# A Stochastic Neural-Network Parameterization for Coarse-grid Climate Models

Stewart Renehan

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2019

Reading Committee:

Chris Bretherton, Chair

Adrian Raftery

Program Authorized to Offer Degree:
Statistics

University of Washington

**Abstract**

A Stochastic Neural-Network Parameterization for Coarse-grid Climate Models

Stewart Renehan

Chair of the Supervisory Committee:
Professor Chris Bretherton
Atmospheric Sciences

Coarse-grid climate models require parameterizations to include the effect of unresolved sub-grid processes. Recently, machine learning approaches have shown promise in producing more accurate parameterizations than existing physical-based approaches. However, these machine learning approaches to parameterization are deterministic so they fail to capture the probability distribution of possible sub-grid processes, which can cause mean state bias in simulations. A stochastic parameterization that is built on top of a machine learning parameterization is introduced. The stochastic approach consists of a Markov chain model that switches between states that represent residual ranges of the machine learning parameterization and a regression model that produces a stochastic adjustments to the machine learning parameterization given a Markov state. The approach produces the correct probability distribution of outputs when predicting on a training a dataset obtained from coarsening a high resolution simulation over 160-km by 160-km grid cells, but fails to correct the mean state bias issues in a climate model simulation. However, the climate model used for the simulation has systematic issues that make it difficult to effectively evaluate parameterizations: a hyper-diffusion scheme wipes out stochastic effects, and it is optimized for high resolution, not coarse resolution, simulations. Further evaluation is necessary to determine whether the stochastic method improves a deterministic machine learning scheme in a full climate model simulation.

# TABLE OF CONTENTS

Page

List of Figures . . . . . iii

List of Tables . . . . . iv

Glossary . . . . . v

Chapter 1: Introduction . . . . . 1
1.1 Introduction to Climate Models . . . . . 1
1.2 Convective Parameterization . . . . . 2
1.3 Known Problems with Existing Parameterizations . . . . . 2
1.4 Machine Learning Parameterization . . . . . 2

Chapter 2: The Dataset . . . . . 4
2.1 Description of the Simulation . . . . . 4
2.2 Coarse Graining the High Resolution Data . . . . . 5
2.3 Climate Model Variables . . . . . 5

Chapter 3: Motivation for Stochastic Parameterization . . . . . 14
3.1 Brenowitz and Bretherton (2018) Approach to Parameterization . . . . . 14
3.2 Evaluation of Brenowitz and Bretherton (2018) Approach . . . . . 16
3.3 Problems with Deterministic Parameterization . . . . . 17
3.4 Goal of this Thesis . . . . . 24

Chapter 4: Methods . . . . . 25
4.1 Primary Stochastic Parameterization Approaches . . . . . 25
4.2 Markov Chain Stochastic Parameterization . . . . . 25
4.3 Mathematical Formulation and Assumptions . . . . . 27
4.4 Maximum Likelihood Formulation . . . . . 28

4.5 Markov States . . . . . 29
4.6 Transitioner Model $T$ . . . . . 30
4.7 Residual Model $h$ . . . . . 31
4.8 Overview of Stochastic Parameterization Method . . . . . 32
4.9 Qualitative Motivation for Approach . . . . . 34

Chapter 5: Results . . . . . 35
5.1 Evaluation Methods . . . . . 35
5.2 Model Fit Evaluation . . . . . 36
5.3 Stochastic Bin Transition Evaluation . . . . . 38
5.4 Coupled-Simulation Evaluation . . . . . 41
5.5 Other Problems with the Simulation . . . . . 43
5.6 Further Research . . . . . 44

Chapter 6: Conclusion . . . . . 46

# LIST OF FIGURES


| Figure Number | | Page |
|---|---|---|
| 2.1 | Near surface Zonal Winds (m/s) | 6 |
| 2.2 | Sensible Heat Flux | 7 |
| 2.3 | Latent Heat Flux | 8 |
| 2.4 | Solar TOA Insolation | 9 |
| 2.5 | Sea Surface Temperature | 10 |
| 2.6 | Total Non-Precipitating Water (QT) | 11 |
| 2.7 | Liquid/Ice Static Energy (SLI) | 11 |
| 2.8 | QT-Forcing / Moistening | 12 |
| 2.9 | SLI-Forcing / Heating (FSLI) | 12 |
| 3.1 | Neural Network Parameterization Inputs | 15 |
| 3.2 | Net Precip. Comparison | 18 |
| 3.3 | Net Heating Comparison | 19 |
| 3.4 | Log PDF Net Heating and Net Precip. NN Comparison | 20 |
| 3.5 | Net Precip. / Precipitable Water Joint PDF Comparison | 21 |
| 3.6 | Neural Network PW over Time in Simulation | 22 |
| 3.7 | Neural Network Net Precip. over Time in Coupled-Simulation | 23 |
| 4.1 | Net Precip. Residual Bins | 30 |
| 4.2 | Process for Training the Stochastic Parameterization | 32 |
| 4.3 | Stochastic Parameterization Process | 33 |
| 5.1 | Stochastic Parameterization Distribution Comparison | 39 |
| 5.2 | Stochastic Parameterization Joint PW / Net Precip. PDF Comparison | 40 |
| 5.3 | Mean Precipitable Water in Tropics over the Course of a Simulation | 41 |
| 5.4 | Stochastic Neural Network Net Precip. over Time in Coupled-Simulation | 42 |

# LIST OF TABLES

Table Number Page

5.1 Evaluation Methods . . . . . 36
5.2 Residual Model Performance . . . . . 36
5.3 Transitioner Performance . . . . . 37
5.4 Stochastic Bin-Transition Kolmorogov-Smirnov Divergence . . . . . 39

# GLOSSARY

COLUMN INTEGRATED: For variables with a z-dimension, column integrating means taking the mass-weighted integral of a single x, y coordinate column. The masses are given by the layer mass of the z-dimensional layers. Column-integrated variables are useful because they collapse a z-dimension vector into a single quantity, easing interpretation.

QT: Total non-precipitating Water (g/kg) (see Figure 2.6). This is a conserved quantity in the System for Atmospheric Modeling (SAM), and is best conceptually thought of as "moisture".

SLI: Liquid/Ice Static Energy (K) (see Figure 2.7). This is a conserved quantity in the System for Atmospheric Modeling (SAM), and is best conceptually thought of as "heat".

NET PRECIP.: The negative column-integrated QT parameterization. This is thought of as the column-integrated moisture sink due to sub-grid-scale processes.

NET HEATING: Column-integrated SLI parameterization. This is thought of as the column-integrated heat source due to sub-grid-scale processes.

GCM: General circulation model. GCMs solve simplified versions Navier-Stokes equations for fluid dynamics to define updates of a climate model.

PDF: Probability Density Function.

LOG PDF: The log of the probability density function. Graphs in this thesis use the Log of the PDF instead of the raw PDF to better visualize long tails of low probability. This is necessary because this thesis focuses on producing the correct long tails of distributions.

PRECIPITABLE WATER (PW): The column-integrated QT for a given column and given point in time. This describes the total amount of water in a column.

COMMUNITY ATMOSPHERE MODEL (CAM): Community Atmosphere Model. This is a state of the art climate model. The parameterization developed in this thesis is compared against the parameterization scheme in the CAM.

SYSTEM FOR ATMOSPHERIC MODELING (SAM): The System for Atmospheric Modeling is a climate model developed by Marat Khairoutdinov that is used to generate training data and to evaluate parameterizations in this thesis.

NEAR-GLOBAL AQUA PLANET (NG-AQUA): A planet modeled after Earth that is covered in ocean, and is cylindrical instead of spherical. The properties of the planet, such as sea surface temperature, energy received from the sun, Coriolis effect, length of day, etc, are designed to mimic Earth.

# Chapter 1

# INTRODUCTION

## *1.1 Introduction to Climate Models*

The state of the atmosphere develops from one point in time to the next according to the Navier-Stokes equations for fluid dynamics. Climate models approximate these equations for discrete time and space. The state of the atmosphere is represented in a climate model by a three dimensional grid of cells, where everywhere within each cell is assumed to have the same atmospheric conditions.

High resolution atmospheric models resolve small scale climate processes such as cloud formations (Bretherton and Khairoutdinov, 2015). They typically have a grid scale of less than 5-km and are able to produce realistic weather simulations (Bretherton and Khairoutdinov, 2015). However, high resolution climate models are computationally expensive so it is not feasible to run them for long periods of time or for large areas such as the entire globe. Furthermore, real-world climate data is often only available at a coarse resolution. Consequently, coarse-grid climate models with grid scales between 25 km and 200 km are important for modeling global weather and long term climate patterns.

Coarse-grid climate models have do not resolve sub-grid-scale processes that have a significant effect on grid scale weather. For example, the grid scale of coarse-grid models is larger than the scale on which cloud formation occurs so this process is not resolved. As such, coarse-grid climate models must have two components: a grid-scale component which models the inter-grid flow of heat and moisture based off of the Navier-Stokes equations for fluid mechanics, and an additional sub-grid scale component called a parameterization which models the overall effect of sub-grid processes on the grid-scale weather (Palmer, 2001). Traditionally, parameterizations are determined by climate experts based off of their intuition

and observations of real world weather or high resolution simulations.

### *1.2 Convective Parameterization*

Cumulus convection in the tropics is perhaps the most difficult and dynamically important sub-grid process that requires parameterization in coarse-grid climate models (Majda, 2007, Nakazawa, 1988). Cumulus convection is the process in which a temperature differential between the top of the atmosphere and bottom of the atmosphere causes warm moist air to rise. As the air rises and cools, its moisture condenses. This phase change releases heat, resulting in a significant grid scale exchange of heat between the top and the bottom of the atmosphere which is important for a climate model to capture.

### *1.3 Known Problems with Existing Parameterizations*

Global climate models are known to have mean state bias, and much of this due to inadequate parameterization (Hwang and Frierson, 2013). The double inter-tropical convergence zone problem in which climate models produce too much rainfall in the southern hemisphere tropics demonstrates the sensitivity of global climate models to convective parameterization (Woelfle et al., 2018). Climate models also have trouble producing the correct daily cycle (diurnal cycle) of precipitation and the Madden Julian Oscillation, two important features of the tropical atmosphere (Jiang, 2017, Stratton and Stirling, 2012). Thus, an improved method for parameterization is called for.

### *1.4 Machine Learning Parameterization*

Recently, climate scientists have experimented with a different approach to parameterization. Instead of using physical intuition to construct a parameterization, they approach it as a function approximation problem (Brenowitz and Bretherton, 2018, Krasnopolsky et al., 2005, O'Gorman and Dwyer, 2018, Rasp et al., 2018). Given a current state of the atmosphere as an input, a parameterization function should produce an output representing the effects

of sub-grid processes. With an accurate training set, the field of machine learning provides methods for constructing a function that can perform this task.

Brenowitz and Bretherton (2018) train a neural network to produce a parameterization using training data from a high resolution 4-km simulation that resolves convection processes. Data from the 4-km simulation is averaged over 160-km by 160-km cells (coarse-grained) such that data is in the coarse-grid setting where a parameterization is necessary. A neural network is then trained to predict the residual of the GCM's resolved tendency due to fluid mechanics on this coarse-grid data. This residual is the unresolved sub-grid processes that must be parameterized. The neural network is able to predict 60% to 70% of the variance of the sub-grid processes. They show that this parameterization outperforms conventional methods (Brenowitz and Bretherton, 2018).

This thesis builds upon Brenowitz and Bretherton (2018)'s approach by introducing stochasticity into their neural network parameterization. The method and results are based on the same training dataset used by Brenowitz and Bretherton (2018); this dataset is described in detail in Chapter 2. The theoretical motivation for the stochastic parameterization is presented in Chapter 3. Chapter 4 presents previous approaches to stochastic parameterizations and details the new methodology for stochastic machine learning parameterization. Chapter 5 presents the results of the stochastic parameterization, and the conclusion is stated in Chapter 6.

# Chapter 2

# THE DATASET

The dataset is obtained from a high resolution simulation that is run for 80 days with outputs stored every 3 hours. The important variables in the climate model are detailed here to provide the flavor of the training data.

## *2.1 Description of the Simulation*

The System for Atmospheric Modeling (SAM) version 6.10 (Khairoutdinov and Randall, 2003) is used with 4-km grid spacing for the simulation. The model is run on a near-global aqua planet (NG-Aqua) measuring 20480 km zonally by 10240 km meridionally. This means that the planet is roughly the size of Earth but is entirely covered in ocean, and is cylindrical instead of spherical. The properties of the planet, such as sea surface temperature, energy received from the sun, Coriolis effect, length of day, etc, are designed to mimic Earth but with some simplifications. For example, the time of day is equivalent everywhere on Earth (Brenowitz and Bretherton, 2018). The NG-Aqua setting is desirable for evaluating machine-learned parameterizations because including land introduces a lot of complexity and computational expense, and cumulus convection, the main parameterization event of interest, occurs frequently in an ocean-covered planet. If machine-learned parameterizations are successful in the NG-Aqua setting, then they will likely be successful in the Earth setting.

The simulation has 34 vertical levels. The grid layer closest to the surface has a height of 75 meters, and the grid layer farthest from the surface has a height of 1200 meters. Each 34 dimension vertical vector is referred to as a "column". To generate the training date, a 4-km grid cell simulation is run for 100 days with updates every 10 seconds. Every 3 hours, the full state of the model is stored; analysis is only considered on the last 80 days after memory

of initial conditions is lost (Brenowitz and Bretherton, 2018).

### *2.2 Coarse Graining the High Resolution Data*

The high resolution output is averaged over 160-km by 160-km grid cells with the vertical grid cell unchanged. At this scale, a climate model can not resolve convective processes. The dynamical tendency variables (also referred to as "forcing variables") which describe the rate of change of moisture, heat and wind are averaged along the edges of the 160-km by 160-km boxes instead of over the entire box. As expected, in the coarse grained setting, the forcing variables do not correctly describe the change in state between each 3-hour time step; a parameterization is necessary.

The final coarse grained dataset on which the machine learning parameterization is trained has 80 days of data separated at 3-hour intervals with 128 x-coordinates, 64 y-coordinates and 34 z-coordinates. This results in 8,192 columns and 278,528 grid cells for each time step, and 5,242,880 column observations over the entire time series (Brenowitz and Bretherton, 2018). For each column, there are 15+ variables; the most important ones are presented in the next section. This is many more observations than is usually available for machine-learning use cases. Machine-learning parameterization is a data rich problem, so over-fitting is rarely an issue.

### *2.3 Climate Model Variables*

Figures 2.1-2.9 show the most important variables in the dataset used in this thesis. The plots are extracted from the coarse grained data that is averaged from the high resolution simulation as described above.

Sensible heat flux (SHF), latent heat flux (LHF) and solar TOA insolation (SOLIN) describe the sources and sinks of energy in the atmosphere and are shown in Figures 2.2, 2.3, 2.4, respectively. These have a large effect on temperature, as they describe how the atmosphere is heated or cooled.

