B.Tech Project 2 Report

Generative Adversarial Networks (GANs) to Simulate Dynamics of Weather and Climate

by

Jeevankur Talukdar

(17ME10029)

Under the supervision of

Dr. Adway Mitra

Assistant Professor, Centre of Excellence in Artificial Intelligence (AI)

Indian Institute of Technology Kharagpur

April 19, 2021

Abstract

Clustering and recognizing patterns is one of the major applications of machine learning recently. Recognizing patterns in naturally occurring events helps us to predict future events more accurately. One obvious example is weather and climate prediction. It has always been difficult to accurately predict the weather and climate at a future point in time. However recent research has shown that with modern machine learning techniques we are able to achieve better predictions. Generative Adversarial Networks (GANs for short) have had a huge success since they were introduced in 2014 by Ian J. Goodfellow and co-authors in the article Generative Adversarial Nets. They are a class of machine learning framework where two neural networks contest with each other to produce better results. These networks have a vast range of applications. In this project we develop and study the application of various GANs to simulate the dynamics of Weather and Climate. Weather has fast dynamics, and climate has slow dynamics. The interactions between a fast and a slow dynamics is modeled well by the Lorenz96 system, which is a common baseline model for evaluating both parameterization and data assimilation techniques.

Simulations of the atmosphere must approximate the effects of small-scale processes with simplified functions called parameterizations. Standard parameterizations only predict one output for a given input, but stochastic parameterizations can sample from all the possible outcomes that can occur under certain conditions. We have developed and evaluated a machine learning stochastic parameterization, which builds a mapping between large-scale current conditions and the range of small-scale outcomes from data about

both. We test the machine learning stochastic parameterization in a simplified mathematical simulation that produces multi-scale chaotic waves like the atmosphere. We find that some configurations of the machine learning stochastic parameterization perform slightly better than a simpler baseline stochastic parameterization over both weather and climate like time spans.

Keywords: Machine Learning, Stochastic, Parameterization

Contents

List of Figures

1. Introduction

1.1 Brief

The activities of many primary sectors depend on the weather for production, e.g. farming. The climate is changing at a drastic rate nowadays, which makes the old weather prediction methods less effective and more hectic. To overcome these difficulties, the improved and reliable weather prediction methods are required. These predictions affect a nation's economy and the lives of people.

A large source of weather and climate model uncertainty is the approximate representation of unresolved sub-grid processes through parameterization schemes. Traditional, deterministic parameterization schemes represent the mean or most likely sub-grid scale forcing for a given resolved-scale state. Machine learning models offer an approach to parameterize complex nonlinear sub-grid processes in a potentially computationally efficient manner from data describing those processes.

1.2 Literature Review

Irreducible uncertainties in weather forecasts result from a lack of scale separation between resolved and unresolved processes. Uncertainty also arises because the chaotic nature of

the atmosphere gives rise to sensitivity to uncertain initial conditions. Practically, uncertainty is represented in forecasts using ensembles of integrations of comprehensive weather and climate prediction models, first suggested by Leith (1975). To produce reliable probabilistic forecasts, the generation of the ensemble must include a representation of both model and initial condition uncertainty.

Initial condition uncertainty is addressed by perturbing the initial conditions of ensemble members, for example by selecting directions of optimal perturbation growth using singular vectors (Buizza Palmer, 1995), or by characterizing initial condition uncertainty during the data assimilation cycle (Isaksen et al., 2010). One approach for representing irreducible model uncertainty is stochastic parameterization of unresolved physical processes. A more detailed motivation for including stochastic parameterizations in weather and climate models is presented in Palmer (2012).

Stochastic approaches for numerical weather prediction (NWP) were originally proposed for use in the European Center for Medium-Range Weather Forecasts (ECMWF) ensemble prediction system (Palmer et al., 1997; Buizza et al., 1999).

