Machine Learning for Stochastic Parameterization: Generative Adversarial Networks(GAN) in the Lorenz '96 Model

Presenter: Wenlin Li

Outline

1. Background and Introduction
2. Methodology/Model (Lorenz 96 System, GANs)
3. Results
4. Discussion
5. Conclusions

Background and Introduction

- Background:
  - Weather and climate models face uncertainties, largely due to unresolved subgrid processes.
  - Traditional deterministic parameterization methods can't fully eliminated these errors.
- Introduction of Stochastic Methods:
  - Initially proposed for the European Center for Medium-Range Weather Forecasts, they have been shown to enhance forecast quality.
- Application of Machine Learning:
  - GANs (Generative Adversarial Networks) offer a potential solution for parameterizing complex subgrid processes.
  - The study aims to evaluate the efficacy of GANs in parameterizing across weather and climate time scales.

Methodology/Model

1. Lorenz ’96 System

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b} \sum_{j=J(k-1)+1}^{kJ} Y_j; \; k = 1, \ldots, K \tag{1a}$$

$$\frac{dY_j}{dt} = -cbY_{j+1}(Y_{j+2} - Y_{j-1}) - cY_j + \frac{hc}{b} X_{(\lfloor (j-1)/J \rfloor + 1)}; \; j = 1, \ldots, JK, \tag{1b}$$

In the present paper, the number of X variables is **K = 8** and the number of Y variables per X variable is **J = 32**. Further, we set the coupling constant to **h = 1**, the **spatial-scale ratio to b = 10**, and the **temporal-scale ratio** to **c = 10**. The forcing term **F = 20** is set large enough to ensure chaotic behavior.

Due to limited computational resources, it is not possible to explicitly simulate the **smallest scales**, which are instead **parameterized.** Motivated by this requirement for weather and climate prediction, a forecast model for the L96 system is constructed by truncating the model equations and parameterizing the impact of the small Y scales on the resolved X scales:

$$\frac{dX_k^*}{dt} = -X_{k-1}^*(X_{k-2}^* - X_{k+1}^*) - X_k^* + F - \hat{U}(X_k^*, t); \; k = 1, \ldots, K, \tag{2}$$

where $X_k^*(t)$ is the forecast estimate of $X_k(t)$ and $\hat{U}(X_k^*, t)$ is the parameterized subgrid tendency. The parameterization $\hat{U}$ approximates the true subgrid tendencies:

$$U(X, Y) = \frac{hc}{b} \sum_{j=J(k-1)+1}^{kJ} Y_j, \tag{3}$$

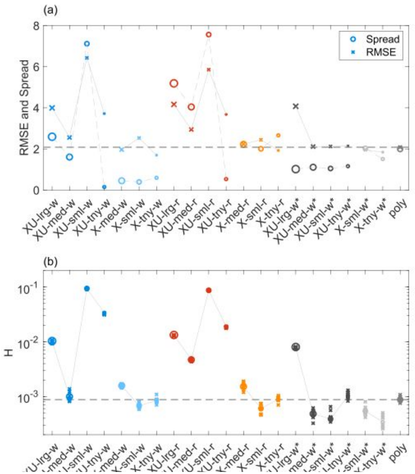
which can be estimated from realizations of the "truth" time series as

$$U_k(t) = [-X_{k-1}(t)\left(X_{k-2}(t) - X_{k+1}(t)\right) - X_k(t) + F] - \left(\frac{X_k(t + dt_f) - X_k(t)}{dt_f}\right), \tag{4}$$

They used a **fourth-order Runge-Kutta (RK4)** time-stepping scheme with a time step of 0.001 MTU (Model Time Unit) and ran the simulation for 20000 MTU. The output from the **first 2000 MTU** was used as **training data**, and the **remaining 18000 MTU** was used as **testing data**.All parameterized L96 prediction models used a prediction time step of **0.005 MTU** and a **second-order Runge-Kutta (RK2)** time-stepping scheme.

