nature computational science

ARTICLES

https://doi.org/10.1038/s43588-022-00281-6

Check for updates

Automated discovery of fundamental variables hidden in experimental data

Boyuan Chen[1]✉, Kuang Huang[2], Sunand Raghupathi[2], Ishaan Chandratreya[1], Qiang Du[2,3] and Hod Lipson[3,4]

All physical laws are described as mathematical relationships between state variables. These variables give a complete and non-redundant description of the relevant system. However, despite the prevalence of computing power and artificial intelligence, the process of identifying the hidden state variables themselves has resisted automation. Most data-driven methods for modelling physical phenomena still rely on the assumption that the relevant state variables are already known. A longstanding question is whether it is possible to identify state variables from only high-dimensional observational data. Here we propose a principle for determining how many state variables an observed system is likely to have, and what these variables might be. We demonstrate the effectiveness of this approach using video recordings of a variety of physical dynamical systems, ranging from elastic double pendulums to fire flames. Without any prior knowledge of the underlying physics, our algorithm discovers the intrinsic dimension of the observed dynamics and identifies candidate sets of state variables.

Mathematical relationships are known to describe nearly all physical laws in nature[1], and these mathematical expressions are almost always formulated as relationships between physical state variables that describe the physical system. This suggests that, before any natural law can be discovered, the relevant state variables must first be identified[2,3].

For example, it took civilizations millennia to formalize basic mechanical variables such as mass, momentum and acceleration. Only once these notions were formalized could laws of mechanical motion be discovered. Laws of thermodynamics were discovered only after concepts such as temperature, pressure, energy and entropy were formalized. Laws of solid mechanics could only be discovered once variables such as stress and strain were formalized. Electromagnetism, fluid dynamics, quantum mechanics and so forth all required their own set of fundamental state variables to be defined before they could be formalized into existence. Without the proper state variables, even a simple system may appear enigmatically complex.

The set of state variables for modelling any system is not only hidden, but also not unique (Extended Data Fig. 1). In fact, even for well studied systems in classical mechanics, such as a swinging pendulum, many sets of possible state variables exist. For the pendulum, the state variables are typically the angle of the arm $q_1 = \theta$ and the angular velocity of the arm $q_2 = \dot{\theta}$. The angle and angular velocity are convenient choices because they can be directly measured. However, alternative sets of state variables, such as kinetic and potential energies of the arm, could also be used as state variables.

A key challenge, however, occurs when the system is new, unfamiliar or complex, and the relevant set of state variables is unknown. Although various techniques such as dynamic mode decomposition and singular value decomposition[4] have been developed to learn dynamical systems on the basis of observations, none of these methods has the ability to process a video of a pendulum, for example, and without any further knowledge output the double pendulum's four state variables. Such an ability, if possessed, could help scientists gain insight into the physics underlying increasingly complex phenomena, especially when theory is not keeping pace with observations.

Data-analytics tools have impacted almost every aspect of scientific discovery[5,6]: machines can measure, collect, store and analyse vast numbers of data. New machine learning techniques can create predictive models, find analytical equations[7] and invariants[8], and even generate causal hypotheses along with new experiments to validate or refute these hypotheses[9–11]. Yet, a longstanding question is whether it is possible to automatically uncover the hidden state variables themselves. Finding such variables is still a laborious process requiring teams of human scientists toiling over decades.

The ability of human scientists to distil vast streams of raw observations into laws governing a concise set of relevant state variables has played a key role in many scientific discoveries. It is thus of great importance to have tools for automated scientific discovery that could help distil raw sensory perceptions into a compact set of state variables and their relationships.

Numerous machine learning tools have been demonstrated to model the dynamics of physical systems automatically, but most of them were already provided with measurements of the relevant state variables in advance[7,8,12–26]. In this Article, by state variables we refer to compact and complete sets of quantitative variables that fully describe the observed dynamical system evolving with time. For example, our own previous work on distilling natural laws[8] assumed an input stream corresponding to state variables such as angle and angular velocity of a pendulum arm. Brunton et al.[24] required access to spatial coordinates and their derivatives for modelling a Lorenz system, Udrescu and Tegmark[26] combined neural networks with known physical properties to solve equations from the Feynman Lecture on Physics, given provided variables, Mrowca et al.[27] required access to the position, velocity, mass and material properties of the particles that compose the objects being modelled and Champion et al.[28] predefined possible basis functions to constrain the training of an autoencoder for observation reconstruction.

[1]Department of Computer Science, Columbia University, New York, USA. [2]Department of Applied Physics and Applied Mathematics, Columbia University, New York, USA. [3]Data Science Institute, Columbia University, New York, USA. [4]Department of Mechanical Engineering, Columbia University, New York, USA. ✉e-mail: bchen@cs.columbia.edu

The goal of this work is to find a way to overcome this key barrier to automated discovery—by explicitly identifying the intrinsic dimensionality of a system and the corresponding hidden state variables, purely from the visual information encoded in raw camera observations. A key challenge in identifying state variables is that they are often hidden and might be hard to measure directly. An even more challenging aspect of state variable identification is that there might be a large number of potential descriptive variables that are related to the varying state of the system, but are neither compact nor complete in their description of the system.

For example, a camera observing a swinging pendulum with an imaging resolution of 128 × 128 pixels in three colour channels will measure 49,152 variables per frame. Yet this enormous set of measurement, while intuitively descriptive, is neither compact nor complete: in fact, we know that the state of a swinging pendulum can be described fully by only two variables: its angle and angular velocity. Moreover these two state variables cannot be measured from a single video frame alone. In other words, a single frame, despite the large number of measurements, is insufficient to describe the full state of a pendulum.

The questions that we aim to answer are whether, given a series of video frames of a swinging pendulum that contain the full and accurate motion trajectories, for example, there is a way to know that only two variables are required to describe its dynamics in full, and whether there is an automated process to reduce the vast deluge of irrelevant and superfluous pixel information into representations in terms of the two state variables. Naturally, we would like this process to work across a variety of physical systems and phenomena.

The starting point of our approach is to model the system dynamics directly from video representations via a neural network with bottleneck latent embeddings[29–31]. If the network is able to make accurate future predictions, the network should internally encapsulate a relationship connecting relevant current states with future states. Our main challenge is to distil and extract the hidden state variables from the network encoding.

Our approach involves two major stages. First, after training the dynamics predictive neural network, we calculate the minimum number of independent variables needed to describe the dynamical systems, known as its intrinsic dimension (ID), with geometric manifold learning algorithms. This initial stage produces accurate ID estimations on a variety of systems from the model's bottleneck latent embeddings which are already reduced by hundreds of times compared with the raw image space.

Armed with the ID obtained in the first stage, in the second stage we design a latent reconstruction neural network to further identify the governing state variables with the exact dimension of the ID. We term these identified state variables neural state variables. Through both quantitative and qualitative experiments, we demonstrate that neural state variables can accurately capture the overall system dynamics.

Beyond our two-stage approach to reveal the system ID and the possible set of state variables, another major contribution of our work is to leverage the discovered neural state variables as both an intermediate representation and an evaluation metric for stable long-term future predictions of system behaviours. Due to the special reduced-dimension property of neural state variables, they can provide very stable long-term predictions, while higher-dimensional autoencoders often yield blurred or plain background predictions if iterated just a few steps into the future.

Finally, we present a hybrid prediction scheme that achieves both accurate and stable long-term predictions. Furthermore, we derive a quantitative evaluation metric for long-term prediction stability with neural state variables by approximating the true system dynamics using the most compact latent space. Additionally, we also demonstrate that neural state variables can offer a robust representation space for modelling system dynamics under various visual perturbations.

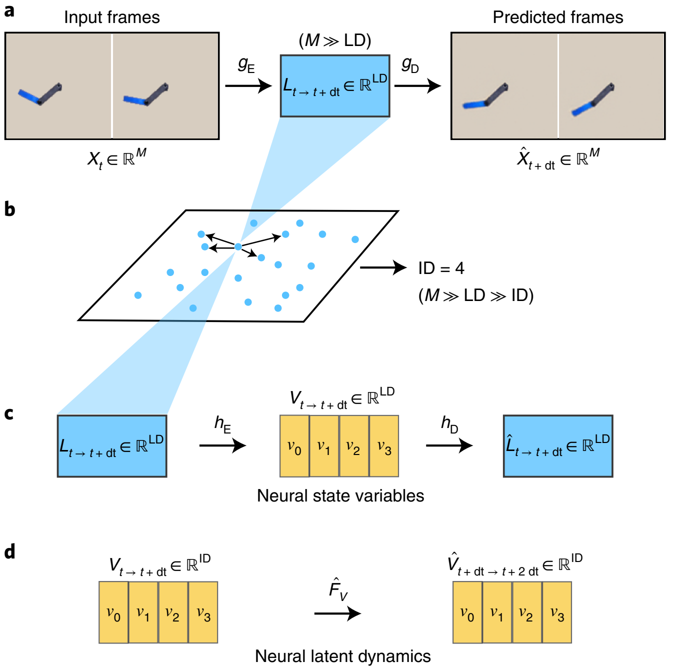

a
Input frames
Predicted frames
$(M \gg \mathrm{LD})$
g_E
g_D
$L_{t\rightarrow t+\mathrm{dt}} \in \mathbb{R}^{\mathrm{LD}}$
$X_t \in \mathbb{R}^M$
$\hat{X}_{t+\mathrm{dt}} \in \mathbb{R}^M$
b
ID = 4
$(M \gg \mathrm{LD} \gg \mathrm{ID})$
c
$L_{t\rightarrow t+\mathrm{dt}} \in \mathbb{R}^{\mathrm{LD}}$
h_E
$V_{t\rightarrow t+\mathrm{dt}} \in \mathbb{R}^{\mathrm{LD}}$
v_0 v_1 v_2 v_3
h_D
$\hat{L}_{t\rightarrow t+\mathrm{dt}} \in \mathbb{R}^{\mathrm{LD}}$
Neural state variables
d
$V_{t\rightarrow t+\mathrm{dt}} \in \mathbb{R}^{\mathrm{ID}}$
v_0 v_1 v_2 v_3
$\hat{F}_V$
$\hat{V}_{t+\mathrm{dt}\rightarrow t+2\,\mathrm{dt}} \in \mathbb{R}^{\mathrm{ID}}$
v_0 v_1 v_2 v_3
Neural latent dynamics