Recent work has assessed the impact of stochastic parameterization schemes in both idealized and state-of-the-art climate models for long term integration (P. D. Williams, 2012; Ajayamohan et al., 2013; Juricke Jung, 2014; Dawson Palmer, 2015; Wang et al., 2016; H. M. Christensen et al., 2017; Davini et al., 2017; Strømmen et al., 2018). These studies demonstrate that including a stochastic representation of model uncertainty can go beyond improving initialized forecast reliability, and can also lead to improvements in the model mean state (Palmer, 2001; Berner et al., 2012), climate variability (Ajayamohan et al., 2013; Dawson Palmer, 2015; H. M. Christensen et al., 2017), and change a model's climate sensitivity (Seiffert von Storch, 2010).

Recent years have seen substantial interest in the development of stochastic parameterization schemes. Pragmatic approaches, such as the Stochastically Perturbed Parameterization Tendencies (SPPT) scheme (Buizza et al., 1999; Palmer, Buizza, et al., 2009) are widely used due to their ease of implementation and beneficial impacts on the model (Sanchez et al., 2016; Leutbecher et al., 2017; H. M. Christensen et al., 2017). Other schemes predict the statistics of model uncertainty using a theoretical understanding of the atmospheric processes involved, such as the statistics of convection (Craig Cohen, 2006; Khouider et al., 2010; Sakradzija Klocke, 2018; Bengtsson et al., 2019). A third approach is to make use of observations or highresolution simulations to characterize variability that is unresolved in a low-resolution forecast model (Shutts Palmer, 2007).

1.3 Scope

Machine learning models offer an approach to parameterize complex nonlinear sub-grid processes in a potentially computationally efficient manner from data describing those processes. The family of machine learning models consist of mathematical models whose structure and parameters (often denoted weights) optimize the predictive performance of a priori unknown relationships between input ("predictor") and output ("predictand") variables. One active area in current machine learning research is generative modeling, which focuses on models that create synthetic representative samples from a distribution of arbitrary complexity without the need for a parametric representation of the distribution. Generative adversarial networks, or GANs (Goodfellow et al., 2014), are a class of generative models that consist of two neural networks in mutual competition. Because the stochastic parameterization problem can be framed as sampling from the distribution of sub-grid tendencies conditioned on the resolved state, conditional GANs have the potential to perform well on this task.

1.4 Objective

The purpose of this study is to evaluate how well GANs can parameterize the subgrid tendency component of an atmospheric model at weather and climate timescales. A key question is whether a GAN can learn uncertainty quantification within the parameterization framework, removing the need to retrospectively develop separate stochastic representations of model uncertainty. Simple chaotic dynamical systems such as L96 are useful for testing methods in atmospheric modeling due to their transparency and computational cheapness. The L96 system has been widely used as a testbed in studies including development of stochastic parameterization schemes (Wilks, 2005; Crommelin Vanden-Eijnden, 2008; Kwasniok, 2012; Arnold et al., 2013), data assimilation methodology (Fertig et al., 2007; Law et al., 2016; Hatfield et al., 2018), as well as using ML approaches to learn improved deterministic parameterization schemes (Schneider et al., 2017; Dueben Bauer, 2018; Watson, 2019).

2. Methods

2.1 Lorenz'96 Model

The Lorenz 96 model is a dynamical system formulated by Edward Lorenz in 1996. It was designed as a 'toy model' of the extratropical atmosphere, with simplified representations of advective nonlinearities and multi-scale interactions (Lorenz, 1996). It consists of two scales of variables arranged around a latitude circle. The large scale, low-frequency X variables are coupled to a larger number of small scale high-frequency Y variables, with a two-way interaction between the Xs and Y s. It is the interaction between variables of different scales that makes the L96 system ideal for evaluating new ideas in parameterization development.

The X and Y variables evolve following:

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b}\sum_{j=J(k-1)+1}^{k} JY_j; \quad k = 1, ..., K \tag{2.1}$$

$$\frac{dY_j}{dt} = -cbY_{j+1}(Y_{j+2} - Y_{j-1}) - cY_j + \frac{hc}{b}X_{(int[(j-1)/J]+1)}; \quad j = 1, ..., JK \tag{2.2}$$

where h is the coupling constant, b is spatial-scale ratio, c is temporal-scale ratio and F is the forcing term.

