Data-driven model discovery with Kolmogorov-Arnold networks

Shirin Panahi,[1] Mohammadamin Moradi,[1] Erik M. Bollt,[2] and Ying-Cheng Lai[1,3,*]
[1]School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, Arizona 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

(Received 9 September 2024; revised 4 February 2025; accepted 3 April 2025; published 10 April 2025)

Data-driven model discovery of complex dynamical systems can be 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 functions. Examples where sparse optimization fails abound, including the classic Ikeda or optical-cavity map in nonlinear dynamics, as well as a wide variety of ecosystems. Another approach is based on machine learning, e.g., deep neural networks, which excels at capturing system behavior from data but functions as black boxes, offering little insight into how inputs influence the outputs. We propose a general model-discovery framework based on the Kolmogorov-Arnold networks (KANs) that are not constrained by the sparsity condition. The KAN framework with a simple structure is capable of accurately capturing the complex behavior of dynamical systems that do not meet the sparsity requirement while offering greater interpretability compared to conventional neural networks. This interpretability provides insights into the dynamics generating the data, which are typically inaccessible in traditional black-box function approximation methods. We demonstrate nonuniqueness in that a large number of approximate models of the system can be found that generate the same invariant set with the true statistics such as the Lyapunov exponents and Kullback–Leibler divergence. An analogy to shadowing of numerical trajectories in chaotic systems is pointed out.

DOI: 10.1103/PhysRevResearch.7.023037

I. INTRODUCTION

Uncovering the underlying model of dynamical systems from observational or measurement data is a central pursuit in science. While it is desirable to obtain explicit mathematical equations governing the system, developing a fully data-driven approach is challenging, particularly when the physical mechanisms underlying the observed dynamics are unknown or incomplete. Finding the governing equations from data can be useful for tasks such as better understanding the system, predicting the future state, and designing control strategies [1]. In nonlinear dynamics and complex networks, modeling systems from data has been an active area of research [2–16]. In recent years, two main approaches have been extensively studied: machine learning and sparse optimization.

Machine learning techniques have been applied to modeling dynamical systems [17–20]. Techniques such as deep neural networks are adept at predicting the system behavior from data [21]. For example, convolutional neural networks, recurrent neural networks [22–24], transformers [25], and reservoir computing [26–28] have been used for tasks such as predicting chaotic systems [29–32], parameter tracking [33], control of complex trajectories [34], and digital twins of nonlinear dynamical systems [20,35], anticipating critical transitions and tipping points [18,36]. While these methods excel in predictive accuracy, they often function as black boxes, lacking interpretability and making it difficult to extract explicit governing equations.

In nonlinear dynamics, the problem of constructing explainable mathematical models from data was investigated quite early, where a method based on calculating the information contained in sequential observations to deduce the deterministic equations was proposed [1]. Other approaches include approximating a nonlinear system by a large collection of linear equations [37–39], fitting differential equations to chaotic data [40], exploiting chaotic synchronization [41] or genetic algorithms [9,42], an inverse Frobenius-Perron approach to designing a dynamical system "near" the original system [4], and finding the least-squares best approximation for modeling [8]. In recent years, 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 estimating 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

*Contact author: Ying-Cheng.Lai@asu.edu

of finding these nontrivial coefficients can then be formulated [10,43] as a compressive-sensing problem [44–48]. Under the same idea, a method known as SINDy (sparse identification of nonlinear dynamics) was later developed, which has gained popularity [49,50].

The sparse-optimization approach is effective for systems whose governing equations are sufficiently simple in the sense of sparsity, such as the chaotic Lorenz [51] and Rössler [52] 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 [53,54], 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 [55,56] also violate the sparsity condition [57]. For these systems, the sparse-optimization approach to model discovery completely fails.

In this paper, we articulate a general approach to uncovering the models of dynamical systems, including those that do not satisfy the sparsity condition - a common limitation of the existing sparse optimization methods. We consider a dynamical system described by

$$d\mathbf{x}/dt = \mathbf{F}(\mathbf{x}) \text{ or}$$
$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n), \tag{1}$$

including where $\mathbf{F}(\mathbf{x})$ *does not* possess a sparse structure. The goal is to find an approximation of $\mathbf{F}(\mathbf{x})$, denoted as $\mathbf{G}(\mathbf{x})$, such that the system

$$d\mathbf{x}/dt = \mathbf{G}(\mathbf{x}), \text{ or}$$
$$\mathbf{x}_{n+1} = \mathbf{G}(\mathbf{x}_n) \tag{2}$$

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). Our method leverages Kolmogorov-Arnold networks (KANs), a recent computational framework for representing complex mathematical functions [58,59], rooted in the classical Kolmogorov-Arnold theorem [60–62] that any multivariate mathematical function can be decomposed as a sum of single-variate functions. KANs lie between the traditional sparse identification methods such as SINDy that fail for systems lacking a sparse structure and neural network-based approaches such as reservoir computing that can model complex dynamics but provide little insight into the underlying system. By decomposing high-dimensional problems into simpler, interpretable univariate functions, KANs address the shortcomings of both types of methods.

To demonstrate the effectiveness of KANs, we apply our framework to benchmark systems (Ikeda map and a chaotic ecosystem) where the SINDy-based approach fails to recover the governing equations. We show that some function $\mathbf{G}(\mathbf{x})$ in Eq. (2) in an implicit form can indeed be found by the KANs. Additionally, we illustrate the robustness of KANs against noise and its ability to enhance interpretability by analyzing synthetic data from the Atlantic Meridional Overturning Circulation (AMOC) model. Our results show that, by leveraging limited domain knowledge, KANs provide a bridge between purely data-driven techniques and the traditional model-based approaches, expanding the frontier of research on dynamical-system discovery.

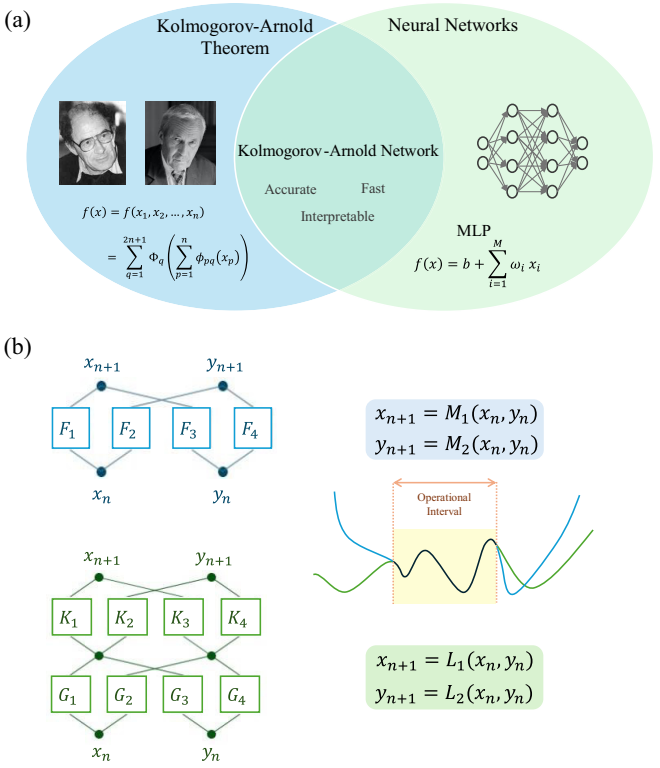

FIG. 1. Basics of KAN. (a) Kolmogorov-Arnold theorem and neural network. (b) Schematic illustration of two different structures (blue and green) leading to two different functions $\mathbf{M}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x})$ that generate the same dynamics as $\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n)$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

II. METHOD

A. 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 [58,59] as illustrated in Fig. 1(a). 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, in some applications a conventional deep neural network may have a large number nodes with an equally large number of weights and biases, even though the nodes all have the same activation functions. A KAN, however, may be dramatically smaller with dozens of nodes, all empirically fitted with different threshold functions that give meaning and interpretability to the results.

