DARPA ATCM PI Meeting
Aug 16-17, 2022

Hybridizing Knowledge-Based and Machine Learning Models for Climate and Tipping-Point Prediction

Presented By
Edward Ott (U. Maryland, PI)

Team: University of Maryland, Texas A&M University, Potomac Research LLC

Climate change is necessarily a non-stationary process.

Predicting Non-Stationary Dynamics and Tipping Point Behavior

Overview

- Due to non-stationarity, the predicted future dynamical behavior will differ from past observed data available for training.

- This requires rethinking ML procedures, as well as protocols, for training and validation that are commonly used for predicting stationary system behavior.

- We have developed machine learning techniques for prediction of non-stationary systems. (See Patel et al., Chaos **31**, 033149 (2021), Patel and Ott, arXiv:2207.0051 [cs.LG]). We plan to incorporate these techniques into our full terrestrial atmospheric ML/knowledge-based climate model.

Overview

- We now give an illustrative example applying our methods on a small test system.

Stationary Lorenz ‘63 equations

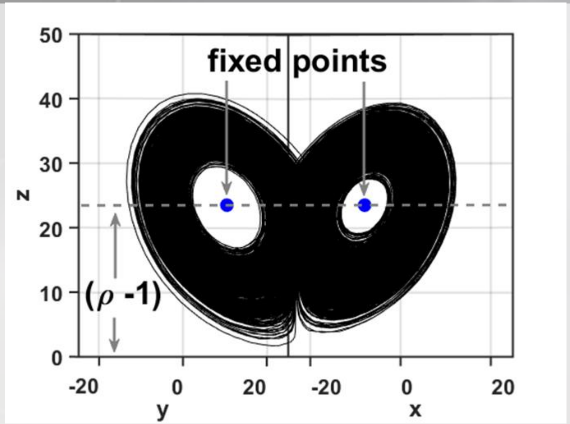
- Models a two-dimensional fluid layer uniformly warmed from below and cooled from above

$$\frac{dx}{dt} = \sigma(y - x) + \xi_x(t)$$

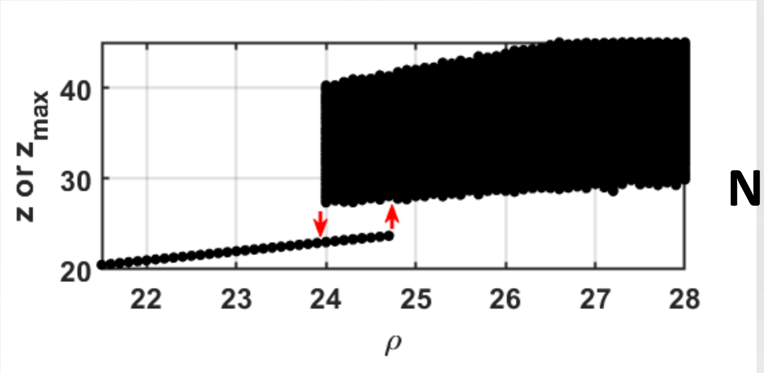
$$\frac{dy}{dt} = x(\rho - z) - y + \xi_y(t)$$

$$\frac{dz}{dt} = xy - \beta z + \xi_z(t)$$

- $\xi_i(t)$ is uncorrelated and white (in time) dynamical noise



Noiseless

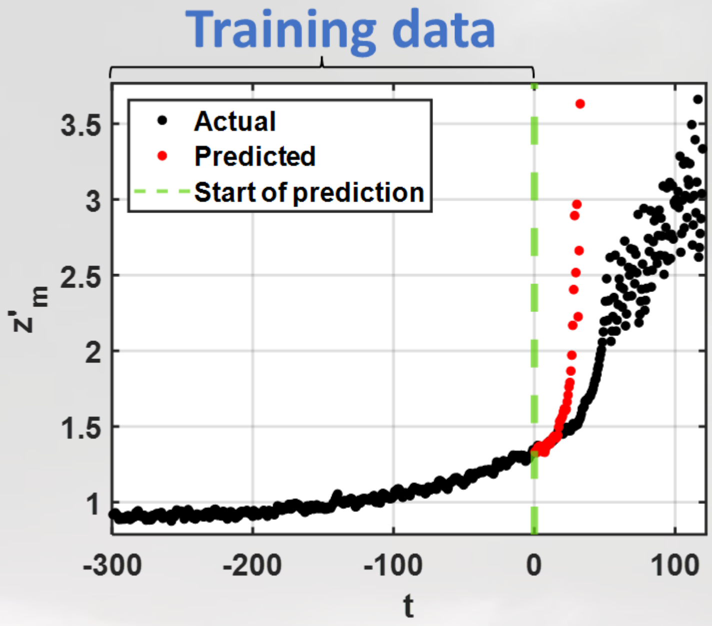


Noiseless

Non-Stationary, Noisy Lorenz '63 equations: ML-Only

System parameter time-dependence:

$$\rho(t) = \rho_0 + \rho_1 \exp\left(\frac{t}{\tau}\right)$$



z'_m are the (rescaled) z-values at which $xy - \beta z = 0$

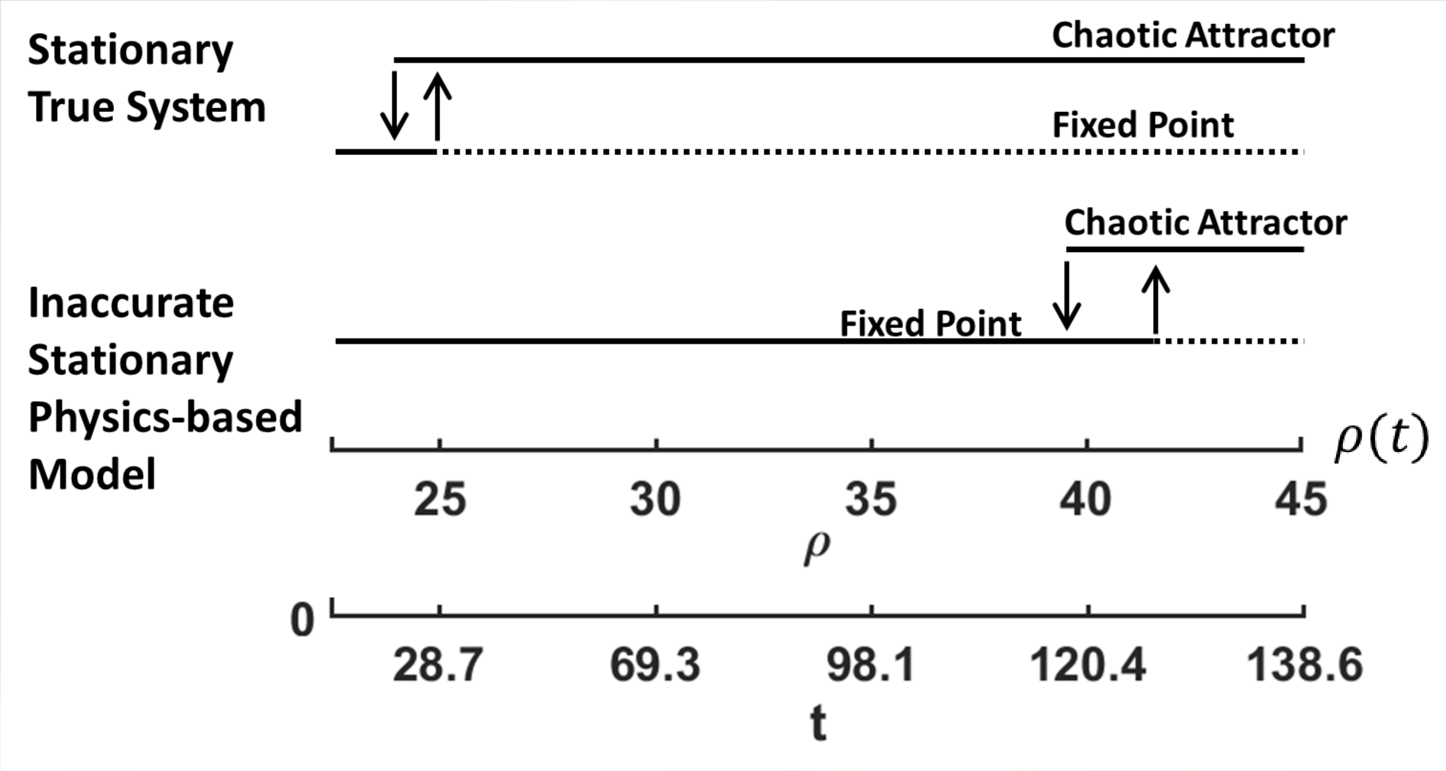
The ML succeeds at predicting an approximation to the tipping point, but fails to get the post-tipping-point behavior.

Can we do better by incorporating an available, although inaccurate, physics-based model?

⇓

ML/knowledge-based hybrid model

Non-Stationary Lorenz ‘63 equations: physics-based model

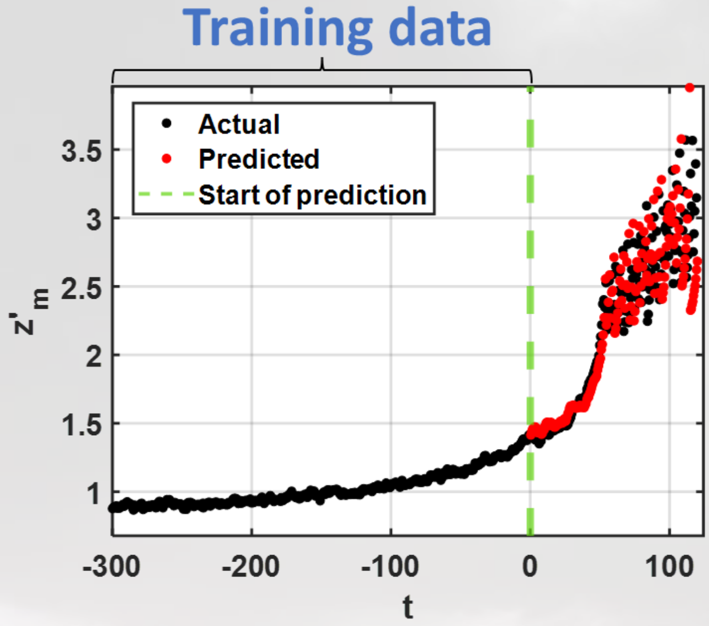


$$\rho(t) = \rho_0 + \rho_1 \exp\left(\frac{t}{\tau}\right)$$

Non-Stationary, Noisy Lorenz ‘63 equations: hybrid

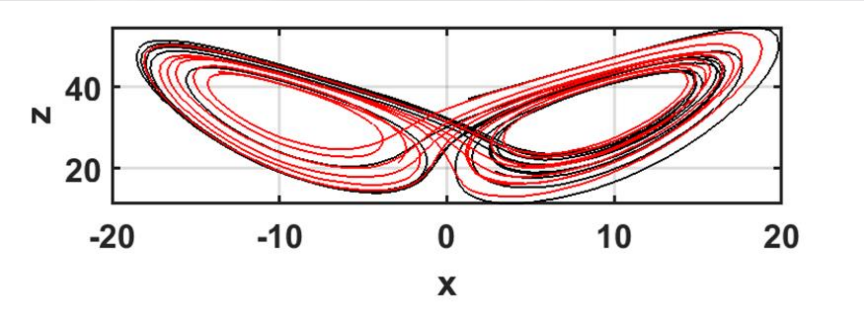
System parameter time-dependence:

$$\rho(t) = \rho_0 + \rho_1 \exp\left(\frac{t}{\tau}\right)$$



z'_m are the (rescaled) z-values at which $xy - \beta z = 0$

$90 \leq t \leq 100$



The hybrid ML is:

- ✓ able to predict the tipping point
- ✓ able to predict post-tipping-point behavior on the larger region of state space

Our Team

Prof. Edward Ott, Dept. of Physics and Dept. of EE (Univ. of MD)

Prof. Michelle Girvan, Dept. of Physics (Univ. of MD)

Prof. Brian Hunt, Dept. of Math (Univ. of MD)

Dr. Andrew Pomerance, Potomac Research LLC

Prof. Istvan Szunyogh, Dept. of Atmos. Sci. (Texas A&M)

Graduate Students

Troy Arcomano (Texas A&M, Atmos. Sci.)

Dhruvit Patel (UMD, Physics)

Mitchell Tsokatos (Texas A&M, Atmos. Sci.)

Alexander Wikner (UMD, Physics)

Weather vs. Climate

<u>Weather prediction task:</u> Produce short-term (~days) forecasts of atmospheric state variable (temperature, rainfall, etc.)

<u>Climate prediction task:</u> Predict long-term (years, decades, …) statistics of atmospheric and oceanic dynamical patterns and average properties (e.g., average temperature and rainfall, frequency of droughts, …)

<u>SOME IMPORTANT POINTS FOR TODAY'S PRESENTATIONS</u>

Climate prediction requires long-term ML system <u>stability</u>.

Climate change by its nature treats a system that is <u>non-stationary</u> in time.

