Data-driven discovery of coordinates and governing equations

Kathleen Champion[a,1], Bethany Lusch[b], J. Nathan Kutz[a], and Steven L. Brunton[a,c]

[a]Department of Applied Mathematics, University of Washington, Seattle, WA 98195; [b]Leadership Computing Facility, Argonne National Laboratory, Lemont, IL 60439; and [c]Department of Mechanical Engineering, University of Washington, Seattle, WA 98195

Edited by David L. Donoho, Stanford University, Stanford, CA, and approved September 30, 2019 (received for review April 25, 2019)

The discovery of governing equations from scientific data has the potential to transform data-rich fields that lack well-characterized quantitative descriptions. Advances in sparse regression are currently enabling the tractable identification of both the structure and parameters of a nonlinear dynamical system from data. The resulting models have the fewest terms necessary to describe the dynamics, balancing model complexity with descriptive ability, and thus promoting interpretability and generalizability. This provides an algorithmic approach to Occam's razor for model discovery. However, this approach fundamentally relies on an effective coordinate system in which the dynamics have a simple representation. In this work, we design a custom deep autoencoder network to discover a coordinate transformation into a reduced space where the dynamics may be sparsely represented. Thus, we simultaneously learn the governing equations and the associated coordinate system. We demonstrate this approach on several example high-dimensional systems with low-dimensional behavior. The resulting modeling framework combines the strengths of deep neural networks for flexible representation and sparse identification of nonlinear dynamics (SINDy) for parsimonious models. This method places the discovery of coordinates and models on an equal footing.

model discovery | dynamical systems | machine learning | deep learning

Governing equations are of fundamental importance across all scientific disciplines. Accurate models allow for understanding of physical processes, which in turn gives rise to an infrastructure for the development of technology. The traditional derivation of governing equations is based on underlying first principles, such as conservation laws and symmetries, or from universal laws, such as gravitation. However, in many modern systems, governing equations are unknown or only partially known, and recourse to first-principles derivations is untenable. Instead, many of these systems have rich time-series data due to emerging sensor and measurement technologies (e.g., in biology and climate science). This has given rise to the new paradigm of data-driven model discovery, which is the focus of intense research efforts (1–14). A central tension in model discovery is the balance between model efficiency and descriptive capabilities. Parsimonious models strike this balance, having the fewest terms required to capture essential interactions (1, 3, 8, 10, 15), thus promoting interpretability and generalizability. Obtaining parsimonious models is fundamentally linked to the coordinate system in which the dynamics are measured. Without proper coordinates, standard approaches may fail to discover simple dynamical models. In this work, we simultaneously discover effective coordinates via a custom autoencoder (16–18), along with the parsimonious dynamical system model via sparse regression in a library of candidate terms (8). The joint discovery of models and coordinates is critical for understanding many modern systems.

Numerous recent approaches leverage neural networks to model time-series data (18–26). When interpretability and generalizability are primary concerns, it is important to identify parsimonious models that have the fewest terms required to describe the dynamics, which is the antithesis of neural networks whose parameterizations are exceedingly large. A breakthrough approach used symbolic regression to learn the form of dynamical systems and governing laws from data (1, 3). Sparse identification of nonlinear dynamics (SINDy) (8) is a related approach that uses sparse regression to find the fewest terms in a library of candidate functions required to model the dynamics. Because this approach is based on a sparsity-promoting linear regression, it is possible to incorporate partial knowledge of the physics, such as symmetries, constraints, and conservation laws (27). Successful modeling requires that the dynamics are measured in a coordinate system where they may be sparsely represented. While simple models may exist in one coordinate system, a different coordinate system may obscure these parsimonious representations. For modern applications of data-driven discovery, there is no reason to believe that we measure the correct variables to admit a simple representation of the dynamics. This motivates the present study to enable systematic and automated discovery of coordinate transformations that facilitate this sparse representation.

The challenge of discovering an effective coordinate system is as fundamental and important as model discovery. Many key scientific breakthroughs were enabled by the discovery of appropriate coordinate systems. Celestial mechanics, for instance, was revolutionized by the heliocentric coordinate system of Copernicus, Galileo, and Kepler, thus displacing Ptolemy's doctrine of the perfect circle, which was dogma for more than a millennium. The Fourier transform was introduced to simplify the representation of the heat equation, resulting in a sparse,

Significance

Governing equations are essential to the study of physical systems, providing models that can generalize to predict previously unseen behaviors. There are many systems of interest across disciplines where large quantities of data have been collected, but the underlying governing equations remain unknown. This work introduces an approach to discover governing models from data. The proposed method addresses a key limitation of prior approaches by simultaneously discovering coordinates that admit a parsimonious dynamical model. Developing parsimonious and interpretable governing models has the potential to transform our understanding of complex systems, including in neuroscience, biology, and climate science.

Author contributions: K.C., B.L., J.N.K., and S.L.B. designed research; K.C. performed research; and K.C., B.L., J.N.K., and S.L.B. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

Data deposition: The source code used in this work is available at GitHub (https://github.com/kpchamp/SindyAutoencoders).

[1]To whom correspondence may be addressed. Email: kpchamp@uw.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1906995116/-/DCSupplemental.