1. Activation functions

In the KAN's numeric training phase, each one-dimensional (1D) function is parametrized 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 $B_j(x)$:

$$f_i(x) = \sum_j c_j B_j(x), \tag{3}$$

where c_js are the learnable coefficients and one of the hyperparameters (see Appendix A for a detailed description of these hyperparameters) 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

$$\phi_i(x) = b(x) + f_i(x), \tag{4}$$

where $b(x) = x/(1 + e^{-x})$. They, along with the structure of the KAN, play a crucial role in determining the number of model parameters (see Appendix B for more details).

Besides B-splines, various activation functions for KAN have been proposed, such as wavelets [63], radial basis functions [64], Fourier series [65], finite basis functions [66], Jacobi basis functions [67], polynomial basis functions [68], and rational functions [69].

While KANs do not yield the governing equations of the underlying system, they offer a greater level of interpretability than conventional machine-learning methods that often operate as "black boxes". The interpretability of KAN structure lies in the accessibility of the internal mechanisms of the model, such as the activation functions $F_i(\mathbf{x})$, $G_i(\mathbf{x})$, and $K_i(\mathbf{x})$ shown in Fig. 1(b). This provides a more transparent view of how inputs are transformed into outputs, which allows for a better understanding of the underlying dynamics and how each function influences the system's behavior. This balance between performance and interpretability positions KANs as a promising alternative method for learning and understanding the dynamics of complex systems.

The accessibility of the activation function in the KAN structure is illustrated in Fig. 1(b) that presents two different KAN structures, highlighted in blue and green, respectively. The blue KAN has two inputs and two outputs without any hidden nodes, where the functions

$$\begin{aligned} M_1 &= F_1(\mathbf{x_n}) + F_3(\mathbf{y_n}), \\ M_2 &= F_2(\mathbf{x_n}) + F_4(\mathbf{y_n}) \end{aligned}$$

are linear combinations of the activation functions F_i for $i = 1, \ldots, 4$. The green structure has two extra hidden nodes where

$$\begin{aligned} L_1 &= K_1(G_1(\mathbf{x_n}) + G_3(\mathbf{y_n})) + K_3(G_2(\mathbf{x_n}) + G_4(\mathbf{y_n})), \\ L_2 &= K_2(G_1(\mathbf{x_n}) + G_3(\mathbf{y_n})) + K_4(G_2(\mathbf{x_n}) + G_4(\mathbf{y_n})). \end{aligned}$$

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.

2. Training and testing

To assess the performance of KANs in discovering dynamical systems, we develop a structured training and evaluation protocol. Our framework follows a three-stage process, in which the ground truth data is divided into three distinct subsets: a training set, a validation set, and a testing set. Each stage serves a specific purpose in ensuring both short-term predictive accuracy and long-term dynamical fidelity.

Open-loop training and validation. We employ an open-loop strategy for training, where KANs are trained to perform one-step-ahead predictions based on input-output mappings. The training dataset is used to optimize the network parameters, where KANs learn the underlying system dynamics by minimizing the error between predicted and actual values. Subsequently, validation is performed on a separate set of inputs that the network has never encountered before. This ensures that KANs generalize beyond the training data and do not overfit to any specific patterns. For both training and validation, the primary performance metric is the Root Mean Square Error (RMSE), which quantifies the deviation of the predicted from the actual values.

Closed-loop testing for long-term predictions. After establishing the short-term predictive accuracy, we assess the ability of KANs to capture the long-term system behavior in a closed-loop configuration. Here, KANs operate autonomously by recursively feeding their own predictions back into the system to generate multistep forecasts. This assessment verifies whether KANs have effectively learned the system's overall dynamics and can remain bounded within the attractor over extended time horizons. To validate the long-term performance, we compare the predicted trajectories with the ground truth from the third part of the dataset. This comparison relies on a statistical analysis to ensure that KANs preserve the key dynamical properties and accurately replicate the system's behavior over time.

B. Dynamical invariants and statistical analysis

A statistical analysis offers quantitative metrics to assess how accurately a KAN model replicates the underlying behavior of the target system, facilitating performance comparison in the testing phase. Quantities commonly used in the statistical analysis are Lyapunov exponents, power spectra, correlation dimensions, and statistical distances, allowing us to assess the accuracy and fidelity of the KAN-produced models in capturing the complex dynamics of the target system. Here we provide a brief description of each of these statistical quantities.

1. 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. For the KAN model, we compute the Lyapunov exponents using the standard numerical approach that involves the Jacobian matrix and QR decomposition [70]. 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. We use the standard finite difference method to evaluate the derivatives [71,72].

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 the exponent values.

2. Power spectrum

Power spectra provide another way to compare the KAN-produced model with the ground truth. It is a fundamental tool for analyzing the frequency components of a time series. It reveals how the power of a signal is distributed over different frequencies, thus providing insights into the periodic or aperiodic nature of the dynamics. To compare the KAN-produced model with the ground truth, we compute the power spectra for both time series. This involves transforming the time series data into the frequency domain using the Fourier transform. The resulting power spectrum allows us to identify dominant frequencies and their corresponding amplitudes.

3. 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 [73].

4. 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 [74,75]. 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:

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

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:

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

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.

III. RESULTS

A. KAN model discovery of the Ikeda optical-cavity map

From the standpoint of data-driven model discovery, the Ikeda map represents perhaps one of the most difficult kinds of systems – far there has been no success with any sparse optimization method. The two-dimensional map is given by [53,54]:

$$x_{n+1} = 1 + \mu(x_n \cos(\phi_n) - y_n \sin(\phi_n)), \tag{5}$$

$$y_{n+1} = \mu(x_n \sin(\phi_n) + y_n \cos(\phi_n)), \tag{6}$$

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 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 Appendix C for more details.

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 Appendix A 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 and validation losses as a function of time are shown as the red and black curve in

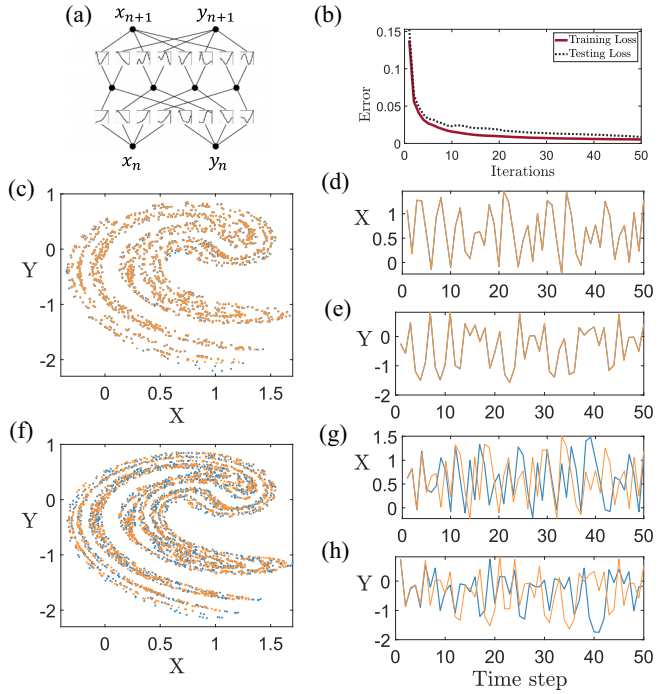

FIG. 2. KANs applied to the Ikeda map. (a) A KAN structure with 2 input, 4 hidden, and 2 output nodes. (b) Training (red) and validation (black dashed) loss curves. (c) Chaotic attractor during the validation phase (blue - ground truth; orange - KAN produced). (d), (e) Time series during the validation. 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.