In this study the full Lorenz '96 equations are treated as the 'truth' which must be forecast or simulated. In the case of the atmosphere, the physical equations of motion of the system are known. However, due to limited computational resources, it is not possible to explicitly simulate the smallest scales, which are instead parameterized.

The parameterization approximates the true sub-grid tendencies.

$$U(X,Y) = \frac{hc}{b} \sum_{j=J(k-1)+1}^{k} JY_j, \tag{2.3}$$

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

$$U_k(t) = [-X_{k-1}(t)(X_{k-2}(t) - X_{k+1}(t)) - X_k(t) + F] - \frac{X_k(t+dt_f) - X_k(t)}{dt_f} \tag{2.4}$$

The time step dt_f equals the time step used in the forecast model for consistency.

Implementation For our implementation we use the number of X variables, K = 8 and the number of Y variables per X variable, J = 32. Further, we set the coupling constant, h = 1, the spatial-scale ratio, b = 10 and the temporal-scale ratio c = 10. The forcing term F = 20 is set large enough to ensure chaotic behavior. The chosen parameter settings, which were used in (Arnold et al., 2013), are such that one model time unit (MTU) is approximately equivalent to five atmospheric days, deduced by comparing error doubling times in L96 and the atmosphere (Lorenz, 1996).

A long "truth" run of the L96 model is performed to generate both training data for the machine learning models and a test period for both weather and climate evaluations. The "truth" run is integrated for 20 MTU (20,000 samples) using a linear timestepping scheme and a time step dt = 0.001 MTU. Output from the first 2 MTU (2000 samples) are used for training, and the remaining 16 MTU (32000 samples)are used for testing.

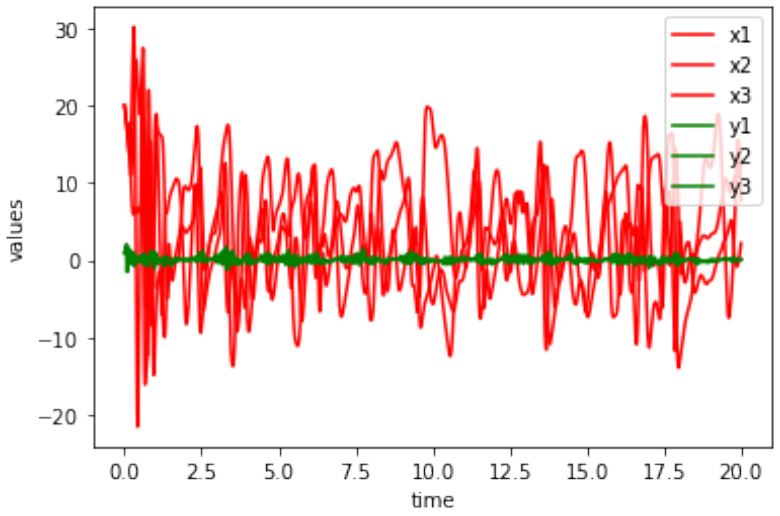

FIGURE 2.1: Plot - first few variables from L96

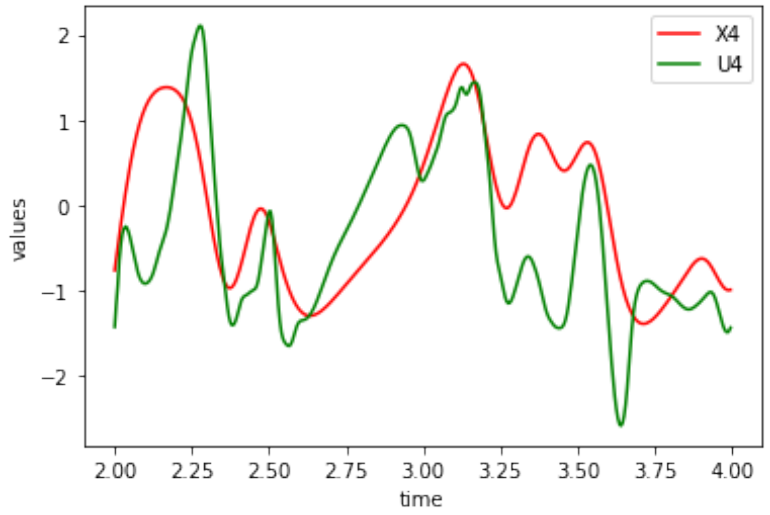

FIGURE 2.2: Plot - True subgrid tendencies of X_4(*Standardized*)

A burn-in period of 2 MTU is discarded. All parameterized forecast models of the L96 use a forecast timestep of dtf = 0.005 MTU .