Long time scale dynamical interactions of the atmosphere with slowly evolving components (ocean conditions, ice, plant ecology, …) are important considerations for [redacted]on).

Some Challenges For Our Project

- The Earth climate system is large and complex
- It has extreme heterogeneity
- Subgrid-scale processes are important
- Evolution takes place on multiple time scales
- Evolution, by the nature of climate change, is

Features Of Our System Inspired By These Challenges

- Reservoir computing.
- Hybrid knowledge-based/machine-learning.
- Many convolutionally connected RCs, each trained independently in parallel to predict its own local spatial region.
- A new regularization technique to promote stability.

Main Accomplishments Since Our Last Virtual Site Visit (May 5, 2022)

- Grad student Troy Arcomano got married.
- Progress and initial results on incorporating two-way coupling between atmospheric and oceanic dynamics. (To be presented by Prof. Istvan Szunyogh of Texas A and M.)
- Further development of machine-learning-aided prediction of nonstationary systems. (To be presented by grad student Dhruvit Patel of the University of Maryland.)

Hybrid Machine Learning and General Circulation Model

Reservoir Computing (RC)

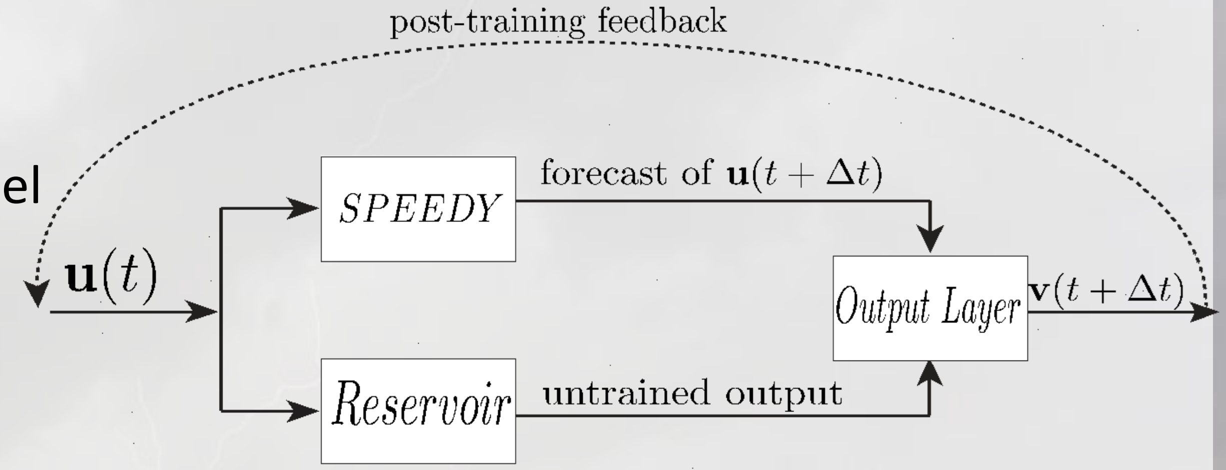
- For ML we use reservoir computing (e.g., Lukoševicius and Jaeger 2009)
  - Recurrent Neural Network (RNN) – enables memory of time series input
  - Simplified training – only the output layer parameters are trained

- Much faster training than other ML methods allows us to perform a lot of exploratory numerical experiments

- Our hybrid approach can be used with any ML method that is trained to make a short-term forecast
  - Short-term forecasts are cycled iteratively to make a long-term forecast

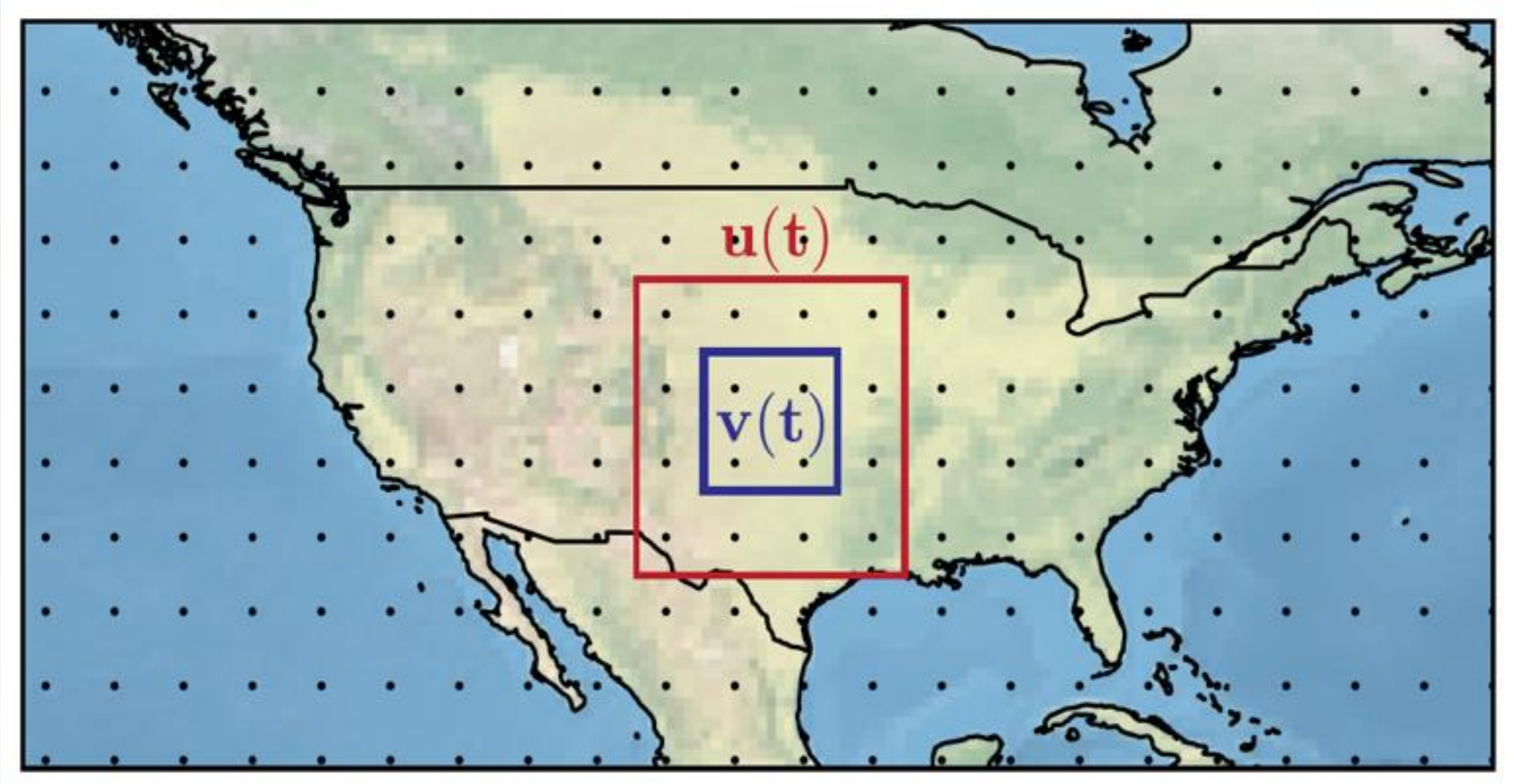
General Hybrid Modeling Approach

- **Atmospheric state: u**(t)
- **SPEEDY:** Atmospheric model
- **Reservoir:** RNN