Fig. 2(b), and the KAN-produced attractor and time series during the validation phase in comparison with the ground truth are shown in Figs. 2(c)–2(e), respectively. The training loss decreases rapidly to near 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, where the KAN structure operates as an autonomous system. The KAN predicted attractor and time series in the testing phase are shown in Figs. 2(f)–2(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 validation and prediction results are shown in Figs. 3(b)–3(h).

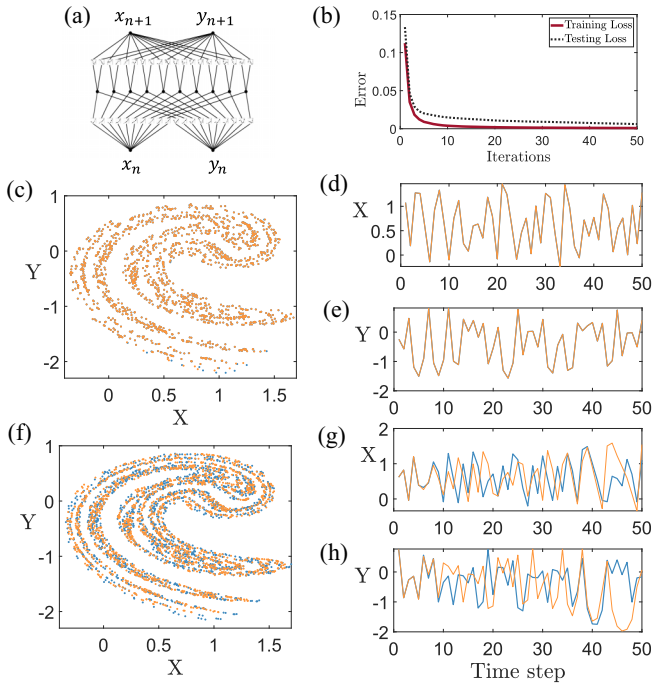

FIG. 3. A KAN configuration generating a different representation of the Ikeda map but with the same chaotic attractor. The KAN has 2 input, 10 hidden, and 2 output nodes. Legends are the same as those in Fig. 2. The two Lyapunov exponents of the KAN predicted model are [0.5033, −0.7311], which again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

B. KAN model discovery of a chaotic ecosystem

For generality, we now present results from a continuous-time system, a chaotic ecosystem [76] of three dynamical variables:

$$\dot{N} = N\left(1 - \frac{N}{K}\right) - x_p y_p \frac{NP}{N + N_0}, \tag{7}$$

$$\dot{P} = x_p P\left(y_p \frac{N}{N + N_0} - 1\right) - x_q y_q \frac{PQ}{P + P_0}, \tag{8}$$

$$\dot{Q} = x_q Q\left(y_q \frac{P}{P + P_0} - 1\right), \tag{9}$$

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 [76]. A power-series expansion of the velocity field contains an infinite number of terms, violating the sparsity condition - see Appendix C for more details.

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 1155 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 $G = 3$, $\lambda = 0$,

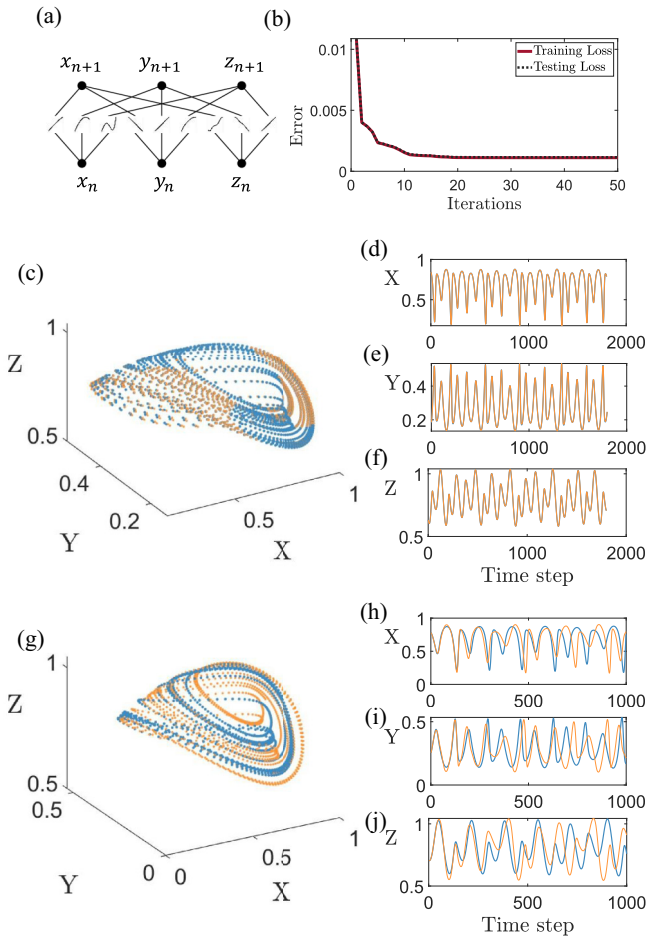

FIG. 4. KAN applied to a chaotic ecosystem. (a) KAN structure with 3 input and 3 output nodes. (b) Validation and testing loss curves. (c) KAN generated attractor during the validation phase (orange), which agrees well with the ground truth (blue). (d)–(f) KAN generated time series (orange) in agreement 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 $[0.0095, -5.8 \times 10^{-6}, -0.3932]$. The exponents of the KAN-generated attractor are consistent: $[0.0053, 0, -0.2288]$. The errors arise from the implicit numerical evaluation of the Jacobian matrix.

$\lambda_{\text{entropy}} = 10$, 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)–4(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)–4(j), demonstrating the KAN's forecasting power.

In addition to these results, we demonstrate the performance of KAN in discovering dynamical systems for three chaotic discrete maps: Logistic, Circle, and Henon maps. Results are presented in Appendix D.

TABLE I. Lyapunov exponents of the Ikeda dynamics.

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

C. Statistical analysis

The calculated Lyapunov exponents for the KAN-produced Ikeda and food-chain systems are listed in Tables I and II. Despite different models, the resulting Lyapunov exponents are essentially the same, revealing the same attractor. Figure 5 shows that the KAN model power spectra capture the underlying periodicities and complex oscillations present in the ground truth dynamics for both the Ikeda optical-cavity map and the food-chain system. The results listed in Table III 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.

The divergence measures results for both the Ikeda and Food-Chain systems and is presented in Table IV, 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: $\text{KL}(P||Q) = 1.0532$, $H(P, Q) = 0.6024$, and $\delta(P, Q) = 0.5999$. These results demonstrate that the KAN-produced attractors are essentially statistically identical to the ground truth (within numerical errors), highlighting the effectiveness of the KAN model in capturing the true system's behavior.

D. Robustness of KAN model discovery against noise

The Atlantic Meridional Overturning Circulation (AMOC) plays a critical role in regulating global climate patterns but is challenging to monitor directly due to the difficulty to obtain long-term observational data. As a result, researchers have utilized fingerprint analysis techniques, such as sea surface temperature (SST), as proxies to assess the AMOC strength and anticipate potential tipping [77–81]. SST has proven to be a reliable fingerprint due to its sensitivity to changes in ocean circulation and heat transport in the North Atlantic [82–84].

Ditlevsen and Ditlevsen [85] introduced a one-dimensional (1D) stochastic model to capture the tipping dynamics of the AMOC using a slowly evolving control parameter λ. This parameter represents a driver of the AMOC dynamics, evolving towards a critical value $\lambda_c = 0$, where a saddle-node

TABLE II. Lyapunov exponents of the food-chain dynamics.