The true sub-grid tendencies $U_k(t)$ are approximated for all $X_k, k = 1, .., K$.

2.2 GAN Parameterizations

The GAN parameterization developed for the Lorenz '96 model in this study utilizes a conditional dense GAN to predict the sub-grid tendency at the current time step given information about the state at the previous time step.

The GAN generator accepts $X_{t1,k}$, $U_{t1,k}$, and a latent random vector $Z_{t1,k}$ as input to estimate $\hat{U}_t, k$, or the predicted U at time t. The discriminator accepts $X_{t1,k}, U_{t1,k}$, and $V_{t,k}$ as inputs (where $V_{t,k}$ may be either $U_{t,k}$ if from the training data or $\hat{U}_{t,k}$ if from the generator) and outputs the probability that $V_{t,k}$ comes from the training data. All inputs and outputs are re-scaled to have a mean of 0 and standard deviation of 1 based on the training data distributions.

$$Z = \frac{X - mean}{StandardDeviation} \tag{2.5}$$

Implementation Each GAN we consider consists of the same neural network architecture with variations in the inputs and how noise is scaled and inserted into the network. Both the generator and discriminator networks contain two hidden layers with 16 neurons in each layer. Scaled exponential linear unit (SELU) activation functions (Klambauer et al., 2017) follow each hidden layer. SELU is a variation of the common Rectified Linear Unit (ReLU) activation function with a scaled exponential transform for the negative values that helps ensure the output distribution retains a mean of 0 and standard deviation of 1. A batch normalization (Ioffe Szegedy, 2015) output layer ensures that the output values retain a mean of 0 and and standard deviation of 1, which helps the generator converge to the true distribution faster.

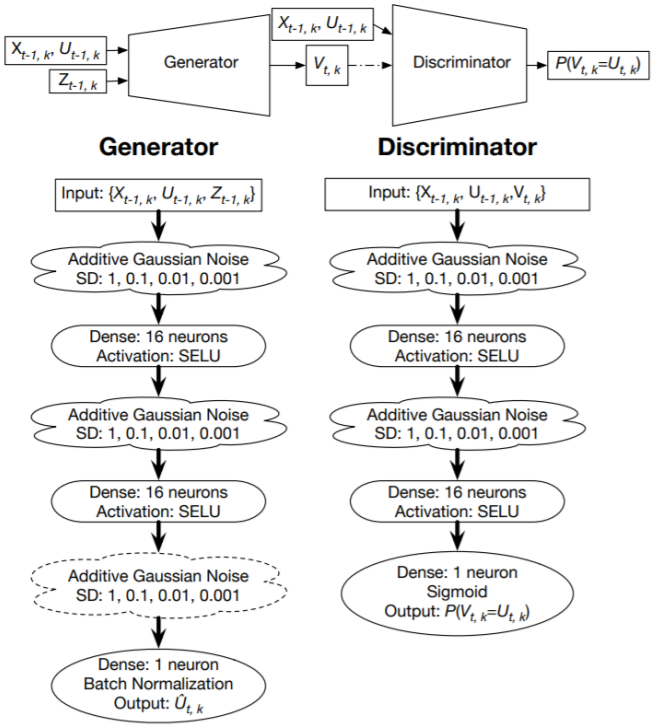

FIGURE 2.3: (Top) A diagram of how the GAN networks are connected for training. (Bottom) A diagram of the GAN network architectures used for the stochastic parameterization

The GAN training procedure iteratively updates the discriminator and generator networks until the networks reach an adversarial equilibrium in which the discriminator should not be able to distinguish "true" data from generator samples.

