Physics-informed Autoencoders for Lyapunov-stable Fluid Flow Prediction

N. Benjamin Erichson
Department of Statistics
ICSI and UC Berkeley
erichson@berkeley.edu

Michael Muehlebach
Department of EECS
UC Berkeley
michaelm@berkeley.edu

Michael W. Mahoney
Department of Statistics
ICSI and UC Berkeley
mahoneymw@berkeley.edu

Abstract

In addition to providing high-profile successes in computer vision and natural language processing, neural networks also provide an emerging set of techniques for scientific problems. Such data-driven models, however, typically ignore physical insights from the scientific system under consideration. We investigate whether it is possible to include physics-informed prior knowledge for improving the model quality (*e.g.*, generalization performance, or robustness in the presence of noisy data). To that extent, we focus on the stability of an equilibrium, one of the most basic properties a dynamic system can have, via the lens of Lyapunov analysis.

1 Introduction

Many problems in science and engineering can be modeled as a dynamical system. Examples include physical fluid flows, atmospheric-ocean interactions, neurophysiological responses, and economic and financial time series, to name only a few. These systems often exhibit rich dynamics that give rise to multiscale structures, in both space and time. Since these systems are typically identified using data, machine learning methods are increasingly of interest for these problems. Deep learning and related neural network techniques, in particular, provide a useful framework for modeling such systems. The merits of deep learning have been demonstrated for scientific applications, in particular for prototypical fluid flow applications, such as fluid flow modeling [12, 15, 7, 11, 9, 23, 10], flow reconstruction [5, 4], flow control and prediction [13, 19, 20, 17, 16], and flow simulation [8, 21, 22].

Thus far, however, neural network models for scientific applications largely ignore knowledge of physics and other domain-specific aspects of the system under consideration. As a consequence, this domain-agnostic learning approach can lead to models that are brittle. One would hope that domain-specific assumptions can improve the algorithmic performance and predictive accuracy of scientific-based machine learning models. For example, physically-informed priors can introduce some degree of *stability* and *robustness*, in that a small change of the input will not dramatically change the output of the learning algorithm [24, 3, 25].

Motivated by this idea, we design stability-preserving models for fluid flow prediction. This is a prototypical scientific problem with a dynamical systems interpretation. More concretely, we learn an end-to-end mapping between the input and target fluid flow snapshot, where the mapping is represented as an autoencoder, with an additional component that attempts to learn the dynamics of the underlying physical process. To illustrate the promise of the method, we show results for simulated and real-world problems, including laminar flow and climate problems, and we demonstrate the use of this physics-informed approach both for improved model training and for improved *a posteriori* model analysis. For model training, we show that constraining the empirical risk minimization problem by using a Lyapunov stability-promoting prior (a physical-meaningful regularization mechanism that corresponds to properties of the physical system being modeled) leads to better training and helps to improve the generalization performance, compared to physics-agnostic models.

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

2 Problem setup

Modeling nonlinear dynamics can be challenging and thus often linear time-invariant approximations of nonlinear systems are used. These take the form

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \eta_t, \quad t = 0, 1, 2, \ldots, \tag{1}$$

where $\mathbf{A} : \mathbb{R}^n \to \mathbb{R}^n$ denotes a linear map. The perturbation η_t might incorporate modeling errors, such as unmodeled dynamics or discretization errors. If η_t is small, then the dynamics simply specify here that the state $\mathbf{x}_{t+1}$ depends just on the value of the previous state $\mathbf{x}_t$, i.e., given the rule $\mathbf{A}$, the state $\mathbf{x}_t$ provides all information needed for predicting the future state at $\mathbf{x}_{t+1}$.

