Discovering Dynamical Parameters by Interpreting Echo State Networks

Oreoluwa Alao*
Department of Physics
Massachusetts Institute of Technology
Cambridge, MA 02139
orealao@mit.edu

Peter Y. Lu*
Department of Physics
Massachusetts Institute of Technology
Cambridge, MA 02139
lup@mit.edu

Marin Soljačić
Department of Physics
Massachusetts Institute of Technology
Cambridge, MA 02139
soljacic@mit.edu

Abstract

Reservoir computing architectures known as echo state networks (ESNs) have been shown to have exceptional predictive capabilities when trained on chaotic systems. However, ESN models are often seen as black-box predictors that lack interpretability. We show that the parameters governing the dynamics of a complex nonlinear system can be encoded in the learned readout layer of an ESN. We can extract these dynamical parameters by examining the geometry of the readout layer weights through principal component analysis. We demonstrate this approach by extracting the values of three dynamical parameters (σ, ρ, β) from a dataset of Lorenz systems where all three parameters are varying among different trajectories. Our proposed method not only demonstrates the interpretability of the ESN readout layer but also provides a computationally inexpensive, unsupervised data-driven approach for identifying uncontrolled variables affecting real-world data from nonlinear dynamical systems.

1 Introduction

The analysis and modeling of complex nonlinear dynamics is a key problem across a broad range of disciplines in science and engineering [1]. While supervised deep learning approaches can provide excellent predictive performance on dynamical systems [2–4], difficulties with interpretability limit their usefulness in some scientific applications. For example, consider the problem of identifying uncontrolled dynamical parameters causing variations in the dynamics of a set of observed trajectories. Directly learning a black-box predictor for each trajectory does not provide any insight into how the dynamics are changing between trajectories. Potential approaches for solving this problem include methods that attempt to directly recover the symbolic governing equations [5–9], which often require a large library of possible terms along with assumptions about the sparsity of the equations, and Koopman operator-based methods [10–13] that map the nonlinear dynamics to an interpretable linear system but struggle with chaotic dynamics. Unsupervised learning approaches that use deep autoencoder architectures to provide interpretability have also been investigated [14, 15].

We propose an alternative approach for discovering varying dynamical parameters that is based on echo state networks (ESNs), a type of reservoir computing architecture [16, 17]. ESNs provide a

*The first two authors have equal contributions.

NeurIPS 2021 AI for Science Workshop.

fast and easy way to build predictive models for dynamical systems [18–20] and have been shown to perform remarkably well when trained to forecast complex nonlinear dynamics exhibiting high-dimensional chaos [4, 19, 21]. While superficially similar to recurrent architectures in deep learning, ESNs are inexpensive to train and do not suffer from exploding or vanishing gradients due to their use of a large sparse reservoir with fixed weights [4, 18]. Correctly constructed ESNs also exhibit the echo state property [16, 22], which provides many nice theoretical guarantees and allows for the ESN to be effectively trained by only fitting the output weights of the final readout layer via linear regression.

However, ESNs are generally considered black-box predictors that have the same issues with interpretability as deep learning approaches. Our work demonstrates that this is not the case. By sharing the fixed weights across a series of separately trained ESNs, we can interpret the learned output weights from the readout layer as a representation of the system dynamics and use manifold learning to extracted the varying dynamical parameters affecting the data. Using the chaotic Lorenz system [23] as an example, we show that this approach is able to quickly and accurately learn high-quality predictive models and then extract an embedding which encodes three distinct varying parameters that affect the dynamics of the system.

2 Methods

To produce a set of interpretable representations of the system dynamics, we train a series of echo state networks (ESNs) that share the same reservoir and input weights. Each ESN is trained on a distinct trajectory in the dataset and, after training, produces a linear readout layer which we use as our representation of the dynamics. We then analyze the geometry of the readout layer weights using a manifold learning method—in our case, principal component analysis (PCA)—and extract parameters that account for the varying dynamics among the trajectories in the dataset.

2.1 ESN training and prediction

ESNs are reservoir computing architectures that consist of a large, fixed sparse reservoir $\mathbf{A} \in \mathbb{R}^{n\times n}$ with hidden size n, a fixed set of input weights $\mathbf{W}_{\text{in}} \in \mathbb{R}^{n\times d}$, and a trained set of output or readout weights $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d\times n}$. A standard discrete time ESN can model a d-dimensional dynamical system

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} = \mathbf{F}_\theta(\mathbf{u}(t)) \tag{1}$$

with dynamical parameters θ and states $\mathbf{u}(t) \in \mathbb{R}^d$ discretized as a time series $\{\mathbf{u}_{t_0}, \mathbf{u}_{t_0+\Delta t}, \ldots\}$ with a fixed time step Δt.