The networks a trained for various values of noise and input combinations.

In forecast mode, we test providing both white, or uncorrelated noise, and red, or correlated noise to the GAN. The red noise is generated using an AR(1) process with a correlation equal to the lag-1 autocorrelation of the deterministic residuals of the GAN. An autoregressive model is when a value from a time series is regressed on previous values from that same time series. For example in an AR(1) auto regression with lag-1, y_t and y_{t-1} satisfy:

TABLE 2.1: GAN Configurations

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
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	$X_{t-1,k}, U_{t-1,k}$	0.1	white	yes
X-sml-w	$X_{t-1,k}, U_{t-1,k}$	0.01	white	yes
X-tny-w	$X_{t-1,k}, U_{t-1,k}$	0.001	white	yes
X-med-r	$X_{t-1,k}, U_{t-1,k}$	0.1	red	yes
X-sml-r	$X_{t-1,k}, U_{t-1,k}$	0.01	red	yes
X-tny-r	$X_{t-1,k}, U_{t-1,k}$	0.001	red	yes

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t \tag{2.6}$$

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 1024 and a learning rate of 0.0001 with the Adam optimizer (Kingma Ba, 2015). The GANs are trained for 30 epochs, or passes through the training data.

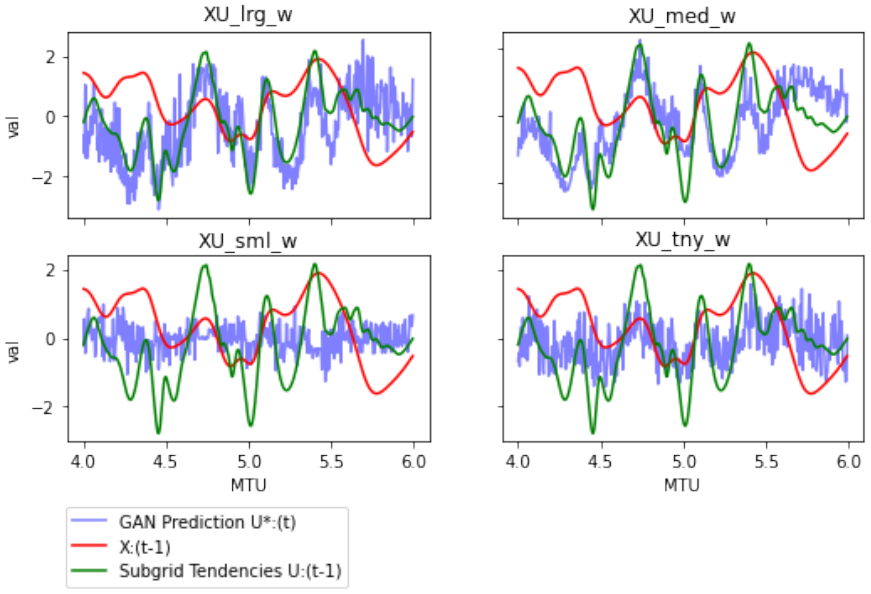

FIGURE 2.4: Plot - Forecast from GANs trained with X_{t-1} and U_{t-1} for white input noise

