Learning a reduced basis of dynamical systems using an autoencoder

David Sondak[*] and Pavlos Protopapas
Institute for Applied Computational Science, Harvard University, Cambridge, Massachusetts 02138, USA

(Received 20 October 2020; revised 28 April 2021; accepted 19 July 2021; published 3 September 2021)

Machine learning models have emerged as powerful tools in physics and engineering. In this work, we use an autoencoder with latent space penalization to discover approximate finite-dimensional manifolds of two canonical partial differential equations. We test this method on the Kuramoto-Sivashinsky (K-S), Korteweg-de Vries (KdV), and damped KdV equations. We show that the resulting optimal latent space of the K-S equation is consistent with the dimension of the inertial manifold. We then uncover a nonlinear basis representing the manifold of the latent space for the K-S equation. The results for the KdV equation show that it is more difficult to recover a reduced latent space, which is consistent with the truly infinite-dimensional dynamics of the KdV equation. In the case of the damped KdV equation, we find that the number of active dimensions decreases with increasing damping coefficient.

DOI: 10.1103/PhysRevE.104.034202

I. INTRODUCTION

Evolution of physical and engineering systems is generally expressed as nonlinear partial differential equations (PDEs). These PDEs are able to capture a wide range of complex phenomena and are therefore indispensable for making predictions of scientific interest. However, most PDEs of practical interest are not analytically tractable. Highly efficient numerical methods can obtain solutions to these PDEs but are severely limited by the strong multiscale nature of the underlying dynamics and limitations of hardware resources. There is therefore considerable interest in the development of reduced models that capture only the most important dynamics of the physical phenomenon of interest [1–3]. In recent years, machine learning algorithms have been borrowed from the computer vision community and adapted for physical applications [4–12], offering an enticing approach for blending data with physical principles. In fact, a key goal in merging machine learning with physics problems is to embed known physical laws into machine learning algorithms [7]. In the present work we use a combination of neural networks and techniques from data analysis to find a dynamically relevant manifold of canonical PDEs using autoencoders. Other recent work has applied autoencoders to learning inertial manifolds of PDEs in a physically meaningful manner [13–17]. In particular, Ref. [16] introduces the Hybrid Neural Network (HNN), in which an autoencoder is used to learn the difference between the data and a linear projection onto the principle component analysis basis. The HNN embeds translation invariance and energy conservation and learns the dynamics on the inertial manifold of the Kuramoto-Sivashinsky equation. In the present work, the traditional mean-squared error loss function between the input and reconstruction of the autoencoder is augmented with the sparsity-promoting mean absolute error loss function, which is applied to the latent space. Sparsity-promoting norms have been studied extensively in statistics for sparse regression [18,19] and mathematics via compressed sensing to reconstruct signals from incomplete data [20–24]. Recent work has also applied sparse regression and compressed sensing to discover equations from data [15,25]. In our approach, the latent space of the trained autoencoder only contains the minimal dimensions needed to reconstruct the solution. This approach is tested on two equations with known reduced dynamics [the Kuramoto-Sivashinsky (K-S) equation and the damped Korteweg-de Vries (KdV) equation] and one equation whose dynamics are truly infinite dimensional (the undamped KdV equation). In the case of the Kuramoto-Sivashinsky equation, the trained autoencoder and latent space is used to find a nonlinear reduced solution basis whose dimension is consistent with that of the inertial manifold.

II. GOVERNING EQUATIONS AND DATASETS

The K-S equation is

$$u_t + uu_x + u_{xx} + u_{xxxx} = 0, \tag{1}$$

where $u = u(x, t)$ is the solution field, $x \in [0, L]$, and $t \in \mathbb{R}^+$. The final integration time is denoted by T. Equation (1) is subject to periodic boundary conditions $u(0, t) = u(L, t)$ and initial condition $u_0 = \cos(\frac{2\pi}{L}x)[1 + \sin(\frac{2\pi}{L}x)]$. The K-S equation has its roots in physics [26–29] and is a frequently studied equation in mathematics [30–37]. The dynamics of the K-S equation are confined to an inertial manifold [31]. That is, despite the K-S equation being a nonintegrable equation whose solutions exhibit spatiotemporal chaos, the underlying dynamics are exponentially attracted to a finite-dimensional manifold. The dimension of the inertial manifold increases with the bifurcation parameter L.

[*] Present address: Dassault Systèmes Simulia Corp., 175 Wyman St, Waltham, MA 02451, USA; david.sondak@3ds.com