Fig. 1 | Two-stage modelling of dynamical systems. a,**b**, First stage: ID estimation. We first modelled the dynamical systems via the evolution from $\boldsymbol{Xt}$ to $\boldsymbol{X}_{t+dt}$ with a fully convolutional encoder–decoder network directly from video observations (**a**). $\boldsymbol{X}_t \in \mathbb{R}^M$ represents the input video frames with dimension M. The dimension of the latent embedding $\boldsymbol{L}_{t\rightarrow t+dt}$ is defined as LD. We then applied a geometric manifold learning algorithm on the latent embedding to identify the ID (**b**). **c**, Second stage: discover neural state variables. We applied another encoder–decoder network on top of the above latent vectors to automatically determine the neural state variables, denoted as $\boldsymbol{V}_{t\rightarrow t+dt}$, by limiting the latent dimension of this network to the identified ID. h_E is the encoder and h_D is the decoder of this second network. The objective of this network is to predict the reconstructed latent embedding denoted as $\hat{\boldsymbol{L}}_{t\rightarrow t+dt}$. **d**, Once we determine the neural state variables, we can leverage the system dynamics in the space of neural state variables as an indicator of dynamics stability. Here we approximate the latent system dynamics through a neural network $\hat{F}_V$ to predict the future neural state variables denoted as $\hat{\boldsymbol{V}}_t + \mathrm{d}t \rightarrow t + 2\,\mathrm{d}t$.

Results

Modelling dynamical systems from videos. The dynamics of a physical system defines the rule that governs how the current system states will evolve into their successive states in the future. Mathematically, provided the ambient space $\mathcal{X}$ and the state space $\mathcal{S} \subset \mathcal{X}$, one can formulate the dynamical system as

$$\boldsymbol{X}_{t+\mathrm{d}t} = F(\boldsymbol{X}_t), \quad t = 0, \mathrm{d}t, 2\,\mathrm{d}t, 3\,\mathrm{d}t, \ldots, \tag{1}$$

where $\boldsymbol{X}_t \in \mathcal{S}$ is the system's current state at time t and dt is the discrete time increment. $F : \mathcal{S} \rightarrow \mathcal{S}$ describes the state evolution from $\boldsymbol{X}_t$ to the system's successive state $\boldsymbol{X}_{t+\mathrm{d}t}$ at time $t+\mathrm{d}t$. Throughout this paper we will consider the system as discrete in time. Any dynamical system continuous in time can also be discretized to formulation (1) with an appropriate dt.

Our goal is to learn the most compact space that implicitly captures the entire system dynamics from only high-dimensional visual data. To achieve this, we formulate a self-supervised learning problem to leverage the natural supervision from future video streams.

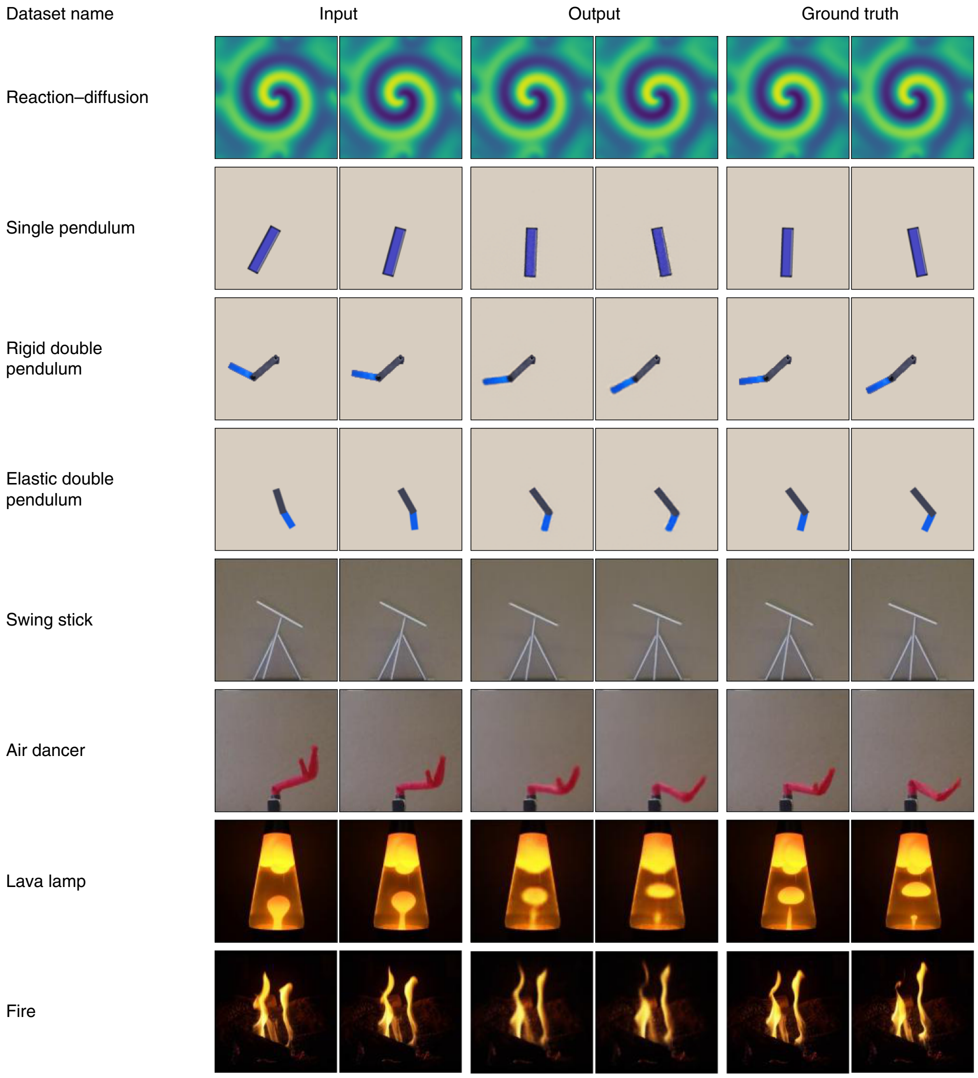


Fig. 2 | Prediction visualizations and physics evaluations. Visualizations of our basic prediction results across multiple dynamical systems with their input frames and the corresponding ground-truth frames. For systems where physical variables happen to be available, we performed physics evaluations on these systems in Supplementary Section 4.

Therefore, $\mathcal{X}$ is the high-dimensional image space. Our model is based on an autoencoder neural network to map high-dimensional visual observations to a relatively low-dimensional embedding, which will then be projected to future video frames. Formally, our framework comprises five major components, as shown in Fig. 1a: a pair of input image frames $\boldsymbol{X}_t$, an encoder network g_{E}, a latent embedding vector $\boldsymbol{L}_{t\to t+\mathrm{d}t}$, a decoder network g_{D} and a pair of output future frames $\boldsymbol{X}_{t+\mathrm{d}t}$. In Methods, we give more information on our choice of state representations and their advantages.

First, to study the generality of the proposed approach, we compiled a dataset comprising video recordings of nine physical dynamical systems from various experimental domains (Extended Data Fig. 1), ranging from simple periodic motion (circular motion, single pendulum), chaotic kinematics (rigid double pendulum, elastic double pendulum, swing stick), nonlinear wave (reaction–diffusion system) and multiphase flow (lava lamp) to aeroelasticity (air dancer) and combustion (flame dynamics). We include full details of the dataset in Supplementary Section 1.

In Fig. 2, we show comparisons between the predicted video frames and the ground-truth recordings. Our model was able to produce accurate video predictions. Our model also substantially outperforms linear extrapolation and copying input data baselines as shown in Supplementary Section 4 suggesting that our model captures non-trivial understanding of the system's second-order dynamics. For a dataset with ground-truth physical quantities such as an elastic double pendulum, our system was able to predict the physical variables accurately compared with the ground truth. For more intuitive and quantitative understanding of the physics evaluation results, we provide more statistics of the evaluation in Supplementary Section 2. Overall, the evaluation results suggest that the model successfully captured a non-trivial understanding of the system dynamics.

