# The Intersection of AI and Climate Modeling: Emulators for Physical Parameterizations and Frameworks for Testing AI Weather Models

by

Garrett C. Limon

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Climate and Space Sciences and Engineering)
in the University of Michigan
2025

Doctoral Committee:

Professor Christiane Jablonowski, Chair
Professor Mark Flanner
Professor Mohammed Ombadi
Professor Alexander Rodríguez

Garrett C. Limon
glimon@umich.edu
ORCID iD: 0000-0002-6504-7710

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Christiane Jablonowski, for their guidance, patience, and support throughout the course of my studies. Their mentorship has been invaluable in shaping both my research and my development as a scientist. Similarly, I greatly appreciate the guidance of my previous mentor, Tyler Luchko, for getting me ready for my doctoral studies. I am also grateful to the members of my dissertation committee, Mark Flanner, Mohammed Ombadi, and Alexander Rodríguez, for their thoughtful feedback, encouragement, and support, all of which strengthened this work. Special thanks go to my collaborators and colleagues, particularly William Chapman and Joshua Elms, whose ideas, discussions, and assistance were essential to the progress of this research. I would also like to acknowledge the support of the NSF GRFP, without which this work and my enrollment here at the University of Michigan would not have been possible.

Finally, I am deeply thankful to my family and friends for their unwavering encouragement, patience, and love throughout this process. Their support sustained me through many personal challenges and allowed me to make it to the end of this journey. Mom, Dad, Allison, Lisa. Korona, Arbor. Liz, Blake, Natasha, Gopal, Ali, Alan, Zeke, Laren, Roderick, Maya, Erica, Mason, Tanner... And so many other loved ones, thank you.

And to the person who set me on this academic journey, I did it.

# TABLE OF CONTENTS

# LIST OF FIGURES

FIGURE

# LIST OF TABLES

TABLE

# LIST OF ACRONYMS

**AI** Artificial Intelligence

**AI2ES** Artificial Intelligence for Environmental Sciences

**CAM** Community Atmosphere Model

**CESM** Community Earth System Model

**CMIP** Climate Model Intercomparison Project

**ECMWF** European Centre for Medium-Range Weather Forecasts

**ENSO** El Niño Southern Oscillation

**ERA5** ECMWF Reanalysis version 5

**ESM** Earth System Model

**FKB** Fortran-Keras-Bridge

**GCM** General Circulation Model

**IPCC** Intergovernmental Panel on Climate Change

**ML** Machine Learning

**NCAR** National Center for Atmospheric Research

**NN** Neural Network

**NWP** Numerical Weather Prediction

**ONNX** Open Neural Network Exchange

**QBO** Quasi-Biennial Oscillation

**RF** Random Forest

**TR** Tendency Reversion

# ABSTRACT

The increasing intersection of Machine Learning (ML) and Earth system modeling presents both transformative opportunities and fundamental challenges. This dissertation examines the integration of ML techniques across three critical fronts in climate and weather modeling: emulating physical parameterizations, coupling ML components into traditional general circulation models, and probing dynamical responses in fully data-driven forecasting systems.

In the first part, we evaluate the skill of Random Forests (RFs) and Neural Network (NN) emulators trained on a hierarchy of physics configurations within the Community Atmosphere Model (CAM). The latter is the atmospheric component of the Community Earth System Model from the National Center of Atmospheric Research. Results show that while RFs can accurately reproduce tendencies and precipitation fields in simplified settings, their performance degrades as complexity increases. We further demonstrate that by incorporating domain knowledge through targeted feature selection, we can improve emulator skill. This work underscores the importance of offline benchmarking and highlights the limitations of tree-based methods in these highly nonlinear regimes, while maintaining skill relative to simple NN benchmarks.

The second part transitions from offline evaluation to online deployment, embedding both RF and NN emulators into CAM. This stage reveals practical challenges in real-time ML–physics coupling, including NN instability in particularly active regions, RF memory constraints, and interface incompatibilities between ML frameworks and CAM's highly optimized Fortran infrastructure. These findings show that successful ML integration must address a variety of both scientific and software engineering constraints.

The final project peers into the evaluation of next-generation AI-based weather forecasting systems, focusing primarily on Google's GraphCast model. We extend the tendency reversion diagnostic framework to GraphCast's 37-level model to assess dynamical responses to perturbations such as tropical heating anomalies. Although architectural constraints may limit full tendency reversion applicability, this work demonstrates how idealized dynamical tests can be adapted for various data-driven forecasting systems, offering a complementary perspective to traditional score-based metrics. In applying this framework, we also find that the 37-level GraphCast exhibits substantially improved representation of expected wave re-

sponses to our perturbed input compared to the 13-level operational version.

Together, these studies chart a path toward responsible development, deployment, and diagnosis of ML-based Earth system models. They point to the need for architecture-aware diagnostics, scalable coupling tools, and hybrid modeling paradigms that combine data-driven flexibility with physical interpretability. As ML continues to reshape the modeling landscape, this work argues for rigorous, domain-informed testing and integration strategies that bridge modern computational approaches with the foundational principles of atmospheric science.

# CHAPTER 1

# Introduction

## 1.1 Understanding the Earth System

The Earth system is characterized by a series of interconnected processes that regulate the planet's atmospheric, geologic, hydrologic, and biologic phenomena [Flato, 2011]. Grasping the intricacies of these interrelated aspects is crucial for projecting and preparing for future scenarios, especially in the context of a changing climate. Extensive research has been conducted on various components of the Earth system over many decades, yielding critical insights into phenomena related to climate change. This includes understanding the greenhouse effect and its repercussions on the stratospheric ozone layer, cryospheric changes, and sea level rise, among other issues [Meehl et al., 2007].

Recent advances in technology have enhanced our capacity to simulate the Earth system, allowing for more precise projections of these climatological impacts at multiple scales. An Earth System Model (ESM), also commonly known as a climate model, strives to depict each component of the Earth system and their interactions, thus producing an integrated depiction of the Earth system. This typically involves the integration of models that encapsulate the distinct processes of the Earth system, such as atmospheric and oceanic dynamics, alongside terrestrial and hydrologic models, each commonly operating on their own unique temporal and spatial scales. Although every element of the Earth system contributes to its overall representation, the atmosphere is particularly critical due to its extensive interactions with all other components. Consequently, accurate atmospheric modeling is imperative for the development of a comprehensive ESM.

### 1.1.1 Weather versus Climate

To more thoroughly examine atmospheric modeling, it is imperative to distinguish between the concepts of 'weather' and 'climate,' which are often conflated in discussions surrounding atmospheric science, particularly in terms of modeling and prediction. Weather represents

an instantaneous manifestation of atmospheric conditions at local-to-regional scales, characterized by state variables such as temperature, wind speed, precipitation, and humidity. However, weather is not merely a collection of these variables at a given moment; it also reflects the evolving interplay of atmospheric dynamics that gives rise to transient patterns and phenomena. This includes routine fluctuations and the development of significant meteorological events, such as thunderstorms, hurricanes, wildfires, hail, and various storm systems, which emerge from complex interactions between atmospheric dynamics, moisture availability, and large-scale circulation patterns.

In contrast to weather, climate denotes the long-term trends and conditions across the Earth system. It encompasses not only the atmosphere but also interactions with the ocean, cryosphere, and land surface, which become increasingly significant over longer timescales. The traditional climatological period for the assessment spans thirty years, although this standard has recently been subject to scrutiny as it is quite an arbitrary timescale [Meehl et al., 2007]. What remains consistent is the recognition that the climate system is conceptualized on a regional-to-global scale, as opposed to the local-to-regional emphasis of weather. Thus, rather than representing an instantaneous state, the climate is understood through sustained observations of atmospheric conditions and trends over extended time periods.

Climatological research is instrumental in identifying and analyzing multiple interconnected indicators, commonly referred to as teleconnections, within Earth system data. These large-scale climate patterns influence atmospheric variability over extended timescales and regions, offering predictive insights into weather and climate dynamics. For instance, phenomena such as the El Niño Southern Oscillation (ENSO) and the Quasi-Biennial Oscillation (QBO) illustrate the complex interplay of global climatic factors. Ma et al. [2023] emphasize the nonlinear characteristics of these teleconnections, highlighting how Machine Learning (ML) techniques can improve the detection and prediction of their influences on regional climate variability. Such indicators are valuable for forecasting seasonal weather patterns, as they modulate temperature, precipitation, and atmospheric circulation across multiple regions. For example, Dai and Wigley [2000] demonstrate that a robust ENSO phase can significantly alter precipitation patterns, particularly in the western United States, where El Niño events are often associated with increased rainfall, while La Niña conditions tend to promote drier conditions. By integrating advanced analytical techniques with climatological research, these studies enhance our ability to predict and understand how global-scale oscillations drive regional weather anomalies.

## 1.2 Climate Modeling and Weather Prediction

Vilhelm Bjerknes was the pioneer in positing the feasibility of forecasting weather by specifying a series of non-linear partial differential equations, specifically the Navier-Stokes equations and state equations applicable to atmospheric processes [Bjerknes, 1904, Washington and Parkinson, 2005, Platzman, 1967]. Subsequent to Bjerknes' work, Lewis Fry Richardson endeavored to predict changes in atmospheric pressure at a single point. Despite the inaccurate result, his approach laid the groundwork for contemporary climate modeling and numerical weather prediction techniques. Moreover, Richardson's practical application of forecast computation is often acknowledged as a seminal contribution to the parallel computing framework, known as the message-passing interface (MPI), within the high-performance computing domain.

In the mid-20th century, the advent of enhanced computational capabilities allowed for the development of early climate and weather models by various research groups. The inaugural regional weather prediction model was crafted by a team at the University of Stockholm in 1954 [Persson, 2005]. Concurrently, Princeton researchers formulated and executed the first General Circulation Model (GCM) simulations as well [Phillips, 1956]. This era marked a significant milestone, endowing the scientific community with computational instruments to interrogate meteorological and climatological queries. Within the succeeding decade, the establishment of the National Center for Atmospheric Research (NCAR), in Boulder, CO, USA, was witnessed, and the proliferation of weather and climate models ensued globally across leading universities and research institutions.

Given the shared origins of weather and climate models in their fluid dynamics foundations, it is important to reexamine their distinctions in the context of simulation. Distinguishing between weather prediction models and climate models can be predicated on the nature of the system each strives to solve. Weather prediction models are formulated as initial value problems, requiring input of the most accurate current weather conditions and propagated forward in time by the model. Due to the chaotic nature of weather systems, forecast accuracy decreases over time.