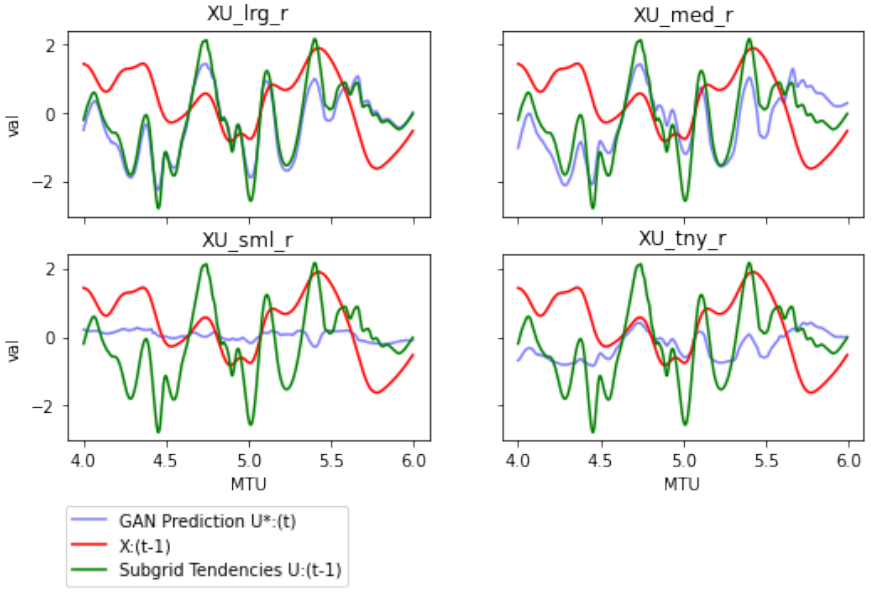

FIGURE 2.5: Plot - Forecast from GANs trained with X_{t-1} and U_{t-1} for red input noise

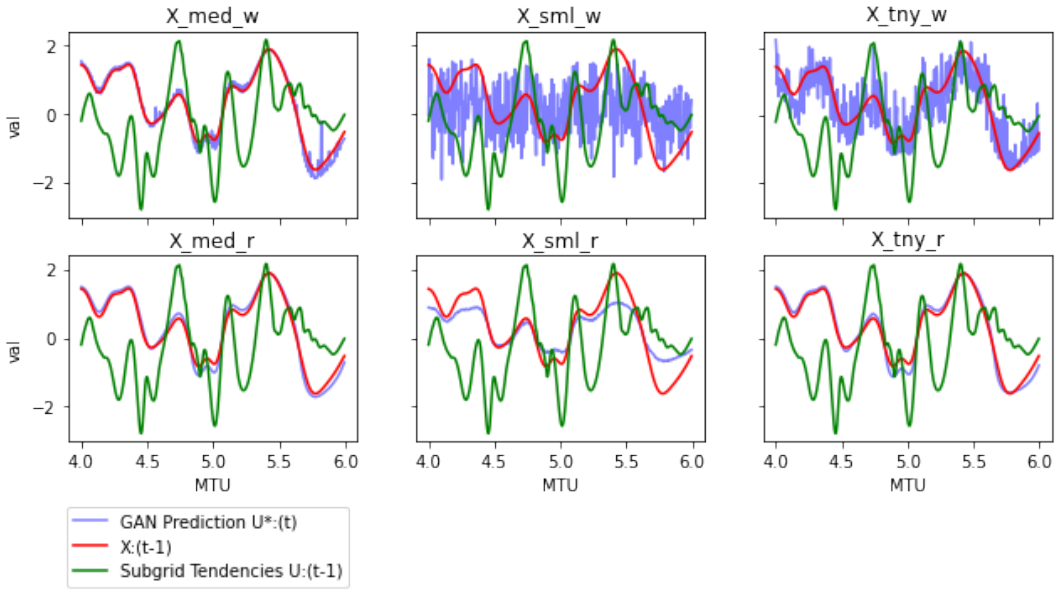

FIGURE 2.6: Plot - Forecast from GANs trained with X_{t-1} for white and red input noises

3. Results

The 14 sets of generated forecast from the 7 different neural networks are evaluated through various metrices.

3.1 Hellinger distance H

The simplest definition of the 'climate' of the L96 system is the probability density function (PDF) of the individual Xt,k values. The climatological skill can therefore be summarized by quantifying the difference between the true and forecast PDF. The Hellinger distance H, is calculated for each forecast model:

$$H(p,q) = \frac{1}{2}\int (\sqrt{p(x)} - \sqrt{q(x)})^2 dx \tag{3.1}$$

where p(x) is the forecast PDF, and q(x) is the verification PDF (Pollard, 2002). The smaller H, the closer the forecast pdf is to the true pdf. We also considered the Kullback-Leibler (KL) divergence (Kullback Leibler, 1951), motivated by information theory, but found it provided no additional information over the Hellinger distance, so results for the KL are not shown for brevity.