ID estimation. ID has served as a fundamental concept in many advances in physical sciences. In general, the ID refers to the minimum number of independent variables needed to fully describe the state of a dynamical system. The ID is independent of specific representations of the system or choice of a particular set of state variables. In a more quantitative way, the ID could be equivalently defined as the topological dimension of the state space $\mathcal{S}$ as a manifold in the ambient space $\mathcal{X}$ (refs. [32–34]).

A common assumption when analysing a physical system is that the ID is known a priori. An even stronger assumption is that the corresponding state variables themselves are given. Yet these

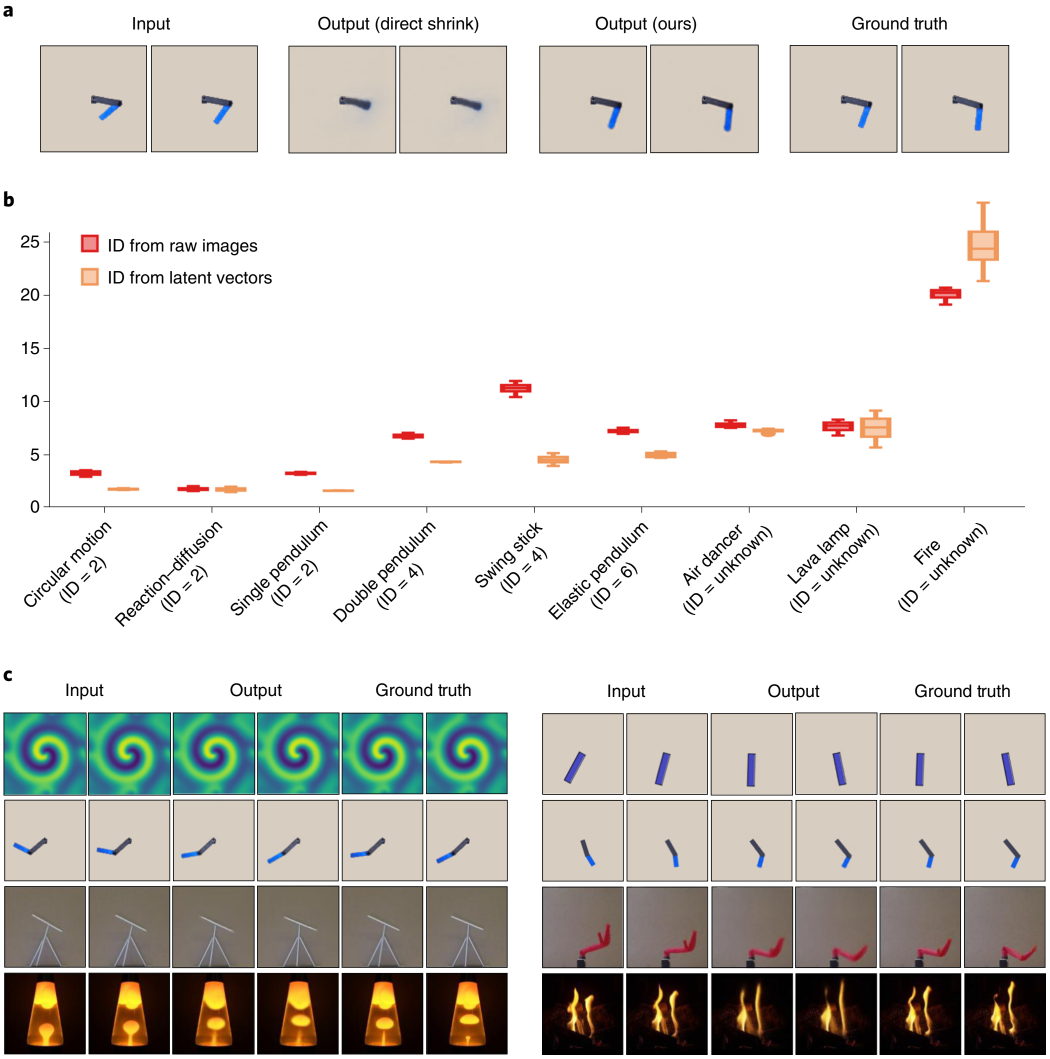


Fig. 3 | ID and neural state variables. a, The first column is the input frames to the predictive models. The second column shows the unsatisfactory prediction by directly reducing the size of the latent embedding on the original autoencoder, and the third column shows the accurate predictions produced by our two-stage method. The fourth column gives the ground-truth future frames. **b**, The ID estimation results. Our method estimates ID without prior knowledge about the systems' state variables as shown in orange box plots. We calculate the estimated ID values over 15 groups of random samples using models trained with three random seeds for each system. The box is drawn from the first quartile Q_1 to the third quartile Q_3 with a horizontal line representing the median. The lower and upper whiskers represent the data range within the interval $[Q_1-1.5(Q_3-Q_1), Q_3+1.5(Q_3-Q_1)]$. Our method outperforms direct estimations from raw images as in red box plots. The ground-truth IDs for known systems are included in the parentheses of each label of the *x* axis. We include the exact numbers in Supplementary Section 5. **c**, By making predictions through the space of our discovered neural state variables, we show the one-step prediction results for two systems. On each side, the first column shows the input to the network, and the second and third columns show the output and ground truth of the prediction results.

assumptions do not hold for unknown or partially known systems. To uncover the underlying dynamics of a wide range of systems and make future predictions of their future behaviours, we need to automatically identify the ID of the systems and extract the corresponding state variables from observed data, which is often high dimensional and noisy.

A naive approach using an autoencoder predictive framework is to keep reducing the size of the latent embedding vector through trial and error until the output is no longer valid. However, this approach does not yield satisfactory results because the output deteriorates long before the minimal set of state variables is reached. As shown in Fig. 3a, the model predictions broke down when we directly shrank the size of the latent space to the ID.

Inspired by traditional manifold learning methods that utilize geometric structures of the embedding vectors (such as their nearest distances), we propose a solution that can automatically discover the ID of a dynamical system from the latent vectors. Our approach only needs a one-time network training step. Specifically, we applied the Levina–Bickel algorithm[35] on the latent embedding space. We discuss more algorithm details in Methods.

Figure 3b shows the estimations across all the systems in our holdout dataset along with baseline comparisons from raw image observations and partial ground truths. The error bar represents the s.d. of the estimated IDs. We include the exact numbers in a table in Supplementary Section 5. The estimated ID values are rounded to the nearest even integer, as position and velocity variables are in pairs in our systems. In practice, it may be necessary to explore some nearby values on the basis of our algorithm's results. Our method demonstrates highly accurate estimations of the ID of all known systems. Although we cannot account for the ground-truth ID of

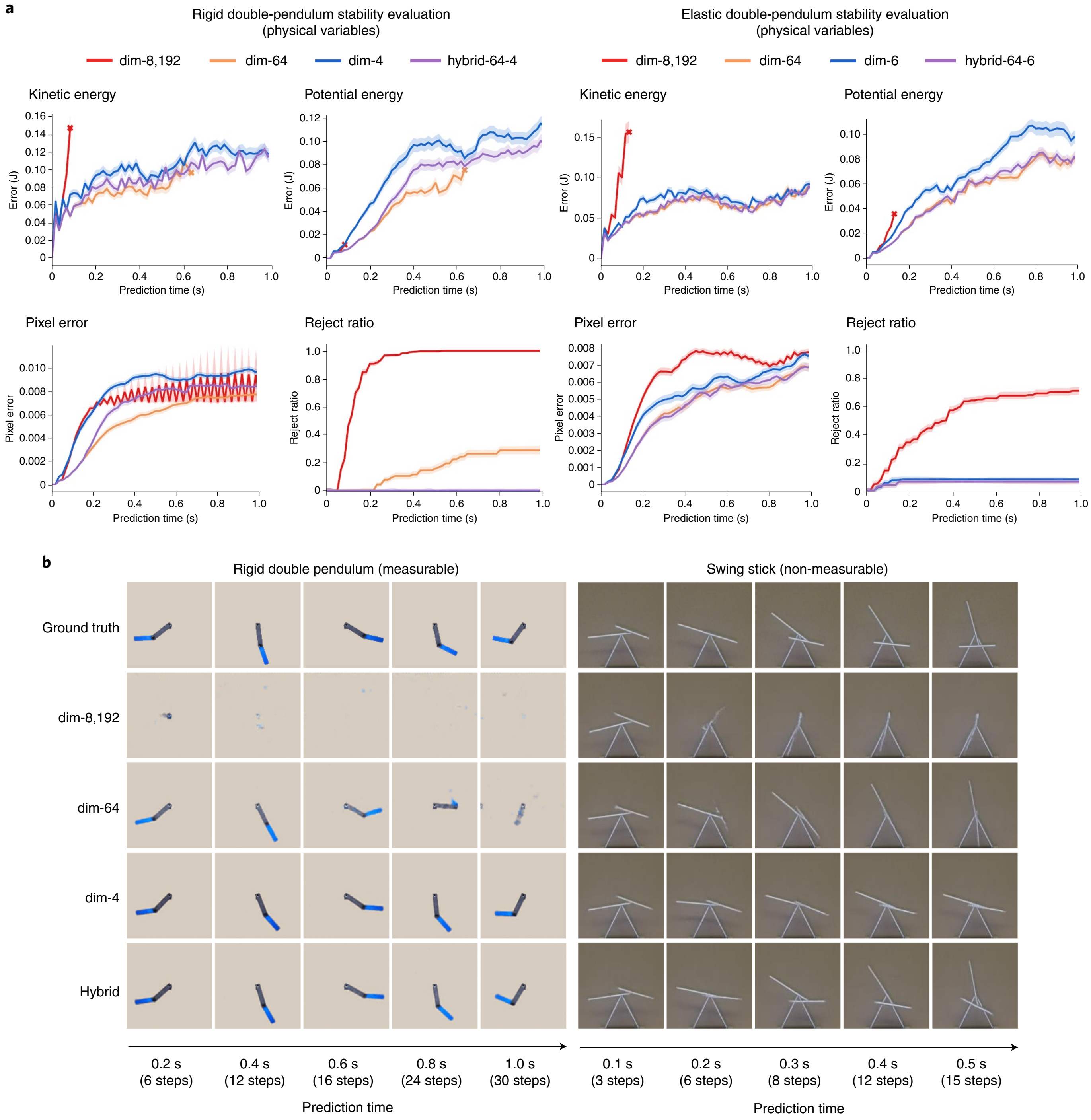
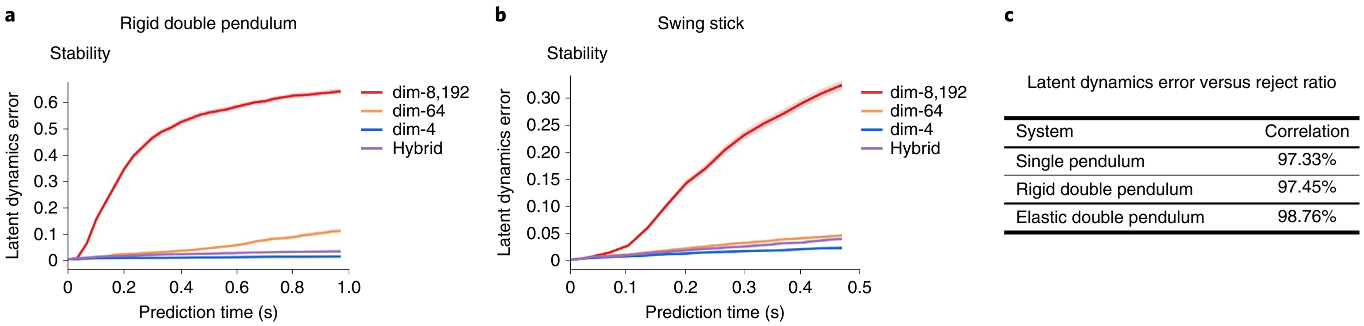