©2021 American Physical Society

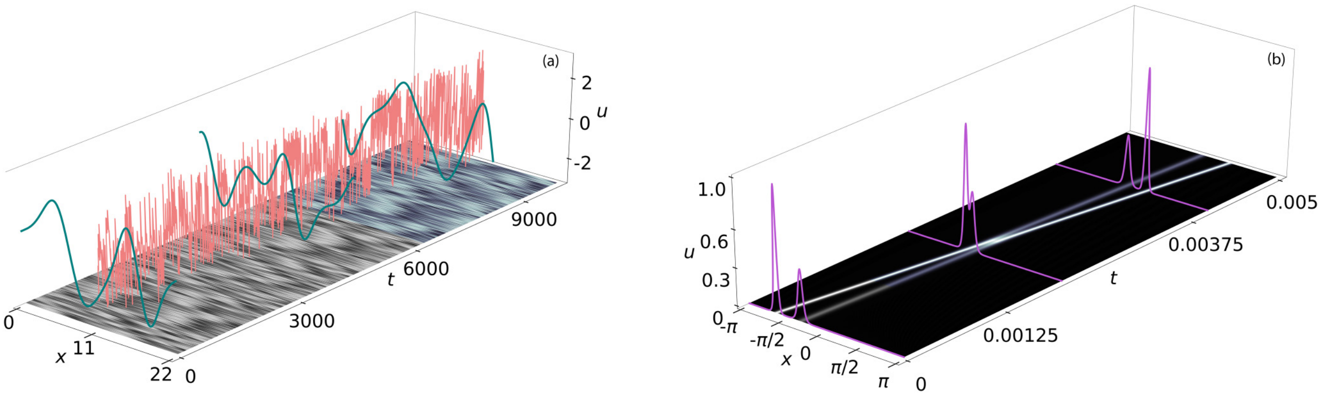

FIG. 1. Datasets from the K-S and KdV equations used in this work. (a) Contours of the K-S equation in space-time with three spatial snapshots at different points in time and a snapshot in time at the center of the domain. The gray shaded area of the contour was not used in the training. (b) Space-time contours of the KdV equation with spatial snapshots in time. The shading is the same as for the K-S equation.

The damped KdV equation, also considered in this work, is

$$u_t + uu_x + u_{xxx} - \eta u_{xx} = 0, \qquad (2)$$

where $x \in [-\pi, \pi]$ and $\eta \geqslant 0$ is a damping coefficient. The KdV equation is subject to periodic boundary conditions $u(-\pi, t) = u(\pi, t)$ and initial condition $u_0 = 3A^2\text{sech}^2[A(x+2)/2] + 3B^2\text{sech}^2[B(x+1)/2]$ with $A = 16$ and $B = 25$. The classical KdV equation is recovered for $\eta = 0$. Similarly to the K-S equation, the KdV equation is a paradigmatic equation in mathematical physics [38,39], leading to the discovery of solitons, which are a direct bridge between observed coherent structures in PDEs and nature. The dynamics of the undamped KdV equation are truly infinite dimensional while those of the damped KdV equation are finite dimensional [39].

The datasets for this work were generated by solving (1) and (2) using an exponential time-differencing fourth-order Runge-Kutta method [40] and a pseudospectral Fourier method in space. Spatial snapshots of these solution fields are used as input to the autoencoder. Figure 1 shows examples of the K-S dataset [Fig. 1(a)] and the KdV dataset [Fig. 1(b)]. The numerical parameters used for the cases in this work are presented in Table I.

III. METHODOLOGY

Autoencoders are a self-supervised neural network architecture that can be used to find a low-dimensional manifold that represents the data [41,42]. The input to the encoder is mapped to a lower-dimensional space called the latent space. The latent space is then expanded through the decoder to reproduce the input to the encoder. An autoencoder with linear activation functions can be shown to be equivalent to the singular value decomposition [43]. In the present work, the input is a snapshot of the solution field obtained from a high-fidelity numerical simulation that used N points in space and N_t points in time. A snapshot in space is denoted by $\mathbf{u}_n = \mathbf{u}(t_n)$ for $n = 1, \ldots, N_t$ and $\mathbf{u} \in \mathbb{R}^N$ is a vector representing the solution at N discrete points in space. This snapshot is mapped to a latent space of dimension N_z with $z_j(t_n)$ the jth component of the latent space corresponding to snapshot n. The reconstructed output of the autoencoder is denoted by $\widehat{\mathbf{u}}_n$. The weights and biases associated with each node of the autoencoder are tuned to minimize the total loss,

$$\mathcal{L} = \mathcal{L}_u + \lambda \mathcal{L}_z, \qquad (3)$$

where

$$\mathcal{L}_u(u, \widehat{u}) = \frac{1}{N_t N} \sum_{n=1}^{N_t} \sum_{i=1}^{N} [u(x_i, t_n) - \widehat{u}(x_i, t_n)^2] \qquad (4)$$

is the mean-squared error (MSE) reconstruction loss and

$$\mathcal{L}_z = \frac{1}{N_t N_z} \sum_{n=1}^{N_t} \sum_{j=1}^{N_z} |z_j(t_n)| \qquad (5)$$

is the mean absolute error (MAE) penalization loss on the latent dimensions. The sparsity of the latent space is controlled by the regularization parameter $\lambda \geqslant 0$. A classical autoencoder corresponds to $\lambda = 0$. The dimension of the latent space, N_z, is not known *a priori*, but the MAE penalization promotes sparsity in the latent dimensions while the MSE loss boosts the reconstruction performance. The appropriate value of λ will therefore restrict the latent space to the dimension necessary for a good reconstruction. Figure 2 depicts the autoencoder and loss functions used in this work. All networks used in the current work used fully connected networks with sinusoidal activation functions. For conciseness, we denote the encoder architecture by $N \to H_1 \to H_2 \cdots \to H_D \to N_z$,

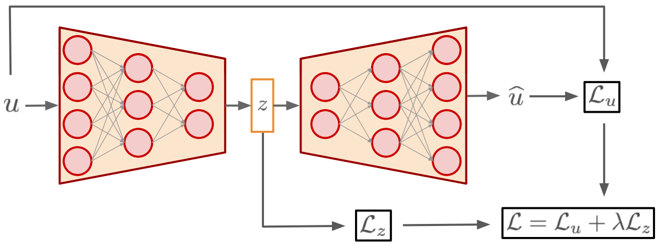

FIG. 2. The autoencoder architecture with latent space penalization. The reconstruction loss L_u (4) is combined with a mean absolute error loss L_z (5). The result is a solution reconstruction that uses only relevant latent dimensions.

