

<https://doi.org/10.1038/s44260-025-00062-y>

# Automated global analysis of experimental dynamics through low-dimensional linear embeddings

Check for updates

Samuel A. Moore<sup>1</sup> ✉, Brian P. Mann<sup>1</sup> & Boyuan Chen<sup>1,2,3</sup> ✉

Dynamical systems theory has long provided a foundation for understanding evolving phenomena across scientific domains. Yet, the application of this theory to complex real-world systems remains challenging due to issues in mathematical modeling, nonlinearity, and high dimensionality. In this work, we introduce a data-driven computational framework to derive low-dimensional linear models for nonlinear dynamical systems directly from raw experimental data. This framework enables global stability analysis through interpretable linear models that capture the underlying system structure. Our approach employs time-delay embedding, physics-informed deep autoencoders, and annealing-based regularization to identify novel low-dimensional coordinate representations, unlocking insights across a variety of simulated and previously unstudied experimental dynamical systems. These new coordinate representations enable accurate long-horizon predictions and automatic identification of intricate invariant sets while providing empirical stability guarantees. Our method offers a promising pathway to analyze complex dynamical behaviors across fields such as physics, climate science, and engineering, with broad implications for understanding nonlinear systems in the real world.

Since Isaac Newton published *Philosophiæ Naturalis Principia Mathematica* in 1687 and society entered the Age of Enlightenment, the study of dynamics has shaped our understanding of the natural world. Initially focused on the forces between bodies, this paradigm has evolved into the broader framework of Dynamical Systems Theory. This theory now extends beyond the motion of physical bodies to encompass the study of time-varying state variables in diverse fields, including mechanical and electrical engineering, climate science, neuroscience, physiology, and ecology.

Analyzing real-world dynamical systems remains challenging due to difficulties in modeling, nonlinearity, and high dimensionality. For instance, although researchers can often measure systems, they frequently struggle to identify the underlying dynamics. Even when models are accurate, nonlinearity can restrict analysis to local regions of the state space, making it harder to understand the system's overall behavior<sup>1–3</sup>. Additionally, high dimensionality—characterized by a large number of states in a system—can hinder meaningful interpretation and analysis.

In 1931, Bernard Koopman showed that a suitable change of coordinates could globally linearize nonlinear dynamics, offering a simple mathematical structure, albeit in an infinite-dimensional space<sup>4</sup>. Unlike general dynamical systems, linear systems allow for straightforward global analysis through spectral decomposition and provide numerous control options<sup>5–7</sup>.

Inspired by Koopman's work, recent research demonstrates that low-dimensional exact linearizations are achievable through eigenfunctions<sup>6,8</sup>. Beyond being low-dimensional and linear, eigenfunction coordinates can uncover hidden properties of the system, such as attractor structures and Lyapunov functions, which are not easily found in other coordinate representations or traditional techniques in dynamical systems. This makes eigenfunction coordinates, and even pseudoeigenfunction coordinates, a powerful technique for the automated global analysis of nonlinear dynamics<sup>9–12</sup>.

In data science, researchers consider dimensionality to be one of the most important characteristics of data. They view high-dimensional data as an expression of a low-dimensional underlying manifold with an intrinsic dimension. As a result, researchers propose various techniques for dimensionality reduction and estimation<sup>13–17</sup>. Although the significance of dimensionality in data science is widely recognized, researchers still understand little about its role in Koopman-inspired models for dynamical systems. Moreover, whether low-dimensional eigenfunction representations exist for most systems remains unclear.

Researchers propose numerous methods for finding approximate linear models of nonlinear dynamics, starting with the now ubiquitous Dynamic Mode Decomposition (DMD) and extended Dynamic Mode

<sup>1</sup>Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA. <sup>2</sup>Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. <sup>3</sup>Department of Computer Science, Duke University, Durham, NC, USA. ✉e-mail: [sam.a.moore@duke.edu](mailto:sam.a.moore@duke.edu); [boyuan.chen@duke.edu](mailto:boyuan.chen@duke.edu)

Decomposition (eDMD)<sup>18,19</sup>. Although these methods are generally straightforward to implement, DMD struggles with nonlinear dynamics, and eDMD often produces representations that are much higher dimensional than the original state space, leading to the curse of dimensionality<sup>18,19</sup>.

Deep learning has become an effective tool for nonlinear dimensionality reduction, learning structured latent representations of data, and modeling dynamical systems<sup>20–24</sup>. A prime example is the use of deep convolutional autoencoders to discover the intrinsic dimension of dynamical systems directly from high-dimensional video observations<sup>25,26</sup>. Similarly to this line of research, our work seeks an alternative set of variables to describe dynamical systems beyond the space in which they are measured.

Researchers have used deep learning to find linear embeddings for nonlinear dynamics<sup>27–33</sup>. However, much like DMD-based methods, these approaches still struggle to produce low-dimensional linear models for even simple nonlinear systems. For example, researchers used deep learning to find a 100-dimensional (100D) linear embedding space for the Duffing equation, a 2D bi-stable nonlinear system<sup>28</sup>. Similarly, they used 21D and 1000D embedding spaces with eDMD for the same system<sup>19,34</sup>. Another widely studied benchmark nonlinear system, the Van der Pol Oscillator, was represented as a 100D system<sup>35</sup>, a 20D system<sup>11,36</sup>, and a 28D linear system<sup>37</sup>, among others.

While the examples above suggest that high-dimensional linear embeddings can model nonlinear dynamics, the few analytical examples of Koopman eigenfunctions demonstrate the potential to capture the dynamics with only a few key variables<sup>6</sup>. Moreover, modeling systems with low-dimensional representations in higher-dimensional spaces increases the likelihood of redundancy, spurious modes, and overfitting. However, current methods for discovering eigenfunctions either lack representational capacity or fail to address dimensionality altogether<sup>11,36,38–40</sup>.

We demonstrate that low-dimensional linearization is possible for a wide class of nonlinear systems using our novel, data-driven machine learning approach. Specifically, we show that 3D and 6D representations are sufficient to accurately model the Van der Pol and Duffing oscillators, respectively. Beyond these prototypical systems, we extend our study to numerous previously unstudied, experimental, and real-world nonlinear dynamical systems by discovering their low-dimensional linear representations. Our results demonstrate that this approach improves generalization, enables long-horizon predictions, reduces the occurrence of false modes, and facilitates interpretability, empirical stability analysis, and the discovery of intricate invariant sets.

We achieve these results with a multi-step procedure and a deep autoencoder network with physics-informed optimization to structure the latent space in accordance with Koopman Operator Theory (Fig. 1). This approach works directly from experimental data (Fig. 1a). We use time-delay embedding as a central feature to enhance prediction performance and implement a method to select the length of time-delay before training using mutual information (Fig. 1b). While fitting the networks to learn long-horizon predictions (Fig. 1c), we apply regularization through hyperparameter annealing (Fig. 1d). Once the model is trained, we perform long-horizon predictions (Fig. 1e), discover eigenfunctions (Fig. 1f), and conduct empirical global stability analysis (Fig. 1g). With this framework, we discover entirely new representations for electrical systems, neural circuits, magnetic pendulums, and atmospheric processes. Consequently, our work can facilitate future discoveries in fields such as physics, robotics, biology, and neuroscience, where the choice of coordinate representation is non-trivial, the behaviors are poorly understood, and experimental data are abundant.

## Results

### Studied dynamical systems

We created nine datasets across multiple scientific fields using trajectories from simulated and experimental nonlinear dynamical systems (Fig. 2). The studied systems exhibit a range of behaviors, from relatively simple to highly complex. The single pendulum was the simplest system studied, with only two state variables describing its angular position and velocity (Fig. 2a). Its

simplicity is further highlighted by the presence of a single fixed-point attractor. In contrast, the more complex Van der Pol oscillator dataset features a limit-cycle attractor and was introduced by Dutch electrical engineer Balthasar van der Pol while studying nonlinear circuits in vacuum tubes (Fig. 2b)<sup>41,42</sup>. The Hodgkin-Huxley model, introduced by Alan Hodgkin and Andrew Huxley in 1952 to describe the excitation mechanisms of neurons, features four state variables, strong nonlinearity, and self-sustained oscillations (Fig. 2c). The mechanisms uncovered by this model were foundational in advancing our understanding of neural excitation, earning Hodgkin and Huxley the Nobel Prize in Physiology or Medicine in 1963<sup>43</sup>. The Lorenz-96 system, developed by Edward Lorenz, the founder of chaos theory, served as the next dataset (Fig. 2d). Originally conceived to explore weather predictability, this model was characterized by its high-dimensional state space and its ability to exhibit both periodic and chaotic solutions<sup>44</sup>.

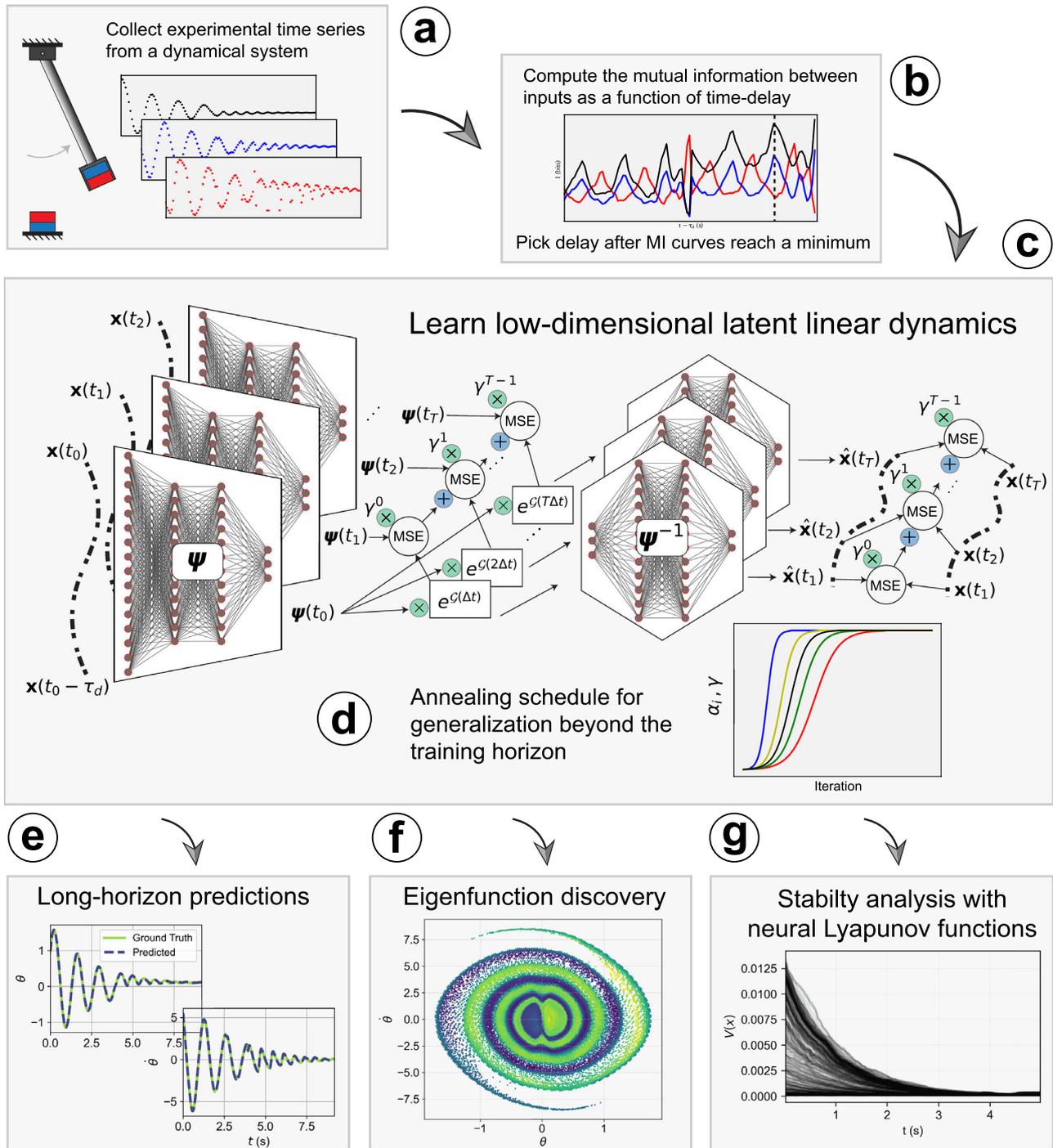
We also constructed four additional datasets from dynamical systems characterized by multistability, in contrast to the previously discussed models. The initial dataset was derived from the Duffing oscillator, introduced by George Duffing in 1918 to analyze mechanical vibrations<sup>45</sup>. Often depicted as a particle mass situated in a double-well potential energy landscape, this system displays more complex stability behavior compared to the earlier datasets (Fig. 2e). The subsequent dataset came from a model involving two interacting magnetic dipoles: one fixed and the other free to oscillate while attached to a spring and damper (Fig. 2f). This model exhibited pronounced nonlinear behavior due to magnetic repulsion compared to the Duffing oscillator and featured particularly asymmetric basins of attraction<sup>46</sup>. We also examined the dynamics of a system with multiple nested limit cycles, as opposed to fixed points (see Supplementary Information). Lastly, we constructed two experimental pendulum datasets; the first was derived from a single pendulum with magnetically induced multi-stability, and the second was from a double pendulum that exhibited chaotic behavior (Fig. 2g, h). For the detailed equations of motion for each of the simulated systems, see the Supplementary Information.