Fig. 4 | Long-term prediction stability. a, Long-term prediction stability evaluation through multiple prediction schemes and evaluation metrics, given measurable physical variables shown on the *y* axis. dim-8,192 and dim-64 are prediction schemes directly through relatively high-dimensional latent space with dimensions 8,192 and 64 respectively. dim-4 and dim-6 are prediction schemes through the space of neural state variables. hybrid-64-4 and hybrid-64-6 are hybrid schemes that iterate between the 64-dimensional latent space and our compact neural state variable space. s.e.m. values are reflected as the shaded regions. Each plot is performed on all the test data with models trained on three random seeds. **b**, Each row shows visualizations of predicted frames under different prediction schemes. The *x* axis represents the future prediction time.

other systems, we do see that our experiments present a reasonable and intuitive relative ranking among all listed systems.

We also compared the performance of the Levina–Bickel algorithm with other popular intrinsic dimensionality estimation algorithms including MiND_ML, MiND_KL, Hein and CD[34,36–39] by following the original implementations[37,38]. We present full evaluations in Supplementary Section 5. Though all the algorithms demonstrated promising performance, we found that the Levina–Bickel algorithm gives the most robust and reliable estimation.

Neural state variables. As we have discussed above, the minimum set of independent state variables $\boldsymbol{V}$ used to describe the dynamical system has the dimension known as the ID, namely $\boldsymbol{V} \in \mathbb{R}^{\mathrm{ID}}$.



System	Correlation
Single pendulum	97.33%
Rigid double pendulum	97.45%
Elastic double pendulum	98.76%

Fig. 5 | Neural state variables for dynamics stability indicators. a,**b**, The effective results of using the latent dynamics on the space of neural state variables, namely neural latent dynamics, as an indicator to quantify the stability of long-term future predictions for the rigid double pendulum (**a**) and the swing stick (**b**). The *y* axis measures the latent dynamics error and the *x* axis represents the future prediction time. s.e.m. values of the latent dynamics errors are reflected as the shaded regions. dim-8,192 and dim-64 are prediction schemes directly through relatively high-dimensional latent space with dimensions 8,192 and 64 respectively. dim-4 is our prediction scheme through the space of neural state variables. The hybrid scheme iterates between the 64-dimensional latent space and our compact neural state variable space. **c**, The strong Pearson correlation results between the latent dynamics error and the physics reject ratio.

To simplify the terminology, we refer them as state variables directly throughout the rest of this paper.

Now that we have identified the number of state variables, we need to find the actual variables themselves (bearing in mind that the set is not unique). We propose a two-stage framework to retrieve possible state variables as shown in Fig. 1a–c. We term our subset of state variables as neural state variables. Hence, with neural state variables $\boldsymbol{V}$, the dynamical system can be expressed as the evolution of the trajectory $\{\boldsymbol{V}_{0\to\mathrm{d}t}, \boldsymbol{V}_{\mathrm{d}t\to 2\,\mathrm{d}t}, \ldots\}$.

The first stage is to identify ID as discussed previously. This stage yields a relatively low-dimensional latent embedding $\boldsymbol{L} \in \mathbb{R}^{\mathrm{LD}}$, where LD is the dimension of this latent embedding. However, LD is still much larger than ID. The second stage operates directly on the latent embedding to further distil the neural state variables. We trained a second autoencoder network to take in the pretrained latent embedding and output the reconstruction of the input. The special property of this network is that the size of the latent embedding is equal to the ID obtained from the first step. With a minimum reconstruction error, we can identify this latent embedding vector as the neural state variables.

Overall, our two-stage method bypasses the optimization challenges and avoids the risk of underfitting the observed data. In Fig. 3c, we qualitatively demonstrate the effectiveness of our approach. For all the systems in our dataset, our framework is able to predict accurate future frames from supercompact variables with dimension ID (for example, ID = 4 for rigid double pendulum and ID = 6 for elastic double pendulum).

Neural state variables for stable long-term prediction. Forecasting the long-term future behaviours of unknown physical systems by learning to model their dynamics is critical for numerous real-world tasks. With a dynamics predictive model giving the one-step prediction, we can perform model rollout to feed each step's prediction as the input to predict the next state. However, there are two main challenges to obtaining satisfactory long-term predictions.

- Non-iterative one-step prediction accuracy. The learned dynamics may not be accurate since prediction errors are iteratively introduced at every prediction step. This issue is mainly attributable to the one-step prediction accuracy.
- Long-term prediction stability. Due to iterative error accumulation, the predicted sequences may not be able to maintain the ground-truth state space: one repeated observation from past studies is that the long-term predicted sequences become blurred, heavily distorted or plain background within only a few rollouts. We also observed similar phenomena in our experiments as shown in Fig. 3a. This is a very important issue to resolve because if objects deform or entirely disappear without following the system dynamics it will be impossible to follow the system evolution faithfully.

Long-term prediction stability refers to the deviation between the predicted sequences generated from the learned dynamics and the ground-truth state space governed by the system dynamics. Given a metric $M_{\mathcal{S}}(\cdot)$ that measures the deviation of a predicted state from the true state space $\mathcal{S}$, and a prediction sequence $\{\hat{\boldsymbol{X}}_0, \hat{\boldsymbol{X}}_{\mathrm{d}t}, \ldots\}$ from any initial state $\hat{\boldsymbol{X}}_0$, we can quantify the stability of a prediction scheme as the growth rate of $M_{\mathcal{S}}(\hat{\boldsymbol{X}}_t)$ as a function of *t*.

One challenge is to define at what point the predicted image becomes so degraded that it does not count as a prediction at all. We define an image quality test as follows (used only for evaluation): for systems for which we have prior knowledge about their conventional state variables, and we can extract these physical variables from the corresponding videos through classic computer vision techniques (for example, colour and contour extraction), $M_{\mathcal{S}}^{\mathrm{phys}}(\cdot)$ can be readily defined as a binary value indicating whether the same set of physical variables can still be distilled from a predicted state $\hat{\boldsymbol{X}}$ as its corresponding ground-truth state. Consequently, if the predicted frame is heavily blurred or distorted, we will not be able to distil meaningful physical variables. Thus, $M_{\mathcal{S}}^{\mathrm{phys}}(\hat{\boldsymbol{X}})$ will be 1. Otherwise $M_{\mathcal{S}}^{\mathrm{phys}}(\hat{\boldsymbol{X}})$ will be 0.

Moreover, to more generally capture the long-term predictive stability of various prediction schemes, $M_{\mathcal{S}}^{\mathrm{phys}}(\cdot)$ should be evaluated on prediction sequences with multiple initial states. Therefore, we further define $M_{\mathcal{S}}^{\mathrm{phys}}(\cdot)$ as the reject ratio to indicate how many predicted frames at each time step from different initial states will fail to pass the physical variables extraction test.