TABLE I. Details of the numerical simulations that were used to generate the datasets. The damping coefficient η only applies to the damped KdV equation. N represents the number of discrete points in the physical domain. The final integration time is T and the constant time step is given by Δt.

Case	Equation	Domain	η	N	T	Δt
1	K-S	[0, 22]	—	512	4×10^4	0.125
2	K-S	[0, 22]	—	1024	4×10^4	0.125
3	K-S	[0, 26]	—	1024	4×10^4	0.125
4	K-S	[0, 30]	—	1024	4×10^4	0.125
5	K-S	[0, 35]	—	1024	4×10^4	0.125
6	K-S	[0, 43]	—	1024	4×10^4	0.125
7	K-S	[0, 45]	—	1024	4×10^4	0.125
8	K-S	[0, 50]	—	1024	4×10^4	0.125
9	KdV	$[-\pi, \pi]$	0	512	5×10^{-3}	0.125
10	KdV	$[-\pi, \pi]$	0.01	512	10^{-2}	10^{-7}
11	KdV	$[-\pi, \pi]$	0.05	512	10^{-2}	10^{-7}
12	KdV	$[-\pi, \pi]$	0.1	512	10^{-2}	10^{-7}
13	KdV	$[-\pi, \pi]$	0.25	512	10^{-2}	10^{-7}
14	KdV	$[-\pi, \pi]$	0.63	512	10^{-2}	10^{-7}
15	KdV	$[-\pi, \pi]$	1.0	512	10^{-2}	10^{-7}
16	KdV	$[-\pi, \pi]$	1.5	512	10^{-2}	10^{-7}
17	KdV	$[-\pi, \pi]$	1.58	512	10^{-2}	10^{-7}
18	KdV	$[-\pi, \pi]$	2.0	512	10^{-2}	10^{-7}
19	KdV	$[-\pi, \pi]$	2.5	512	10^{-2}	10^{-7}
20	KdV	$[-\pi, \pi]$	3.0	512	10^{-2}	10^{-7}
21	KdV	$[-\pi, \pi]$	3.5	512	10^{-2}	10^{-7}
22	KdV	$[-\pi, \pi]$	3.98	512	10^{-2}	10^{-7}
23	KdV	$[-\pi, \pi]$	4.0	512	10^{-2}	10^{-7}
24	KdV	$[-\pi, \pi]$	5.7	512	10^{-2}	10^{-7}
25	KdV	$[-\pi, \pi]$	10.0	512	10^{-2}	10^{-7}
26	KdV	$[-\pi, \pi]$	15.0	512	10^{-2}	10^{-7}
27	KdV	$[-\pi, \pi]$	20.0	512	10^{-2}	10^{-7}
28	KdV	$[-\pi, \pi]$	30.0	512	10^{-2}	10^{-7}
29	KdV	$[-\pi, \pi]$	50.0	512	10^{-2}	10^{-7}
30	KdV	$[-\pi, \pi]$	100.0	512	10^{-2}	10^{-7}

where H_k represents the number of nodes in layer k and D is the number of hidden layers. The decoder uses the reverse form of the encoder portion. The autoencoders were trained used gradient descent with gradient clipping [44] to limit the maximum gradient to 10. The Adamax optimizer [45] was used in all experiments. The transient portion of the dataset was excluded from both the training and validation sets. In general, 80% of the remaining dataset was retained for training and 20% was used for validation. Table II contains specific details on all autoencoder architectures used in this work including their hyperparameters.

IV. RESULTS

Autoencoders were trained on datasets generated from the K-S equation (1) and the KdV equation (2). The cases considered in this work are summarized in Tables I and II, respectively. Each case was run for a range of regularization parameter values. For each value of λ, the dataset was split into a training and validation set and the MSE (4) was monitored on the validation set during training. Two models were saved for each experiment. One model was saved at the minimum of the training loss curve while the other model was saved at the minimum of the MSE loss on the validation set. The model with the lowest MSE loss on the validation set at each value of λ, $\mathcal{L}_u^*(u_{\text{valid}}, \widehat{u}_{\text{valid}})$, was taken to represent the optimal regularization parameter for that case and was used for analysis.

A. Kuramoto-Sivashinsky equation

Using the K-S dataset with $L = 22$, the autoencoder was trained to find $\widehat{u}(x, t)$ with the total loss (3) for $\lambda \in [10^{-4}, 2]$. The architecture of the encoder was $512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32$, corresponding to $N_z = 32$. Figure 3(a) presents $\mathcal{L}_u^*(u_{\text{valid}}, \widehat{u}_{\text{valid}})$ for different values of λ and shows a minimum at $\lambda = 0.398$. Figure 3(b) shows the reconstruction by the trained autoencoder using $\lambda = 0.398$ on a snapshot from the validation set. More insight can be obtained by passing each snapshot through the trained network and extracting the latent dimensions corresponding to each snapshot. This process results in N_t vectors (one for each snapshot), each of size N_z. Figure 4 shows that of the 32 latent dimensions, only 10 are consistently active. This is consistent with, but slightly larger than, the known dimension of the inertial manifold for the K-S equation [37,46]. The interquartile range (IQR) can be used as an indicator of the variability of the latent dimensions. When normalized by the largest IQR (the most active

TABLE II. Parameters used for the trained autoencoders. The cases correspond to those in Table I. The Start Index column represents the snapshot index from which the training and validation sets were taken. Data before this index were not used.