Climate models, in contrast, are less concerned with initial states, focusing instead on replicating climatological statistics and atmospheric flow over extended time periods. Numerically, climate simulation is akin to addressing a boundary value problem. An effective climate model should maintain consistent climatology and a stable atmospheric representation throughout the simulated time frame, despite potential variations in initial conditions. Commonly, climate models necessitate an initial 'spin-up' period to achieve their stable atmospheric general circulation [Ma et al., 2021]. Discerning these differences is vital for

pinpointing areas of model enhancement that may benefit weather forecasting, climate projection, or both. The principal objective of this thesis is to explore the application and refinement of an atmospheric GCM within the purview of climate simulation, rather than weather prediction.

## 1.3 The Atmosphere: A System of Scales

The atmosphere represents a highly complex and chaotic system characterized by dynamic behavior and intricate interconnections with various Earth system components. It encompasses numerous physical processes operating across diverse temporal and spatial scales. As an illustrative example, consider atmospheric radiative transfer, primarily dictated by the equilibrium between incoming solar radiation and outgoing radiation, which includes both surface fluxes and reflected solar radiation. Surface emissions of greenhouse gases add to the demonstration of the interactive linkages between atmospheric phenomena and the broader Earth system.

Focusing on atmospheric composition, nitrogen and oxygen constitute over 99 percent of atmospheric gas volume and do not directly influence radiative transfer. Conversely, trace greenhouse gases, encompassing carbon dioxide, methane, and other relevant species, though constituting less than one percent of the atmosphere, exert significant effects on the energy balance due to their interactions with radiative processes [Petty, 2006]. In addition, minor constituents such as aerosols, including sea salt and soot, serve as cloud condensation nuclei, facilitating cloud formation. Therefore, all of these particulates significantly impact energy balance, as clouds are central to the modulation of radiative scattering and absorption. Moreover, cloud dynamics are integrally connected to atmospheric general circulation, exhibiting development and transport patterns that span the globe.

Appreciating the multitude of scales and interrelations involved in representing solely the radiative budget reveals the overarching complexity inherent to atmospheric and Earth system modeling. From molecular-level abundances of greenhouse gases and aerosol particles, to cloud formation and dynamics, and overall general circulation patterns, a myriad of spatially extensive physical processes influence the radiative equilibrium and, consequently, the atmospheric state.

Furthermore, extending this analytic approach to other facets of atmospheric dynamics, such as the interplay between meridional temperature gradients and wind shear or other circulation drivers associated with extreme weather events, uncovers a similarly intricate web of physical interactions occurring across the various spatial and temporal scales. Thus, comprehensive understanding of process-level impacts and significance across the spectrum

of scales is imperative for researchers to consider when modeling the atmospheric system.

### 1.3.1 Standard Model Configuration

A GCM comprises two primary components: the dynamical core, which performs geophysical fluid dynamic calculations, and the physical parameterization schemes, which approximate subgrid-scale processes not explicitly resolved by the dynamical core [Washington and Parkinson, 2005, Jacobson, 2005, Randall et al., 2007]. Standard methodologies typically utilize a three-dimensional spherical grid as the foundational structure for solving the system of fluid equations. Various numerical techniques, each with their own advantages and limitations, may be employed for the dynamical core to simulate the geophysical flow of the atmosphere.

This work does not engage deeply with dynamical core development; for an in-depth analysis, the reader is referred to Washington and Parkinson [2005] and Jacobson [2005]. However, a critical consideration in climate model design is the spatial resolution of the dynamical core. Contemporary models often operate with a horizontal resolution of approximately 100 kilometers and employ between 30 and 60 vertical levels. However, state-of-the-art models have begun to reach resolutions nearing 3 km kilometers with up to 120 vertical levels [Brenowitz and Bretherton, 2019]. Temporal resolutions in operational scenarios vary, contingent on the horizontal resolution, computational resources, and the length of the simulation period [Randall et al., 2007, Hourdin and Armengaud, 1999]. Even with these high-resolution configurations, many atmospheric processes remain below the representational capacity of the model grid. Therefore, the necessity for parameterization schemes to estimate the unresolved physical processes becomes significant.

Reflecting upon the scale-dependent nature of atmospheric processes, parameterizations aim to incorporate as broad a spectrum of physical interactions as possible. Typical processes represented by these schemes include, but are not limited to, radiative transfer, convection, cloud microphysics, and turbulence. Parameterized processes are essential to simulate the climate system.

Moreover, the efficacy of climate models relies on the synchrony of these two components functioning collectively within the coupled system, thereby facilitating comprehensive investigations into the atmospheric system. Nonetheless, parameterization schemes serve as a primary source of model bias and uncertainty due to their heuristic construction and the diverse range of approaches employed across modeling platforms [Held and Suarez, 1994, Held, 2005, Stevens and Bony, 2013, Hourdin et al., 2017]. The complexity of such schemes and the consequent variability in tunable parameters, generally calibrated against observational

data, add significant, yet quantifiable error to GCM simulations. The cumulative effects of parameter tuning, along with the increased intricacy of additional coupled schemes, pose considerable challenges as models advance in complexity.

### 1.3.2 Model Development Workflow

The development of climate models proceeds through a well-established process. This typically begins with deterministic evaluations of the dynamical core through benchmark tests such as shallow-water test cases, which assess the horizontal and temporal discretization strategies applied in fluid dynamics solvers [Williamson et al., 1992]. Subsequently, the scope of testing broadens to encompass vertical discretization, either through two-dimensional frameworks that include a vertical axis or comprehensive three-dimensional dynamical core assessments. Specific examples of three-dimensional dynamical core tests include examinations of a dry atmosphere to scrutinize discretizations across all fundamental dimensions, and specialized dynamic tests such as the analytical initialization of tropical cyclone vortices [Held and Suarez, 1994, Reed and Jablonowski, 2011].

Following these initial dynamical core tests, simplified physical parameterizations are integrated to drive the model's atmosphere towards or away from an equilibrium state, fostering the development of quasi-realistic flow dynamics [Thatcher and Jablonowski, 2016]. This stage constitutes a significant portion of work presented in this dissertation, as it enables incremental and transparent modifications to the complexity of parameterization schemes. Proceeding from the 'minimal physics' frameworks, which offer various standard scenarios and combinations for modeling environmental phenomena, researchers tend to proceed to aquaplanet simulations [Neale and Hoskins, 2000]. These involve a comprehensive GCM configured for an entirely ocean-covered Earth analogue, typically with an assigned sea surface temperature distribution [Blackburn et al., 2013].

The final phase of advancement encompasses contemporary models, such as those participating in Atmospheric Model Intercomparison Projects and subsequently the Climate Model Intercomparison Project (CMIP) runs, wherein the atmospheric component is coupled to models of the other aspects of the Earth system for contributions to the Intergovernmental Panel on Climate Change (IPCC) reports [Meehl et al., 2007]. Models at this stage integrate advanced parameterization schemes at competitive resolutions and incorporating explicit topography. They are also the models that are incorporated within comprehensive ESMs, where the atmospheric component works alongside ocean, terrestrial, cryospheric, and biogeochemical models, thus simulating the Earth system. In this document, the term 'climate model' will pertain exclusively to atmospheric GCMs, rather than ESMs, unless explicitly

stated otherwise in the context of the text.

## 1.4 Potential for Artificial Intelligence

In recent years, advancements in Artificial Intelligence (AI) and ML have garnered significant interest, not only within our day-to-day life, but also throughout the scientific community. The proliferation of ML applications across diverse disciplines and sectors has led to its rapid adoption by researchers in atmospheric science. The integration of ML into this field has been unsurprising, considering the extensive datasets produced by research in atmospheric science, meteorology, and climate modeling and the fact that the efficacy of data-driven techniques hinges on the availability of substantial and high-quality datasets.

ML, a prominent sub-discipline of AI, has demonstrated its potential to propel advancements in various scientific arenas. Fundamentally, ML involves applying algorithmic strategies to detect patterns and infer functional relationships from data. Consequently, ML methodologies are increasingly being utilized in different branches of atmospheric science, facilitating novel insights from the discipline's expansive and evolving datasets, along with other strategies to leverage the power of ML to advance our knowledge. Section 1.4.2 provides a comprehensive exploration of the diverse applications of ML within this field.

### 1.4.1 Machine Learning

In the broad domain of ML, applications are typically classified into two primary categories: supervised learning and unsupervised learning, as illustrated in Figure 1.1. Unsupervised learning focuses on uncovering underlying patterns or structures within data without the need for labeled output. Common tasks in this category include clustering, where the goal is to group data points based on similarity. Clustering methods exemplify this by discerning data groupings and patterns, potentially revealing associations that may elude conventional analytical approaches [Ikotun et al., 2023]. For instance, clustering could categorize major league baseball players by team affiliation or performance metrics, potentially reconstructing team compositions from statistics alone. This highlights the potential of such algorithms to unveil latent connections within datasets, possibly beyond the reach of traditional data analysis techniques. In contrast, supervised learning involves the identification of correlations or functional relationships between labeled input-output pairs. This category encompasses two principal types of tasks: classification and regression, with the latter being particularly relevant for emulating physical parameterizations in atmospheric climate models, or predicting the entire atmospheric state based on the previous.