With the above test, we can quantitatively compare the stability of various long-term prediction schemes, namely iterative long-term predictions through high-dimensional latent vectors (8,192 variables or 64 variables), and predictions through neural state variables. Please refer to Methods for a formal definition of each scheme.

Figure 4a shows the stability results for the rigid double pendulum and elastic double pendulum where we can extract the physical variables from videos. The 8,192-dimensional and 64-dimensional schemes cannot give stable long-term future predictions. In our

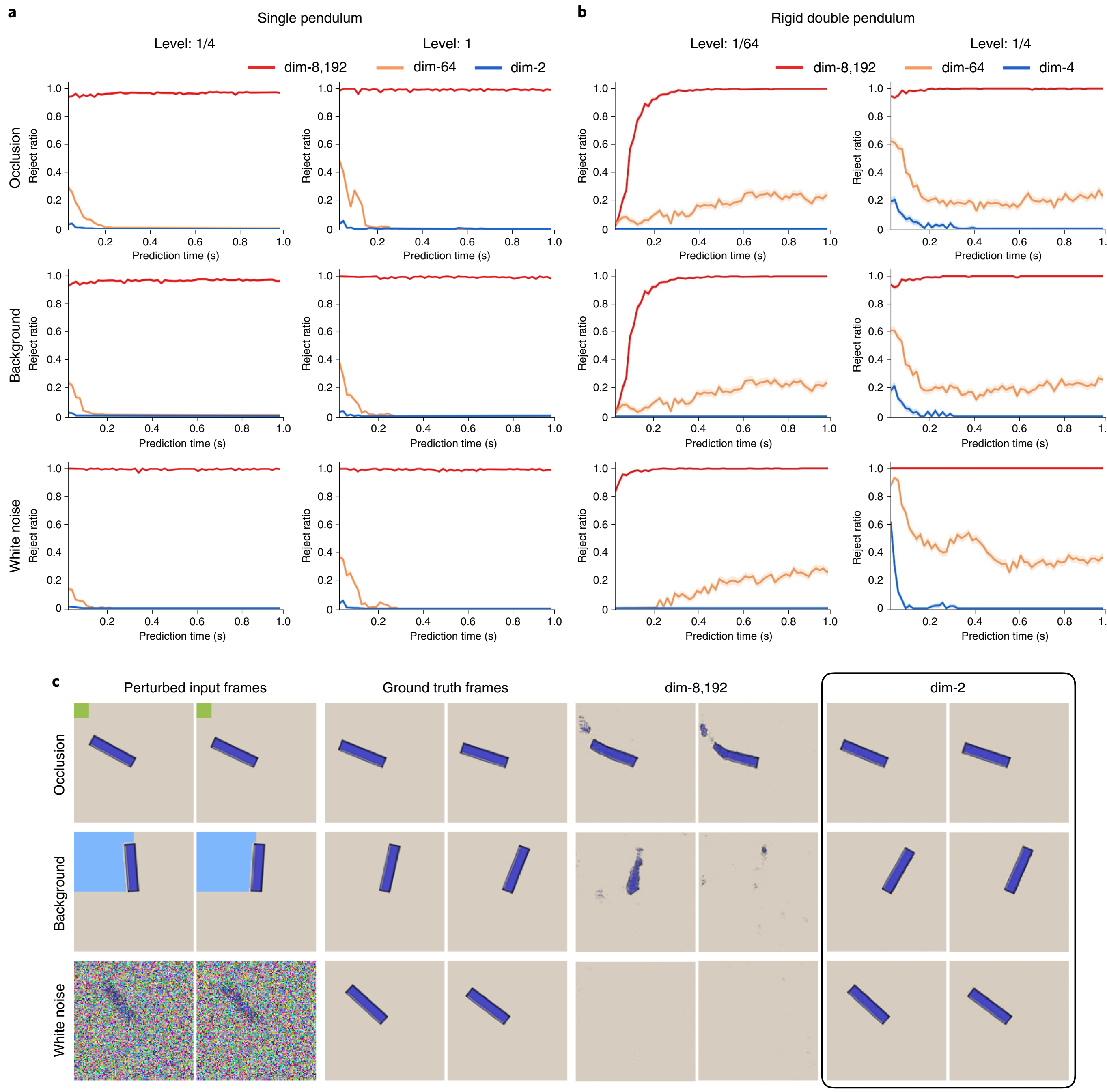


Fig. 6 | Neural state variables for robust long-term prediction. a,**b**, The perturbation evaluation results using the physical reject ratio metric to measure the long-term prediction robustness on the single-pendulum system (**a**) and the rigid double-pendulum system (**b**). Each subplot depicts the results under different perturbation levels across all prediction schemes. Each row implies a different perturbation type. s.e.m. values of the reject ratios are reflected as the shaded regions. dim-8,192 and dim-64 are prediction schemes directly through relatively high-dimensional latent space with dimensions 8,192 and 64 respectively. dim-2 and dim-4 are prediction schemes through the space of neural state variables. **c**, Visualizations of the perturbation evaluations. We show the perturbed input frames in the first two columns, followed by the ground-truth future frames and the predicted future frames from two different prediction schemes. The dim-2 column is the output from our prediction scheme through the space of neural state variables.

experiments, we found that the same conclusion still holds for various relatively high dimensions. Therefore, our finding is robust to different dimensions. We also noticed that both schemes can provide stable predictions when the system ID is smaller or equal than 2.

Inspired by lessons learned from computational physics, an effective fix for the unstable long-term prediction is to construct a prediction scheme where the predicted states will be projected into a small neighbourhood of the state space. Here neural state variables serve as a reasonable candidate solution. This is because neural state variables have the same dimension as the ID. This fact prevents predictions from falling off the system manifold into new dimensions. We provide more detailed theoretical analysis in Supplementary Section 7.

The blue curves in Fig. 4a illustrate the stability introduced by using neural state variables as intermediate representations for long-term predictions. Neural state variables provide the most stable predictions across all the systems. However, since neural state variables were obtained by performing reconstruction on a relatively high-dimensional latent embedding, they have an inferior performance on one-step prediction accuracy.

To leverage the advantages from both worlds, we propose a hybrid scheme as our final solution: using neural state variables as stabilizers while performing long-term predictions with their corresponding high-dimensional latent embeddings. Formally, the hybrid scheme follows an $N+1$ pattern, where for every N steps performed with the high-dimensional latent vectors a one-step prediction is followed with the neural state variables. As shown by the purple curves in Fig. 4a, our hybrid scheme offers stable and accurate long-term predictions. In Fig. 4, the hybrid scheme was implemented with specific values of N between 3 and 6. We also conducted experiments described in Supplementary Section 6 with different choices of N and found that the outcomes were not sensitive to the particular values of N.

Another important note is that the use of pixel error as the evaluation metric, though easy to compute, can be misleading for the evaluation of long-term predictions when the predictions quickly become unstable. As quantitatively and qualitatively demonstrated in Fig. 4, pixel errors remain roughly the same after the predicted images become plain backgrounds. These pixel errors are even smaller than the pixel errors computed from a slightly inaccurate but clear prediction. This observation further emphasizes the importance of designing an appropriate $M_{\mathcal{S}}(\cdot)$ metric.

Neural state variables for dynamics stability indicators. So far, to evaluate long-term prediction stability, we have been assuming that we can extract the physical variables from the system states during evaluation. However, in most of the video representations in our dataset we do not know either which variables to extract or how to extract them directly from videos. As noted above, pixel errors are also not reliable. In this case, a very challenging but important problem is how we can evaluate the long-term prediction stability from videos. Resolving this problem can potentially open up the door to quantitatively evaluate prediction stability of various schemes for many complex and unknown systems, all directly from videos.

Following the framework in the last section, the key is the design of the metric $M_{\mathcal{S}}(\cdot)$. Here we propose a solution based on neural state variables, namely $M_{\mathcal{S}}^{\mathrm{neur}}$. Specifically, $M_{\mathcal{S}}^{\mathrm{neur}}$ is a metric on a pair of states $(\hat{\boldsymbol{X}}_t, \hat{\boldsymbol{X}}_{t+\mathrm{d}t})$.

$$M_{\mathcal{S}}^{\mathrm{neur}}(\hat{\boldsymbol{X}}_t, \hat{\boldsymbol{X}}_{t+\mathrm{d}t}) = \left|h_{\mathrm{E}} \circ g_{\mathrm{E}}(\hat{\boldsymbol{X}}_{t+\mathrm{d}t}) - \hat{F}_V(h_{\mathrm{E}} \circ g_{\mathrm{E}}(\hat{\boldsymbol{X}}_t))\right|,$$

where $\hat{F}_V$ is a neural network trained to approximate the latent dynamics on the space of neural state variables $\hat{\boldsymbol{V}}_{t+\mathrm{d}t\to t+2\,\mathrm{d}t} \leftarrow \hat{F}_V(\boldsymbol{V}_{t\to t+\mathrm{d}t})$, $h_{\mathrm{E}} \circ g_{\mathrm{E}}(\hat{\boldsymbol{X}}_t) = \hat{\boldsymbol{V}}_{t\to t+\mathrm{d}t}$ and $h_{\mathrm{E}} \circ g_{\mathrm{E}}(\hat{\boldsymbol{X}}_{t+\mathrm{d}t}) = \hat{\boldsymbol{V}}_{t+\mathrm{d}t\to t+2\,\mathrm{d}t}$ are neural states in $\mathbb{R}^{\mathrm{ID}}$ and $|\cdot|$ is the Euclidean norm in $\mathbb{R}^{\mathrm{ID}}$.