However, despite the simplicity of this rule, it often turns out to be a challenge to find an estimate for $\mathbf{A}$. This is because, in a data-driven setting, we only have access to (high-dimensional) observations

$$\mathbf{y}_t = \mathcal{G}(\mathbf{x}_t) + \xi_t, \quad t = 0, 1, 2, \ldots, T, \tag{2}$$

where the function $\mathcal{G} : \mathbb{R}^n \to \mathcal{Y} \subseteq \mathbb{R}^m$ maps the state $\mathbf{x}_t$ to a subspace $\mathcal{Y}$, and the variable ξ_t represents measurement errors. For example, one may think of the function $\mathcal{G}$ as a sensor which collects measurements at time t. We assume that the dynamics of the flow are low-dimensional, in the sense that $\mathcal{G}(\mathbf{x}_t)$, $t = 1, 2, \ldots$, lies on an n-dimensional manifold embedded in $\mathbb{R}^m$ [2]. Further, we assume that the function $\mathcal{G}$ has an inverse, which implies that a single data-point $y_t \in \mathcal{Y}$ is enough to uniquely determine the corresponding state $\mathbf{x}_t$.

3 Autoencoder-type models for fluid flow prediction

Given a sequence of observations $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_T \in \mathbb{R}^m$ for training, the objective of this work is to learn a model which maps the snapshot $\mathbf{y}_t$ to $\mathbf{y}_{t+1}$. The model is composed of three functions

$$\hat{\mathbf{y}}_{t+1} = \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi}(\mathbf{y}_t), \tag{3}$$

where $\mathbf{\Phi}$ approximates $\mathcal{G}$, $\mathbf{\Omega}$ approximates $\mathbf{A}$, and $\mathbf{\Psi}$ approximates $\mathcal{G}^{-1}$. The design architecture of the model is sketched in Figure 1. The encoder $\mathbf{\Psi} : \mathcal{Y} \to \mathbb{R}^n$ maps the high-dimensional snapshot $\mathbf{y}_t$ to a low-dimensional feature space, where $n \ll m$. The encoder should be designed so that it preserves the coherent structure of the fluid flow, while suppressing uninformative variance (fine scale features) in the data. The dynamics $\mathbf{\Omega} : \mathbb{R}^n \to \mathbb{R}^n$ evolves the state in time, modeled as

$$\mathbf{z}_{t+1} = \mathbf{\Omega}\mathbf{z}_t. \tag{4}$$

where $\mathbf{z}_t = \mathbf{\Psi}(\mathbf{y}_t)$. Finally, the decoder $\mathbf{\Phi} : \mathbb{R}^n \to \mathcal{Y}$ maps the low-dimensional features (evolved in time) back to the high-dimensional measurement space.

During inference time we can obtain predictions $\hat{\mathbf{y}}_t$ for an initial point $\mathbf{y}_0$ by composing the learned model t-times. This leads to the following expansion

$$\hat{\mathbf{y}}_t = \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi} \circ \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi} \circ \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi} \circ \ldots \circ \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi}(\mathbf{y}_0). \tag{5}$$

If the model obeys the assumption that $\mathbf{\Psi}$ approximates $\mathcal{G}^{-1}$, then we have that $\mathbf{I} \approx \mathbf{\Psi} \circ \mathbf{\Phi}$, where $\mathbf{I} \in \mathbb{R}^{n \times n}$ denotes the identity. Thus, Eq. (5) reduces approximately to

$$\hat{\mathbf{y}}_t \approx \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Omega} \circ \mathbf{\Omega} \ldots \circ \mathbf{\Omega} \circ \mathbf{\Psi}(\mathbf{y}_0) = \mathbf{\Phi} \circ \mathbf{\Omega}^t \circ \mathbf{\Psi}(\mathbf{y}_0). \tag{6}$$

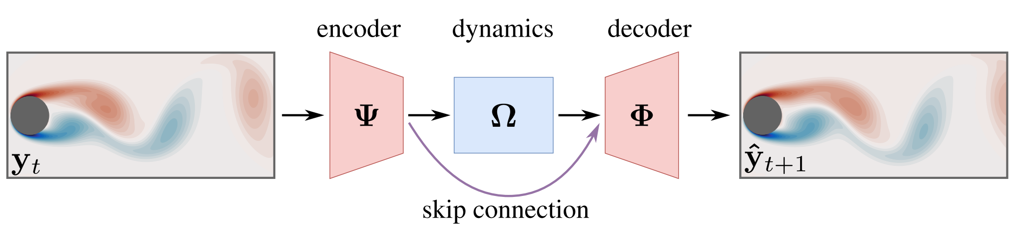


Figure 1: Design architecture of the autoencoder-type flow prediction model. The skip connection allows one to enforce the identity-persevering constraint posed on the encoder. This constraint is important, because we aim to design the model so that only $\mathbf{\Omega}$ captures the dynamics. Note, we do not train $\mathbf{\Omega}$ with the identity map, i.e., $\mathbf{\Omega}$ is a discrete-time flow map.

