

Physics-informed time series analysis with Kolmogorov-Arnold networks under Ehrenfest constraints

Abhijit Sen^{1,*}, Illya V. Lukin^{2,3,†}, Kurt Jacobs^{4,5,‡}, Lev Kaplan^{1,§}, Andrii G. Sotnikov^{2,3,||} and Denys I. Bondar^{1,¶}

¹*Department of Physics and Engineering Physics, Tulane University, New Orleans, Louisiana 70118, USA*

²*Karazin Kharkiv National University, Svobody Square 4, 61022 Kharkiv, Ukraine*

³*Akhiezer Institute for Theoretical Physics, NSC KIPT, Akademichna 1, 61108 Kharkiv, Ukraine*

⁴*United States DEVCOM Army Research Laboratory, Adelphi, Maryland 20783, USA*

⁵*Department of Physics, University of Massachusetts at Boston, Boston, Massachusetts 02125, USA*



(Received 24 September 2025; accepted 11 March 2026; published 6 April 2026)

The prediction of quantum dynamical responses lies at the heart of modern physics. Yet, modeling these time-dependent behaviors remains a formidable challenge because quantum systems evolve in high-dimensional Hilbert spaces, often rendering traditional numerical methods computationally prohibitive. While large language models have achieved remarkable success in sequential prediction, quantum dynamics presents a fundamentally different challenge: forecasting the entire temporal evolution of quantum systems rather than merely the next element in a sequence. Existing neural architectures such as recurrent and convolutional networks often require vast training datasets and suffer from spurious oscillations that compromise physical interpretability. In this work, we introduce an approach: Kolmogorov-Arnold networks (KANs) augmented with physics-informed loss functions that enforce the Ehrenfest theorems. Our method achieves superior accuracy with significantly less training data: It requires only 5.4% of the samples (200) compared to temporal convolution networks (3700). We further introduce the chain of KANs, an architecture that embeds temporal causality directly into the model design, making it particularly well suited for time series modeling. Our results demonstrate that physics-informed KANs offer a compelling advantage over conventional black-box models, maintaining both mathematical rigor and physical consistency while dramatically reducing data requirements.

DOI: [10.1103/84vm-tvzm](https://doi.org/10.1103/84vm-tvzm)

I. INTRODUCTION

The main purpose of physics is to furnish mathematical models that describe the dynamics of observables. The model should have predictive power—Given data about the present and perhaps the past, it should provide an estimator of observable quantities that well approximates actual values observed in future dynamics.

While machine learning (ML) approaches can achieve initial success with time series forecasting, they fundamentally fail to capture the underlying physical laws governing real-world systems. These black-box models often violate conservation principles, produce physically impossible predictions, and require excessive training data to rediscover

known physics. Hence, the task is to develop physics-informed time series analysis. This is easier said than done since theoretical physics deals with very different mathematical models such as quantum and classical mechanics. It turns out that nearly all physical theories can be derived from the corresponding Ehrenfest theorems [1]—These are dynamical constraints relating different physical observables.

For example, both nonrelativistic classical and quantum particles moving in one dimension without dissipation obey the Ehrenfest theorems [2]:

$$\frac{d}{dt}\langle\hat{x}\rangle = \frac{1}{m}\langle\hat{p}\rangle, \quad \frac{d}{dt}\langle\hat{p}\rangle = \langle F(\hat{x})\rangle, \quad (1)$$

where $\langle\hat{x}\rangle = \langle\hat{x}\rangle(t)$ and $\langle\hat{p}\rangle = \langle\hat{p}\rangle(t)$ are time series for a particle's position and momentum, m is the particle's mass, and $F(\hat{x})$ is a shorthand notation for the sum of high-order statistical moments of the coordinate $\langle F(\hat{x})\rangle = \sum_n c_n \langle\hat{x}^n\rangle$. Note that these differential relations cannot be solved since they do not form a closed system of equations for the unknown $\langle\hat{x}\rangle$ and $\langle\hat{p}\rangle$ [3]. However, we can infer the classical Liouville equation of motion from the Ehrenfest theorems (1) assuming $\hat{x}$ and $\hat{p}$ commute, whereas the Schrödinger equation is obtained if $\hat{x}$ and $\hat{p}$ do not commute [1]. This Ehrenfest-centric approach to physics has led to a number of findings in quantum control [4,5], formulating relativistic dynamics [6,7], open system theory [8,9], entropic forces [10], and entropic gravity [11,12].

*Contact author: abhijit913@gmail.com

†Contact author: illya.lukin11@gmail.com

‡Contact author: dr.kurt.jacobs@gmail.com

§Contact author: lkaplan@tulane.edu

||Contact author: a_sotnikov@kipt.kharkov.ua

¶Contact author: dbondar@tulane.edu

This motivates us to merge time series analysis with the Ehrenfest theorems.

Modeling how many-body quantum systems respond to external fields presents significant challenges due to their nonlinear behavior, high dimensionality, and sensitivity to initial conditions. This is clearly seen in high harmonic generation (HHG), where strong input pulses create output signals at multiple frequencies of the input. HHG signals are complex because the interplay of quantum coherence and many-body interactions produces broadband, nonstationary waveforms with rapidly varying amplitudes and phases across multiple harmonics. Further, HHG has garnered significant interest across scientific and technological domains due to its role in generating coherent extreme ultraviolet and soft x-ray radiation. In ultrafast spectroscopy, HHG enables attosecond time resolution, thereby allowing real-time tracking of electronic dynamics [13]. The ability to generate high-order harmonics has catalyzed advances in nanoscale imaging and lithography [14,15]. Recent studies indicate that HHG can enable single-atom computing [16] and enhance spectroscopic characterization of complex mixtures [17]. Within quantum spin systems, HHG serves as a probe for spin dynamics and holds promise for terahertz laser generation [18–20]. In our work, the driven spin-chain model belongs precisely to this class of spin-based HHG systems, where the nonlinear time evolution of magnetization under external driving is the microscopic origin of the harmonic spectrum. Accurate modeling of such nonlinear quantum-optical processes allows for rapid prediction of system responses to arbitrary optical inputs, eliminating the need for expensive trial-and-error experimentation.

With the increasing availability of experimental and theoretical data in HHG, ML methods offer substantial advantages over traditional first-principles modeling. For example, Lytova *et al.* [21] demonstrated that a deep neural network (DNN) can be trained to predict HHG spectra from molecular parameters (laser intensity, internuclear distance, molecular orientation) and solve the inverse problem of determining molecular parameters from observed HHG spectra or dipole data. In the macroscopic regime, Serrano *et al.* [22] used DNN to overcome the computational impossibility of exactly simulating HHG by coupling the three-dimensional time-dependent Schrödinger equation with Maxwell’s equations, enabling macroscopic HHG simulations that reveal hidden attosecond pulse signatures missed by standard approximations. Further, ML has also been applied to HHG in broader contexts: from harmonic generation microscopy [23], nonlinear spectroscopy in solids [24], and plasma physics where particle-in-cell simulations were combined with ML to predict harmonic generation features [25], to molecular HHG modeling and spectrum inversion [21], time series forecasting of nonlinear spectra [26], and prediction of HHG driven by spatially structured input fields [27]. Beyond direct HHG spectral modeling, DNNs have further been used for ultrafast science tasks such as ultrashort pulse reconstruction [28,29], estimation of free-electron laser pulse characteristics [30], and denoising of experimental photoelectron spectra [31]. In Ref. [32], unlike in earlier HHG-ML work, the authors train a single ML model on many input-output signal pairs, achieving dataset-wide generalization rather than

fitting a separate model for each case. The modeling involved the use of a temporal convolution network (TCN), a physics-uninformed approach in the sense that the model is only made aware of the dataset and not the underlying physics principle. While effective, this approach presents key limitations (see Appendix B): computational overhead from convolutional matrix operations, a dependency on large datasets for reliable fitting, and the emergence of residual oscillations in the output time series, indicative of nonsmooth predictions.

In this work, we fuse a state-of-the-art Artificial Intelligence technique—the Kolmogorov-Arnold network [33] (KAN)—with Ehrenfest theorems to develop a physically informed ML toolbox—the KAN-Ehrenfest time series analysis (KAN-ETS). The versatility of this technique is illustrated by modeling the nonlinear optical response of correlated many-body quantum systems. It is also shown that KAN-ETS outperforms the current state-of-the-art “physics-uninformed” time series analysis when used on the same dataset.

We organize the paper as follows: Section II introduces the input-output time series mapping framework and formulates the learning problem in terms of paired input and output sequences. Section III presents physics-informed Kolmogorov-Arnold networks (PIKANs) with Ehrenfest constraints, establishing the theoretical foundation of our approach. Section IV describes the datasets, model architectures, loss functions, and optimization strategies that we employ in our study. Section IV also introduces the chain of KANs architecture, which embeds temporal causality directly into the modeling process. Section V reports numerical results across single- and multi-amplitude datasets, evaluating predictive accuracy and architectural stability. Finally, Sec. VI concludes with a summary of our contributions and outlines potential directions for extending physics-informed KAN frameworks to broader classes of dynamical systems.

II. INPUT-OUTPUT TIME SERIES MAPPING

In input $\rightarrow$ output time series modeling, the dataset is typically organized into multiple paired sequences, each capturing a system’s response to a time-varying external input or stimulus. This setup reflects a common structure in scientific problems, where understanding the temporal relationship between cause (input) and effect (output) is crucial. Each pair in the dataset corresponds to a single system run, where the input shapes the dynamics of the system and the output reflects the time evolution of relevant physical quantities.

To facilitate machine learning approaches in such continuous-time systems, both the input and output functions are discretized at uniform time steps δt . This discretization transforms the continuous input signal [say, $h(t)$] into a vector $\mathbf{h} = [h_1, \dots, h_T]$ and the continuous output signal [say, $Y(t)$] into a vector $\mathbf{Y} = [Y_1, \dots, Y_T]$, where $h_k = h(k\delta t)$ and $Y_k = Y(k\delta t)$. The learning task is then framed as discovering a mapping $\mathbf{h} \mapsto \mathbf{Y}$ that captures the entire dynamical transformation of the system over time. In order to train the machine learning model to do the mapping, we need an entire dataset of signals. The dataset can be expressed in the

general form

$$\mathcal{D}(A) \equiv \{\mathbf{h}^{(i)} \rightarrow \mathbf{Y}^{(i)}\}_{i=1}^n. \quad (2)$$

The complete dataset $\mathcal{D}(A)$ is subsequently partitioned into training and test subsets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, to enable supervised learning and unbiased performance evaluation.

Unlike traditional autoregressive forecasting models that focus on predicting the next time step, this formulation seeks a global sequence-to-sequence transformation, allowing the model to infer complex temporal dependencies and nonlinear behavior across the full input duration.

Further note that our approach explicitly requires the data to be generated and represented on a fixed, uniform time grid with step δt . Once temporal resolution δt is chosen, the model learns a mapping $\mathbf{h} \mapsto \mathbf{Y}$ between vectors of fixed length. The current approach is not discretization invariant; rather, it is tied to a physically meaningful sampling interval δt . An alternative approach would be to make the KAN-based architecture resolution agnostic, e.g., via multiresolution training or Fourier-operator layers.

III. PHYSICS-INFORMED KAN WITH EHRENFEST CONSTRAINTS

Physics-informed neural networks (PINNs) represent a paradigm shift in scientific machine learning by incorporating fundamental physical laws directly into the learning process of neural networks [34–38]. By embedding physical constraints through partial differential equations into the loss function, PINNs enable the learning of solutions that are both data consistent and physically plausible. Following the revolutionary impact of PINNs and the recent introduction of KANs, the scientific machine learning community has witnessed the emergence of PIKANs [39], addressing the drawbacks of PINNs [40–45].

KANs are grounded in the Kolmogorov-Arnold representation theorem [46], which provides exact function decomposition rather than mere approximation. Unlike DNNs that act as black boxes, KANs replace static activations with trainable spline-based edge functions, enabling compact architectures that capture the compositional structures inherent in physics. Furthermore, beyond achieving better accuracy, KANs act as a scientific discovery engine through sparsification, pruning, and symbolification; they can recover interpretable formulas from data. More details on KANs can be found in Appendix A.