### Long-Horizon Predictions with Low-Dimensional Linear Dynamics

We first evaluated the ability of our approach to make accurate long-horizon predictions, which is the first step towards trusting the model for system analysis and knowledge discovery. We developed a set of algorithmic components to achieve accurate long-horizon predictions, including time-delay, loss functions, curriculum learning, and regularization techniques. During training and inference, we conducted prediction rollouts in the latent space,  $\psi$ , after encoding an initial time-delayed state. The length of the time-delay had a large impact on future prediction error (Fig. S4a, b). However, it remains unclear how to select the time-delay parameter in existing data-driven methods. Therefore, a method to select the time-delay in our latent linear models was developed based on mutual-information (Fig. 1b, S13b,c, Methods), inspired by traditional dynamical systems research in time-delay embedding<sup>47–50</sup>.

We performed model rollouts using the analytical solution to our learned linear model for the dynamics (Fig. 1c, Methods). To ensure accurate predictions, we supervised these rollouts using ground truth future embeddings over  $T$  time steps, progressively scaling down each prediction based on its temporal distance from the present using a discount factor  $\gamma$  (Fig. 1c, Methods). After generating the latent space predictions, we re-projected them into state space using  $\psi^{-1}$ , and during training, we supervised these predictions with ground truth future states (Fig. 1c, Methods).

We found that obtaining predictions that generalized beyond the training horizon was essential for learning accurate dynamics in a low-dimensional latent space. We achieved this prediction generalization capability through systematic annealing of the discount factor, which we implemented as a form of curriculum learning for the prediction horizon (Fig. 1d, Methods). By training with a variable horizon, we demonstrated nearly two orders of magnitude improvement in long-horizon predictions compared to the fixed-horizon approaches used in previous work (Fig. S4).



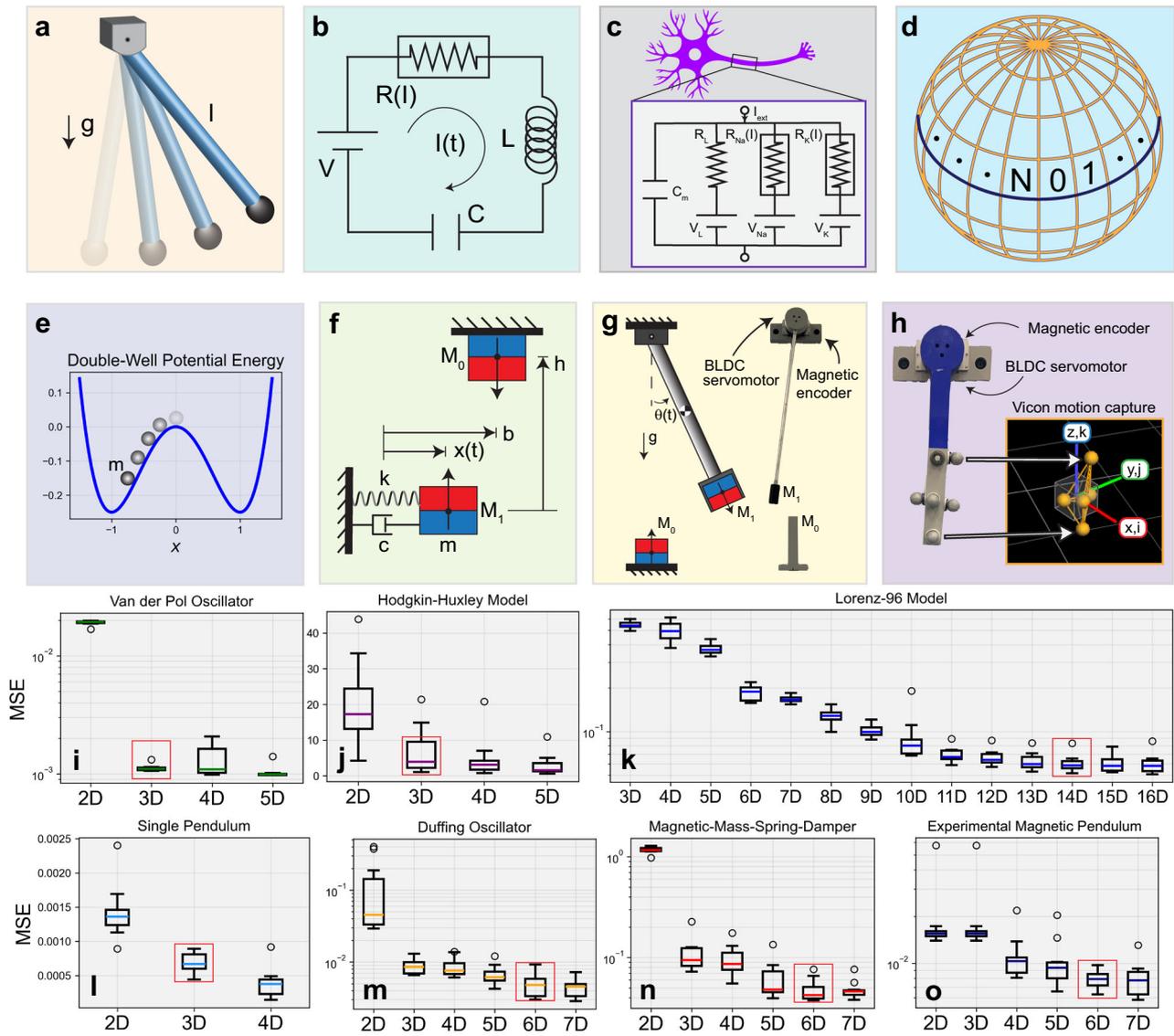
**Fig. 1 | Automated global analysis of experimental dynamics.** An overview of our framework to automate the global analysis of experimental dynamical systems by learning low-dimensional latent linear embeddings. **a** Collect time-series from a dynamical system. **b** Choose the model input dimension by selecting an appropriate time-delay using the mutual-information between trajectories in the system. **c** Train

a deep autoencoder network that constrains the latent space to behave like a linear dynamical system. **d** During training, annealing the coefficient of the loss function and the training prediction horizon to ensure model generalization. **e** Long-horizon predictions. **f** Interpretable eigenfunction discovery. **g** Stability analysis with learned Lyapunov functions.

To select the embedding dimension for each system, we examined the validation prediction error of our trained models across dimensions (Fig. 2i–o). As part of our regularization strategy and to keep the model parsimonious, we chose the smallest embedding dimension that did not significantly degrade performance.

We represented the single pendulum as a 3D linear system. Although the prediction error decreased slightly with the addition of another latent state, the error was already exceptionally small in 3D. We visualized

example predictions in latent space for a trajectory starting at the upright position of the pendulum (Fig. 3a). Leveraging the linear evolution of the latent states, we decomposed the predicted latent states into separate dynamic modes,  $\phi$ , of oscillation, growth, and decay using Koopman Mode Decomposition (Fig. 3b, Methods). The predictions in state space remained accurate across all future time-steps and successfully captured the frequency-shifting behavior in 3D (Fig. 3c, d), as compared to the models trained in 2D (Fig. S2). Furthermore, we found that the 4D models



**Fig. 2 | Datasets and prediction error.** Diagrams detailing the studied dynamical systems and the prediction error as a function of latent dimension. **a** A single pendulum model. **b** A circuit with nonlinear resistance known as the Van der Pol oscillator. **c** A model for how action potential in neurons are initiated and propagated called the Hodgkin-Huxley Model. **d** A model which was devised to study weather predictability known as the Lorenz-96 system. **e** A particle mass situated in a double well potential called the Duffing oscillator. **f** A mass-spring-damper system

with two repelling magnets. **g** An experimental magnetic pendulum. **h** An experimental double pendulum. **i–o** Box and whisker plots showing the mean squared prediction error across embedding dimensions for some of the studied systems. Using our learning approach, the prediction error plateaus with a relatively low-dimensional state space. The red box indicates the latent dimension that the system was modeled in.

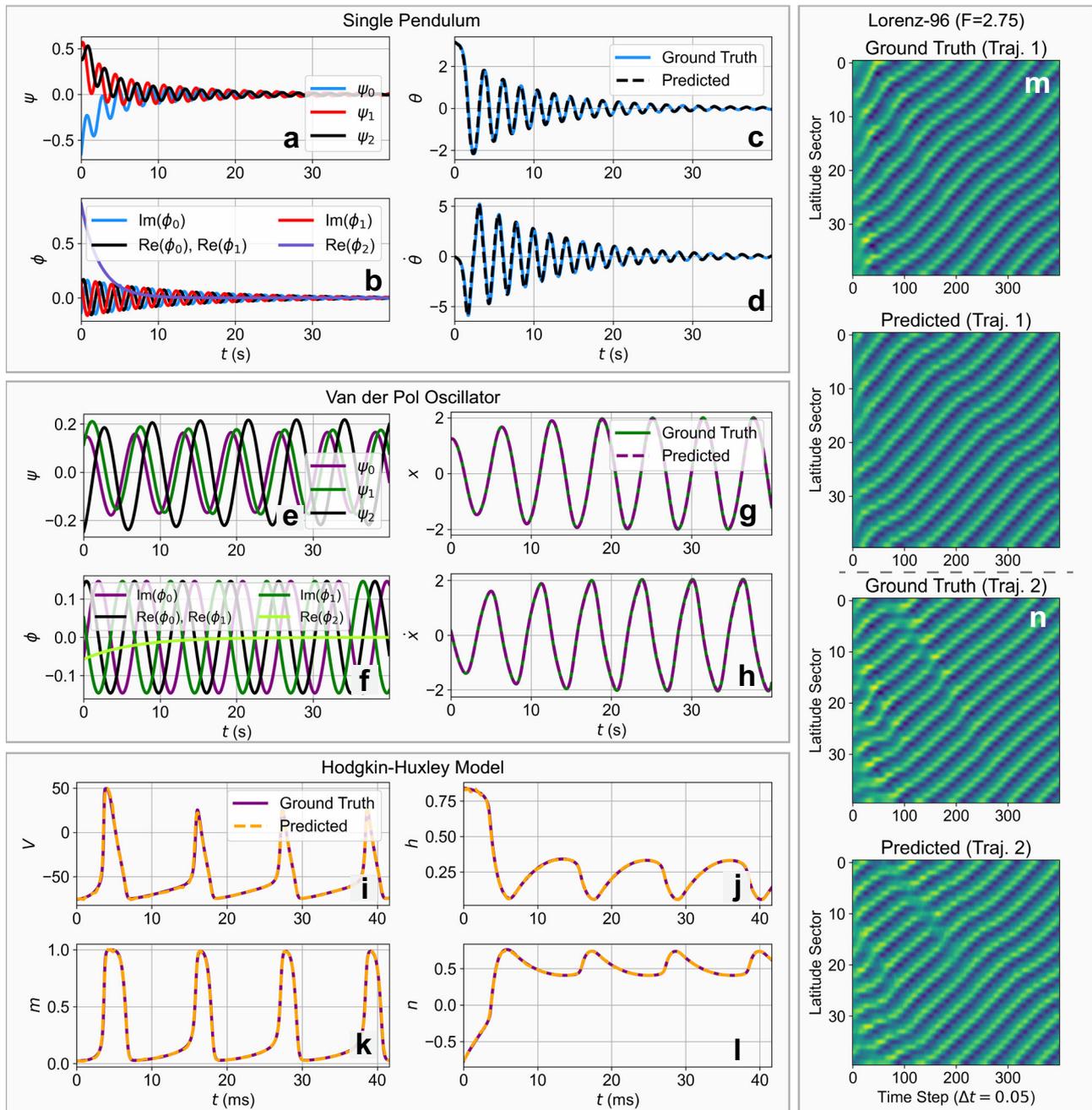
introduced additional modes to the dynamics, which were likely to be false (Fig. S2).