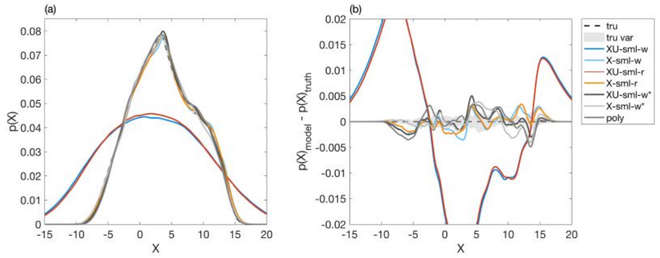
4. Weather Evaluation

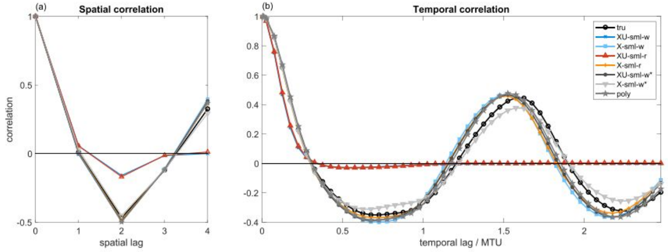
The parameterized model of the Lorenz '96 system was assessed within the weather forecasting framework. The predictive accuracy of all weather experiments was evaluated using **root mean square error (RMSE)** and the measure of spread. Among them, the **X-tny-r, X-tny-w, and X-tny-w* (no output layer noise) models exhibited the best performance in terms of RMSE**.



5. Climate Evaluation

By evaluating their reproduction of the probability density function(PDF) of the X variables and capturing spatiotemporal behavior. Through various analyses, **some GANs successfully replicated the true system's characteristics**, **equaling or surpassing polynomial parameterization methods**, while others performed poorly.





2. GANs

- **Purpose**: Use Generative Adversarial Networks (GANs) to predict the subgrid tendency based on current and previous states.
- **Structure**:
  - The GAN generator accepts Xt−1,k, Ut−1,k, and a latent Gaussian random vector Zt−1,k as input to estimate Ût,k. The discriminator accepts Xt−1,k, Ut−1,k, and Vt,k as inputs (where Vt,k may be either Ut,k if from the training data or Ût,k if from the generator) and outputs the probability that Vt,k comes from the training data.
  - Both components of the GAN have two hidden layers with 16 neurons each and use SELU activation function and L2 regularization.
  - The generator tries to produce data that looks like the real data, while the discriminator tries to discriminate if data is real or generated.
- **Noise Injection**: The study tests both white (uncorrelated) noise and red (correlated) noise as inputs to the GAN. White noise is sampled from a standard normal distribution with a mean of 0 and standard deviation of 1, and red noise is a temporally correlated time series, and its generation equation:

$$\phi_g = \frac{\mathbb{E}[(U_t - \hat{U}_t^d)(U_{t-1} - \hat{U}_{t-1}^d)]}{\sigma_{\hat{U}}^2}; \sigma_g = (1 - \phi_g^2)^{1/2} \quad (7a)$$