As shown in the previous sections, all latent embeddings with dimension higher or equal to the ID may provide an accurate short-term approximation of system dynamics. However, we chose the space of neural state variables as the reference because it has the same dimension as the ID. First, as mentioned above, neural state variables project the predicted states in the small neighbourhood of the ground-truth states. Moreover, the Euclidean distance serves as a good metric to measure the dynamics deviation in this case, while other higher dimensions may suffer the curse of dimensionality when designing the distance metric. Overall, $M_{\mathcal{S}}^{\mathrm{neur}}$ is an ideal alternative candidate to $M_{\mathcal{S}}^{\mathrm{phys}}$.

Similarly to $M_{\mathcal{S}}^{\mathrm{phys}}$, the final $M_{\mathcal{S}}^{\mathrm{neur}}$ is computed across multiple prediction sequences with various initial states. We show the evaluation results with our stability metric based on neural state variables in Fig. 5. $M_{\mathcal{S}}^{\mathrm{neur}}$ produces patterns that strongly match with $M_{\mathcal{S}}^{\mathrm{phys}}$ for the systems where we know how to extract physical variables. This can also be seen in the correlation plot in Fig. 5 where we computed the Pearson correlation coefficient between the reject ratios of all models (dim-8192, dim-64, dim-ID, hybrid) at all prediction steps and the respective latent dynamics errors. For unknown systems, we observed the same trend, where high-dimensional latent embedding schemes are often not stable. In conclusion, our $M_{\mathcal{S}}^{\mathrm{neur}}$ metric can help us measure the long-term prediction stability directly from videos without additional prior knowledge of the system. $M_{\mathcal{S}}^{\mathrm{neur}}$ evaluates the long-term prediction stability in a different space than does $M_{\mathcal{S}}^{\mathrm{phys}}$.

Neural state variables for robust long-term prediction. Another critical factor when modelling system dynamics from videos is the robustness against visual perturbations. Therefore, we applied several visual perturbations in the space of visual sensor observations (that is, the input video frames) during the test time and evaluated the performances of different models. Please refer to Methods for a detailed description of each type of applied perturbation.

We show the test-time results using the physical reject ratio metric in Fig. 6a,b. The quantitative results clearly demonstrate the strong robustness of models on the neural state variable space across all levels of perturbation. The models with very high-dimensional latent space quickly produce unstable predictions. The models with a dimension that is relatively low but still higher than ID can sometimes give stable predictions again after several unstable rollouts. However, even though the predictions can become stable again, this requires a much greater number of prediction steps. We also show qualitative visualizations in Fig. 6c.

Though our models on the neural state variable space consistently produce stable predictions under various noise perturbations, they do not always give accurate predictions, especially when the noise levels are relatively high. We provide more failure examples in Supplementary Section 12.

Analysis. We hypothesize that neural state variables contain rich physical meanings that align with the conventional definition of the physical state variables. In this section, we verify this hypothesis through both quantitative regression experiments and qualitative visualizations.

We trained a small neural network with five layers of multilayer perceptrons to regress conventional physical variables including positions, velocities and energies from learned neural state variables. Our results are shown in Supplementary Section 9. The values are L1 errors with their s.e.m. Using 30% of labelled data, the learned neural state variables can be used to accurately regress the physical variables. We then compared the regression errors with those from the first few principal components of high-dimensional latent vectors from our dynamics predictive model. Using the same number of state variables, which is equal to the ID, and the same labelled data, the regression errors using principal components of high-dimensional latent vectors are much larger than those using neural state variables, especially for velocity variables. Therefore, state variables obtained through principal component analysis, or equivalently through linear neural networks, have difficulty in capturing the dynamics of the system.

Visualizations coloured according to the value of physical variables in Extended Data Fig. 2 can further demonstrate the physical meaning of the neural state variables. We observe that the physical variables are indeed captured in the set of neural state variables chosen by our modelling system. The charts also reveal the inherent symmetrical nature of these variables.

Discussion

Overall, this work proposes several advancements that complement the existing works. There are many promising directions for the future research. First, the learned neural state variables currently do not have units, which make them less interpretable in physics. This is due to the non-uniqueness of the current neural state variables. Therefore, one future direction is to further regularize the neural state variables with prior physical knowledge, existing physics-informed artificial intelligence techniques or dimensional analysis with units, so that they could have better correspondence to traditional physical variables and satisfy physical constraints such as energy conservation[40–47]. Another interesting question is to investigate if there is a governing principle to allow us to to choose one set of variables over another[48]. Moreover, when frames are corrupted or contain incomplete information because of hidden factors, system uncertainty or inappropriate sampling frequency, the observation data may not fully capture the real physics. Regularization techniques[49,50] can help us design algorithms to provide effective and automated remedies for such imperfect observation data. For complex systems with little prior physical knowledge, another interesting direction is to analyse the neural state variables and translate them into human-interpretable physics. Finally, the proposed framework of distilling neural state variables and generating stable long-term predictions can be used as a component of automated control systems.

Methods

Dynamics predictive model from videos. The first step towards modelling a dynamical system is to choose the representation of system states. Previous studies assume that the states are given as the direct measurements of a set of predefined state variables, such as the position and velocity of a rigid body object. However, defining which state variables to measure requires expert prior knowledge of the system. For an unfamiliar physical system, we do not know in advance what quantities to measure. Moreover, most state variables are not directly measurable, as they correspond to properties that are not physically observable in a non-intrusive manner or cannot be uniquely determined without prior knowledge.

In this work, we chose video frames as the state representation. Using the notations above, $\mathcal{X}$ is the high-dimensional image space. This choice comes with several advantages. First, video recordings do not require prior knowledge of the inner working processes of the observed dynamical system. Second, video cameras collect a rich stream of physics signals, without requiring expensive and specialized equipment. If we can apply our method to data collected by video cameras, then this approach could potentially operate with other types of sensor array.

For the dynamical systems studied in our paper, both the input and output image pairs are two consecutive frames with the dimension $128\times128\times3$ RGB channels. The pairs of frames are concatenated to form single input and output image. g_{E} and g_{D} are fully convolutional networks. The network first outputs $\boldsymbol{L}_{t\to t+\mathrm{d}t}=g_{\mathrm{E}}(\boldsymbol{X}_t)$ and then generates the future frames $\hat{\boldsymbol{X}}_{t+\mathrm{d}t}$.

To train the encoder and decoder networks, we use a simple L2 loss function without other constraints:

$$\mathcal{L}=\mathbb{E}_{\boldsymbol{X}}\left[\|\, g_{\mathrm{D}}(g_{\mathrm{E}}(\boldsymbol{X}_t))-\boldsymbol{X}_{t+\mathrm{d}t}\|_2^2\right].$$

The learned mapping $\hat{F}=g_{\mathrm{D}}\circ g_{\mathrm{E}}$ provides a numerical approximation of the system's evolution mapping F through the latent embedding.

Discussion on the dimension of latent embedding. One critical but largely ignored design decision is the dimension of the latent embedding $\boldsymbol{L}_{t\to t+\mathrm{d}t}$ of the first autoencoder network to predict future video frames. Here we define this dimension as LD. In machine learning, LD is often treated as a hyperparameter selected using an 'educated guess' because it is not immediately clear what the best value of LD should be. However, this dimensionality is especially important for physical dynamics modelling. When LD is large, the latent embedding can hold large numbers of useful bits of information about the system dynamics. However, large embedding vectors can overfit the data and have limited capacity for longer-range prediction. More importantly, large latent spaces hide and obfuscate the compact set of state variables we are after. When LD is too small, the network may underfit the data. Therefore, we aim to come as close as possible to the exact number of state variables in the next section.

Levina–Bickel algorithm for ID estimation. The algorithm considers latent vectors $\{\boldsymbol{L}^{(1)},\boldsymbol{L}^{(2)},\ldots,\boldsymbol{L}^{(N)}\}$ collected from the trained neural network that predicts dynamics as N data points on a manifold of dimension ID in the latent embedding space. A key geometric observation is that the number of data points within distance r from any given data point $\boldsymbol{L}^{(i)}$ is proportional to r^{ID} when r is small.