The prediction error for the Van der Pol oscillator plateaued after we lifted the latent space by a single dimension (Fig. 2i). Therefore, we modeled the Van der Pol oscillator as a 3D linear system, in stark contrast to the 100D, 28D, and 28D models in previous work<sup>11,35–37</sup>. As with the other systems, we integrated the dynamics ahead in time in the latent space (Fig. 3e) and decomposed the predictions into separate dynamic modes (Fig. 3f). When projected back into state space, our predictions closely matched the ground truth over an extended horizon, even capturing the transient behavior before settling on the limit-cycle attractor (Fig. 3g, h). In contrast, the models trained in 2D failed to capture this transient behavior, while the 4D model exhibited an unstructured latent space, indicating overfitting (Fig. S2).

Interestingly, we observed that the prediction error for the Hodgkin-Huxley model plateaued before reaching the original state dimension of four (Fig. 2j). This suggests redundancy in the original coordinate choice by

Hodgkin and Huxley and confirms that our approach is capable of discovering novel coordinates for dimensionality reduction in addition to linearization. Like our model for the Van der Pol Oscillator, the model trained with the Hodgkin-Huxley dataset achieved accurate long-horizon predictions, including transient behavior (Fig. 3i–l). This dimensionality reduction of the Hodgkin-Huxley system is likely enabled by a strong correlation between states, which is visible in their synchronous oscillations (Fig. 3i–l). We also reduced the dimensionality of the Lorenz-96 model, which exhibits limit-cycle behavior, from the original state dimension of 40 to 14 (Fig. 2k). Despite this substantial reduction, the predictions in the original state space aligned closely with the ground truth trajectories over a long horizon (Fig. 3m, n). Like the Hodgkin-Huxley system, dimensionality reduction was possible in the Lorenz-96 system due to the shared information between states.

We also achieved long-horizon prediction capabilities with low-dimensional linear models on systems with multiple attractors. We



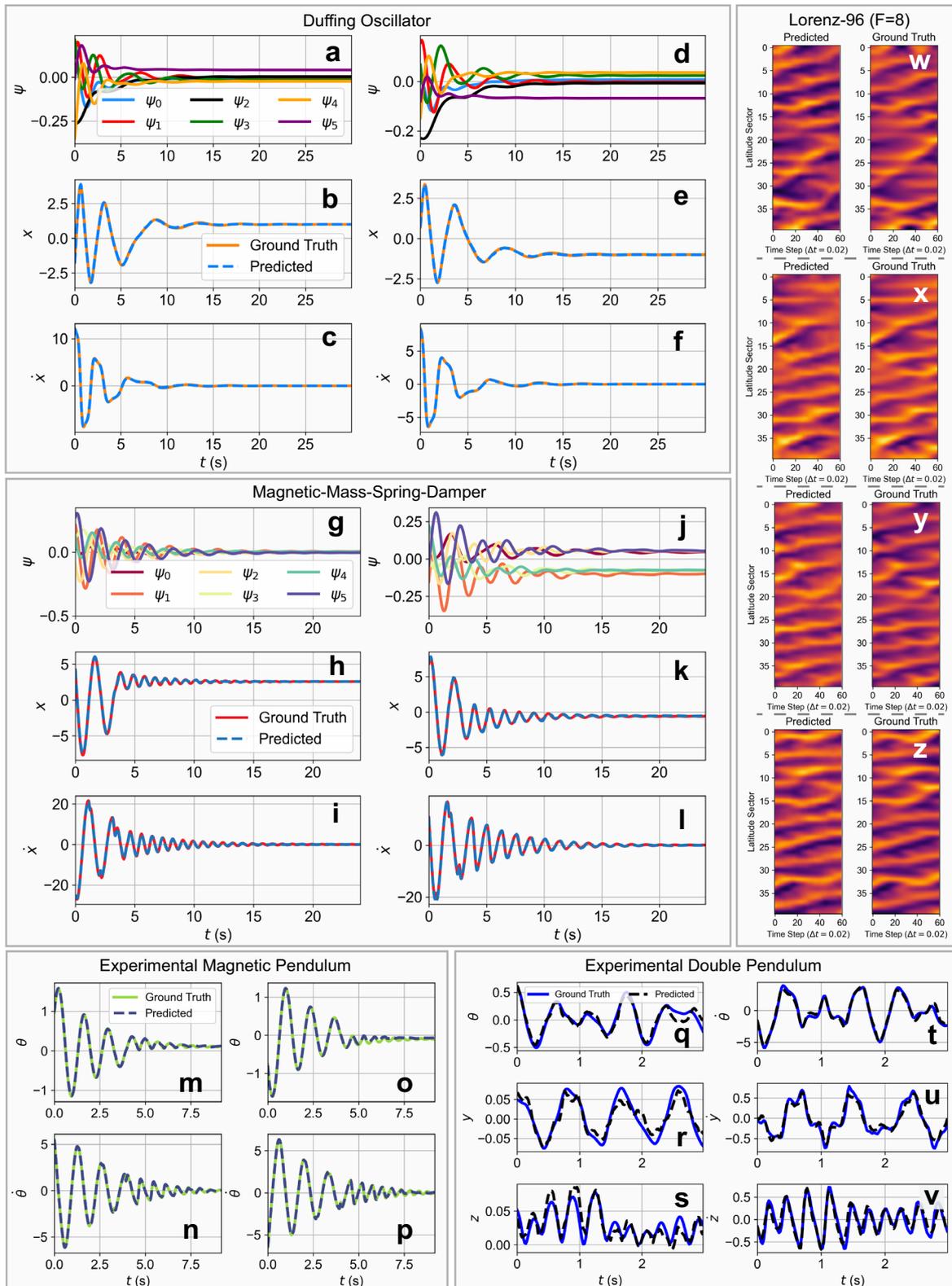
**Fig. 3 | Long-horizon predictions.** Predicted trajectories from low-dimensional linear embeddings of nonlinear dynamics. **a** Predicted trajectories in latent space for the single pendulum modeled as a 3D linear system. **b** The same latent space trajectories for the pendulum, decomposed into separate modes. **c, d** Ground truth and predicted trajectories for angular position and velocity after decoding into state space. **e** Predicted latent states for the Van der Pol oscillator as a 3D linear system.

**f** The same latent space trajectories for the Van der Pol oscillator, decomposed into separate modes. **g, h** Ground truth and predicted trajectories for the state variables of the Van der Pol oscillator after decoding. **i–l** Ground truth and predicted trajectories in state space for the Hodgkin-Huxley model as a 3D linear system. **m, n** Future predicted and ground truth states for the periodic Lorenz-96 model with 40 latitude sectors, represented as a 14D linear system.

modeled the Duffing oscillator as a 6D linear system (Fig. 2m) compared to the 21D, 100D, and 1000D models used in previous work<sup>19,28,34</sup>. The predicted evolution of the six latent states for a high energy initial condition is shown in Fig. 4a. After decoding the predictions, we accurately captured both inter-well and intra-well oscillations, as well as the correct resting attractor (Fig. 4c, d). Predicting the intra-well oscillations was particularly challenging for the 4D model (Fig. S2). Additionally, we plotted a second trajectory starting from an initial condition close to the first in Fig. 4d. The model correctly inferred that the system comes to rest

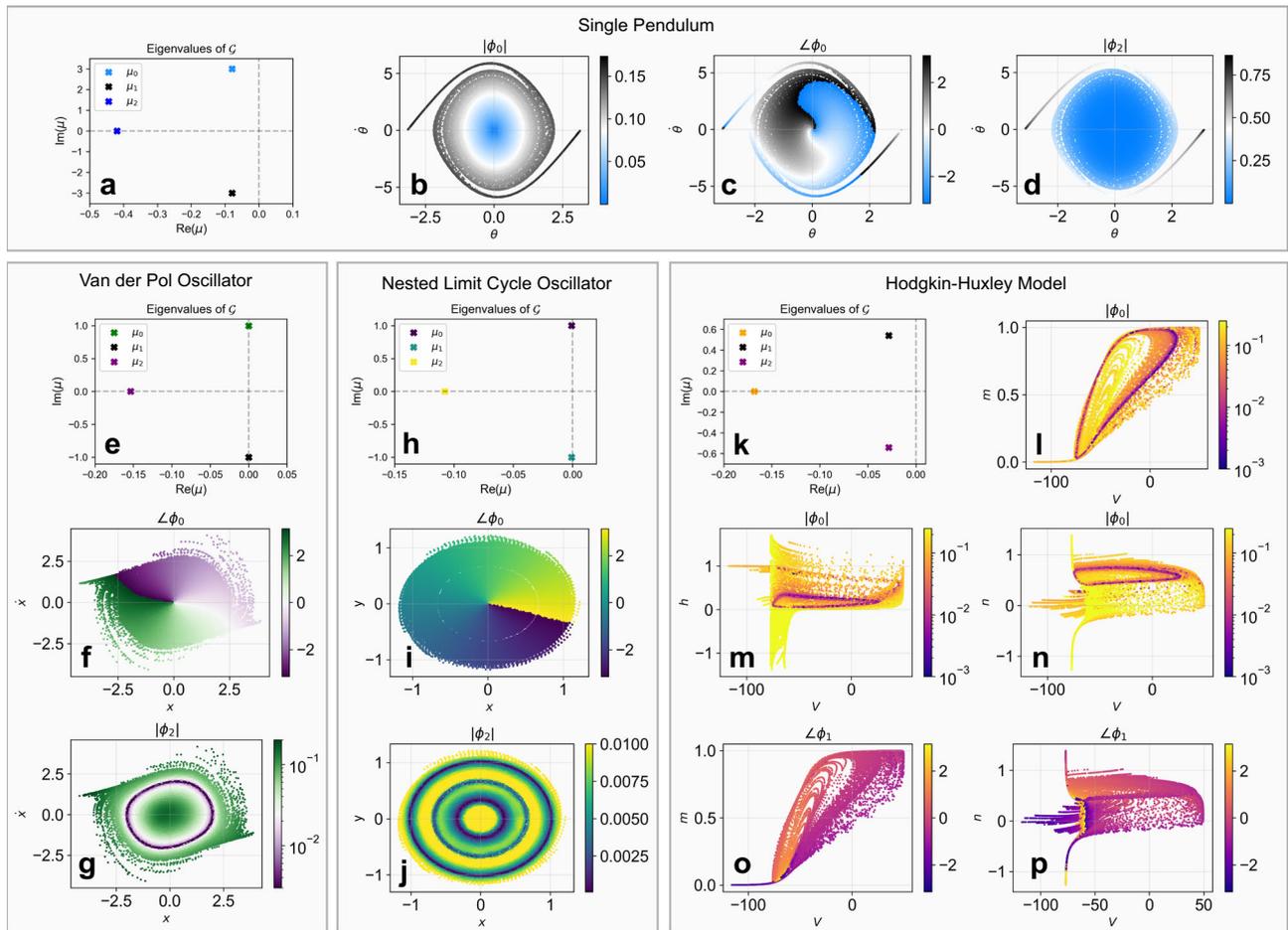
in the opposite attractor, even though the initial conditions were close in state space (Fig. 4e, f).