Training—To train the output weights $\mathbf{W}_{\text{out}}$, we first feed a training trajectory $\{\mathbf{u}_t\}_{t=t_0}^{T_{\text{train}}}$ into the ESN in a recurrent fashion

$$\mathbf{r}_{t+\Delta t} = \tanh(\mathbf{A}\mathbf{r}_t + \mathbf{W}_{\text{in}}\mathbf{u}_t), \tag{2}$$

producing a series of hidden reservoir states $\{\mathbf{r}_{t_0}, \mathbf{r}_{t_0+\Delta t}, \ldots\}$ where each $\mathbf{r}_t \in \mathbb{R}^n$ and $\mathbf{r}_{t_0} = \mathbf{0}$ is initialized as the zero vector. The output weights $\mathbf{W}_{\text{out}}$ are then fitted using a (regularized) linear regression to map the hidden states $\mathbf{r}_t$ back to the physical states $\mathbf{u}_t$ such that $\mathbf{u}_t \approx \mathbf{W}_{\text{out}}\mathbf{r}_t$, corresponding to the final readout layer of the ESN. Because the training only involves passing the inputs through the ESN once and then fitting the final readout layer using linear regression, this process is both fast and computationally inexpensive.

Prediction—To predict the future states of a test trajectory $\{\mathbf{u}_t\}_{t=t_0}^{T_{\text{test}}}$, the new trajectory is fed recurrently into the ESN as during training (2), and then future physical states $\hat{\mathbf{u}}_t = \mathbf{W}_{\text{out}}\mathbf{r}_t$ for $t > T_{\text{test}}$ are computed using the output weights from the readout layer. The predicted states $\hat{\mathbf{u}}_t$ are then fed back into the recurrent architecture to obtain subsequent hidden states

$$\mathbf{r}_{t+\Delta t} = \tanh(\mathbf{A}\mathbf{r}_t + \mathbf{W}_{\text{in}}\hat{\mathbf{u}}_t). \tag{3}$$

2.2 Training ESNs with unique and comparable readout layers

Because our approach treats the trained output weights $\mathbf{W}_{\text{out}}$ from the readout layer as a representation of the system dynamics, we require $\mathbf{W}_{\text{out}}$ to be comparable across ESNs trained on distinct

