Recovering the parameters underlying the Lorenz-96 chaotic dynamics

Soukayna Mouatadid[1], Pierre Gentine[2], Wei Yu[1], Steve Easterbrook[1]

[1]Department of Computer Science, University of Toronto, [2]Department of Earth and Environmental Engineering, Earth Institute, and Data Science Institute, Columbia University

UNIVERSITY OF TORONTO

HOW MUCH CO_2 CAN ACCUMULATE IN THE ATMOSPHERE BEFORE WE CROSS 2°C WARMING?

- Global warming target of 2015 Paris Agreement: 2°C
- How much CO_2 can accumulate in the atmosphere before this threshold is crossed?

➔ No certain answer.

➔ Answers vary from 480 ppm in 2030, to 600 ppm after 2060 (Schneider et al., 2017a).

- Estimated economic value of an accurate answer:

➔ $10 trillions in savings (Hope, 2015).

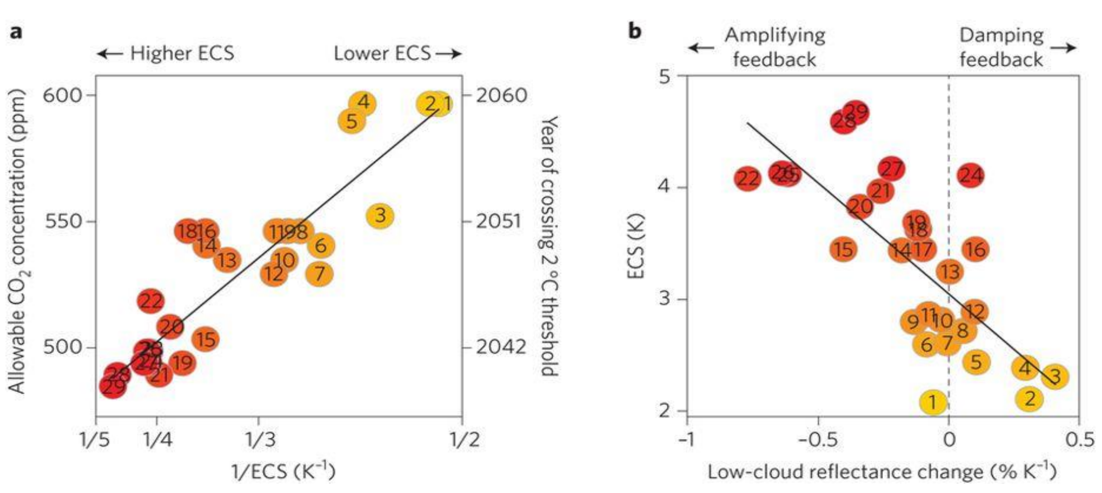

Figure 1. Dependence of climate goals on equilibrium climate sensitivity (ECS) and of ECS on low-cloud feedback (Schneider et al., 2017a).

PARAMETRIZATION SCHEMES

- Typical GCM grid scale: 10 to 150 km.
- Cloud formation scales: ~2 km or less.

➔ Clouds cannot be resolved by current climate models.

➔ Clouds modeled by heuristically approximated parametrization schemes.

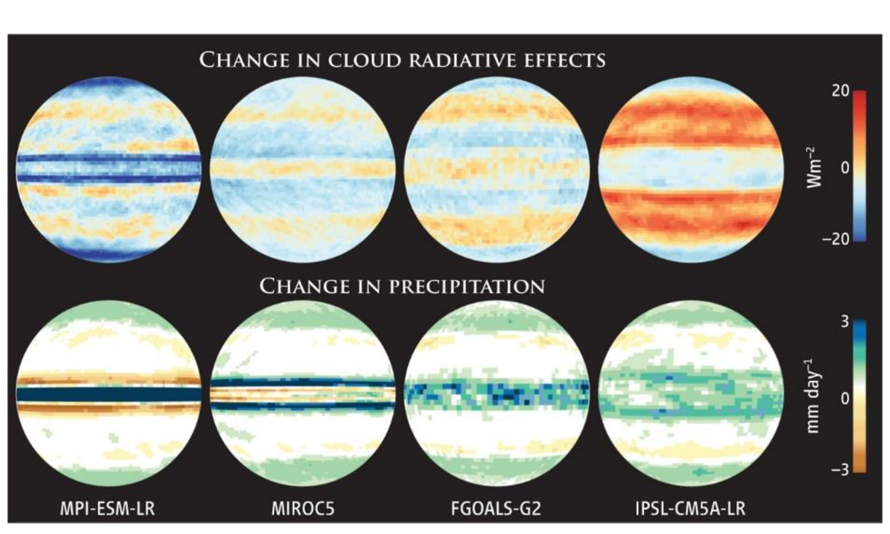

Figure 2. The response patterns of clouds and precipitation to warming vary dramatically depending on the climate model, even in the simplest model configuration. Shown are changes in the radiative effects of clouds and in precipitation accompanying a uniform warming (4°C) (Stevens and Bony, 2013).