$$\epsilon_g \sim \mathcal{N}(0, \sigma_g) \quad (7b)$$

$$Z_{r,t,k} = \phi_g Z_{t-1,k} + \epsilon_g \quad (7c)$$

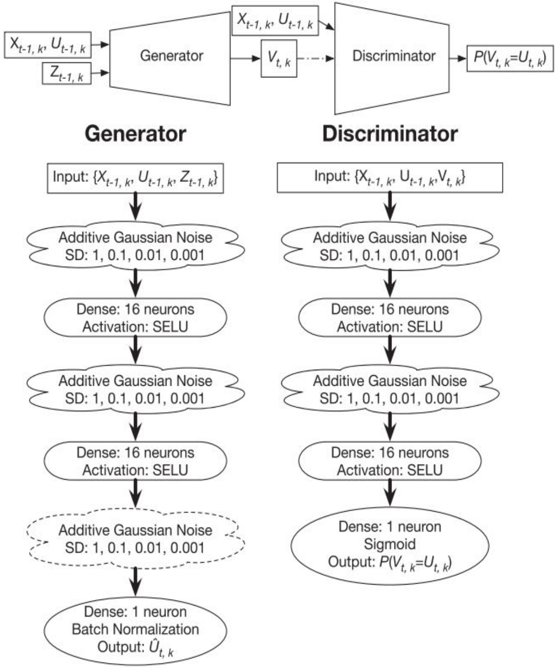


Table 1
Summary of the GAN Configurations Tested

Short name	Input variables	Noise magnitude	Noise correlation	Output layer noise?
XU-lrg-w	$X_{t-1,k}, U_{t-1,k}$	1	white	yes
XU-med-w	$X_{t-1,k}, U_{t-1,k}$	0.1	white	yes
XU-sml-w	$X_{t-1,k}, U_{t-1,k}$	0.01	white	yes
XU-tny-w	$X_{t-1,k}, U_{t-1,k}$	0.001	white	yes
X-med-w	$X_{t-1,k}$	0.1	white	yes
X-sml-w	$X_{t-1,k}$	0.01	white	yes
X-tny-w	$X_{t-1,k}$	0.001	white	yes
XU-lrg-r	$X_{t-1,k}, U_{t-1,k}$	1	red	yes
XU-med-r	$X_{t-1,k}, U_{t-1,k}$	0.1	red	yes
XU-sml-r	$X_{t-1,k}, U_{t-1,k}$	0.01	red	yes
XU-tny-r	$X_{t-1,k}, U_{t-1,k}$	0.001	red	yes
X-med-w	$X_{t-1,k}$	0.1	red	yes
X-sml-r	$X_{t-1,k}$	0.01	red	yes
X-tny-r	$X_{t-1,k}$	0.001	red	yes
XU-lrg-w*	$X_{t-1,k}, U_{t-1,k}$	1	white	no
XU-med-w*	$X_{t-1,k}, U_{t-1,k}$	0.1	white	no
XU-sml-w*	$X_{t-1,k}, U_{t-1,k}$	0.01	white	no
XU-tny-w*	$X_{t-1,k}, U_{t-1,k}$	0.001	white	no
X-sml-w*	$X_{t-1,k}$	0.01	white	no
X-tny-w*	$X_{t-1,k}$	0.001	white	no

- **Training Details**: A batch B, or subset of samples drawn randomly without replacement from the training data, of truth run output is split in half. **One subset** is fed through the **generator G** and then into the **discriminator D,** and **the other** is sent directly to the **discriminator**. The **discriminator weights** are then **updated** based on the following loss function L_d

$$L_d = \mathbb{E}_B[\log(D(X_{t-1,b}, U_{t-1,b}, U_{t,b})] + \mathbb{E}_B[\log(1 - D(G(X_{t-1,b}, U_{t-1,b}, Z_{t-1,b}))]. \quad (5)$$

E_B is the expected value over a single batch of data. Another batch of samples are drawn and sent through the generator and then the discriminator with frozen weights. The **generator loss** L_g is calculated as

$$L_g = \mathbb{E}_B[\log(D(G(X_{t-1,b}, U_{t-1,b}, Z_{t-1,b}))] \quad (6)$$

The GANs are all trained with a consistent set of optimization parameters. The GANs are updated through **stochastic gradient descent** with a batch size (number of examples randomly drawn without replacement from the training data) of **1,024** and a **learning rate of 0.0001** with the **Adam optimizer** (Kingma & Ba, 2015). The GANs are **trained** on **6.4 million samples** and are **validated** on **29 million samples** from different portions of the truth run. The GANs are trained for **30 epochs**, or passes through the training data. The model **weights** are **saved** for analysis **every epoch for the first 20 epochs** and then **every 2 epochs between epochs 20 and 30**. The GANs are developed with the Keras v2.2 machine learning API coupled with Tensorflow v1.13.