In order for this to happen, we enforce that the function $\mathbf{\Psi}$ acts as an (approximate) inverse function for $\mathbf{\Phi}$, i.e., $\mathbf{q}_t \approx \mathbf{\Psi} \circ \mathbf{\Phi}(\mathbf{q}_t)$. This is achieved by introducing an additional penalty

$$\min \frac{1}{T-1} \sum_{t=0}^{T-1} \|\mathbf{y}_{t+1} - \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi}(\mathbf{y}_t)\|_2^2 \ + \ \lambda \|\mathbf{q}_t - \mathbf{\Psi} \circ \mathbf{\Phi}(\mathbf{q}_t)\|_2^2, \tag{7}$$

where λ is a tuning parameter that balances the two objectives. The variable $\mathbf{q}_t$ denotes carefully chosen test points, which we set in our experiments to the encoded flow field $\mathbf{q}_t = \mathbf{\Psi}(\mathbf{y}_t)$ at time t.

4 Lyapunov stability as a tool for physics-informed learning

Here, we focus on Lyapunov stability, which describes a fundamental property of the dynamic system in Eq. (1). We assume that the dynamic system has an equilibrium at the origin. This means that if the system is initialized at the origin ($\mathbf{x}_0 = 0$), its state will remain at the origin for all times (that is, $\mathbf{x}_t = 0$ for all $t = 1, 2, \dots$ provided that $\eta_t = 0$ for all $t = 1, 2, \dots$). We can then ask ourselves what happens when the system is initialized in a region close to the the origin. Will the resulting trajectories remain close to the equilibrium for all times, or will they drift away? If the former is true, the origin of the dynamic system is said to be stable. Compared to the other notions of stability mentioned above, Lyapunov stability is therefore a statement about the robustness of trajectories with respect to small perturbations of their initial conditions about the given equilibrium.

Given that (1) is stable, we are interested in learning a model (4) that is likewise stable. Therefore, we design a stability-promoting penalty based on Lyapunov's method. More precisely, we impose that the symmetric matrix $\mathbf{P}$, defined by

$$\mathbf{\Omega}^\top \mathbf{P} \mathbf{\Omega} - \mathbf{P} = -\mathbf{I}, \tag{8}$$

is positive definite. We design a prior that promotes Lyapunov stability by penalizing eigenvalues of $\mathbf{P}$, denoted as p, that have small negative values. Such a prior can take various forms, but the following choice works particularly well in our experiments:

$$\rho(p) := \begin{cases} \exp\left(-\frac{|p-1|}{\gamma}\right) & \text{if } p < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where γ is a tuning parameter. The physics-informed autoencoder preserves stability if κ is chosen large enough, trained by minimizing the following objective

$$\min \frac{1}{T-1} \sum_{t=0}^{T-1} \|\mathbf{y}_{t+1} - \mathbf{\Phi} \circ \mathbf{\Omega} \circ \mathbf{\Psi}(\mathbf{y}_t)\|_2^2 \ + \ \lambda \|\mathbf{q}_t - \mathbf{\Psi} \circ \mathbf{\Phi}(\mathbf{q}_t)\|_2^2 + \ \kappa \sum_i \rho(p_i). \tag{10}$$

5 Experiments and discussion

Here, we provide empirical results, demonstrating the generalization performance, by studying a laminar flow and a real-world climate problem. We use shallow architectures which are composed of only a few linear layers, connected by non-linear activation functions. These architectures provide an excellent parsimonious-predictability trade-off for our fluid flow prediction problems. Shallow networks have the advantage that they are scalable, fast to train, and easy to tune [4].

The extend version of this paper (`https://arxiv.org/abs/1905.10866`) provides additional results, specifics of the network architecture and details about the tuning parameters.