Let us now delve into the general mathematical foundation of a constrained physics model for our input $\rightarrow$ output dataset. Consider the Hamiltonian

$$H = -\frac{J_z}{4} \sum_{i=1}^{N-1} \sigma_i^z \sigma_{i+1}^z - \frac{h_x + f(t)}{2} \sum_{i=1}^N \sigma_i^x - \frac{h_z}{2} \sum_{i=1}^N \sigma_i^z, \quad (3)$$

where J_z is the ferromagnetic coupling between neighboring spins, while h_x and h_z are the amplitudes of the transverse and longitudinal magnetic fields, respectively. The external driving $f(t)$ is considered to be our input data.

Then, we have

$$\begin{aligned} H_{\text{Ising}} &= -\frac{J_z}{4} \sum_{i=1}^{N-1} \sigma_i^z \sigma_{i+1}^z, \\ H_x &= -\frac{h_x + f(t)}{2} \sum_{i=1}^N \sigma_i^x, \\ H_z &= -\frac{h_z}{2} \sum_{i=1}^N \sigma_i^z. \end{aligned} \quad (4)$$

We define the total magnetization in the x direction,

$$Y_x = \sum_{i=1}^N \sigma_i^x. \quad (5)$$

Following Ehrenfest's theorem, the time derivative of the expectation value of any operator A is given by

$$\frac{d}{dt} \langle A \rangle = i \langle [H, A] \rangle + \left\langle \frac{\partial A}{\partial t} \right\rangle. \quad (6)$$

In our case, $A = Y_x$ does not have explicit time dependence, so

$$\frac{d}{dt} \langle Y_x \rangle = i \langle [H, Y_x] \rangle. \quad (7)$$

A quick calculation reveals the following relationship:

$$\frac{d}{dt} \langle Y_x \rangle = \frac{J_z}{2} \sum_{n=1}^{N-1} \langle \sigma_n^y \sigma_{n+1}^z + \sigma_n^z \sigma_{n+1}^y \rangle + h_z \sum_{i=1}^N \langle \sigma_i^y \rangle. \quad (8)$$

Note that since the state $|\psi(t)\rangle$ itself evolves in time, the expectation value $\langle Y_x \rangle(t) = \langle \psi(t) | Y_x | \psi(t) \rangle$ is explicitly a function of t . It is common to write

$$m_x(t) \equiv \langle Y_x \rangle(t) = \langle \psi(t) | \sum_{i=1}^N \sigma_i^x | \psi(t) \rangle. \quad (9)$$

Generally, we want our neural network predictions to obey Ehrenfest equations. For this reason, we will define our loss function as the sum of the minimal squared error for the signals and an additional penalty term enforcing the Ehrenfest equations. In numerical simulations, the right-hand side of the Ehrenfest equations can be computed simply using a finite-difference derivative of the signal $\langle Y_x \rangle(t)$. However, for the real experimental data, the signal is generally noisy, so it makes sense to additionally measure directly the operators appearing on the right-hand side of the Ehrenfest equations and to use them in the loss function. In this sense, the Ehrenfest equations define additional data, which do not appear in the model predictions or inputs, but help to define the loss.

To conclude this section, we note that the signal (9) learned by our KAN model is directly connected to HHG in quantum spin chains. As shown in Ref. [18], the HHG emission intensity of a driven ferromagnetic chain is given by $I(\omega) \propto \omega^2 |M(\omega)|^2$, where $M(\omega) = \int dt e^{i\omega t} m_x(t)$. Thus, the time-domain magnetization curves presented in Fig. 1 already encode the full spin-HHG response, and the corresponding spectra are obtained by a straightforward Fourier transform.

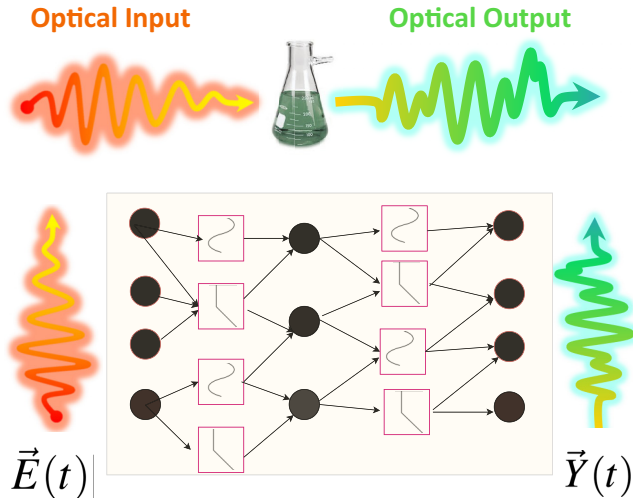


FIG. 1. Optics as an input-output problem. An input field $\mathbf{E}(t)$ (red) interacts with a medium (beaker) to produce an output field $\mathbf{Y}(t)$ (green). A KAN, trained on $[\mathbf{E}(t), \mathbf{Y}(t)]$ pairs, serves as a surrogate model of the medium, enabling prediction of $\mathbf{Y}(t)$ for previously unseen inputs $\mathbf{E}(t)$.

IV. APPROACH

A. Model and datasets

To establish the correspondence between the input and output data, we study the dynamics of the one-dimensional transverse-field Ising model (3) under a time-dependent external driving field $f(t)$.

Given the external driving $f(t)$ (the input data), to obtain the output data we simulate the dynamics with the Hamiltonian (3) starting from the ferromagnetic product state in the z direction. At each time step, we measure the magnetization (5), which corresponds to the output data. To gain access to the dynamical observables, we simulate the Schrödinger equation for the given Hamiltonian on infinite spin chains using the infinite time-evolving block decimation (iTEBD) algorithm [47] with the maximal bond dimension $D = 100$. The values of the Hamiltonian parameters are chosen to be $(J_z, h_z, h_x) = 0.8 \times (1, 0.25, -0.525)$. As for the external driving, we employ the sinusoidal signals $f(t) = A \sin(\omega t)$ with different driving frequencies ω and amplitudes A . In all cases, we calculate the evolution for $N_T = 500$ time steps with maximal time defined as $T = 2\pi/\omega_{\min}$, with $\omega_{\min}$ the minimal driving frequency.

With these data, we should now train the KAN (or several KANs) to fit the magnetization data based on the input data $f(t)$. Note that we employ an additional `MinMaxScaler` from `scikit-learn` on both input and output data to map it into the interval $[0, 1]$, which is shown in several figures below. This data normalization generally improves model training and reduces possible biases from large values.

We study the following four datasets:

(1) Modeling the input $\rightarrow$ output relation for the transverse Ising model based on a precalculated dataset $\mathcal{D}(A = 2.6)$ of input-output pairs of time series:

$$\mathcal{D}(A) = \{\mathbf{h}^{(i)} \rightarrow \mathbf{Y}^{(i)}\}, \quad (10)$$

where $h_k^{(i)} = A \sin(\omega^{(i)} k \delta t)$ represents the k th component of the i th sample of input vector, and where $\omega^{(i)} \in [0.4, 4]$, with either $N_\omega = 200$ or $N_\omega = 400$ frequencies.

(2) Modeling the input $\rightarrow$ output relation for the transverse Ising model from dataset $\mathcal{D}(A = 10)$. Frequencies are $\omega^{(i)} \in [0.4, 4.0]$ and $N_\omega = \{200, 600\}$.

(3) Modeling the input $\rightarrow$ output relation for the transverse Ising model based on the dataset $\mathcal{D}(A_1 = 0.4, A_m = 2.6)$ with inputs of $m = 8$ different intensities,

$$\mathcal{D}(A_1, A_m) = \{\mathbf{h}^{(i,j)} \rightarrow \mathbf{Y}^{(i,j)}\}, \quad (11)$$

$$h_k^{(i,j)} = A_j \sin(\omega^{(i)} k \delta t), \quad (12)$$

$$A_j = A_1 + (j - 1)(A_m - A_1)/(m - 1), \quad (13)$$

where $h_k^{(i,j)}$ represents the k th component of the i th sample of input vector corresponding to the amplitude A_j . Here, A_1 and A_m denote the smallest and largest amplitudes, respectively. The frequencies are taken in the range $\omega^{(i)} \in [0.4, 3]$ with $N_\omega = 200$.

(4) Modeling the input $\rightarrow$ output relation of the transverse Ising model on the dataset $\mathcal{D}(A_1 = 1.0, A_m = 10.0)$, $m = 10$, $\omega^{(i)} \in [0.4, 4]$, and $N_\omega = 400$.

Note that the datasets with input amplitude $A = 10$ are much more complex. In addition, the trainability may depend on the number of frequencies. This is the reason for using datasets with different numbers of frequencies (while all other parameters are fixed) in some cases under study.

B. Loss function and problem statement

Our goal is to predict the output magnetization signal from a given input driving field. We discretize both input and output signals at uniform time intervals with sufficiently small time steps. The input data are represented as $\mathbf{h}^{(i)} = [h_1^{(i)}, h_2^{(i)}, \dots, h_k^{(i)}, \dots, h_{N_T}^{(i)}]$, where k indexes the time steps and i indexes different input signals (corresponding to different driving frequencies and amplitudes). Similarly, we define the output signals as $\mathbf{Y}^{(i)} = [Y_1^{(i)}, Y_2^{(i)}, \dots, Y_{N_T}^{(i)}]$. Our objective is to train a KAN model $\hat{\mathbf{Y}}$ that maps input signals $\mathbf{h}^{(i)}$ to output signals $\mathbf{Y}^{(i)}$.

We construct a loss function with two components that enforce complementary constraints on the model predictions. The first term ensures that model predictions match the target output signals using a mean squared error (MSE) criterion. The second term is a derivative-based penalty inspired by the Ehrenfest equation (8), which relates the time derivative of the magnetization to expectation values of other observables. In principle, the right-hand side of the Ehrenfest equation (8) should be evaluated by measuring the corresponding operators. However, when the training data are obtained from a high-fidelity numerical simulation of the Schrödinger equation (as in our case), the noise level is negligibly small. Under these conditions, the finite-difference derivative of the output signal $\mathbf{Y}^{(i)}$ provides an accurate approximation to the right-hand side of the Ehrenfest relation. We have verified this explicitly in Appendix C: The MSE between the derivatives computed via finite differences and those obtained from direct operator measurements within the iTEBD simulation is of the

order 10^{-6} , with the relative error below 1% in most cases. This justifies the use of numerical derivatives as a practical replacement in the loss function when training on low-noise computational data. For noisy experimental data, in contrast, finite differences amplify the noise, and it becomes essential to replace them with the operator expectation values appearing on the right-hand side of Eq. (8), which can be measured directly. As demonstrated in Appendix D, KAN predictions remain stable even in the presence of significant noise in the training data.

The complete loss function is defined as

$$\mathcal{L} = \text{MSE}(\hat{\mathbf{Y}}(\mathbf{h}^{(i)}), \mathbf{Y}^{(i)}) + \frac{\lambda}{N_{\text{targ}} N_T} \sum_{i,k} |D\hat{\mathbf{Y}}(\mathbf{h}^{(i)})_k - (D\mathbf{Y}^{(i)})_k|^\alpha, \quad (14)$$

where λ controls the weight of the Ehrenfest penalty term, D denotes the finite-difference approximation to the time derivative [or, equivalently, the right-hand side of the Ehrenfest equation (8) when applied to noisy data], N_{targ} is the number of signals in the training part of the dataset, and α is the penalty exponent (typically 2 or 4).

The Ehrenfest penalty term serves an additional purpose beyond physical consistency: It ensures smoothness in the temporal predictions. Without this regularization, predictions at different time points could vary independently, potentially introducing artificial noise and violating the underlying physical causality.

The hyperparameter λ may be adjusted based on the specific system under study and can be scheduled to decrease during the optimization process to balance fitting accuracy with physical constraint satisfaction.

Both the Sobolev loss (see, e.g., Ref. [48]) and Ehrenfest-constrained loss extend the standard MSE loss by incorporating derivative-based penalties, thus enforcing regularity in the learned function beyond pointwise accuracy. The Sobolev approach introduces these penalties via abstract weak derivatives, promoting smoothness in a purely mathematical sense and serving as a generic regularizer. In contrast, the Ehrenfest-constraint loss in Eq. (14) replaces these abstract derivatives with physically motivated ones, specifically those derived from commutators with the system's Hamiltonian. This substitution ensures that the enforced smoothness aligns with the underlying dynamical laws, transforming a mathematical regularizer into a physically interpretable constraint. Although both frameworks share a similar variational structure, the Ehrenfest formulation provides a principled, physics-informed specialization of the Sobolev paradigm, retaining its stability and generalization benefits while grounding every derivative penalty in fundamental physical principles.

