

Learning a Reduced Basis of Dynamical Systems using an Autoencoder

David Sondak and Pavlos Protopapas

Institute for Applied Computational Science, Harvard University

(Dated: November 17, 2020)

Machine learning models have emerged as powerful tools in physics and engineering. In this work, we present an autoencoder with latent space penalization, which discovers finite dimensional manifolds underlying the partial differential equations of physics. 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. The results for the KdV equation imply that there is no 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. We then uncover a nonlinear basis representing the manifold of the latent space for the K-S equation.

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 multi-scale 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 [1–4]. In recent years, machine learning algorithms have been borrowed from the computer vision community and adapted for physical applications [5–13], 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 [8]. The current work presents a general approach to finding 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 [14–18]. In particular, [17] 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 (PCA) 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. In this way, the latent space of the trained autoencoder only contains the minimal dimensions needed to reconstruct the solution. This approach is tested on two equations that are known to have an inertial manifold (the Kuramoto-Sivashinsky equation and the damped Korteweg-de Vries

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 non-linear reduced solution basis whose dimension is consistent with that of the inertial manifold.

II. GOVERNING EQUATIONS AND DATASETS

The Kuramoto-Sivashinsky (K-S) equation is,

$$u_t + uu_x + u_{xx} + u_{xxx} = 0 \quad (1)$$

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 $u_0 = \cos\left(\frac{2\pi}{L}x\right) \left(1 + \sin\left(\frac{2\pi}{L}x\right)\right)$. The K-S equation has its roots in physics [19–22], and is a frequently-studied equation in mathematics [23–30]. Of particular relevance to the present work is the result that the dynamics of the K-S equation are confined to an inertial manifold [24]. That is, despite the K-S equation being a non-integrable equation whose solutions exhibit spatio-temporal chaos, the underlying dynamics are exponentially attracted to a finite-dimensional manifold. The dimension of the inertial manifold increases with the bifurcation parameter L .

The damped Korteweg-de Vries (KdV) equation, also considered in this work, is,

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

where $x \in [-\pi, \pi]$ and $\eta \geq 0$ 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 [31, 32], leading to the discovery of solitons, which are a direct bridge between observed