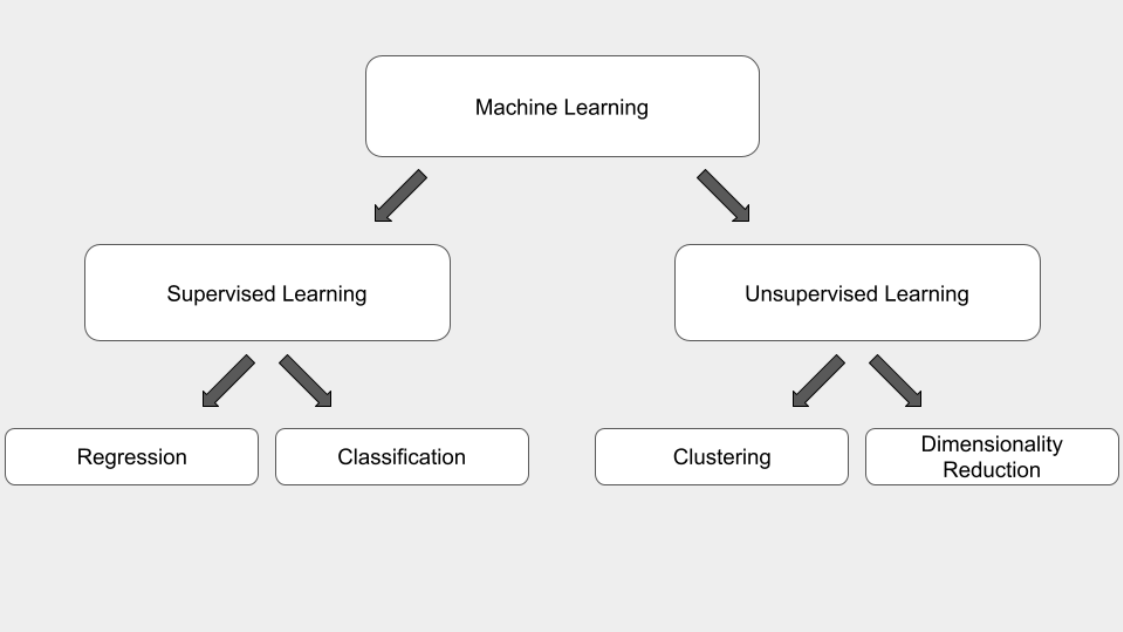



Figure 1.1: Diagram showing different types of common ML approaches.

Regression tasks in ML involve estimating a functional relationship between a dependent variable (the output or target) and one or more independent variables (the inputs or features). In ML terminology, these variables are often referred to as labels (dependent variables) and features (independent variables), respectively. Mathematically, regression aims to identify a function

$$\hat{g}(\vec{X}) \approx f(\vec{X}) \tag{1.1}$$

that approximates the relationship, where $f(\vec{X})$ represents the desired target function and $\vec{X}$ denotes the vector of input features. The objective of regression is to minimize the discrepancy between the prediction of the model, $\hat{g}(\vec{X})$, and the true output, $\vec{y} = f(\vec{X})$, from the data.

Modern ML techniques, such as Neural Networks (NNs) and Random Forests (RFs), stand out for their ability to handle complex non-linear relationships, which makes them particularly useful for non-linear regression tasks. In the context of physical parameterizations, a nonlinear regression model can be used to represent a physical tendency or forcing function, which describes how the dependent variable (such as a physical tendency) depends on the current state of the system (the independent variables, or state variables). Here, the function $f(\vec{x})$ represents the physical tendency and the vector $\vec{X}$ consists of the state variables that define the state of the system at a given time step. The trained ML model $\hat{g}(\vec{X})$, which is a learned approximation of the physical relationship, is then used to predict

the tendency based on the state variables at each time step.

Much of this work exploits the power of ML to engage with such nonlinear regression challenges. For parameterization emulation, we consider labeled datasets encompassing state variables from climate models (e.g., temperature, pressure, specific humidity) and the corresponding outputs from parameterization schemes (e.g., forcing tendencies, precipitation rates). ML techniques can be applied to replicate or augment the functional relationships between the parameterized outputs and the state inputs, which is particularly viable given the inherently nonlinear nature of parameterization schemes. This dissertation focuses on two principal nonlinear regression techniques derived from the ML field: RF and NN, elaborated upon further in the methodology section of Chapter 2.

#### 1.4.1.1 Random Forests

A RF is an ensemble learning method widely used for both classification and regression tasks. It combines multiple decision trees to improve predictive performance and robustness. As a supervised ML algorithm, RFs utilize the power of bootstrap aggregating and random feature selection to mitigate overfitting, reduce variance, and enhance prediction accuracy [Breiman, 1996]. Overfitting is the term used when your ML model performs well when applied to the training data, but poorly we applied to unseen data. RFs have become particularly popular due to its effectiveness, ease of use, and ability to handle complex datasets, even with high-dimensional feature spaces.

Each decision tree within a RF is trained on a random subset of training data, sampled with replacement. This introduces diversity among the trees in the forest, which helps to reduce the overall variance of the model compared to a single decision tree. Each tree in the forest makes an independent prediction, and the final output of the RF is determined by aggregating the predictions from all trees.

For classification, the most common class predicted by the individual trees is selected as the final class (majority voting). However, for regression the average of the predictions from all trees is used as the final output. In addition to random sampling of data, RFs introduce another layer of randomness during the training phase by selecting a random subset of features at each split when constructing each decision tree. This prevents individual trees from relying on the same features, which encourages diversity and further reduces the risk of overfitting. Another advantage of RFs is that they cannot extrapolate their predictions outside of the scope of the data they were trained with. This is beneficial in physical science applications, as it avoids artifacts that might be inconsistent with underlying physics or spurious error growth. For instance, an RF will inherently comply with the non-negative property of precipitation, as it will not have encountered negative precipitation values during

training. This contrasts with techniques like NNs, which can struggle with extrapolation and adherence to underlying physical constraints [Beucler et al., 2021].

Although RFs offer many advantages, they are not without limitations. While individual decision trees are easy to interpret, the ensemble nature of RFs makes it difficult to visualize the decision-making process of the entire model. This can hinder the interpretability, especially within the domains of physical sciences. RFs can also be computationally intensive, particularly for large datasets with a high number of trees and deep branch structures, which can result in longer training times and increased memory usage. While they are not immune to challenges such as reduced interpretability and high computational cost, ongoing developments in ensemble methods and model explainability continue to extend their applicability and usefulness in a wide range of fields, including the atmospheric sciences [O'Gorman and Dwyer, 2018, Connelly and Gerber, 2024].

#### 1.4.1.2 Neural Networks

NNs are a class of computational models inspired by a simplified interpretation of the structure and way the human brain processes information, learning from data to perform tasks such as classification, regression, and pattern recognition. The foundation of NNs lies in their architecture, which consists of interconnected nodes or 'neurons' organized into layers. These layers process input data in a manner that allows the network to learn complex mappings between inputs and outputs [Baldi, 2021, Reichstein et al., 2019].

A typical feed-forwardNN typically consists of three primary layers: Input Layer: The input layer receives the raw data, such as images, numerical values, or text, and passes it to the next layer for further processing. Hidden Layers: Hidden layers are composed of neurons that perform mathematical transformations on the data received from the input layer or previous hidden layers. The complexity of the network increases with the number of hidden layers, enabling the model to capture intricate patterns in data. Output Layer: The output layer generates the final prediction or decision based on the information processed by the network. The neurons within each layer are connected by an associated weight that determines the strength of the signal transferred between neurons. An essential component of a NN is the activation function, which introduces non-linearity to the system. Without it, a NN would behave like a linear regression model, limiting its ability to capture complex patterns. The process of training a NN involves adjusting the weights of the network to minimize an error between the predicted output and the actual target value. This is typically done using an optimization algorithm such as gradient descent. The goal is to find the optimal set of weights that minimize the loss function, which quantifies the error of the model.

There are a number of various types of NN used today. Feed-forward NNs are the simplest

type, where information moves in one direction from input to output without cycles, loops, or other specialized layers. Convolutional NNs (CNNs) are specialized for processing grid-like data, such as images. They use convolutional layers to detect patterns (e.g., edges, textures) and pooling layers to reduce dimensionality, making them highly efficient for tasks like image classification and object detection. Another type of common network architecture is a Recurrent NN (RNN), which has connections that loop back on themselves, allowing them to maintain a memory of previous inputs. This makes RNNs particularly suited for sequential data tasks, such as time series forecasting and natural language processing. Lastly, there are Generative Adversarial Networks (GANs), consisting of two networks: a generator that creates synthetic data, and a discriminator that attempts to differentiate between real and generated data. This adversarial training process enables GANs to generate high-quality synthetic data. The NNs employed in most of this work are in reference to feed-forward NNs, as one of the fundamental goals and question of these projects is whether simple parametrization schemes can be emulated effectively with simple ML techniques.

Despite their successes, NNs face several challenges, including the need for large amounts of data, computational resources, and the risk of overfitting; wherein a model learns to memorize training data rather than generalize to new, unseen data. Regularization techniques, such as dropout and weight decay, as well as advances in deep learning architectures like transformers, are helping to address some of these challenges. Recent developments in NNs, such as transfer learning and semi-supervised learning, allow for more efficient learning from smaller datasets and better generalization across different domains, however these topics are beyond the scope of this work [Choudhary et al., 2022]. These techniques are driving significant advances in fields like computer vision, natural language processing, and reinforcement learning.

#### 1.4.1.3 Trustworthy and Explainable AI

Reproducibility, explainability, and trustworthiness are fundamental concerns in scientific research. These issues are particularly accentuated within the domains of ML and AI, where understanding and confidence in the methodology are essential to the integrity of the results. The capability to replicate and trust findings transcends the methodologies utilized, be they conventional or aided by ML,

ML approaches, which are commonly perceived as 'black box' models, often attract scrutiny and skepticism regarding their reliability and the validity of their outputs. Although these considerations are not the primary focus of this dissertation, it is pertinent to acknowledge their importance. We direct the reader to comprehensive reviews on this topic in the context of our field, such as Yang et al. [2024]. The work categorizes existing methods

into two primary paradigms: post-hoc interpretability techniques, which explain pre-trained models using approaches like perturbation-based, game theory-based, and gradient-based attribution methods; and inherently interpretable models designed from the ground up using architectures such as tree ensembles and explainable NNs. The survey highlights how these techniques offer insights into model predictions, uncovering novel meteorological relationships captured by ML. Additionally, the authors highlight challenges related to attaining deeper mechanistic interpretations grounded in physical principles, such as establishing standardized evaluation benchmarks, embedding interpretability within iterative model development processes, and enabling explainability for large foundational models like Google's NeuralGCM and GraphCast.

The ethical implications, as well as the credibility and explainability of AI and ML applications, are the subject of ongoing discourse among professionals and researchers. For example, the works by Bostrom et al. [2024], McGovern et al. [2022], and Flora et al. [2024] offer comprehensive narratives on the topic. Dialogue is advanced in Ebert-Uphoff and Hilburn [2023] and Mamalakis et al. [2022].

To us, addressing the so-called 'appropriateness' of employing AI and ML in scientific inquiry necessitates a reflection on the root of hesitation associated with their use. The misinterpretation of scientific terminology can cause mistrust of established knowledge, a challenge familiar in climate science discourse. Terms such as 'bias' and 'uncertainty' possess precise, quantifiable meanings within their scientific and modeling contexts, yet they can be misconstrued or misappropriated in broader, often politically influenced debates. Similarly, unfamiliarity with AI and ML can lead to reluctance to trust the findings derived from these technologies, even among scientists.