Case	Architecture	Learning rate	Batch size	Epochs	Start Index
1	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	40	3×10^4	5×10^4
2	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
3	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
4	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
5	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
6	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
7	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
8	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	3×10^4	5×10^4
9 ($\lambda = 0$)	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	5×10^{-3}	100	10^5	10^4
9 ($\lambda > 0$)	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	5×10^{-3}	100	10^5	10^4
10	(D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	5×10^4	10^4
11–30	(D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	7.5×10^4	10^4

latent dimension) a very clear separation between "active" and "nonactive" dimensions emerges as shown in the bottom of Figure 4. The largest IQR of the remaining 22 dimensions is 0.238 of the most active dimension.

The trained autoencoder model also provides a way to develop a nonlinear basis for the learned manifold using the technique of activation maximization. The idea behind this technique is to determine the input that maximizes the output of a specific node in the neural network. In the present work, we were interested in the inputs that would maximize each latent dimension. The input that maximized a given latent dimension was interpreted as a component of the basis of the low-dimensional manifold. The input that maximizes latent dimension z_j is determined from

$$b_j = \arg\max_v S_j(v), \quad j = 1, \dots, N_z, \tag{6}$$

where

$$S_j = z_j(v) - \beta R_{\mathrm{TV}}(v) \tag{7}$$

and

$$R_{\mathrm{TV}}(v) = \frac{1}{N-1} \sum_{i=1}^{N-1} [v(x_{i+1}) - v(x_i)]^2 \tag{8}$$

is a regularization used to smooth the resulting field with the regularization parameter $\beta \geqslant 0$. The function evaluation $z_j(v)$ corresponds to an evaluation of the encoder portion of the trained autoencoder with input v. Note that v is not a temporal snapshot of the dataset but instead represents an arbitrary input to the autoencoder. Gradient ascent was used to solve (6),

$$v_j^{(l+1)} = v_j^{(l)} + \gamma \frac{\partial S_j}{\partial v_j}. \tag{9}$$

The step size γ was set to 1 and the regularization strength β was set to 3. The maximization of each latent dimension was initialized from a random distribution in space where each point was drawn uniformly in [0, 1]. The result is a basis for the reduced manifold. As an alternative to activation maximization, one could consider the prediction from a single latent component propagated through the trained decoder. If the decoder was perfect, then the result may be very similar to

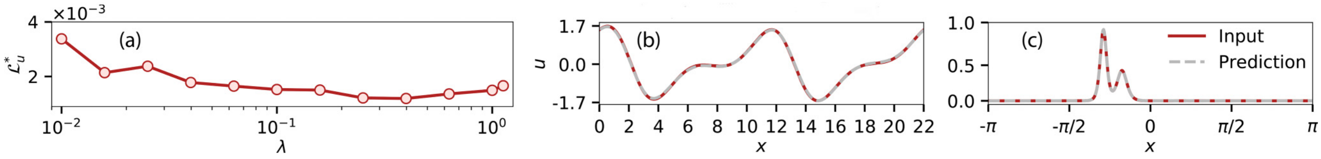

FIG. 3. (a) Reconstruction loss on the validation set for the K-S equation with $L = 22$ across different values of penalization parameter λ. In each case the network architecture was 512 → 256 → 128 → 64 → 32. The minimum value occurs at $\lambda = 0.398$. (b) Solution reconstruction and input solution for the K-S equation. (c) Solution reconstruction and input solution for the KdV equation.

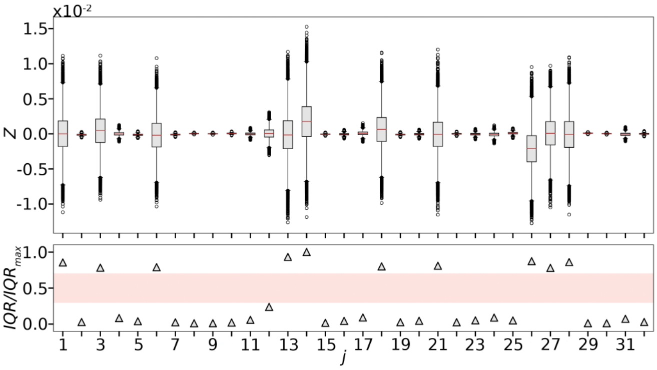

FIG. 4. Active latent dimensions for the model with the lowest total loss (3) ($\lambda = 0.398$). Top: Box plots of each dimension of the latent space generated by passing all snapshots through the autoencoder and extracting the latent space. Bottom: Normalized IQR for each dimension showing a gap between the 10 active and remaining nonactive dimensions.

activation maximization. In the present work, we only present results from activation maximization. Figures 5(a) and 5(c) show the components of the basis and their power spectra, respectively. The power spectra clearly show that the components of the discovered basis consist of a handful of distinct modes. In contrast to the active dimensions, Figures 5(b) and 5(d) show that the inputs that maximize the nonactive dimensions are constants near zero.