Sys \ LEs	L_1	L_2	L_3
Ground Truth	0.009495	−0.00058	−0.393213
Food Chain	0.005354	−0.000005	−0.228759

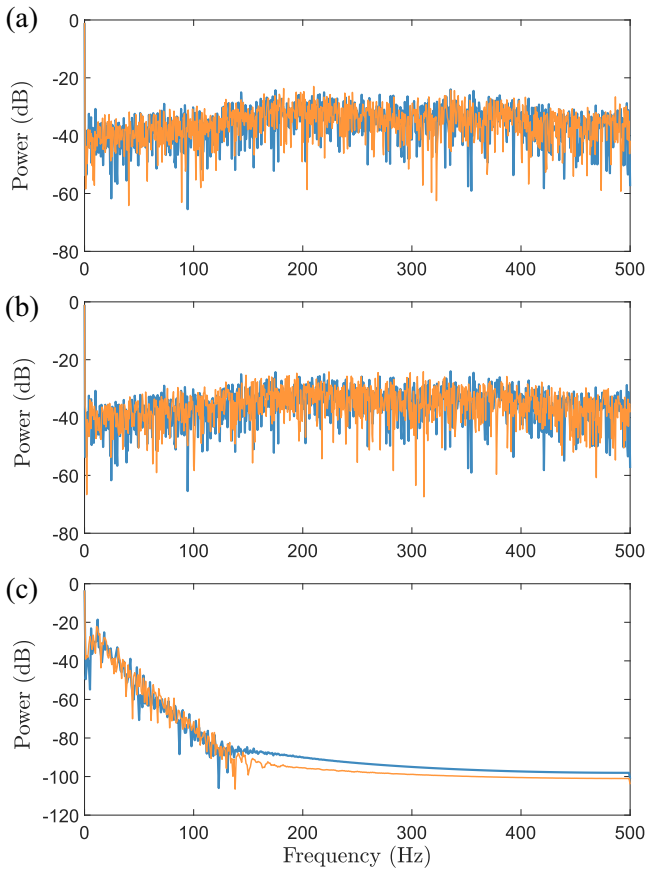

FIG. 5. Power spectrum comparison. Shown are the power spectra of the KAN-produced dynamics (orange) compared with the ground truth (blue): (a), (b) the two KAN models generating the same Ikeda dynamics as described in the text, and (c) the food-chain dynamics.

bifurcation occurs. The model dynamics are governed by the stochastic differential equation:

$$\dot{X}_t = -(A(X_t - m)^2 + \lambda) + \sigma dB_t, \tag{10}$$

$$\lambda = \begin{cases} \lambda_0, & \text{for } t < t_0, \\ \lambda_0\left(1 - \frac{t-t_0}{t_c-t_0}\right), & \text{for } t \geqslant t_0, \end{cases} \tag{11}$$

where X_t represents the stochastic AMOC fingerprint, A is a timescale parameter, $m = \mu - \sqrt{|\lambda|/A}$ with μ as the stable fixed point, B_t represents a Brownian motion, and σ is the noise amplitude. Initially, the system is stable with $\lambda = \lambda_0$, but starting from t_0, λ changes linearly, driving the system toward tipping. Although the theoretical tipping time corresponds to t_c, noise can induce an earlier transition.

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

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

TABLE IV. 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.

System \ Method	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

Figure 6(a) illustrates 10 realizations of the model with parameter values $A = 0.95$, $m = -1.3$, $\lambda_0 = -2.7$, $\sigma = 0.3$, $t_0 = 1924$, and $\lambda_c = 0$. The green and red curves depict the stable and unstable equilibria, respectively, showing the system's progression toward collapse as λ approaches its critical value. The KAN architecture used to model the noisy AMOC finger print data has a simple [2,2] structure (2 input and 2 output nodes, no hidden nodes), as illustrated in Fig. 6(b). The neural network was trained using 1500 data points from 10 different model realizations with the sampling interval $\delta t = 1$, with 80% of the data allocated for training and the remaining 20% for testing. The training process involved 150 iterations for the following hyperparameter values: cubic B-spline ($K = 3$), grid size $G = 3$, $\lambda = 0$, $\lambda_{\text{entropy}} = 10$, learning rate 0.1, and a zero initial random seed. Figure 6(c) shows the rapid decrease in the validation and testing loss with increasing epochs. The KAN generated time series during the training and testing phase are shown in Figs. 6(d) and 6(e), respectively, where a comparison with the ground truth indicates successful training and testing.

E. Physics-informed KANs and explainability in real systems and applications

It is important to demonstrate how KAN enhances explainability in real systems. As outlined in Refs. [58,59], a fundamental advantage of KANs lies in their greater interpretability in comparison with the conventional neural networks, and the goal of KANs is to facilitate "mechanistic interpretability," enabling the discovery of physical laws from data. Previous applications of KANs have already illustrated its potential to uncover physical concepts such as conserved quantities, Lagrangian, hidden symmetries, and constitutive laws [58,59].

An essential feature of KANs lies in its ability to accomplish some of complex tasks with a remarkably simple structure, requiring fewer nodes and layers than conventional neural networks. This simplicity not only reduces computational complexity but also enhances interpretability. For example, the univariate learnable activation functions in KANs are inherently more understandable than the dense weight matrices typical of conventional neural networks. However, scalability remains a challenge. When the scale of a KAN model grows, interpretability may diminish due to the increasing difficulty of analyzing the combined output of the 1D functions. Thus, the current implementation of KANs is

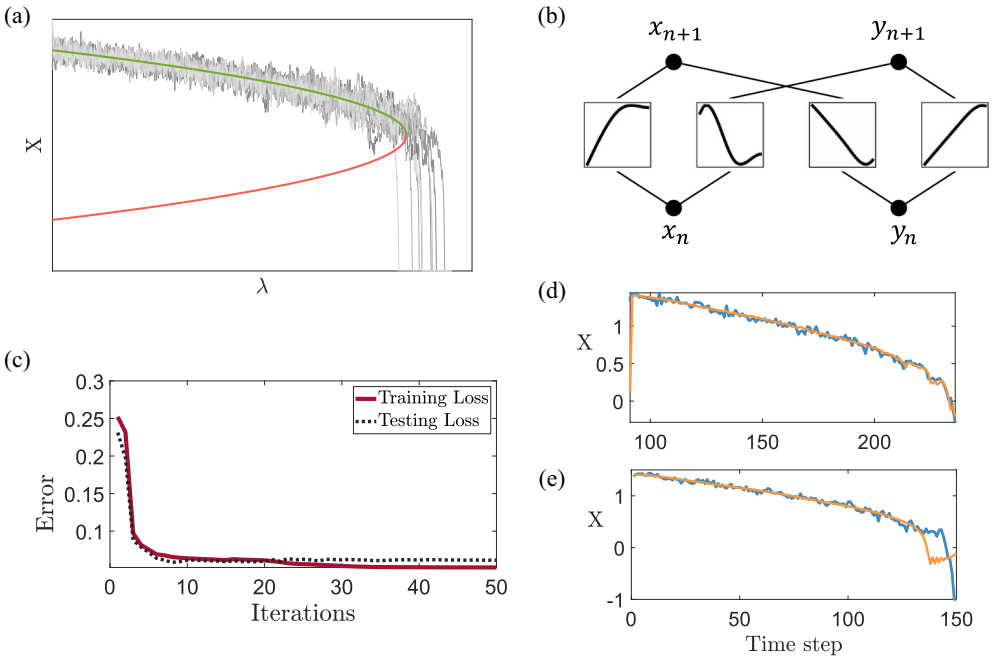

FIG. 6. KANs applied to the AMOC 1D model. (a) Ten different realizations of the 1D stochastic AMOC fingerprint variable for $A = 0.95$, $m = -1.3$, $\lambda_0 = -2.7$, $\sigma = 0.3$, $t_0 = 1924$, and $\lambda_c = 0$. The parameter values are from Ref. [85], which are the best estimates obtained from the AMOC fingerprint data. The green and red curves correspond to the stable and unstable equilibria of the model, respectively. (b) KAN structure with 2 input and 2 output nodes. (c) Validation and testing loss curves. (d) KAN generated time series (orange) in agreement with the true time series (blue). (e) Similar to (d) but for the testing phase.

most effective in scenarios where small-scale, interpretable networks suffice.