RELATED WORK

- Learn cloud parametrizations from cloud resolving model simulation data using:

➔ Single-layers feed-forward neural networks (Krasnopolsky et al., 2013; Gentine et al., 2018; Brenowitz and Bretherton, 2018)

➔ Ensemble Kalman inversion model (Schneider et al., 2017b)

➔ Random forest model (O'Gorman and Dwyer, 2018)

➔ Multi-layer feed-forward neural network (Rasp et al., 2018)

But…

OBJECTIVE

Want:

- A climate model which can objectively zoom in on clouds.

Where does ML fit in?

- Can a deep learning model recover the parameters underlying a cloud parameterization scheme ?

Objective:

- Recover the parameters underlying the chaotic behavior of the Lorenz-96 model.

LORENZ-96 MODEL

Slow large-scale variables x_i (i=1,2,..., I):

$$\frac{dX_i}{dt} = -X_{i-1}(X_{i-2} - X_{i+1}) - X_i + F - \boldsymbol{hc}\bar{Y}_i$$

$$\bar{Y}_i = \frac{1}{J}\sum_{i=1}^{J} Y_{i,j}$$

Fast small-scale variables $y_{i,j}$ (i=1,2,...,I; j=1,2,...,J):

$$\frac{1}{c}\frac{dY_{i,j}}{dt} = -\boldsymbol{b}Y_{i+1,j}(Y_{i+2,j} - Y_{i-1,j}) - Y_{i,j} + \frac{h}{j}X_j$$

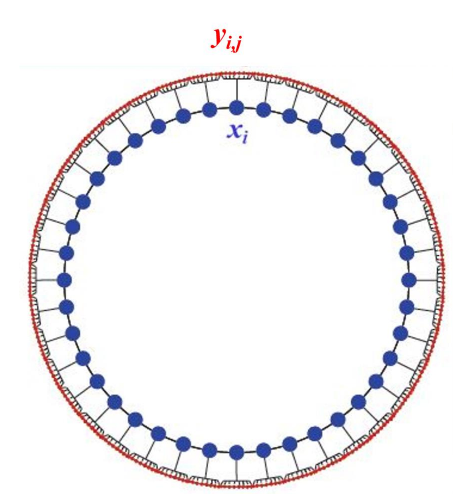

Figure 3. The Lorenz-96 model with coupled slow large-scale variables and fast small-scale variables.

- *b*: controls the amplitude of the nonlinear interactions among the fast variables, while the parameter
- *c*: controls how rapidly the fast variables fluctuate relative to the slow variables and the parameter
- *h*: controls strength of the coupling between the fast and slow variables

- **Lorenz-96 configuration:**
- Four slow variables, each associated with four fast variables
- F = 8 forcing used in original configuration of the Loren-96 model
- Accumulated over 50,000 steps, with a time step of 0.01
- Parameters b, c and h are sampled from gaussian distributions with mean 11 and standard deviation 5 for b and c and mean 1 and standard deviation 0.1.

RECOVERING B, C AND H

➔ Generate 200 trajectories of the 20 variables in the L-96 model

➔ Convert these trajectories to grayscale images of shape (20, 50000)

➔ Chunk images into tiles of shape (20, 20)

➔ Flatten image chunks used as FC models inputs

➔ Image chunks used as convolutional models inputs

➔ FC and Conv models used to predict the b, c and h parameters

Loss function:

- Mean squared error normalized by standard deviation:

$$WMSE = \frac{1}{n}\left[\frac{1}{\sigma_b}\sum_{i=1}^{n}(b_i - \hat{b}_i)^2 + \frac{1}{\sigma_c}\sum_{i=1}^{n}(c_i - \hat{c}_i)^2 + \frac{1}{\sigma_h}\sum_{i=1}^{n}(h_i - \hat{h}_i)^2\right]$$