Total non-precipitating water (QT) and liquid/ice static energy (SLI), shown in Figures

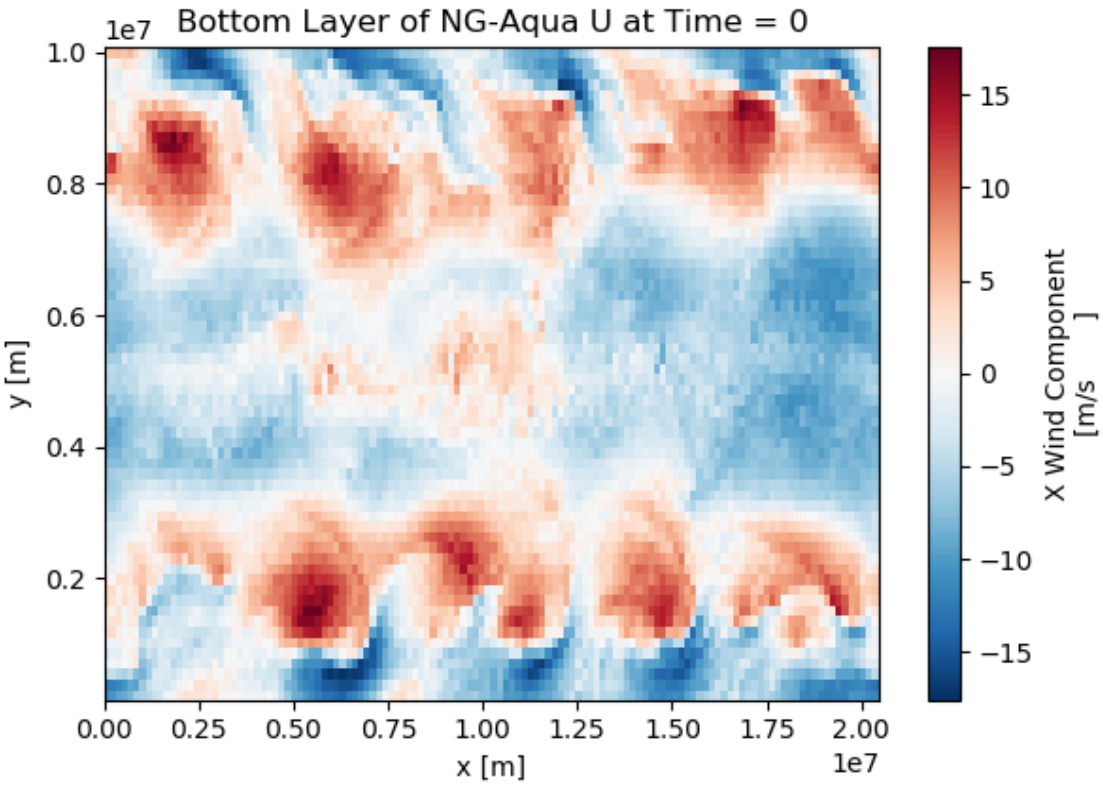


Figure 2.1: Near surface Zonal Winds (m/s).

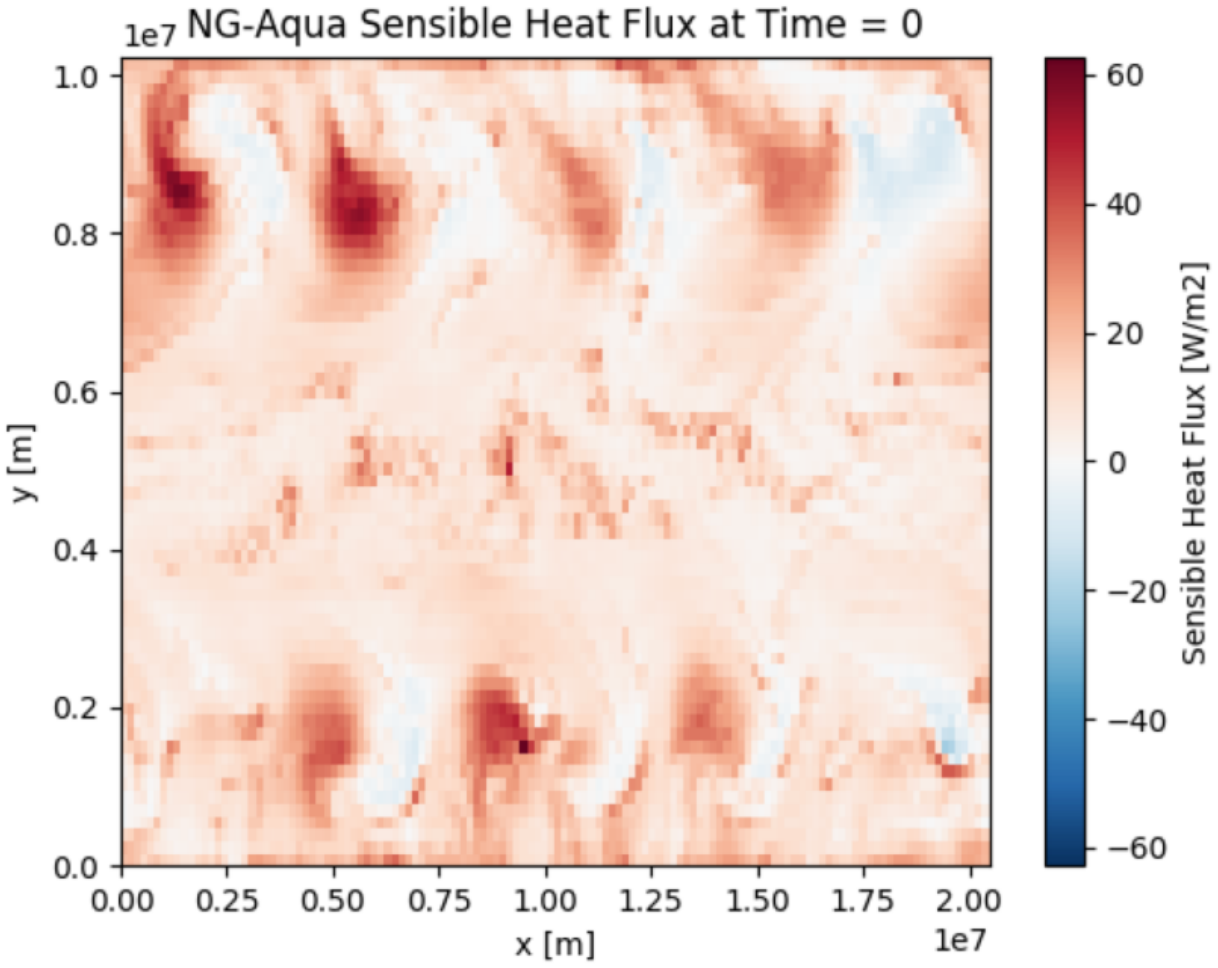


Figure 2.2: Sensible Heat Flux ($W/m^2$). Sensible heat describes the exchange of energy between the surface (the ocean) and the atmosphere due to the difference in temperature.

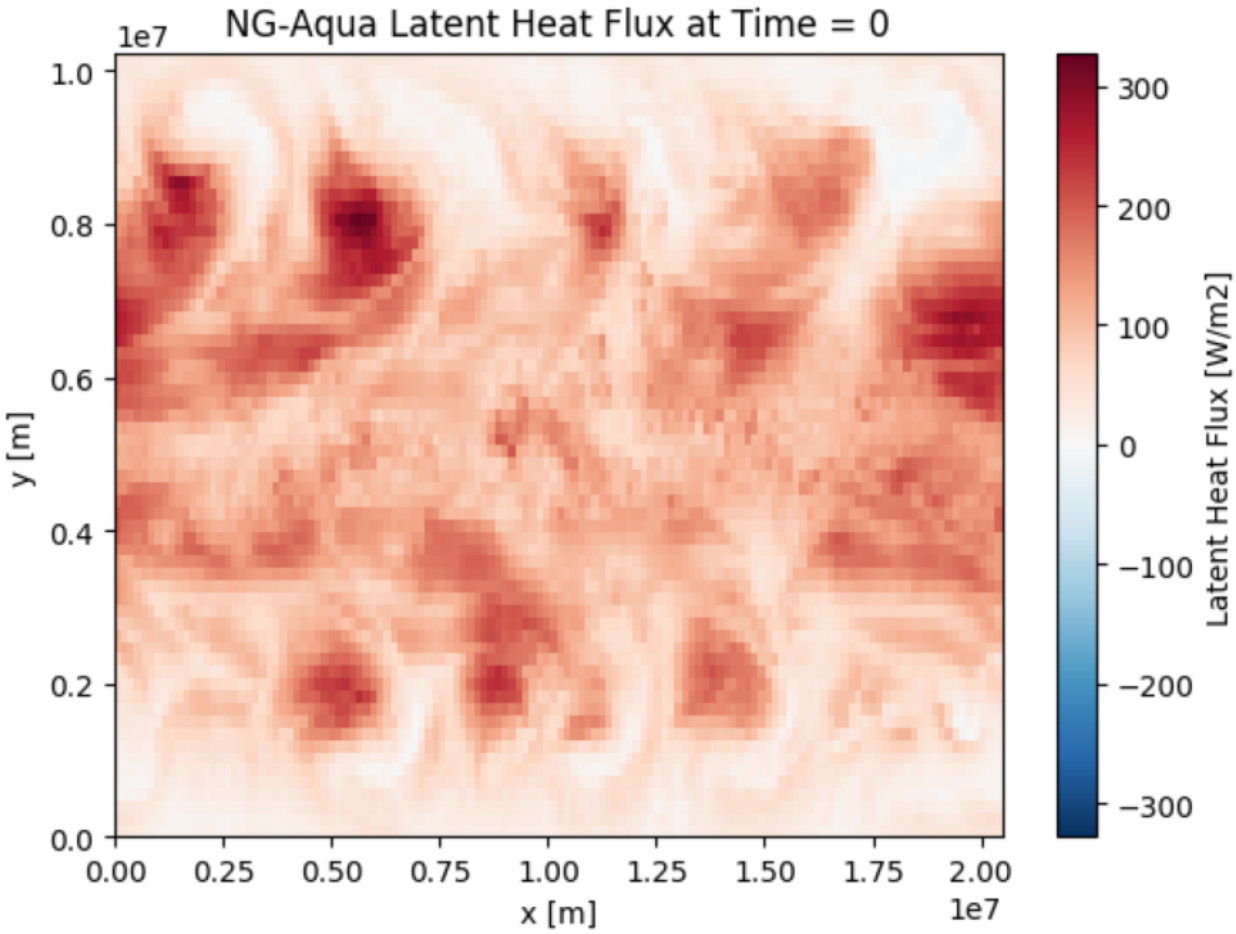


Figure 2.3: Latent Heat Flux ($W/m^2$). Evaporation (or sublimation) takes place at the surface of the ocean. Latent heat flux describes the cooling at the surface due to the latent heat required for this phase transition.

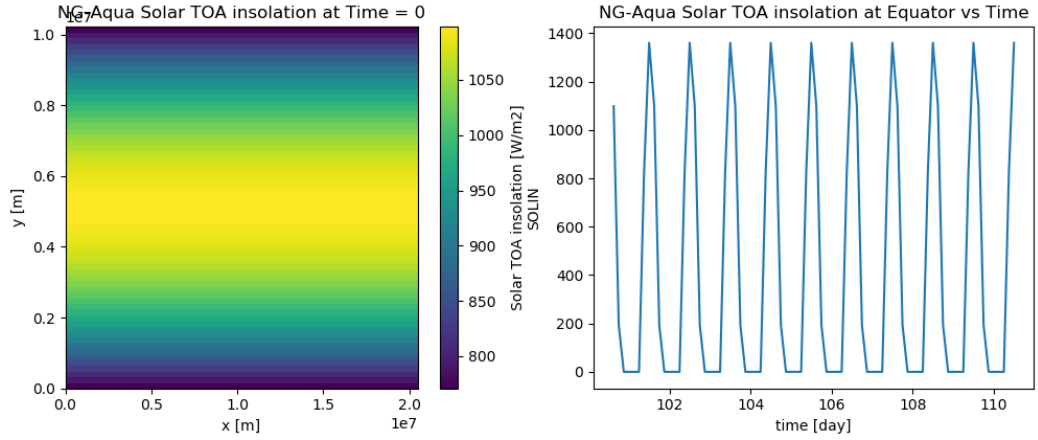


Figure 2.4: Solar TOA Insolation (W/$m^2$) describes the energy received from the sun. For a given latitude, the position of the sun is assumed the same for every longitude in the simulation in order to simplify calculations. The SOLIN for the entire planet goes to zero at night and peaks in the middle of the day.

2.6 and 2.7, are the two important conserved quantities from which the state of the weather can be derived, and as such are denoted as the "state" of the atmosphere. QT and SLI are conceptually thought of as moisture and heat, respectively; a full description of these variables is given by Khairoutdinov and Randall (2003). QT, thought of as moisture, is essential because it describes how much water is in a grid cell. The amount of water, along with SLI, determines the humidity and precipitation.

The right panel of Figure 2.6 shows that the majority of the moisture is near the surface of the ocean. This is because the amount of water vapor that air can hold decreases exponentially with the temperature, which decreases with height above the ocean. This is known as the Clausius-Clapeyron relationship. The right panel of Figure 2.7 shows that SLI actually increasing with height above the ocean. This is because pressure decreases more rapidly than temperature with height and SLI is a measure of heat that is conserved under changes in temperature, so Figure 2.7 is not showing that temperature decreases with height.