C. Model optimization

During the model optimization step, we first divide the dataset into two, using 80% for training and the remainder for testing. Then, we train the model with the Adam optimizer with a learning rate between 0.0002 and 0.001 with several thousand epochs depending on the problem. A brief summary of training configurations for Efficient-KANs is given in Table I.

TABLE I. Summary of training configurations.

Model type	Efficient-KAN
Activation function	SiLU
Grid size	5
Spline order	3
Optimizer	Adam
Learning rate	0.0002–0.001
Train-test split	0.8–0.2
Epochs	3000–9000

The model is then validated on the test data. In the validation process, we employ the coefficients of determination, also referred to as R^2 test scores, computed for all test cases individually. This coefficient is bounded above by 1, with $R^2 \simeq 1$ corresponding to a very good fit. Generally, we plot the R^2 scores as a function of the frequency of the input signal. As an additional measure of accuracy, we calculate the percentage of test cases with R^2 scores higher than some fixed threshold R_{th}^2 , which we usually choose as 0.9, 0.95, or 0.98.

D. Chain of KANs

In addition to employing a single KAN to predict all output points, we also test the idea of using separate KANs to calculate different data points of the output series. Note that the main purpose of this approach is to enforce causality in the predictions, i.e., the requirement that the output at a given time step can only depend on the input at preceding time steps. Hence, we construct the estimator $\hat{\mathbf{Y}}^{(i)} = [\hat{Y}_1(h_1^{(i)}), \hat{Y}_2(h_1^{(i)}, h_2^{(i)}), \hat{Y}_3(h_1^{(i)}, h_2^{(i)}, h_3^{(i)}), \dots, \hat{Y}_{N_T}(h_1^{(i)}, \dots, h_{N_T}^{(i)})]$ by minimizing the MSE loss function

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_{\text{targ}} N_T} \sum_{i \in \text{Train}} \sum_{k=1}^{N_T} [Y_k^{(i)} - \hat{Y}_k(h_1^{(i)}, h_2^{(i)}, \dots, h_k^{(i)})]^2, \quad (15)$$

augmented with the Ehrenfest penalty term as defined in Eq. (14). Here, “Train” denotes the training dataset and as above N_{targ} is the number of signals in this dataset. Note that here any output data point prediction for $Y_k^{(i)}$ depends only on the input signal $h_j^{(i)}$ for $j \leq k$, effectively enforcing causality. Next, every function $\hat{Y}_k$ can be effectively parametrized as a separate KAN. For this reason, we refer to this structure as the “chain of KANs.”

V. RESULTS

The results for the various models are briefly summarized in Table II.

A. Results for the single KAN model

We begin our numerical analysis with the models with fully connected KAN architecture, as described in Appendix A 3, i.e., without any causality constraints, and with datasets 1 and 2, as specified in Sec. IV A. We discuss datasets 3 and 4 with multiple amplitudes separately in Sec. V C.

TABLE II. Performance evaluation of time series algorithms for the datasets listed in Sec. IV A, obtained from driven Ising-type systems. The results indicate the proportion of test set predictions with $R^2 > R_{th}^2$.

Dataset	N_ω	Model	Architecture	Results	R_{th}^2
1	200	Efficient-KAN	[500, 100, 500]	39/40	0.95
2	200	Efficient-KAN	[500, 100, 500]	30/40	0.95
2	600	Efficient-KAN	[500, 400, 400, 500]	115/120	0.95
2	600	Wav-KAN	[500, 240, 240, 500]	116/120	0.95
1	200	Chain of KAN	[500, 3, 1]	37/40	0.95
2	600	Chain of KAN	[500, 100, 1]	108/120	0.95
3	200	Efficient-KAN	[500, 100, 500]	95%	0.95
3	200	Chain of KAN	[500, 100, 1]	70%	0.9
4	400	Efficient-KAN	[500, 1000, 500]	85%	0.9

Both our input and output data contain 500 time steps. Hence, the corresponding KAN models generally have an architecture of the form [500, a , 500], where a characterizes the number of nodes in the intermediate KAN layer (we start with KAN models with a single intermediate layer and briefly describe deeper models at the end of this subsection); see also Appendix A 4. Note that even one intermediate layer appears to be sufficient to approach good accuracy. Note also that the number of architecture parameters significantly depends on the number of input and output time steps. Hence, the number of parameters will be significantly reduced when treating data with lower time resolution.

We first train the model on dataset 1 from Sec. IV A, which corresponds to the amplitude $A = 2.6$ and $N_\omega = 200$. This dataset is the easiest to learn, since the signals are generally not very complex. In Fig. 2, we show the R^2 scores for different test samples and also display the target signal along with the KAN predictions for four different input frequencies. We can see that the predictions are generally rather good, with the only exception being at the lowest frequency, where the prediction R^2 score is near 0.7. For all other cases, the scores are greater than 0.95. At high frequencies, we generally obtain very precise fits, which is partially related to the fact that the output signal behavior at high frequencies is more consistent from one frequency to another. Note that here the results are presented for the KAN model with rather high internal width $a = 100$. For higher widths and depths of KAN, the accuracy of the data does not change significantly. We discuss the accuracy of KAN models with lower widths (in some sense, the minimal KAN architecture) in Sec. V D.

Note that the prediction accuracy at the lowest frequencies can be enhanced by giving the model more input time series, thereby covering more input frequencies in the same range and providing higher frequency resolution. In particular, in Fig. 3 we show the results of the KAN model with the same width $a = 100$, but with a larger dataset consisting of $N_\omega = 400$ frequencies and the same amplitude $A = 2.6$. Here, we see that the accuracy of the predictions at low frequencies is significantly improved, as is generally expected in machine learning when the size of the training dataset increases. In particular, we find that the R^2 scores exceed 0.95 at all frequencies in the dataset. Note that in the previous study [32], a

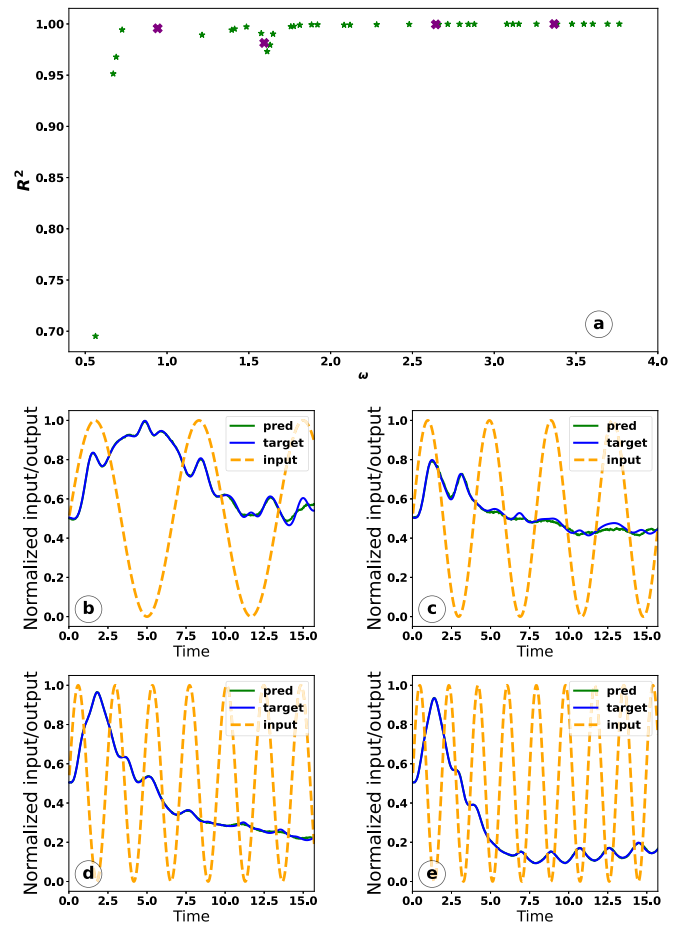


FIG. 2. Performance illustration for a KAN model with intermediate layer width 100 constructed for dataset 1 from Sec. IV A with 200 frequencies. Panel (a) shows the R^2 values vs the frequency $\omega^{(i)}$ for all i in the test dataset. Panels (b)–(e) show the optical inputs h_i (labeled as input), the corresponding output time series $Y^{(i)}$ (labeled as target), and the outputs $\hat{Y}^{(i)}$ predicted by the time series model (labeled as pred), for frequencies $\omega^{(i)}$ in $\{0.94, 1.59, 2.64, 3.37\}$, respectively, marked by purple crosses in panel (a).

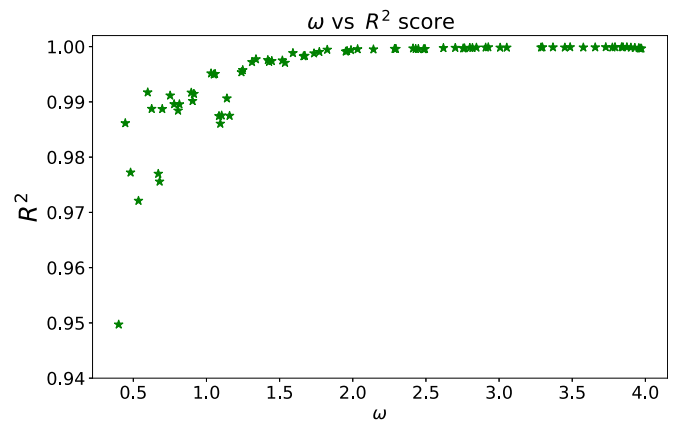


FIG. 3. Performance illustration for a KAN model with intermediate layer width 100 for dataset 1 from Sec. IV A with 400 frequencies. The figure shows the R^2 score for each driving frequency $\omega^{(i)}$ in the test set.

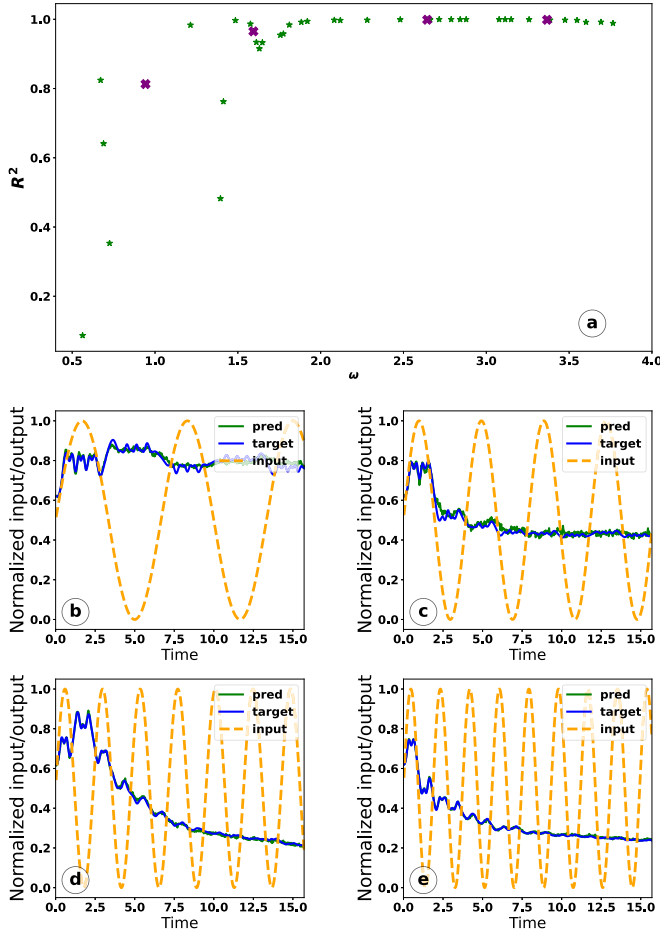


FIG. 4. Performance illustration of a KAN model with intermediate layer width 100 trained on dataset 2 from Sec. IV A with 200 frequencies. See the caption of Fig. 2 for further details.

dataset size $N_\omega = 3700$ was necessary to accurately train the models. In contrast, for KAN models, much smaller dataset sizes ($N_\omega = 200$ – 400) are sufficient to obtain accurate results.