First published October 21, 2019.

diagonal, decoupled linear system. Eigen-coordinates have been used more broadly to enable sparse dynamics, for example in quantum mechanics and electrodynamics, to characterize energy levels in atoms and propagating modes in waveguides, respectively. Principal component analysis (PCA) is one of the most prolific modern coordinate discovery methods, representing high-dimensional data in a low-dimensional linear subspace. Nonlinear extensions of PCA have been enabled by a neural network architecture, called an autoencoder (16, 17, 28). However, PCA and autoencoders generally do not take dynamics into account and, thus, may not provide the right basis for parsimonious dynamical models. In related work, Koopman analysis seeks coordinates that linearize nonlinear dynamics (29); while linear models are useful for prediction and control, they cannot capture the full behavior of many nonlinear systems. Thus, it is important to develop methods that combine simplifying coordinate transformations and nonlinear dynamics. We advocate for a balance between these approaches, identifying coordinate transformations where only a few nonlinear terms are present, as in near-identity transformations and normal forms.

In this work we present a method to discover nonlinear coordinate transformations that enable parsimonious dynamics. Our method combines a custom autoencoder network with a SINDy model for parsimonious nonlinear dynamics. The autoencoder enables the discovery of reduced coordinates from high-dimensional data, with a map back to reconstruct the full system. The reduced coordinates are found along with nonlinear governing equations for the dynamics in a joint optimization. We demonstrate the ability of our method to discover parsimonious dynamics on 3 examples: a high-dimensional spatial dataset with dynamics governed by the chaotic Lorenz system, the nonlinear pendulum, and a spiral wave resulting from the reaction–diffusion equation. These results demonstrate how to focus neural networks to discover interpretable dynamical models. Critically, the proposed method provides a mathematical framework that places the discovery of coordinates and models on equal footing.

Background

We review the SINDy (8) algorithm, which is a regression technique for extracting parsimonious dynamics from time-series data. The method takes snapshot data $\mathbf{x}(t) \in \mathbb{R}^n$ and attempts to discover a best-fit dynamical system with as few terms as possible:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)). \quad [1]$$

The state of the system $\mathbf{x}$ evolves in time t, with dynamics constrained by the function $\mathbf{f}$. We seek a parsimonious model for the dynamics, resulting in a function $\mathbf{f}$ that contains only a few active terms: It is sparse in a basis of possible functions. This is consistent with our extensive knowledge of a diverse set of evolution equations used throughout the physical, engineering, and biological sciences. Thus, the types of functions that compose $\mathbf{f}$ are typically known from modeling experience.

SINDy frames model discovery as a sparse regression problem. If snapshot derivatives are available, or can be calculated from data, the snapshots are stacked to form data matrices $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_m]^T$ and $\dot{\mathbf{X}} = [\dot{\mathbf{x}}_1\ \dot{\mathbf{x}}_2\ \cdots\ \dot{\mathbf{x}}_m]^T$ with $\mathbf{X}, \dot{\mathbf{X}} \in \mathbb{R}^{m \times n}$. Although $\mathbf{f}$ is unknown, we can construct an extensive library of p candidate functions $\boldsymbol{\Theta}(\mathbf{X}) = [\boldsymbol{\theta}_1(\mathbf{X}) \cdots \boldsymbol{\theta}_p(\mathbf{X})] \in \mathbb{R}^{m \times p}$, where each $\boldsymbol{\theta}_j$ is a candidate model term. We assume $m \gg p$ so the number of data snapshots is larger than the number of library functions; it may be necessary to sample transients and multiple initial conditions to improve the condition number of $\boldsymbol{\Theta}$. The choice of basis functions typically reflects some knowledge about the system of interest: A common choice is polynomials in $\mathbf{x}$ as these are elements of many canonical models. The library is used to formulate an overdetermined linear system

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi},$$

where the unknown matrix $\boldsymbol{\Xi} = (\boldsymbol{\xi}_1\ \boldsymbol{\xi}_2\ \cdots\ \boldsymbol{\xi}_n) \in \mathbb{R}^{p \times n}$ is the set of coefficients that determine the active terms from $\boldsymbol{\Theta}(\mathbf{X})$ in the dynamics $\mathbf{f}$. Sparsity-promoting regression is used to solve for $\boldsymbol{\Xi}$ that result in parsimonious models, ensuring that $\boldsymbol{\Xi}$, or more precisely each $\boldsymbol{\xi}_j$, is sparse and only a few columns of $\boldsymbol{\Theta}(\mathbf{X})$ are selected. For high-dimensional systems, the goal is to identify a low-dimensional state $\mathbf{z} = \varphi(\mathbf{x})$ with dynamics $\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z})$, as in Eq. **2**. The standard SINDy approach uses a sequentially thresholded least-squares algorithm to find the coefficients (8), which is a proxy for ℓ_0 optimization (30) and has convergence guarantees (31). Yao and Bollt (2) previously formulated system identification as a similar linear inverse problem without including sparsity, resulting in models that included all terms in $\boldsymbol{\Theta}$. In either case, an appealing aspect of this model discovery formulation is that it results in an overdetermined linear system for which many regularized solution techniques exist. Thus, it provides a computationally efficient counterpart to other model discovery frameworks (3).

SINDy has been widely applied to identify models for fluid flows (27), optical systems (32), chemical reaction dynamics (33), convection in a plasma (34), and structural modeling (35) and for model predictive control (36). There are also a number of theoretical extensions to the SINDy framework, including for identifying partial differential equations (10, 37), and models with rational function nonlinearities (38). It can also incorporate partially known physics and constraints (27). The algorithm can also be reformulated to include integral terms for noisy data (39) or handle incomplete or limited data (40, 41). The selected modes can also be evaluated using information criteria for model selection (42). These diverse mathematical developments provide a mature framework for broadening the applicability of the model discovery method.