Ultimately, ML represents a tool at the disposal of the scientific community, with a multitude of potential benefits across various research areas. Crucially, it is incumbent upon researchers to discern the appropriate contexts for deploying ML and to commit to transparency regarding the results being ethical, interpretable, reproducible, and credible.

### 1.4.2 Machine Learning and Atmospheric Science

As mentioned, ML has already proliferated throughout the field of atmospheric science. McGovern et al. [2022] provides a comprehensive analysis of the motivations and limitations associated with the deployment of ML in atmospheric sciences, emphasizing the critical need to understand the circumstances under which ML can benefit specific scientific inquiries. The integration of domain expertise into ML development and the interpretation of its outputs is also underlined as crucial. These concepts form the basis of the discussions here and shape

the research presented in this dissertation.

Additionally, Barnes et al. [2019] advocate for increased collaboration between physical scientists and data scientists to foster the most effective and tailored ML applications within the field. Karpatne et al. [2019] further explore these challenges, highlighting the diversity of geoscience datasets, their unique characteristics, and how their varying properties influence the appropriate choice of ML approaches and architectures for specific research problems.

Throughout the past decade, ML has seen a proliferation of applications in geoscience. This section outlines several key examples, with further instances and relevant studies discussed in subsequent chapters. Highlighting an application from the broader field of geoscience, researchers at NCAR have employed probabilistic ML methods to approximate the mixed layer depth of the oceans using satellite observations [Foster et al., 2021]. By also incorporating sea surface data derived from model simulations, the researchers have enhanced their ML model's performance, which has demonstrated competitive results in comparison with other ML and conventional strategies. Such work exemplifies the potential of ML to advance geoscience research when scientific questions are approached with consideration of the pertinent data's availability, choice of ML techniques, and their applicability to the research objectives.

#### 1.4.2.1 Recent Applications of ML in Post-processing of Weather Forecasts and Climate Projections

Weather forecasting is a prominent domain within the atmospheric sciences that has notably integrated ML methodologies in recent decades. Early initiatives pioneering the use of ML for real-time prediction of severe weather phenomena include the work of Lagerquist et al. [2017] on forecasting convective wind events and Gagne et al. [2017] on predicting extreme hail occurrences. Subsequent advancements were introduced by Lagerquist et al. [2019], who employed deep learning techniques to detect synoptic-scale fronts that significantly influence both daily weather and extreme weather patterns. More recent efforts have been directed toward enhancing the operational, real-time delineation of these frontal boundaries [Justin et al., 2023].

In addition to deep learning, other ML techniques have been applied to weather forecasting. For instance, Loken et al. [2022] investigated the utility of RFs for predicting severe weather hazards one day in advance. Their study is notable for comparing two distinct methodologies and examining the relative benefits of using ensemble mean outcomes from trained models versus individual model members for optimizing forecast accuracy.

A comprehensive review by McGovern et al. [2023] discusses the extensive range of ML applications in convective weather system forecasting. This review elaborates on the extensive

body of research at the intersection of ML and meteorological forecasting.

Keller and Potthast [2024] introduced an innovative AI-based variational data assimilation approach that integrates the data assimilation process directly into a NN. The method leverages deep learning techniques to minimize the variational cost function, enabling data assimilation without relying on pre-existing analysis datasets. Their proof-of-concept demonstrated that their model can efficiently assimilate observations and produce accurate initial conditions for numerical weather prediction, highlighting another example of the potential for AI to enhance computational efficiency in weather forecasting systems.

Numerous studies have demonstrated the application of ML techniques to the post-processing of climate and weather model outputs. Yorgun and Rood [2016], for instance, utilized RFs to address biases resulting from the coupling of physical and dynamical processes in climate models. In earlier work related to Loken et al. [2022], RFs were employed to refine next-day precipitation ensemble forecasts [Loken et al., 2019].

Chapman et al. [2019] applied deep learning methods to identify and correct biases in weather forecasting, particularly looking at atmospheric rivers. Watt-Meyer et al. [2021] developed an RF-based approach to adjust parameterized tendencies in hindcast simulations to align more closely with observational data. In another innovative use of ML, Bretherton et al. [2022] trained algorithms to determine nudging tendencies that can improve physical parameterization schemes in coarse-resolution model runs.

Another recent example of AI-based post processing is the work of Stachura et al. [2024] with their tool aimed at improving the calibration of Numerical Weather Prediction (NWP) outputs, specifically for near-surface air temperature forecasts. Their approach leverages statistical and deep learning techniques to correct biases inherent in raw model outputs by learning from historical forecast errors and observed weather data. Unlike traditional bias correction methods, which often rely on linear regression or simple statistical adjustments, their AI-driven approach dynamically adjusts forecast outputs based on complex, nonlinear relationships between atmospheric variables. Their findings demonstrated that ML-based post-processing significantly enhances forecast accuracy, particularly in extreme temperature scenarios where traditional models struggle. This work highlights the potential of AI in refining operational weather forecasting by reducing systematic biases and improving probabilistic forecast reliability.

#### 1.4.2.2 Recent Applications of ML: Model Development and Augmentation

Addressing the concerns over model bias and uncertainty associated with parameterization schemes is an extensive area of research at the intersection of ML and climate model development. This is also the area aligned with the core subject of the research presented in this

dissertation. While many foundational studies are detailed in the introduction to Chapter 2, we mention additional research here for a comprehensive look at the field.

Significant contributions in this field of emulation include the work by Rasp [2020], who explored online learning methods with the aim of reducing biases and uncertainties in parameterization schemes using the Lorenz 96 framework [Lorenz, 1995]. Although their model yielded encouraging results, they acknowledged limitations and expressed reservations regarding the generalizability of their approach, particularly in the context of a changing climate. In contrast, Brenowitz and Bretherton [2019] successfully emulated moistening and heating processes within a cloud-resolving model, utilizing a synergistic approach of coarse graining and NN techniques. Subsequent studies by the same group focused on interpreting and stabilizing an ML emulator for convection in the super-parameterized Community Atmosphere Model (SP-CAM) [Brenowitz et al., 2020]. Yuval and O'Gorman [2023] further advanced the field by using NNs to predict momentum fluxes tied to convective parameterization in climate models. These initiatives underscore the potential of ML emulation in parameterization schemes, an area of active research that necessitates in-depth understanding of the interplay between the ML algorithms and climate model numerics, thereby providing inspiration for the present thesis.

More recently, Heuer et al. [2024] developed ML models to improve the representation of convective processes in the icosahedral non-hydrostatic (ICON) climate model. They trained various ML algorithms, including U-Net architectures and Gradient Boosted Trees, on data from high-resolution ICON simulations over the tropical Atlantic [Friedman, 2002, Ronneberger et al., 2015]. A key finding was that while the U-Net demonstrated strong performance in offline tests, it inadvertently learned non-causal relationships with precipitation, leading to instability when integrated into the ICON model. By modifying the U-Net to exclude these non-causal connections, the researchers achieved stable, long-term simulations and improved predictions of precipitation extremes. This work underscores the importance of interpretability in ML-based parameterizations and highlights the potential of tailored ML models to enhance climate simulations. Additionally, Otness et al. [2023] explored a data-driven multiscale modeling approach for subgrid parameterizations, utilizing NNs to predict unresolved forcings across different scales. This highlighted the benefits of incorporating additional multiscale information to enhance prediction accuracy and generalization in climate models.

Similarly, in the related domain of ocean modeling, there are a number of examples of ML emulators being utilized. Ross et al. [2023] systematically evaluated ML models designed to emulate subgrid-scale processes in an ocean model. Their findings highlighted that while NN-based parameterizations improved upon low-resolution models, they often strug-

gled to generalize to unseen oceanic conditions. To address this, the authors proposed a novel equation-discovery approach combining linear regression and genetic programming, resulting in a symbolic parameterization that demonstrated robust performance across diverse scenarios. Zhang et al. [2023] implemented a machine-learned mesoscale eddy parameterization into an ocean circulation model, demonstrating its ability to improve the representation of subgrid processes. Their study assessed the generalization capabilities of the learned parameterization and its impact on large-scale ocean dynamics.

#### 1.4.2.3 Recent Applications of ML: Emulators for General Circulation Models and Benchmark Data Sets

Recently, many groundbreaking developments at the intersection of ML and atmospheric modeling have expanded well beyond enhancing and augmenting parameterization schemes. Researchers affiliated with industry giants such as Google, Microsoft, and NVIDIA have introduced advanced digital twin weather forecasting models such as FourCastNet [Kurth et al., 2023], GraphCast [Lam et al., 2023], and Pangu-Weather [Bi et al., 2023]. These models share common attributes, including the utilization of generative and transformer architectures and training on reanalysis data, which is a product that blends numerical model outputs with observational data to create a standardized, high-resolution global dataset of atmospheric conditions spanning the past 80 years [Hersbach et al., 2020]. While reanalysis datasets offer a valuable resource forAI-driven models, they are subject to inherent uncertainties, particularly in extreme weather events and regional-scale predictions, where observational coverage is sparse or inconsistent.

Rasp et al. [2020] introduced a foundational framework for benchmarking these kinds of AI-driven weather forecasting models, providing a standardized dataset and evaluation metrics to compare ML approaches with traditional numerical weather prediction (NWP) models. WeatherBench utilizes reanalysis data from the European Centre for Medium-Range Weather Forecasts (ECMWF). In particular, the ECMWF Reanalysis version 5 (ERA5) dataset offers global atmospheric variables at various pressure levels and surface conditions [Hersbach et al., 2020]. ERA5 is the fifth-generation ECMWF atmospheric reanalysis product, combining model output with a wide range of observations to produce a globally complete, physically consistent estimate of past weather and climate. It provides fields at hourly temporal resolution and a native spatial grid at 0.25° resolution (roughly 31 km spacing) for most variables, with vertical coverage up to 37 pressure levels. The framework simplifies model evaluation by providing predefined metrics such as root mean square error (RMSE) and anomaly correlation coefficient, ensuring consistency across studies. More recently, WeatherBench 2 has expanded on this foundation by incorporating higher-resolution

data, additional variables, and more comprehensive assessment tools tailored to extreme weather events and data-driven downscaling tasks [Rasp et al., 2024]. By establishing a common benchmark, WeatherBench has accelerated the development of ML-based forecasting techniques, offering a clear path for comparing innovations and guiding the community toward models that balance predictive accuracy with physical interpretability.