Next, we study dataset 2 from Sec. IV A, which corresponds to driving with the amplitude $A = 10$. The signals from this dataset are generally more difficult to learn, and larger architectures are necessary. The results for this dataset (and with $N_\omega = 200$) are shown in Fig. 4. The KAN model is the same as previously used for dataset 1 with intermediate layer width $a = 100$. We see that the quality of the results is poorer than in the case of lower amplitude $A = 2.6$. As before, the predictions at higher frequencies are sufficiently accurate, while some discrepancies arise at small frequencies. Due to these errors at small frequencies, only 33 of 40 test cases have R^2 scores higher than 0.9, and 30 test cases have $R^2 > 0.95$. The quality of the results can be further improved by a finer frequency resolution and a larger KAN architecture. From the comparison of these two datasets, we conclude that the complexity of the data can significantly alter the necessary KAN architecture and frequency resolution for an accurate fit.

Finally, to validate the potential ability of KANs to capture these complex correlations, we train a deep KAN network with widths $[500, 400, 400, 500]$ on dataset 2 with $N_\omega = 600$. The results can be seen in Fig. 5. Here, 115 out of 120 test

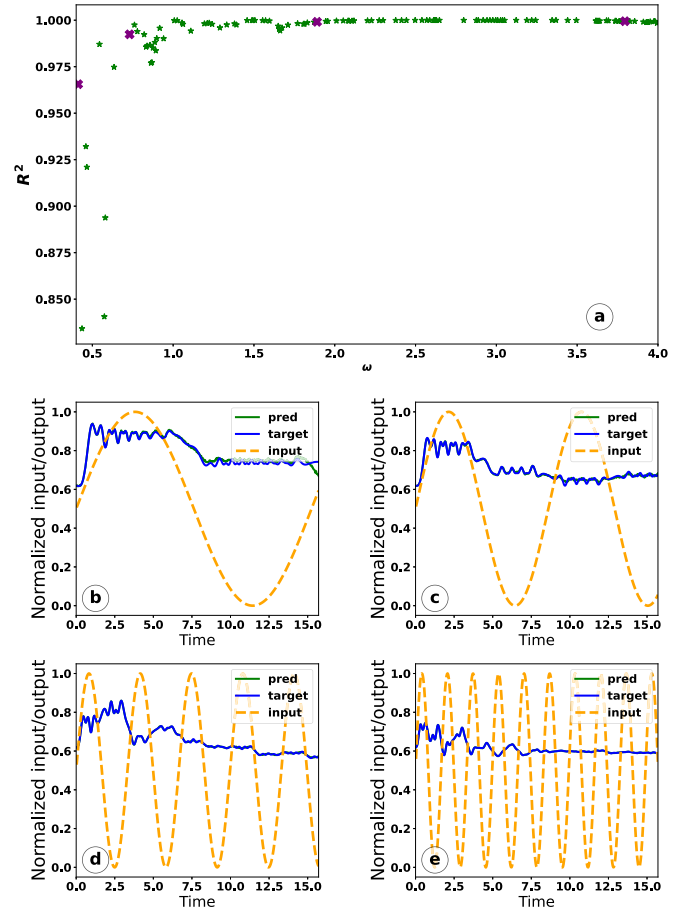


FIG. 5. Performance illustration for a two-layer KAN with intermediate width 400 trained on dataset 2 from Sec. IV A with 600 frequencies. For further details, see the caption of Fig. 2. As the test set for the 600-frequency dataset was somewhat different, in panels (b)–(e) we show frequencies $\omega^{(i)} \in \{0.41, 0.73, 1.89, 3.80\}$.

cases have $R^2 > 0.95$ and 117 cases have $R^2 > 0.9$. Note that the accuracy is much better than in the previous cases, but this is achieved at the price of larger width, depth, and dataset size. We have also tested the wavelet-based KAN (Wav-KAN) architecture with layers $[500, 240, 240, 500]$ on the same problem and obtained the results shown in Fig. 6. Here, 116 out of 120 test cases have $R^2 > 0.95$ and 118 test cases have $R^2 > 0.9$. Generally, both normal spline-based Efficient-KAN and Wav-KAN are capable of achieving very high accuracy given enough training data and sufficient width.

B. Results for the chain of KANs model

Next, we can discuss the chain of KANs model, which consists of a different KAN for every output point. This model was introduced in Sec. IV D. An example of predictions made using a chain of KANs with an intermediate layer of width 3 trained on dataset 1 is shown in Fig. 7. Note that here we plot the results for a chain of KANs both with full window size (500) and with restricted window size (200). It can be seen that some residual unphysical peaks may appear in the prediction, especially at small window sizes. Due to this effect, below we generally use the full window size of 500.

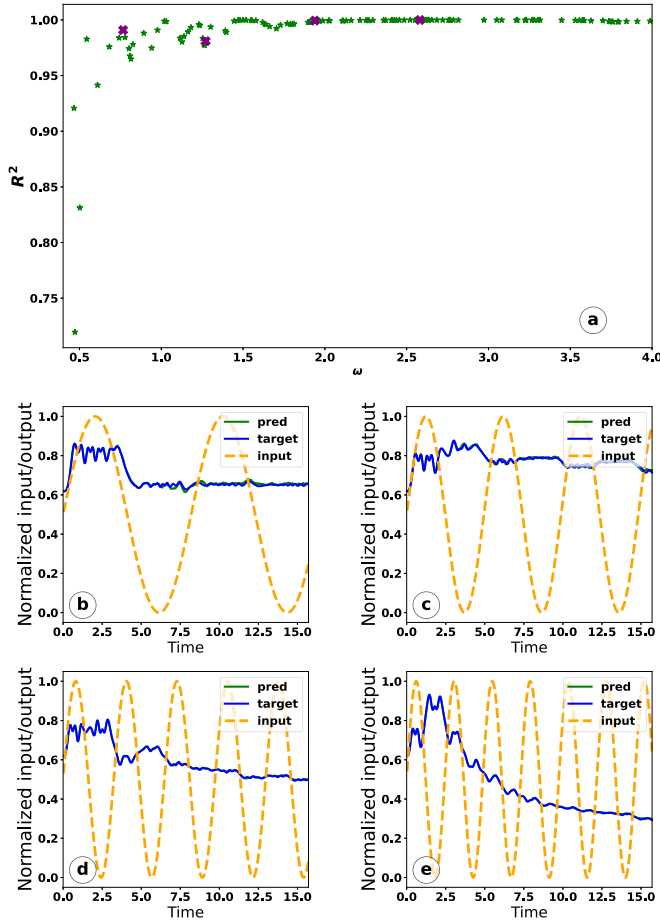


FIG. 6. Performance illustration for Wav-KAN with layers [500, 240, 240, 500] trained on dataset 2 from Sec. IV A with 600 frequencies. For further details, see the caption of Fig. 2. The frequencies shown in panels (b)–(e) correspond to $\omega^{(i)} = \{0.77, 1.27, 1.94, 2.58\}$.

We now discuss the prediction accuracy of the chain of KANs for datasets 1 and 2 from Sec. IV A. The results for dataset 1 (amplitude $A = 2.6$) are shown in Fig. 8. Of 40 test cases, 37 have scores $R^2 > 0.95$ and 38 cases have $R^2 > 0.9$. The results are generally similar to the single KAN model from the previous subsection, with the only inaccurate point corresponding to the lowest frequency.

For dataset 2, which corresponds to large-amplitude driving ($A = 10$), the results are shown in Fig. 9. Unfortunately, it was difficult to train the chain of KANs model for 200 frequencies and we were able to train it successfully only for a large dataset of 600 frequencies. 108 out of 120 test cases have $R^2 > 0.95$ and 114 cases have $R^2 > 0.9$. In addition, the necessary width of the intermediate layer was also much larger than that for dataset 1.

C. Results for the datasets with multiple amplitudes

In this subsection, the results for the datasets consisting of multiple amplitudes are presented (datasets 3 and 4 in Sec. IV A). These datasets contain a much larger number of input signals and the complexity of the output signals is also higher than that for the single-amplitude datasets.

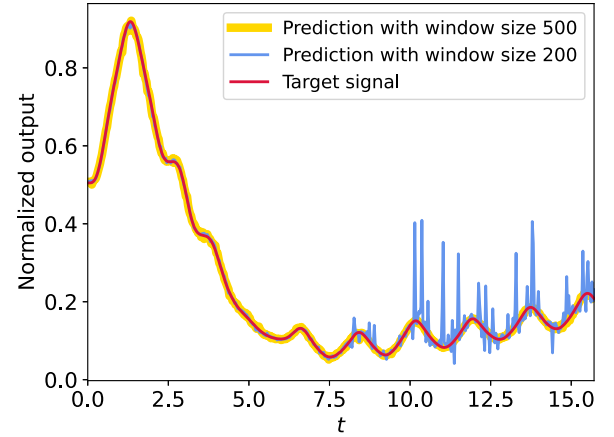


FIG. 7. Prediction and target signal for a chain of KANs model with intermediate layer width 3 for frequency $\omega = 3.55$ from dataset 1 from Sec. IV A. Fits are shown for different window sizes.

The results for the single KAN architecture with intermediate layer width 100 for dataset 3 are shown in Fig. 10. The percentage of test cases with scores above 0.95 is 95%, while the percentage of scores above 0.9 is 98%. It can be

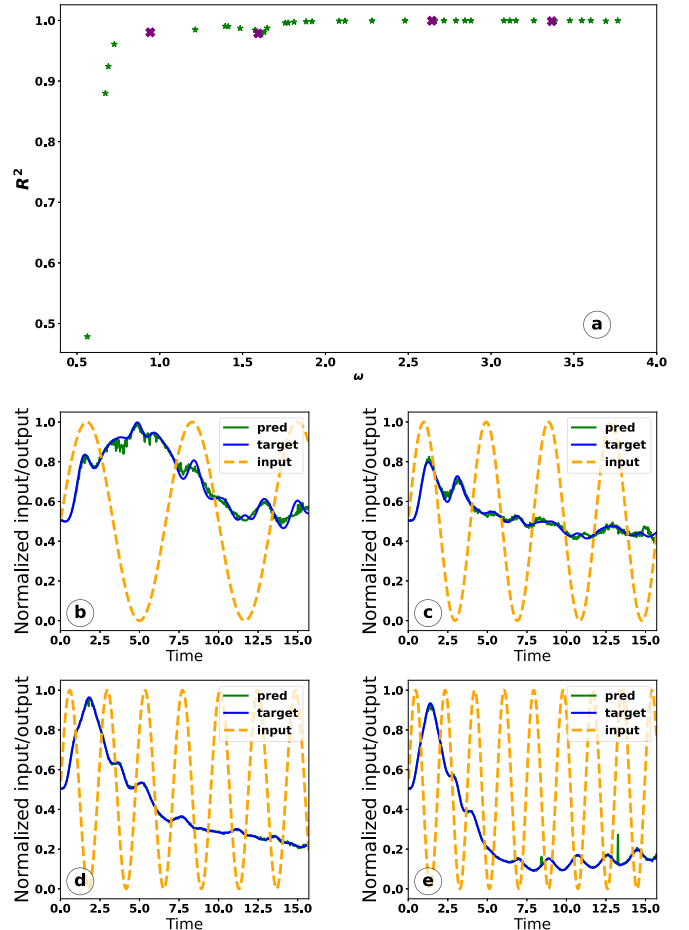


FIG. 8. Performance illustration for the chain of KANs model with intermediate layer width 3 trained on dataset 1 from Sec. IV A. For further details, see the caption of Fig. 2.