- **Output Layer:** trained by linear regression
  - **v**(t+Δt) approximates **u**(t+Δt) [or a sub-vector in the parallel version]
- **Parallel version:** ~1000 reservoirs, each forecasts on a separate local region
  - each output layer is independently trained to treat geographic heterogeneity

Parallel-Hybrid Training local input **u**(t), output **v**(t+Δt)

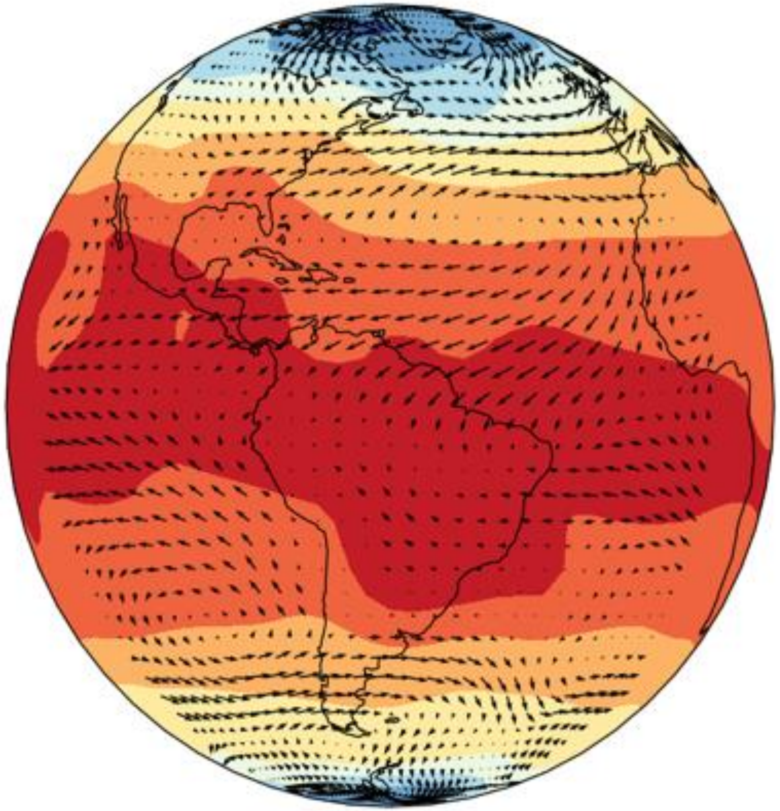


SPEEDY

- **S**implified **P**arameterizations, primitive **E**quation **DY**namics Version 42 of the International Centre for Theoretical Physics (ICTP)
  - (Molteni 2003, Kucharski 2006)

- Equations
  - Fluid dynamics based equations
  - Simplified modeling of unresolved processes

- Configuration
  - 8 vertical layers
  - T30 (~300km)



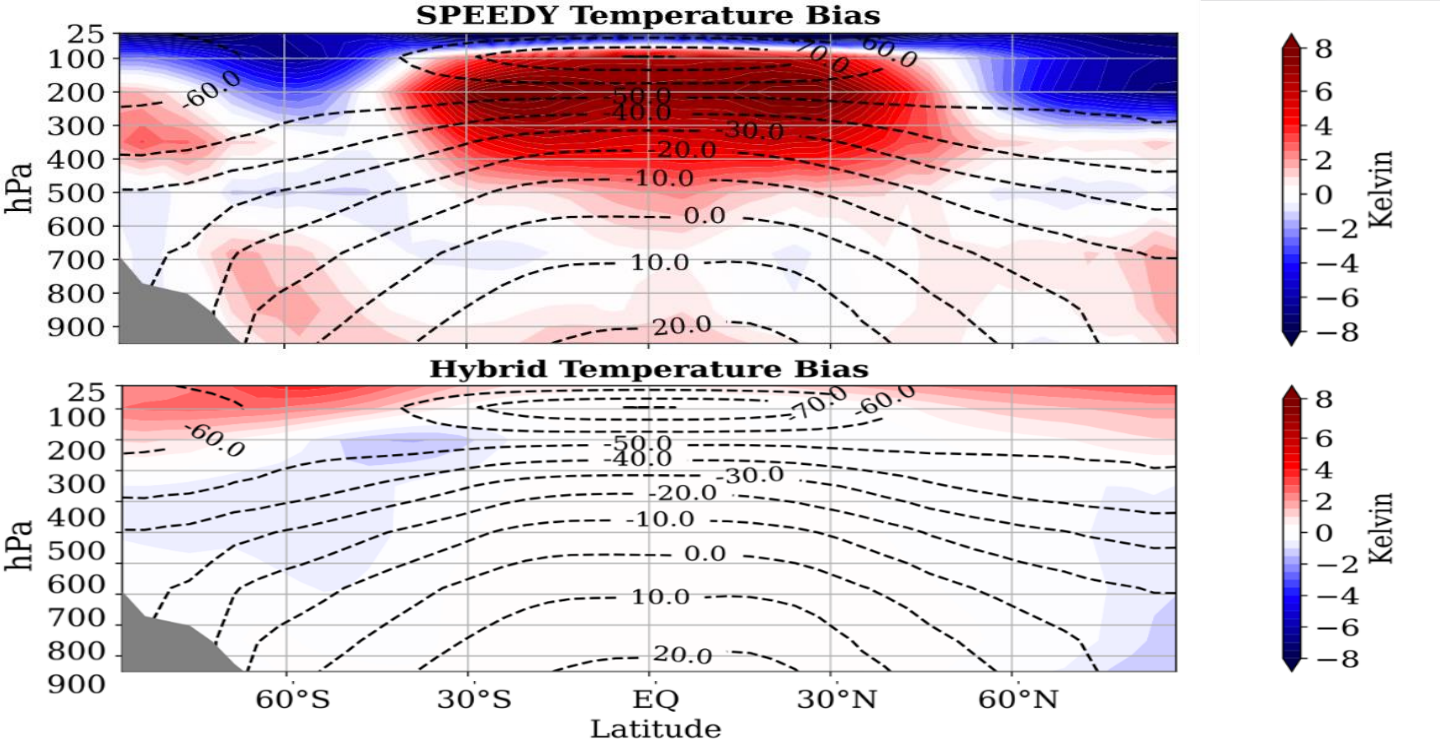
Training

- Regridded ERA-5 Reanalysis data to SPEEDY horizontal and vertical grid
- Used the 5 prognostic variables for SPEEDY
  - Temperature
  - 2 components of the wind
  - Specific Humidity
  - Surface Pressure
- 19 years of data from 1981-1999
- Training minimizes 6 hour forecast error