In Sec. III D, we applied the KAN framework directly to the noisy data without any prior knowledge of its origin. The results, illustrated in Fig. 6, demonstrate the robustness of KAN in learning the dynamics of the AMOC, even when subjected to high levels of noise. This highlights the capacity and robust performance of KANs in scenarios where the data are limited or noisy.

However, in real-world applications, prior knowledge about the system being studied is often quite limited. In such cases, leveraging the partial and limited information can significantly improve both the model structure and interpretability. To demonstrate this, we integrated domain knowledge into the KAN framework. Specifically, we inferred from the physics of the AMOC system that the driver of climate change (the control parameter λ), such as the freshwater flux or the logarithm of atmospheric CO_2 concentration, affects the AMOC fingerprint. However, the AMOC fingerprint does not exert an immediate effect on the control parameter itself.

Incorporating this causal knowledge, we modified the KAN structure by removing the link from the first input (representing x) to the second input (representing λ), as shown in Fig. 7(a). This adjustment not only aligns the model with the underlying physical reality but also enhances the explainability of the learned dynamics. Furthermore, we exploited a unique advantage of KAN-direct access to its activation functions. Examining these activation functions closely, as illustrated in Figs. 7(b)–7(d), we obtained additional insights into the system. For instance, the first activation function exhibits a quadratic (or square) shape, while the remaining two activation functions are approximately linear. Remarkably, this observation agrees perfectly with the mathematical structure of the original AMOC model, confirming the interpretability and reliability of the KAN framework.

By leveraging limited domain knowledge, the KAN framework becomes a powerful tool for bridging the gap between purely data-driven and SINDy-type of approaches. Its ability to incorporate prior information not only enhances its interpretability but also helps one get closer to finding the physical principles governing the system. This combination of data-driven learning and physics-informed modeling positions KAN as a compelling alternative for studying real-world systems, particularly those characterized by noisy and limited data.

IV. MATHEMATICAL INSIGHTS

To gain insights into the meaning of the interpretability of the KAN-discovered models, we offer some mathematical insights by interpreting the machine-learning modeling errors 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 **G** is said to produce identical behavior as the true system **F** if numerically computed orbits of **G** shadow some true orbits of **F**, at least for the observed finite time of the data set. For maps, if a true orbit of **F** is a sequence

$$\text{Orbit}_{\mathbf{F}}(\mathbf{x}_0) = \{\mathbf{x}_0, \mathbf{F}(\mathbf{x}_0), \mathbf{F}^2(\mathbf{x}_0)...\} \equiv \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, ...\},$$

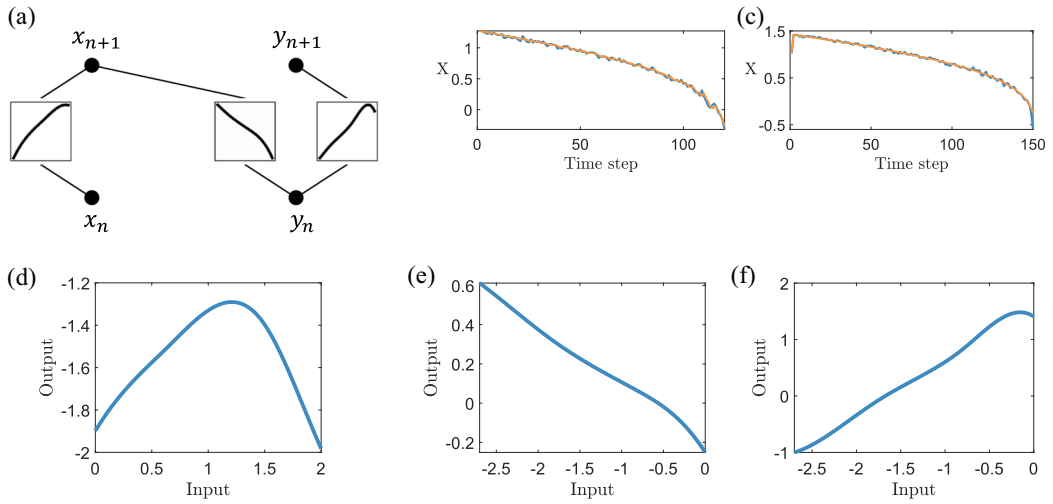

FIG. 7. Physics-informed KAN applied to AMOC fingerprint data. (a) Physics-informed KAN structure. (b), (c) Validation and testing KAN generated time series (orange) in agreement with the true time series (blue). (d)–(f) KAN activation functions.

it is unreasonable to expect that a good but imperfect model **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, $0 \leqslant |\epsilon_i| < \epsilon$. Nonetheless, a small normalized error of the function difference between the system and model alone does not prevent the model from producing an unrealistic orbit $\text{Orbit}_G(i\mathbf{x}_0)$ that behaves quite differently from any orbit of **F**, e.g., a model orbit that diverges to infinity even if the true orbit produces a 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 multilayer perceptron (MLP) [58] 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 [58], and therefore each has many internal fitted parameters of the scalar piecewise cubic. 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:

$$G_\Theta(x) = \sum_{q=1}^{2n+1} \Phi_{q,\Theta_q} \circ \sum_{p=1}^{n} \phi_{q,p,\Theta+q,p}(x_p), \tag{12}$$

and for a multivariate argument $x = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$. It is shown [58] 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" [58]. 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 [86]. 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 interpretable states, including sparsification.

V. DISCUSSION

Model discovery has been a central topic of interest for many years, with efforts generally falling into two main categories: machine learning methods and nonlinear-dynamics/sparse-optimization approaches, each with its own limitations. Machine learning techniques achieve high performance but function as black boxes, often offering little insight into the underlying system dynamics. In contrast, the sparse-optimization method provides interpretable mathematical equations but is only applicable to systems with an inherently sparse representation. In this paper, we leveraged KANs as a bridge between these two paradigms. KANs not only demonstrate predictive performance but also enhance interpretability by revealing how inputs influence outputs through their activation functions. We also investigated the robustness of our proposed method against noise by employing noisy synthetic data generated from the AMOC model. Our analysis revealed that the KAN framework remains effective even under noise, highlighting its potential for practical applications where data imperfections are inevitable. Furthermore, by integrating domain knowledge into the KAN framework, we demonstrated that our approach can enhance the interpretability of dynamical systems in the real-world AMOC system.

Overall, we exploited KANs as a data-driven model discovery method 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.

ACKNOWLEDGMENTS

The work at Arizona State University was supported by the Air Force Office of Scientific Research under Grant No. FA9550-21-1-0438. The work at Clarkson University was supported by the Army Research Office under Grant No. W911NF-17-S-0002.

DATA AVAILABILITY

All codes are available at [87].

APPENDIX A: KAN HYPERPARAMETERS

Hyperparameter tuning in KANs plays a crucial role not only in optimizing the 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.

TABLE V. Number of trainable parameters in KAN.