In addition to WeatherBench, other benchmarking frameworks have emerged to support the development and evaluation of AI-driven weather and climate models, notably Artificial Intelligence for Environmental Sciences (AI2ES) and ClimateBench. TheAI2ES initiative focuses on applying ML techniques to improve environmental hazard prediction, such as severe weather events and subseasonal-to-seasonal forecasting [McGovern et al., 2023]. AI2ES emphasizes interdisciplinary collaboration, providing datasets and tools to help researchers develop AI models that enhance the prediction of high-impact phenomena like tornadoes, hurricanes, and extreme precipitation. By offering standardized datasets and evaluation protocols tailored to these extreme events, AI2ES bridges the gap between ML innovation and operational forecasting, ensuring that new models meet the practical needs of meteorological agencies and disaster response teams.

Meanwhile, ClimateBench addresses the distinct challenge of modeling long-term climate dynamics. Developed as part of the AI for Climate Change Initiative, ClimateBench uses data from the Coupled Model Intercomparison Project Phase 6 (CMIP6) to benchmark ML models tasked with emulating complex climate processes [Watson-Parris et al., 2022]. The framework facilitates comparison between ML-based climate emulators and traditional climate models, with a focus on simulating responses to anthropogenic forcing scenarios. Importantly, ClimateBench highlights the role of AI in accelerating climate projections, enabling rapid exploration of future climate states under various emission pathways. Together, AI2ES and ClimateBench complement WeatherBench and WeatherBench 2 by extending the focus beyond short-term weather forecasting to encompass extreme weather events and long-term climate modeling, providing a comprehensive landscape for benchmarking AI in atmospheric sciences.

The work of Li et al. [2024] introduces the concept of using a generative ensemble forecasting approach that integrates AI with traditional NWP models. This method employs generative models to create ensemble forecasts, enhancing the representation of uncertainty in weather predictions. Unlike purely AI-driven models that rely on ML algorithms trained on observational data, this approach synergizes AI techniques with established NWP frameworks, leveraging the strengths of both to improve forecast accuracy and reliability.

Another interesting approach to global AI-driven modeling approaches is NeuralGCM that represents a significant advancement in climate modeling by integrating ML techniques with

traditional physics-based approaches [Kochkov et al., 2024]. This hybrid model combines a differentiable atmospheric dynamical core with NN components to emulate small-scale processes, aiming to enhance both accuracy and computational efficiency in weather and climate simulations. In contrast to purely data-driven models like FourCastNet and Pangu-Weather, which rely heavily on extensive reanalysis datasets, NeuralGCM incorporates fundamental physical principles directly into its architecture. This integration allows NeuralGCM to maintain stability over extended simulations, accurately tracking climate metrics such as global mean temperature over multiple decades [Kochkov et al., 2024].

While these kinds of models mark a paradigm shift in numerical weather prediction, their practical applications still face open questions, particularly regarding their ability to replicate key dynamical properties of the atmosphere. Efforts are ongoing to develop test cases designed to assess the reliability and robustness of these AI-driven weather models. We deployed these test cases within the framework of an AI-focused working group at the 2025 Dynamical Core Model Intercomparison Project (DCMIP), where various AI-driven weather models were compared against each other. This work situates itself within a rapidly evolving field that extends beyond industry, as government and academic institutions, such as ECMWF and NCAR, are also actively developing AI-driven weather models.

## 1.5 Overview of the Thesis

This dissertation explores machine learning approaches for Earth system modeling, grounded in foundational scientific principles. It is structured around three core projects that explore the integration and evaluation of ML within climate and weather modeling frameworks. The first chapter, taken directly from our published manuscript Limon and Jablonowski [2023], investigates the use of simple ML algorithms to emulate simplified physical processes in an offline setup, providing insight into the relationship between complexity of parameterization and the offline skill of ML emulators, focused primarily on RFs. The second chapter addresses the challenges of coupling such emulators, RFs and NNs, into CAM, the atmospheric component of the Community Earth System Model (CESM). Here, we highlight key issues related to stability, generalization, and physical consistency in online settings. The third chapter shifts focus to the development and extension of dynamical tests for modern AI-driven weather forecasting systems like GraphCast. In particular, it includes a preliminary analysis of differences in dynamic responses in GraphCast's 37-level model versus its 13-level operational counterpart to an imposed tropical heating anomaly. Together, these studies aim to advance the understanding of how ML tools can be robustly integrated into the physical modeling of the climate system.

# CHAPTER 2

# Probing the Skill of Random Forest Emulators for Physical Parameterizations via a Hierarchy of Simple CAM6 Configurations

*Note for the reader:* This chapter features a published manuscript with minor reformatting to be featured in the thesis. The original published work comes from Limon and Jablonowski [2023].

## 2.1 Introduction

In recent decades machine learning (ML) has become an intriguing tool for atmospheric scientists. It provides the unique ability to bridge data science with the physical sciences in order to improve our understanding of the Earth system [Reichstein et al., 2019, Boukabara et al., 2021]. While ML is still a relatively novel approach to applications in climate science, there is already an abundance of research utilizing these techniques. Some examples include identifying mixed layer depths in the ocean via observations [Foster et al., 2021], attributing model biases from physics-dynamics coupling in climate models [Yorgun and Rood, 2016], improving severe hail predictions over the US high plains [Gagne et al., 2017], post-processing bias corrections of weather forecasts [Chapman et al., 2019], and implementing corrective schemes like 'nudging' physics tendencies via coarse-graining or hindcasting [Bretherton et al., 2022, Watt-Meyer et al., 2021].

General Circulation Models (GCMs) are made up of a dynamical core, responsible for the geophysical fluid flow calculations, and physical parameterization schemes. The latter estimate subgrid-scale processes that are generally not resolved by the dynamical core's computational grid. These processes include aspects of the Earth system such as radiation, convection, turbulence, and microphysical processes, among others. They are a source of significant bias and model uncertainty due to the heuristic nature of their development [Held,

2005, Stevens and Bony, 2013, Hourdin et al., 2017]. Parameterization schemes can range significantly in complexity, from simple forcing mechanisms that produce quasi-realistic and stable atmospheric flow conditions, to state-of-the-art packages wherein the various unresolved processes work in conjunction with each other [Bogenschutz et al., 2013, Gettelman and Morrison, 2015]. In this paper, we focus primarily on the former, wherein simplified forcing mechanisms for wind, temperature, moisture, and precipitation are used to produce quasi-realistic atmospheric flow.

Beginning with the work of Krasnopolsky and Fox-Rabinovitz [2006] applying neural networks (NN)s to climate and weather prediction model development, ML became an attractive candidate for augmenting the subgrid-scale physics schemes within weather and climate models. In recent years, ML techniques have already been shown to be capable of replicating parameterizations schemes to various degrees of effectiveness [Beucler et al., 2021, Yuval and O'Gorman, 2020]. Specifically, Ukkonen [2022] was able to develop ML emulators for radiative transfer processes, O'Gorman and Dwyer [2018] and Gentine et al. [2018] used random forests (RF) and NNs to emulate moist convection processes, respectively, Gettelman et al. [2021] utilized NNs to emulate a component in the micro-physics scheme within a GCM, Chantry et al. [2021] developed a nonorographic gravity wave drag emulator, and Rasp et al. [2018] and Brenowitz and Bretherton [2018] tackled a full physics emulator of cloud-resolving and near-global aquaplanet simulations, respectively, via NNs. These are just a few examples showing both the promise of ML emulation and some limitations, particularly in regards to model stability and physical realism [Beucler et al., 2021, Yuval et al., 2021].

Our work is inspired by many of these recent studies into ML emulation for parameterization schemes, with a focus on multiple simplified physics configurations within version 6 of the Community Atmosphere Model (CAM6). CAM6 is the atmospheric GCM within the Community Earth System Model (CESM) [Danabasoglu et al., 2020] framework, developed by the National Center for Atmospheric Research (NCAR). In particular, we utilize a hierarchy of three physical forcing setups of varying complexities. Each setup contains a well-defined increase in non-linearity associated with its mathematical expressions. The parameterization schemes begin with a dry model setup, described in Held and Suarez [1994] and referred to as HS hereon. This is followed by a moist version of the HS scheme developed by Thatcher and Jablonowski [2016], referred to as TJ. Lastly, a modified version of the TJ scheme is used in which we couple a simple Betts-Miller (BM) convection scheme to the physics processes [Betts and Miller, 1986, Frierson, 2007]. These three parameterization packages may also be referred to throughout the papers as dry, moist, and convection, respectively. None of these physics schemes include topography or seasonal and diurnal cycles.

The primary focus of this work utilizes RFs that are uniquely trained and tuned for

each case, allowing for an investigation into the relationship between the degree of non-linearity within the parameterization scheme and the corresponding effectiveness of the RF to emulate the forcing. Probing the limits of an RF emulator in an offline mode with respect to simplified parameterization schemes allows for a better understanding of an ideal baseline for these methods in the pursuit of identifying areas in which they may be applicable. Of course, NNs are an alternative ML technique that has effectively become the standard in this field in recent years. It is useful to keep in mind that this work does not aim to find the 'best possible' emulator for our simplified schemes, rather we ask more fundamental questions about the dependence of the ML skill on the physical complexity of a parameterization. This is why we chose RFs to be our main focus, as they are an adequate tool to address this question and possess properties that are of interest to us as physical scientists. That being said, we do provide results from baseline NN emulators for each case in the interest of completeness.

In this work, we show that various physical forcing tendencies and precipitation rates can be emulated by both the RF and NN models in an offline mode. We do not include an online evaluation of our emulators. This is intentional as we strive to understand the limits of the RF emulators and raise questions about the feasibility of RFs for use in more complex parameterization schemes. In many cases, our ML models are shown to be highly skilled, both from a statistical perspective and from direct comparisons. We begin with an explanation of the three model configurations, our model run setup and data processing steps, and a background discussion on ML techniques in section 2. This is followed by our results and discussion in section 3 before culminating with concluding thoughts in section 4.