Climate Simulations

- Ran SPEEDY and the trained parallel-hybrid model from Jan 1, 2000 to Dec 31, 2010
  - 11-year free run with training optimized for 6-hour forecasts
- Both simulations use climatological sea surface temperatures (SSTs)
- We compare our climate statistics to ERA5 for the ten years from 2001 to 2010

Climate Simulations Results (Arcomano et al. 2022)

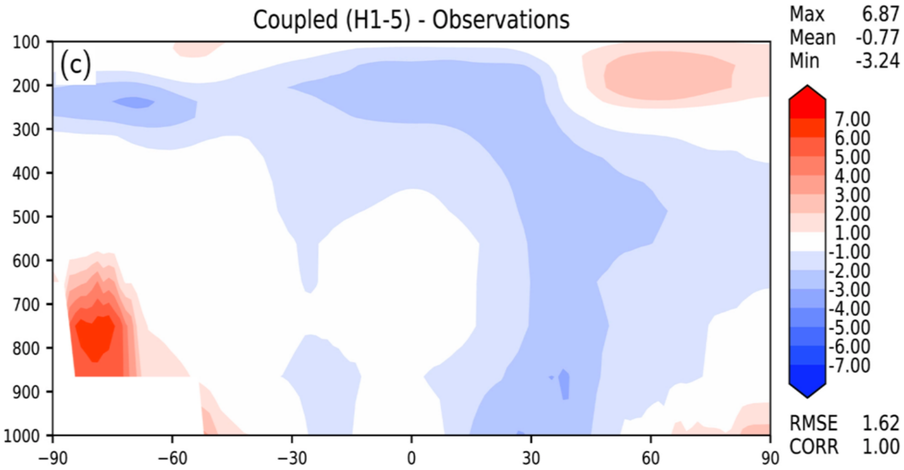


Lower bias indicates better simulation of the 2001-2010 climate represented by ERA5

Comparison with Coupled Ocean-Atmosphere Model

- Energy Exascale Earth System Model
  - Fully coupled Earth system model
  - Run on ~6000 CPU cores
- Our Hybrid Model
  - Climatological boundary conditions
  - Once trained can be run using desktop
    - 10 year simulation ~2 hours

E3SM Coupled Model Version 1



J Adv Model Earth Syst, Volume: 11, Issue: 7, Pages: 2089-2129, First published: 15 March 2019, DOI: (10.1029/2018MS001603)

	Temperature			
Model	Min Bias	Max Bias	Mean Bias	RMSE
E3SM	-3.24	6.87	-0.77	1.62
Hybrid	-0.92	2.38	0.19	0.49
SPEEDY	-6.56	7.11	0.83	3.22

Current Work

- Add vertical localization
  - Limit local regions also in the vertical direction
  - Will improve scalability to higher resolution
  - Account for vertical heterogeneity

- Preliminary results show improved year to year variability for the stratosphere

- Include additional dynamics needed for earth system climate modeling
  - Two way coupling of the atmosphere to the ocean, sea ice, etc.
  - Interim goal is to simulate and predict El Nino dynamics
  - First approach: using ML-component to forecast sea surface temperatures (SSTs)

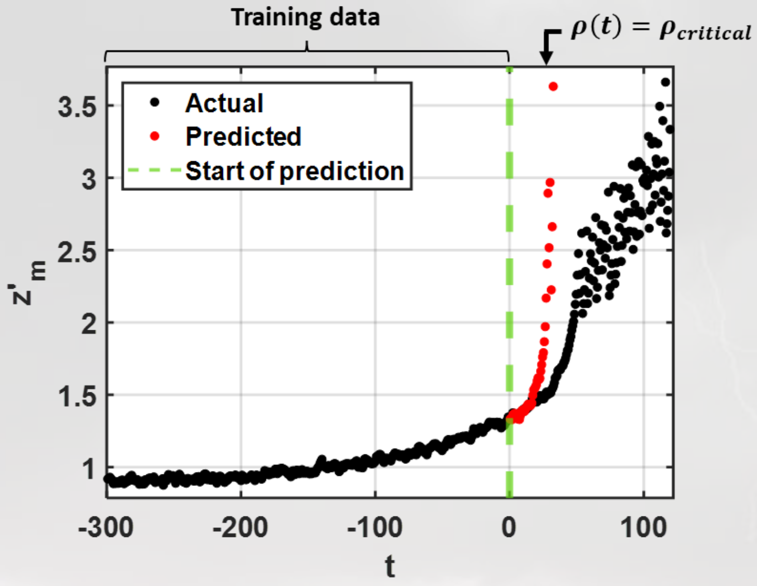
Ocean-Atmosphere Coupling: Preliminary Result

- http://drive.google.com/file/d/1U8Tz_iUGtmXb8M5Qlfcc7MK2U4S1OGOv/view

Non-Stationary Lorenz '63 equations: ML-Only

System parameter time-dependence:

$$\rho(t) = \rho_0 + \rho_1 \exp\left(\frac{t}{\tau}\right)$$



z'_m are the (rescaled) z-values at which $xy - \beta z = 0$

The ML is trained on a pre-tipping-point trajectory restricted to a small region of state space (noisy motion about one of the two slowly-changing fixed points)

The ML is:

- ✓ **able to predict a tipping point (i.e., fixed point loses stability)**
- ✗ **unable to predict the post-tipping-point behavior on the larger region of state space**

A paper reporting our findings related to the type of problem we just illustrated is complete and posted on arxiv at this link.