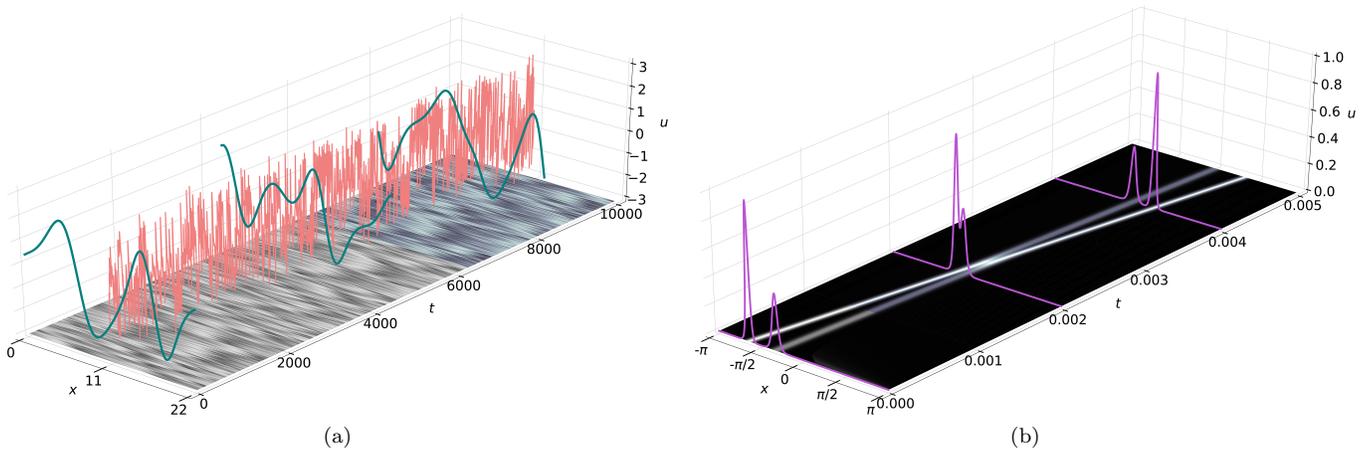


FIG. 1. Datasets from the K-S and KdV equations used in this work. (a) Contours of the K-S equation in space-time with three spatial snapshots at different points in time and a snapshot in time at the center of the domain. The grey shaded area of the contour was not used in the training. (b) Space-time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

coherent structures in PDEs and nature. The dynamics of the undamped KdV equation are truly infinite dimensional and are therefore not confined to an inertial manifold while those of the damped KdV equation are finite-dimensional [32].

The datasets for this work were generated by solving (1) and (2) using an exponential time-differencing fourth-order Runge-Kutta method [33] and a pseudo-spectral 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 (Figure 1a) and the KdV dataset (Figure 1b). The numerical parameters used for the cases in this work are presented in Table I in Appendix A.

III. METHODOLOGY

Autoencoders are a self-supervised neural network architecture that can be used to find a low-dimensional manifold that represents the data [34, 35]. 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 [36]. 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 $n = 1, \dots, N_t$ 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 j^{th} component of the latent space corresponding to snapshot n . The reconstructed output of the autoencoder is denoted by $\hat{\mathbf{u}}_n$. The weights and

biases associated with each node of the autoencoder are tuned to minimize the total loss,

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

where

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

is the mean squared error (MSE) reconstruction loss and

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

is the mean absolute error (MAE) penalization loss on the latent dimensions. The sparsity of the latent space is controlled by the regularization parameter $\lambda \geq 0$. 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 dimensions 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 $N \rightarrow H_1 \rightarrow H_2 \cdots \rightarrow H_D \rightarrow N_z$ 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 [37] to limit the maximum gradient to 10. The Adamax optimizer [38] was used in all experiments. The transient portion of

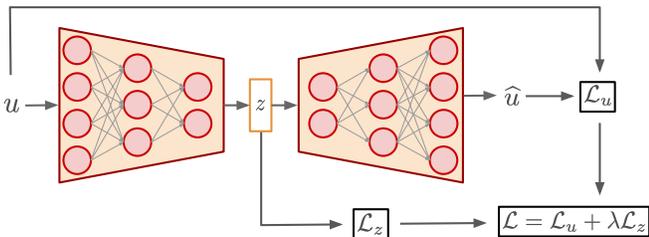


FIG. 2. The autoencoder architecture with latent space penalization. The reconstruction loss L_u (4) is combined with a mean absolute error loss L_z (5). The result is a solution reconstruction that uses only relevant latent dimensions.

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 in Appendix B 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 in Appendix A and B, 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 , $\mathcal{L}_u^*(u_{\text{valid}}, \hat{u}_{\text{valid}})$, 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 $\hat{u}(x, t)$ 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 $\mathcal{L}_u^*(u_{\text{valid}}, \hat{u}_{\text{valid}})$ 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 [30, 39]. 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 latent dimension) a very clear separation between “active” and “non-active” 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,

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

where

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

and

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

is a regularization used to smooth the resulting field with the regularization parameter $\beta \geq 0$. 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),

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

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. Figures 5a and 5c 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 5b and 5d show that the inputs that maximize the non-active 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 ten times for each value of L . 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 [40].