3. Polynomial Regression Parameterization(similar with the PLR in last paper)

- **Purpose:** In practical modeling scenarios, for reasons like **computational resource constraints**, one cannot directly utilize this "truth" model. Therefore, there's a need to **parameterize** certain sub-processes within the model, which often involves simplifying or approximating certain aspects of the model. Polynomial Regression Parameterization is a **stringent benchmark** against which to test GAN parameterization.
- **Structure:**

$$\frac{dX_k^*}{dt} = -X_{k-1}^*(X_{k-2}^* - X_{k+1}^*) - X_k^* + F - \hat{U}(X_k^*, t);\ k = 1, \ldots, K, \quad (2)$$

$$\begin{aligned} \hat{U}_{t,k} &= U_{t,k}^d + \epsilon_{t,k} \\ U_{t,k}^d &= aX_{t-1,k}^3 + bX_{t-1,k}^2 + cX_{t-1,k} + d \end{aligned} \quad (8)$$

noise term:

$$\epsilon_{t,k} = \phi\epsilon_{t-1,k} + \sigma_\epsilon(1-\phi^2)^{1/2} z_{t,k}, \quad (9)$$

where $z \sim \mathcal{N}(0,1)$, the first-order autoregressive parameters (ϕ, σ_ϵ) are fit from the residual time series $r_t = U_t - U_t^d$, and the ϵ_k processes are independent for different X variables.

The polynomial parameterization has been specifically designed to represent the impact of the X variables in this version of the L96 model. This polynomial regression parameterization model consists of two components: **a deterministic part, U_d t,k**, which is **a cubic polynomial of X_t-1,k**; and **a stochastic noise term, ε_t,k**.
The parameters[a, b, c, d] are determined by **a least squares** fit to the (X, U) data from the L96 “truth” training run.

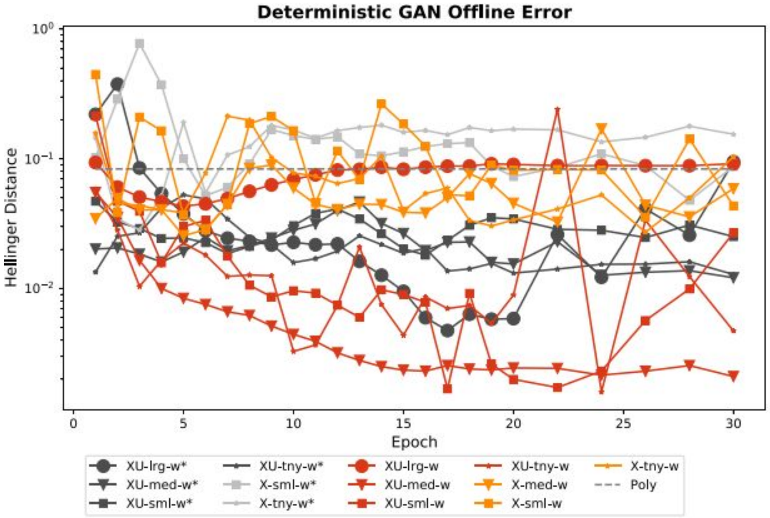
Results

1. Metrics

- The accuracy of ensemble weather forecasts can be assessed by calculating the **Root Mean Square Error (RMSE)** of the ensemble mean. A lower RMSE indicates higher **prediction accuracy**.
- The "climate" of the L96 system can be simply defined as the **probability density function (PDF) of individual Xt,k values.** Therefore, climate skill can be characterized by **quantifying** the **differences** between the **real PDF** and the **predicted PDF**. The **Hellinger distance** is used to **measure** the **differences** of **each forecasting model**.

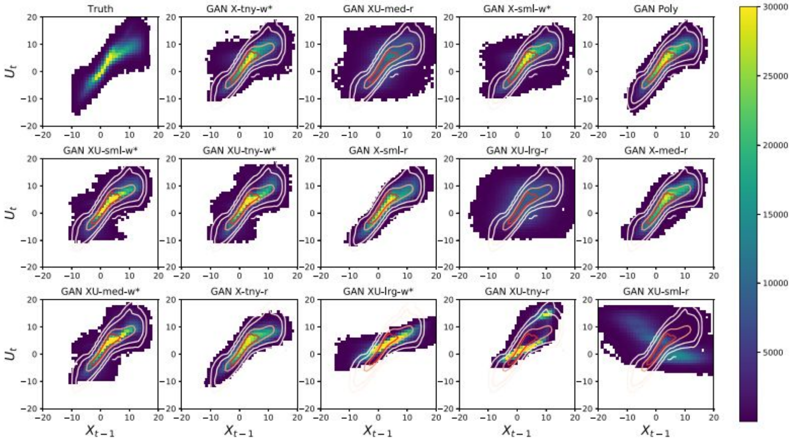
2.Off-line Assessment of GAN Performance