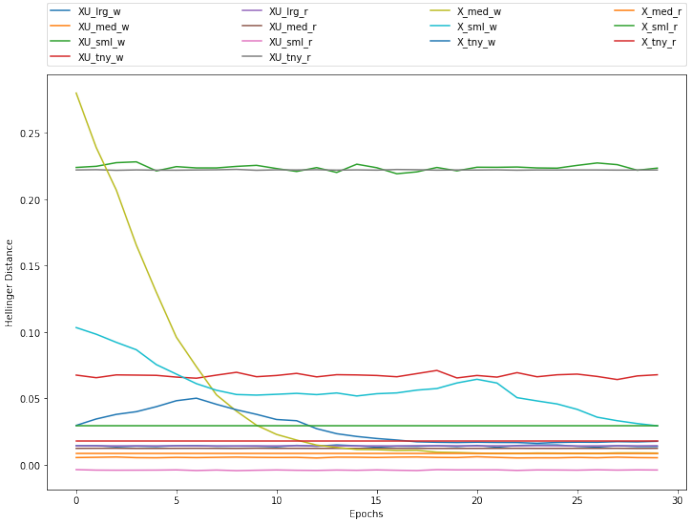

FIGURE 3.1: Deterministic GAN Offline Error for all X

3.2 Offline assessment of GAN performance

The GAN parameterizations are first evaluated on how closely their output subgrid forcing distributions match those of the truth run when the GANs are supplied with input X and U values from the truth run. This is summarized by the Hellinger distance in Figure 3.1. Most of the GANs show a trend of decreasing Hellinger distance for the first few epochs followed by mostly stable oscillations. GANs with both Xt1,k and Ut1,k as input tend to perform better in the offline analysis than those with only Xt1,k. Larger input noise standard deviations seem to reduce the amount of fluctuation in the Hellinger distance between epochs, but there does not appear to be a consistent correlation with noise standard deviation and Hellinger distance.

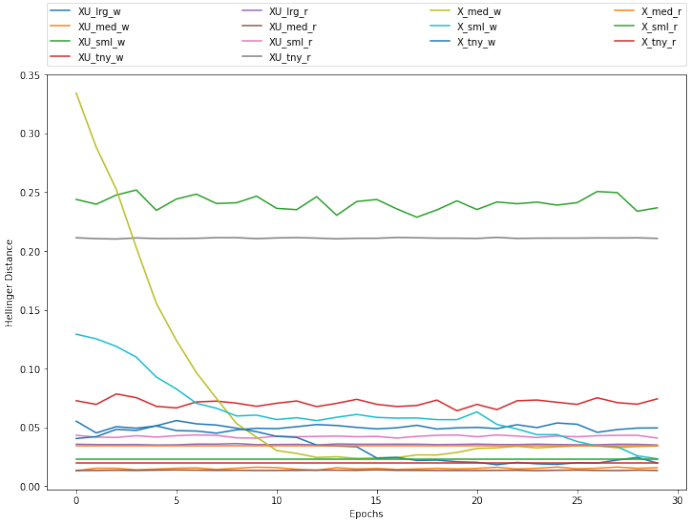

FIGURE 3.2: Deterministic GAN Offline Error for X_1

3.3 GAN simulation of sub-grid-scale tendency distribution

The joint distributions of X_{t1} and U_t from the different model runs reveal how the noise standard deviation affects the model climate (Fig. 3.3). Larger noise standard deviations increase the range of X values appearing in the run but do not appear to change the range of U values output by the GAN. The behavior of the XU-tny-r GAN devolved into oscillating between two extremes. The X-only GANs did the best job in capturing the shape of the truth distribution although they underestimated the variance at the extremes.