5.1 Flow behind a cylinder

As a canonical example, we consider a downsampled fluid flow behind a cylinder, which is characterized by a periodically shedding wake structure [14]. The dataset comprises 250 fluid flow snapshots in time, each consisting of 64×64 spatial grid points. We split the sequence into a training (first 100 snapshots) and test set (remaining 150 snapshots).

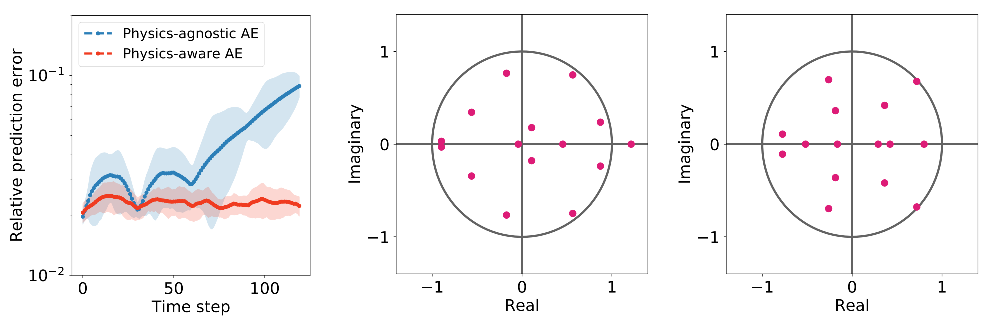
We evaluate the quality of the physics-agnostic model (minimizing (7)) and the physics-aware model (minimizing (10)) by studying their ability to estimate future fluid flow fields. For both models, we expect that the extrapolation in time will eventually break down, but we expect that there will be a larger range of time over which the extrapolation is valid for models that are designed to have the stability properties of the underlying physical system. The physics-aware model shows an improved generalization performance, when averaged over 30 initial conditions, as shown in Figure 2. Further, it can be seen that the stability-promoting prior reduces the prediction uncertainty.

Of course, one can also fiddle around with the amount of weight decay until all eigenvalues of $\boldsymbol{\Omega}$ have magnitude less than one (*i.e.*, increasing the amount of weight decay shrinks the eigenvalues towards the origin). However, a physics-informed prior appears to be a more elegant solution.

5.2 Sea surface temperature of the gulf of Mexico

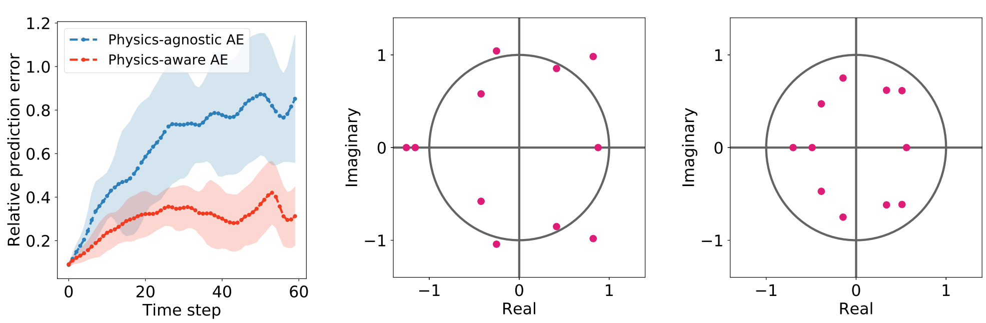
Next, we model the sea surface temperature (SST) of the the Gulf of Mexico as a real-world example to demonstrate the performance of our physics-informed autoencoder. The National Oceanic & Atmospheric Administration (NOAA) provides daily sea surface temperatures for the last 26 years. We consider the daily SSTs for the Gulf of Mexico over a period of six years (2012-2018). The data comprise 2190 snapshots in time with spatial resolution of 64×64. We split the sequence into a training (first 1825 snapshots) and test set (remaining 365 snapshots).

Again, the physics-aware model shows an improved generalization performance for a larger range of time, as shown in Figure 3. The prediction error is overall substantially larger than in the previous example. This is because predicting the fluctuations (in this non-toy model) is a challenging problem, since complex ocean dynamics lead to rich flow phenomena, featuring various seasonal fluctuations.