Finally, we compute the optimal number of latent dimensions for $L = [22, 26, 30, 35, 43, 45, 50]$ and repeat the experiment 10 times for each value of L. Each experiment was run under the exact same conditions and using the same data. The number of input points for each snapshot was 1024, which was selected so that the dimension of the inertial manifold is captured for all values of L considered. The network architecture was the same across cases (see Table II) and used 64 dimensions for the latent space. Given that stochastic gradient descent and a finite data set was used to train the network, we expect some variability in the final model and the corresponding optimal number of latent dimensions. Figure 6 depicts the scaling of the number of latent dimensions with L, along with uncertainty bounds, and shows a nearly linear scaling, consistent with the scaling of the number of active modes [47].

B. Undamped and damped KdV equations

In contrast to the K-S equation, the undamped KdV equation does not possess an inertial manifold and, moreover, the dynamics are truly infinite dimensional [39]. The undamped KdV equation therefore provides a test for the method in which the dimension of the optimal latent space is expected to be full. One may expect that this corresponds to an optimal latent space penalization of zero. However, as

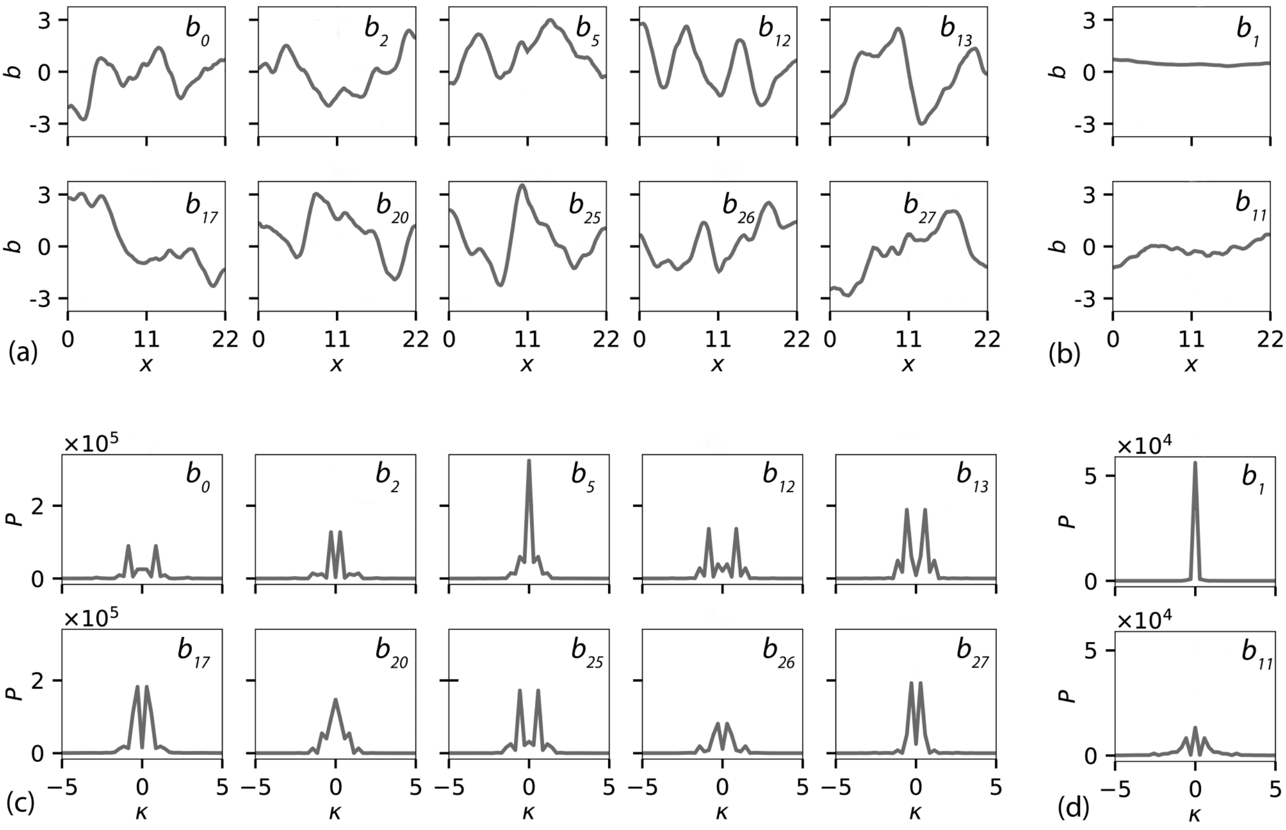

FIG. 5. (a) The nonlinear basis for the 10-dimensional manifold found for the K-S equation with $L = 22$. (b) Examples of the nonactive dimensions. (c) Power spectrum of the basis for the active dimensions. (d) Power spectrum of two nonactive dimensions. Wave numbers in the power spectra have been truncated to highlight the peaks in the power spectra. There are no peaks for $|\kappa| > 5$.