FQT and FSLI describe the grid scale rates of change of QT and SLI due to grid-scale

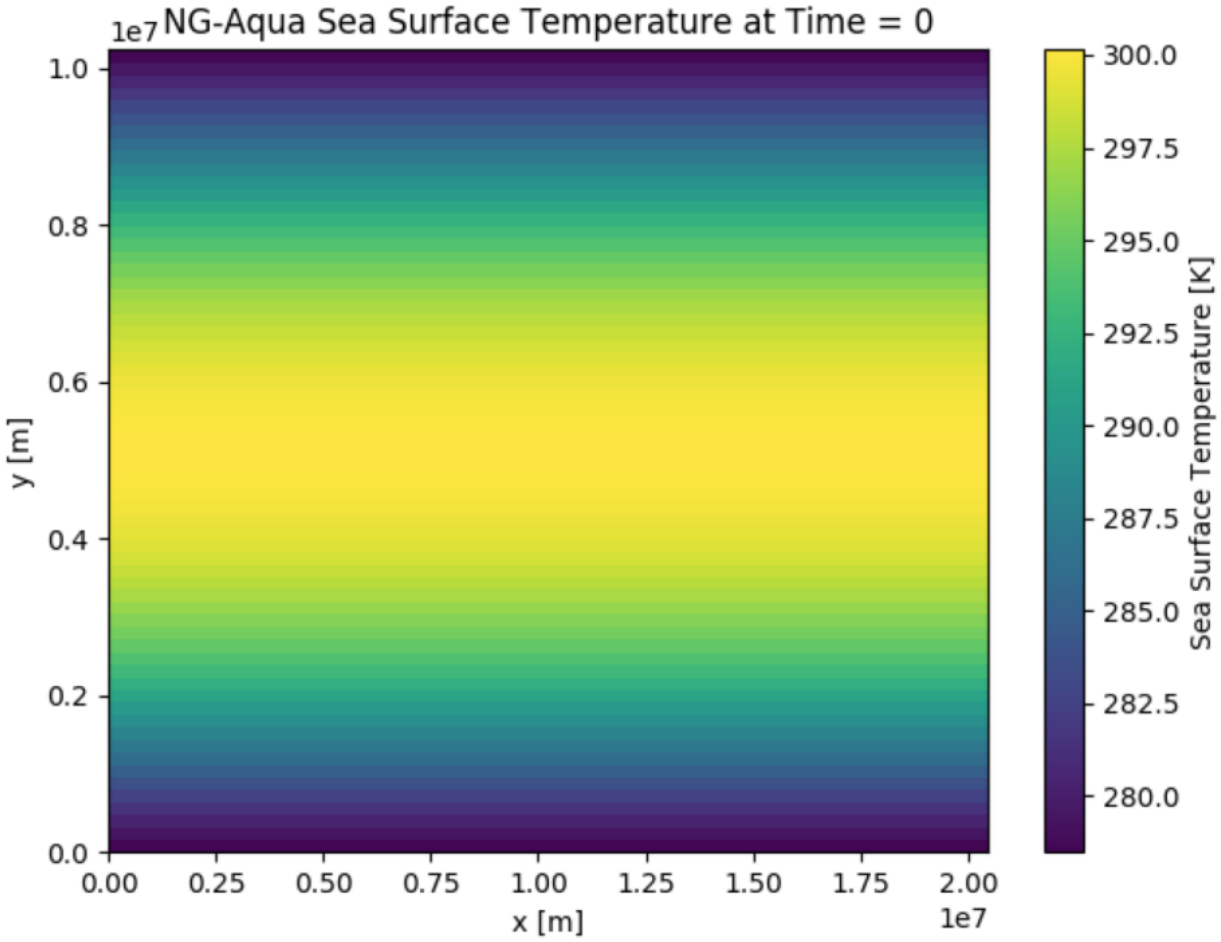


Figure 2.5: Sea Surface Temperature (K) (SST) describes the temperature of the surface of the ocean at each location on the planet. The sea surface temperature does not change throughout the simulation.

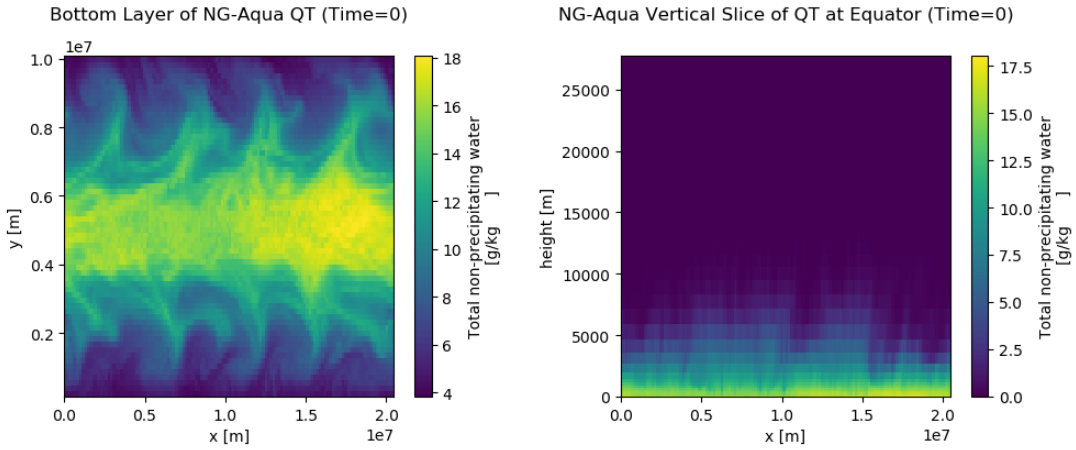


Figure 2.6: Total Non-Precipitating Water (g/kg) (QT). QT is the sum of mixing ratios of water vapor, cloud liquid water, and cloud ice, where the mixing ratio is the mass of the variable of interest divided by the mass of dry air in a grid cell. The left panel shows the QT for the bottom layer of the atmosphere at time = 0. The right panel is a vertical slice at the equator at time = 0.

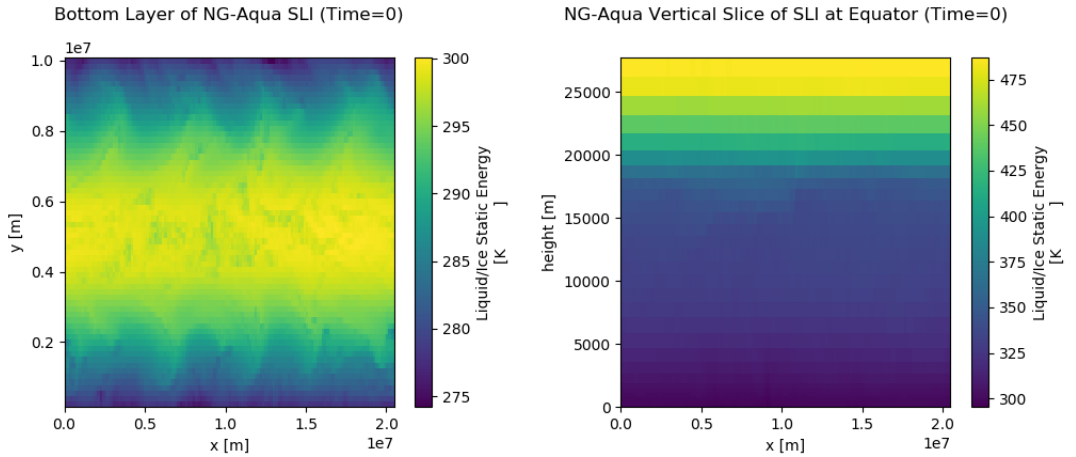


Figure 2.7: Liquid/Ice Static Energy (Kelvin) (SLI). SLI is a conserved quantity of energy that is conserved under changes in pressure, and has units Kelvin (Khairoutdinov and Randall, 2003).

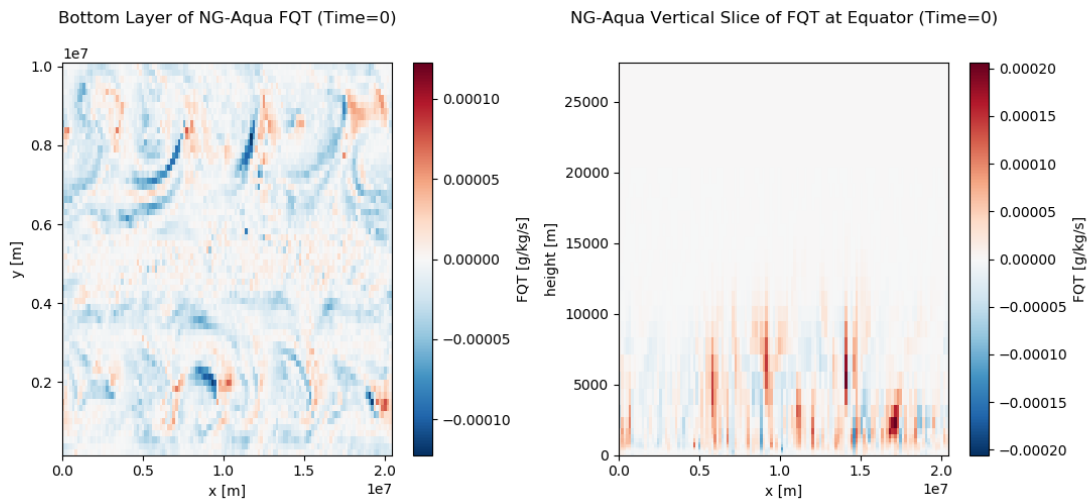


Figure 2.8: QT-Forcing / Moistening (g/kg/s) (FQT). FQT is the grid scale rate of change of QT (Figure 2.6) due to grid scale dynamics.

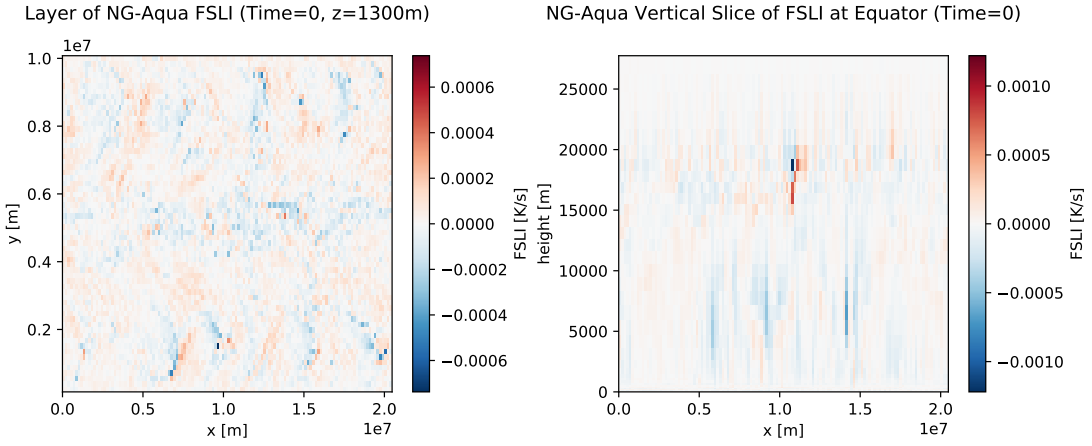


Figure 2.9: SLI-Forcing / Heating (K/s) (FSLI). FSLI is the grid scale rate of change of the SLI (Figure 2.7) due to grid scale dynamics.

dynamics. In the right panel of Figure 2.8 near $x = 0.9$, there is a red vertical line between 500 meters and 1000 meters above the ocean where the grid scale forcing of QT is high, indicating that QT is increasing rapidly, and a blue area below that where QT is decreasing. This is a convective event where moisture is being transferred from the bottom of the atmosphere to the top of the atmosphere. This event is happening at a scale that is visible on the 160-km grid scale, indicating that it is resolved by a coarse-grid climate model. However, FQT and FSLI do not fully describe the rates of change of QT and SLI so a parameterization is necessary.

# Chapter 3
# MOTIVATION FOR STOCHASTIC PARAMETERIZATION

This section describes the work of Brenowitz and Bretherton (2018) that uses machine learning to build a parameterization, the problems with their approach, and the motivation for a stochastic modification to their approach.

## *3.1 Brenowitz and Bretherton (2018) Approach to Parameterization*

The coarse-grid variables FQT and FSLI shown in Figure 2.8 and Figure 2.9 do not represent the sub-grid processes such as cumulus convection which affect the state of the climate, QT and SLI. The parameterization adjusts the rate of change of QT and SLI as determined by the Navier-Stokes differential equation solver such that the climate model updates QT and SLI from time $t$ to time $t + \Delta t$ by

$$x(t + \Delta t) = x(t) + \Delta t(g(t) + Q(t)) \tag{3.1}$$

where $Q = \{Q_{QT}, Q_{SLI}\}$ is the parameterization, $g = \{FQT, FSLI\}$ is the grid scale forcing, and $x = \{QT, SLI\}$ is the current state. Thus, the parameterization can be thought of as a correction of FQT and FSLI. A machine learning problem is readily formed from this setup: the desired output of the parameterization is obtained by rearranging Equation 3.1 to solve for $Q$, and using the values of $g$ and $x$ from the coarse grained data.

Brenowitz and Bretherton (2018) take the single-column approach to their machine-learned parameterization, as does this thesis. This assumes that the parameterization depends only on the value of variables within a given grid column such that the parameterization for each cell of a column depends only on the current state of all of the other cells in the column, but not on the state of neighboring columns, and not on previous states. This is a reasonable assumption for large 160-km by 160-km grid cells.

The machine learning model is trained to predict the parameterization from the current state $x$, or QT and SLI, and other variables $y$ which include LHF, SHF, SOLIN, and SST. Given that each column has 34 layers, the machine learning model has a 72-dimensional input and a 68-dimensional output as shown in Figure 3.1. The model couples the QT and SLI forcings because they are so physically coupled that two separate models are not necessary.

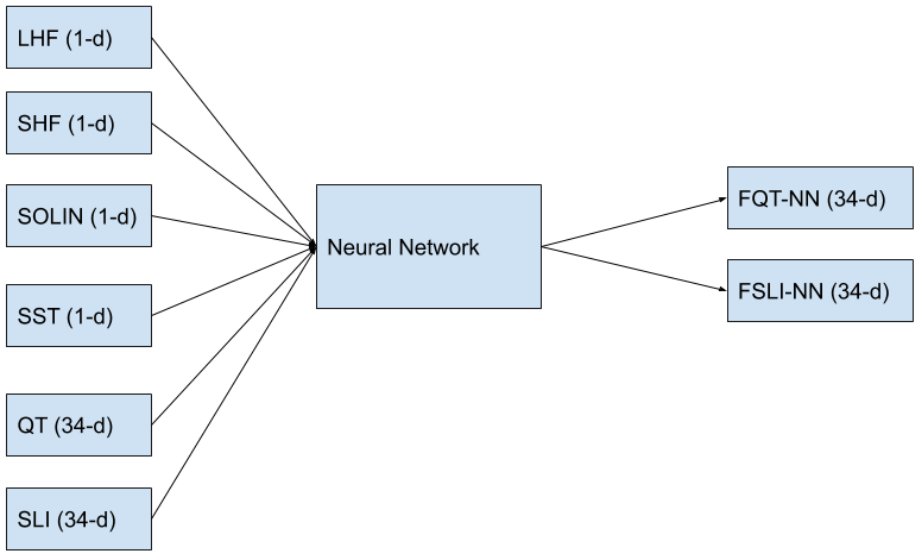


Figure 3.1: Neural Network Parameterization Inputs. The neural network is trained with three densely connected hidden layers with 256 nodes per layer, and rectified linear unit (ReLU) activation; the model has 160,266 free parameters. It is trained using stochastic batch gradient descent with the Adam optimizer for five epochs.

### *3.2 Evaluation of Brenowitz and Bretherton (2018) Approach*

Brenowitz and Bretherton (2018) show that their approach is an improvement over existing parameterization approaches using a "single-column model" evaluation method. The single-column model method does not evaluate the performance of the parameterization in a full climate model simulation with the parameterization. Instead, the coarse-grained dataset is used as inputs to the neural network. The state is initialized from the beginning of the coarse-grained dataset and then updated with three hour time intervals using the FQT and FSLI from the coarse-grained dataset and the output of the neural network parameterization. The inputs to the neural network are the values of LHF, SHF, SOLIN, and SST from the coarse-grained dataset and values of QT and SLI that are predicted from the simulation. The single-column model is run forward for 80 days with a three hour time step using the update rule defined in Equation 3.1, and the simulated state is compared to the true states observed in the coarse grained dataset.