Neural Networks for Dynamical Systems. The success of neural networks (NNs) on image classification and speech recognition has led to the use of NNs to perform a wide range of tasks in science and engineering (17). One recent focus has been the use of NNs to study dynamical systems, which has a surprisingly rich history (43). In addition to improving solution techniques for systems with known equations (24–26), deep learning has been used to understand and predict dynamics for complex systems with unknown equations (18–23). Several methods have trained NNs to predict dynamics, including a time-lagged autoencoder which takes the state at time t as input data and uses an autoencoder-like structure to predict the state at time $t + \tau$ (21). Other approaches use a recurrent architecture, particularly long short-term memory (LSTM) networks, for applications involving sequential data (44). LSTMs have recently been used for forecasting of chaotic dynamical systems (20). Reservoir computing has also enabled impressive predictions (13). Autoencoders are increasingly being leveraged for dynamical systems because of their close relationship to other dimensionality reduction techniques (28, 45–47).

Another class of NNs uses deep learning to discover coordinates for Koopman analysis. Koopman theory seeks to discover coordinates that linearize nonlinear dynamics (29). Methods such as dynamic mode decomposition (DMD) (4, 5, 9), extended DMD (48), and time-delay DMD (49) build linear models for dynamics, but these methods rely on a proper set of coordinates for linearization. Several recent works have focused on the use of deep-learning methods to discover the

proper coordinates for DMD and extended DMD (22, 23). Other methods seek to learn Koopman eigenfunctions and the associated linear dynamics directly using autoencoders (18). While autoencoders are particularly useful when reconstruction of the original state space is necessary, there are many applications in which full reconstruction is unnecessary. Koopman analysis and its combination with neural networks have also shown impressive results for use in such forecasting applications (19, 50).

Despite their widespread use, NNs face 3 major challenges: generalization, extrapolation, and interpretation. The hallmark success stories of NNs (computer vision and speech, for instance) have been on datasets that are fundamentally interpolatory in nature. The ability to extrapolate, and as a consequence generalize, is known to be an underlying weakness of NNs. This is especially relevant for dynamical systems and forecasting, which is typically an extrapolatory problem by nature. Thus models trained on historical data will generally fail to predict future events that are not represented in the training set. An additional limitation of deep learning is the lack of interpretability of the resulting models. While attempts have been made to interpret NN weights, network architectures are typically complicated with the number of parameters (or weights) far exceeding the original dimension of the dynamical system. The lack of interpretability also makes it difficult to generalize models to new datasets and parameter regimes. However, NN methods still have the potential to learn general, interpretable dynamical models if properly constrained or regularized. In addition to methods for discovering linear embeddings (18), deep learning has also been used for parameter estimation of partial differential equations (PDEs) (24, 25).

SINDy Autoencoders

We present a method for the simultaneous discovery of sparse dynamical models and coordinates that enable these simple representations. Our aim is to leverage the parsimony and interpretability of SINDy with the universal approximation capabilities of deep neural networks (51) to produce interpretable and generalizable models capable of extrapolation and forecasting. Our approach combines a SINDy model and a deep autoencoder network to perform a joint optimization that discovers intrinsic coordinates which have an associated parsimonious nonlinear dynamical model. The architecture is shown in Fig. 1. We again consider dynamical systems of the form **1**. While this dynamical model may be dense in terms of functions of the original measurement coordinates $\mathbf{x}$, our method seeks a set of reduced coordinates $\mathbf{z}(t) = \varphi(\mathbf{x}(t)) \in \mathbb{R}^d$ $(d \ll n)$ with an associated dynamical model

$$\frac{d}{dt}\mathbf{z}(t) = \mathbf{g}(\mathbf{z}(t)) \qquad [2]$$

that provides a parsimonious description of the dynamics; i.e., $\mathbf{g}$ contains only a few active terms. Along with the dynamical model, the method provides coordinate transforms φ, ψ that map the measurements to intrinsic coordinates via $\mathbf{z} = \varphi(\mathbf{x})$ (encoder) and back via $\mathbf{x} \approx \psi(\mathbf{z})$ (decoder).

The coordinate transformation is achieved using an autoencoder network architecture. The autoencoder is a feedforward neural network with a hidden layer that represents the intrinsic coordinates. Rather than performing a task such as prediction or classification, the network is trained to output an approximate reconstruction of its input, and the restrictions placed on the network architecture (e.g., the type, number, and size of the hidden layers) determine the properties of the intrinsic coordinates (17); these networks are known to produce nonlinear generalizations of PCA (16). A common choice is that the dimensionality of the intrinsic coordinates $\mathbf{z}$, determined by the number of units in the corresponding hidden layer, is much lower than that of the input data $\mathbf{x}$: In this case, the autoencoder learns a nonlinear embedding into a reduced latent space. Our network takes measurement data $\mathbf{x}(t) \in \mathbb{R}^n$ from a dynamical system as input and learns intrinsic coordinates $\mathbf{z}(t) \in \mathbb{R}^d$, where $d \ll n$ is chosen as a hyperparameter prior to training the network.