Similar to the Duffing oscillator, we observed the prediction error level-off at 6D in the simulated and experimental magnetic systems (Fig. 2n, o). Our approach successfully modeled the full scope of nonlinear behavior in the magnetic-mass-spring-damper and magnetic pendulum systems. This included the impact-like velocity jumps as the magnets come into close proximity, the various frequencies of oscillation, and the multi-stability (Fig. 4g–l).



**Fig. 4 | Long-horizon predictions for multi-stable and chaotic systems.** **a** The six learned latent states for the Duffing oscillator, forecasting over an extended horizon. **b, c** Predicted and ground truth trajectories after decoding, including the correctly anticipated resting attractor. **d–f** Forecasted latent variables and states for the Duffing oscillator that come to rest in the opposite potential well compared to the previous trajectory. **g–l** Forecasted latent variables (in 6D) and states for the magnetic-mass-spring-damper system with asymmetric basins of attraction. **g–i**

Predicted and ground truth trajectories that come to rest in the smaller attractor. **j–l** Predicted latent variables and states for the magnetic-mass-spring-damper system that come to rest in the larger attractor. **m–p** Ground truth and predicted trajectories for each attractor of the experimental magnetic pendulum when modeled as a 6D linear system. **q–v** Long-horizon predicted and ground truth trajectories for the measured states of the experimental double pendulum. **w–z**, Four forecasted and ground truth trajectories for the chaotic Lorenz-96 model with 40 states.



**Fig. 5 | Learned eigenvalues and eigenfunctions.** **a** The learned eigenvalues for the single pendulum model in a 3D embedding space. The complex part corresponds to the frequency of oscillation, and the real part corresponds to decay (or growth if positive). **b, c** The magnitude and phase of one of the complex-valued, learned eigenfunctions as a function of the input states. **d** The magnitude of the purely real, learned eigenfunction. **e** The learned eigenvalues for the Van der Pol oscillator in a

3D embedding space. **f, g** The phase of one of the purely complex-valued eigenfunctions and the magnitude of the purely real-valued eigenfunction. **h–j** The learned eigenvalues and eigenfunctions for a model with three nested limit cycle attractors as a 3D linear system. **k–p** The learned eigenvalues and eigenfunctions for the Hodgkin-Huxley model as a 3D linear system. Some of the depicted color maps are clamped for visual clarity.

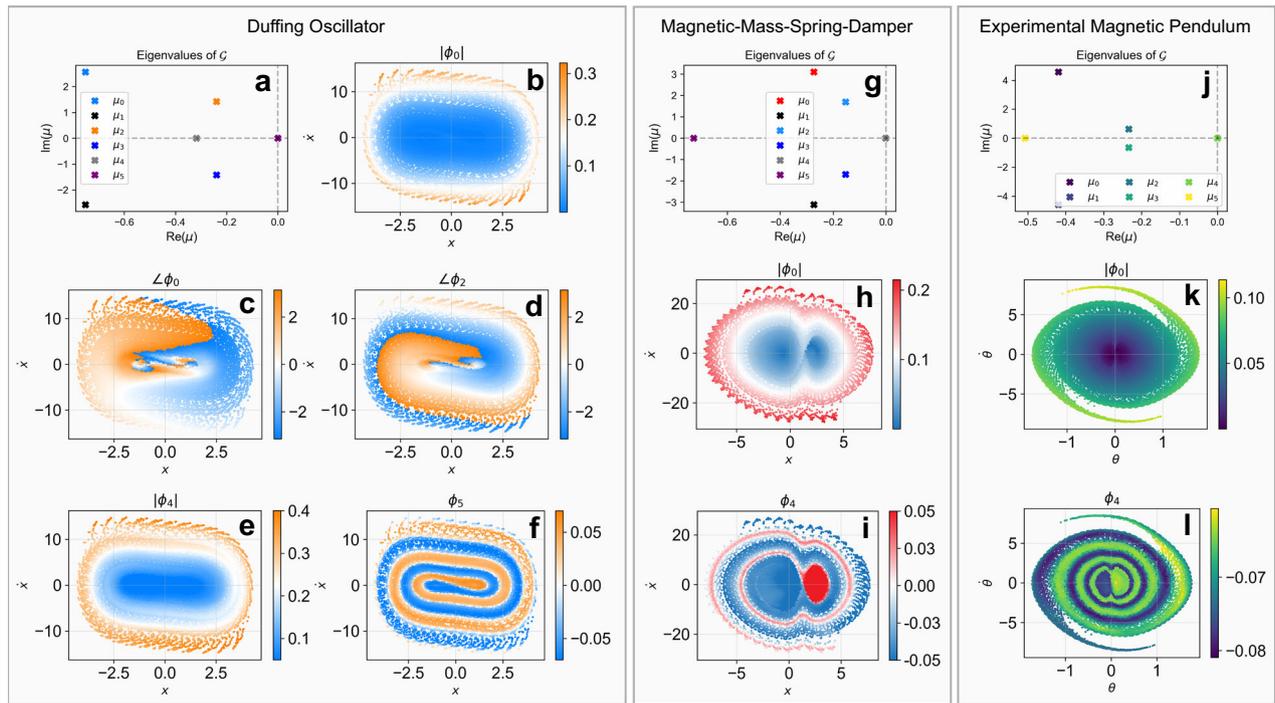
Lastly, chaotic systems, such as the experimental double pendulum, pose significant challenges for finite-dimensional Koopman models because of their spectral characteristics<sup>51</sup>. Due to this, we observed little correlation between the prediction error and the embedding dimension (Fig. S2) and thus chose to model the experimental double pendulum and chaotic Lorenz system’s in their measured dimensions of 6D and 40D. While latent linear models can only be described as pseudoeigenfunction coordinates for chaotic systems (Methods), they still provide an interpretable and accurate framework for forecasting future states in the experimental double pendulum and the chaotic Lorenz-96 model (Fig. 4q–z).

**Eigenfunction discovery and spectral analysis**

Eigenfunctions play a crucial role in understanding physical phenomena across various fields, including quantum mechanics, vibration analysis, and thermal processes. Traditionally, eigenfunctions are derived from linear systems theory, which limits their direct application to nonlinear systems. However, because our approach results in a linear representation of nonlinear systems, we can directly apply modal decomposition to characterize the latent variables as complex-valued eigenfunctions or pseudoeigenfunctions, providing a detailed description of the system’s behavior. Specifically, the eigenvalues, eigenvectors, eigenfunctions, and pseudoeigenfunctions allowed us to analyze oscillation frequencies, phases, and growth and decay rates within each system (Methods). Additionally, we used the decaying modes of

the system to construct neural Lyapunov functions (Methods), which characterized the attractive and equilibrium states of the system and provided stability guarantees. To ensure that the learned modes were physically realistic, we constructed a term in the loss function that penalized positive real-part eigenvalues, which would otherwise lead to unbounded exponential growth in the predicted trajectories (Methods).

We first examined the spectrum of the pendulum and identified a complex conjugate pair of eigenvalues,  $\mu_{0,1} = \alpha \pm \beta i$ , and a purely real-valued eigenvalue  $\mu_2$  (Fig. 5a). The real-part of the eigenvalues were negative, which corresponds to global exponential decay in the system. The imaginary part of the complex-conjugate pair indicates the natural frequencies in latent space (Methods). We explored the system’s eigenfunctions as functions of the input states (Fig. 5b–d) and as trajectories (Fig. 3b). When we examined the magnitude of the eigenfunction  $\phi_0$ , which corresponded to  $\mu_0$ , it revealed decay toward the pendulum’s downward position,  $\theta = \theta = 0$  (Fig. 5b). The minima in Koopman eigenfunctions indicate an equilibrium or attractor state in the dynamics (Methods), which we will further verify in the studied systems through empirical Lyapunov analysis in the next section. Notably, the equilibrium state in the pendulum was automatically revealed as a feature of the learned latent space. Our model also provided further interpretation of the system through the phase of the oscillatory mode, which described how the periodic latent trajectory shifted as a function of the state variables (Fig. 5c). Lastly, we studied the magnitude of the final learned



**Fig. 6 | Learned eigenvalues and eigenfunctions for multi-stable systems.** **a** The learned eigenvalues of the 6D latent linear model for the Duffing oscillator. **b–e** The magnitude and phase of two complex-valued learned eigenfunctions for the Duffing oscillator. **f** The learned eigenfunction for the system with the smallest magnitude eigenvalue. **g** The six learned eigenvalues for the magnetic-mass-spring-damper

system. **h** The magnitude of one of the complex-valued eigenfunctions for the system. **i** The learned eigenfunction for the magnetic-mass-spring-damper system with the smallest magnitude eigenvalue. **j–l** The learned eigenvalues and two eigenfunctions from the experimental magnetic pendulum system.

eigenfunction,  $|\phi_2|$ , which also identified decay toward the downward position, but at a faster rate (Fig. 5d).

We also performed spectral decomposition of the model for the Van der Pol oscillator and discovered a pair of purely oscillatory modes and a purely decaying mode (Fig. 5e). When we examined the phase of the oscillatory mode as a function of the inputs, it revealed an intricate symmetry (Fig. 5e). The magnitude of the purely real eigenvalue served as a neural Lyapunov function, allowing us to identify the limit-cycle attractor (Fig. 5g). Our approach automatically uncovered this attractor by modeling the Van der Pol oscillator as a 3D linear system, eliminating the need for a high-dimensional representation that would have required further refinement after learning<sup>11</sup>. We observed a similar decomposition of modes with the more complex nested limit cycle and Hodgkin-Huxley models (Fig. 5h–p).

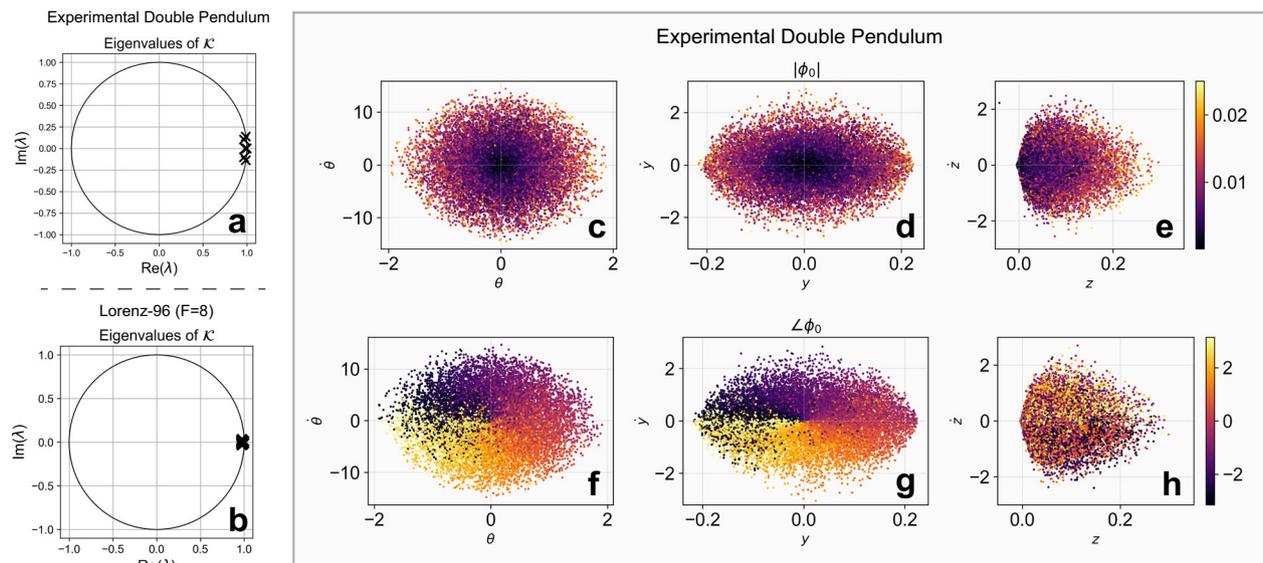
Most of the multi-stable systems required additional dimensions for accurate predictions, which lead to more modes of behavior compared to the mono-stable systems. We identified six eigenvalues for the Duffing oscillator, revealing four oscillatory and decaying modes, each characterized by two distinct frequencies and decay rates, along with a purely decaying mode and a static mode with negligible real or imaginary components (Fig. 6a). Similar to the previous systems, we used the decaying modes to construct neural Lyapunov functions and identified the equilibria of the system. By examining the phases of the various oscillation frequencies, we uncovered interesting patterns (Fig. 6b–e). Unlike the mono-stable systems, our learned model for the Duffing oscillator includes a static mode that separates the state space into two parts corresponding to the basins of attraction (Fig. 6f). The basin of attraction for an equilibrium represents the set of initial conditions that lead to that equilibrium, as  $t \rightarrow \infty$ . Previous methods have used eigenfunctions to identify the basins of attraction for the Duffing oscillator, but they were limited to small regions in state space and did not capture the inter-well oscillations that result in the spiral pattern seen in Fig. 6f<sup>19,28</sup>. Not only did our approach improve upon previous Koopman-based methods, but, unlike brute-force techniques in traditional dynamical

systems, we automatically discovered the basins of attraction as a continuous function of the original state variables.