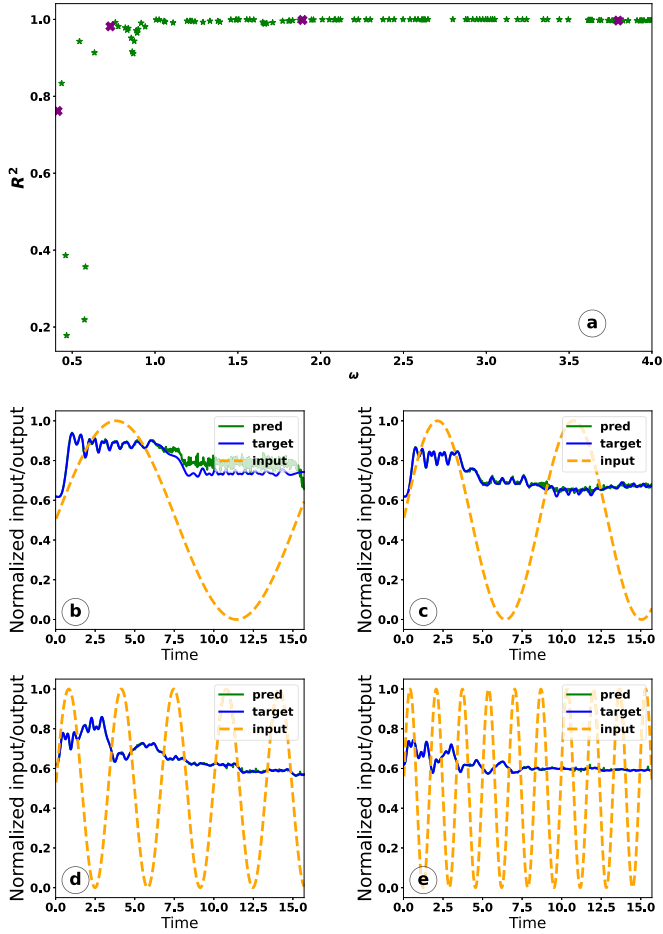


FIG. 9. Performance illustration for the chain of KANs model with intermediate layer width 100 trained on dataset 2 with 600 frequencies from Sec. IV A. See the caption of Fig. 2 for further details. Note that frequencies in panels (b)–(e) are the same as in Fig. 5.

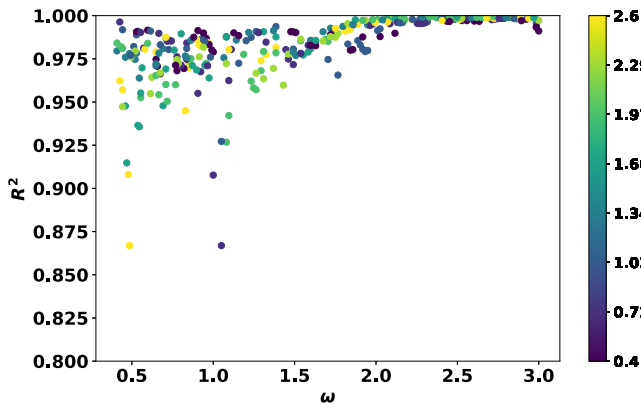


FIG. 10. Performance illustration of the single KAN model with the intermediate layer width 100 trained on dataset 3 in Sec. IV A. The figure shows dependence of R^2 scores on the driving frequency for different amplitudes for the input signals from the test set. Different amplitudes are shown in different colors, with largest amplitudes corresponding to yellow dots and darker colors for smaller amplitudes.

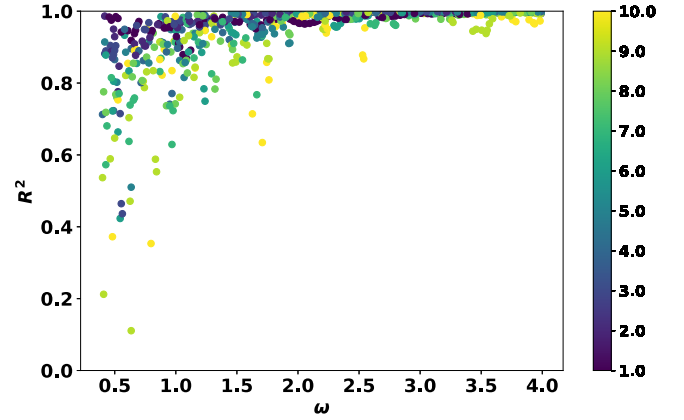


FIG. 11. Performance illustration for the KAN model with intermediate layer width 1000 trained on dataset 4 in Sec. IV A. See the caption of Fig. 10 for further details. Note that one point is excluded from the plot due to $R^2 < 0$.

seen that quite generally the R^2 scores are very high, at least for sufficiently high frequencies. The major drawback is that we still need to use quite a large intermediate layer width to achieve good accuracy.

The results for the most complex dataset 4 are shown in Fig. 11. Note that here we use $N_\omega = 400$ per amplitude, since it was difficult to train the model with only 200 frequencies. The percentage of $R^2 > 0.9$ in the test set is 85%, while the percentage of $R^2 > 0.95$ is 74%. The predictions do not demonstrate good accuracy at small frequencies. Besides, there was one point where the prediction was completely unsuccessful, with a negative R^2 score. Still, taking into account the data complexity, these results can be considered quite promising. In Fig. 12, we additionally show the results for dataset 3 and the chain of KANs architecture. The percentage of scores with $R^2 > 0.95$ is 52%, while for $R^2 > 0.9$ it is 70%.

D. Architecture stability

An additional question concerns the choice of the optimal KAN architecture. The optimal architecture is the one that has

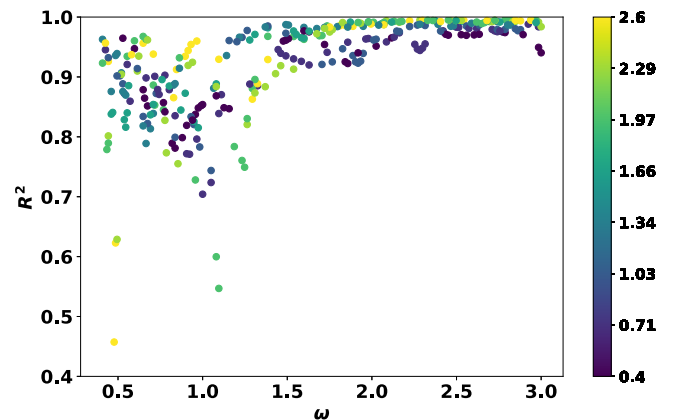


FIG. 12. Performance illustration for the chain of KANs model with $a = 100$ trained on dataset 3 in Sec. IV A. See the caption of Fig. 10 for further details.

TABLE III. KAN with $a = 10$, $A = 2.6$, and $N_\omega = 200$.

Model No.	$R^2 > 0.98$ (%)	$R^2 > 0.95$ (%)	$R^2 > 0.9$ (%)
1	67.5	77.5	80.5
2	52.5	65	77.5
3	50	70	80
4	57.5	70	77.5
5	60	75	85
6	57.7	75	90
7	62.5	75	87.5
8	80	82.5	95
9	70	82.5	82.5
10	55	67.5	77.5

a minimal number of layers as well as minimal widths for the intermediate layers, while still providing sufficiently accurate predictions. Obviously, there are potentially many different combinations of the numbers and widths of layers, and it is difficult to check them all. In this subsection, we focus on the architectures with just one intermediate layer and search for the minimal width for this layer.

We also note that the results for the training may vary depending on the partitioning of the dataset into the training and test parts. For this reason, we train our models for various different partitions (generally for ten different partitions) and summarize the results in the form of tables, where we show the percentage of test results with R^2 scores exceeding the threshold values of 0.98, 0.95, and 0.9.

First, we should note that the minimal stable architecture depends on the dataset. For dataset 1, which has amplitude $A = 2.6$, a relatively small intermediate width $a = 10$ or $a = 20$ is already sufficient to reach good accuracy. From a comparison of Tables III and IV, one can see that the accuracy gradually increases with the width of the intermediate layer. Note that the accuracy also depends on the number of frequencies in the dataset. In particular, in Table V we show the results with width $a = 10$ trained on dataset 1 with $N_\omega = 400$, rather than $N_\omega = 200$ as in Table III. In this case, width $a = 10$ gives the same accuracy level as we obtained previously with $a = 20$ for the dataset with $N_\omega = 200$ (Table IV). Still, in both cases, rather small intermediate layer widths were sufficient to reach good accuracy. This is in contrast to dataset 2 (with $A = 10$), where the accuracy increases very slowly with the

TABLE V. KAN with $a = 10$, $A = 2.6$, and $N_\omega = 400$.

Model No.	$R^2 > 0.98$ (%)	$R^2 > 0.95$ (%)	$R^2 > 0.9$ (%)
1	50	76	91
2	47	94	97
3	65	86	91
4	65	89	94
5	53	73	96
6	68	88	90
7	41	74	94
8	81	95	99
9	63	90	95
10	56	80	91

intermediate layer width. We show an example of results for dataset 2 with intermediate width $a = 100$ in Table VI. Even for such a relatively large width, on average only about 60%–70% of our results have $R^2 > 0.9$.

We can conclude that the optimal architecture for KAN depends rather strongly on the properties of the signal. For complex signals, the network must be sufficiently wide (or deep).

VI. CONCLUSION

We have developed and validated KAN-Ehrenfest time series analysis, a physics-informed machine learning framework that combines KANs with Ehrenfest's theorems to predict quantum system dynamics while preserving the underlying physics of the model. The resulting KAN-ETS approach addresses a critical limitation in existing time series methods in physics: the tendency to violate physical constraints in favor of empirical accuracy. Our method embeds Ehrenfest's relations directly into the loss function, additionally regularizing it and resulting in smoother predictions. The code examples are available in Ref. [49].

Our method accurately reconstructs magnetization patterns in Ising chains; over 95% of test cases achieve an R^2 above 0.95 using inputs that sample just 200 equispaced frequencies. KANs match or outperform traditional models such as TCNs [32] while requiring 95% less data and far fewer computational resources. This proves that KANs are both dataefficient and faster to train.

TABLE IV. KAN with $a = 20$, $A = 2.6$, and $N_\omega = 200$.

Model No.	$R^2 > 0.98$ (%)	$R^2 > 0.95$ (%)	$R^2 > 0.9$ (%)
1	73	83	90
2	58	75	93
3	53	75	90
4	70	85	93
5	80	90	98
6	58	83	90
7	70	88	93
8	88	95	100
9	65	85	90
10	63	68	78

TABLE VI. KAN with $a = 100$, $A = 10$, and $N_\omega = 200$.

Model No.	$R^2 > 0.98$ (%)	$R^2 > 0.95$ (%)	$R^2 > 0.9$ (%)
1	58	63	75
2	48	55	68
3	33	43	58
4	45	53	75
5	55	65	83
6	38	50	63
7	38	50	63
8	60	65	70
9	58	60	73
10	48	50	70

Further, another significant advancement of our framework lies in its treatment of the multi-amplitude problem through prompt-conditioned mapping. Rather than requiring separate models for different drive amplitudes (a computationally expensive approach that limits practical applications), the KAN-ETS framework learns a continuous response surface across all amplitude regimes within a single model, seamlessly interpolating from linear response at small drives to highly nonlinear behavior at extreme amplitudes. This mirrors how large language models handle diverse prompts within a unified architecture, adapting their responses based on input context, rather than requiring separate models for different query types. The approach scales naturally with the available data through curriculum learning that progresses from moderate to extreme amplitudes, creating a robust model that faithfully reproduces material behavior across the entire amplitude spectrum and providing a clear path from controlled laboratory studies to large-scale applications without architectural modifications.

We further enriched our modeling toolkit with the chain of KANs, enforcing strict temporal causality along with the Ehrenfest constraint. Unlike traditional models that predict entire output sequences at once, the chain of KANs decomposes the prediction into a sequence of causally ordered steps. Each output point is predicted using only current and past input values, ensuring that future information does not leak into the prediction, preserving causality. By assigning a separate KAN to each time step, the model allows independent training and flexible deployment of individual components, which is particularly useful for online and real-time applications. Despite this decomposition, the chain of KANs maintains strong predictive performance and offers improved interpretability, making it especially valuable for physics-informed forecasting tasks such as quantum control, spectroscopy, and modeling nonlinear optical systems.

ACKNOWLEDGMENT

This work was supported by Army Research Office (ARO) (Grant No. W911NF-23-1-0288; program manager Dr. James Joseph).

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARO, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

DATA AVAILABILITY

The data that support the findings of this article are openly available [49].

APPENDIX A: KOLMOGOROV-ARNOLD THEOREM AND KOLMOGOROV-ARNOLD NETWORK

1. Quick introduction

The development of DNNs, also referred to as feed-forward neural networks or multilayer perceptrons (MLPs), and convolutional neural networks (CNNs) has played a

foundational role in the evolution of modern machine learning architectures. While DNNs serve as a general-purpose framework applicable to a wide variety of learning tasks, CNNs are architecturally specialized for data with spatial structure, particularly images.