The neural network parameterization is compared to the Community Atmosphere Model (CAM) version 5 (Mapes and Neale, 2011) parameterization, a commonly used parameterization developed using the standard method: the physical intuition of climate scientists based on observed climate patterns and high resolution simulations. Brenowitz and Bretherton (2018) show that their neural network parameterization thoroughly outperforms CAM in the single-column model evaluation setting, obtaining a mean absolute deviation of less than half of the CAM over the 80 day single-column model simulation (Brenowitz and Bretherton, 2018).

The single-column model evaluation method is fundamentally different from evaluating the parameterization in a full simulation because the forcings, $g$ from Equation 3.1, and other variables are not updated based on the output of the parameterization and the Navier-Stokes' equations are never solved. In the true simulation setting, the output of the parameterization has a significant effect on the the grid scale forcings and wind which in turn change the state. Consequently, the observed strong performance of the neural network parameterization in

the single-column model setting does not guarantee that the neural network parameterization will work well in a full climate model simulation.

### *3.3 Problems with Deterministic Parameterization*

A significant pitfall of the neural network approach is that it is deterministic. Given a specified input, the neural network always produces the same output. However, in reality, for a given set of coarse-grid scale inputs there is a probability distribution of outputs that can occur. In other words, if equivalent coarse-grid conditions occur two separate times, the two sub-grid process will be different. Intuitively this makes sense: given weather conditions averaged over a 160-km by 160-km grid cell, it unlikely that it is possible to predict the exact timing, location, and magnitude of a convective event. Thus, it is not realistic for the neural network to always produce the same output for a given input. As demonstrated below, this limits the applicability of the neural network parameterization.

Figures 3.2, 3.3 and 3.4 illustrate the effects of the deterministic prediction. The distribution of outputs produced by the neural network is higher towards the expectation of the distribution than the true distribution (Figure 3.4), and it does not produce as extreme of values as the true parameterization (Figures 3.2 and 3.3). This is because the the neural network is trained to output the expected value of the parameterization for a given input, as the expected value minimizes the mean squared error which is the loss function used for training. The inability of the neural network to produce the same distribution of parameterizations as the training set is not due to under-fitting; adding additional layers and increasing the number of epochs for training does not improve training or testing performance of the neural network as would be expected if the neural network was under-fitting.

As shown in Figure 3.5, the relationship between net precipitation and precipitable water is roughly exponential. This exponential relationship is well established (Bretherton et al., 2004, Rushley et al., 2018), and demonstrates the importance for a parameterization to produce the correct distribution of outputs. Net precip. describes the rate of change of precipitable water because precipitation removes water from the atmosphere. So, if there is

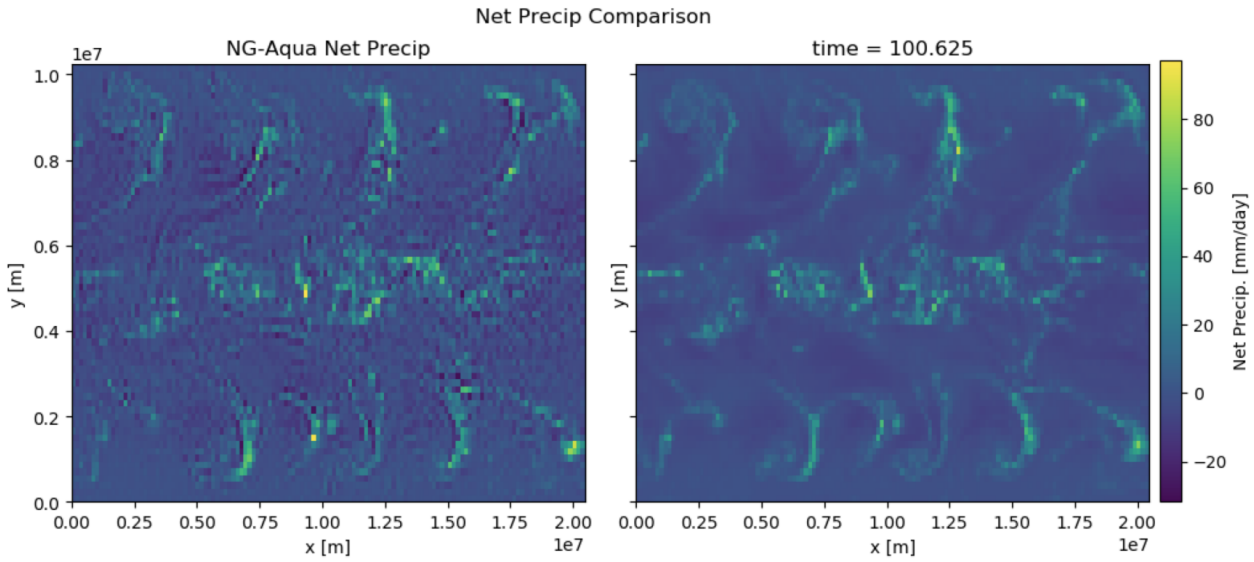


Figure 3.2: Net Precip. Comparison for time=0. The left panel shows the true net precip. parameterization based on rearrangement of Equation 3.1 of the training data; the right panel shows the net precip. produced by the neural network parameterization. The true parameterization has values of net precip. that are much higher than any values produced by the neural network, and values that are much lower than any produced by the neural network; the NG-Aqua parameterization also has higher zonal variance.

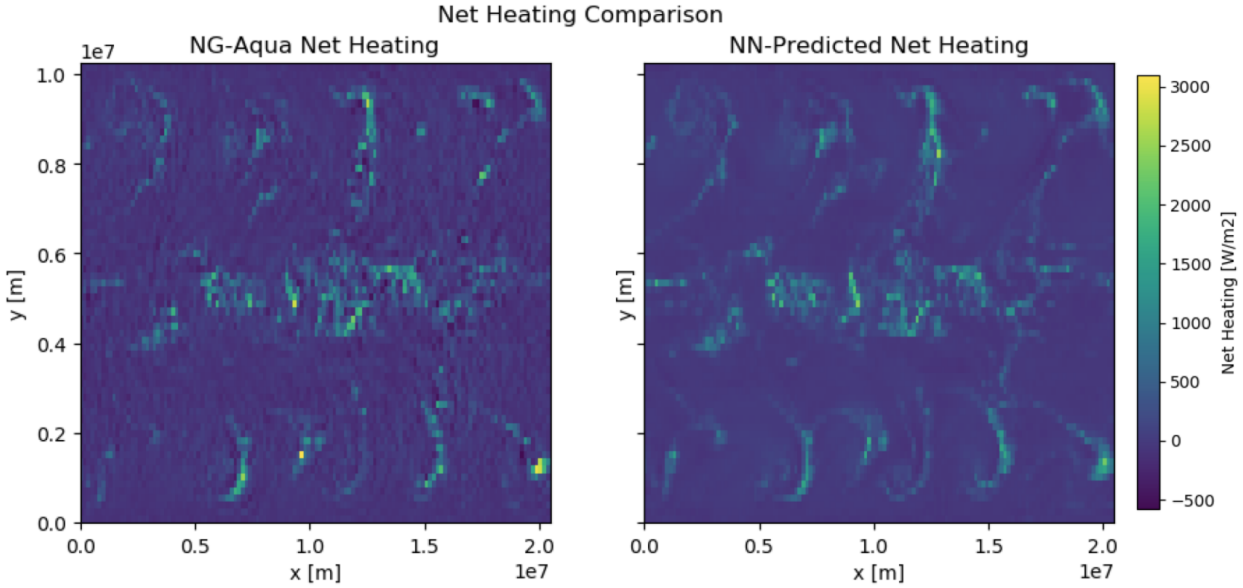


Figure 3.3: Net Heating Comparison for time=0. The left panel shows the true net heating parameterization based on rearrangement of Equation 3.1 of the training data; the right panel shows the net heating produced by the neural network parameterization. The true parameterization has much higher variance in net heating than the neural network.

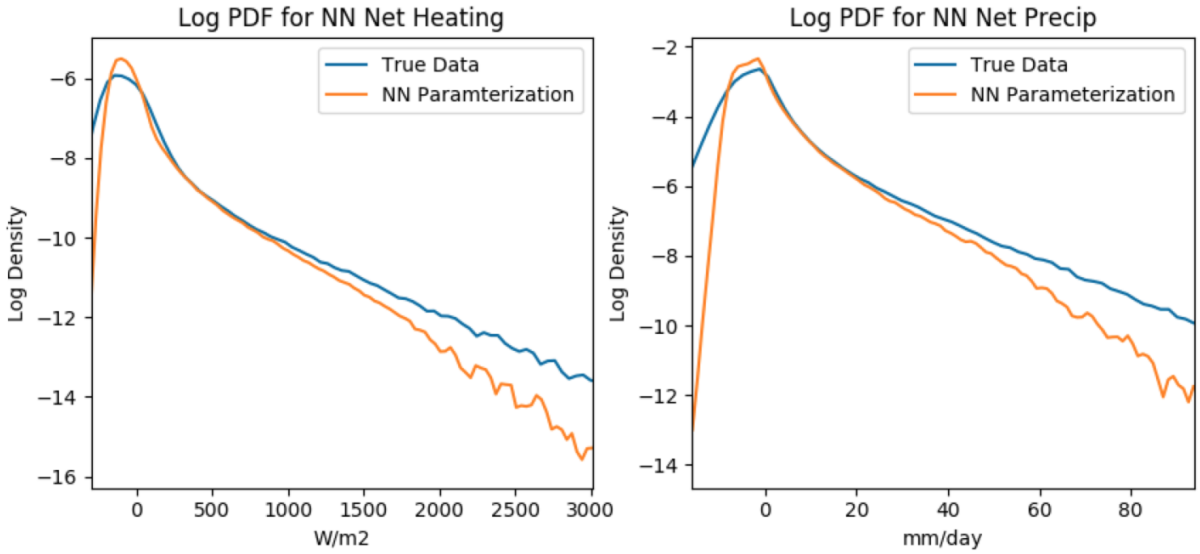


Figure 3.4: Log PDF Net Heating and Net Precip. NN Comparison. This is the log PDF of parameterizations over the entire globe for six days of data for net heating and net precip. The figure compares the log-pdf of the true parameterization to that produced by the neural network. The neural network has a more central distribution with higher density in near the expectation and lower density near the tails than the true parameterization.

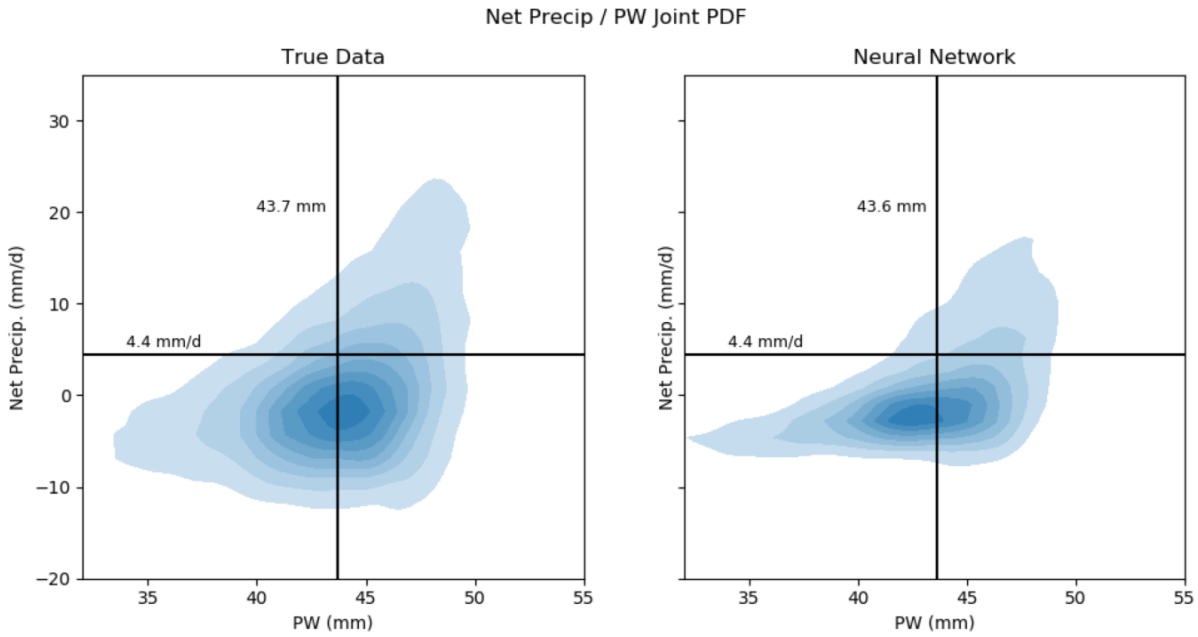


Figure 3.5: Net Precip. / Precipitable Water Joint PDF. For a given value of precipitable water, the neural network does not produce a wide enough distribution of precipitation.

a lack of variance in net precip., then the high states of precipitable water where the most precipitation occurs will never be reached.

This can be illustrated as follows, considering the left panel of Figure 3.5. Assume that PW is at its mean, 43.7 mm. At this state, the average net precip. is roughly zero, but it varies between -10 mm/day and 14 mm/day. The neural network will only produce values between roughly -8 mm/day and 10 mm/day as shown in the right panel of Figure 3.5. Now assume that in order for net precip. to moisten the atmosphere to the point where PW reaches its maximum observed value of 50 mm, the minimum possible net precip. of -10 mm/day must occur. This net precip. value is observed in the true data, but is never produced by the neural network when PW is at is at 43.7 mm. Thus, in this simplified scenario that ignores grid-scale moistening, a simulation with the neural network parameterization will never produce PW values of 50mm where net precip. values above 20 mm/day can occur, so extreme rainfall events will never happen. Because this moisture-precipitation relationship

is convex, a simulation with the neural network parameterization is expected to not produce enough precipitation on average.