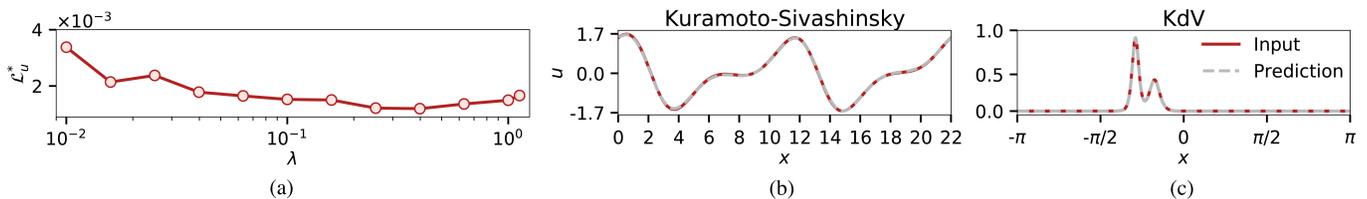


FIG. 3. (a) Reconstruction loss on the validation set for the K-S equation with $L = 22$ across different values of penalization parameter λ . In each case the network architecture was $512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32$. The minimum value occurs at $\lambda = 0.398$.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

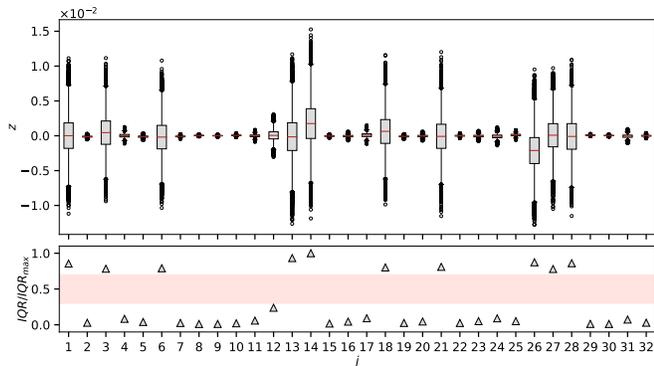


FIG. 4. Active latent dimensions for the model with the lowest total loss (3) ($\lambda = 0.398$). Top: Box plots of each dimension of the latent space generated by passing all snapshots through the autoencoder and extracting the latent space. Bottom: Normalized IQR for each dimension showing a gap between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

B. KdV Equation

The latent space penalization enabled the discovery of a reduced basis for an equation that has an inertial manifold. 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 [32]. The undamped KdV equation therefore provides a test for the method in which the optimal latent space penalization is expected to be zero. Before training the autoencoder, the input was normalized by its maximum value. Following the same procedure as for the K-S equation, the optimal latent space regularization was determined to be $\lambda = 0$ implying that the system does not possess an underlying finite-dimensional manifold. Figure 3 (c) presents a snapshot of a solution from the validation set and the corresponding prediction from the autoencoder at $\lambda = 0$. The lack of a finite-dimensional manifold is further supported by Figure 7, in which the IQR for each snapshot of the dataset is visualized for each dimension in the latent space. There is no separation of the latent space dimensions into “active” and “nonactive” components.

C. Damped KdV Equation

As a final test case, the damped KdV equation was studied over a range of damping coefficients, η (see Table I in Appendix A). The damped KdV equation possesses an inertial manifold and it therefore provides a fertile test ground for understanding the behavior of the latent space as the damping coefficient is varied. For each damping coefficient, the autoencoder with latent space penalization was trained across a number of regularization parameters. The number of active dimensions for each damped KdV equation had to be determined. However, the minima in the MSE loss, L_u vs. regularization parameter, λ , on the validation sets were shallow. Figure 8a presents a representative example for one particular model. Rather than extract a single minimum as was done in the K-S case, a different procedure was employed to provide a more robust result. As before, the model with the minimum validation loss was extracted at each λ and used for the analysis. Instead of retaining a single model at the minimum of the $L_u(\lambda)$ curve, all models were retained in the shallow minimum. The latent dimensions for every snapshot of the dataset were computed for each of these models and the IQR was again used to assess the activity of each latent dimension. The number of active dimensions was computed by counting the number of latent dimensions with IQR above a certain threshold, IQR_{thresh} , for each model. To guard against sensitivity in the choice of IQR_{thresh} , a range of thresholds was tested between 0.1 and 0.8. This process results in an array representing the number of active dimensions for each value of η . The array contains the number of active dimensions for each model in the shallow minimum and each threshold value. The median number of dimensions over each model in the shallow minimum for each case and each threshold value was computed, which resulted in an array for each case that represents the number of active dimensions for each choice of threshold. As expected, the threshold value only shifts the number of active dimensions up and down, but the behavior of the number of active dimensions over η for each threshold value remains the same. The final result is obtained by taking the median over the threshold values. Figure 8b shows the variation of active dimensions as the threshold is increased. The number of active latent dimensions