In traditional MLPs and CNNs, activation functions are essential nonlinear components applied after each linear transformation or convolution. Their role is to introduce nonlinearity into the model, allowing the network to approximate complex, nonlinear mappings rather than collapsing into a series of linear operations. Common activation functions such as ReLU, Tanh, or GELU are fixed and applied uniformly across the network. Without them, even a deep stack of layers would behave like a single linear transformation. In contrast, KANs depart from this paradigm by embedding nonlinearity directly in the connections between neurons. Instead of static scalar weights followed by a global activation, each connection in a KAN is a learnable univariate function, often implemented as a B -spline that maps input values in a data-driven, nonlinear fashion (see Fig. 1).

2. Kolmogorov-Arnold representation theorem in KANs

KAN is a neural architecture that leverages the functional decomposition guaranteed by the Kolmogorov-Arnold representation theorem [46]. The theorem is stated below.

Let $n \in \mathbb{N}$ and $I = [0, 1]$. For any continuous function

$$f \in C(I^n, \mathbb{R}),$$

there exist continuous “inner” functions

$$\phi_{i,j} : I \rightarrow \mathbb{R}, \quad i = 0, 1, \dots, 2n, \quad j = 1, 2, \dots, n,$$

and continuous “outer” functions

$$\psi_i : \mathbb{R} \rightarrow \mathbb{R}, \quad i = 0, 1, \dots, 2n,$$

such that for every $\mathbf{x} = (x_1, \dots, x_n) \in I^n$, one has

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2n} \psi_i \left(\sum_{j=1}^n \phi_{i,j}(x_j) \right). \quad (\text{A1})$$

In 1900, David Hilbert famously asked whether any continuous function of three or more variables could be built from just continuous functions of two variables [50]. Then, in a stunning breakthrough in 1957, Andrey Kolmogorov proved that any continuous function of several variables can be exactly expressed as a finite sum of single-variable functions composed with addition, and soon thereafter Vladimir Arnold refined and streamlined that construction [46]. This surprise solution not only laid Hilbert’s question to rest, but also inspired today’s Kolmogorov-Arnold networks, which use trainable univariate subnetworks to capture complex, high-dimensional behaviors.

3. KANs as networks

In this Appendix, we will present the underlying mathematics behind KAN and how it differs from traditional DNN. We will first present how each neuron functions in DNN and then move on to how, in a KAN, each connection itself implements a learnable univariate mapping (see Fig. 13). In traditional DNN, the input data are initially received by the

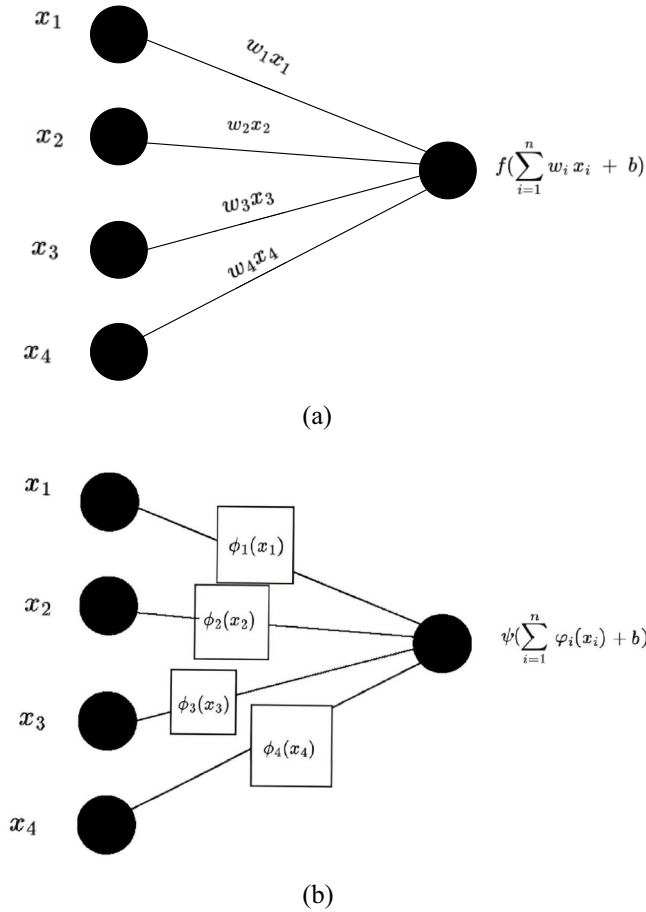


FIG. 13. Comparison of (a) a traditional deep neural network and (b) a Kolmogorov-Arnold network.

input neurons, which then undergo a preactivation transformation where the inputs are linearly combined with learned weights and adjusted by bias parameters. Subsequently, this weighted sum passes through a neuron and is subjected to a nonlinear activation function f . Formally, let $\{x_i\}_{i=1}^n$ denote the inputs to a given neuron and $\{w_i\}_{i=1}^n$, b the corresponding weights and bias. The neuron first computes the *preactivation*

$$z = \sum_{i=1}^n w_i x_i + b. \quad (\text{A2})$$

To introduce nonlinearity, which is essential for the network's ability to approximate complex functions, the preactivation value is then passed through an elementwise activation function f , yielding the *postactivation*

$$a = f(z). \quad (\text{A3})$$

In KAN, each input connection applies a learnable univariate function $\varphi_i : \mathbb{R} \rightarrow \mathbb{R}$, replacing scalar weight parameters and embedding nonlinearity at every edge. Formally, let $\{x_i\}_{i=1}^n$ denote the inputs to a neuron, $\{\varphi_i\}_{i=1}^n$ the corresponding edge functions, and b the bias term. The neuron then computes the *preactivation*

$$z = \sum_{i=1}^n \varphi_i(x_i) + b, \quad (\text{A4})$$

without requiring a separate activation step, since each φ_i is itself nonlinear by design. Optionally, to fully mirror the classical superposition form, one may apply an *outer* univariate function $\psi : \mathbb{R} \rightarrow \mathbb{R}$, which results in

$$a = \psi(z). \quad (\text{A5})$$

By composing L such layers, denoting the ℓ th layer's function matrix by $\phi^{(\ell)}$, one immediately recovers the general deep KAN mapping

$$\text{KAN}(x) = (\phi^{(L-1)} \circ \phi^{(L-2)} \circ \dots \circ \phi^{(0)})(x), \quad (\text{A6})$$

which naturally extends the single-neuron formulation to architectures of arbitrary depth and width.

We conclude this Appendix with a brief overview of the different types of KAN that have been proposed. The KAN architecture has rapidly evolved into several specialized variants such as FastKAN, Efficient-KAN, Wav-KAN, T-KAN, etc., each tailored to address specific computational challenges or to suit distinct application domains [51]. In this article, we tend to use Efficient-KAN [52] and Wav-KAN [53]. Efficient-KAN optimizes the original KAN architecture for memory and computational efficiency, making it more practicable for large-scale or resource-constrained settings. On the other hand, Wav-KAN enhances the representational power and interpretability by integrating multiresolution wavelet activation functions, well suited for data with rich frequency structure.

4. KAN architecture notation

In this article, we represent a KAN model by the four-tuple

$$[I, a, b, O]$$

(the same notation being used in our implementation code). Here, I denotes the dimensionality of the input layer (i.e., the number of input features), a and b denote the widths of the first and second hidden layers, respectively, and O denotes the dimensionality of the output layer (i.e., the number of output features).

For the specific configuration

$$[500, 100, 100, 500],$$

we have $I = 500$, meaning each data example is a univariate time series of length 500 (treated as a 500-dimensional input vector). The hidden layers each contain 100 nodes ($a = b = 100$), providing a compact nonlinear embedding, and $O = 500$, so the network reconstructs or forecasts a 500-step output sequence.

5. Introduction to Efficient-KAN and Wav-KAN

In the previous section, we introduced a KAN layer as a map where each connection (edge) from input component x_j to output unit i is not a scalar weight but a learnable univariate function. Efficient-KAN follows exactly this modeling principle. Its contribution is mainly computational. The idea is to rewrite the same spline-parametrized edge functions into a form that can be evaluated using a small number of graphics processing unit (GPU)-friendly matrix multiplications, without constructing large intermediate edge activation tensors.

In order to understand the Efficient-KAN and the Wav-KAN, we must first delve into the details of the general formulation of KAN [33].

a. Details of KAN

Let us understand KAN more deeply. Let us begin by generalizing Eq. (A4). In the previous Appendix, Eq. (A4) was for a single-output neuron. This formulation can be interpreted as follows:

- (1) The architecture consists of a single neuron producing one output z .
- (2) The neuron receives inputs $x_1, \dots, x_n$.
- (3) Each input x_i is transformed through its own learnable univariate function $\phi_i(\cdot)$.
- (4) The neuron computes the sum of all edge functions and adds a bias term b .

The *output* of the neuron is then obtained by applying an outer activation function $\psi : \mathbb{R} \rightarrow \mathbb{R}$ to the preactivation as in Eq. (A5).

We now extend this formulation to accommodate a complete KAN layer with multiple outputs. Consider the ℓ th layer characterized by input dimensionality $n_\ell \in \mathbb{N}$ and output dimensionality $m_\ell \in \mathbb{N}$. Let the input activation vector be denoted as

$$\mathbf{x}^{(\ell)} = (x_1^{(\ell)}, \dots, x_{n_\ell}^{(\ell)})^\top \in \mathbb{R}^{n_\ell}, \quad (\text{A7})$$

which the layer transforms into the preactivation vector

$$\mathbf{z}^{(\ell+1)} = (z_1^{(\ell+1)}, \dots, z_{m_\ell}^{(\ell+1)})^\top \in \mathbb{R}^{m_\ell}. \quad (\text{A8})$$

For each ordered pair of indices (i, j) where

$$i \in \{1, \dots, m_\ell\}, \quad j \in \{1, \dots, n_\ell\}, \quad (\text{A9})$$

we associate a distinct learnable univariate edge function

$$\phi_{ij}^{(\ell)} : \mathbb{R} \rightarrow \mathbb{R}, \quad (\text{A10})$$

which governs the contribution of input component $x_j^{(\ell)}$ to preactivation $z_i^{(\ell+1)}$ in the next layer. The natural multioutput generalization of Eq. (A4) is therefore

$$z_i^{(\ell+1)} = \sum_{j=1}^{n_\ell} \phi_{ij}^{(\ell)}(x_j^{(\ell)}) + b_i^{(\ell)}, \quad i = 1, \dots, m_\ell, \quad (\text{A11})$$

where $b_i^{(\ell)}$ is the bias associated with output i of the ℓ th layer. After computing the preactivation $z_i^{(\ell+1)}$ using Eq. (A11), the postactivation (output activation) of layer $\ell + 1$ is obtained by applying an outer nonlinearity $\psi^{(\ell)} : \mathbb{R} \rightarrow \mathbb{R}$ componentwise:

$$a_i^{(\ell+1)} = \psi^{(\ell)}(z_i^{(\ell+1)}), \quad i = 1, \dots, m_\ell. \quad (\text{A12})$$

Further, in the original KAN formulation [33], each edge function $\phi_{ij}^{(\ell)}$ is a learnable univariate mapping. A convenient and expressive choice is to represent these edge functions using a spline expansion. In particular, we decompose each $\phi_{ij}^{(\ell)}$ into a fixed base nonlinearity and a flexible spline term:

$$\phi_{ij}^{(\ell)}(u) = w_{ij}^{(\ell)} \sigma(u) + s_{ij}^{(\ell)}(u), \quad (\text{A13})$$

where $\sigma(\cdot)$ is a fixed base activation function (e.g., SiLU) and $s_{ij}^{(\ell)}(\cdot)$ is a learnable spline component. The spline component

is expanded in a set of B -spline basis functions $\{B_m^{(\ell)}\}_{m=1}^{M_\ell}$ defined on a grid in layer ℓ :

$$s_{ij}^{(\ell)}(u) = \sum_{m=1}^{M_\ell} c_{ijm}^{(\ell)} B_m^{(\ell)}(u), \quad (\text{A14})$$

where $c_{ijm}^{(\ell)}$ are learnable spline coefficients. In practice, the number of basis functions M_ℓ is typically chosen as the sum of the grid size and the spline order, which matches our implementation. Substituting Eqs. (A13) and (A14) into Eq. (A11) yields the fully expanded KAN layer preactivation:

$$z_i^{(\ell+1)} = \sum_{j=1}^{n_\ell} w_{ij}^{(\ell)} \sigma(x_j^{(\ell)}) + \sum_{j=1}^{n_\ell} \sum_{m=1}^{M_\ell} c_{ijm}^{(\ell)} B_m^{(\ell)}(x_j^{(\ell)}) + b_i^{(\ell)}. \quad (\text{A15})$$

Now, we are in a position to understand the difference between original KAN implementation [33] and Efficient-KAN [52].

b. Efficient-KAN: An efficient implementation of KAN

The original KAN implementation suffers from significant computational and memory overhead. For each sample in a minibatch and each input coordinate, the network must evaluate all edge functions explicitly, producing a large intermediate tensor of size $B \times m_\ell \times n_\ell$ for a minibatch of size B . This inefficiency grows as the network scales in width, depth, or batch size.