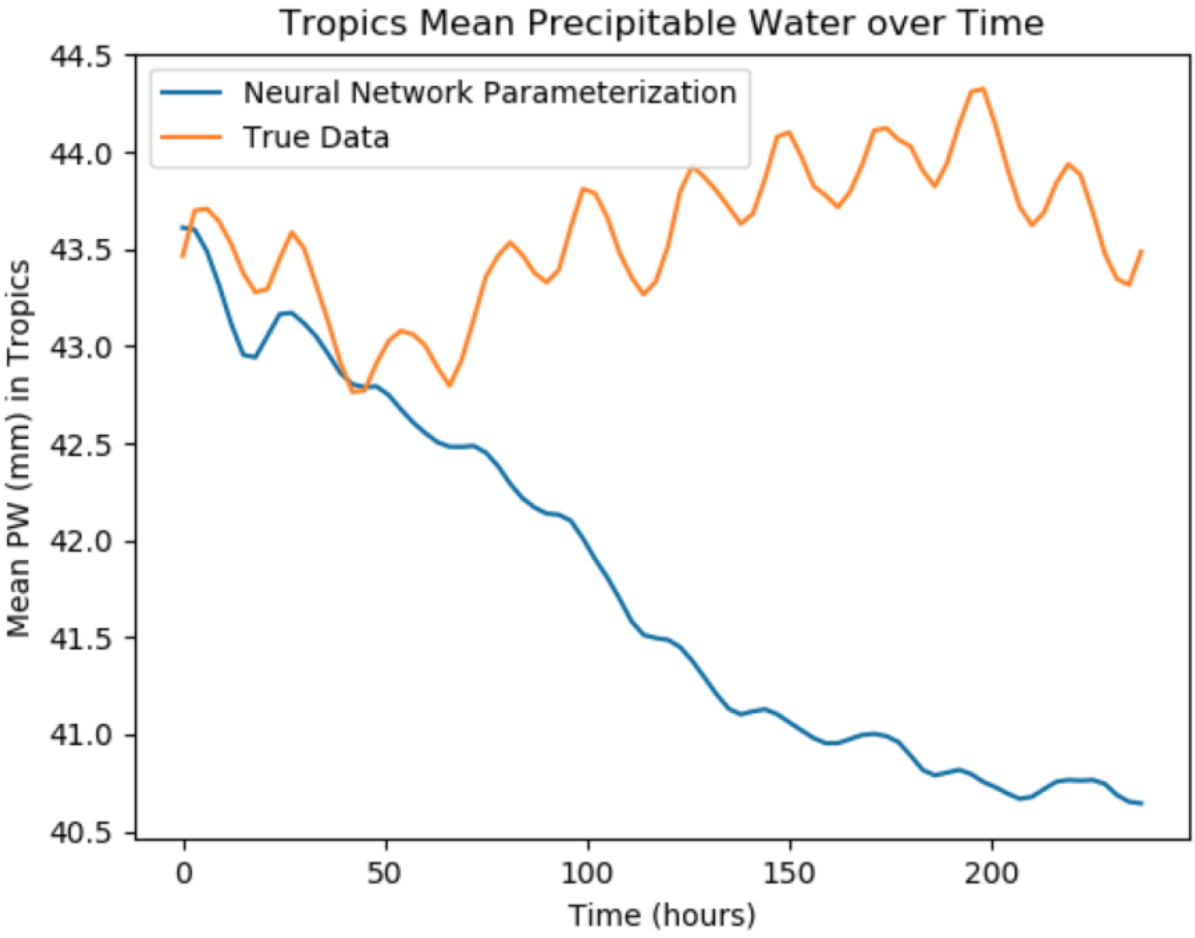


Figure 3.6: Neural Network PW over Time in Simulation. When a simulation is run with a neural network parameterization, the precipitable water decreases over time in the tropics compared to the true data.

When the neural network is used as the parameterization in a simulation, the expected problem of lack of precipitation occurs. The initial conditions for the simulation are obtained from the training dataset such that the results can be compared directly to the training data. The System for Atmospheric Modeling (SAM) is the climate model used for the simulation. This is the same climate model that generates the high-resolution training dataset except it

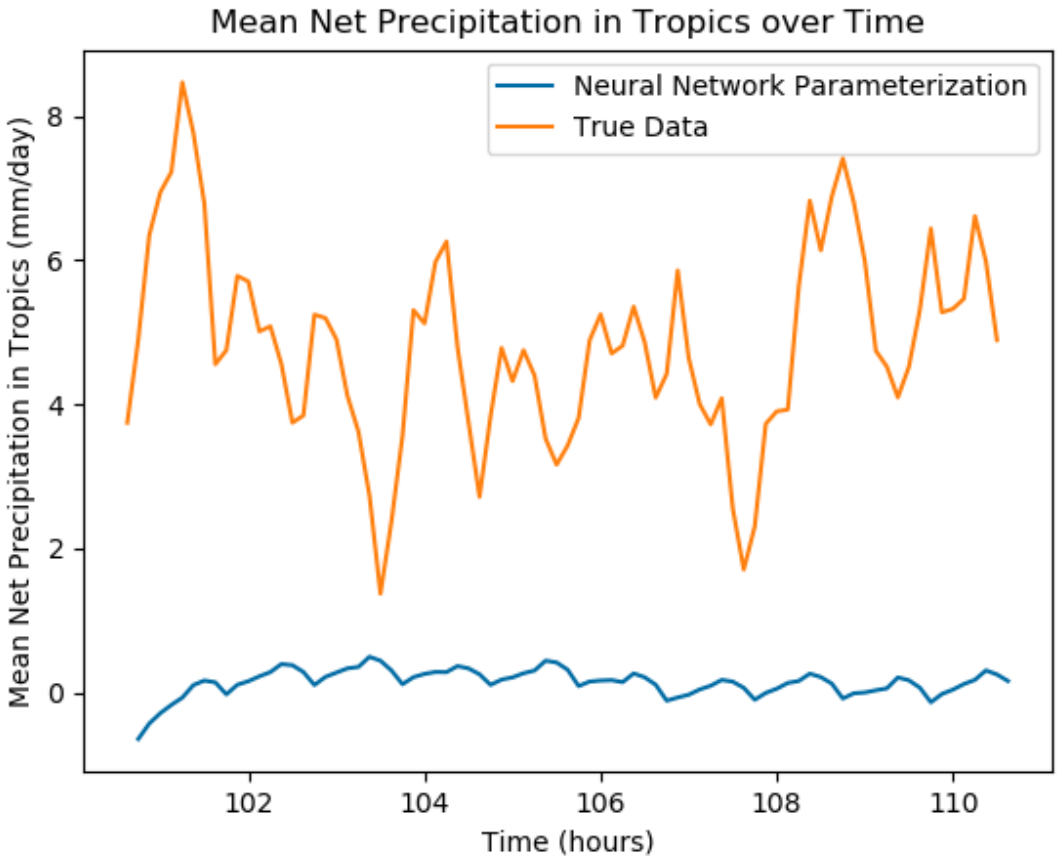


Figure 3.7: Neural Network Net Precip. over Time in Coupled-Simulation. When a simulation is run with a neural network parameterization, the net precip. is well below that of the true data.

uses a 160-km by 160-km grid size, the neural network parameterization, and a 120-second time step.

As shown in Figures 3.6 and 3.7, the net precip. is uniformly below the true data in the tropics, and the precipitable water decreases every day in the simulation in the tropics while in the true data it remains relatively constant. This is exactly the nature of the problem that is expected with a deterministic parameterization that produces a distribution of outputs that is too central.

The precipitation bias is most prevalent in the tropics where it weakens the Hadley circulation to the point where it disappears within a ten day simulation. The Hadley circulation is a constant wind pattern where heavy rainfall near the equator releases heat, warming the air. The hot air rises and moves north in the northern hemisphere, or south in the southern hemisphere, descends in the subtropics, and then circles back towards the equator. The circulation continues throughout the high resolution NG-Aqua simulation, but completely dies out in under ten days in the coarse-grid simulation with the neural network parameterization.

### *3.4 Goal of this Thesis*

The loss of the Hadley circulation, net precip. bias, and precipitable water bias in the SAM simulation with the neural network parameterization motivate this thesis. The goal of this thesis is to modify the neural network parameterization to be stochastic, such that it draws from the probability distribution of possible outputs for a given input, and produces a distribution of outputs that matches that of the training data. The stochastic parameterization should maintain the flexibility of the neural network and its improvements over existing methods. The approach should be computationally efficient so it does not significantly slow down the simulation. Lastly, the approach should be physically motivated so that it is an appealing solution to climate scientists. If the stochastic parameterization is implemented in this way, the hypothesis that the observed mean-state bias of precipitable water and precipitation in a simulation are caused by an improper representation of the parameterization pdf can be tested.

# Chapter 4

# METHODS

Existing approaches to stochastic parameterization are discussed in this chapter, and a Markov-chain approach that motivates the methods in this thesis is detailed. The mathematical construction of the stochastic parameterization problem is then stated. The assumptions for the approach developed in this thesis are then presented which yield a maximum likelihood optimization problem. Next, the implementation of the solution to the maximum likelihood optimization is presented. Lastly, and the full stochastic neural network parameterization is described.

## *4.1 Primary Stochastic Parameterization Approaches*

Several methods for stochastic parameterization have been proposed and implemented, including stochastic differential equations (Majda et al., 1999, 2003, Wilks, 2005), cellular automata (Shutts, 2005), the stochastic multicloud model (Goswami et al., 2017, Khouider et al., 2010), and multiplicative random noise (Buizza et al., 1999). Multiplicative random noise remains the most popular scheme in operational climate models; it consists of multiplying a parameterization by a random number near 1. While this method has been shown to significantly improves climate models (Berner et al., 2017), implementations are criticized as they are manually tuned to correct existing parameterizations after the fact and they lack strong theoretical motivation for their parameters (Berner et al., 2017).

## *4.2 Markov Chain Stochastic Parameterization*

A Markov chain based stochastic parameterization approach is proposed by Crommelin and Vanden-Eijnden (2008). A training dataset of observed states, denoted by $x(t)$, is divided into

16 evenly-spaced discrete bins. Then for each bin the observed parameterizations are divided into four discrete bins. The four bins are Markov states, and the parameterization changes between these Markov states stochastically via a Markov process as a simulation progresses. The transition probabilities depend on the previous Markov state, the bin for $x(t-1)$, and the current bin for $x(t)$, and they are computed directly from the observed transitions in the training dataset (Crommelin and Vanden-Eijnden, 2008). The parameterization output is the average parameterization in the training dataset for the bin for $x(t)$ and current Markov state.

The approach directly addresses the deterministic prediction problem. For a given $x(t)$, their parameterization produces one of four possible outputs instead of the same output every time. Furthermore, the probabilities of each output are computed directly from the training set, so the distribution of parameterizations is approximately correct. Given more training data and more Markov states, the distribution of parameterizations will approach the distribution in the training dataset (Crommelin and Vanden-Eijnden, 2008).

The approach is evaluated on the Lorenz '96 model which is a toy model that is frequently used for testing parameterization schemes (Crommelin and Vanden-Eijnden, 2008). It is shown to represent a much larger part of the distribution of observed parameterizations than a deterministic parameterization scheme (Crommelin and Vanden-Eijnden, 2008). However, the approach and is not evaluated with a real climate model so it is unclear whether it is operationally useful. Furthermore, the parameterization scheme is relatively simple as it only considers the bin for $x(t)$, the bin for $x(t-1)$, the previous Markov state, and the current Markov state, so it may not be able model more complex sub-grid process. Lastly, the method does not address multicollinearity for high dimensional parameterizations. Nonetheless, the Markov chain approach is a straightforward and effective method for introducing stochasticity.

The approach developed in this thesis is partially motivated by this Markov chain approach and by the stochastic multicloud model; it uses a Markov process to stochastically choose from a discrete set of parameterizations. However, the approach in this thesis re-

tains the neural network and its ability to model complex processes, it takes into account many of the variables in the climate model, it directly models the multicollinearity of the parameterization, and it is evaluated in a GCM simulation.

### *4.3 Mathematical Formulation and Assumptions*

In order to eliminate the need for a subscript indicating the latitude/longitude coordinate of a column, the methods developed below consider only a single grid column. The results for a single-column are easily generalized to all grid columns. In addition, the subscript $t$ gives the index of the time step, such that $x_{t+1}$ represents the state at the time that occurs $\Delta t$ after the time step indexed by $t$.

The neural network is a function $f$ that maps state $x_t$ (QT and SLI), and other variables $y_t$ (SOLIN, SHF, LHF, and SST) to the parameterization $Q_t$. The parameterization $Q_t$ is assumed to be equivalent to

$$Q_t = f(x_t, y_t) + \epsilon_t \tag{4.1}$$

where $f(x_t, y_t) = E[Q_t|x_t, y_t]$ and $\epsilon_t$ is the noise term. The neural network is assumed to be $f$. Plugging Equation 4.1 into Equation 3.1 yields

$$x_{t+1} = x_t + \Delta t(f(x_t, y_t) + g_t + \epsilon_{t+1}). \tag{4.2}$$

The values of $\epsilon_t$ are readily observable from the dataset as the residual of the neural network output

$$\epsilon_{t+1} = Q_t - f(x_t, y_t). \tag{4.3}$$

The error term $\epsilon_{t+1}$ is given by

$$\epsilon_{t+1} = h(\eta_{t+1}, x_t, y_t, \theta_h) + \tilde{\epsilon}_{t+1} \tag{4.4}$$

where $h$ is a continuous function, $\eta_t \in I^N$ represents the state at time $t$ of a Markov process with $N$ states, $\tilde{\epsilon_t}$ is a centered multivariate normal Gaussian with unknown covariance $\Sigma_t$, and $\theta_h$ is a parameter vector for $h$. $\Sigma_t$ is assumed to be diagonal. For instance,

$$\tilde{\epsilon_t} \sim N(0, \Sigma_t). \tag{4.5}$$

Each Markov state $\eta$ represents a range of of values for the $QT$ portion of column-integrated $\epsilon_t$. It is assumed that the probability of changing from $\eta = i$ at time $t$ to $\eta = j$ at time $t + 1$ is given by

$$P(\eta_{t+1} = j | \eta_t = i, x_t, y_t) = T_{i,j}(x_t, y_t, \theta_T) \tag{4.6}$$

where $T_i$ is a model for $\eta = i$, and there are $N$ such models, one for each Markov state, and $T_i$ defines the transition probabilities such that

$$\sum_{j=0}^{N-1} T_{i,j}(\cdot) = 1. \tag{4.7}$$

### *4.4 Maximum Likelihood Formulation*

Under these assumptions, a likelihood equation for the observed data is formed. The goal of the parameterization is to find parameter vectors $\theta_h$ and $\theta_T$ for functions $h$ and $T$ that maximize the likelihood of the observed data

$$P(x_1, x_2, ..., x_{n_t}, \eta_1, \eta_2, ..., \eta_{n_t}) = \prod_{t=0}^{n_t-1} P(x_{t+1}, \eta_{t+1} | x_0, y_0, ..., x_t, y_t, \eta_0, ..., \eta_t). \tag{4.8}$$

The Markov assumption simplifies the probability term in the product such that

$$P(x_{t+1}, \eta_{t+1} | x_0, y_0, ..., x_t, y_t, \eta_0, ..., \eta_t) = P(x_{t+1}, \eta_{t+1} | x_t, y_t, \eta_t). \tag{4.9}$$

This factors to

$$P(x_{t+1}, \eta_{t+1} | x_t, y_t, \eta_t) = P(x_{t+1} | x_t, y_t, \eta_{t+1}, \eta_t) P(\eta_{t+1} | x_t, y_t, \eta_t). \tag{4.10}$$

The Markov assumption of $\eta$ simplifies $P(x_{t+1}|x_t, y_t, \eta_{t+1}, \eta_t)$ to $P(x_{t+1}|x_t, y_t, \eta_{t+1})$. The shape of the density of the probability term $P(x_{t+1}|x_t, y_t, \eta_{t+1})$ is given by $\tilde{\epsilon}_t$ in Equation 4.4; it follows a multivariate Gaussian distribution with mean $x_{t+1}$ and covariance $\Sigma_t$, so

$$P(x_{t+1}|x_t, y_t, \eta_{t+1}) = (2\pi)^{\frac{-k}{2}}|\Sigma_t|^{-\frac{1}{2}}exp(\frac{-1}{2}(\hat{x}_{t+1} - x_{t+1})\Sigma_t^{-1}(\hat{x}_{t+1} - x_{t+1})) \tag{4.11}$$

where $k$ is the dimensionality in the $z$ direction, and

$$\hat{x}_{t+1} = x_t + \Delta t(f(x_t, y_t) + g(t) + h(\eta_{t+1}, x_t, y_t, \theta_h)). \tag{4.12}$$

The second part of the factor in Equation 4.10, $P(\eta_{t+1}|x_t, y_t, \eta_t)$, is a probability mass function with probabilities for each Markov states given by Equation 4.6. Plugging these into the likelihood Equation 4.6 gives

$$L(\theta_h, \theta_T) = \prod_{t=0}^{n_t-1}(2\pi)^{\frac{-k}{2}}|\Sigma_t|^{-\frac{1}{2}}exp(\frac{-1}{2}(\hat{x}_{t+1} - x_{t+1})\Sigma_t^{-1}(\hat{x}_{t+1} - x_{t+1}))T_{\eta_t,\eta_{t+1}}(x_t, y_t, \theta_T). \tag{4.13}$$

This equation is maximized when $h$ minimizes the mean squared error of $\hat{x}_{t+1} - x_{t+1}$ and $T$ maximizes the classification accuracy of the observed $\eta$ values.