(a) Generalization performance. (b) Physics-agnostic model in (a). (c) Physics-aware model in (a).

Figure 2: Summary of results for the flow behind a cylinder. The physics-aware model outperforms the physics-agnostic model for predicting future flow fields over a time horizon of 120 snapshots. The results are averaged over 30 initial conditions and the error bands show the 5% and 95% percentile. The plots (b) and (c) show the complex eigenvalues of $\boldsymbol{\Omega}$, which correspond to the models in (a).



(a) Generalization performance. (b) Physics-agnostic model in (a). (c) Physics-aware model in (a).

Figure 3: Summary of results for the SST data. The physics-informed model shows a better generalization performance over a prediction horizon of 60 days. The results are averaged over 30 initial conditions and the error bands show the 5% and 95% percentile. The plots (b) and (c) show the complex eigenvalues of $\boldsymbol{\Omega}$, which correspond to the models in (a).

6 Conclusion

Neural networks have been shown to be a highly valuable tool for dynamical modeling, prediction and control of fluid flows. Surprisingly, these data-driven models have the ability to learn implicitly some of the physical properties (encoded in the data) reasonably well, if a sufficient amount of data is provided for training. However, often the amount of data is limited, and one has knowledge about the data generation mechanisms. In this case, physics-informed learning might help to improve considerably the generalization performance. To accomplish this, we introduced a method for training autoencoders that preserve Lyapunov stability. This simple, yet effective, approach of including a physics-informed stability-enhancing prior into the learning process shows a substantial performance boost for several fluid flow prediction tasks. A minor disadvantage is that we need an additional tuning parameter, but we have observed that tuning this relatively-robust parameter is not a problem.

Acknowledgments

MWM and NBE would like to acknowledge the generous funding support from the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (FA8750-17-2-0122) as well as Amazon Web Services (AWS) for supporting the project.

References

[1] Ravi P Agarwal. *Difference equations and inequalities: theory, methods, and applications*. CRC Press, 2 edition, 2000.

[2] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

[3] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[4] N Benjamin Erichson, Lionel Mathelin, Zhewei Yao, Steven L Brunton, Michael W Mahoney, and J Nathan Kutz. Shallow learning for fluid flow reconstruction with limited sensors and limited data. *arXiv preprint arXiv:1902.07358*, 2019.

[5] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.

[6] Wolfgang Hahn. *Stability of Motion*. Grundlehren der mathematischen Wissenschaften. Springer, 1967.

[7] David Hartman and Lalit K Mestha. A deep learning framework for model reduction of dynamical systems. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1917–1922. IEEE, 2017.

[8] Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. *arXiv preprint arXiv:1806.02071*, 2018.

[9] Kookjin Lee and Kevin Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *arXiv preprint arXiv:1812.08373*, 2018.

[10] Hugo FS Lui and William R Wolf. Construction of reduced order models for fluid flows using deep feedforward neural networks. *arXiv preprint arXiv:1903.05206*, 2019.

[11] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[12] Michele Milano and Petros Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

[13] Jeremy Morton, Antony Jameson, Mykel J Kochenderfer, and Freddie Witherden. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems*, pages 9258–9268, 2018.

[14] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[15] Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencodernetworks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[16] Shaowu Pan and Karthik Duraisamy. Long-time predictive modeling of nonlinear dynamical systems using neural networks. *Complexity*, 2018, 2018.

[17] Shivakanth Chary Puligilla and Balaji Jayaraman. Deep multilayer convolution frameworks for data-driven learning of fluid flow dynamics. In *2018 Fluid Dynamics Conference*, page 3091, 2018.

[18] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer, 2013.

[19] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.

[20] Pantelis R Vlachas, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[21] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):95, 2018.

[22] Kyongmin Yeo and Igor Melnyk. Deep learning algorithm for data-driven simulation of noisy dynamical system. *Journal of Computational Physics*, 376:1212–1231, 2019.

[23] Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. *arXiv preprint arXiv:1708.06850*, 2017.

[24] Bin Yu et al. Stability. *Bernoulli*, 19(4):1484–1500, 2013.

[25] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4488, 2016.