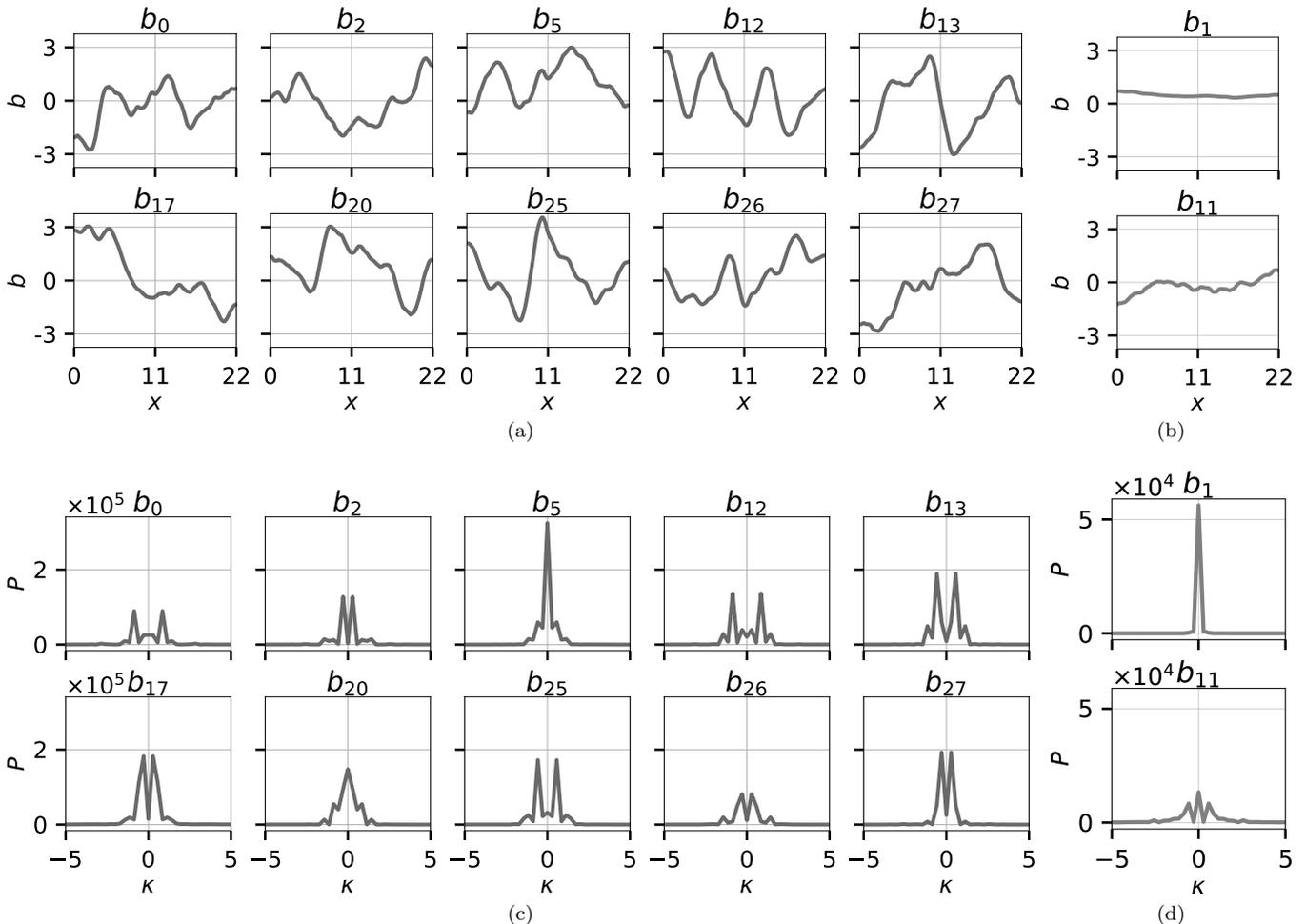


FIG. 5. (a) The nonlinear basis for the 10– dimensional manifold found for the K-S equation with $L = 22$. (b) Examples of the non-active dimensions. (c) Power spectrum of the basis for the active dimensions. (d) Power spectrum of two non-active dimensions. Wavenumbers in the power spectra have been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

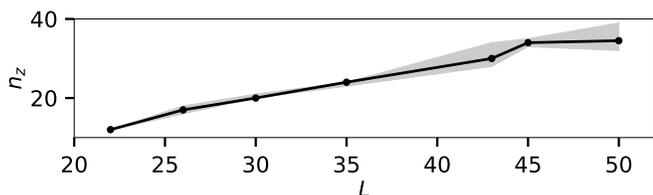


FIG. 6. Scaling of the optimal number of latent dimensions with domain size L for the K-S equation.

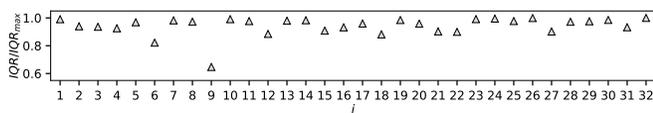


FIG. 7. Active latent dimensions for the KdV equation at $\lambda = 0$ visualized with the normalized IQR . The least active dimension is more than 60% of the most active dimension indicating that all latent dimensions are required to reconstruct the solution.

decreases nearly monotonically with increasing damping coefficient as shown in Figure 9.

V. CONCLUSIONS

We introduced an autoencoder with latent space penalization applied to datasets generated from nonlinear partial differential equations in physics. The latent space penalization is used to restrict the dimensionality of the latent space to the minimal number of dimensions, thereby ensuring that the dynamics embodied in the dataset are captured by a low-dimensional manifold. 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. The KdV equation does not possess an inertial manifold and