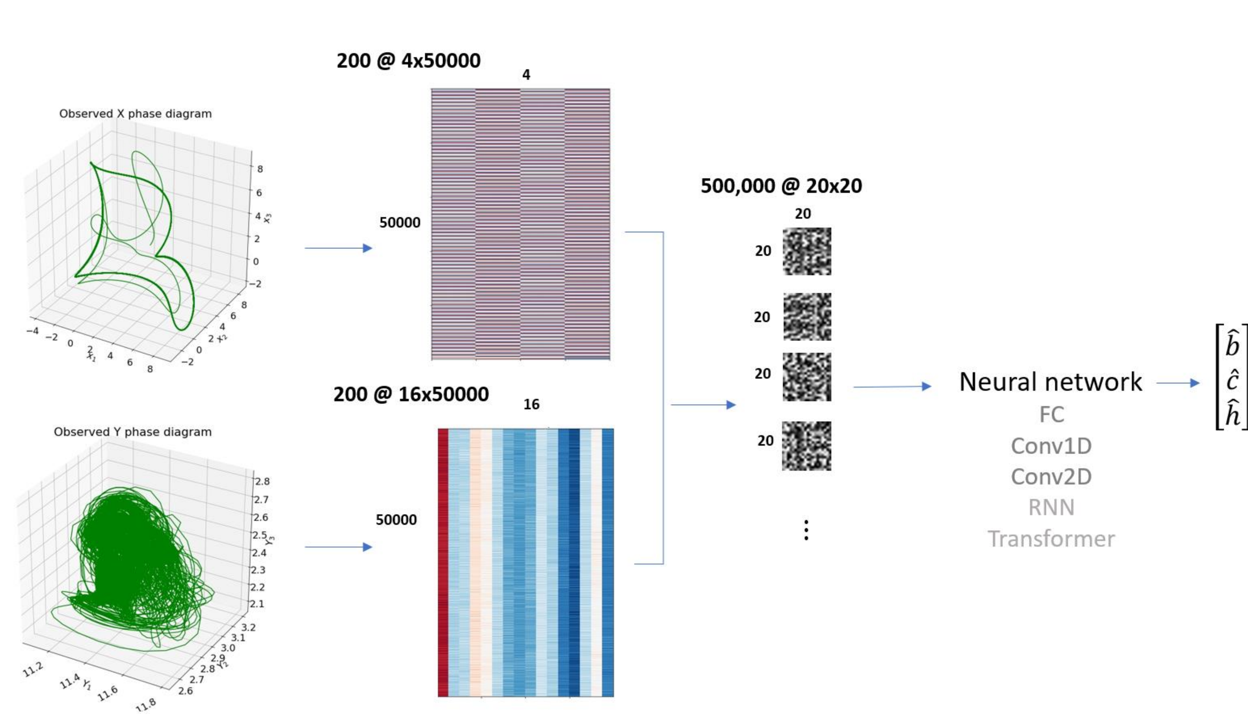

Figure 4. Overview of the L-96 trajectories pre-processing and their underlying parameters recovery.

LEARNING MODELS

Learning algorithms:

➔ Fully connected with 3 layers

➔ 1D convolutional with 2 layers (each with 32 filters of size 3) followed by 3 FC layers