While autoencoders can be trained in isolation to discover useful coordinate transformations and dimensionality reductions, there is no guarantee that the intrinsic coordinates learned will have associated sparse dynamical models. We require the network to learn coordinates associated with parsimonious dynamics by simultaneously learning a SINDy model for the dynamics of the intrinsic coordinates $\mathbf{z}$. This regularization is achieved by constructing a library $\boldsymbol{\Theta}(\mathbf{z}) = [\boldsymbol{\theta}_1(\mathbf{z}), \boldsymbol{\theta}_2(\mathbf{z}), \ldots, \boldsymbol{\theta}_p(\mathbf{z})]$ of

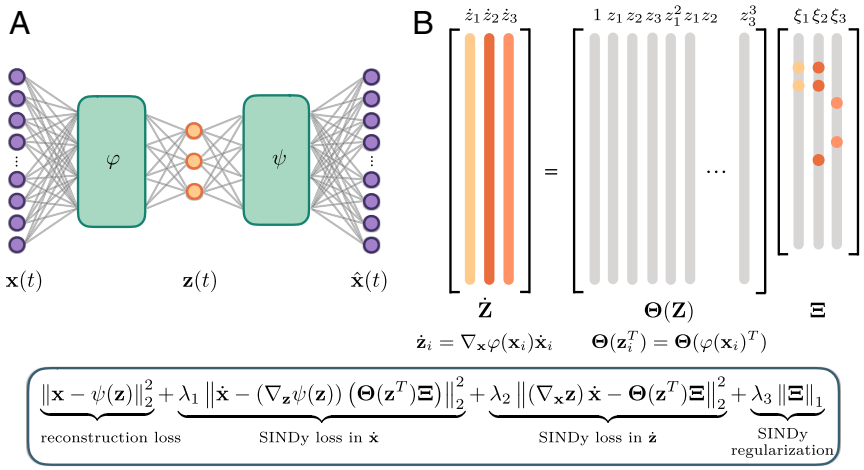

Fig. 1. Schematic of the SINDy autoencoder method for simultaneous discovery of coordinates and parsimonious dynamics. (*A*) An autoencoder architecture is used to discover intrinsic coordinates $\mathbf{z}$ from high-dimensional input data $\mathbf{x}$. The network consists of 2 components: an encoder $\varphi(\mathbf{x})$, which maps the input data to the intrinsic coordinates $\mathbf{z}$, and a decoder $\psi(\mathbf{z})$, which reconstructs $\mathbf{x}$ from the intrinsic coordinates. (*B*) A SINDy model captures the dynamics of the intrinsic coordinates. The active terms in the dynamics are identified by the nonzero elements in $\boldsymbol{\Xi}$, which are learned as part of the NN training. The time derivatives of $\mathbf{z}$ are calculated using the derivatives of $\mathbf{x}$ and the gradient of the encoder φ. *Inset* shows the pointwise loss function used to train the network. The loss function encourages the network to minimize both the autoencoder reconstruction error and the SINDy loss in $\mathbf{z}$ and $\mathbf{x}$. L_1 regularization on $\boldsymbol{\Xi}$ is also included to encourage parsimonious dynamics.

candidate basis functions, e.g., polynomials, and learning a sparse set of coefficients $\Xi = [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d]$ that defines the dynamical system

$$\frac{d}{dt}\mathbf{z}(t) = \mathbf{g}(\mathbf{z}(t)) = \boldsymbol{\Theta}(\mathbf{z}(t))\Xi.$$

While the library must be specified prior to training, the coefficients Ξ are learned with the NN parameters as part of the training procedure. Assuming derivatives $\dot{\mathbf{x}}(t)$ of the original states are available or can be computed, one can calculate the derivative of the encoder variables as $\dot{\mathbf{z}}(t) = \nabla_{\mathbf{x}}\varphi(\mathbf{x}(t))\dot{\mathbf{x}}(t)$ and enforce accurate prediction of the dynamics by incorporating the following term into the loss function:

$$\mathcal{L}_{d\mathbf{z}/dt} = \left\| \nabla_{\mathbf{x}}\varphi(\mathbf{x})\dot{\mathbf{x}} - \boldsymbol{\Theta}(\varphi(\mathbf{x})^T)\Xi \right\|_2^2. \quad [3]$$

This term uses the SINDy model along with the gradient of the encoder to encourage the learned dynamical model to accurately predict the time derivatives of the encoder variables. We include an additional term in the loss function that ensures SINDy predictions can be used to reconstruct the time derivatives of the original data:

$$\mathcal{L}_{d\mathbf{x}/dt} = \left\| \dot{\mathbf{x}} - (\nabla_{\mathbf{z}}\psi(\varphi(\mathbf{x})))\left(\boldsymbol{\Theta}(\varphi(\mathbf{x})^T)\Xi\right) \right\|_2^2. \quad [4]$$

We combine Eqs. **3** and **4** with the standard autoencoder loss

$$\mathcal{L}_{\text{recon}} = \|\mathbf{x} - \psi(\varphi(\mathbf{x}))\|_2^2,$$

which ensures that the autoencoder can accurately reconstruct the input data. We also include an L_1 regularization on the SINDy coefficients Ξ, which promotes sparsity of the coefficients and therefore encourages a parsimonious model for the dynamics. The combination of the above 4 terms gives the overall loss function

$$\mathcal{L}_{\text{recon}} + \lambda_1\mathcal{L}_{d\mathbf{x}/dt} + \lambda_2\mathcal{L}_{d\mathbf{z}/dt} + \lambda_3\mathcal{L}_{\text{reg}},$$

where the hyperparameters $\lambda_1, \lambda_2, \lambda_3$ determine the relative weighting of the 3 terms in the loss function.