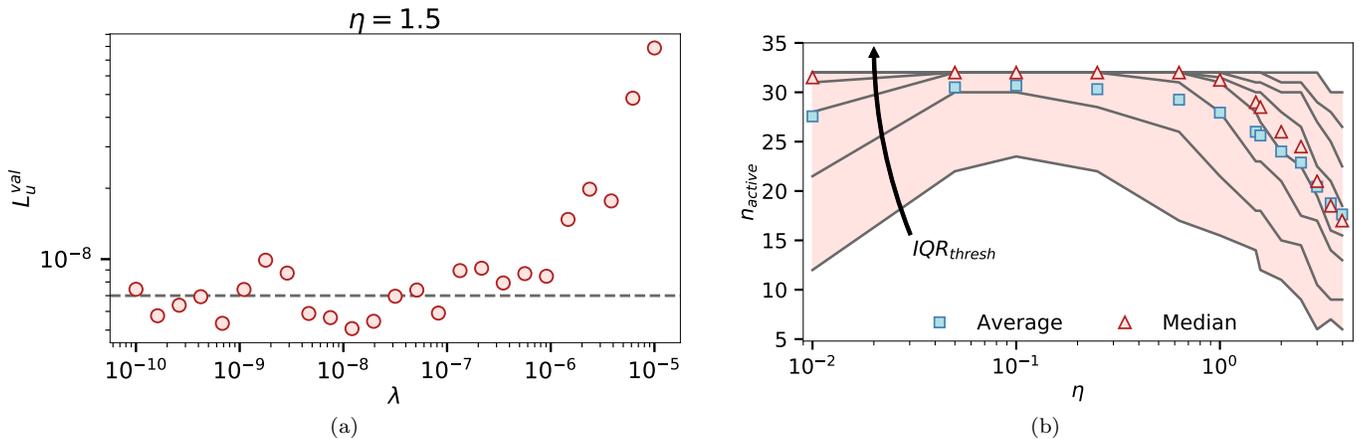


FIG. 8. (a) Validation loss as a function of regularization parameter for the damped KdV equation with $\eta = 1.5$. The dashed line is presented as a reference to highlight the shallow minimum / plateau. (b) The number of active dimensions as a function of damping parameter for different IQR thresholds. The mean and median show the overall scaling. The paper presents the median.

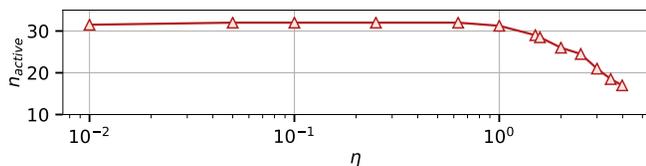


FIG. 9. Active latent space dimensions vs. damping parameter η for the damped KdV equation.

has truly infinite dimensional dynamics. In this case, we found optimal results without any latent space penalization, which is consistent with the known properties of the KdV equation. Finally, when applying this technique to the damped KdV equation, which once again has an iner-

tial manifold, we found the beginnings of power law scaling for the number of active dimensions for sufficiently large damping coefficient. This technique opens interesting directions for determining reduced order models.

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.

-
- [1] P. Schmid and J. Sesterhenn, *APS* **61**, MR (2008).
[2] P. J. Schmid, *Journal of fluid mechanics* **656**, 5 (2010).
[3] P. Benner, S. Gugercin, and K. Willcox, *SIAM review* **57**, 483 (2015).
[4] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, *Computers & Fluids* **179**, 704 (2019).
[5] I. E. Lagaris, A. Likas, and D. I. Fotiadis, *IEEE transactions on neural networks* **9**, 987 (1998).
[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, arXiv preprint arXiv:1711.10561 (2017).
[7] J.-L. Wu, H. Xiao, and E. Paterson, *Physical Review Fluids* **3**, 074602 (2018).
[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Journal of Computational Physics* **378**, 686 (2019).
[9] S. Greydanus, M. Dzamba, and J. Yosinski, in *Advances in Neural Information Processing Systems* (2019), pp. 15379–15389.
[10] M. Brenner, J. Eldredge, and J. Freund, *Physical Review Fluids* **4**, 100501 (2019).
[11] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, *Physica D: Nonlinear Phenomena* **406**, 132401 (2020).
[12] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopoulos, arXiv preprint arXiv:2001.11107 (2020).
[13] J. Page, M. P. Brenner, and R. R. Kerswell, arXiv preprint arXiv:2008.07515 (2020).
[14] F. J. Gonzalez and M. Balajewicz, arXiv preprint arXiv:1808.01346 (2018).
[15] 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, 2019), vol. 11067, p. 110670N.
[16] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, *Proceedings of the National Academy of Sciences* **116**, 22445 (2019).
[17] A. J. Linot and M. D. Graham, *Physical Review E* **101**, 062209 (2020).
[18] W. Gilpin, arXiv preprint arXiv:2002.05909 (2020).
[19] Y. Kuramoto and T. Tsuzuki, *Progress of theoretical physics* **55**, 356 (1976).
[20] B. I. Cohen, J. Krommes, W. Tang, and M. Rosenbluth, *Nuclear fusion* **16**, 971 (1976).