Future Work

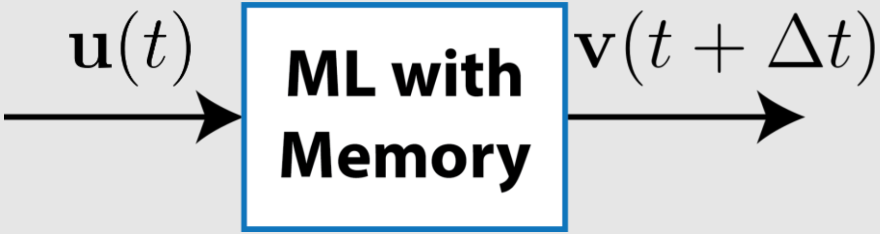
- Transition the methods developed here using small test systems to our more realistic hybrid climate model

- Investigate the ability of machine learning models to learn subgrid scale physics

Long-Term Prediction Stability using Linearized Multi-Noise Training (LMNT)

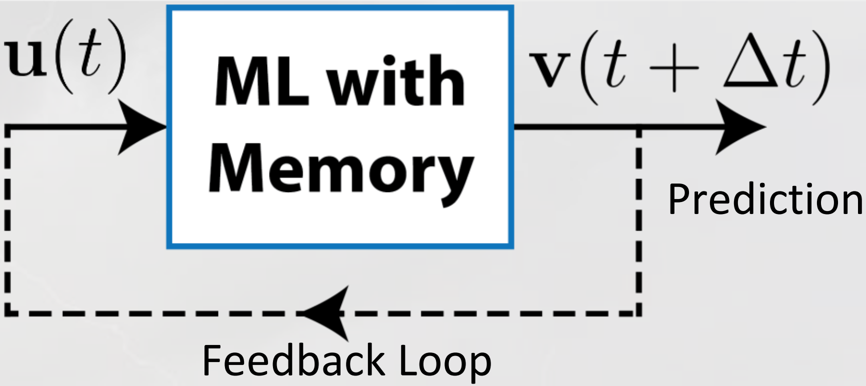
Using ML Models for Long-Term Prediction

"Open-Loop" Training



- ML Model is trained such that $\mathbf{v}(t + \Delta t) \approx \mathbf{u}(t + \Delta t)$
- Δt is short compared to the $\mathbf{u}(t)$ time scale

"Closed-Loop" Prediction



- Long-term predictions of the climate are generated by feeding the Δt predictions back into the ML Model over very many feedback iterations

Feedback Stability During Prediction

- While the training ensures that the Δt prediction error is small, the error during long-term prediction will accumulate with each Δt iteration of the closed-loop prediction model. Thus, long-term weather prediction fails, but it has been found that these long-term ML model iterations may be good as climate predictions, provided they remain stable.

- **Goal:** Achieve long-term stability

- **Method:** Devise appropriate regularization to be applied during training

Stabilization Techniques

- Regularization is often used to prevent ML over-fitting (e.g., Tikhonov Regularization, which penalizes large ML model weights)

- A regularization technique for improving stability during feedback is to add noise to each sample input to the ML model during training
  - This technique is already used in our Hybrid atmospheric prediction model

- Such regularization techniques require the addition of hyperparameters to tune their effect on performance

- In the case of our noise regularization for improved prediction stability, a drawback is that hyperparameter tuning is extremely time consuming and places limits on how many experiments we can run

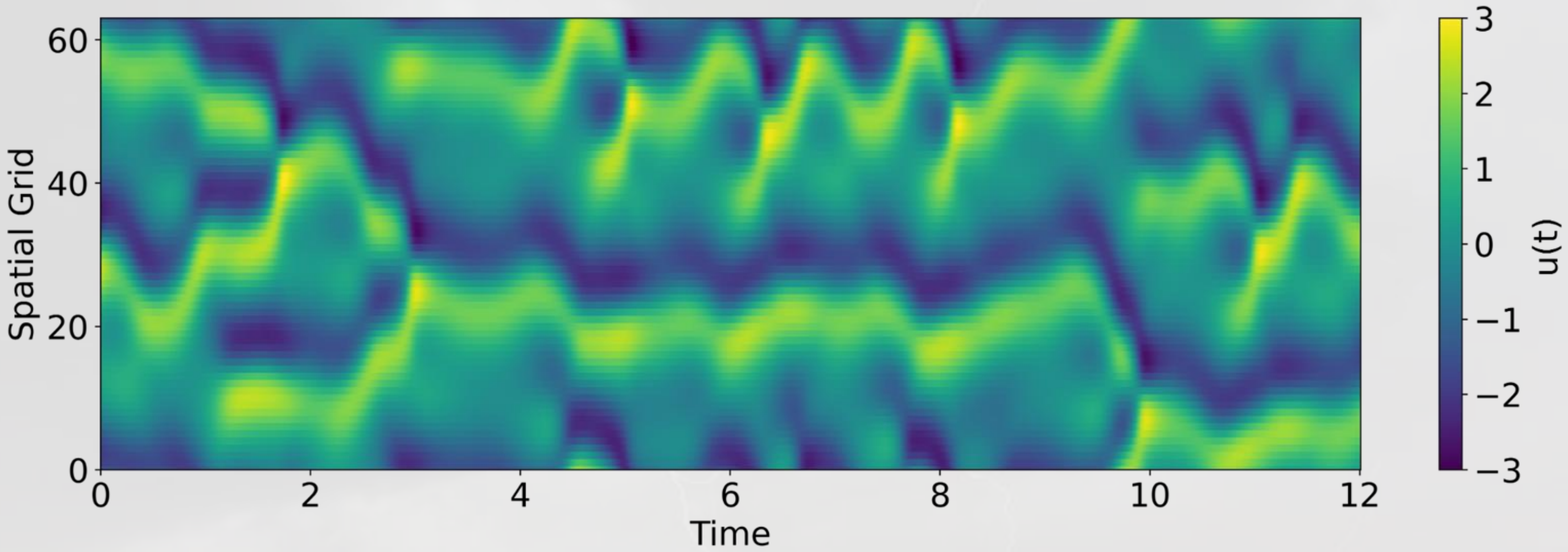
Linearized Multi-Noise Training (LMNT)

- Motivated by the success of the noise training technique, we have analytically devised deterministic, non-stochastic regularization techniques that mimic the effect of the stochastic noise training technique, for which hyperparameter optimization is much faster and for which better performance is often obtained.

- We refer to training with this new regularization as **LMNT**

- This regularization technique has been implemented in our Hybrid climate prediction model, and is being tested

Test System: Kuramoto-Sivashinsky Equation



- 1-Dimensional model of the complex, chaotic spatial evolution of flame fronts, simulated with 64 spatial grid points with periodic boundary conditions

Summary of Some Stability Test Results on the KS Equation

- Tests were conducted using a 500 node reservoir with a long training time.

- Stability tests with no regularization were universally unstable and produced results that diverged to very large, unphysical values.

- Stability tests with Tikhonov regularization (discouraging large ML model weights) required tuning the regularization to such a strong effect that the climate produced was very different from the desired true climate.