System \ Parameters	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

APPENDIX B: 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 ($N_{h1}, N_{h2}, ..., N_{hj}$), 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

$$N_a = (N_i \times N_{h1}) + (N_{h1} \times N_{h2}) + \cdots + (N_{hj} \times N_o). \quad \text{(B1)}$$

Consider the numeric training phase of KAN. Each activation function within the KAN is parametrized by a B-spline curve represented as a linear combination of the basis functions, as outlined in Eq. (3). Each B-spline curve is characterized by $(G + K)$ trainable coefficients. The total number of trainable parameters in a KAN is then given by

$$(G + K) \times N_a, \quad \text{(B2)}$$

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 V presents the number of trainable parameters for the KAN structures used to generate the Ikeda and food-chain dynamics.

APPENDIX C: SPECTACULAR FAILURES OF SPARSE OPTIMIZATION APPROACH TO FINDING EQUATIONS FOR THE IKEDA OPTICAL-CAVITY MAP AND CHAOTIC FOOD-CHAIN SYSTEM

The sparse-optimization approach to finding the governing equations of nonlinear dynamical systems from data was first

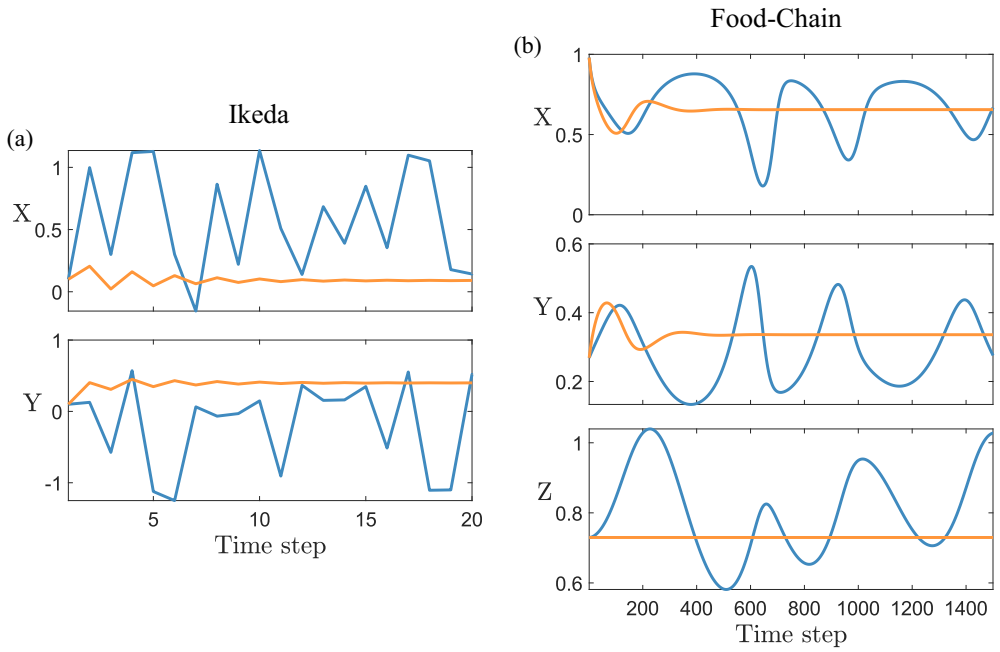

FIG. 8. Comparison of time series from the equations found by sparse optimization and the ground truth. Shown are two sets of time series (orange: equations from sparse optimization; blue: ground truth) for (a) Ikeda map and (b) chaotic food-chain system described in the main text. The sparse-optimization method fails to find the correct equations.

introduced in 2011 [10]. The idea is that power-series or 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 sparse optimization methods is that they require 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. Here we present an example of such a failure when attempting to estimate the dynamical equation of the Ikeda map using a commonly used sparse-optimization algorithm [49] that employs library of base functions including polynomials, inverse functions, products, exponential, and sinusoidal functions, etc. The estimated map functions are

$$\begin{aligned} x^+ =& 14.413e^x + 21.543e^y - 10.137x - 18.639y \\ &+ 5.308\sin(x) + 15.552\sin(y) - 41.853\cos(x) \\ &+ 6.222\cos(y) - 0.218\sin(x+y) - 10.137x \\ &- 18.638y - 28.135x^2 - 8.364y^2 + 0.152xy, \end{aligned}$$

$$\begin{aligned} y^+ =& -3.604e^x + 0.170e^y + 5.990x - 0.282y \\ &- 9.407\sin(x) + 3.029\cos(x) + 0.903\cos(y) \\ &+ 0.246\sin(x+y) + 5.990x - 0.283y \\ &+ 2.439x^2 + 0.493y^2 + 0.552xy. \end{aligned} \tag{C1}$$

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

$$\begin{aligned} x^+ =& 3.445e^x + 6.290\sin(x) + 0.869\sin(y) \\ &- 3.424\cos(x) + -0.391\sin(x+y) - 8.528x \\ &- 0.710y + -3.967x^2 + -1.363\frac{xy}{1+x}, \\ y^+ =& 0.572y^2 + 2.050xy - 3.879\frac{xy}{1+x} \\ z^+ =& 0. \end{aligned} \tag{C2}$$

The time series produced by these equations (orange) versus the ground truth (blue) are shown in Figs. 8(a) and 8(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.

APPENDIX D: KAN MODEL DISCOVERY OF THREE ADDITIONAL CLASSICAL NONLINEAR DYNAMICAL SYSTEMS

1. Logistic map

The map is given by [88]

$$X(n+1) = rX(n)(1 - X(n)) \tag{D1}$$

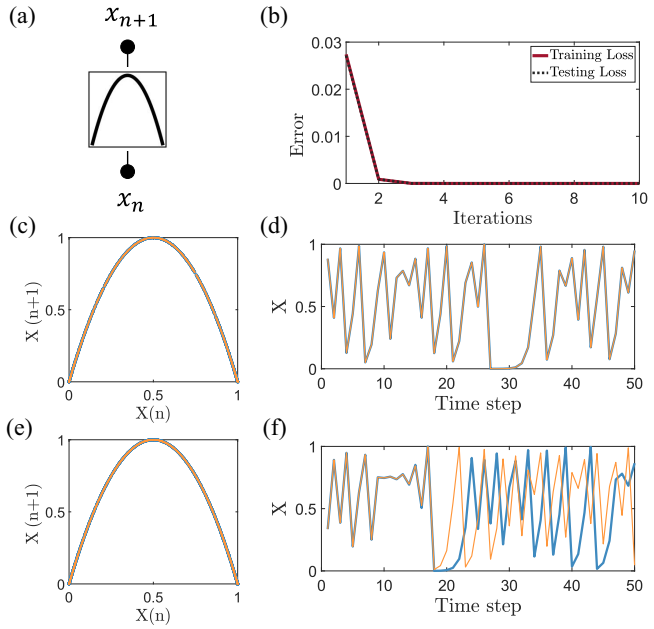

FIG. 9. KAN Model applied to the logistic map. (a) KAN model structure with one input and one output node, without hidden nodes. (b) Training and testing loss curves over 10 iterations. (c) Attractor and (d) time series during the training phase (orange) in comparison with the ground truth (blue), indicating 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.

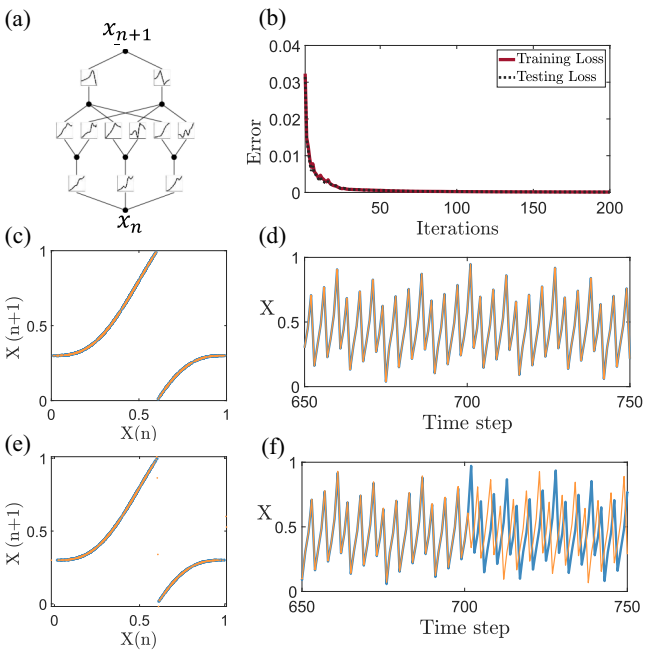

FIG. 10. KAN applied to the circle Map. (a) Structure of the KAN, which includes a single input and output node with two hidden layers. (b) Training and testing loss curves over 200 iterations. (c) Attractor and (d) time series during the training phase (orange) in comparison with the ground truth (blue), 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.