In addition to the L_1 regularization, to obtain a model with only a few active terms, we also incorporate sequential thresholding into the training procedure as a proxy for L_0 sparsity (30). This technique is inspired by the original algorithm used for SINDy (8), which combined least-squares fitting with sequential thresholding to obtain a sparse model. To apply sequential thresholding during training, we specify a threshold that determines the minimum magnitude for coefficients in the SINDy model. At fixed intervals throughout the training, all coefficients below the threshold are set to zero and training resumes using only the terms left in the model. We train the network using the Adam optimizer (52). In addition to the loss function weightings and SINDy coefficient threshold, training requires the choice of several other hyperparameters including learning rate, number of intrinsic coordinates d, network size, and activation functions. Details of the training procedure are discussed in *SI Appendix*. Alternatively, one might attempt to learn the library functions using another neural network layer, a double sparse library (53), or kernel-based methods (54) for more flexible library representations.

Results

We demonstrate the success of the proposed method on 3 example systems: a high-dimensional system with the underlying dynamics generated from the canonical chaotic Lorenz system, a 2D reaction–diffusion system, and a 2D spatial representation (synthetic video) of the nonlinear pendulum. Results are shown in Fig. 2.

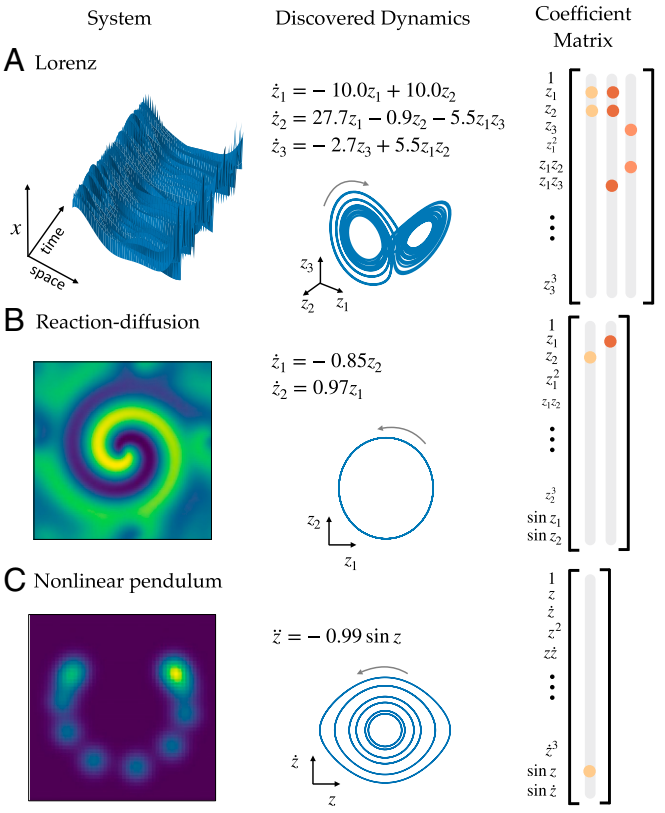

Fig. 2. Discovered models for examples. (*A*–*C*) Equations, SINDy coefficients Ξ, and attractors for Lorenz (*A*), reaction–diffusion (*B*), and nonlinear pendulum (*C*) systems.

Chaotic Lorenz System. We first construct a high-dimensional example problem with dynamics based on the chaotic Lorenz system. The Lorenz system is a canonical model used as a test case, with dynamics given by the following equations:

$$\dot{z}_1 = \sigma(z_2 - z_1) \quad [5a]$$
$$\dot{z}_2 = z_1(\rho - z_3) - z_2 \quad [5b]$$
$$\dot{z}_3 = z_1 z_2 - \beta z_3. \quad [5c]$$

The dynamics of the Lorenz system are chaotic and highly nonlinear, making it an ideal test problem for model discovery. To create a high-dimensional dataset based on this system, we choose 6 fixed spatial modes $\mathbf{u}_1, \ldots, \mathbf{u}_6 \in \mathbb{R}^{128}$, given by Legendre polynomials, and define

$$\mathbf{x}(t) = \mathbf{u}_1 z_1(t) + \mathbf{u}_2 z_2(t) + \mathbf{u}_3 z_3(t) + \mathbf{u}_4 z_1(t)^3 + \mathbf{u}_5 z_2(t)^3 + \mathbf{u}_6 z_3(t)^3. \quad [6]$$

This results in a dataset that is a nonlinear combination of the true Lorenz variables, shown in Fig. 3*A*. The spatial and temporal modes that combine to give the full dynamics are shown in Fig. 3*B*. Full details of how the dataset is generated are given in *SI Appendix*.

Fig. 3*D* shows the dynamical system discovered by the SINDy autoencoder. While the resulting model does not appear to match the original Lorenz system, the discovered model is parsimonious, with only 7 active terms, and the dynamics exhibit an attractor with a 2-lobe structure, similar to that of the original Lorenz attractor. Additionally, by choosing a suitable variable transformation the discovered model can be rewritten in the same form as the original Lorenz system. This demonstrates that the SINDy autoencoder is able to recover the correct sparsity