- Stability tests with LMNT allowed for tuning strengths with excellent stability and climate reproduction.

Future Work

- A paper is being written on LMNT

- Test how LMNT improves stability and accuracy in our Hybrid climate forecasting model

- Implement a fast hyperparameter optimization scheme using LMNT in our Hybrid climate forecasting model

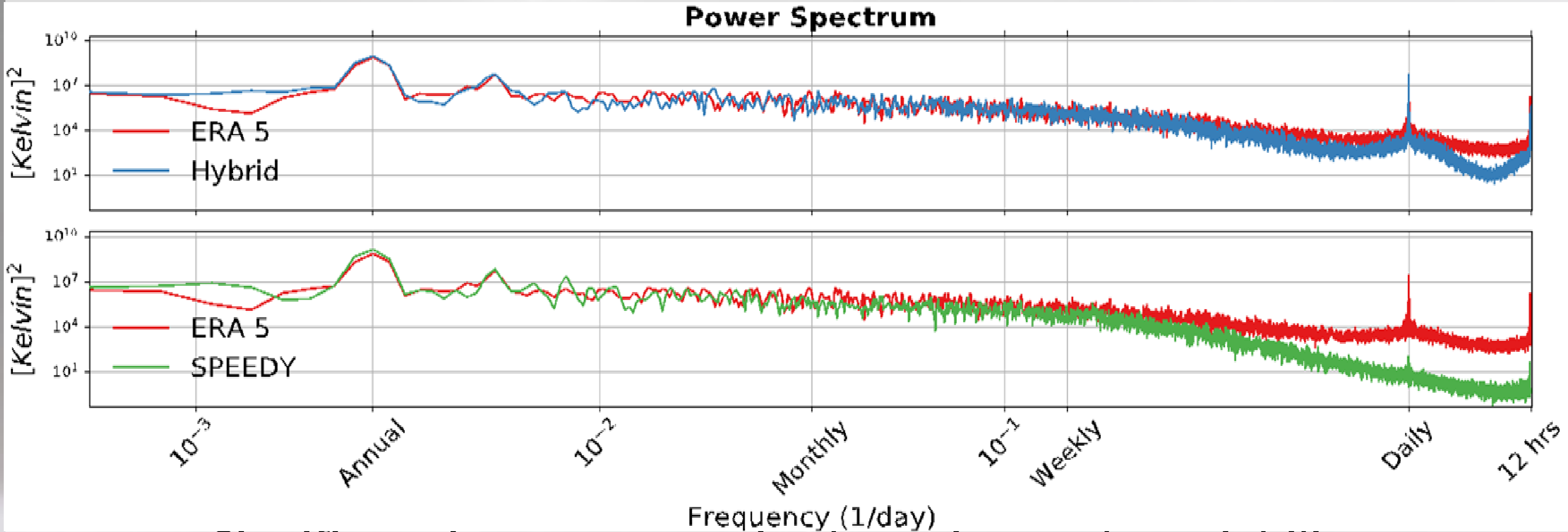
Conclusion/Discussion

- **An unexpected result:** Our low-resolution ML/hybrid atmospheric prediction system does at least as well (apparently better) in reproducing climate as a very much more computationally intensive, slower state-of-the-art DOE climate system.
  - Not a completely fair comparison.
  - Will be fairer after we self-consistently include a ML (or ML/hybrid) ocean component.
  - **Possible implication:** Even at our current, relatively low resolution, our approach may provide a very much faster alternative for obtaining results of similar quality to those of a conventional system.
- Initial results on new stability-promoting techniques indicate that instability may no longer be a problem.

Conclusion/Discussion

- **A main near-term future thrust:** Incorporate, test, and demonstrate non-stationary operation of our system.
- **A question for DARPA:** Status of plans for future follow-on DARPA opportunities for funding support in this field?