Our learned model for the magnetic-mass-spring-damper had a similar structure to the Duffing oscillator, with four oscillating and decaying modes, a decaying mode, and a static mode (Fig. 6g). We used the magnitude of the first mode, acting as a neural Lyapunov function, to reveal decay in the state space, indicating the presence of equilibrium states (Fig. 6h). By analyzing the static mode, we uncovered the vastly asymmetric basins of attraction (Fig. 6i), which validated that our approach is not limited to systems with simple nonlinearities, such as the Duffing equation. When we examined the latent dynamics of the experimental magnetic pendulum, we found modes very similar to those of the two previous systems (Fig. 6j–l). It is important to note that comparing system behavior, especially from observational data, is not straightforward when using traditional local methods for dynamical systems analysis<sup>1</sup>. However, with our approach, we automatically revealed the system’s behavior through the learned modes, allowing us to directly compare the systems. Our results further suggest that we can make this comparison based on dimensionality, as systems with similar behavior are often modeled with the same number of dimensions (Fig. 2i–o).

Chaotic or mixing systems typically lack nontrivial Koopman eigenvalues or eigenfunctions; instead, they are characterized by eigenmeasures<sup>52</sup>. In such cases, our framework naturally learns pseudoeigenfunctions (Methods) without requiring additional modifications. Pseudoeigenfunctions extend the notion of eigenfunctions to points in the continuous spectrum, providing approximate yet meaningful descriptors of coherent structures in the dynamics.

For the systems we studied, the learned discrete-time operators have eigenvalues with magnitudes close to one, i.e., they are approximately neutrally stable or unitary (Fig. 7a, b). Under these conditions, every spectral point admits an associated pseudoeigenfunction. An example for the experimental double pendulum is shown in Fig. 7c–h. The pseudoeigenfunction’s magnitude (Fig. 7c–e) exhibits a distinct minimum at the



**Fig. 7 | Learned eigenvalues and pseudo-eigenfunctions for chaotic systems.** a, b The eigenvalues for the learned linear (discrete-time) operator of the experimental double pendulum and the chaotic Lorenz-96 system. c–h The magnitude and phase of a pseudo-eigenfunction for the experimental double pendulum.

downward equilibrium, while its phase captures the oscillatory structure around the top-link angle  $\theta$  and the bottom-link horizontal displacement  $y$ .

### Empirical Lyapunov stability

At large, stability analysis has been used to provide safety guarantees in chemical reactors, understand the spread of disease, construct controllers for autonomous systems, and design aircraft wings. However, in most cases, stability analysis is limited to local regions in state space, making global stability analysis challenging for nonlinear systems. While Lyapunov’s direct method provides a global approach, its practical application has been limited due to the difficulty of finding an appropriate Lyapunov function. In fact, there are no known general analytical methods for finding Lyapunov functions. Our method, as noted earlier, automatically provides a neural Lyapunov function, enabling the application of Lyapunov stability analysis on observational data from nonlinear systems. We empirically analyzed the stability of the studied systems using this approach. A Lyapunov function  $V(\mathbf{x}) \in \mathbb{R}$  satisfies  $V(\mathbf{x}) \geq 0$  and  $\dot{V}(\mathbf{x}) \leq 0$  for all states  $\mathbf{x} \in \mathbb{R}^{n_2}$ . If the neural Lyapunov functions we have learned meet these conditions for observational data, we can conclude that the system is asymptotically stable. Moreover, each sub-level set of the function is forward invariant, which includes static equilibria, more general attractors like limit cycles, and basins of attraction<sup>29,11,53</sup>. We constructed the neural Lyapunov functions by taking the magnitude of the eigenfunctions associated with eigenvalues that have negative real parts. The positive semi-definite condition is satisfied by taking the magnitude, and the rate of change is negative due to the eigenvalue (Methods). With this in mind, we used the learned eigenfunctions as powerful tools for stability analysis.

The superimposed time-series trajectories on the neural Lyapunov functions for the pendulum clearly showed evolution toward the minima, which coincided with the downward position of the pendulum at zero velocity (Fig. 8a, b). We evaluated the function on ground truth trajectories from the pendulum to verify the negative rate of change of the Lyapunov function and confirm that the downward position of the pendulum is globally asymptotically stable. These evaluated trajectories demonstrated exponential decay toward zero, certifying the automatically generated stability hypothesis (Fig. 8c, d).

The neural Lyapunov function we learned for the Hodgkin-Huxley model revealed a particularly intricate structure when projected into the system’s state space (Fig. 5l–n). The complexity of this representation is evident, especially considering that no analytical form of this function has yet been derived. By superimposing the function onto trajectories from the

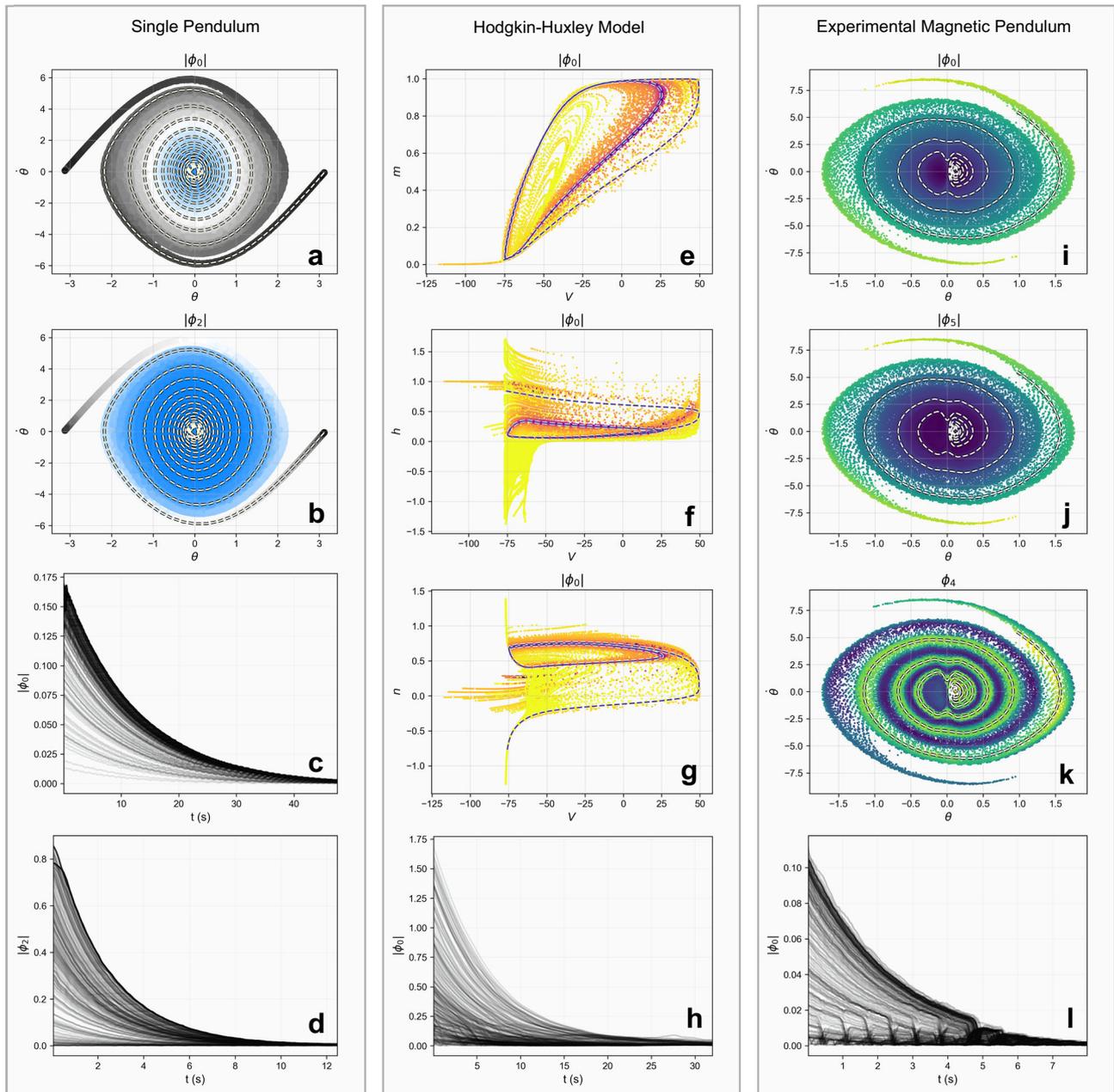
system, we observed that our learned function clearly describes the system’s limit-cycle attractor (Fig. 8e–g). Similar to our approach with the single pendulum, we verified the stability of this attractor by evaluating the Lyapunov function on trajectories from the system, confirming the exponential decay toward the attractor (Fig. 8h).

We demonstrated our method’s ability to perform stability analysis on noisy experimental data using the magnetic pendulum. By plotting the system’s trajectories on the learned eigenfunctions, we emphasized their temporal evolution and identified equilibrium states at the minima (Fig. 8i, j). The plotted trajectory remained entirely within the estimated basin of attraction, which we determined from the static mode of the dynamics (Fig. 8k). We verified the asymptotic stability of the system’s fixed points by analyzing trajectories from the system (Fig. 8l). Similarly, the neural Lyapunov functions we generated for the other dynamical systems indicated the stability of their respective attractors (Fig. S2).

### Discussion

In this work, we introduce a novel framework for the automated global analysis and forecasting of nonlinear dynamical systems by learning low-dimensional linear embeddings directly from experimental data. Our approach stands out by discovering significantly lower-dimensional linear embeddings, often an order of magnitude smaller than those found by previous methods. While it is clear that some systems require higher-dimensional embeddings than others to be represented by linear dynamics. Our framework flexibly adapts to this, as it aims to find the minimal dimensionality sufficient for accurate long-term predictions. Additionally, we achieved substantial improvements in long-horizon prediction accuracy and generalization performance over prior work. Unlike earlier methods that required refinement after learning, our approach directly produces neural Lyapunov functions, enabling straightforward global stability analysis. This was made possible through the integration of deep autoencoder networks, time-delay observables, and new regularization techniques. Using this framework, we derived nine novel coordinate representations for both prototypical and experimental dynamical systems, offering new insights into their behavior.