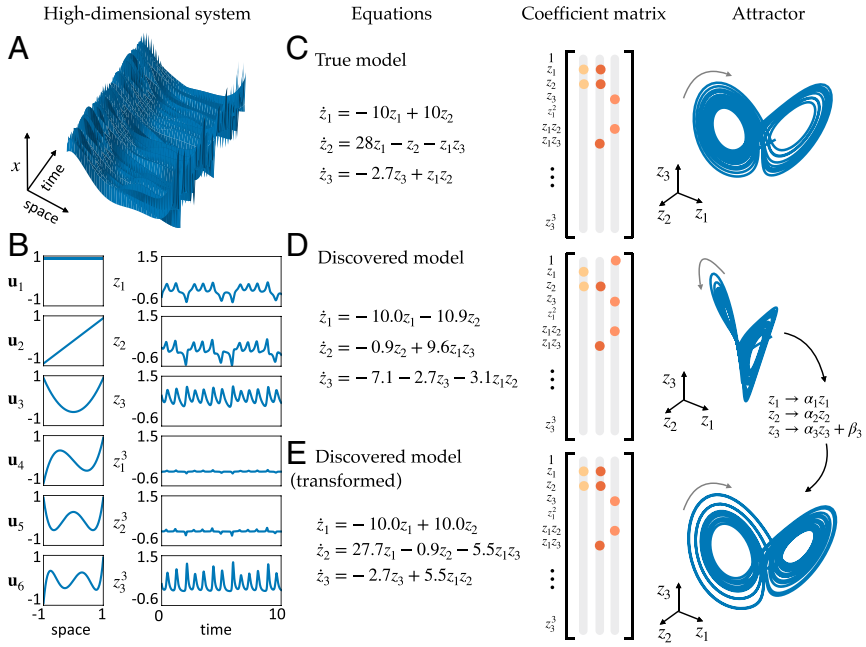

Fig. 3. Model results on the high-dimensional Lorenz example. (*A*) Trajectories of the chaotic Lorenz system ($\mathbf{z}(t) \in \mathbb{R}^3$) are used to create a high-dimensional dataset ($\mathbf{x}(t) \in \mathbb{R}^{128}$). (*B*) The spatial modes are created from the first 6 Legendre polynomials and the temporal modes are the variables in the Lorenz system and their cubes. The spatial and temporal modes are combined to create the high-dimensional dataset via **[6]**. (*C* and *D*) The equations, SINDy coefficients Ξ, and attractors for the original Lorenz system and a dynamical system discovered by the SINDy autoencoder. The attractors are constructed by simulating the dynamical system forward in time from a single initial condition. (*E*) Applying a suitable variable transformation to the system in *D* reveals a model with the same sparsity pattern as the original Lorenz system. The parameters are close in value to the original system, with the exception of an arbitrary scaling, and the attractor has a similar structure to the original system.

pattern of the dynamics. The coefficients of the discovered model are close to the original parameters of the Lorenz system, up to an arbitrary scaling, which accounts for the difference in magnitude of the coefficients of $z_1 z_3$ in the second equation and $z_1 z_2$ in the third equation.

On test trajectories from 100 initial conditions sampled from the training distribution, the relative L_2 errors in predicting $\mathbf{x}$, $\dot{\mathbf{x}}$, and $\dot{\mathbf{z}}$ are 3×10^{-5}, 2×10^{-4}, and 7×10^{-4}, respectively. For initial conditions outside of the training distribution, the model has higher relative L_2 errors on 100 test trajectories of 0.016, 0.126, and 0.078 for $\mathbf{x}$, $\dot{\mathbf{x}}$, and $\dot{\mathbf{z}}$. In both cases, the resulting SINDy models produce dynamics that are qualitatively similar to the true trajectories, although due to the chaotic nature of the Lorenz system and its sensitivity to parameters and initial conditions, the phase of most predicted trajectories diverges from the true trajectories after a short period. Improved prediction over a longer duration may be achieved by increased parameter refinement or training with longer trajectories.

Reaction–Diffusion. In practice, many high-dimensional datasets of interest come from dynamics governed by PDEs with more complicated interactions between spatial and temporal dynamics. To test the method on data generated by a PDE, we consider a lambda–omega reaction–diffusion system governed by

$$u_t = (1 - (u^2 + v^2))u + \beta(u^2 + v^2)v + d_1(u_{xx} + u_{yy})$$
$$v_t = -\beta(u^2 + v^2)u + (1 - (u^2 + v^2))v + d_2(v_{xx} + v_{yy})$$

with $d_1, d_2 = 0.1$ and $\beta = 1$. This set of equations generates a spiral wave formation, whose behavior can be approximately captured by 2 oscillating spatial modes. We apply our method to snapshots of $u(x, y, t)$ generated by the above equations. Snapshots are collected at discretized points of the xy domain, resulting in a high-dimensional input dataset with $n = 10^4$.

We train the SINDy autoencoder with $d = 2$. The resulting model is shown in Fig. 2*B*. The network discovers a model with nonlinear oscillatory dynamics. On test data, the relative L_2 error for the input data $\mathbf{x}$ and the input derivatives $\dot{\mathbf{x}}$ is 0.016. The relative L_2 error for $\dot{\mathbf{z}}$ is 0.002. Simulation of the dynamical model accurately captures the low-dimensional dynamics, with relative L_2 error of $\mathbf{z}$ totaling 1×10^{-4}.

Nonlinear Pendulum. As a final example, we consider a simulated video of a nonlinear pendulum. The nonlinear pendulum is governed by the following second-order differential equation:

$$\ddot{z} = -\sin z.$$

We simulate the system from several initial conditions and generate a series of snapshot images with a 2D Gaussian centered at the center of mass, determined by the pendulum's angle z. This series of images is the high-dimensional data input to the autoencoder. Despite the fact that the position of the pendulum can be represented by a simple 1-dimensional variable, methods such as PCA are unable to obtain a low-dimensional representation of this dataset. A nonlinear autoencoder, however, is able to discover a 1-dimensional representation of the dataset.