Efficient-KAN addresses this bottleneck by exploiting the linearity of the spline term in its coefficients $c_{ijm}^{(\ell)}$ once the B -spline basis values are known. Rather than materializing the full edge activation tensor

$$S_{b,i,j}^{(\ell)} = \phi_{ij}^{(\ell)}(x_{b,j}^{(\ell)}) \in \mathbb{R}^{B \times m_\ell \times n_\ell}, \quad (\text{A16})$$

the method proceeds as follows. First, it evaluates B -spline bases for each input coordinate:

$$\mathcal{B}^{(\ell)}(X^{(\ell)})_{b,j,m} = B_m^{(\ell)}(x_{b,j}^{(\ell)}) \in \mathbb{R}^{B \times n_\ell \times M_\ell}. \quad (\text{A17})$$

Next, it flattens (j, m) into a single basis feature index:

$$\Phi^{(\ell)}(X^{(\ell)})_{b,(j-1)M_\ell+m} = \mathcal{B}^{(\ell)}(X^{(\ell)})_{b,j,m} \in \mathbb{R}^{B \times (n_\ell M_\ell)}. \quad (\text{A18})$$

The spline coefficients are then reshaped into a matrix:

$$W_{\text{spline}}^{(\ell)}[i, (j-1)M_\ell+m] = c_{ijm}^{(\ell)} \in \mathbb{R}^{m_\ell \times (n_\ell M_\ell)}. \quad (\text{A19})$$

Finally, the entire spline contribution is computed via a single matrix multiplication:

$$X_{\text{spline}}^{(\ell+1)} = \Phi^{(\ell)}(X^{(\ell)})(W_{\text{spline}}^{(\ell)})^\top \in \mathbb{R}^{B \times m_\ell}, \quad (\text{A20})$$

replacing the summation over a $B \times m_\ell \times n_\ell$ tensor with GPU-friendly matrix operations.

c. Wav-KAN: Implementing wavelets in KAN

Wav-KAN [53] is a variant of KAN that fundamentally changes the function class used on each edge by replacing spline bases with wavelets, aiming to obtain localized, multiscale expressivity without relying on knot grids and refinement procedures.

The wavelet construction begins with a mother wavelet $\psi : \mathbb{R} \rightarrow \mathbb{R}$, from which one forms a family of shifted and scaled

wavelets

$$\psi_{s,\tau}(u) = \frac{1}{\sqrt{s}} \psi\left(\frac{u-\tau}{s}\right), \quad s > 0, \tau \in \mathbb{R}. \quad (\text{A21})$$

Here, the parameter s controls resolution (large s yields coarse structure, while small s yields sharp, localized detail) and τ specifies the location of the wavelet on the input axis. This is the same principle that underlies the continuous wavelet transform, where a signal is analyzed by its correlations with wavelets across different (s, τ) :

$$C(s, \tau) = \int_{-\infty}^{+\infty} g(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) dt. \quad (\text{A22})$$

Wav-KAN leverages this multiscale viewpoint at the level of individual KAN edges. Instead of approximating $\phi_{ij}^{(\ell)}$ with a spline expansion tied to a knot grid, the edge function is modeled using wavelet atoms whose shift and scale are learned directly from data.

A standard Wav-KAN parametrization assigns to each edge (i, j) a learnable amplitude, translation, and scale:

$$\phi_{ij}^{(\ell)}(u) = w_{ij}^{(\ell)} \psi\left(\frac{u - \tau_{ij}^{(\ell)}}{s_{ij}^{(\ell)}}\right), \quad s_{ij}^{(\ell)} > 0, \quad (\text{A23})$$

where $w_{ij}^{(\ell)}$ is the amplitude (wavelet coefficient), $\tau_{ij}^{(\ell)}$ shifts the wavelet along the input coordinate, and $s_{ij}^{(\ell)}$ sets the effective bandwidth of the localized feature. Substituting Eq. (A23) into the general KAN layer definition yields the Wav-KAN preactivation

$$z_i^{(\ell+1)} = \sum_{j=1}^{n_\ell} w_{ij}^{(\ell)} \psi\left(\frac{x_j^{(\ell)} - \tau_{ij}^{(\ell)}}{s_{ij}^{(\ell)}}\right) + b_i^{(\ell)}, \quad i = 1, \dots, m_\ell, \quad (\text{A24})$$

followed by the usual outer nonlinearity

$$a_i^{(\ell+1)} = \psi^{(\ell)}(z_i^{(\ell+1)}), \quad i = 1, \dots, m_\ell. \quad (\text{A25})$$

In summary, Efficient-KAN accelerates spline-based KANs by reorganizing computation into matrix multiplications, whereas Wav-KAN modifies the edge-function parametrization by replacing spline bases with wavelets that learn location and scale, thereby providing an explicit multiresolution inductive bias.

APPENDIX B: FAILURE MODES OF TEMPORAL CONVOLUTION NETWORKS

In Ref. [32], the input $\rightarrow$ output mapping for both the transverse and nonintegrable Ising models is formulated as follows. We drive a quantum spin chain of length N with a monochromatic magnetic field of fixed amplitude A ,

$$h^{(i)}(t) = A \sin(\omega^{(i)} t) \quad \text{with} \quad i = 1, \dots, n, \quad (\text{B1})$$

where $n = 3700$. The corresponding optical output is the total magnetization

$$Y^{(i)}(t) = \sum_{k=1}^N \langle \sigma_x^{(k)}(t) \rangle. \quad (\text{B2})$$

Then, discretizing time into $T = 512$ uniform steps of size δt transforms each continuous trajectory into a pair of

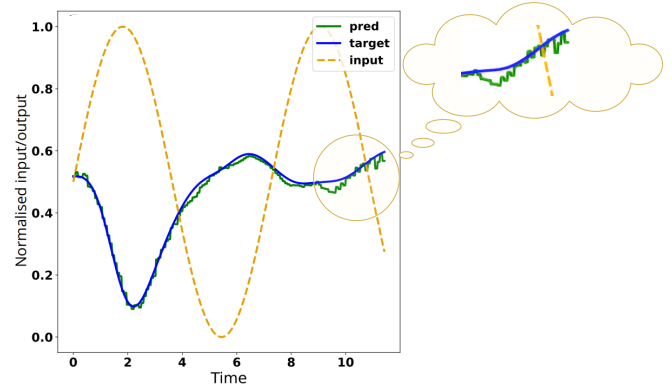


FIG. 14. Predicted output signal using a TCN model. The TCN output (green) compared with the target signal (blue), for a given input (orange), exhibits step-size residual oscillations, highlighted in the inset, which remain unresolved even after training on large data samples.

T -dimensional vectors:

$$\begin{aligned} \text{Input: } h_k^{(i)} &= A \sin(\omega^{(i)} k \delta t), \\ \text{Output: } Y_k^{(i)} &= Y^{(i)}(k \delta t), \end{aligned} \quad (\text{B3})$$

where $k = 1, 2, \dots, T$. Thus, the full dataset can be written as

$$\{\mathbf{h}^{(i)} \rightarrow \mathbf{Y}^{(i)}\}_{i=1}^n, \quad \mathbf{h}^{(i)}, \mathbf{Y}^{(i)} \in \mathbb{R}^T, \quad (\text{B4})$$

It was observed that when the number of input frequencies falls below this threshold, the TCNs, regardless of architectural depth or filter size, tend to become trapped in poor local minima during training, resulting in plateau loss and consequently inadequate input-output fitting. Moreover, owing to the inherently large receptive fields and repeated convolutional operations required to process such high-dimensional data, the training time for TCNs scales unfavorably. In contrast, we will see that KANs achieve comparable or superior fit performance with as few as 200 frequency samples, while incurring substantially lower computational overhead. In particular, where a TCN baseline required approximately 3700 data samples to reach a given level of precision, KANs reached that regime with only 200, underscoring both their data efficiency and reduced training cost. Another unresolved issue in Ref. [32] lies with tiny step-size residual oscillations in the predicted output signal (see Fig. 14). Although the original article does not investigate the source of these oscillations/fluctuations, in the present work we demonstrate that they vanish when employing KANs.

APPENDIX C: COMPUTATION OF THE RIGHT-HAND SIDES OF EHRENFEST EQUATIONS

Our loss definition uses derivative regularization, which includes the signal derivative obtained with Ehrenfest equations. Note that in a real experiment this derivative should be computed with direct measurements of the operators appearing on the right-hand sides of the Ehrenfest equations.

Our own data collection was based on the nearly exact iTEBD simulation. Due to this near-exactness, we have two

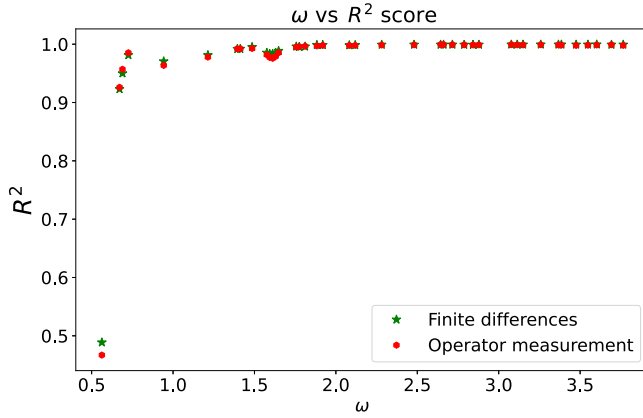


FIG. 15. Comparison of the training results with Ehrenfest terms computed either with finite differences or with operator measurements. The KAN model was [500, 100, 500] and it was trained on dataset 1. Except for the different methods of Ehrenfest term computation, all other training details were the same. Note that the results are slightly different from the results shown in Fig. 2, as we used slightly different training parameters.

ways to obtain the terms on the right-hand side of the Ehrenfest equations: We could either measure the operators from the right-hand side directly in the iTEBD algorithm or simply compute the numerical derivative of the signal. Due to the near-exactness of the simulation, both approaches lead to nearly identical results. In particular, the MSE loss between the operators from the right-hand sides of Ehrenfest equations computed in two different ways was of the order 10^{-6} , with average relative difference below 1%. This is why we used the derivatives obtained from finite differences of the target signal in our training. As real experimental data are inevitably noisy, and the noise influences the finite-difference computation of derivatives much more than it influences operator measurements, it is better to use Ehrenfest equations to access derivatives in an experimental setting.

To further emphasize that the usage of finite differences instead of operator measurements does not significantly influence the results (in the case of training data obtained from nearly exact numerical simulation), we add Fig. 15 with a

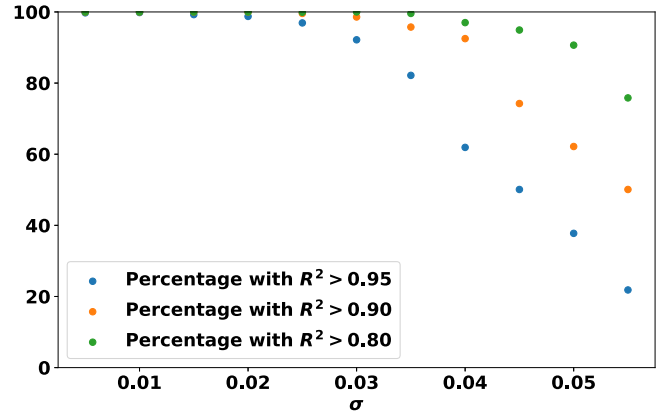


FIG. 16. Percentage of test predictions with R^2 scores exceeding 0.95, 0.9, and 0.8 as a function of the additive Gaussian noise standard deviation σ . The KAN model was trained on dataset 1, and each data point represents an average over 15 independent noise realizations.

comparison of training results for dataset 1 (with 200 frequencies) and with Ehrenfest terms computed either with finite differences or with operator measurements inside of iTEBD. We used the same model and the same train/test data splitting for this comparison.