➔ 2D convolutional with 1 layer (32 filters of size 3x3) followed by 3 FC layers

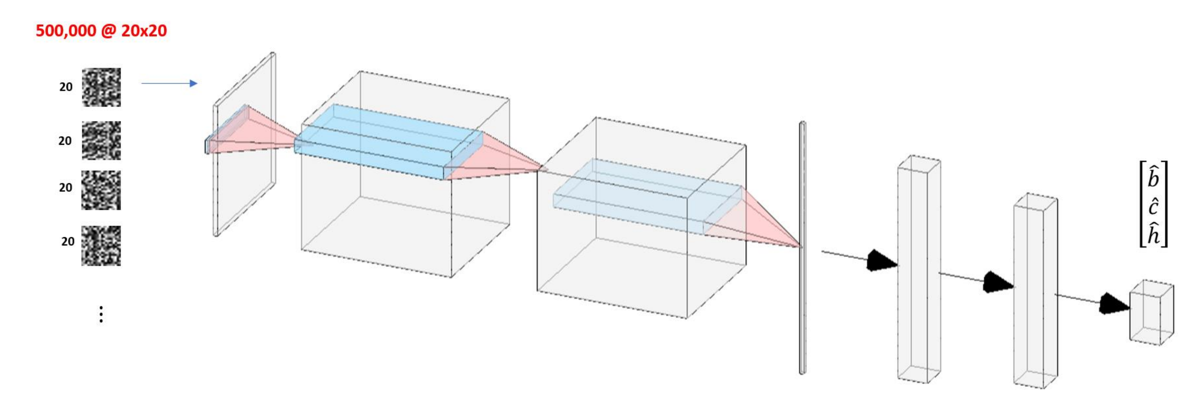

Figure 5. The image chunks used as inputs to a 1D convolutional model with 2 layers of 32 filters (size 3) each the Lorenz-96 model with coupled slow large-scale variables and fast small-scale variables.

RESULTS

Table 1. Training and testing loss and coefficient of determination.

TEST MODE	MODEL	TRAIN LOSS	TEST LOSS	TRAIN r^2	TEST r^2
		LEARNING FROM X AND Y			
False	LR	1.7512	1.7560	0.7588	0.7578
	FC	0.6583	0.6714	0.9094	0.9074
	CONV1D	0.6682	0.6812	0.9079	0.9060
	CONV2D	0.6502	0.6861	0.9105	0.9054
		LEARNING FROM Y ONLY			
	LR	1.7394	1.7429	0.7605	0.7597
	FC	0.6647	0.6808	0.9084	0.9061
	CONV1D	0.6968	0.7073	0.9041	0.9024
	CONV2D	0.6744	0.7063	0.9071	0.9026
		LEARNING FROM X AND Y			
True	LR	1.7371	2.9112	0.7609	0.6059
	FC	0.7064	1.3262	0.9028	0.8212
	CONV1D	0.7029	1.2822	0.9031	0.8263
	CONV2D	0.6577	1.3260	0.9070	0.8125
		LEARNING FROM Y ONLY			
	LR	1.7407	2.9268	0.7604	0.6039
	FC	0.6805	1.3197	0.9063	0.8220
	CONV1D	0.6898	1.2726	0.9050	0.8276
	CONV2D	0.6577	1.3260	0.9094	0.8210