### *4.5 Markov States*

The Markov states represent adjacent ranges of net precip. residuals that together cover every possible residual. The net precip. residuals are calculated by running the first ten days of training data covering every latitude/longitude coordinate through the neural network. Ten days is enough data to generate a large sample that covers the full range of observed residuals. The residuals are split into seven bins, where the bin split locations are chosen such that the within-bin variance of net precip. residuals weighted by the number of observations in each bin is minimized. This method for choosing bins limits the amount of variance that the $h$ models must predict on average, as for each bin, there is minimal variance of net precip. residual that $h$ must model. This has the effect of enabling the full stochastic model to fit the residuals more closely than if bin split locations are chosen arbitrarily. Seven bins are used because this number appears to be low enough that the transitioner can accurately model

bin transitions, but high enough that the outer bins adequately capture the most extreme values of net precip. residual.

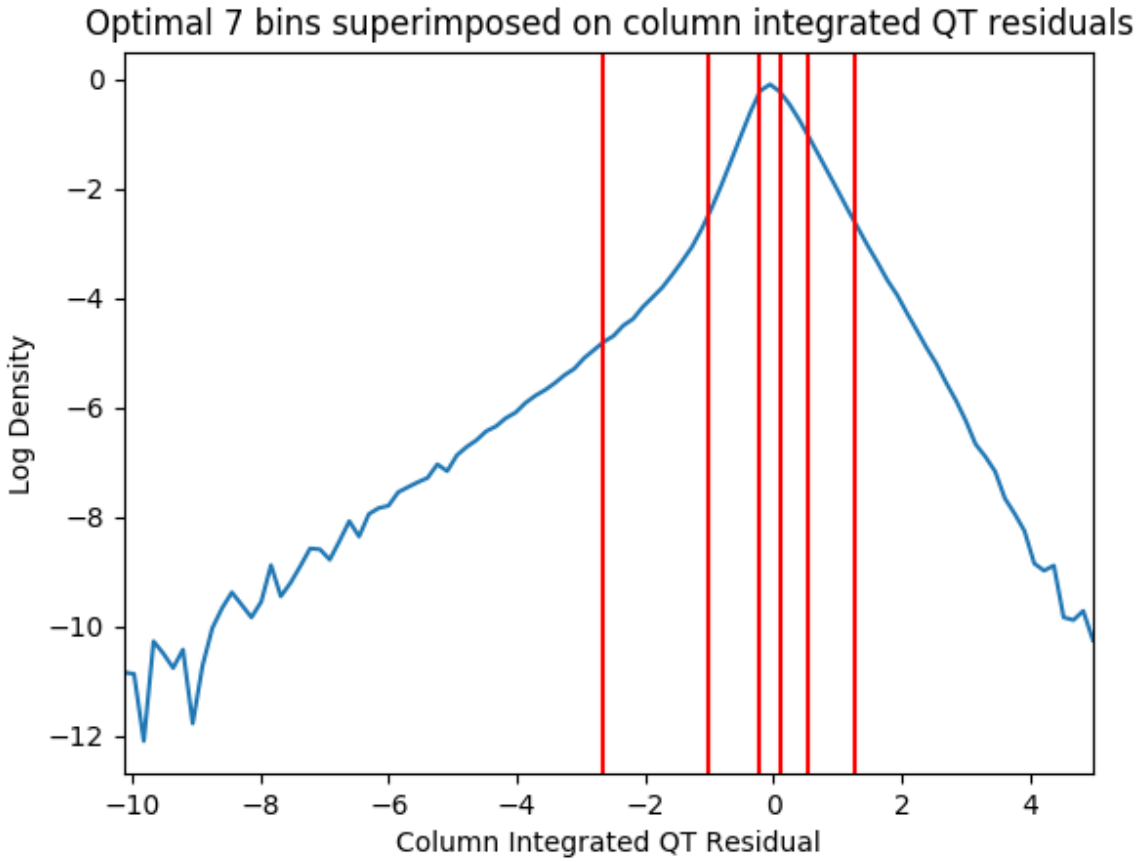


Figure 4.1: Net Precip. Residual Bin Choices. The bins are chosen to minimize the within-bin variance weighted by the number of observations in a bin.

### *4.6 Transitioner Model $T$*

A multinomial logistic regression model is trained for the transitioner model. A separate model is trained for each bin. Each model accepts cubic polynomials of column-integrated QT, column-integrated SLI, SHF, SOLIN, and SST as inputs. Cubic polynomials are used because on average over the time series, each variable tends to be nonlinear with latitude, such

that they reach a maximum near the equator and then drop with distance from the equator. Cubic polynomials of the variables enable a logistic regression model to capture relationship such that the parameterization can be latitude-specific. The transitioner models have a softmax output so they output probabilities of transitioning to each bin, satisfying Equation 4.7. Transitions are randomly chosen according to these probabilities. The transitioner model coefficient matrix $\theta_T$ is chosen to maximize the log likelihood

$$\hat{\theta}_T = argmax \sum_t log(T_{\eta_t,\eta_{t+1}}(x_t, y_t, \theta_T)) \tag{4.14}$$

where the $\eta$ values are obtained from the training dataset. This equation maximizes the transitioner term in Equation 4.13.

### *4.7 Residual Model* $h$

The residual model $h$ is a $\eta$-specific linear regression model that takes QT, SLI, SHF, LHF, SST, and SOLIN as inputs and outputs a 68-dimensional value representing the residual of the neural network.

The residual models are trained to minimize the mean squared difference between the observed residuals and predicted residuals

$$\hat{\theta}_{h,i} = argmin_{\theta_h} \sum_{t=0}^{n_t-1} \sum_{z=1}^{k} (h_i(x_t, y_t, \theta_h) - (x_t - \frac{x_{t+1} - x_t}{\Delta t} - f(x_t, y_t) - g_t))_z^2 * I_i(\eta_{t+1}) \tag{4.15}$$

where $z$ represents the vertical layer and $I_i(\eta_{t+1})$ is the indicator function that is 1 when $\eta_{t+1} = i$ and 0 otherwise. Equation 4.15 is the sum of squared errors of the neural network with the stochastic residual adjustment given by $h_i(x_t, y_t, \theta_h)$. Since Equation 4.13 has the squared difference between the predicted $\hat{x}$ and the observed $x$ in the exponential multiplied by a diagonal unknown covariance matrix, the solution in Equation 4.15 will maximize the exponential term in Equation 4.13. Combined with the transitioner model solution, the likelihood Equation 4.13 is maximized without any knowledge of the covariance matrix of the residuals.

The residual models know the range of net precip. residual that they should produce because they are $\eta$-specific, and $\eta$ represents a range of values for net precip. residuals. Conceptually, the residual models transform this residual range and the current state variables QT, SLI, SHF, LHF, SST and SOLIN into a 68-dimensional residual. They predict the magnitude of the net precip. residual within $\eta$'s range, and how that net precip. residual should be split up among the vertical layers, directly addressing multicollinearity among the vertical layers.

### *4.8 Overview of Stochastic Parameterization Method*

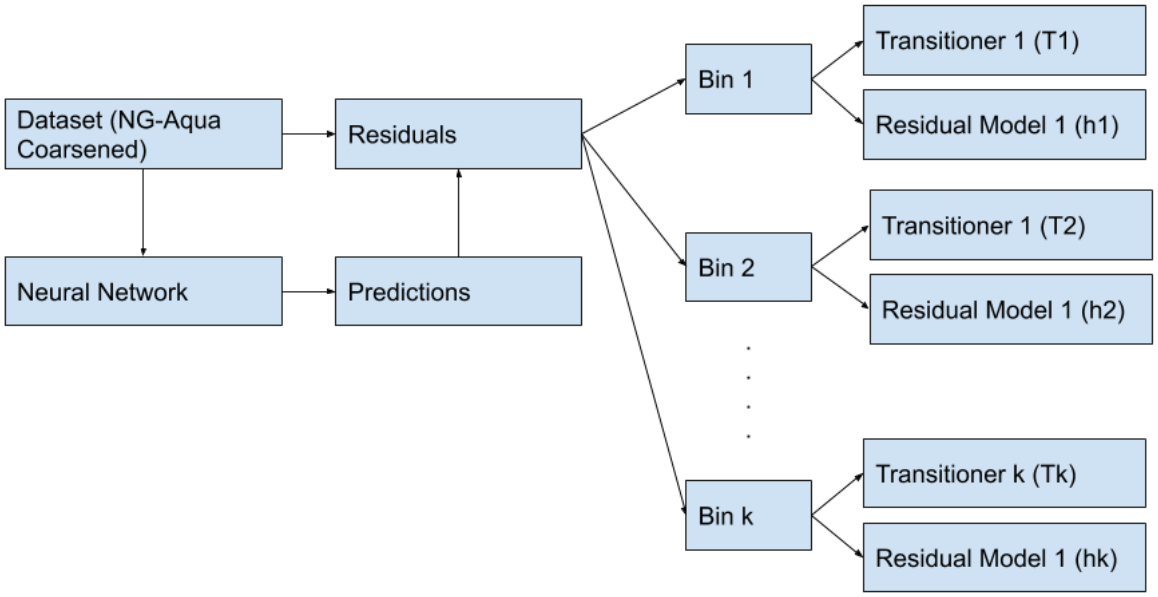


Figure 4.2: Process for Training the Stochastic Parameterization

Figure 4.2 shows the process for training the stochastic parameterization. The dataset is fed through the neural network to generate predictions. Then the residuals of the neural network are calculated by comparing the neural network output to the observed parameterizations in the dataset. Each data point (a specific column and time) is then assigned an

$\eta$ bin based on its net precip. residual, according to the bins defined in Figure 4.1. Next, for each bin all of the observations are collected and a transitioner model is trained that models the transition probabilities from the subject bins to all of the other bins. Separately, a residual model $h$ is trained for each bin, modeling the 68-dimensional residual.

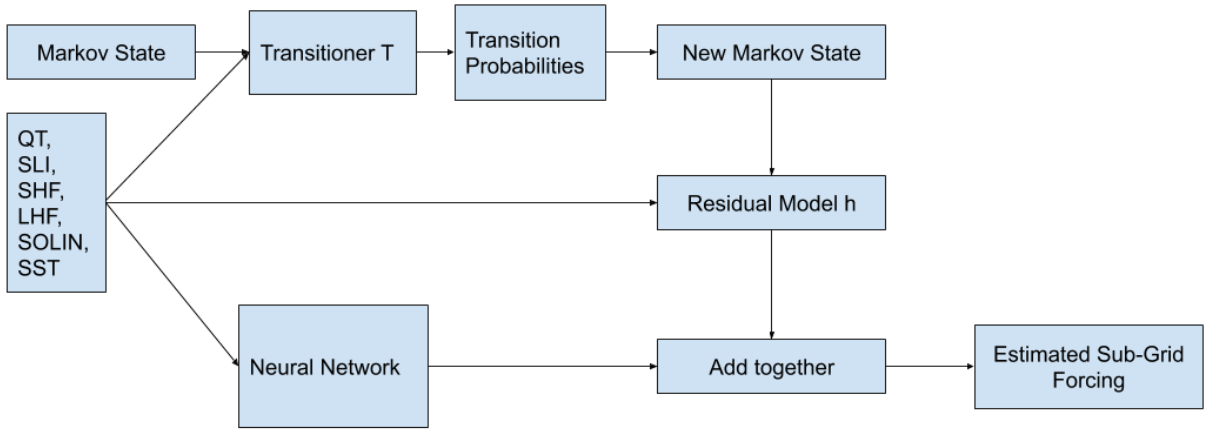


Figure 4.3: Stochastic Parameterization Process

Figure 4.3 shows how stochastic neural network parameterizations are produced for a single-column for a single time point. The existing Markov state $\eta$, as well as the current state of the model QT and SLI, and other variables SHF, LHF, SOLIN, and SST are fed through the transitioner model. The transitioner model outputs transition probabilities, and a new Markov state is randomly chosen based on these transition probabilities. This defines which residual model $h$ is used, and the state and other variables are fed through this residual model. The output of the residual model is added to the output of the neural network to produce an estimated parameterization.

### *4.9 Qualitative Motivation for Approach*

As described in Chapter 3, the goal for stochastic parameterization is to move from producing the expected value of the probability distribution to drawing from the full probability distribution. This Markov chain method approximates the probability distribution by producing a seven-state probability mass function, with probabilities approximately equal to the integrated true pdf nearby the seven values. The probabilities of each output are modeled from the training data with a logistic regression model, such that the probabilities are an estimate of the seven discrete options. Given infinite computing power, as the amount of training data increases and the number of Markov states is increased, the methods described here will approach the true probability distribution. A few of the assumptions, such as the Markov assumption, and off-zero diagonal covariance matrix are not completely true. Subsequent Markov states depend slightly on states prior to the current state, and $\tilde{\epsilon}_t$ has a nonzero off-diagonal covariance. However, these are not strong assumptions so the approach can produce a sound estimate of the probabilities of the seven bins.

# Chapter 5

# RESULTS

## *5.1 Evaluation Methods*

Multiple methods are used to evaluate the stochastic parameterization. First, the thirty days of data is run through the neural network and formatted into training data for the transitioner models and residual models as detailed in Figure 4.3. Thirty days of training data are used as this appears to be enough that the training and validation errors are close, but small enough such that the models train in a reasonable amount of time. The data from these thirty days is randomly broken up into 90% training data and 10% validation data. The models are fit to the training data, and then their performance is evaluated on the training data and then validation data to obtain an estimate of population fit. These performance metrics are reported in Tables 5.2 and 5.3. This evaluation method is referred to "model fit evaluation", as its metrics describe how well the models fit the data.