- [21] G. Sivashinsky, *AcAau* **4**, 1177 (1977).
- [22] G. I. Sivashinsky and D. Michelson, *PThPh* **63**, 2112 (1980).
- [23] B. Nicolaenko, B. Scheurer, and R. Temam, *Physica D: Nonlinear Phenomena* **16**, 155 (1985).
- [24] C. Foias, B. Nicolaenko, G. Sell, and R. Temam, *Journal de Mathématiques Pures et Appliquées* **67**, 197 (1988).
- [25] R. Conte and M. Musette, *Journal of Physics A: Mathematical and General* **22**, 169 (1989).
- [26] M. S. Jolly, I. Kevrekidis, and E. S. Titi, *Physica D: Nonlinear Phenomena* **44**, 38 (1990).
- [27] J. C. Robinson, *Physics Letters A* **184**, 190 (1994).
- [28] M. S. Jolly, R. Rosa, R. Temam, et al., *Advances in Differential Equations* **5**, 31 (2000).
- [29] 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, 2012).
- [30] X. Ding, H. Chaté, P. Cvitanović, E. Siminos, and K. Takeuchi, *Physical review letters* **117**, 024101 (2016).
- [31] D. J. Korteweg and G. De Vries, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **39**, 422 (1895).
- [32] J.-M. Ghidaglia, *Journal of Differential Equations* **74**, 369 (1988).
- [33] A.-K. Kassam and L. N. Trefethen, *SIAM Journal on Scientific Computing* **26**, 1214 (2005).
- [34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Tech. Rep.*, California Univ San Diego La Jolla Inst for Cognitive Science (1985).
- [35] P. Baldi, in *Proceedings of ICML workshop on unsupervised and transfer learning* (2012), pp. 37–49.
- [36] M. Milano and P. Koumoutsakos, *Journal of Computational Physics* **182**, 1 (2002).
- [37] T. Mikolov et al., Presentation at Google, Mountain View, 2nd April **80**, 26 (2012).
- [38] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).
- [39] H.-l. Yang and G. Radons, *Physical review letters* **108**, 154101 (2012).
- [40] H.-l. Yang, K. A. Takeuchi, F. Ginelli, H. Chaté, and G. Radons, *Physical review letters* **102**, 074102 (2009).

Appendix A: Numerical Simulations

The one-dimensional Kuramoto-Sivashinsky (K-S) and Korteweg-de Vries (KdV) equations were solved using a pseudo-spectral Fourier discretization in space and an exponential fourth-order Runge-Kutta method [33] 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.

Appendix B: Autoencoder Architectures

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	$4 \cdot 10^4$	0.125
2	K-S	[0, 22]	—	1024	$4 \cdot 10^4$	0.125
3	K-S	[0, 26]	—	1024	$4 \cdot 10^4$	0.125
4	K-S	[0, 30]	—	1024	$4 \cdot 10^4$	0.125
5	K-S	[0, 35]	—	1024	$4 \cdot 10^4$	0.125
6	K-S	[0, 43]	—	1024	$4 \cdot 10^4$	0.125
7	K-S	[0, 45]	—	1024	$4 \cdot 10^4$	0.125
8	K-S	[0, 50]	—	1024	$4 \cdot 10^4$	0.125
9	KdV	$[-\pi, \pi]$	0	512	$5 \cdot 10^{-3}$	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}

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 was 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	$3 \cdot 10^4$	$5 \cdot 10^4$
2	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
3	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
4	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
5	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
6	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
7	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
8	(E): 1024 → 256 → 64 (D): 64 → 256 → 1024	10^{-3}	40	$3 \cdot 10^4$	$5 \cdot 10^4$
9 ($\lambda = 0$)	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	$5 \cdot 10^{-3}$	100	10^5	10^4
9 ($\lambda > 0$)	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	$5 \cdot 10^{-3}$	100	10^5	10^4
10	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$5 \cdot 10^4$	10^4
11	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
12	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
13	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
14	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
15	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
16	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
17	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
18	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
19	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
20	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
21	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4
22	(E): 512 → 256 → 128 → 64 → 32 (D): 32 → 64 → 128 → 256 → 512	10^{-3}	60	$7.5 \cdot 10^4$	10^4