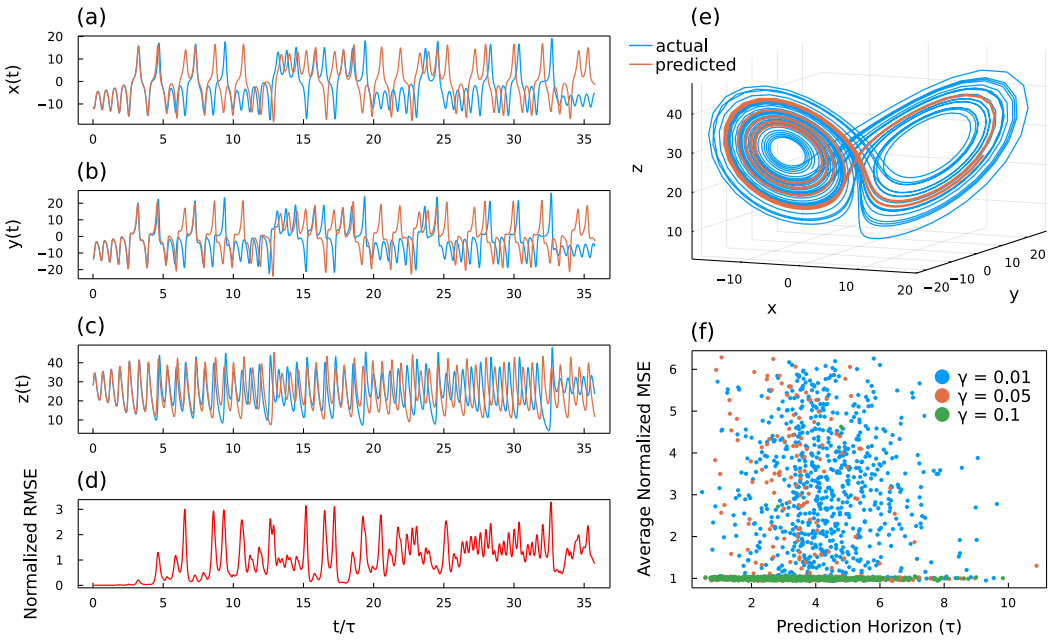

Figure 1: Prediction example for a test trajectory with dynamical parameters $\sigma = 10.3$, $\rho = 29.2$, $\beta = 2.58$. (a)–(c) The plots of the individual states x, y, z of the Lorenz system show the predicted dynamics (orange) compared with the actual ground truth dynamics (blue). Times are given in units of τ, the Lyapunov time of the dynamics. (d) The root mean squared error (RMSE) between the predicted and actual dynamics (normalized by the standard deviation over the entire trajectory) is used to define the prediction horizon, which corresponds to the first time at which the normalized RMSE rises above 1. The prediction horizon for this example is 4.56τ. (e) The phase space plot consists of the full trajectory of the actual dynamics (blue) alongside a partial trajectory of the predicted dynamics (orange), showing the predicted trajectory up to the prediction horizon where the two trajectories diverge. (f) For three sets of ESNs trained with ridge regression parameters $\gamma = 0.01$, 0.05, 0.1, the average mean squared error (MSE) over the entire test trajectory (normalized by twice the variance) is plotted versus the prediction horizon for each example in the test set. We ultimately chose to use $\gamma = 0.1$, which results in good long-term prediction stability as indicated by an average normalized MSE that is consistently low and close to 1.

trajectories with varying dynamics. To ensure this, we use the same fixed reservoir and input weights for all our ESNs and impose L_2 regularization on the linear regression (i.e. ridge regression) when fitting $\mathbf{W}_{\text{out}}$. Along with proper initialization of the ESN to ensure the echo state property (ESP) is fulfilled [16, 22], these conditions also maintain uniqueness among the learned readout layers corresponding to different dynamics. The ESP guarantees that each input sequence $\{\mathbf{u}_t\}_{t=-\infty}^{T-1}$ corresponds to a unique reservoir state $\mathbf{r}_T$. Then, because the dynamical system is Markovian (1), each set of dynamical parameters θ and physical state $\mathbf{u}_T$ have a unique input sequence history and so correspond to a unique $\mathbf{r}_T$. Therefore, the mapping $\mathbf{r}_T \to \mathbf{u}_T$, which is approximated by the readout layer, is also unique for each θ. Finally, by using ridge regression to fit the readout layer—a common practice for training ESNs—we obtain a unique set of output weights $\mathbf{W}_{\text{out}}$ for each θ.

2.3 Extracting dynamical parameters using manifold learning

Once we have the dynamics encoded in the output weights $\mathbf{W}_{\text{out}}$, we can analyze the geometry of this representation by performing manifold learning. In our experiments, we found that applying PCA to the flattened output weights was enough to extract a reasonable low-dimensional embedding that encodes the dynamical parameters θ.