## 2.2 Methods

### 2.2.1 CAM6 Configurations

#### 2.2.1.1 Dry Scheme

The dry CAM6 model configuration utilizes two physical forcing mechanisms as described in HS. The dissipation of the horizontal wind is represented by Rayleigh friction at the lower levels of the model (below 700 hPa) and thereby mimics the surface friction and the planetary boundary layer (PBL) mixing of momentum. The Rayleigh friction is expressed as

$$\frac{\partial \vec{v}_h}{\partial t} = -k_v(p)\,\vec{v}_h. \tag{2.1}$$

In addition, radiation is mimicked by a Newtonian temperature relaxation described by

$$\left(\frac{\partial T}{\partial t}\right)_{\text{HS}} = -k_T(\phi, p)\,[T - T_{\text{eq}}(\phi, p)]. \tag{2.2}$$

Here, $\partial/\partial t$ represents a sub-grid physics tendency (forcing) of a variable over a physics time step, $p$ symbolizes the pressure, $\phi$ denotes the latitude, $\vec{v}_h$ is the horizontal velocity vector, $T$ stands for the temperature, $T_{\text{eq}}$ is a pre-defined equilibrium temperature profile, and $k_v$ and $k_T$ are the dissipation and relaxation coefficients, respectively, with the inverse time unit $\text{s}^{-1}$. The details are provided in HS. These forcings are coupled to the dry dynamical core and produce stable atmospheric fluid flow, triggering quasi-realistic processes such as Rossby waves in the midlatitudes. This model configuration comes implemented within CAM6's 'Simpler Models' framework and is set with the 'FHS94' compset choice.

#### 2.2.1.2 Moist Scheme

The moist TJ physics scheme is similarly forced by Rayleigh friction and the Newtonian temperature relaxation. However, the equilibrium temperature is now slightly different than its HS variant and additional forcing mechanisms are used. These include large-scale condensation with its associated heating or cooling effects, surface fluxes of latent and sensible heat, and a PBL mixing scheme for temperature and moisture via a second-order diffusion mechanism. The PBL mixing and surface friction of momentum is kept identical to the HS Rayleigh friction approach. All details of the TJ moist physics package are provided in Thatcher and Jablonowski [2016]. To illustrate the enhanced complexity in comparison to HS, the TJ temperature forcing now takes the form

$$\left(\frac{\partial T}{\partial t}\right)_{\text{TJ}} = -k_T(\phi, p)\,[T - \tilde{T}_{\text{eq}}(\phi, p)] + \frac{L}{c_p}C + \frac{C_H|\vec{v}_a|(T_s - T_a)}{z_a} + \text{PBL Diffusion} \tag{2.3}$$

where $\tilde{T}_{\text{eq}}$ is a modified equilibrium profile defined in TJ, $L$ is the latent heat of vaporization, $C$ is the large-scale condensation rate, $c_p$ is the specific heat at constant pressure, $C_H$ is the transfer coefficient for sensible heat, $|\vec{v}_a|$ is the horizontal wind speed at the lowest model level, $T_s$ is the surface temperature, $T_a$ is the temperature of the lowest model level, and $z_a$ is the height of the lowest model level. The latter five are needed for the computation of the sensible heat flux at the surface. The details of the PBL temperature diffusion algorithm are provided in TJ and Reed and Jablonowski [2012]. This model setup is also implemented within the 'Simpler Models' framework in CAM6 via the 'FTJ16' compset, which assumes an ocean-covered lower boundary with a prescribed sea surface temperature and no topography.

The inclusion of moisture brings an additional forcing tendency for specific humidity,

which is similarly impacted by the large-scale condensation rate, the latent heat flux at the surface, and PBL diffusion

$$\left(\frac{\partial q}{\partial t}\right)_{\text{TJ}} = -C + \frac{C_E|\vec{v}_a|(q_{\text{sat},s} - q_a)}{z_a} + \text{PBL diffusion} \tag{2.4}$$

Here, $q$ refers to the specific humidity, $C_E$ is the bulk transfer coefficient for water vapor, $q_{\text{sat},s}$ is the saturation specific humidity at the surface, and $q_a$ is the specific humidity at the lowest model level. Again, mathematical details of the PBL diffusion of $q$ are provided in TJ and and Reed and Jablonowski [2012]. Additionally we chose to emulate the large-scale precipitation rate which is modeled via the equation

$$P_{\text{ls}} = \frac{1}{\rho_{\text{water}} g} \int_{p_{top}}^{p_s} C dp \tag{2.5}$$

where $\rho_{\text{water}}$ is the density of water, $g$ is gravity, $p_{top}$ is the pressure at the model top, and $p_s$ is the surface pressure.

#### 2.2.1.3 Convection Scheme

The final step in our CAM6 model hierarchy couples the BM convection scheme to the TJ setup [Betts and Miller, 1986, Betts, 1986, Frierson, 2007]. This configuration is not built into the CAM6 'Simpler Models' framework and required some minor modifications to the TJ setup. The simplified BM technique follows the description by Frierson [2007] and we recommend this paper for a more complete description. To summarize, the resulting tendencies with the addition of the BM convection scheme can be written as

$$\left(\frac{\partial T}{\partial t}\right)_{\text{BM}} = -\frac{T - T_{\text{ref}}}{\tau} + \left(\frac{\partial T}{\partial t}\right)_{\text{TJ}} \tag{2.6}$$

$$\left(\frac{\partial q}{\partial t}\right)_{\text{BM}} = -\frac{q - q_{\text{ref}}}{\tau} + \left(\frac{\partial q}{\partial t}\right)_{\text{TJ}} \tag{2.7}$$

where $\tau$ is the convective relaxation time and $T_{\text{ref}}$ and $q_{\text{ref}}$ are reference temperature and specific humidity profiles for the convection. Within our implementation, the BM scheme is calculated first, before the rest of the TJ scheme.

The convection scheme utilizes regimes of precipitation due to warming, $P_T$, and precipitation due to drying, $P_q$. In the regime of $P_T > 0$ and $P_q > 0$, 'convection' is triggered. Frierson [2007] described in detail how extra steps are taken with regards to the reference profiles in order to ensure the conservation of enthalpy in the deep convection regime. The author also describes three approaches to handling shallow convection. In our work we use

the so-called "shallower" scheme, in which the reference temperature is further modified in order to lower the depth at which shallow convection occurs. This is considered the simplest technique within the BM scheme that allows for both deep and shallow convection to occur.

The BM convection scheme has a dependency on two coefficients: the relative humidity threshold for the reference temperature profile ($\mathrm{RH_{BM}}$) and $\tau$, the convective relaxation time. In order to choose these values, we examined various profiles of a variety of fields and compared them to fields from a CAM6 aquaplanet configuration [Williamson et al., 2012, Medeiros et al., 2016]. Details on the aquaplanet model setup and how it was used to identify our choices of $\mathrm{RH_{BM}}$ and $\tau$ can be found in the Supporting Information Text S1. The aquaplanet configuration acts as a loose reference for these choices as it is a widely used model configuration in which the planet's surface is covered by an ocean. This allows for surface-ocean interactions to become an integral component of the underlying physics. It is useful for exploring many aspects of geophysical fluid flow in a controlled model setting. The chosen values were $\tau = 4$ hr and $\mathrm{RH_{BM}} = 0.7$.

### 2.2.2 Machine Learning

Broadly speaking, there are two categories of ML applications: supervised and unsupervised learning. Unsupervised learning encompasses tasks that attempt to identify general patterns in data, for example, clustering algorithms. Supervised learning strives to identify correlations or functional relationships between a labeled input and output. There are two primary tasks that can be done with supervised learning: classification and regression; the latter is applicable to emulating physical parameterizations. Regression is the process of estimating a functional relationship between a dependent variable (the predictant), referred to as the label or output, and one or more independent variables, referred to as features or input variables when using ML terminology. With this framework in mind, we can think of regression as the process of identifying the function $\hat{g}(\vec{X})$ such that

$$\hat{g}(\vec{X}) \approx f(\vec{X}) \tag{2.8}$$

where $f(\vec{X})$ is the function we seek to identify and $\vec{X}$ is the vector of input variables (features).

What separates modern machine learning techniques like NNs, support vector machines, and RFs are their applications to nonlinear systems, providing methods for nonlinear regression tasks. In its simplest form, a physical parameterization is a nonlinear function that describes a tendency or precipitation rate (dependent variable) given the (independent) state variables. In the analogy to Equation 8, the tendency would be $f$ while the state variables

make up the vector $\vec{X}$ and our trained ML model will be $\hat{g}(\vec{X})$.

We primarily focus on RFs to emulate the parameterization schemes, but we also include a brief investigation into simple NNs as well for comparison. An RF is an ensemble of decision trees, which can themselves be considered an ML technique. Decision trees identify thresholds among a branch network, forming a structure of conditional operations that produce a prediction [Breiman, 1996]. Random forests are commonly used in classification applications of ML, but have been shown to be effective for nonlinear regression tasks in atmospheric science as well [O'Gorman and Dwyer, 2018]. Various trees in the forest are initialized at random and are then trained along side each other. The final result is an ensemble average of the results from all trees in the forest. Neural networks are another approach we use to show the effectiveness of ML techniques to emulate these processes. Neural networks are the baseline approach to the field of deep learning, in which densely connected layers of 'neurons' are linked via an activation function that is able to map nonlinear functions between the labeled input and output. The field of deep learning is vast and has been undergoing rapid advancements within Earth system science, but for the purposes of this work, we just focus on the case of standard feed forward NNs [Baldi, 2021, Reichstein et al., 2019].

When applicable, RF approaches are of interest due to both its relative simplicity as an application of non-linear regression, its interpretability, along with inherently preserving some underlying physical properties of our predicted fields. Since each individual tree produces an output that is within the scope of the training data, their average is also inherently within the scope of the data. This means that RFs cannot extrapolate to a prediction outside of the range established by their training data. In the context of using ML techniques for physical science applications, this is a welcome property because it can avoid potential artifacts that could be inconsistent with the physics at play. For example, an RF will inherently adhere to the non-negative property of precipitation, as it will have never encountered negative precipitation in its training data. This is in contrast to techniques such as NNs, which historically have difficulty with extrapolation and adhering to underlying physical constraints [Beucler et al., 2021].

We developed a streamlined workflow from data generation to training, testing, and analysis by utilizing CAM6's built-in 'Simpler Models' physics framework along with the Python libraries Xarray, scikit-learn, and Keras [Hoyer and Hamman, 2017, Pedregosa et al., 2011, Chollet, 2017]. Xarray allows for straightforward data manipulations of NetCDF data, scikit-learn is a well-maintained ML library that includes user-friendly RF implementations for Python, and Keras is a Python library that provides and approachable interface for the Tensorflow deep learning framework.