It was observed that in the initial stages, most GANs showed a **gradual decrease** in the **Hellinger distance**, followed by a relatively stable period. GANs that incorporated inputs **Xt−1,k and Ut−1,k** typically **performed better** in offline analysis compared to GANs that **only used Xt−1,k** as input.



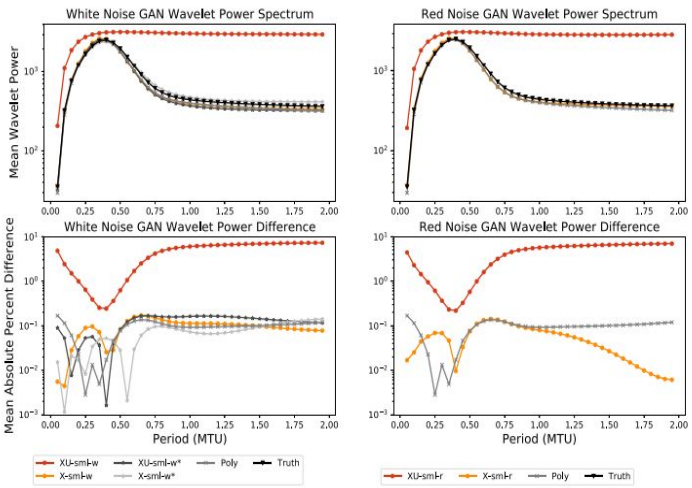
3.GAN Simulation of Subgrid-Scale Tendency Distribution

By comparing the joint distribution of Xt-1 and Ut from model simulations, they investigated how the **noise standard deviation affects the climate model**. Furthermore, They compared the **capabilities** of various GANs and polynomial models in **capturing the shape of the real distribution**.



6.Wavelet Analysis

- Energy Evaluation:
  - A continuous wavelet transform using the Ricker wavelet was used to decompose time series to analyze contributions from different periods.
  - **All sml GANs, except for the XU-sml-w and XU-sml-r , closely followed the true power curve.**
- Evaluation of Wavelet Differences:
  - A clearer assessment of the wavelet differences was obtained by computing the Mean Absolute Percentage Difference (MAPD) from the truth run at different wavelengths.
  - The evaluation of wavelet differences was achieved by computing the **Mean Absolute Percentage Difference (MAPD) at various periods.** The results revealed that while **no GAN** model **performed best** across all periods, **the X-sml-r GAN** exhibited **lower MAPD in the increased periods**.



$$E = \frac{1}{T}\sum_{t=1}^{T} w_t^2 \tag{13}$$

$$MAPD = \frac{1}{T}\sum_{t=1}^{T} \frac{|E_{g,t} - E_{u,t}|}{E_{u,t}}. \tag{14}$$

Discussion

This study primarily **focuses** on the **application of GANs** in **stochastic parameterization**, as GANs offer **a framework** to directly **embed randomness** in the model and training process, rather than introducing **stochasticity after deterministic parameterization**. **However**, the relative simplicity of the L96 system may lead to more complex **GANs overfitting** the data compared to simple polynomial parameterization.

Conclusions

In this study, the authors developed a Generative Adversarial Networks (GANs) framework for the Lorenz '96 dynamic system to parameterize subgrid-scale processes. **Some GANs outperformed** the **benchmark models(Polynomial Regression Parameterization)** in forecast accuracy. **In particular**, **some GANs with red noise** produce **reliable weather forecasts**, in which the ensemble spread is a good indicator of the error in the ensemble mean. **The noise** is **most critical** for producing **reliable forecasts**. Although GANs are very **sensitive** to **noise** magnitude and **other hyperparameter settings**, they show **potential** for using in stochastically parameterized physical processes in more **complex weather and climate models**.