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 $X \in [0, 1]$.

We utilize a simple KAN structure, as depicted in Fig. 9(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. 9(b) shows the validation loss over time, while Figs. 9(c) and 9(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. 9(b) represents the validation loss, and Figs. 9(e) and 9(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.

2. Circle map

The map is given by

$$X(n+1) = X(n) + \Omega - \frac{K}{2\pi}\sin(2\pi X(n)) \mod 1, \quad \text{(D2)}$$

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 10(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%

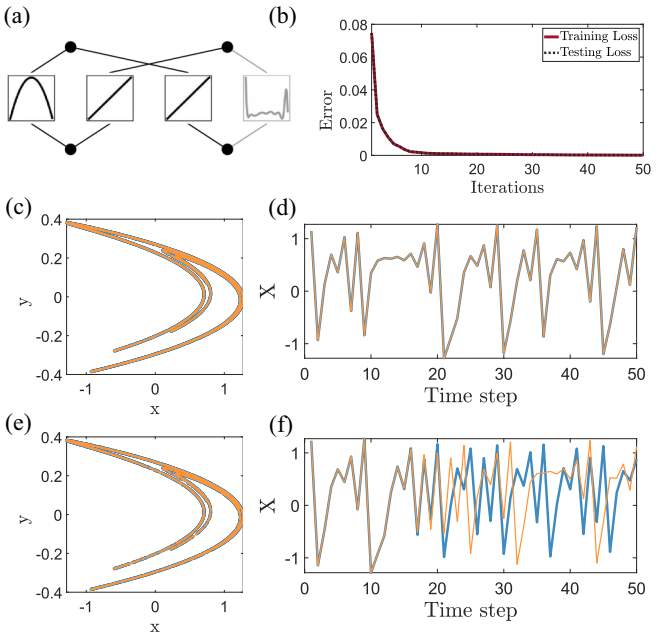

FIG. 11. KAN applied to the Hénon map. (a) KAN structure: two input and two output nodes, no hidden layers. (b) Training and testing loss curves over 50 iterations. (c) Attractor and (d) time series during the training phase (orange) in comparison with the ground truth. (e) Attractor and (f) time series during the testing phase, demonstrating that the KAN model effectively replicates the chaotic dynamics of the Hénon map.

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 $\lambda_{\text{ent}} = 10$, learning rate 0.1, and a random seed set to zero. The training (validation) is carried out in a feedforward (recurrent) process. The red (black dashed) curve in Fig. 10(b) illustrates the training (validation) loss over time. Figures 10(c), 10(e) and 10(d), 10(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.

3. Hénon map

The two-dimensional map is given by [89]

$$x(n+1) = 1 - ax(n)^2 + y(n),$$
$$y(n+1) = bx(n), \tag{D3}$$

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. 11(a), with two input and two output nodes. The dataset comprises 5×10^4 points with 90% for training

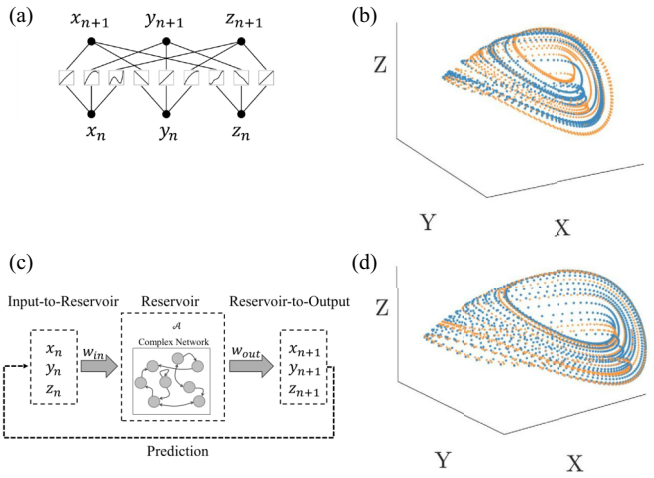

FIG. 12. Comparing KAN with reservoir computing applied to a chaotic ecosystem that does not possess a sparse structure. (a) KAN structure with 3 inputs and 3 outputs. (b) KAN-generated attractor during the testing phase (orange), which agrees with the ground truth (blue). (c) Testing structure of reservoir computing with 500 nodes. (d) Reservoir-computing generated attractor during the testing phase.

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. 11(b)–11(f) demonstrate that KAN represents a data-discovered model that faithfully generate the ground-truth Hénon chaotic dynamics.

APPENDIX E: KANs AND STATE-OF-THE-ART NEURAL NETWORKS

A direct comparison with existing state-of-the-art neural networks model is beyond the scope of this paper. Our primary goal is to demonstrate that complex dynamics can be effectively represented by KANs that feature a simple structure and offers enhanced interpretability compared to conventional neural network architectures.

Here we provide a comparison of the trainable parameters in the KAN and reservoir computing to highlight the computational efficiency of KANs. For instance, we applied KAN to a chaotic ecosystem governed by Holling-type of equations (Fig. 4 in the main text), a scenario where traditional methods such as SINDy fail. The results in Fig. 12 show that, while this complex system could also be learned using a reservoir-computing model with 1500 trainable parameters (using hyperparameters similar to those in Ref. [33]), the KAN framework achieved comparable accuracy with significantly lower computational complexity: it requires only nine activation functions and 54 trainable parameters. In addition, KAN provides a unique advantage in terms of interpretability, granting access to all nine learned univariate functions and 54 trainable parameters. This enables a deeper understanding of the system's underlying dynamics, an insight that is often inaccessible with conventional black-box models such as reservoir computing or transformers. This balance between simplicity, accuracy, and interpretability is the characteristic strength of KANs.