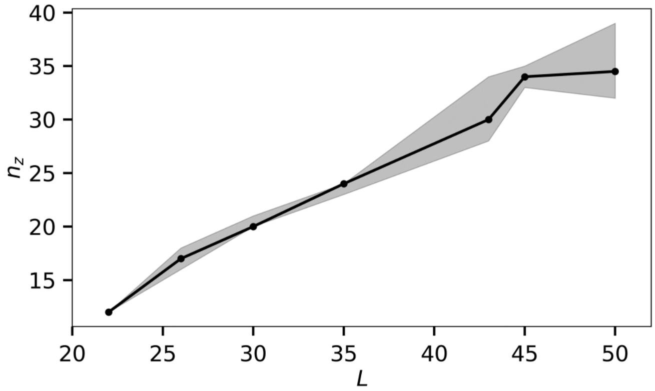

FIG. 6. Scaling of the optimal number of latent dimensions with domain size L for the K-S equation. The black circles indicate the median value over 10 runs and the region between the maximum and minimum values is shaded.

will be shown, this is not necessarily correct. The autoencoder attempts to learn the correct dimensionality of the underlying manifold regardless of the regularization used. Figure 3(c) presents a snapshot of a solution from the validation set of the undamped KdV equation and the corresponding prediction from the autoencoder at $\lambda = 2.37 \times 10^{-4}$ and shows excellent reconstruction. On the other hand, the damped KdV equation does have finite-dimensional dynamics and therefore provides a fertile test ground for understanding the behavior of the latent space as the damping coefficient is varied. We consider the damped KdV equation over a range of damping coefficients, $\eta \in [0, 100]$ (see Table I). Before training the autoencoder, the input was normalized by its maximum value. For each damping coefficient, including the undamped case, the autoencoder with latent space penalization was trained across a number of regularization parameters to determine the optimal value. It is expected to be easier to learn the underlying manifold for problems with larger damping coefficients because the dimensionality of the underlying manifold is smaller for these types of problems. Following this line of reasoning, the optimal regularization parameter will be larger for smaller damping coefficients. Because of the stochastic nature of the optimization, the procedure to determine the optimal regularization parameter was repeated five times at each value of the damping coefficient in order to demonstrate the robustness of the results. Figure 7 shows the optimal regularization parameter for each damping coefficient. The undamped and weakly damped cases have similar optimal regularization parameters just less than $\lambda = 10^{-3}$, but as the damping increases the optimal regularization parameter decreases until it reaches very small values for large damping coefficients. This indicates that for very large damping coefficients the autoencoder is able to estimate the dimensionality of the manifold nearly unaided. We observe that a larger optimal λ is required to find an acceptable latent space to capture the dynamics for cases with smaller damping. This does not imply that the dimensionality of the learned latent space is smaller for smaller damping coefficients. Once again, inspecting the latent space directly sheds light on this detail. Figure 8 shows the IQR normalized by its maximum value for two extremes in damping coefficient (no damping on the top and strong damping on the bottom). Unlike in the K-S problem, there is no obvious clustering into "active" and "nonactive" dimensions for the KdV equations and an additional step must be performed to determine the dimensionality of the underlying manifold. For each damping coefficient, we pass all snapshots through the trained autoencoder with the optimal regularization parameter and extract the resulting latent space, which leads to matrix of size $n_z \times N_t$. An SVD on the learned latent space is performed to explain how the autoencoder is learning the dimension of the underlying manifold. Figure 9 shows the explained variance extracted from the SVD of the latent space for each damping coefficient, which is given by

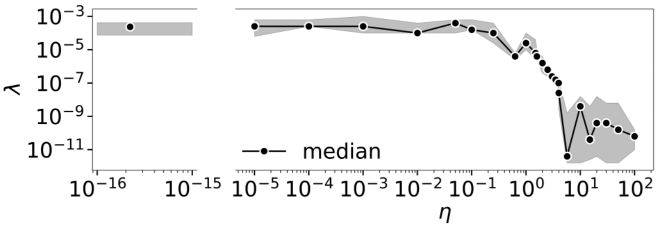

FIG. 7. Optimal regularization parameters for each value of damping coefficient. The black circles indicate the median value over five runs at each damping coefficient and the region between the maximum and minimum values for each run is shaded.

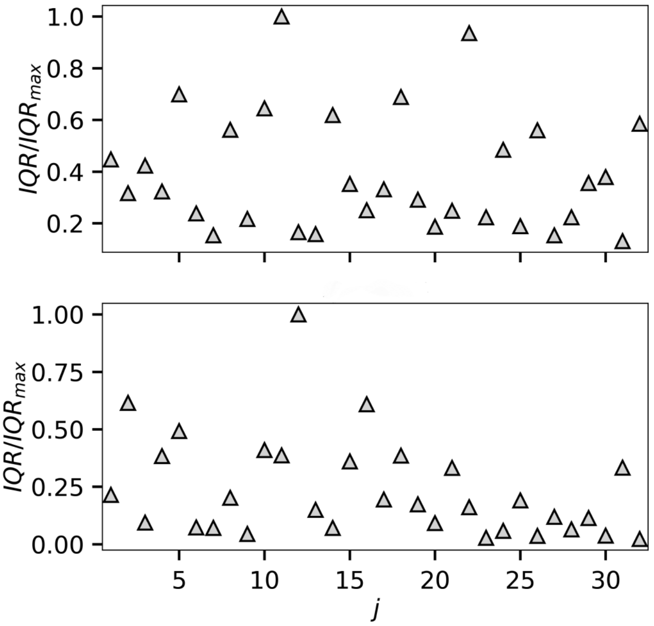

FIG. 8. Interquartile ranges of each component of the optimal latent space for the undamped (top) and strongly damped ($\eta = 100$, bottom) KdV equations. There is no clear clustering into "active" and "nonactive" dimensions in either case.