On the basis of the observation, the Levina–Bickel algorithm derives the local ID estimator near $\boldsymbol{L}^{(i)}$ as $\frac{1}{k-2}\sum_{j=1}^{k-1}\log\frac{T_k(\boldsymbol{L}^{(i)})}{T_j(\boldsymbol{L}^{(i)})}$, where $T_k(\boldsymbol{L}^{(i)})$ is the Euclidean distance between $\boldsymbol{L}^{(i)}$ and its kth nearest neighbour in $\{\boldsymbol{L}^{(1)},\boldsymbol{L}^{(2)},\ldots,\boldsymbol{L}^{(N)}\}$. The global ID estimator is then calculated as

$$\mathrm{ID}_{\mathrm{L-B}}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{k-2}\sum_{j=1}^{k-1}\log\frac{T_k(\boldsymbol{L}^{(i)})}{T_j(\boldsymbol{L}^{(i)})}.$$

Model details to discover neural state variables. In this section, we detail our model that reconstructs the pretrained latent embedding from the aforementioned dynamics predictive model. Specifically, the network can be expressed as follows: $\boldsymbol{V}_{t\to t+\mathrm{d}t}=h_{\mathrm{E}}(\boldsymbol{L}_{t\to t+\mathrm{d}t})$ and $\hat{\boldsymbol{L}}_{t\to t+\mathrm{d}t}=h_{\mathrm{D}}(\boldsymbol{V}_{t\to t+\mathrm{d}t})$. We train the latent reconstruction model with the L2 loss: $\mathcal{L}=\mathbb{E}_{\boldsymbol{L}}\left[\|\, h_{\mathrm{D}}(h_{\mathrm{E}}(\boldsymbol{L}_{t\to t+\mathrm{d}t}))-\boldsymbol{L}_{t\to t+\mathrm{d}t}\|_2^2\right]$.

Definition of long-term prediction schemes. These schemes are based on iterative model rollouts but they differ in the size of intermediate variables. When the model rollouts are through high-dimensional latent vectors (8,192 variables or 64 variables), the iterative scheme is given by $\hat{\boldsymbol{X}}_{t+\mathrm{d}t}=g_{\mathrm{D}}\circ g_{\mathrm{E}}(\hat{\boldsymbol{X}}_t),\quad t=0,\mathrm{d}t,\ldots$, where g_{E} and g_{D} represent the first autoencoder that transforms input frames to the predicted frames via latent embeddings. When the model rollouts are through neural state variables, the iterative scheme is given by $\hat{\boldsymbol{X}}_{t+\mathrm{d}t}=g_{\mathrm{D}}\circ h_{\mathrm{D}}\circ h_{\mathrm{E}}\circ g_{\mathrm{E}}(\hat{\boldsymbol{X}}_t),\quad t=0,\mathrm{d}t,\ldots$. The original latent embeddings are computed from input frames with g_{E}, and the reconstructed latent embeddings will be sent to g_{D} to produce the final predicted frames.

Intuition of evaluating dynamics stability with neural state variables. Intuitively, M_S^{neur} measures how far the predicted dynamics, reflected by the given predicted sequence, deviates from the reduced system dynamics projected onto the space of neural state variables. The above equation can be conceptually thought of measuring the distance between two quantities. The first quantity is calculated by projecting the predicted system dynamics from the high-dimensional model to the neural state variable space, and the second quantity is the reduced system dynamics from the ID model.

Perturbations for evaluating long-term prediction robustness. We performed three types of perturbation. The first type is to simulate camera occlusions by covering a certain portion of the input frames with a randomly generated colour square. The area of the square indicates the level of the perturbation. For example, $\frac{1}{64}$ means that the area of the square is $\frac{1}{64}$ of the area of one input frame.

The second type is to simulate background colour change by covering a certain portion of the input frame background with a randomly generated colour square. The main difference between this perturbation and the first one is that the colour square will not cover the object. The level definition is the same as for the first perturbation type.

Finally, to simulate possible sensor noise, we added random Gaussian noise to the input frames. The Gaussian noise has a zero mean and different levels of s.d. For example, $\frac{1}{64}$ means that the Gaussian noise has an s.d. of $\frac{1}{64}\times255$, where 255 is the highest pixel value in the input frames.

Data availability

All of our simulated and physical dataset repository is available[51]. Source data for Figs. 2b, 3b, 4a, 5 and 6a and Extended Data Fig. 2 are available for this Article.

Code availability

The open-source code to reproduce our training and evaluation results is available at the Zenodo repository[52] and GitHub (https://github.com/BoyuanChen/neural-state-variables).

Received: 24 February 2022; Accepted: 21 June 2022;
Published online: 25 July 2022

References

1. Anderson, P. W. More is different. *Science* **177**, 393–396 (1972).
2. Thompson, J. M. T. & Stewart, H. B. *Nonlinear Dynamics and Chaos* (Wiley, 2002).
3. Hirsch, M. W., Smale, S. & Devaney, R. L. *Differential Equations, Dynamical Systems, and an Introduction to Chaos* (Academic, 2012).
4. Kutz, J. N., Brunton, S. L., Brunton, B. W. & Proctor, J. L. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (SIAM, 2016).
5. Evans, J. & Rzhetsky, A. Machine science. *Science* **329**, 399–400 (2010).
6. Fortunato, S. et al. Science of science. *Science* **359**, eaao0185 (2018).
7. Bongard, J. & Lipson, H. Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104**, 9943–9948 (2007).
8. Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).

9. King, R. D., Muggleton, S. H., Srinivasan, A. & Sternberg, M. Structure–activity relationships derived by machine learning: the use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. Natl Acad. Sci. USA* **93**, 438–442 (1996).
10. Waltz, D. & Buchanan, B. G. Automating science. *Science* **324**, 43–44 (2009).
11. King, R. D. et al. The robot scientist Adam. *Computer* **42**, 46–54 (2009).
12. Langley, P. BACON: a production system that discovers empirical laws. In *Proc. Fifth International Joint Conference on Artificial Intelligence* Vol. 1 344 (Morgan Kaufmann, 1977).
13. Langley, P. Rediscovering physics with BACON.3. In *Proc. Sixth International Joint Conference on Artificial Intelligence* Vol. 1 505–507 (Morgan Kaufmann, 1979).
14. Crutchfield, J. P. & McNamara, B. Equations of motion from a data series. *Complex Syst.* **1**, 417–452 (1987).
15. Kevrekidis, I. G. et al. Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Commun. Math. Sci.* **1**, 715–762 (2003).
16. Yao, C. & Bollt, E. M. Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems. *Physica D* **227**, 78–99 (2007).
17. Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P. & Henningson, D. S. Spectral analysis of nonlinear flows. *J. Fluid Mech.* **641**, 115–127 (2009).
18. Schmidt, M. D. et al. Automated refinement and inference of analytical models for metabolic networks. *Phys. Biol.* **8**, 055011 (2011).
19. Sugihara, G. et al. Detecting causality in complex ecosystems. *Science* **338**, 496–500 (2012).
20. Ye, H. et al. Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. *Proc. Natl Acad. Sci. USA* **112**, E1569–E1576 (2015).
21. Daniels, B. C. & Nemenman, I. Automated adaptive inference of phenomenological dynamical models. *Nat. Commun.* **6**, 8133 (2015).
22. Daniels, B. C. & Nemenman, I. Efficient inference of parsimonious phenomenological models of cellular dynamics using S-systems and alternating regression. *PloS ONE* **10**, e0119821 (2015).
23. Benner, P., Gugercin, S. & Willcox, K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**, 483–531 (2015).
24. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937 (2016).
25. Rudy, S. H., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614 (2017).
26. Udrescu, S.-M. & Tegmark, M. AI Feynman: a physics-inspired method for symbolic regression. *Sci. Adv.* **6**, eaay2631 (2020).
27. Mrowca D. et al. Flexible neural representation for physics prediction. In *Advances in Neural Information Processing Systems* Vol. 31 (eds Bengio, S. et al.) (Curran Associates, 2018).
28. Champion, K., Lusch, B., Kutz, J. N. & Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116**, 22445–22451 (2019).
29. Baldi, P. & Hornik, K. Neural networks and principal component analysis: learning from examples without local minima. *Neural Netw.* **2**, 53–58 (1989).
30. Hinton, G. E. & Zemel, R. S. Autoencoders, minimum description length, and Helmholtz free energy. *Adv. Neural Inf. Process. Syst.* **6**, 3 (1994).
31. Masci, J., Meier, U., Cireşan, D. & Schmidhuber, J. Stacked convolutional autoencoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks* 52–59 (Springer, 2011).
32. Bishop C. M. et al. *Neural Networks for Pattern Recognition* (Oxford Univ. Press, 1995).
33. Camastra, F. & Staiano, A. Intrinsic dimension estimation: advances and open problems. *Inf. Sci.* **328**, 26–41 (2016).
34. Campadelli, P., Casiraghi, E., Ceruti, C. & Rozza, A. Intrinsic dimension estimation: relevant techniques and a benchmark framework. *Math. Probl. Eng.* **2015**, 759567 (2015).
35. Levina, E. & Bickel, P. J. Maximum likelihood estimation of intrinsic dimension. In *Proc. 17th International Conference on Neural Information Processing Systems* 777–784 (MIT Press, 2005).
36. Rozza, A., Lombardi, G., Ceruti, C., Casiraghi, E. & Campadelli, P. Novel high intrinsic dimensionality estimators. *Mach. Learn.* **89**, 37–65 (2012).
37. Ceruti, C. et al. DANCo: an intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern Recognit.* **47**, 2569–2581 (2014).
38. Hein, M. & Audibert, J.-Y. Intrinsic dimensionality estimation of submanifolds in R^d. In *Proc. 22nd International Conference on Machine Learning* 289–296 (Association for Computing Machinery, 2005).
39. Grassberger, P. & Procaccia, I. in *The Theory of Chaotic Attractors* 170–189 (Springer, 2004).
40. Pukrittayakamee, A. et al. Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks. *J. Chem. Phys.* **130**, 134101 (2009).
41. Wu, J., Lim, J. J., Zhang, H., Tenenbaum, J. B. & Freeman, W. T. Physics 101: Learning physical object properties from unlabeled videos. In *Proc. British Machine Vision Conference (BMVC)* (eds Wilson, R. C. et al.) 39.1-39.12 (BMVA Press, 2016).
42. Chmiela, S. et al. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, e1603015 (2017).
43. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 13890 (2017).
44. Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
45. Lutter, M., Ritter, C. & Peters, J. Deep Lagrangian networks: using physics as model prior for deep learning. In *International Conference on Learning Representations* (2019).
46. Bondesan, R. & Lamacraft, A. Learning symmetries of classical integrable systems. Preprint at https://arxiv.org/abs/1906.04645 (2019).
47. Greydanus, S. J., Dzumba, M. & Yosinski, J. Hamiltonian neural networks. Preprint at https://arxiv.org/abs/1906.01563 (2019).
48. Swischuk, R., Kramer, B., Huang, C. & Willcox, K. Learning physics-based reduced-order models for a single-injector combustion process. *AIAA J.* **58**, 2658–2672 (2020).
49. Lange, H., Brunton, S. L. & Kutz, J. N. From Fourier to Koopman: spectral methods for long-term time series prediction. *J. Mach. Learn. Res.* **22**, 1–38 (2021).
50. Mallen, A., Lange, H. & Kutz, J. N. Deep probabilistic Koopman: long-term time-series forecasting under periodic uncertainties. Preprint at https://arxiv.org/abs/2106.06033 (2021).
51. Chen B. et al. Dataset for the paper titled Discovering State Variables Hidden in Experimental Data (1.0). *Zenodo* https://doi.org/10.5281/zenodo.6653856 (2022).
52. Chen B. et al. BoyuanChen/neural-state-variables: (v1.0). *Zenodo* https://doi.org/10.5281/zenodo.6629185 (2022).