Future research could expand this method to process higher-dimensional data streams, such as video and audio. Enhancing the data efficiency of the framework is another path for improvement, particularly when data collection is costly or labor-intensive. Another promising direction is to extend this framework to controlled systems. Moreover, the current reliance on a brute force search to determine the appropriate embedding dimension suggests that future work could focus on developing methods for automatic



**Fig. 8 | Lyapunov analysis and invariant sets.** **a, b** A trajectory from the test dataset superimposed on two neural Lyapunov functions for the single pendulum. Each sublevel set for the neural Lyapunov functions serve as forward invariant sets. **c, d** Empirical global stability analysis by evaluating ground truth trajectories using the neural Lyapunov function. This confirms the negative definite condition for the time rate of change of the Lyapunov function. **e–g** A ground truth trajectory overlaid

upon the neural Lyapunov function for the Hodgkin-Huxley limit-cycle attractor. **h** Empirical verification of the Lyapunov function by evaluation with ground truth data. **i, j** A trajectory superimposed upon two neural Lyapunov functions for the experimental magnetic pendulum. **k** The same trajectory overlaid on the neural eigenfunction the corresponds to system’s basin of attraction. **l** Empirical stability analysis using trajectories from the experimental magnetic pendulum dataset.

embedding dimension discovery or uncovering theoretical connections between system behavior and latent linear dimensions.

## Methods

### Koopman operator theory

Consider the following continuous-time dynamical system on a state space  $\mathcal{M} \subseteq \mathbb{R}^n$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{1}$$

where  $\mathbf{x}(t) \in \mathcal{M}$  is the state at time  $t \in \mathbb{R}$ ,  $\mathbf{f} : \mathcal{M} \rightarrow \mathcal{M}$  is the dynamics operator, and the overdot indicates the derivative with respect to time. The

flow map  $\mathbf{F}^t : \mathcal{M} \rightarrow \mathcal{M}$  integrates initial conditions  $\mathbf{x}(t_0) \equiv \mathbf{x}_0$  to time  $t$ , and is defined as

$$\mathbf{F}^t(\mathbf{x}_0) := \mathbf{x}_0 + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\tau))d\tau. \tag{2}$$

Now, let us consider  $\Psi(\mathcal{M})$  the set of scalar measurements or observable functions  $\psi : \mathcal{M} \rightarrow \mathbb{C}$ . Then, the time-parameterized family of Koopman operators  $\mathcal{K}^t : \Psi(\mathcal{M}) \rightarrow \Psi(\mathcal{M})$  is given by

$$\mathcal{K}^t \psi(\mathbf{x}) = \psi \circ \mathbf{F}^t(\mathbf{x}). \tag{3}$$

If the flow,  $\mathbf{F}^t$ , is smooth and continuous we can define the infinitesimal generator  $\mathcal{G} : \Psi(\mathcal{M}) \rightarrow \Psi(\mathcal{M})$  of  $\mathcal{K}^t$  as

$$\mathcal{G}\psi := \lim_{t \rightarrow 0^+} \frac{\mathcal{K}^t \psi - \psi}{t} = \lim_{t \rightarrow 0^+} \frac{\psi \circ \mathbf{F}^t - \psi}{t} = \dot{\psi}, \quad (4)$$

and  $\mathcal{K}^t = \exp(\mathcal{G}t)$ . The generator  $\mathcal{G}$  gives rise to a continuous-time linear dynamical system in observable coordinates

$$\dot{\psi} = \mathcal{G}\psi. \quad (5)$$

When applying the chain rule to  $\dot{\psi}$  we also get that

$$\dot{\psi}(\mathbf{x}) = \partial_{\mathbf{x}}\psi(\mathbf{x})\dot{\mathbf{x}} = \partial_{\mathbf{x}}\psi(\mathbf{x})\mathbf{f}(\mathbf{x}). \quad (6)$$

An observable  $\phi$  is an eigenfunction to the generator  $\mathcal{G}$  if, for a corresponding eigenvalue  $\mu \in \mathbb{C}$ ,

$$\dot{\phi}(\mathbf{x}) = \mathcal{G}\phi(\mathbf{x}) = \mu\phi(\mathbf{x}). \quad (7)$$

This also implies that  $\phi$  is an eigenfunction for  $\mathcal{K}^t$  with the eigenvalues  $\lambda^t = \exp(\mu t)$ . Eigenfunctions can also form a basis for observables with spectral decomposition.

$$\psi(\mathbf{x}) = \sum_{j=1}^{\infty} \phi_j(\mathbf{x})\mathbf{v}_j, \quad (8)$$

where  $\mathbf{v}_j$  is the  $j$ -th Koopman mode and  $\psi$  is a vector of observables. With the spectral decomposition (Eq. (8)), we can represent the flow of observables as

$$\begin{aligned} \psi(\mathbf{x}(t_k)) &= \mathcal{K}^{t_k} \sum_{j=1}^{\infty} \phi_j(\mathbf{x}_0)\mathbf{v}_j \\ &= \sum_{j=1}^{\infty} \lambda_j^{t_k} \phi_j(\mathbf{x}_0)\mathbf{v}_j. \end{aligned} \quad (9)$$

which is known as Koopman Mode Decomposition<sup>54</sup>. Koopman Mode Decomposition highlights the benefits of the operator-theoretic perspective of dynamical systems, showing that a system's behavior is completely characterized by the interplay of  $\lambda_j$ ,  $\phi_j$ , and  $\mathbf{v}_j$ .

Some dynamics, such as chaotic or mixing systems, have a continuous spectrum and thus  $\mathcal{K}^t$  may not have nontrivial eigenvalues and eigenvectors<sup>52</sup>. In this case, we instead consider the spectrum as

$$\sigma(\mathcal{K}^t) = \{z \in \mathbb{C} : (\mathcal{K}^t - zI)^{-1} \text{ is unbounded}\}. \quad (10)$$

The spectrum  $\sigma(\mathcal{K}^t)$  includes all eigenvalues of  $\mathcal{K}^t$ , but may also contain points that are not eigenvalues. Moreover, if  $\mathcal{K}^t$  is normal ( $\mathcal{K}\mathcal{K}^* = I$ ), then for every spectral point  $\lambda_{\sigma}^t \in \sigma(\mathcal{K}^t)$  and  $\epsilon > 0$  there exists an observable

$$\|(\mathcal{K}^t - \lambda_{\sigma}^t I)\phi_{\epsilon}\| \leq \epsilon. \quad (11)$$

These observables are called  $\epsilon$ -pseudoeigenfunctions and satisfy

$$\|\mathcal{K}^t \phi_{\epsilon} - \lambda_{\sigma}^t \phi_{\epsilon}\| = \mathcal{O}(t\epsilon), \quad \forall t. \quad (12)$$

These pseudoeigenfunctions describe persistent behavior and coherent structures within the system without proper diagonalization, as in the Koopman Mode Decomposition (Eq. (9)).

Lastly, the eigenfunctions of a dynamical system can be used to construct Lyapunov functions for stability analysis. A Lyapunov function  $V(\mathbf{x}) \in \mathbb{R}$  satisfies  $V(\mathbf{x}) \geq 0$  and  $\dot{V}(\mathbf{x}) \leq 0$  for all  $\mathbf{x} \in \mathcal{M}$ . Then, the magnitude of eigenfunctions  $|\phi|$  with  $\text{Re}(\mu) < 0$  satisfies these conditions

(Eq. (7)). Moreover, any sub-level set

$$\mathcal{Q} = \{x \mid V(\mathbf{x}) \leq c\} \quad (13)$$

with  $c \geq 0$  is forward invariant and the zero-level set is globally asymptotically stable<sup>11,53</sup>.

### Loss function

Instead of considering the observables  $\psi$  as simple functions of the state, we input a small trajectory or set of delayed states. Time-delayed inputs have been widely used in a Koopman-informed models across various contexts and have proven to be a rich set of observables<sup>28,37,55–57</sup>. To illustrate this, we consider a vector of time-delayed states  $\mathbf{X}_j$  starting at time  $t_j$ :

$$\mathbf{X}_j = \left[ \mathbf{x}(t_j)^T, \mathbf{x}(t_j - \tau_0)^T, \dots, \mathbf{x}(t_j - \tau_d)^T \right]^T, \quad (14)$$

where  $\tau_i$  are delays and  $i \in (0, 1, \dots, d)$ . We use the encoder  $\psi : \mathbb{R}^{(d+2)n} \rightarrow \mathbb{R}^m$  to transform the states into their latent representation in  $\mathbb{R}^m$ , and  $\psi^{-1} : \mathbb{R}^m \rightarrow \mathbb{R}^n$  to transform them back into state space, but only for the leading time  $t_j$ . In other words, the estimated or reconstructed state vector is given by  $\hat{\mathbf{x}}_j = \psi^{-1}(\psi(\mathbf{X}_j))$ . This reconstruction requirement gives us the first term of the loss function:

$$\mathcal{L}_{\mathbf{x}_0} = \frac{1}{n} \|\hat{\mathbf{x}}_0 - \mathbf{x}_0\|_2^2 = \frac{1}{n} \|\psi^{-1}(\psi(\mathbf{X}_0)) - \mathbf{x}_0\|_2^2. \quad (15)$$

In addition to reconstructing the leading input, the autoencoder and the latent space need to meet several more requirements informed by Koopman Operator Theory. First, by applying the chain rule to the observables (Eq. (6)), we derive the second term in the loss function:

$$\mathcal{L}_{\dot{\psi}} = \frac{1}{m} \|\mathcal{G}\psi(\mathbf{X}_0) - \partial_{\mathbf{x}}\psi(\mathbf{X}_0)\dot{\mathbf{X}}_0\|_2^2. \quad (16)$$

Next, to ensure good predictions with the linear model in latent space, we include the following term in the loss function:

$$\begin{aligned} \mathcal{L}_{\psi_t} &= \frac{1}{mT} \sum_{j=1}^T \gamma^{j-1} \|\hat{\psi}(\mathbf{X}_j) - \psi(\mathbf{X}_j)\|_2^2 \\ &= \frac{1}{mT} \sum_{j=1}^T \gamma^{j-1} \|e^{\mathcal{G}(t_j - t_0)} \psi(\mathbf{X}_0) - \psi(\mathbf{X}_j)\|_2^2. \end{aligned} \quad (17)$$

Here, the matrix exponential is the analytical solution to the linear dynamics, integrating initial conditions from time  $t_0$  to  $t_j$ . The sum in the latent space prediction loss accounts for future predictions over a time horizon  $T$  (Eq. (17)). It is worth noting that computing the prediction loss is computationally inexpensive. In practice, we can form a stack of time-scaled matrix exponentials (i.e.,  $e^{\mathcal{G}(t_j - t_0)}$ ) and multiply them by a batch of latent vectors  $\psi(\mathbf{X}_0)$  to predict a batch of future latent embeddings. This method is far more efficient than performing multi-step predictions for nonlinear systems due to its use of recurrence relationships.

In theory, if the terms presented so far are sufficiently minimized (Eq. (15)–(17)), future predictions in state space should also perform well. In fact, without the time-delay, these terms form the loss function for Physics Informed Koopman Networks (PIKNs) (Eq. (15)–(17))<sup>33</sup>. However, not explicitly enforcing prediction quality in state space can result in a shrinking latent space during optimization, where the prediction loss (Eq. (17)) could be minimized by allowing the encoder to output zero for every input. To avoid such shortcuts, we introduced an additional term that directly quantifies the quality of predictions in state space:

$$\mathcal{L}_{\mathbf{x}_t} = \frac{1}{nT} \sum_{j=1}^T \gamma^{j-1} \|\hat{\mathbf{x}}_j - \mathbf{x}_j\|_2^2 = \frac{1}{nT} \sum_{j=1}^T \gamma^{j-1} \|\psi^{-1}(e^{\mathcal{G}(t_j - t_0)} \psi(\mathbf{X}_0)) - \mathbf{x}_j\|_2^2. \quad (18)$$

We replaced the 2-norm in the state space prediction loss with the Mahalanobis distance with diagonal covariance for the Hodgkin-Huxley dataset to account for the scaling of the states. Lastly, we directly penalized the spectrum of  $\mathcal{G}$  to encourage the learning of stable or neutrally stable dynamics with the following term:

$$\mathcal{L}_\mu = \sum_{j=0}^m \max(0, \text{Re}(\mu_j)). \tag{19}$$

The total loss is then given by:

$$\mathcal{L} = \mathcal{L}_{x_0} + \alpha_1 \mathcal{L}_\psi + \alpha_2 \mathcal{L}_{\psi_t} + \alpha_3 \mathcal{L}_{x_t} + \alpha_4 \mathcal{L}_\mu. \tag{20}$$

To assess the impact of the additional loss terms (Eqs. (18), (19)) on the performance of the learned model and compare our approach to PIKNs, we conducted ablation experiments using simulated data from the magnetic mass-spring-damper system. We selected this system because of its demonstration of nonlinear behavior and global stability, implying that the learned eigenvalues should be negative. In these experiments, we assessed several metrics, including latent space prediction error, state space prediction error, reconstruction error, chain rule loss error, and the number and magnitude of positive eigenvalues (Fig. S4e–j).