The model fit evaluation does not describe how the parameterization behaves when the Markov states are updated stochastically. The "stochastic bin-transition evaluation" method is intended to test the stochastic update process. This occurs on the training dataset without the use of a GCM. The Markov states are initialized according to the first time step in the training dataset. Then, the Markov states are stochastically updated for each column, and the stochastic parameterization output is produced based on the method detailed in Figure 4.3. The inputs to the neural network, residual models, and transitioners come from the training dataset at each time step. Unlike the single-column model validation setting, the predicted state variables QT and SLI are not used as inputs to the next time step; instead, the state variables come from the training dataset every time. This process is run for ten days.

Lastly, the stochastic parameterization is evaluated in the "coupled-simulation evaluation" setting, in which the stochastic neural-network method is used as the parameterization in a GCM simulation. The simulation is initialized using the first time step of the training data and then run for ten days. The results of the simulation are compared to the training data. This method is the ultimate test of how well the parameterization works because the final goal of the parameterization is to improve the accuracy of a GCM simulation.

An overview of the evaluation methods is given in Table 5.1. Analysis of the results of each evaluation method are presented in the sections below.

Table 5.1: Evaluation Methods

| **Evaluation Method** | **Metrics** | **Stochastic Transitions** | **GCM** |
|---|---|---|---|
| Model Fit | $r^2$, Accuracy, Log-Loss | No | No |
| Stochastic Bin-Transition | KS-Divergence | Yes | No |
| Coupled-Simulation | PW & Net Precip. Bias | Yes | Yes |

## *5.2 Model Fit Evaluation*

Table 5.2: Residual Model Performance

| **Bin** | **Train $r^2$** | **Validation $r^2$** |
|---|---|---|
| 0 | 0.24 | 0.23 |
| 1 | 0.14 | 0.14 |
| 2 | 0.09 | 0.09 |
| 3 | 0.08 | 0.08 |
| 4 | 0.09 | 0.09 |
| 5 | 0.13 | 0.13 |
| 6 | 0.18 | 0.18 |

Table 5.2 shows the mass-weighted coefficient of determination, $r^2$, on training and validation data for the residual models. The average $r^2$ is around 0.15 meaning that given an $\eta$ representing a net precip. residual range, the model is able to predict 15% of the variance of residual in that bin on average. This is somewhat low because seven bins are used: there is only a small amount of variance to be explained within each bin. As shown in Figure 4.1, the outer bins (bins 0 and 6) have a larger variance in net precip. residuals than the middle bins (bins 2-4). Table 5.2 shows that the residual models are able to explain more variance for these outer bins that have higher variance, with a validation $r^2$ of 0.24 for bin 0 and 0.18 for bin 6, than the middle bins which have validation $r^2$s of 0.08 and 0.09. Thus, the low $r^2$s are a reflection of the method for choosing bins: minimizing the within-bin variance of residuals. The equality of the train and validation $r^2$s reflects the large amount of training data available compared to the number of parameters that need to be estimated for the linear regression models.

Table 5.3: Transitioner Performance

| **Bin** | **Train Accuracy** | **Val Accuracy** | **Train Log Loss** | **Val Log Loss** |
|---|---|---|---|---|
| 0 | 43% | 41% | 1.45 | 1.46 |
| 1 | 36% | 35% | 1.64 | 1.65 |
| 2 | 40% | 40% | 1.37 | 1.37 |
| 3 | 45% | 44% | 1.24 | 1.25 |
| 4 | 37% | 36% | 1.42 | 1.42 |
| 5 | 35% | 35% | 1.43 | 1.44 |
| 6 | 40% | 39% | 1.57 | 1.57 |

Table 5.3 shows the train and validation accuracy and log loss of the transitioner models. The log loss reported in this table is the negative log likelihood of the predicted class probabilities averaged over the seven bins. The accuracy is the percentage of the time that bin with the highest predicted transition probability is the same bin to which a transition

is observed. The validation accuracy and log loss are nearly identical to the train accuracy and log loss because there is ample training data and logistic regression is a relatively high bias model. The accuracies are substantially better than a baseline model that predicts the highest probability bin every time, which is 25% accurate. The log losses are substantially better than a baseline model that outputs the prior probabilities each time, which has log losses of 3.1, 2.1, 1.5, 1.4, 1.5, 1.8, and 2.3 for bin 0 through 6, respectively.

The goal of the transitioner is not to achieve as high of an accuracy, or as low of a log loss, as possible. If the accuracy is 100% and log loss is 0, then there is no stochasticity and the transitions would be deterministic. This thesis assumes that the probability of transitioning from the current residual range to another is a stochastic process for a given transitioner model input. As such, the accuracy of a transitioner model should not be close to 100% and the log loss should not be close to zero. Under this assumption of stochasticity, the observed transitions in the training dataset are not always to the bin with the highest true transition probability, so the definition of accuracy used for Table 5.3 is an insufficient measure of transitioner model performance; additional evaluation methods are necessary.

### *5.3 Stochastic Bin Transition Evaluation*

To evaluate the residual models and transitioner models holistically, the stochastic bin-transition evaluation method is used.

Figure 5.1 shows the distribution of outputs in the stochastic bin-transition setting. It shows that the stochastic parameterization produces a distribution of outputs that is much closer to the parameterization distribution in the training dataset than the deterministic neural network approach.

Table 5.4 shows the Kolmorogov-Smirnov divergence comparison between the parameterization distribution in the training dataset and that produced by the stochastic and non-stochastic parameterizations in the stochastic bin-transition setting. The distributions that Table 5.4 compares are shown in Figure 5.1. The stochastic parameterization has a KS-divergence of 0.01 for the distribution of net precip. outputs, meaning the furthest

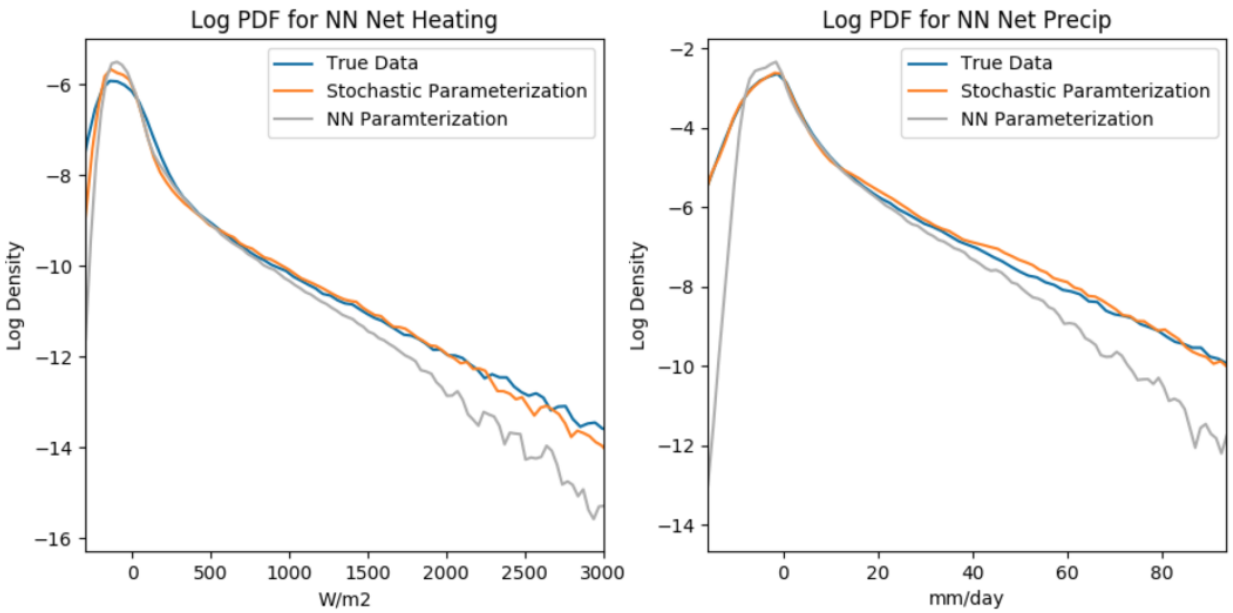


Figure 5.1: Stochastic Parameterization Distribution Comparison

Table 5.4: Stochastic Bin-Transition Kolmorogov-Smirnov Divergence

| **Parameterization** | **Net Heating KS-Diverg.** | **Net Precip KS-Diverg.** |
|---|---|---|
| Non-Stochastic | 0.14 | 0.11 |
| Stochastic | 0.07 | 0.01 |

distance between the pdfs of the stochastic net precip. outputs and the true net precip. outputs is 0.01. Comparatively, the non-stochastic neural network parameterization has a KS-Divergence of 0.11; the stochastic approach provides a ten-fold improvement.

The KS-Divergence improvement for net heating from the non-stochastic model to the stochastic model is 0.14 to 0.07, which is not as substantial as the net precip. improvement. This is because the bins are based on the net precip. residual, not the net heating residual. If the binning method is switched to use net heating residual, the KS-Divergence improvement for net heating and net precip. flips. The stochastic model described in this thesis bins on net precip. residual because the physical motivation is based on the fact that the lack of net

precip. variance produced by the parameterization as shown in Figure 3.5 is expected to be the source of the precipitation bias observed in the simulation shown in Figures 3.6 and 3.7.

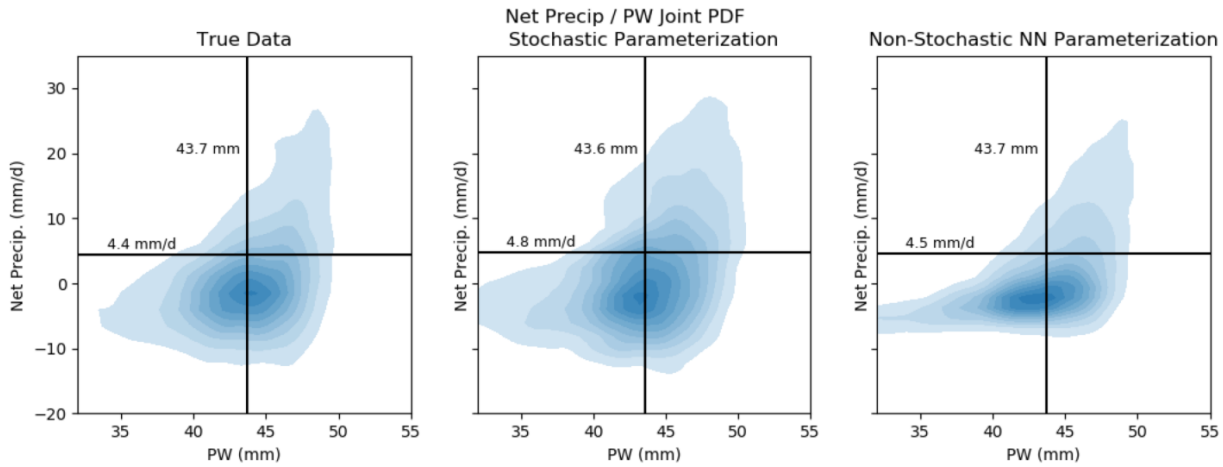


Figure 5.2: Stochastic Parameterization Joint PW / Net Precip. PDF Comparison

Figure 5.2 shows the net precip. / precipitable water joint pdfs for the training data, the stochastic parameterization, and the deterministic parameterization. The lack of variance produced by the deterministic parameterization shown in the right panel is a major motivation for this thesis, as discussed in Section 3.3. The middle panel shows the joint pdf for the stochastic parameterization. For a given value of precipitable water, the stochastic parameterization produces a distribution of net precip. that is close to that of the observed parameterizations, and that has a larger variance than the non-stochastic neural network parameterization. If the precipitation and precipitable water bias observed in the simulation, shown in Figures 3.6 and 3.7, is due to the parameterization not producing enough variance as described in the motivation chapter, then a simulation with the stochastic parameterization should not be significantly biased.

The shape of the joint pdf of the stochastic parameterization in Figure 5.2 is slightly distorted compared to the training dataset distribution in the left panel. However, for each value of precipitable water, the conditional pdf of net precip. is roughly the same width as in

the left panel, compared to the non-stochastic parameterization where the net precip. pdf is uniformly more concentrated towards the middle of the conditional distribution than the left panel. Thus, this stochastic parameterization can test the hypothesis that the problem of precipitation and precipitable water bias in a simulation is due to an improper representation of the probability distribution of the parameterization.

### *5.4 Coupled-Simulation Evaluation*

A simulation with the System for Atmospheric Modeling in the NG-Aqua setting with 160-km grid boxes is run with the stochastic parameterization developed in this thesis.