### 2.2.3 Model Setup and Data Preparation

The simple model configurations allow us to generate large quantities of model output to train our machine learning models. Working with CAM6, we utilize its Finite Volume (FV) dynamical core [Lin, 2004] with 30 pressure-based vertical levels and a model top at roughly 2.2 hPa. The exact placement of the model levels is specified in Reed and Jablonowski [2012] (see their Appendix B). The model is run for 60 years with a latitude-longitude grid of resolution $1.9^\circ \times 2.5^\circ$ - simply referred to as 2-degree resolution and corresponds to roughly 200 km grid spacing. We output data for state variables, including temperature, surface pressure, specific humidity, and the diagnostic quantity relative humidity, once every week of the simulation just before the prognostic states are updated by the physics package. Additionally, we output the tendencies due to the physical parameterization package after they are updated with the same output frequency. This is an important modification since by default both the state variables and physical tendencies are output after the physics update. We chose to output once per week in order to avoid close correlations between the time snapshots. Strong correlations are present in data snapshots that are only separated by short time intervals, such as a day. This allows for our data to include a larger range of the functional space, while avoiding redundancies within the scope of the training data. It should be reiterated that our configurations do not include a diurnal or seasonal cycle, which allows us to be able to take weekly output without risking an incomplete representation of the functional space. For more complicated systems, care would need to be taken in choosing output intervals that effectively sample the functional space.

Here, we define the input fields for our ML models to be the state variables used by the underlying schemes, such as temperature and pressure. Similarly, the output fields are the resulting tendency or precipitation rate being predicted. For preprocessing, we focus primarily on the shape of the data, input choices, and the distribution of the data between training and testing. The state variables and tendencies, using temperature ($T$) as an example, are generally output from the model in the shape

$$T(N_{\text{time}}, N_{\text{lev}}, N_{\text{lat}}, N_{\text{lon}})$$

where $N_{\text{time}}$, $N_{\text{lev}}$ $N_{\text{lat}}$, and $N_{\text{lon}}$ correspond to the number of temporal snapshots, vertical levels, latitudes, and longitudes, respectively. Some variables are surface fields, such as the precipitation rates, and correspond to $\mathrm{N_{lev}} = 1$. Due to the nature of the physical parameterizations being column-wise implementations in the atmospheric model, we carry

this over as our feature/label dimension. This means our number of samples becomes

$$N_{\text{samples}} = N_{\text{time}} \times N_{\text{lat}} \times N_{\text{lon}}$$

The number of features becomes

$$\mathrm{N}_{\text{features}} = \mathrm{N}_{\text{lev}} \times \mathrm{N}_{\text{input fields}}$$

where 'input fields' include temperature, specific humidity, relative humidity, and pressure, among others. The number of labels becomes

$$\mathrm{N}_{\text{labels}} = \mathrm{N}_{\text{lev}} \times \mathrm{N}_{\text{output fields}} = \mathrm{N}_{\text{lev}}$$

where $N_{\text{output fields}} = 1$ for all cases in this work since we train a unique RF for each predicted tendency or precipitation rate. This was a conscious decision that allows for a robust investigation into the effectiveness of RFs for these emulation tasks as the functional form slowly increases in complexity within our hierarchy. This is in contrast to other similar efforts, such as Rasp et al. [2018] and Yuval and O'Gorman [2020], wherein a single ML model is trained to predict all fields of interest.

Finally, we partition the data into training and testing subsets. The training data comes from the first 50 years of the 60-year model run. We choose a selection of roughly 15-20 million samples (grid columns), which represents the majority of the available data from the 50 years for training. This number depends primarily on the complexity of the chosen RF parameters, the size and shape of the variable, and our computational wallclock limit for training of roughly 24 hours. This wallclock limit is determined by NCAR's data analysis platform 'Casper' used for this work. Furthermore, the physical characteristics of the CAM6 data impact the ML input data. For example, the moisture tendency is zero above roughly 250 hPa. This means that the six model levels between 250 hPa and the model top can be omitted from the process, resulting in significantly fewer data to be processed. Likewise, the precipitation rate is a surface field, which leads to significantly reduced computational cost for training since $N_{\text{labels}} = N_{\text{lev}} = 1$. This allows us to use closer to $N_{\text{samples}} \approx 20$ million for RF emulators, which is just below the upper limit of our generated data. In contrast, the moist and convective temperature tendencies use 15 million samples. The discrepancy between these two cases is a result of the size and complexity of each individually-optimized RF. The number of samples used in training for each case is included in Tables S1 to S8 in the Supporting Information.

The testing data are used to quantify the ability of our RF configurations to emulate

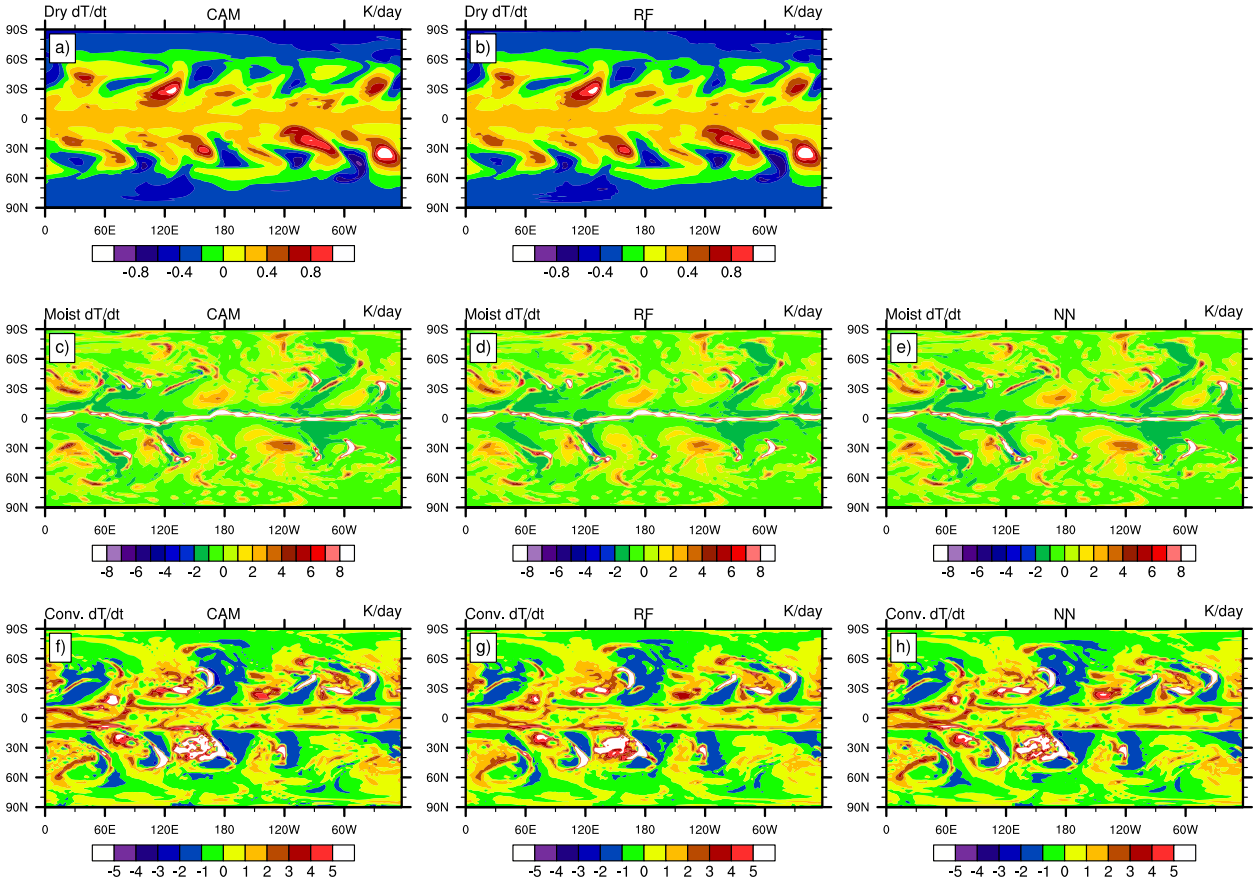



Figure 2.1: Snapshots of the predicted temperature tendencies near 850 hPa for the (top) dry, (middle) moist, and (bottom) convective cases: (left) CAM6 output, (middle column) RF predictions, (right) NN predictions. The magnitude of the extremes in (c), (d), and (e) is around $50-60$ K/day and close to 20 K/day in (f), (g) and (h), but were left out in order to avoid over-saturating the contours.

the parameterization. The testing data were not available during the hyperparameter optimization process or training and come from the final six years of the 60-year CAM6 model run. The time gap between the training and testing data is built into our framework in order to avoid potentially correlated signals between time samples. The chosen 4-year gap is generous, and shorter multi-months gap periods could also be sufficient. It is important to evaluate model performance on data that the ML models have not seen while training in order to ensure that the emulators do not show signs of overfitting. Overfitting in ML occurs when the ML model has been trained well on the subset of data that it has seen, but is unable to generalize to a new set of data from the same source. Lastly, the ML algorithms need to have their hyperparameters tuned in order to obtain an optimized RF architecture for the problem. This is an important part of the ML workflow, albeit less important for RFs relative to other ML approaches, and we utilized the SHERPA hyperparameterization library to

accomplish it in the case of our RFs [Hertel et al., 2020]. Our NN hyperparameters were chosen based on tuning choices made in Beucler et al. [2021], which led to very skillful emulators for our work. We note here that all NNs use the same architecture/hyperparamter choices, meaning that while each case is uniquely trained, they are not uniquely tuned, whereas each RF is both uniquely trained and tuned and can be interpreted as our 'best case' RF for each emulated field. We also incorporated a unitary invariance transform for our NN input, combined with a simple min/max scaler for our output fields. Further details about the process of hyperparameter tuning and the final choices of the selected hyperparameters can be found in Tables A1 to A9 in the Appendix.