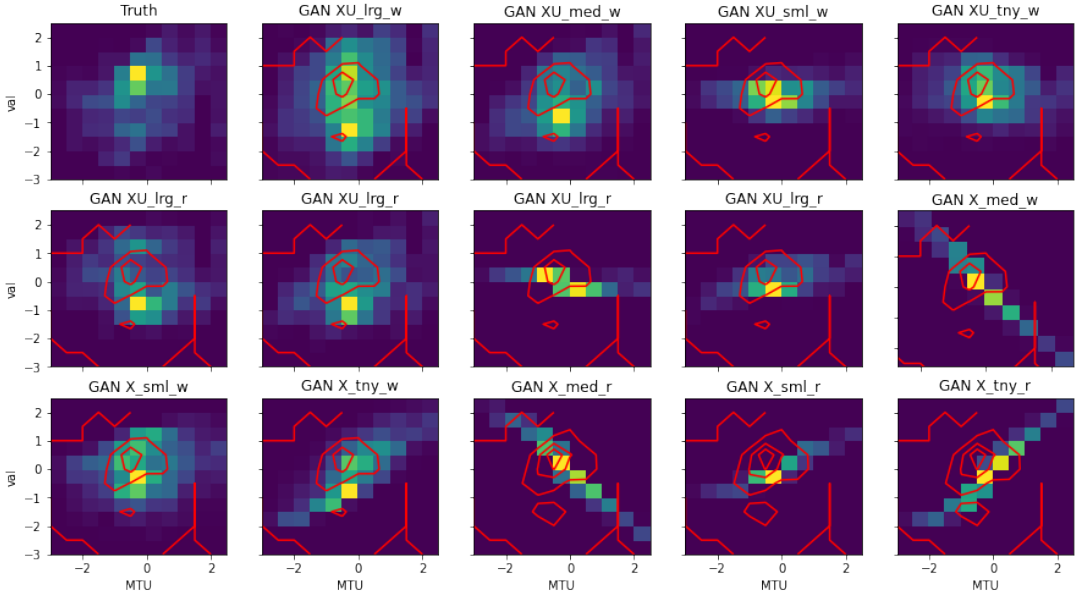

FIGURE 3.3: Joint distributions (2D histograms) of Xt1 and Ut for each GAN configuration. The truth joint distribution is overlaid in red contours on each forecast model distribution

4. Conclusion

Several of the GANs tested show a weather and climate skill that is competitive with a bespoke polynomial parameterization scheme. The good performance of the GAN is encouraging, demonstrating that GANs can indeed be used as explicit stochastic parameterizations of uncertain sub-grid processes directly from data.

The L96 system is commonly used as a testbed for new ideas in parameterization, and ideas tested using the system can be readily developed further for use in higher complexity Earth system models. However, the L96 system has many fewer dimensions than an Earth system model and a relatively simple target distribution. The relative simplicity of the L96 system may have also led to the more complex GAN overfitting to the data compared with the simpler polynomial parameterization. For more complex, higher dimensional systems, the extra representational capacity of the GAN may provide more benefit than can be realized in L96. The computational simplicity of L96 also allows for the production of extremely large training data sets with little compute resources. Higher complexity Earth system model output can pro-vide training set coverage spatially but will be limited temporally by the amount of computational resources available.

Future work will use these observations to develop machine-learned stochastic parameterization schemes for use in higher complexity Earth system models. GANs of a similar level of complexity to those used for L96 could emulate local effects, such as some warm rain

formation processes. Other generative neural network frameworks, such as variational autoencoders, should also be investigated to determine if they can provide similar or better performance with a less sensitive training process.

5. References

[0] Brenowitz, N., Bretherton, C. (2019). Spatially extended tests of a neural network parameterization trained by coarse-graining [https://arxiv.org/abs/1904.03327]

[1] Simulating weather regimes: impact of model resolution and stochastic parameterization [https://doi.org/10.1007/s00382-014-2238-x]

[2] The Lorenz' 96 Model [https://en.wikipedia.org/wiki/Lorenz96model]

[3] Noise Correlation Matrix [https://www.sciencedirect.com/topics/engineering/noise-correlation-matrix]

[4] Autocorrelation and Time Series Methods [https://online.stat.psu.edu/stat462/node/188/]

[5] Matplotlib [https://matplotlib.org/]

[6] Additive white Gaussian noise [https://en.wikipedia.org/wiki/Add..noise]

[7] Noise of GAN training [https://www.inference.vc/instance-noise-a-trick-for-stabilising-gan-training/]