APPENDIX D: KAN TRAINING WITH NOISY DATA

In real-world experiments, measured data are typically noisy due to shot noise and decoherence. To assess the robustness of our method under such conditions, we train KAN models in the presence of synthetic additive noise. Specifically, we use dataset 1 with the KAN architecture [500, 100, 500], adding Gaussian noise with standard deviation σ to both the normalized training signal and the normalized derivative obtained from the Ehrenfest equations.

We trained KAN models on the resulting noisy data for various values of σ . Predictions were validated on noiseless test data and averaged over 15 independent noise realizations for each σ . Figure 16 shows the percentage of test predictions achieving R^2 scores above 0.95, 0.9, and 0.8 as a function of σ . The results demonstrate that KAN predictions remain stable even in the presence of significant noise.

- [1] D. I. Bondar, R. Cabrera, R. R. Lompay, M. Y. Ivanov, and H. A. Rabitz, Operational dynamic modeling transcending quantum and classical mechanics, *Phys. Rev. Lett.* **109**, 190403 (2012).
- [2] P. Ehrenfest, Bemerkung über die angenäherte Gültigkeit der klassischen Mechanik innerhalb der Quantenmechanik, *Z. Phys.* **45**, 455 (1927).
- [3] L. E. Ballentine, Y. Yang, and J. P. Zibin, Inadequacy of Ehrenfest's theorem to characterize the classical regime, *Phys. Rev. A* **50**, 2854 (1994).
- [4] A. G. Campos, D. I. Bondar, R. Cabrera, and H. A. Rabitz, How to make distinct dynamical systems appear spectrally identical, *Phys. Rev. Lett.* **118**, 083201 (2017).
- [5] G. McCaul, C. Orthodoxou, K. Jacobs, G. H. Booth, and D. I. Bondar, Driven imposters: Controlling expectations in many-body systems, *Phys. Rev. Lett.* **124**, 183201 (2020).
- [6] R. Cabrera, A. G. Campos, D. I. Bondar, and H. A. Rabitz, Dirac open-quantum-system dynamics: Formulations and simulations, *Phys. Rev. A* **94**, 052111 (2016).
- [7] R. Cabrera, A. G. Campos, H. A. Rabitz, and D. I. Bondar, Operational dynamical modeling of spin 1/2 relativistic particles: The Dirac equation and its classical limit, *Eur. Phys. J. Spec. Top.* **227**, 2195 (2019).
- [8] D. I. Bondar, R. Cabrera, A. Campos, S. Mukamel, and H. A. Rabitz, Wigner–Lindblad equations for quantum friction, *J. Phys. Chem. Lett.* **7**, 1632 (2016).

- [9] D. V. Zhdanov, D. I. Bondar, and T. Seideman, No thermalization without correlations, *Phys. Rev. Lett.* **119**, 170402 (2017).
- [10] S. L. Vuglar, D. V. Zhdanov, R. Cabrera, T. Seideman, C. Jarzynski, and D. I. Bondar, Nonconservative forces via quantum reservoir engineering, *Phys. Rev. Lett.* **120**, 230404 (2018).
- [11] A. J. Schimmoller, G. McCaul, H. Abele, and D. I. Bondar, Decoherence-free entropic gravity: Model and experimental tests, *Phys. Rev. Res.* **3**, 033065 (2021).
- [12] E. J. Sung, A. G. Campos, H. Abele, and D. I. Bondar, Decoherence-free entropic gravity for a Dirac fermion, *Phys. Rev. D* **108**, 104036 (2023).
- [13] F. Krausz and M. Ivanov, Attosecond physics, *Rev. Mod. Phys.* **81**, 163 (2009).
- [14] S. Ghimire, A. D. DiChiara, E. Sistrunk, P. Agostini, L. F. DiMauro, and D. A. Reis, Observation of high-order harmonic generation in a bulk crystal, *Nat. Phys.* **7**, 138 (2011).
- [15] T. T. Luu, M. Garg, S. Y. Kruchinin, A. Moulet, M. T. Hassan, and E. Goulielmakis, Extreme ultraviolet high-harmonic spectroscopy of solids, *Nature (London)* **521**, 498 (2015).
- [16] G. McCaul, K. Jacobs, and D. I. Bondar, Towards single atom computing via high harmonic generation, *Eur. Phys. J. Plus* **138**, 123 (2023).
- [17] A. B. Magann, G. McCaul, H. A. Rabitz, and D. I. Bondar, Sequential optical response suppression for chemical mixture characterization, *Quantum* **6**, 626 (2022).
- [18] S. Takayoshi, Y. Murakami, and P. Werner, High-harmonic generation in quantum spin systems, *Phys. Rev. B* **99**, 184303 (2019).
- [19] T. N. Ikeda and M. Sato, High-harmonic generation by electric polarization, spin current, and magnetization, *Phys. Rev. B* **100**, 214424 (2019).
- [20] R. K. Malla, D. Kim, D. E. Kim, A. Chacón, and W. J. M. Kort-Kamp, Ultrafast nonequilibrium dynamics and high-harmonic generation in two-dimensional quantum spin Hall materials, *Phys. Rev. Mater.* **7**, L051201 (2023).
- [21] M. Lytova, M. Spanner, and I. Tamblin, Deep learning and high harmonic generation, *Can. J. Phys.* **101**, 132 (2023).
- [22] J. Serrano, J. M. Pablos-Marín, and C. Hernández-García, Machine-learning applied to the simulation of high harmonic generation driven by structured laser beams, *EPJ Web Conf.* **287**, 13018 (2023).
- [23] Y. Shen, E. Liao, T. Tai, Y. Liao, C. Sun, C. Lee, S. See, and H. Chen, Deep learning-based photodamage reduction on harmonic generation microscope at low-level optical power, *J. Biophotonics* **17**, e202300285 (2023).
- [24] N. D. Klimkin, A. Jiménez-Galán, R. E. F. Silva, and M. Ivanov, Symmetry-aware deep neural networks for high harmonic spectroscopy in solids, *Opt. Express* **31**, 20559 (2023).
- [25] A. Mihailescu, A new approach to theoretical investigations of high harmonics generation by means of fs laser interaction with overdense plasma layers. Combining particle-in-cell simulations with machine learning, *J. Instrum.* **11**, C12004 (2016).
- [26] J.-Z. Yan, S.-S. Zhao, W.-D. Lan, S.-Y. Li, S.-S. Zhou, J.-G. Chen, J.-Y. Zhang, and Y.-J. Yang, Calculation of high-order harmonic generation of atoms and molecules by combining time series prediction and neural networks, *Opt. Express* **30**, 35444 (2022).
- [27] J. M. Pablos-Marín, J. Serrano, and C. Hernández-García, Simulating macroscopic high-order harmonic generation driven by structured laser beams using artificial intelligence, *Comput. Phys. Commun.* **291**, 108823 (2023).
- [28] T. Zahavy, A. Dikopoltsev, D. Moss, G. I. Haham, O. Cohen, S. Mannor, and M. Segev, Deep learning reconstruction of ultrashort pulses, *Optica* **5**, 666 (2018).
- [29] C. Brunner, A. Duensing, C. Schröder, M. Mittermair, V. Golkov, M. Pollanka, D. Cremers, and R. Kienberger, Deep learning in attosecond metrology, *Opt. Express* **30**, 15669 (2022).
- [30] A. Sanchez-Gonzalez *et al.*, Accurate prediction of X-ray pulse properties from a free-electron laser using machine learning, *Nat. Commun.* **8**, 15461 (2017).
- [31] S. Kumar Giri, U. Saalmann, and J. M. Rost, Purifying electron spectra from noisy pulses with machine learning using synthetic Hamilton matrices, *Phys. Rev. Lett.* **124**, 113201 (2020).
- [32] A. Sen, B. K. Parida, K. Jacobs, and D. I. Bondar, Input-output optics as a causal time series mapping: A generative machine learning solution, *Phys. Rev. Res.* **7**, 023015 (2025).
- [33] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, KAN: Kolmogorov-Arnold networks, in *Proceedings of the 13th International Conference on Learning Representations (ICLR)* (OpenReview.net, Singapore, 2025), pp. 1–13.
- [34] M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [35] G. Pang, L. Lu, and G. E. Karniadakis, fPINNs: Fractional physics-informed neural networks, *SIAM J. Sci. Comput.* **41**, A2603 (2019).
- [36] L. Yuan, Y.-Q. Ni, X.-Y. Deng, and S. Hao, A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations, *J. Comput. Phys.* **462**, 111260 (2022).
- [37] J. O’Leary, J. A. Paulson, and A. Mesbah, Stochastic physics-informed neural ordinary differential equations, *J. Comput. Phys.* **468**, 111466 (2022).
- [38] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what’s next, *J. Sci. Comput.* **92**, 88 (2022).
- [39] S. Patra, S. Panda, B. K. Parida, M. Arya, K. Jacobs, D. I. Bondar, and A. Sen, Physics informed Kolmogorov-Arnold neural networks for dynamical analysis via efficient-KAN and WAV-KAN, *J. Mach. Learn. Res.* **26**, 1 (2025).
- [40] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, An expert’s guide to training physics-informed neural networks, [arXiv:2308.08468](https://arxiv.org/abs/2308.08468).
- [41] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, *Adv. Neural Inf. Process. Syst.* **34**, 26548 (2021).
- [42] Z. K. Lawal, H. Yassin, D. T. C. Lai, and A. C. Idris, Physics-informed neural network (PINN) evolution and beyond: A systematic literature review and bibliometric analysis, *Big Data Cognit. Comput.* **6**, 140 (2022).
- [43] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, Stiff-PINN: Physics-informed neural network for stiff chemical kinetics, *J. Phys. Chem. A* **125**, 8098 (2021).

- [44] O. Fuks and H. A. Tchelepi, Limitations of physics informed machine learning for nonlinear two-phase transport in porous media, *J. Mach. Learn. Model. Comput.* **1**, 19 (2020).
- [45] V. Dwivedi, N. Parashar, and B. Srinivasan, Distributed learning machines for solving forward and inverse problems in partial differential equations, *Neurocomputing* **420**, 299 (2021).
- [46] V. I. Arnold, On the representation of functions of several variables as a superposition of functions of a smaller number of variables, in *Collected Works*, edited by A. B. Givental, B. A. Khesin, J. E. Marsden, A. N. Varchenko, V. A. Vassiliev, O. Y. Viro, and V. M. Zakalyukin (Springer, Berlin, 2009), pp. 25–46.
- [47] G. Vidal, Classical simulation of infinite-size quantum lattice systems in one spatial dimension, *Phys. Rev. Lett.* **98**, 070201 (2007).
- [48] A. De Gaetano, PINNs and Sobolev spaces, in *2025 IEEE 29th International Conference on Intelligent Engineering Systems (INES)* (IEEE, Piscataway, NJ, 2025), pp. 000011–000012.
- [49] I. Lukin and A. Sen, Physics-informed time series analysis with Kolmogorov-Arnold networks under Ehrenfest constraints, GitHub (2025), https://github.com/illyaLukin/KAN_TS.
- [50] S. A. Morris, Hilbert 13: Are there any genuine continuous multivariate real-valued functions? *Bull. Am. Math. Soc.* **58**, 107 (2021).
- [51] S. Somvanshi, S. A. Javed, M. M. Islam, D. Pandit, and S. Das, A survey on Kolmogorov-Arnold network, *ACM Comput. Surv.* **58**, 1 (2025).
- [52] Blealtan, Efficient KAN: A memory-efficient KAN implementation, GitHub (2024), <https://github.com/Blealtan/efficient-kan>.
- [53] Z. Bozorgasl and H. Chen, Wav-KAN: Wavelet Kolmogorov-Arnold networks, [arXiv:2405.12832](https://arxiv.org/abs/2405.12832).