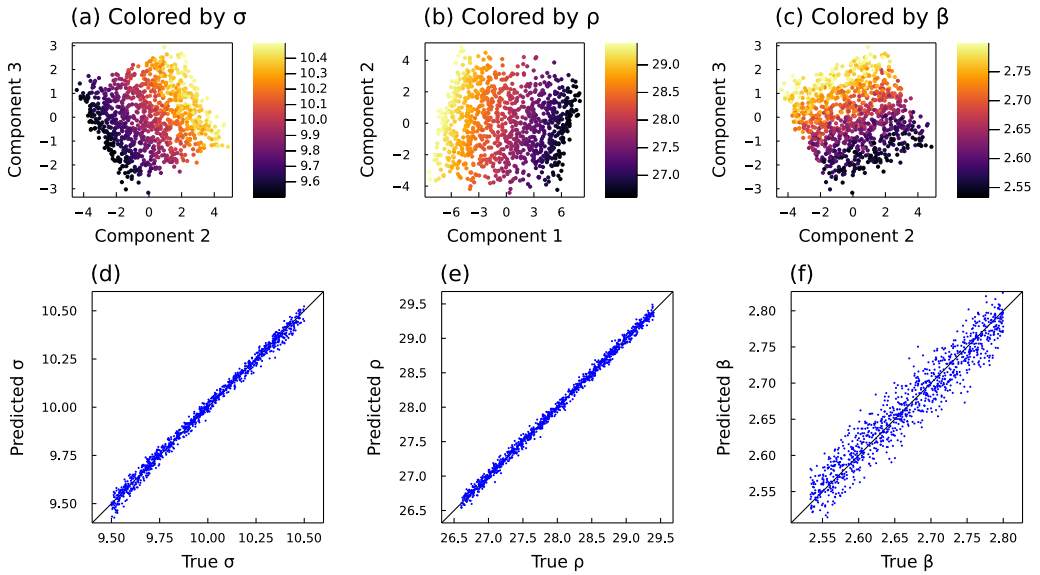

Figure 2: Three-dimensional embedding of the varying dynamical parameters σ, ρ, β of the Lorenz system extracted from the trained output weights $\mathbf{W}_{\text{out}}$ using PCA. (a)–(c) The scatter plots show cross-sections of the extracted embedding colored by the ground truth parameters σ, ρ, β. ρ is primarily aligned with component 1, while σ and β are embedded in components 2 and 3. (d)–(f) The predicted dynamical parameters, based on a linear fit from the extracted embedding, are plotted versus the ground truth parameters. The parameters σ and ρ are very well represented with $R^2 = 0.993$ and 0.996, respectively, while the embedding of β is slight noisier with $R^2 = 0.925$.

3 Lorenz System Experiment

The Lorenz system [23] is a three-dimensional chaotic dynamical system governed by

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y - x) \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho - z) - y \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z \end{aligned} \tag{4}$$

with the parameters σ, ρ, β traditionally chosen to be 10, 28, $8/3$, respectively. For our dataset, we generate 1000 trajectories each with 300,000 time steps ($\Delta t = 0.02$) for training and a subsequent 50,000 time steps for testing. For each trajectory, we independently sample the three dynamical parameters σ, ρ, β from uniform distributions that cover $\pm 5\%$ of their standard values to mimic uncontrolled variations in the dynamics. Using a sparse reservoir (with size 300, degree 6, and a spectral radius of 0.1 as suggested by [20]), an input weight scaling of 0.1, as well as a ridge regression regularization parameter $\gamma = 0.1$, we achieve long-term stability and good prediction performance (Fig. 1) with a mean prediction horizon of 3.2τ. Here, $\tau = \Lambda_{\text{max}}^{-1}$ refers to the Lyapunov time of the chaotic system, i.e. the inverse of the maximum Lyapunov exponent Λ_{max}, which characterizes how quickly two similar trajectories diverge.

Applying PCA to the trained output weights $\mathbf{W}_{\text{out}}$, we can identify the 3 relevant components based on their explained variance ratios (0.750, 0.165, 0.062) using a total explained variance threshold of 0.95. These PCA components provide a three-dimensional embedding that directly corresponds to the varying dynamical parameters σ, ρ, β of the Lorenz system (Fig. 2(a)–(c)). We also obtain high-quality linear fits for the parameters σ, ρ, β from the extracted three-dimensional embedding (Fig. 2(d)–(f)) with R^2 correlation coefficients of 0.993, 0.996, 0.925, respectively.

4 Conclusion

We have empirically confirmed that parameters governing the dynamics of a nonlinear system are encoded in the readout layer of a trained ESN and have demonstrated a method for extracting an embedding of the varying dynamical parameters using PCA. Our proposed method for analyzing data from dynamical systems, including chaotic systems, provides a fast and computationally inexpensive approach for both predicting future states and understanding the uncontrolled variables causing variations in the dynamics. While our method only extracts a latent embedding of the dynamical parameters without an explicit symbolic interpretation, the geometry of this embedding is already very informative on its own: trajectories with similar dynamics are embedded nearby; each relevant component of the embedding corresponds to a varying parameter; and the intrinsic dimensionality of the embedding manifold (i.e. the number of relevant components) gives the number of varying parameters. Given the extracted embedding, it should also be feasible to approximately reconstruct ESN output weights as a function of the identified dynamical parameters, effectively creating a tunable model for the dynamics learned from the data. This would aid in directly interpreting unknown dynamical parameters extracted using our method, so we hope to further explore this direction in the near future. Also, because our approach builds on top of standard ESN training methods, it would be straightforward to adapt for any applications where ESNs are currently being used purely for prediction. These applications include modeling EEG signals [24], making atmospheric climate predictions [25], and scaling up ESNs to model high-dimensional chaotic attractors [21]. As such, we plan to test our method on a wider variety of examples, including systems exhibiting spatiotemporal chaos, as well as investigate the effects of noise on our approach.

Acknowledgments and Disclosure of Funding

We would like to acknowledge useful discussions with Rumen Dangovski, Samuel Kim, Andrew Ma, and Charlotte Loh, and Dr. John Rickert. This research is supported in part by the U.S. Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program and the Center for Excellence in Education (CEE) through the Research Science Institute (RSI) summer program. This work is further supported in part by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, `http://iaifi.org/`). It is also based upon work supported in part by the U.S. Army Research Office through the Institute for Soldier Nanotechnologies at MIT, under Collaborative Agreement Number W911NF-18-2-0048.