In the first trials, we examined PIKNs, which struggled with predicting future states, assessing stability, and maintaining a stable latent space. In subsequent trials, we augmented the PIKN model with time-delayed inputs, which marginally improved both stability and state space prediction capabilities. We then introduced the state space prediction loss term alongside the time-lagged inputs, which significantly enhanced the accuracy of state space predictions by an order of magnitude compared to the original PIKN models. However, despite these improvements, the models still failed to accurately estimate stability, with a median of one positive eigenvalue per model. Ultimately, the introduction of the eigenvalue penalty (Eq. (19)) resulted in the best performance.

### Time-delay selection

Using time-delayed data to analyze or model a dynamical systems is an established approach. In fact, Taken’s theorem, first published in 1981, demonstrated that attractors can be reconstructed from time-delayed partial state measurements<sup>58</sup>. Since, numerous methods have been proposed for modelling and forecasting dynamics from time-delayed measurements including local polynomial methods<sup>49</sup>, time-delayed or Hankel DMD<sup>37,56,57,59,60</sup>, and deep learning-based models<sup>28,55,61</sup>.

Regardless of the method used, the length of time-delay is an important hyperparameter that significantly impacts model performance (Fig. S4a, b). Information theory has been suggested in previous work as a useful tool for selecting the length of time-delay<sup>47–50</sup>. However, much of this research is focused on attractor reconstruction as opposed to Koopman-informed modelling. Furthermore, few studies, if any, have explored the relationship between model performance and information-theoretic metrics. In this section, we examine the relationship between mutual information (MI) and future state prediction error. From this relationship, we propose a principled approach to selecting a lower bound on the length of time-delay for latent linear models.

The MI in bits between two discrete random variables  $A$  and  $B$  is given by:

$$I(A; B) = \sum_{a \in A} \sum_{b \in B} P_{(A,B)}(a, b) \log_2 \left( \frac{P_{(A,B)}(a, b)}{P_A(a)P_B(b)} \right) \tag{21}$$

where  $P_{(A, B)}$  is the joint probability mass function, and  $P_A, P_B$  are the marginal probability mass functions for  $A$  and  $B$ , respectively. In the context of collected trajectories from a dynamical system, we aim to calculate the MI between every, or many, combinations of states and their time-delayed counterparts. For example, the MI between a set of states at an arbitrary time

$x_a(t_j)$  and another state at a previous instant  $x_b(t_j - \tau_d)$  is expressed as:

$$I(x_a(t); x_b(t - \tau_d)) = \sum_{t_j} P(x_a(t_j), x_b(t_j - \tau_d)) \log_2 \left( \frac{P(x_a(t_j), x_b(t_j - \tau_d))}{P(x_a(t_j))P(x_b(t_j - \tau_d))} \right). \tag{22}$$

Naturally, if  $a = b$ , this measures the mutual information between a state at an arbitrary instant and a previous instant.

We calculated the mutual information for a single pendulum (Fig. S4c). The plot shows the mutual information vs. time delay ( $t - \tau_d$ ) for each combination of the states  $\theta$ , and  $\dot{\theta}$ . We observe a nearly periodic pattern in  $I$  as a function of the delay. Moreover, the mutual information between states and themselves is out of phase with mutual information between  $\theta$  and  $\dot{\theta}$ . Similarly, we calculated the mutual information between each combination of states in a double pendulum (Fig. S4d). Due to the chaotic dynamics, the mutual information did not exhibit a periodic pattern as a function of time delay, nor did the decrease in information occur monotonically. Interestingly, we observed an increase in mutual information between the angular positions and their velocities at approximately a half-second delay.

When we consider the prediction error as a function of delay alongside the time-delayed mutual information, we observe that predictions reach a point of diminishing returns when plotted against  $t - \tau_d$ . In the case of the single pendulum, this point coincides with one full-period in the mutual information signal, suggesting that for periodic systems, one full period of information should be included in the model. We also observed a relationship between the mutual information in the double pendulum and the predictive ability of the trained models. The prediction error appeared to plateau once each of the mutual information curves reached a local minimum. For instance, while the blue and grey mutual information curves reached a local minima at approximately  $t - \tau_d = 0.6s$ , the yellow and olive curves did not reach a minima until  $t - \tau_d = 1.25s$ , which coincided with the point of diminishing returns (Fig. S4d).

With these results in mind, we can use the mutual information between the states in a system to approximate the model’s predictive capability as a function of time delay. This provides us with an estimate for the lower bound of the time delay, and this process is significantly faster than performing hyperparameter search or running controlled experiments.

### Annealing strategy for the loss function and discount factor

Minimizing the total loss (Eq. (20)) presents several challenges. First, the loss landscape for a function with this many terms is likely filled with local extrema, making global optimization difficult. Second, the outcome of the optimization heavily depends on the values of the weights of each loss term,  $\alpha$ , and the discount factor  $\gamma$ . Many researchers address a weighted multi-parametric loss function with random search or hand-tuning. While this type of brute force search can still yield an adequate model, it does not improve training stability or model generalization and is often computationally expensive.

The practice of annealing, which entails systematically adjusting the hyperparameters during training, is widely recognized for improving training stability and enhancing model generalization. This technique is most frequently applied to the learning rate. For instance, Smith et.al. proposed oscillating learning rates in a periodic or quasi-periodic pattern to balance exploration and exploitation during training<sup>62,63</sup>. Recent work on variational autoencoders (VAEs) has shown that cyclically annealing the coefficients in the loss function can improve learning by mitigating vanishing of the Kullback-Liebler divergence<sup>64</sup>.

In accordance with Fu et.al., we set the  $\alpha$  coefficients in the loss function (Eq. (20)) with the following schedule:

$$\alpha^i = \begin{cases} g(k), & k \leq R \\ 1, & k > R \end{cases} \text{ with } k = \frac{\text{mod}(i-1, T_c/M)}{T_c/M}, \tag{23}$$

where  $\alpha^i$  represents the hyperparameter value at the  $i^{\text{th}}$  training iteration<sup>64</sup>. We define integers  $M$  as the number of cycles and  $T_c$  as the total number of training iterations. The parameter  $R \in [0, 1]$  indicates the proportion of each cycle during which  $\alpha$  remains at its maximum value of 1, and  $g$  is a function that increases monotonically from 0 to 1. For simplicity, we consider linear or sigmoid functions for  $g$ . Importantly, training invariably concludes with  $\alpha$  set to 1.

In this study, we categorize the annealing process into two types: when  $M = 1$ , we refer to it as monotonic or standard annealing, and when  $M > 1$ , we call it cyclical annealing. The schedule parameters  $M$  and  $R$  can vary across different coefficients in the loss function. Notably, we do not apply annealing to the reconstruction loss, as we regard reconstruction as fundamental to the model's effectiveness throughout training. Inadequate reconstruction quality would adversely impact the model's capability to make accurate future predictions in state space. To ensure accurate reconstruction, we warm-start or pre-train each network on the reconstruction loss alone. Examples of both cyclical and standard annealing schedules, using linear and sigmoid functions, demonstrate how they vary over the course of training iterations (Fig. S4).

In addition to the  $\alpha$  parameters in the loss function, the discount factor and prediction horizon have a disproportionate impact on model training and generalization performance. Many learning methods opt to optimize for single or few time-step predictions. This approach can be effective if the loss is sufficiently small, as the simple objective has a regularization effect. However, in many cases, these models are prone to underfitting, as we will show. On the other hand, optimizing for multiple time-step predictions comes with its own challenges, such as overfitting and exploding gradients. To address these issues, we use discount factor annealing as a method to balance time-horizon trade-offs and improve model performance.

While annealing the different terms in the loss function can be viewed as altering the importance of each term throughout training, discount factor annealing effectively changes the prediction horizon. Gradually increasing the discount factor over time acts as a form of temporal curriculum learning. In other words, the model must first learn to make short-term predictions before progressing to longer-term predictions.

To further motivate the need for discount factor annealing, we examined the model predictions for a pendulum, as depicted in (Fig. S4o–q). This figure contrasts three models that differ solely in their prediction horizon or discount factor. First, we compared the predicted and actual time series when using a model with a relatively short prediction horizon, where each prediction step is weighted equally with  $\gamma = 1$  (Fig. S4o). We observed that the predictions remain accurate up to, and slightly beyond, the training horizon, after which they deteriorate.

A straightforward strategy to improve the model's predictive accuracy might be to extend the prediction horizon. While extending the horizon does improve the model's predictive performance within the training horizon, its ability to generalize beyond this point is limited and nonphysical (Fig. S43p). Thus, training with longer prediction horizons may lead to overfitting. Finally, after implementing a cyclical annealing schedule for  $\gamma$ , we observe that the model is able to generalize well past the training horizon (Fig. S4q). These results also illustrate the importance of using a longer validation horizon than the training horizon during model selection. Even if a model performs well over many time-steps, it may still fail to generalize to extended horizons.

A high-level overview of the loss curves when employing linear monotonic annealing for the loss coefficients  $\alpha$ , and linear cyclical annealing for the discount factor provides insight into the training dynamics (Fig. S4e–l). These loss curves highlight the challenges associated with annealing the discount factor while training with extended prediction horizons. Notably, we observed a significant uptick in the loss metrics at the 5000th iteration (Fig. S4i–l). This increase in the loss aligns with the discount factor elevating to a value of 1, suggesting a correlation between the rate of change of the discount factor and a temporary regression of the model performance.

When we examined the relationship between the discount factor  $\gamma$  and its application over various future time steps  $\gamma^T$ , we uncovered distinct behaviors depending on the chosen annealing function (Fig. S4m–p). A linear annealing function showed a pronounced escalation in the value of  $\gamma^T$  as  $T$  increases, particularly noticeable at extended horizons. This change becomes starkly apparent even at relatively short horizons, such as  $T = 25$ , where the weighting applied to those predictions essentially becomes binary (Fig. S4n). This phenomenon is attributed to the fact that, under a linear framework, the effective horizon remains small until  $\gamma$  approaches 1, which is a consequence of the exponential decay inherent in the geometric series used for discounting. So, the impact of the annealing function on the effective prediction horizon and the overall training dynamics should not be underestimated.

To resolve the abrupt change in weighting at long prediction horizons observed with linear annealing, we adopted a sigmoid function for the main trials in this study. Using a sigmoid function ensured a more gradual adjustment in the weights assigned to time steps farther into the future compared to linear annealing (Fig. S4p). By incrementing the discount factor according to a sigmoid curve, we extended the effective prediction horizon in a more gradual and controlled manner, mitigating the binary weighting effect observed with linear annealing.

Lastly, we performed experiments to evaluate the efficacy of different annealing strategies. We compared monotonic and cyclical sigmoid annealing strategies against a baseline method that employed simple random search to determine the coefficients of the loss function and the discount factor. For each strategy—monotonic, cyclical, and baseline—we set the training and testing horizons to  $T = 400$  and  $T = 600$ , respectively. In the baseline approach, we sampled the discount factor from a uniform distribution with a range of 0.9 to 1.0, allowing for variations in the effective prediction horizon. As previously noted, we fixed the coefficient for reconstruction loss at 1, whereas we uniformly sampled the coefficients  $\alpha_i$  from a range of 0.01 to 1.0. The detailed hyperparameter settings for the annealing experiments are documented in Tables S4 and S5.