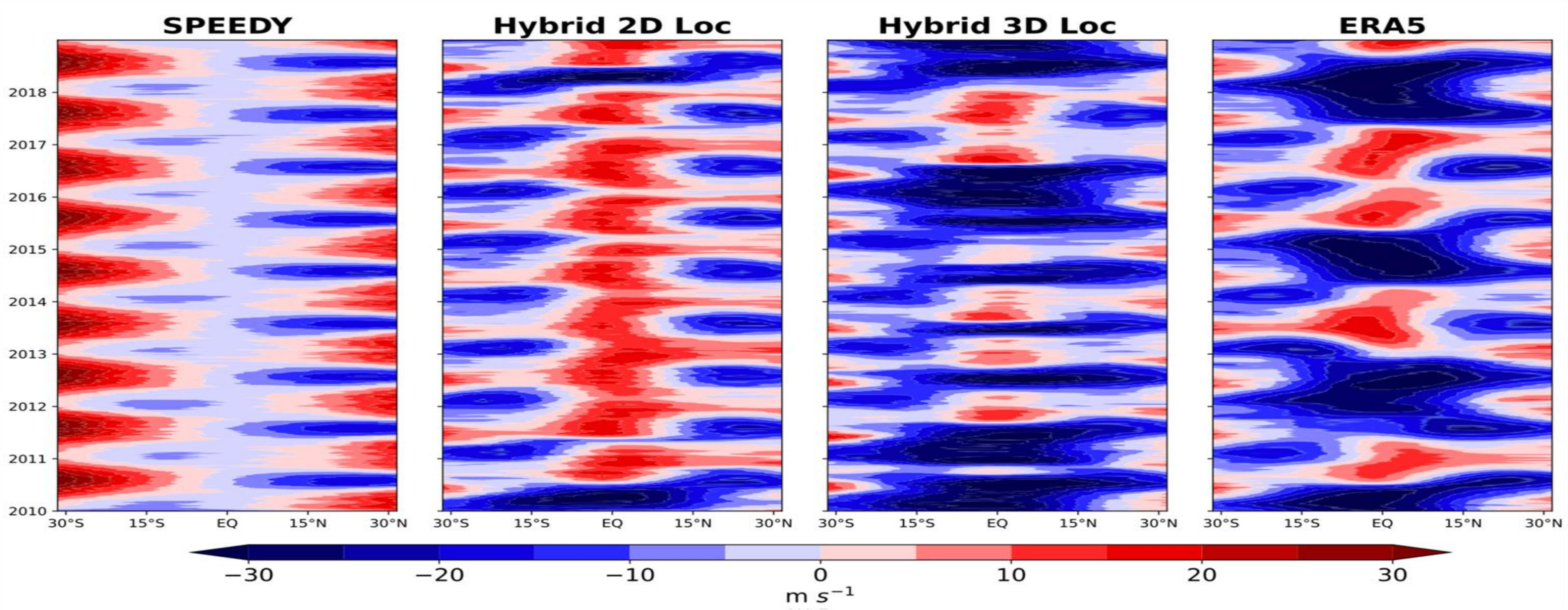
Climate Dynamics



Significant improvement in short timescale variability

Vertical Localization: Preliminary Result

Equatorial Zonal Wind in the Stratosphere



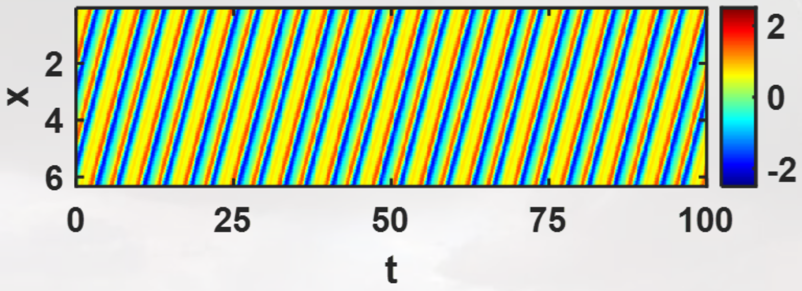
Stationary Kuramoto-Sivashinsky equation

- One-dimensional model of the complex chaotic spatial evolution of flame fronts

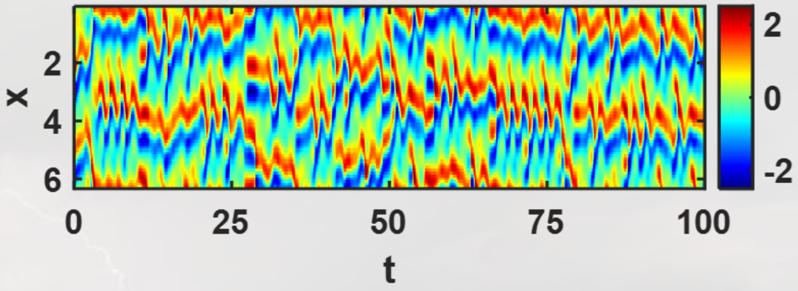
$$\frac{\partial w(x,t)}{\partial t} + w(x,t)\frac{\partial w(x,t)}{\partial x} + \frac{\partial^2 w(x,t)}{\partial x^2} + \kappa\frac{\partial^4 w(x,t)}{\partial x^4} = \xi(x,t)$$

- Periodic boundary conditions $w(x,t) = w(x+2\pi,t)$
- $\xi(x,t)$ is uncorrelated and white (in time) dynamical noise

Periodic motion: $\boldsymbol{\kappa = 0.079}$



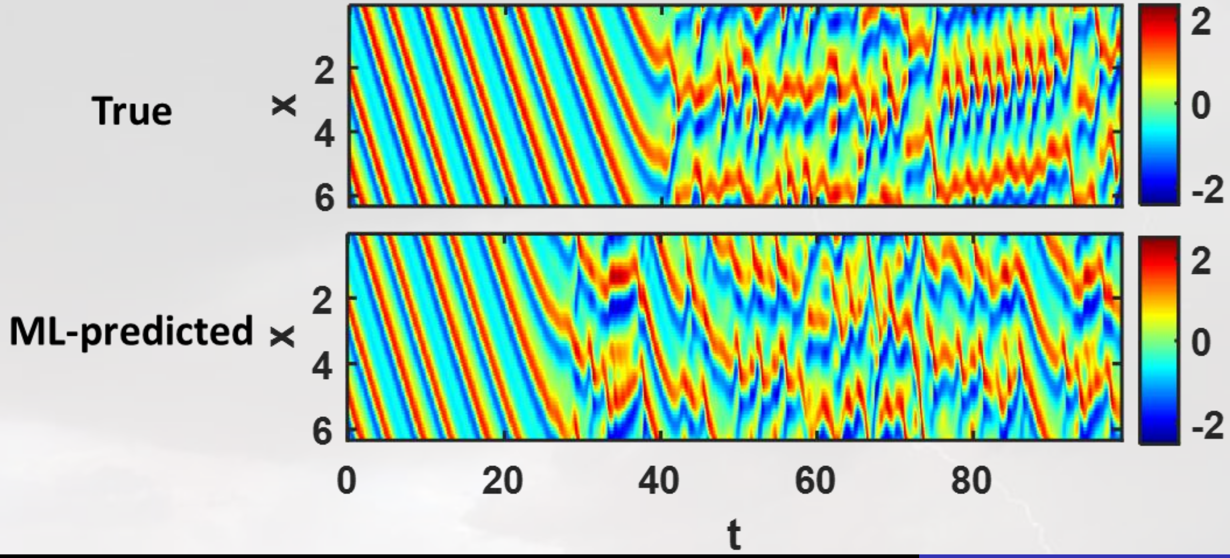
Chaotic motion: $\boldsymbol{\kappa = 0.081}$



Non-Stationary Kuramoto-Sivashinsky equation

System parameter time-dependence: $$\kappa(t) = \kappa_0 + \kappa_1 \exp\left(\frac{t}{\tau}\right)$$

Training data: noisy periodic orbit from t = -60 to t = 0.



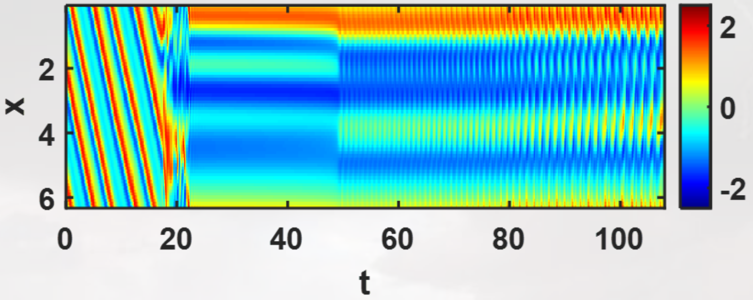
Prediction begins before tipping point and the ML is able to:

- ✓ **Predict a tipping point transition**
- ✓ **Predict the post-tipping point system dynamics**

Stability of ML prediction

- Long-term stability of ML forecasts is a well-known problem, and one method we employ to address this problem is noise training.
- In the previous example, we found that we did not need to do noise training to obtain stable predictions if the hyperparameters were carefully chosen. However, as the size of the system gets larger, hyperparameter tuning becomes increasingly more computationally expensive. In such cases, **noise training may help by making the ML stability less sensitive to the choice of hyperparameters**.

Forecast from an ML model with non-optimal hyperparameters and trained without noise training



Forecast from an ML model with non-optimal hyperparameters but trained with noise training