References

[1] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. doi: 10.1017/9781108380690.

[2] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Comput.*, 31(7):1235–1270, July 2019. ISSN 0899-7667. doi: 10.1162/neco_a_01199. URL `https://doi.org/10.1162/neco_a_01199`.

[3] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf`.

[4] Hamidreza Eivazi, Luca Guastoni, Philipp Schlatter, Hossein Azizpour, and Ricardo Vinuesa. Recurrent neural networks and koopman-based frameworks for temporal predictions in a low-order model of turbulence. *International Journal of Heat and Fluid Flow*, 90:108816, 2021. ISSN 0142-727X. doi: https://doi.org/10.1016/j.ijheatfluidflow.2021.108816. URL `https://www.sciencedirect.com/science/article/pii/S0142727X21000461`.

[5] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of*

the National Academy of Sciences, 113(15):3932–3937, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1517384113. URL https://www.pnas.org/content/113/15/3932.

[6] S. Rudy, A. Alla, S. Brunton, and J. Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019. doi: 10.1137/18M1191944. URL https://doi.org/10.1137/18M1191944.

[7] Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1906995116. URL https://www.pnas.org/content/116/45/22445.

[8] Kadierdan Kaheman, J. Nathan Kutz, and Steven L. Brunton. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2242):20200279, 2020. doi: 10.1098/rspa.2020.0279. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2020.0279.

[9] Peter Y. Lu, Joan Ariño, and Marin Soljačić. Discovering sparse interpretable dynamics from partial observations, 2021.

[10] Alexandre Mauroy, Yoshihiko Susuki, and Igor Mezić. *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer International Publishing, Cham, 2020. ISBN 978-3-030-35713-9. doi: 10.1007/978-3-030-35713-9_1. URL https://doi.org/10.1007/978-3-030-35713-9_1.

[11] Carl Folkestad, Daniel Pastor, Igor Mezic, Ryan Mohr, Maria Fonoberova, and Joel Burdick. Extended dynamic mode decomposition with learned Koopman eigenfunctions for prediction and control. In *2020 American Control Conference (ACC)*, pages 3906–3913, 2020. doi: 10.23919/ACC45564.2020.9147729.

[12] Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern Koopman theory for dynamical systems, 2021.

[13] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3a835d3215755c435ef4fe9965a3f2a0-Paper.pdf.

[14] David Zheng, Vinson Luo, Jiajun Wu, and Joshua B. Tenenbaum. Unsupervised Learning of Latent Physical Properties Using Perception-Prediction Networks. In *Conference on Uncertainty in Artificial Intelligence*, 2018. URL http://auai.org/uai2018/proceedings/papers/191.pdf.

[15] Peter Y. Lu, Samuel Kim, and Marin Soljačić. Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning. *Phys. Rev. X*, 10:031056, Sep 2020. doi: 10.1103/PhysRevX.10.031056. URL https://link.aps.org/doi/10.1103/PhysRevX.10.031056.

[16] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[17] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009. ISSN 1574-0137. doi: https://doi.org/10.1016/j.cosrev.2009.03.005. URL https://www.sciencedirect.com/science/article/pii/S1574013709000173.

[18] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004. doi: 10.1126/science.1091277. URL https://www.science.org/doi/abs/10.1126/science.1091277.

[19] Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, and Edward Ott. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(12):121102, 2017. doi: 10.1063/1.5010300. URL https://doi.org/10.1063/1.5010300.

[20] Alexander Haluszczynski and Christoph Räth. Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103143, 2019. doi: 10.1063/1.5118725. URL https://doi.org/10.1063/1.5118725.

[21] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120:024102, Jan 2018. doi: 10.1103/PhysRevLett.120.024102. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.024102.

[22] Izzet B. Yildiz, Herbert Jaeger, and Stefan J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 2012. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2012.07.005. URL https://www.sciencedirect.com/science/article/pii/S0893608012001852.

[23] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml.

[24] Rahma Fourati, Boudour Ammar, Javier Sanchez-Medina, and Adel M. Alimi. Unsupervised learning in reservoir computing for EEG-based emotion recognition. *IEEE Transactions on Affective Computing*, pages 1–1, 2020. doi: 10.1109/TAFFC.2020.2982143.

[25] Troy Arcomano, Istvan Szunyogh, Jaideep Pathak, Alexander Wikner, Brian R. Hunt, and Edward Ott. A machine learning-based global atmospheric forecast model. *Geophysical Research Letters*, 47(9):e2020GL087776, 2020. doi: https://doi.org/10.1029/2020GL087776. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020GL087776.