$$w_i = \sum_{j=1}^{i} \sigma_j \Bigg/ \sum_{j=1}^{n_z} \sigma_j, \tag{10}$$

where σ_j is the jth singular value. A clear trend is present in which smaller values of η are closer to the diagonal and large values of η exhibit a fuller profile. This implies that

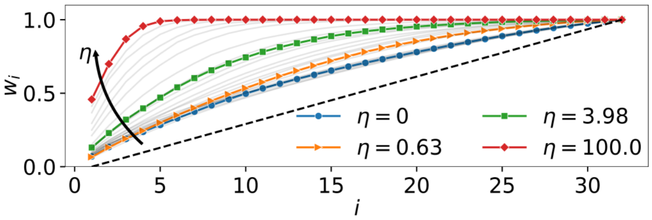

FIG. 9. The explained variance for various damping coefficients in the KdV equation. The dashed line shows the situation where all dimensions contribute equally. As the damping coefficient is increased, the explained variance profile becomes fuller indicating that fewer latent dimensions contribute.

smaller values of η have more "active" latent dimensions than large values of η, as expected. Although the trend is very clear for larger values of η, it is less obvious for small η for which some profiles are less full than that of $\eta = 0$. To disentangle this, we consider the area under the curve of each profile. If all modes were equally important, then the area under the curve (AUC) would be $(n_z - 1)/2$, which here is 15.5. At the other extreme, if only one mode was necessary, then the area under the curve would be $n_z - 1$. Considering $Z = n_z - \text{AUC}$ provides information on how many modes are required to represent the latent space. Considering the extremes again, if all modes were equally important, then $Z = (n_z - 1)/2$, whereas if only one mode was necessary then $Z = 1$. Figure 10 presents Z as a function of damping coefficient. We observe that the number of latent parameters is close to (but not quite) 15.5 for small values of η. For moderate values of η, the number of latent parameters drops quickly before plateauing at nearly two at large values of η. We note that for very large values of η the damped KdV equation is dominated by diffusion and for periodic boundary conditions the damped KdV equation tends to a constant for very large damping coefficients.

V. CONCLUSIONS

An autoencoder with latent space penalization was used to find approximate low-dimensional manifolds for two differential equations in physics. In the case of the K-S equation, the optimal latent space dimension was consistent with the known dimensionality of the inertial manifold for bifurcation parameter $L = 22$. We then determined a nonlinear basis for this manifold, which could in principle be used in a reduced order model. In the case of the KdV equation, which has truly infinite-dimensional dynamics, we found optimal results with a somewhat larger regularization parameter. This can be understood by recognizing that the autoencoder attempts to learn a reduced manifold regardless of the damping of the problem and that for small (or zero) damping coefficient, the autoencoder requires more regularization to find the correct manifold. Finally, when applying this technique to the damped KdV equation, which once again has finite-dimensional dynamics we found the beginnings of a power-law scaling for the number of active dimensions for sufficiently large damping coefficient. At very large damping coefficients the number of active dimensions for the KdV equation began to plateau at nearly 2. This is consistent with the behavior of the strongly damped KdV equation with periodic boundary conditions.

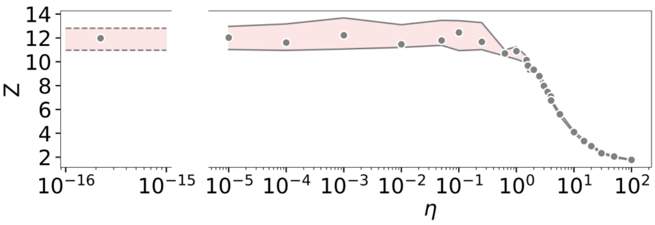

FIG. 10. The area under the explained variance curve as a function of damping coefficient. The undamped and weakly damped KdV cases show almost equal contributions from all latent dimensions. The strongly damped KdV cases begin to plateau at 1 to 2 latent dimensions.

Several avenues exist for extensions of this work. It would be interesting to rigorously connect the latent space with known mathematical objects such as the inertial manifold (when it is known to exist) and attractors [17]. In the present work, we empirically show that the dimensionality of the latent space is consistent with that of the inertial manifold for the K-S equation. However, it would be interesting to attempt to connect the two in a rigorous manner. Another route to take is to use the new basis for the K-S equation developed in this work to simulate dynamics with the reduced model. These simulations could be performed using a recurrent neural network [13] or traditional numerical solvers. Of particular interest is extending this work to more realistic problems in science and engineering including Rayleigh-Bénard convection. These problems will require different network architectures including convolutional neural networks.

ACKNOWLEDGMENTS

The authors thank Michael Jolly and Robert Moser for insightful discussions. The computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University.

APPENDIX: NUMERICAL SIMULATIONS

The one-dimensional K-S and KdV equations were solved using a pseudospectral Fourier discretization in space and an exponential fourth-order Runge-Kutta method [40] in time. The spatial domain was discretized using N points in physical space. The nonlinear terms were computed in physical space using the 2/3 dealiasing rule. Table I presents the runs that were used to generate the figures in the paper.