We observed significant enhancements in model performance with both cyclical and monotonic annealing when compared to the baseline method, particularly in forecasting accuracy (Fig. S4k–n). Moreover, we observed nearly two orders of magnitude improvement in both latent-space and state-space prediction error, with minimal to no compromise on  $\mathcal{L}_{x_0}$  and  $\mathcal{L}_y$ . We note a slight edge in performance for monotonic over cyclical. For this reason, we choose to employ monotonic for the models trained in this work. For especially long training horizons, we increase the time step for predictions to achieve extended forecasts without destabilizing training from a large  $T$ .

### Data collection and model training

We generated the simulated datasets in Python using the SciPy library and 4th order Runge-Kutta numerical integration. The number of trajectories we collected for each dataset, along with the trajectory length, size of the time-step, etc., is given in Tables S1–S5. We sampled initial conditions for the dissipative systems from a beta distribution to prevent bias in the datasets towards low-energy trajectories. The dataset for the magnetic-mass-spring-damper was balanced to contain an equal number of trajectories in each attractor. We trained the neural networks using Pytorch<sup>65</sup> and Lightning<sup>66</sup>. We used the AdamW optimizer and OneCycleLR for all of the models and performed a hyperparameter search with Optuna<sup>63,67,68</sup>. The various training and computational hardware details we used for each experiment are given in Tables S1–S4. We used a combination of the Moteus r4.11 controller and the Moteus mj5208 brushless motor as a servomotor to collect data for the experimental pendulum setups. We modeled the simulated systems according to the equations provided in the Supplementary Information.

### Data availability

All of the experimental data are available on Github: <https://github.com/generalroboticslab/AutomatedGlobalAnalysis>.

## Code availability

The code for this project are available on Github: <https://github.com/generalroboticslab/AutomatedGlobalAnalysis>.

Received: 11 May 2025; Accepted: 3 November 2025;

Published online: 17 December 2025

## References

- Strogatz, S. H. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering* (CRC press, 2018).
- Khalil, H. K. *Control of nonlinear systems* (Pearson, 2015).
- Jordan, D. & Smith, P. *Nonlinear ordinary differential equations: an introduction for scientists and engineers* (OUP Oxford, 2007).
- Koopman, B. O. Hamiltonian systems and transformation in hilbert space. *Proc. Natl. Acad. Sci.* **17**, 315–318 (1931).
- Hespanha, J. P. *Linear systems theory* (Princeton university press, 2018).
- Brunton, S. L., Budišić, M., Kaiser, E. & Kutz, J. N. Modern koopman theory for dynamical systems. arXiv preprint arXiv:2102.12086 (2021).
- Borrelli, F., Bemporad, A. & Morari, M. *Predictive control for linear and hybrid systems* (Cambridge University Press, 2017).
- Brunton, S. L., Brunton, B. W., Proctor, J. L. & Kutz, J. N. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS one* **11**, e0150171 (2016).
- Mauroy, A. & Mezić, I. Global stability analysis using the eigenfunctions of the koopman operator. *IEEE Trans. Autom. Control* **61**, 3356–3369 (2016).
- Mauroy, A., Mezić, I. & Moehlis, J. Isostables, isochrons, and koopman spectrum for the action–angle representation of stable fixed point dynamics. *Phys. D: Nonlinear Phenom.* **261**, 19–30 (2013).
- Deka, S. A., Valle, A. M. & Tomlin, C. J. Koopman-based neural lyapunov functions for general attractors. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 5123–5128 (IEEE, 2022).
- Mezić, I. Koopman operator, geometry, and learning of dynamical systems. *Not. Am. Math. Soc.* **68**, 1087–1105 (2021).
- Vidal, R., Ma, Y. & Sastry, S. Generalized principal component analysis (gpca). *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1945–1959 (2005).
- Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**, 2323–2326 (2000).
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
- Levina, E. & Bickel, P. Maximum likelihood estimation of intrinsic dimension. *Adv. Neural Inf. Process. Syst.* **17** (2004).
- Camastra, F. & Staiano, A. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.* **328**, 26–41 (2016).
- Schmid, P. J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28 (2010).
- Williams, M. O., Kevrekidis, I. G. & Rowley, C. W. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.* **25**, 1307–1346 (2015).
- Tschannen, M., Bachem, O. & Lucic, M. Recent advances in autoencoder-based representation learning. arXiv preprint arXiv:1812.05069 (2018).
- Hafner, D. et al. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565 (PMLR, 2019).
- Cranmer, M. et al. Lagrangian neural networks. arXiv preprint arXiv:2003.04630 (2020).
- Chen, R. T., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. Neural ordinary differential equations. *Adv. Neural Inf. Process. Syst.* **31** (2018). (10.1038/s44260-025-00062-y)
- Ansuini, A., Laio, A., Macke, J. H. & Zoccolan, D. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems* **32** (2019).
- Chen, B. et al. Automated discovery of fundamental variables hidden in experimental data. *Nat. Comput. Sci.* **2**, 433–442 (2022).
- Huang, K., Cho, D. H. & Chen, B. Automated discovery of continuous dynamics from videos. arXiv preprint arXiv:2410.11894 (2024).
- Lusch, B., Kutz, J. N. & Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 4950 (2018).
- Takeishi, N., Kawahara, Y. & Yairi, T. Learning koopman invariant subspaces for dynamic mode decomposition. *Adv. Neural Inf. Process. Syst.* **30** (2017).
- Han, Y., Hao, W. & Vaidya, U. Deep learning of koopman representation for control. In *2020 59th IEEE Conference on Decision and Control (CDC)*, 1890–1895 (IEEE, 2020).
- Yeung, E., Kundu, S. & Hodas, N. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, 4832–4839 (IEEE, 2019).
- Shi, H. & Meng, M. Q.-H. Deep koopman operator with control for nonlinear systems. *IEEE Robot. Autom. Lett.* **7**, 7700–7707 (2022).
- Morton, J., Witherden, F. D. & Kochenderfer, M. J. Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. arXiv preprint arXiv:1902.09742 (2019).
- Liu, Y., Sholokhov, A., Mansour, H. & Nabi, S. Physics-informed koopman network. arXiv preprint arXiv:2211.09419 (2022).
- Peitz, S., Otto, S. E. & Rowley, C. W. Data-driven model predictive control using interpolated koopman generators. *SIAM J. Appl. Dyn. Syst.* **19**, 2162–2193 (2020).
- Iacob, L. C., Beintema, G. I., Schoukens, M. & Tóth, R. Deep identification of nonlinear systems in koopman form. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 2288–2293 (IEEE, 2021).
- Korda, M. & Mezić, I. Optimal construction of koopman eigenfunctions for prediction and control. *IEEE Trans. Autom. Control* **65**, 5114–5129 (2020).
- Kamb, M., Kaiser, E., Brunton, S. L. & Kutz, J. N. Time-delay observables for koopman: Theory and applications. *SIAM J. Appl. Dyn. Syst.* **19**, 886–917 (2020).
- Kaiser, E., Kutz, J. N. & Brunton, S. L. Data-driven discovery of koopman eigenfunctions for control. *Mach. Learn.: Sci. Technol.* **2**, 035023 (2021).
- Folkestad, C. et al. Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control. In *2020 American Control Conference (ACC)*, 3906–3913 (IEEE, 2020).
- Haseli, M. & Cortés, J. Learning koopman eigenfunctions and invariant subspaces from data: Symmetric subspace decomposition. *IEEE Trans. Autom. Control* **67**, 3442–3457 (2021).
- Van der Pol, B. Lxxxviii. on “relaxation-oscillations”. *Lond., Edinb., Dublin Philos. Mag. J. Sci.* **2**, 978–992 (1926).
- Van der Pol, B. & Van Der Mark, J. Frequency demultiplication. *Nature* **120**, 363–364 (1927).
- Hodgkin, A. L. & Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **117**, 500 (1952).
- Lorenz, E. N. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, vol. 1 (Reading, 1996).
- Duffing, G. *Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung*. (Vieweg, 1918).
- Wang, X.-S., Moore, S. A., Turner, J. D. & Mann, B. P. A model-free sampling method for basins of attraction using hybrid active learning (hal). *Commun. Nonlinear Sci. Numer. Simul.* **112**, 106551 (2022).
- Fraser, A. M. & Swinney, H. L. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A* **33**, 1134 (1986).

48. Fraser, A. M. Information and entropy in strange attractors. *IEEE Trans. Inf. Theory* **35**, 245–262 (1989).
49. Abarbanel, H. D., Carroll, T., Pecora, L., Sidorowich, J. & Tsimring, L. S. Predicting physical variables in time-delay embedding. *Phys. Rev. E* **49**, 1840 (1994).
50. Abarbanel, H. Analysis of observed chaotic data (Springer Science & Business Media, 2012).
51. Otto, S. E. & Rowley, C. W. Koopman operators for estimation and control of dynamical systems. *Annu. Rev. Control, Robot., Autonomous Syst.* **4**, 59–87 (2021).
52. Mezić, I. Spectrum of the koopman operator, spectral expansions in functional spaces, and state-space geometry. *J. Nonlinear Sci.* **30**, 2091–2145 (2020).
53. Mauroy, A., Susuki, Y. & Mezić, I. Koopman operator in systems and control, vol. 7 (Springer, 2020).
54. Mezić, I. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dyn.* **41**, 309–325 (2005).
55. Qian, S., Chou, C.-A. & Li, J.-S. Deep multi-modal learning for joint linear representation of nonlinear dynamical systems. *Sci. Rep.* **12**, 12807 (2022).
56. Yuan, Y., Zhou, K., Zhou, W., Wen, X. & Liu, Y. Flow prediction using dynamic mode decomposition with time-delay embedding based on local measurement. *Phys. Fluids* **33** (2021).
57. Le Clainche, S. & Vega, J. M. Higher order dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* **16**, 882–925 (2017).
58. Takens, F. Detecting strange attractors in turbulence. *Lect. Notes Math., Berl. Springe. Verl.* **898**, 366 (1981).
59. Arbabi, H. & Mezić, I. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM J. Appl. Dyn. Syst.* **16**, 2096–2126 (2017).
60. Brunton, S. L., Brunton, B. W., Proctor, J. L., Kaiser, E. & Kutz, J. N. Chaos as an intermittently forced linear system. *Nat. Commun.* **8**, 19 (2017).
61. Bakarji, J., Champion, K., Kutz, J. N. & Brunton, S. L. Discovering governing equations from partial measurements with deep delay autoencoders. arXiv preprint arXiv:2201.05136 (2022).
62. Smith, L. N. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference On Applications Of Computer Vision (WACV)*, 464–472 (IEEE, 2017).
63. Smith, L. N. & Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence And Machine Learning For Multi-domain Operations Applications*, vol. 11006, 369–386 (SPIE, 2019).
64. Fu, H. et al. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. arXiv preprint arXiv:1903.10145 (2019).
65. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019).
66. Team, P. L. Pytorch lightning. GitHub 3 (2019).
67. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017).
68. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631 (2019).

### Acknowledgements

This work was supported by the NSF Graduate Research Fellowship, the ARL STRONG program under awards W911NF2320182 and W911NF2220113, by ARO W911NF2410405, by DARPA FoundSci program under award HR00112490372, DARPA TIAMAT program under award HR00112490419, and NSF award number 2053470.

### Author contributions

S.M. led the conceptualization, methodology development, implementation, experiments, and writing of the original manuscript. B.M and B.C provided guidance on research and experiment design, interpretation of results, manuscript review and editing, and funding acquisition.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s44260-025-00062-y>.

**Correspondence** and requests for materials should be addressed to Samuel A. Moore or Boyuan Chen.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025