Acknowledgements

This research was supported in part by NSF AI Institute for Dynamical Systems grant 2112085 (to H.L.), DARPA MTO Lifelong Learning Machines (L2M) Program HR0011-18-2-0020 (to H.L.), NSF NRI grant 1925157 (to H.L.), NSF DMS grant 1937254 (to Q.D.), NSF DMS grant 2012562 (to Q.D.), NSF CCF grant 1704833 (to Q.D.), DE grant SC0022317 (to Q.D.) and DOE ASCR DE grant SC0022317 (to Q.D.).

Author contributions

B.C. and H.L. proposed the research; B.C., K.H., H.L. and Q.D. performed experiments and numerical analysis, B.C. and K.H. designed the algorithms; B.C., K.H., I.C. and S.R. collected the dataset; B.C., K.H., H.L. and Q.D. wrote the paper; all authors provided feedback.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s43588-022-00281-6.

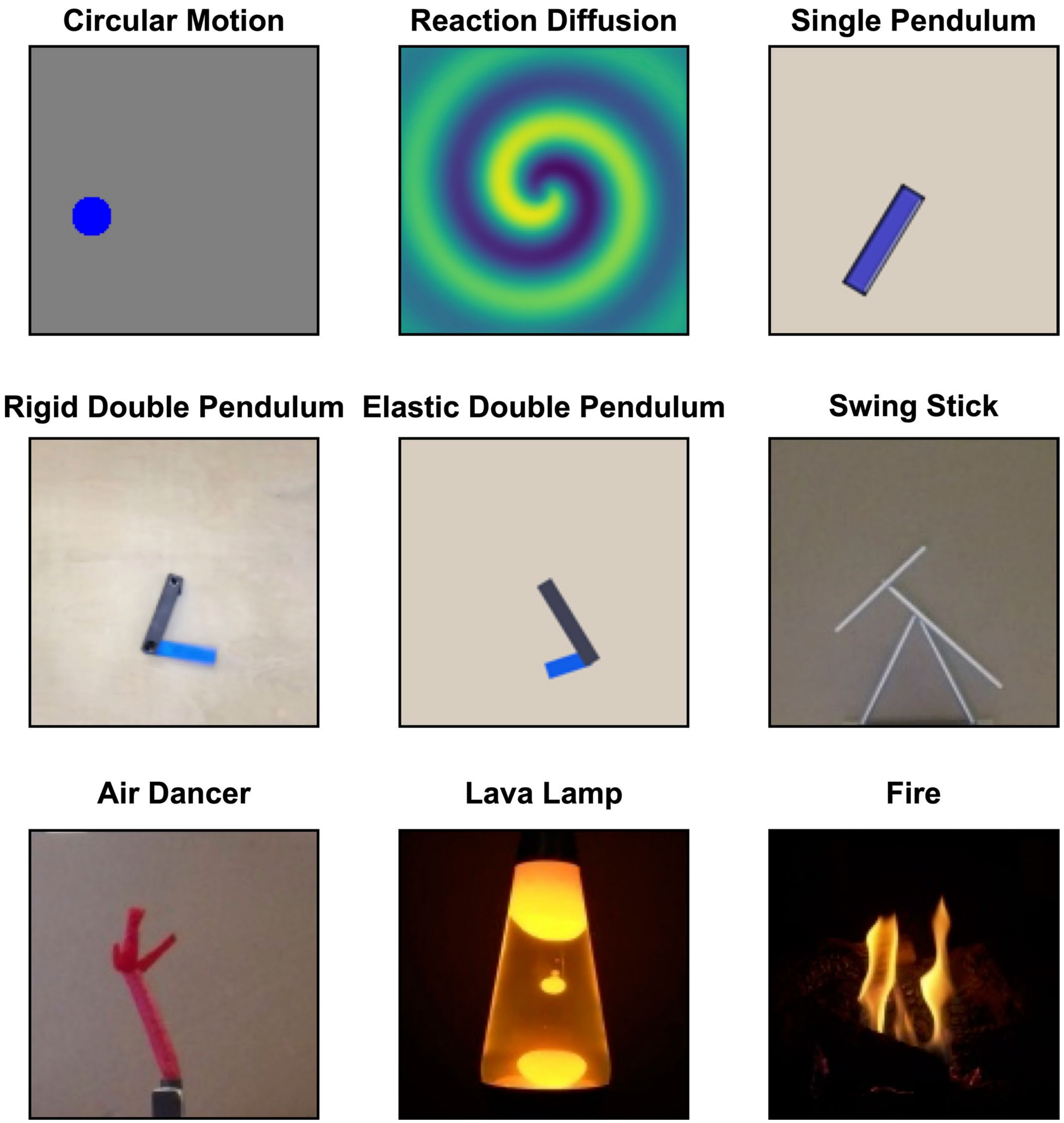
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-022-00281-6.

Correspondence and requests for materials should be addressed to Boyuan Chen.

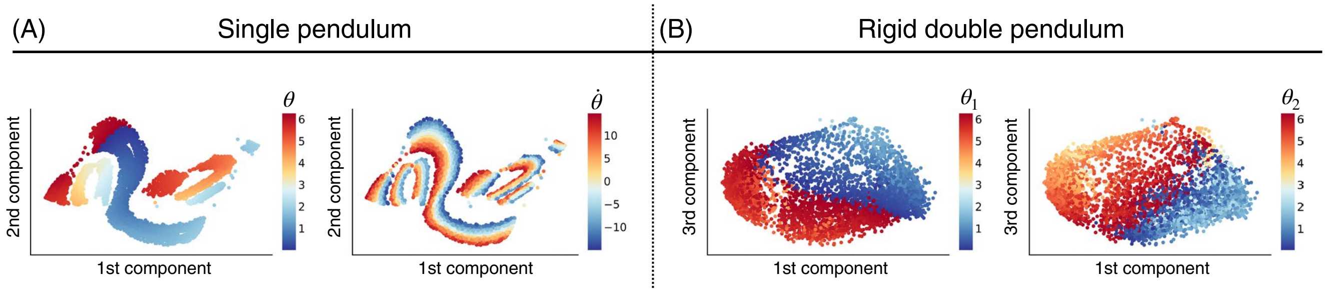
Peer review information *Nature Computational Science* thanks Bryan Daniels and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Jie Pan, in collaboration with the *Nature Computational Science* team. Peer reviewer reports are available.

Reprints and permissions information is available at www.nature.com/reprints.

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



Extended Data Fig. 1 | What state variables describe these dynamical systems?. What state variables describe these dynamical systems? Identifying state variables from raw observation data is a precursor step to discovering physical laws. The key challenge is to figure out how many variables will give a complete and non-redundant description of the system's states, what are the candidate variables, and how the variables are dependent on each other. Our work studies how to retrieve possible set of state variables from data distributions non-linearly embedded in the ambient space.



Extended Data Fig. 2 | PCA and Neural State Variables visualization. PCA and Neural State Variables visualization. Here we visualize the interesting symmetrical structures encoded in the Neural State Variables from single pendulum (A) and rigid double pendulum (B) after applying PCA algorithm on them. The colors represent the value of different physical variables. The x-axis and y-axis represent different components of the Neural State Variables.