[1] P. J. Schmid, J. Fluid Mech. **656**, 5 (2010).
[2] P. Benner, S. Gugercin, and K. Willcox, SIAM Rev. **57**, 483 (2015).
[3] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, Comput. Fluids **179**, 704 (2019).
[4] I. E. Lagaris, A. Likas, and D. I. Fotiadis, IEEE Trans. Neural Netw. **9**, 987 (1998).
[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis, arXiv:1711.10561.
[6] J.-L. Wu, H. Xiao, and E. Paterson, Phys. Rev. Fluids **3**, 074602 (2018).
[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686 (2019).
[8] S. Greydanus, M. Dzamba, and J. Yosinski, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, New York, 2019), pp. 15379–15389.
[9] M. P. Brenner, J. D. Eldredge, and J. B. Freund, Phys. Rev. Fluids **4**, 100501 (2019).
[10] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, Physica D **406**, 132401 (2020).
[11] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, arXiv:2001.11107.
[12] J. Page, M. P. Brenner, and R. R. Kerswell, Phys. Rev. Fluids **6**, 034402 (2021).
[13] F. J. Gonzalez and M. Balajewicz, arXiv:1808.01346.
[14] P. V. Kuptsov and A. V. Kuptsova, in *Saratov Fall Meeting 2018: Computations and Data Analysis: From Nanoscale Tools to Brain Functions* (International Society for Optics and Photonics, Bellingham, Washington, 2019), Vol. 11067, p. 110670N.
[15] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Proc. Natl. Acad. Sci. USA **116**, 22445 (2019).
[16] A. J. Linot and M. D. Graham, Phys. Rev. E **101**, 062209 (2020).
[17] W. Gilpin, *Advances in Neural Information Processing Systems*, Vol. 33 (Curran Associates, Inc., Red Hook, New York, 2020), pp. 204–216.
[18] R. Tibshirani, J. R. Stat. Soc. B **58**, 267 (1996).
[19] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations* (CRC Press, Boca Raton, FL, 2015).
[20] E. J. Candés, J. Romberg, and T. Tao, IEEE Trans. Inf. Theory **52**, 489 (2006).
[21] E. J. Candés, J. K. Romberg, and T. Tao, Commun. Pure Appl. Math. **59**, 1207 (2006).
[22] D. L. Donoho, IEEE Trans. Inf. Theory **52**, 1289 (2006).
[23] R. G. Baraniuk, IEEE Signal Process. Mag. **24**, 118 (2007).
[24] E. J. Candès and M. B. Wakin, IEEE Signal Process. Mag. **25**, 21 (2008).
[25] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Proc. Natl. Acad. Sci. USA **113**, 3932 (2016).
[26] Y. Kuramoto and T. Tsuzuki, Prog. Theor. Phys. **55**, 356 (1976).
[27] B. I. Cohen, J. Krommes, W. Tang, and M. Rosenbluth, Nucl. Fus. **16**, 971 (1976).
[28] G. I. Sivashinsky, Acta astronautica **4**, 1177 (1977).
[29] G. I. Sivashinsky and D. M. Michelson, Progr. Theor. Phys. **63**, 2112 (1980).
[30] B. Nicolaenko, B. Scheurer, and R. Temam, Phys. D: Nonlinear Phenom. **16**, 155 (1985).
[31] C. Foias, B. Nicolaenko, G. Sell, and R. Temam, J. Math. Pures Appl. **67**, 197 (1988).
[32] R. Conte and M. Musette, J. Phys. A: Math. Gen. **22**, 169 (1989).
[33] M. S. Jolly, I. Kevrekidis, and E. S. Titi, Phys. D: Nonlinear Phenom. **44**, 38 (1990).
[34] J. C. Robinson, Phys. Lett. A **184**, 190 (1994).
[35] M. S. Jolly, R. Rosa, R. Temam *et al.*, Adv. Diff. Equ. **5**, 31 (2000).
[36] P. Constantin, C. Foias, B. Nicolaenko, and R. Temam, *Integral Manifolds and Inertial Manifolds for Dissipative Partial Differential Equations*, Vol. 70 (Springer Science & Business Media, New York, 2012).
[37] X. Ding, H. Chaté, P. Cvitanović, E. Siminos, and K. A. Takeuchi, Phys. Rev. Lett. **117**, 024101 (2016).
[38] D. J. Korteweg and G. De Vries, Lond. Edinb. Dubl. Philos. Mag. J. Sci. **39**, 422 (1895).
[39] J.-M. Ghidaglia, J. Diff. Equ. **74**, 369 (1988).
[40] A.-K. Kassam and L. N. Trefethen, SIAM J. Sci. Comput. **26**, 1214 (2005).
[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Technical Report (Institute for Cognitive Science, University of California San Diego, La Jolla, 1985).
[42] P. Baldi, in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (PMLR, Bellevue, Washington, 2012), pp. 37–49.
[43] M. Milano and P. Koumoutsakos, J. Comput. Phys. **182**, 1 (2002).
[44] T. Mikolov, Statistical language models based on neural networks, Ph.D. thesis, Brno University of Technology, 2012.
[45] D. P. Kingma and J. Ba, arXiv:1412.6980.
[46] H.-l. Yang and G. Radons, Phys. Rev. Lett. **108**, 154101 (2012).
[47] H.-l. Yang, K. A. Takeuchi, F. Ginelli, H. Chaté, and G. Radons, Phys. Rev. Lett. **102**, 074102 (2009).