For this example, we use a second-order SINDy model with a library of functions including the first derivatives $\dot{\mathbf{z}}$ to predict the second derivative $\ddot{\mathbf{z}}$. This approach is the same as with a first-order SINDy model but requires estimates of the second derivatives as well. Second-order gradients of the encoder and decoder are therefore also required. Computation of the derivatives is discussed in *SI Appendix*.

The SINDy autoencoder is trained with $d = 1$. Of the 10 training instances, 5 correctly identify the nonlinear pendulum equation. We calculate test error on trajectories from 50 randomly chosen initial conditions sampled from the same distribution as

the training data. The best model has a relative L_2 error of 8×10^{-4} for the decoder reconstruction of the input $\mathbf{x}$. The relative L_2 errors of the SINDy model predictions for $\ddot{\mathbf{x}}$ and $\ddot{\mathbf{z}}$ are 3×10^{-4} and 2×10^{-2}, respectively.

Discussion

We have presented a data-driven method for discovering interpretable, low-dimensional dynamical models and their associated coordinates from high-dimensional data. The simultaneous discovery of both is critical for generating dynamical models that are sparse and hence interpretable. Our approach takes advantage of the power of NNs by using a flexible autoencoder architecture to discover nonlinear coordinate transformations that enable the discovery of parsimonious, nonlinear governing equations. This work addresses a major limitation of prior approaches for model discovery, which is that the proper choice of measurement coordinates is often unknown. We demonstrate this method on 3 example systems, showing that it is able to identify coordinates associated with parsimonious dynamical equations. Our code is publicly available at http://github.com/kpchamp/SindyAutoencoders (55).

A current limitation of our approach is the requirement for clean measurement data that are approximately noise-free. Fitting a continuous-time dynamical system with SINDy requires reasonable estimates of the derivatives, which may be difficult to obtain from noisy data. While this represents a challenge, approaches for estimating derivatives from noisy data such as the total variation regularized derivative can prove useful in providing derivative estimates (56). Moreover, there are emerging NN architectures explicitly constructed for separating signals from noise (57), which can be used as a preprocessing step in the data-driven discovery process advocated here. Alternatively our method can be used to fit a discrete-time dynamical system, in which case derivative estimates are not required. It is also possible to use the integral formulation of SINDy to abate noise sensitivity (39).

A major problem with deep-learning approaches is that models are typically neither interpretable nor generalizable. Specifically, NNs trained solely for prediction may fail to generalize to classes of behaviors not seen in the training set. We have demonstrated an approach for using NNs to obtain classically interpretable models through the discovery of low-dimensional dynamical systems, which are well studied and often have physical interpretations. While the autoencoder network still has the same limited interpretability and generalizability as other NNs, the dynamical model has the potential to generalize to other parameter regimes of the dynamics. Although the coordinate transformation learned by the autoencoder may not generalize to data regimes far from the original training set, if the dynamics are known, the autoencoder can be retrained on new data with fixed terms in the latent dynamics space (see *SI Appendix* for discussion). The problem of relearning a coordinate transformation for a system with known dynamics is simplified from the original challenge of learning the correct form of the underlying dynamics without knowledge of the proper coordinate transformation.

The challenge of utilizing NNs to answer scientific questions requires careful consideration of their strengths and limitations. While advances in deep learning and computing power present a tremendous opportunity for new scientific breakthroughs, care must be taken to ensure that valid conclusions are drawn from the results. One promising strategy is to combine machine-learning approaches with well-established domain knowledge: For instance, physics-informed learning leverages physical assumptions into NN architectures and training methods. Methods that provide interpretable models have the potential to enable new discoveries in data-rich fields. This work introduced a flexible framework for using NNs to discover models that are interpretable from a standard dynamical systems perspective. While this formulation used an autoencoder to achieve full state reconstruction, similar architectures could be used to discover embeddings that satisfy alternative conditions. In the future, this approach could be adapted using domain knowledge to discover new models in specific fields.

ACKNOWLEDGMENTS. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant DGE-1256082. We also acknowledge support from the Defense Advanced Research Projects Agency (PA-18-01-FP-125) and the Army Research Office (W911NF-17-1-0306 and W911NF-19-1-0045). This work was facilitated through the use of advanced computational, storage, and networking infrastructure provided by Amazon Web Services cloud computing credits funded by the Student Technology Fee at the University of Washington. This research was funded in part by the Argonne Leadership Computing Facility, which is a Department of Energy Office of Science User Facility supported under Contract DE-AC02-06CH11357. We also thank Jean-Christophe Loiseau and Karthik Duraisamy for valuable discussions about sparse dynamical systems and autoencoders.