[1] J. P. Crutchfield and B. McNamara, Equations of motion from a data series, Complex Syst. **1**, 417 (1987).
[2] Y.-C. Lai, C. Grebogi, and J. Kurths, Modeling of deterministic chaotic systems, Phys. Rev. E **59**, 2907 (1999).
[3] Y.-C. Lai and C. Grebogi, Modeling of coupled chaotic oscillators, Phys. Rev. Lett. **82**, 4803 (1999).
[4] E. M. Bollt, Controlling chaos and the inverse Frobenius-Perron problem: Global stabilization of arbitrary invariant measures, Int. J. Bifurcation Chaos **10**, 1033 (2000).
[5] R. Hegger, H. Kantz, L. Matassini, and T. Schreiber, Coping with nonstationarity by overembedding, Phys. Rev. Lett. **84**, 4092 (2000).
[6] 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 Process. **49**, 2138 (2001).
[7] T. D. Sauer, Reconstruction of shared nonlinear dynamics in a network, Phys. Rev. Lett. **93**, 198701 (2004).
[8] 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).
[9] 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).
[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] 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).
[12] 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, Europhys. Lett. **94**, 48006 (2011).
[13] R.-Q. Su, W.-X. Wang, and Y.-C. Lai, Detecting hidden nodes in complex networks from time series, Phys. Rev. E **85**, 065201(R) (2012).
[14] 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).
[15] 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).
[16] A. A. R. AlMomani, J. Sun, and E. M. Bollt, How entropic regression beats the outliers problem in nonlinear system identification, Chaos **30**, 013107 (2020).
[17] P. Rajendra and V. Brahmajirao, Modeling of dynamical systems through deep learning, Biophys. Rev. **12**, 1311 (2020).
[18] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Machine learning prediction of critical transition and system collapse, Phys. Rev. Res. **3**, 013090 (2021).
[19] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Emergence of transient chaos and intermittency in machine learning, J. Phys. Complex. **2**, 035014 (2021).
[20] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, Reservoir computing as digital twins for nonlinear dynamical systems, Chaos **33**, 033111 (2023).
[21] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, Deep learning and process understanding for data-driven earth system science, Nature (London) **566**, 195 (2019).
[22] A. Almeida and G. Azkune, Predicting human behaviour with recurrent neural networks, Appl. Sci. **8**, 305 (2018).
[23] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. **8**, 6085 (2018).
[24] D. Durstewitz, G. Koppe, and M. I. Thurm, Reconstructing computational system dynamics from neural data with recurrent neural networks, Nat. Rev. Neurosci. **24**, 693 (2023).
[25] S. d'Ascoli, S. Becker, P. Schwaller, A. Mathis, and N. Kilbertus, ODEFormer: Symbolic regression of dynamical systems with transformers, arXiv:2310.05573.
[26] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report **148**, 13 (2001).
[27] W. Maass, T. Nachtschlaeger, and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. **14**, 2531 (2002).
[28] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).
[29] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. **120**, 024102 (2018).
[30] J. Jiang and Y.-C. Lai, Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius, Phys. Rev. Res. **1**, 033056 (2019).
[31] A. Griffith, A. Pomerance, and D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, Chaos **29**, 123108 (2019).
[32] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, Next generation reservoir computing, Nat. Commun. **12**, 5564 (2021).
[33] Z.-M. Zhai, M. Moradi, B. Glaz, M. Haile, and Y.-C. Lai, Machine-learning parameter tracking with partial state observation, Phys. Rev. Res. **6**, 013196 (2024).
[34] Z.-M. Zhai, M. Moradi, L.-W. Kong, B. Glaz, M. Haile, and Y.-C. Lai, Model-free tracking control of complex dynamical trajectories with machine learning, Nat. Commun. **14**, 5698 (2023).
[35] Y.-C. Lai, Digital twins of nonlinear dynamical systems: A perspective, Eur. Phys. J. Spec. Top. **233**, 1391 (2024).
[36] S. Panahi, L.-W. Kong, M. Moradi, Z.-M. Zhai, B. Glaz, M. Haile, and Y.-C. Lai, Machine learning prediction of tipping in complex dynamical systems, Phys. Rev. Res. **6**, 043194 (2024).
[37] J. D. Farmer and J. J. Sidorowich, Predicting chaotic time series, Phys. Rev. Lett. **59**, 845 (1987).
[38] G. Gouesbet, Reconstruction of standard and inverse vector fields equivalent to a Rössler system, Phys. Rev. A **44**, 6264 (1991).
[39] T. Sauer, Reconstruction of dynamical systems from interspike intervals, Phys. Rev. Lett. **72**, 3811 (1994).
[40] E. Baake, M. Baake, H. G. Bock, and K. M. Briggs, Fitting ordinary differential equations to chaotic data, Phys. Rev. A **45**, 5524 (1992).

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

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

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

[44] 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).

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

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

[47] R. G. Baraniuk, Compressed sensing, IEEE Signal Process. Mag. **24**, 118 (2007).

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

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

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

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

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

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

[54] 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).

[55] 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 (1959).

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

[57] 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).

[58] 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.

[59] Z. Liu, P. Ma, Y. Wang, W. Matusik, and M. Tegmark, KAN 2.0: Kolmogorov-Arnold networks meet science, arXiv:2408.10205.

[60] 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).

[61] A. B. Givental, B. A. Khesin, J. E. Marsden, A. N. Varchenko, V. A. Vassiliev, O. Y. Viro, and V. M. Zakalyukin, 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, 2009), pp. 25–46.

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

[63] Z. Bozorgasl and H. Chen, Wav-KAN: Wavelet Kolmogorov-Arnold networks, arXiv:2405.12832.

[64] Z. Li, Kolmogorov-Arnold networks are radial basis function networks, arXiv:2405.06721.

[65] J. Xu, Z. Chen, J. Li, S. Yang, W. Wang, X. Hu, and E. C.-H. Ngai, FourierKAN-GCF: Fourier Kolmogorov-Arnold network–an effective and efficient feature transformation for graph collaborative filtering, arXiv:2406.01034.

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

[67] A. A. Aghaei, fKAN: Fractional Kolmogorov-Arnold networks with trainable Jacobi basis functions, Neurocomputing **623** 129414 (2025).

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

[69] A. A. Aghaei, rKAN: Rational Kolmogorov-Arnold networks, arXiv:2406.14495.

[70] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Determining Lyapunov exponents from a time series, Physica D **16**, 285 (1985).

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

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

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

[74] T. M. Cover, *Elements of Information Theory* (John Wiley & Sons, New York, 1999).

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

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

[77] L. C. Jackson and R. A. Wood, Fingerprints for early detection of changes in the AMOC, J. Clim. **33**, 7027 (2020).

[78] J. Robson, D. Hodson, E. Hawkins, and R. Sutton, Atlantic overturning in decline? Nat. Geosci. **7**, 2 (2014).

[79] R. K. Haskins, K. I. C. Oliver, L. C. Jackson, R. A. Wood, and S. S. Drijfhout, Temperature domination of AMOC weakening due to freshwater hosing in two GCMs, Clim. Dyn. **54**, 273 (2020).

[80] M. Ben-Yami, V. Skiba, S. Bathiany, and N. Boers, Uncertainties in critical slowing down indicators of observation-based fingerprints of the Atlantic overturning circulation, Nat. Commun. **14**, 8344 (2023).

[81] R. Zhang, On the persistence and coherence of subpolar sea surface temperature and salinity anomalies associated with the

Atlantic multidecadal variability, Geophys. Res. Lett. **44**, 7865 (2017).

[82] G. McCarthy, D. Smeed, W. Johns, E. Frajka-Williams, B. Moat, D. Rayner, M. Baringer, C. Meinen, J. Collins, and H. Bryden, Measuring the Atlantic meridional overturning circulation at 26°*n*, Prog. Oceanogr. **130**, 91 (2015).

[83] G. D. McCarthy, I. D. Haigh, J. J.-M. Hirschi, J. P. Grist, and D. A. Smeed, Ocean impact on decadal Atlantic climate variability revealed by sea-level observations, Nature (London) **521**, 508 (2015).

[84] M. S. Lozier, F. Li, S. Bacon, F. Bahr, A. S. Bower, S. A. Cunningham, M. F. de Jong, L. de Steur, B. deYoung, J. Fischer, S. F. Gary, B. J. W. Greenan, N. P. Holliday, A. Houk, L. Houpert, M. E. Inall, W. E. Johns, H. L. Johnson, C. Johnson, J. Karstensen *et al.*, A sea change in our view of overturning in the subpolar north Atlantic, Science **363**, 516 (2019).

[85] P. Ditlevsen and S. Ditlevsen, Warning of a forthcoming collapse of the Atlantic meridional overturning circulation, Nat. Commun. **14**, 4254 (2023).

[86] K. G. Binmore, *Mathematical Analysis: A Straightforward Approach* (Cambridge University Press, Cambridge, 1982).

[87] https://github.com/AminMoradiXL/KAN_paper/.

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

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