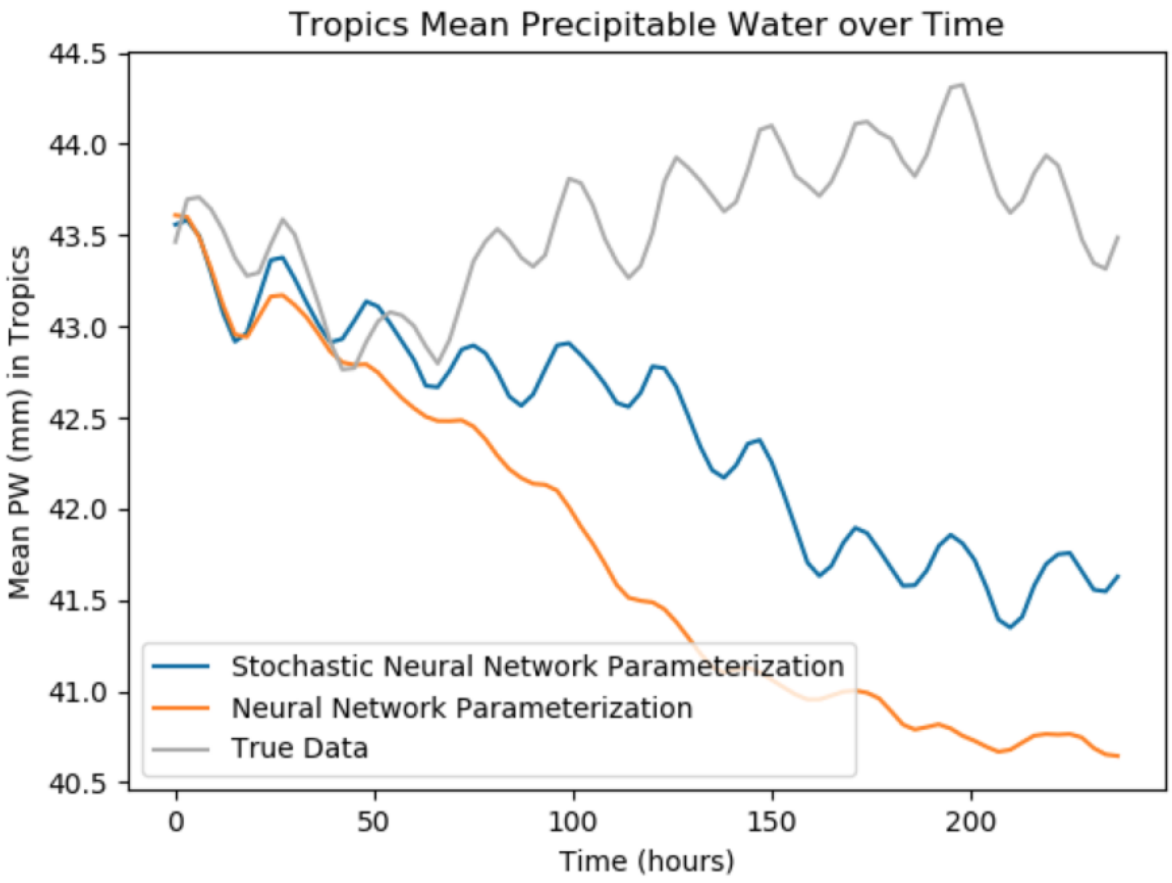


Figure 5.3: Mean Precipitable Water in Tropics over the Course of a Simulation

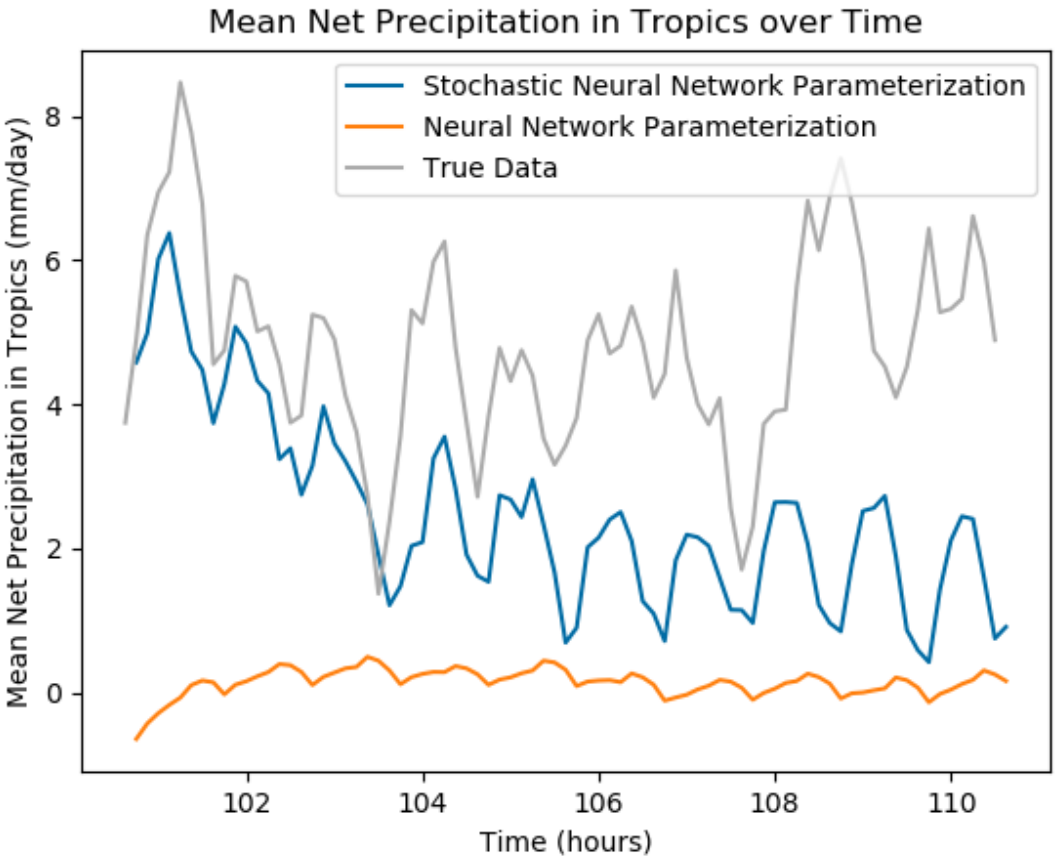


Figure 5.4: Stochastic Neural Network Net Precip. over Time in Coupled-Simulation. When a simulation is run with the stochastic neural network parameterization, the net precip. is well below that of the true data.

As shown in Figure 5.3 and Figure 5.4, the mean state bias problems observed in the simulation with the neural-network parameterization persist. The precipitable water still decreases over the course of the simulation and the net precip. remains consistently below that in the training data. Both Figures 5.3 and 5.4 seem to show that the stochastic parameterization provides some improvement over the neural-network approach. However, this improvement fluctuates significantly when the simulation is repeated and when the neural network is retrained with different initializations, so it is unclear whether the apparent improvement of the stochastic approach is significant. The coupled-simulation suffers from the same issues as the non-stochastic coupled-simulation. Both precipitable water and net precip. remain significantly biased, and the Hadley circulation dies out and disappears over the ten day simulation. Even if the stochastic model provides some improvement over the non-stochastic approach, it is not enough to resolve the most pronounced issues of the simulation.

Despite these negative results, the coupled-simulation with stochastic parameterization is stable. Many issues were encountered over the course of this project with the simulation blowing up, where the state variables increase towards infinity until the simulation crashes (Brenowitz and Bretherton, 2019). Given these troubles, the fact that the final parameterization described in this paper does not produce outputs that are increasingly large in the coupled-simulation setting is an accomplishment. The method is also demonstrably efficient, as it adds a negligible amount of time to a coupled-simulation compared to a simulation with the non-stochastic neural network.

### *5.5 Other Problems with the Simulation*

There are a couple of known issues with the SAM simulation that may be the root source of the persistent precipitation and precipitable water bias. SAM uses a process called "hyper-diffusion" in which after each update of the simulation, the value of variables for adjacent grid cells are smoothed. Without hyper-diffusion, small numerical instabilities can cause oscillation that propagates, resulting in numerical blow-up. The outputs of simulations

without hyper-diffusion look oscillatory, such that adjacent grid cells have large differences in state.

While hyper-diffusion is necessary for the simulation, it ends up cancelling out some of the effects of stochastic parameterizations. At each time step the stochastic parameterization produces a variety of outputs for nearby grid cells that have similar inputs; then the hyper-diffusion averages these grid cells to make their parameterization similar, effectively cancelling out the stochasticity. The stochastic parameterization is local horizontally so it cannot possibly negate hyper-diffusion which is non-local.

The second known issue with this simulation is that the System for Atmospheric Modeling is not designed to be used in the coarse-grid setting. It is designed for high resolution cloud simulations to study cloud formation processes (Khairoutdinov and Randall, 2003), so it should not be expected to work in the coarse resolution setting. SAM solves a simplified version of the Navier-Stokes equations and maintains assumptions that work well in the high resolution setting but have not been well tested in the coarse-grid setting. For example, coarse-resolution SAM does not model the unresolved source of momentum realistically. Though it was too difficult to implement for this thesis, if well-established parameterizations such as the CAM parameterization which the neural network was initially evaluated against (Brenowitz and Bretherton, 2018), is used in a 160-km grid cell SAM simulation, similar problems of PW and net precip. bias would likely occur. This is because these problems may have more to do with hyper-diffusion and the use of SAM with such a coarse resolution than an improper representation of stochasticity by the parameterization.

### *5.6 Further Research*

The promising results in the offline stochastic bin-transition setting shown in Figures 5.1 and 5.2 call for further evaluation of the methods developed in this thesis. The stochastic machine-learned parameterization should be implemented with other GCMs that are intended for coarse-grid use. Simulations with the stochastic parameterization, the deterministic machine-learning parameterization, and established parameterizations should be

compared.

# Chapter 6

# CONCLUSION

A new approach for introducing stochasticity into a machine-learned parameterization in order to produce a desired distribution is presented in this thesis. The approach consists of a Markov chain model that stochastically switches between states that represent residual ranges of the machine-learned model, coupled with a residual model that predicts the full residual from the Markov state and climate model variables. The approach is motivated by problems with a climate model simulation that is run with a deterministic neural-network parameterization. The simulation does not produce enough precipitation is the tropics, resulting in the loss of the Hadley circulation, a persistent wind circulation between the tropics and subtropics. Theoretically, the lack of precipitation can be caused by a parameterization that does correctly replicate the probability distribution of possible parameterizations.

When evaluated offline, the approach produces the correct distribution of parameterizations. As such, it is useful for evaluating the hypothesis that observed mean-state bias in precipitation in the deterministic neural-network climate model simulations is due to an improper representation of the probability distribution of parameterizations.

When evaluated in a climate model simulation, the stochastic parameterization fails to improve these mean state bias issues. The persisting mean-state bias issues are probably not due to problems with the stochastic methodology, but a result of a hyper-diffusion scheme in the climate model which averages nearby grid cells, cancelling out stochastic effects, and the fact that the climate model used for the simulation is not intended for the coarse-grid setting.

The ability of the stochastic approach to correct the distribution of outputs of the deterministic machine learning parameterization, the simplicity of the method, and its stability

in a coupled-simulation calls for further evaluation with a climate model that is suited to coarse-grid simulations. The stochastic parameterization can be built on top of any machine learning parameterization, so hopefully the methods developed in this thesis will be tested by climate scientists developing machine learning parameterizations, and this thesis will contribute to our ability to predict the weather and climate patterns more accurately and further into the future.

# BIBLIOGRAPHY

Judith Berner, Ulrich Achatz, Lauriane Batt, Lisa Bengtsson, Alvaro de la Cmara, Hannah M. Christensen, Matteo Colangeli, Danielle R. B. Coleman, Daan Crommelin, Stamen I. Dolaptchiev, Christian L. E. Franzke, Petra Friederichs, Peter Imkeller, Heikki Jrvinen, Stephan Juricke, Vassili Kitsios, Franois Lott, Valerio Lucarini, Salil Mahajan, Timothy N. Palmer, Ccile Penland, Mirjana Sakradzija, Jin-Song von Storch, Antje Weisheimer, Michael Weniger, Paul D. Williams, and Jun-Ichi Yano. Stochastic parameterization: Toward a new view of weather and climate models. *Bulletin of the American Meteorological Society*, 98(3):565–588, 2017. doi: 10.1175/BAMS-D-15-00268.1. URL https://doi.org/10.1175/BAMS-D-15-00268.1.

Noah D Brenowitz and Christopher S Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.

Noah D. Brenowitz and Christopher S. Bretherton. Spatially extended tests of a neural network parametrization trained by coarsegraining. *Journal of Advances in Modeling Earth Systems*, 2019. doi: 10.1029/2019ms001711.

Christopher S Bretherton and Marat F Khairoutdinov. Convective self-aggregation feedbacks in near-global cloud-resolving simulations of an aquaplanet. *Journal of Advances in Modeling Earth Systems*, 7(4):1765–1787, 2015.

Christopher S. Bretherton, Matthew E. Peters, and Larissa E. Back. Relationships between water vapor path and precipitation over the tropical oceans. *Journal of Climate*, 17(7):1517–1528, 2004. doi: 10.1175/1520-0442(2004)017¡1517:RBWVPA¿2.0.CO;2. URL https://doi.org/10.1175/1520-0442(2004)017<1517:RBWVPA>2.0.CO;2.

Roberto Buizza, M Milleer, and Tim N Palmer. Stochastic representation of model uncertainties in the ecmwf ensemble prediction system. *Quarterly Journal of the Royal Meteorological Society*, 125(560):2887–2908, 1999.

Daan Crommelin and Eric Vanden-Eijnden. Subgrid-scale parameterization with conditional markov chains. *Journal of the Atmospheric Sciences*, 65(8):2661–2675, 2008. doi: 10.1175/2008JAS2566.1. URL https://doi.org/10.1175/2008JAS2566.1.

BB Goswami, B Khouider, R Phani, P Mukhopadhyay, and AJ Majda. Implementation and calibration of a stochastic multicloud convective parameterization in the ncep c limate f orecast s ystem (cfsv2). *Journal of Advances in Modeling Earth Systems*, 9(3):1721–1739, 2017.

Yen-Ting Hwang and Dargan MW Frierson. Link between the double-intertropical convergence zone problem and cloud biases over the southern ocean. *Proceedings of the National Academy of Sciences*, 110(13):4935–4940, 2013.

Xianan Jiang. Key processes for the eastward propagation of the madden-julian oscillation based on multimodel simulations. *Journal of Geophysical Research: Atmospheres*, 122(2): 755–770, 2017.

Marat F. Khairoutdinov and David A. Randall. Cloud resolving modeling of the arm summer 1997 iop: Model formulation, results, uncertainties, and sensitivities. *Journal of the Atmospheric Sciences*, 60(4):607–625, 2003. doi: 10.1175/1520-0469(2003)060¡0607:CRMOTA¿2.0.CO;2. URL https://doi.org/10.1175/1520-0469(2003)060<0607:CRMOTA>2.0.CO;2.

Boualem Khouider, Joseph Biello, Andrew J Majda, et al. A stochastic multicloud model for tropical convection. *Communications in Mathematical Sciences*, 8(1):187–216, 2010.

Vladimir M Krasnopolsky, Michael S Fox-Rabinovitz, and Dmitry V Chalikov. New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation

of longwave radiation in a climate model. *Monthly Weather Review*, 133(5):1370–1383, 2005.

Andrew J Majda. Multiscale models with moisture and systematic strategies for superparameterization. *Journal of the Atmospheric Sciences*, 64(7):2726–2734, 2007.

Andrew J Majda, Ilya Timofeyev, and Eric Vanden Eijnden. Models for stochastic climate prediction. *Proceedings of the National Academy of Sciences*, 96(26):14687–14691, 1999.

Andrew J Majda, Ilya Timofeyev, and Eric Vanden-Eijnden. Systematic strategies for stochastic mode reduction in climate. *Journal of the Atmospheric Sciences*, 60(14):1705–1722, 2003.

Brian Mapes and Richard Neale. Parameterizing convective organization to escape the entrainment dilemma. *Journal of Advances in Modeling Earth Systems*, 3(2), 2011.

Tetsuo Nakazawa. Tropical super clusters within intraseasonal variations over the western pacific. *Journal of the Meteorological Society of Japan. Ser. II*, 66(6):823–839, 1988.

Paul A O'Gorman and John G Dwyer. Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, 10(10):2548–2563, 2018.

T. N. Palmer. A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models. *Quarterly Journal of the Royal Meteorological Society*, 127(572):279–304, 2001. doi: 10.1002/qj.49712757202. URL `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49712757202`.

Stephan Rasp, Michael S Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

SS Rushley, Daehyun Kim, CS Bretherton, and M-S Ahn. Reexamining the nonlinear moisture-precipitation relationship over the tropical oceans. *Geophysical research letters*, 45(2):1133–1140, 2018.

Glenn Shutts. A kinetic energy backscatter algorithm for use in ensemble prediction systems. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(612):3079–3102, 2005.

RA Stratton and AJ Stirling. Improving the diurnal cycle of convection in gcms. *Quarterly Journal of the Royal Meteorological Society*, 138(666):1121–1134, 2012.

Daniel S Wilks. Effects of stochastic parametrizations in the lorenz'96 system. *Quarterly Journal of the Royal Meteorological Society*, 131(606):389–407, 2005.

MD Woelfle, S Yu, CS Bretherton, and MS Pritchard. Sensitivity of coupled tropical pacific model biases to convective parameterization in cesm1. *Journal of Advances in Modeling Earth Systems*, 10(1):126–144, 2018.