1. J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* **104**, 9943–9948 (2007).
2. C. Yao, E. M. Bollt, Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems. *Physica D* **227**, 78–99 (2007).
3. M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
4. C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D. Henningson, Spectral analysis of nonlinear flows. *J. Fluid Mech.* **645**, 115–127 (2009).
5. P. J. Schmid, Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28 (2010).
6. P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**, 483–531 (2015).
7. B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **306**, 196–215 (2016).
8. S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 3932–3937 (2016).
9. J. N. Kutz, S. L. Brunton, B. W. Brunton, J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (Society for Industrial and Applied Mathematics, 2016).
10. S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz, Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614 (2017).
11. O. Yair, R. Talmon, R. R. Coifman, I. G. Kevrekidis, Reconstruction of normal forms by learning informed observation geometries from data. *Proc. Natl. Acad. Sci. U.S.A.* **114**, E7865–E7874 (2017).
12. K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377 (2018).
13. J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
14. P. W. Battaglia *et al.*, Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261 (4 June 2018).
15. H. Schaeffer, R. Caflisch, C. D. Hauck, S. Osher, Sparse dynamics for partial differential equations. *Proc. Natl. Acad. Sci. U.S.A.* **110**, 6634–6639 (2013).
16. P. Baldi, K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.* **2**, 53–58 (1989).
17. I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, *Deep Learning* (MIT Press, 2016), vol. 1.
18. B. Lusch, J. N. Kutz, S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 4950 (2018).
19. A. Mardt, L. Pasquali, H. Wu, F. Noé, VAMPnets: Deep learning of molecular kinetics. *Nat. Commun.* **9**, 5 (2018).
20. P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, P. Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long-short term memory networks. *Proc. R. Soc. A* **474**, 20170844 (2018).
21. C. Wehmeyer, F. Noé, Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **148**, 241703 (2018).
22. E. Yeung, S. Kundu, N. Hodas, "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems" in *2019 American Control Conference* (IEEE, New York), pp. 4832–4839.
23. N. Takeishi, Y. Kawahara, T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition" in *Advances in Neural Information Processing Systems 30* (Curran Assoc Inc., Red Hook, NY), pp. 1130–1140.
24. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv:1711.10566 (28 November 2017).

25. M. Raissi, P. Perdikaris, G. E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems. arXiv:1801.01236 (4 January 2018).
26. Y. Bar-Sinai, S. Hoyer, J. Hickey, M. P. Brenner, Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15344–15349 (2019).
27. J. C. Loiseau, S. L. Brunton, Constrained sparse Galerkin regression. *J. Fluid Mech.* **838**, 42–67 (2018).
28. M. Milano, P. Koumoutsakos, Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182**, 1–26 (2002).
29. I. Mezic, Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dyn.* **41**, 309–325 (2005).
30. P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, A. Y. Aravkin, A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access* **7**, 1404–1423 (2019).
31. L. Zhang, H. Schaeffer, On the convergence of the SINDy algorithm. *Multiscale Model. Simul.* **17**, 948–972 (2019).
32. M. Sorokina, S. Sygletos, S. Turitsyn, Sparse identification for nonlinear optical communication systems: SINO method. *Opt. Express* **24**, 30433 (2016).
33. M. Hoffmann, C. Fröhner, F. Noé, Reactive SINDy: Discovering governing reactions from concentration data. *J. Chem. Phys.* **150**, 025101 (2019).
34. M. Dam, M. Brøns, J. Juul Rasmussen, V. Naulin, J. S. Hesthaven, Sparse identification of a predator-prey system from simulation data of a convection model. *Phys. Plasmas* **24**, 022310 (2017).
35. Z. Lai, S. Nagarajaiah, Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mech. Syst. Signal Process.* **117**, 813–842 (2019).
36. E. Kaiser, J. N. Kutz, S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **474**, 20180335 (2018).
37. H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. A* **473**, 20160446 (2017).
38. N. M. Mangan, S. L. Brunton, J. L. Proctor, J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multiscale Commun.* **2**, 52–63 (2016).
39. H. Schaeffer, S. G. McCalla, Sparse model selection via integral terms. *Phys. Rev. E* **96**, 023302 (2017).
40. G. Tran, R. Ward, Exact recovery of chaotic systems from highly corrupted data. *Multiscale Model. Simul.* **15**, 1108–1129 (2017).
41. H. Schaeffer, G. Tran, R. Ward, Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Math.* **78**, 3279–3295 (2018).
42. N. M. Mangan, J. N. Kutz, S. L. Brunton, J. L. Proctor, Model selection for dynamical systems via sparse regression and information criteria. *Proc. R. Soc. A* **473**, 20170009 (2017).
43. R. Gonzalez-Garcia, R. Rico-Martinez, I. Kevrekidis, Identification of distributed parameter systems: A neural net based approach. *Comput. Chem. Eng.* **22** (suppl. 1), S965–S968 (1998).
44. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
45. K. T. Carlberg *et al.*, Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning. *J. Comput. Phys.* **395**, 105–124 (2019).
46. F. J. Gonzalez, M. Balajewicz, Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. arXiv:1808.01346 (22 August 2018).
47. K. Lee, K. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. arXiv:1812.08373 (20 December 2018).
48. M. O. Williams, I. G. Kevrekidis, C. W. Rowley, A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.* **25**, 1307–1346 (2015).
49. S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, J. N. Kutz, Chaos as an intermittently forced linear system. *Nat. Commun.* **8**, 19 (2017).
50. H. Wu, F. Noé, Variational approach for learning Markov processes from time series data. *J. Nonlinear Sci.* **29**, 1432–1467 (2019).
51. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
52. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization. arXiv:1412.6980 (30 January 2017).
53. R. Rubinstein, M. Zibulevsky, M. Elad, Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Trans. Signal Process.* **58**, 1553–1564 (2009).
54. H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, R. Chellappa (2012) "Kernel dictionary learning" in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, New York), pp. 2021–2024.
55. K. Champion, SindyAutoencoders. GitHub. https://github.com/kpchamp/SindyAutoencoders. Deposited 10 October 2019.
56. R. Chartrand, Numerical differentiation of noisy, nonsmooth data. *ISRN Appl. Math.* **2011**, 1–11 (2017).
57. S. H. Rudy, J. N. Kutz, S. L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *J. Comput. Phys.* **396**, 483–506 (2019).