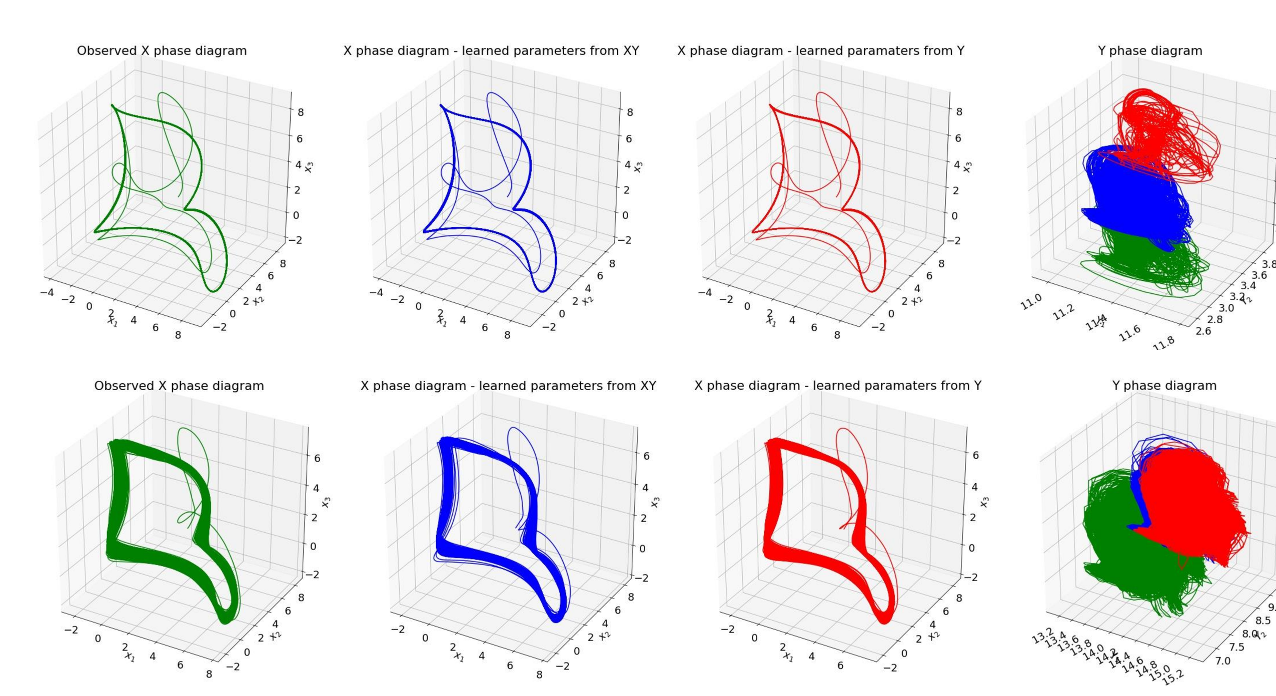

Figure 6. Lorenz-96 phase diagram of the first three slow (X) and fast (Y) variables using observed parameters (green), learned parameters from the X and Y variables (blue) and learned parameters from the Y variables only (red). The learning algorithm is a fully connected network with *test_mode* set to *False* (top row). The learning algorithm is a 1D convolutional model with *test_mode* set to *True* (bottom row).

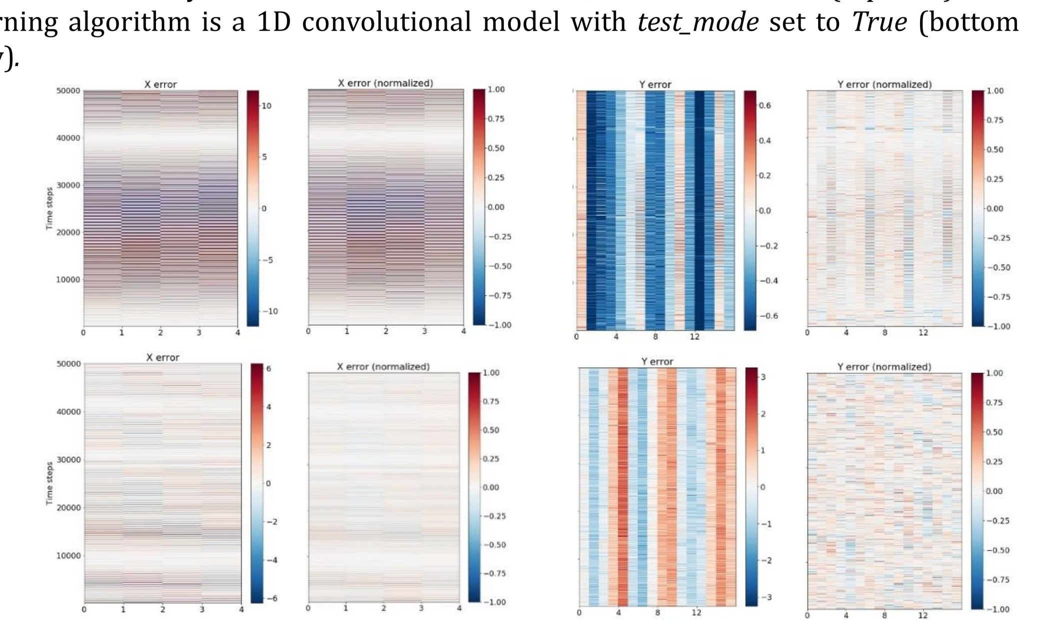

Figure 7. Errors between the Lorenz-96 slow (X) and fast (Y) variables generated using the observed parameters and the inferred parameters using the FC model trained on the Y variables with *test_mode* set to *False* (top row) and using the Conv1D model trained on Y variables with *test_mode* set to *True* (bottom row).

References

- Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*.
- Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock?. *Geophysical Research Letters*.
- Hope, C. (2015). The $10 trillion value of better information about the transient climate response. *Phil. Trans. R. Soc. A*, *373*(2054), 20140429.
- Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. *Advances in Artificial Neural Systems*, *2013*, 5.
- O'Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change and extreme events. *arXiv preprint arXiv:1806.11037*.
- Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent sub-grid processes in climate models. *arXiv preprint arXiv:1806.04731*.
- Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, K. G., Schär, C., & Siebesma, A. P. (2017a). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*(1), 3.
- Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017b). Earth System Modeling 2.0: A Blueprint for Models That Learn From Observations and Targeted High-Resolution Simulations. *Geophysical Research Letters*, *44*(24).
- Stevens, B., & Bony, S. (2013). What are climate models missing?. *Science*, *340*(6136), 1053-1054.