, as illustrated in Fig. S5(a), with two input and two output nodes. The dataset comprises 5×10^4 points with 90% for training and the remaining 10% for testing. The training process spans 50 iterations, with the following hyperparameter values: grid size 10, spline order $K = 3$, λ set to 0, λ_{entropy} set to 10, learning rate of 0.1, and a random seed initialized to zero. The results in Figs. S5(b-f) demonstrate that KAN represents a data-discovered model that faithfully generate the ground-truth Hénon chaotic dynamics.

[1] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, KAN: Kolmogorov-Arnold networks, arXiv:2404.19756 (2024).

[2] E. Ott, *Chaos in Dynamical Systems*, 2nd ed. (Cambridge University Press, Cambridge, UK, 2002).

[3] A. Wolf *et al.*, Quantifying chaos with lyapunov exponents, Chaos **16**, 285 (1986).

[4] S. C. Chapra and R. P. Canale, Numerical methods for engineers (McGraw-Hill Education, New York, 2015) Chap. 23, pp. 819–858, 7th ed., chapter on Numerical Differentiation and Jacobians.

[5] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[6] W. Squire and G. Trapp, Using complex variables to estimate derivatives of real functions, SIAM review **40**, 110 (1998).

[7] P. Grassberger and I. Procaccia, Measuring the strangeness of strange attractors, Physica D **9**, 189 (1983).

[8] T. M. Cover, *Elements of information theory* (John Wiley & Sons, 1999).

[9] L. Pardo, *Statistical Inference Based on Divergence Measures*, Statistics: A Series of Textbooks and Monographs (CRC Press, 2018).

[10] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, Phys. Rev. Lett. **106**, 154101 (2011).

[11] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Nat. Acad. Sci. (USA) **113**, 3932 (2016).

[12] R. M. May, *Stability and Complexity in Model Ecosystems*, 2nd ed. (Princeton University Press, Princeton, NJ, USA, 1974).

[13] M. Hénon, A two-dimensional mapping with a strange attractor, Commun. Math. Phys. **50**, 69 (1976).