## 2.3 Results & Discussion

### 2.3.1 Snapshots & Mean Fields

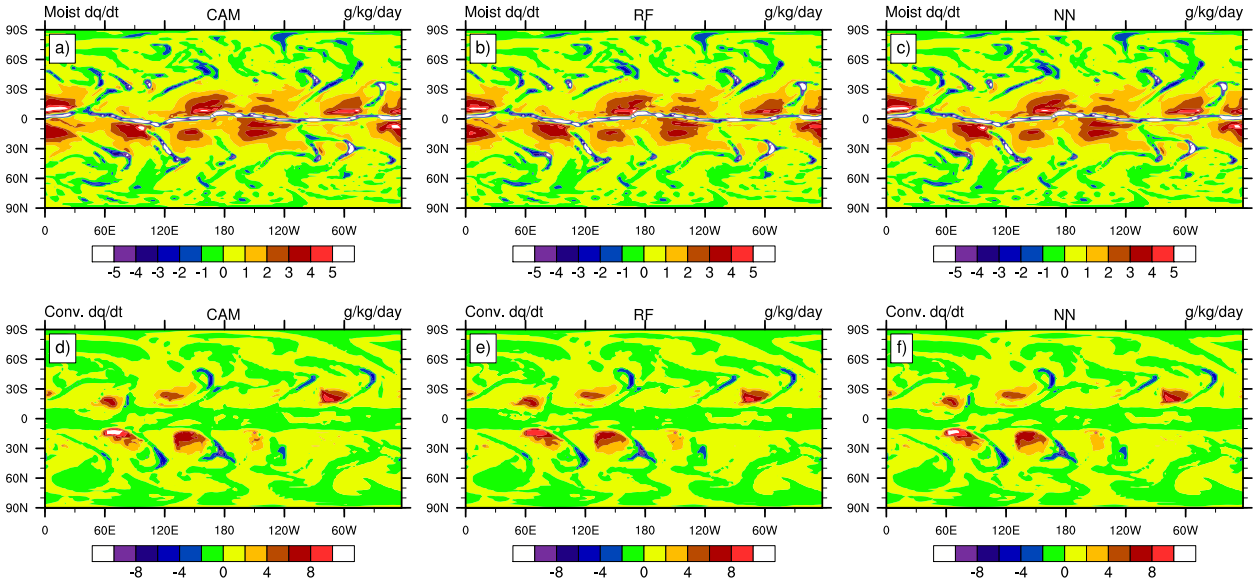



Figure 2.2: Snapshots of the predicted specific humidity tendencies near 850 hPa for the (top) moist and (bottom) convective cases: (left) CAM6 output, (middle column) RF predictions, and (right) NN predictions. The minima in (a), (b), and (c) are around $-20$ g/kg/day, but were left out in order to avoid over-saturating the contours.

Figures 2.1 and 2.2 show horizontal snapshots of the instantaneous CAM6 output, the RF predictions, and the NN predictions for the temperature and moisture tendencies, respectively. From top to bottom, the figures show each of the three physics schemes: dry (Figure 2.1 only), moist, and convection. We chose a snapshot from a randomly chosen time step at the model level closest to 850 hPa. The snapshots in Figures 2.1 and 2.2 show how effective

ML methods can be at emulating simple parameterization schemes in climate models for any given time step. These temporal snapshots allow us to appreciate the agreement between the CAM output and the ML predictions, while still being able to identify areas and magnitudes of discrepancy. They also show how at a given time step, the ML prediction can reproduce the flow properties associated with baroclinic waves in the midlatitudes. This is apparent in the heating tendencies along the frontal zones, as well as decreasing moisture levels in these areas, corresponding to precipitation bands. As an aside, we aim at displaying the results

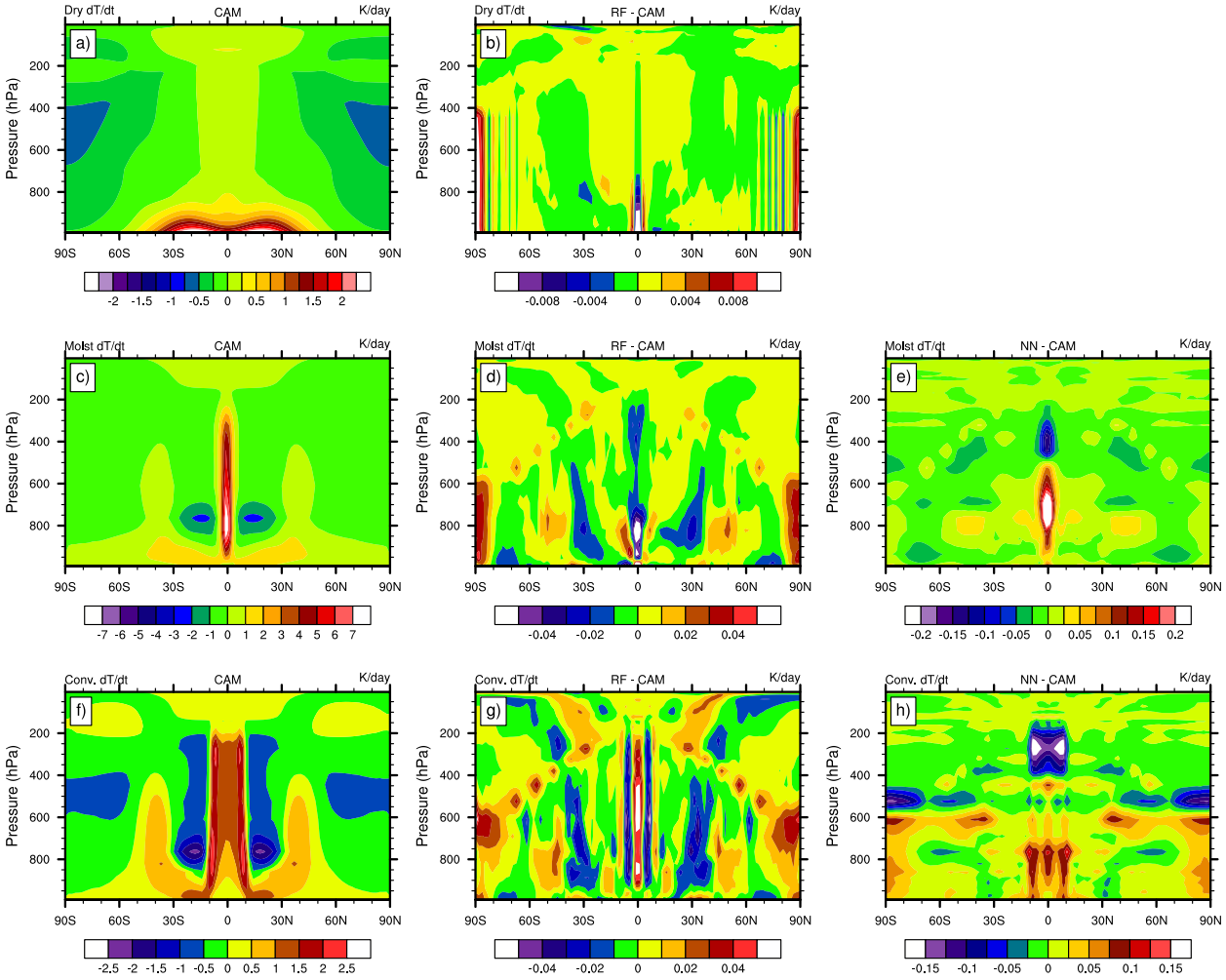



Figure 2.3: Zonal-mean time-mean temperature tendency output from CAM6 and the ML anomalies over the full testing data set. Ordered by dry (top), moist (middle), and convection (bottom) cases; left column is CAM6 output, middle column is RF difference, and right column is NN differences. The maxima in (d), (e), and (g) are around 0.12, 0.32, and 0.07 K/day, respectively, while the minimum in (h) is around $-0.19$ K/day. These were left out in order to avoid over-saturating the contours.

with consistent color schemes and, whenever possible, similar scales on the color bars. In

some instances this makes it infeasible to capture the true min/max range or to utilize the same scales for various plots within a given panel. For these cases, we note the maxima and/or minima in the captions for completeness.

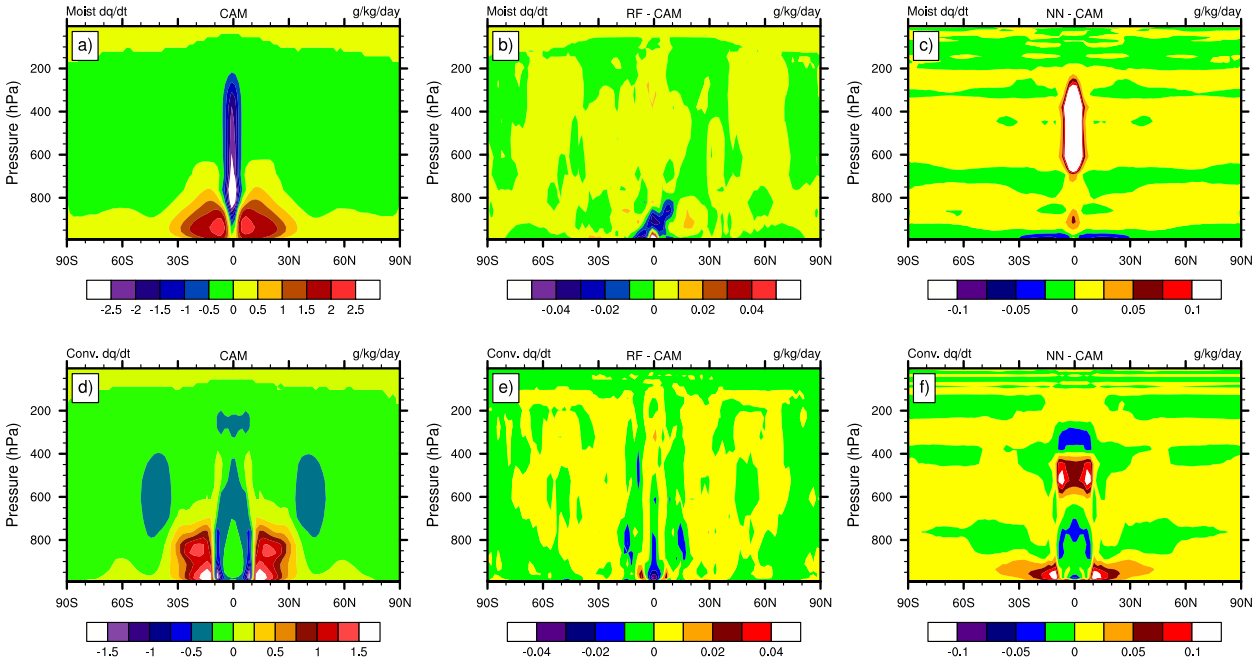



Figure 2.4: Zonal-mean time-mean moisture tendencies over the full testing data set for the (top) moist and (bottom) convective cases: (left) CAM6 output, (middle column) RF ML predictions, (right) their differences. The minimum in (a) is around $-3.6$ g/kg/day and the maximum in (c) is around 0.46 g/kg/day, but were left out in order to avoid over-saturating the contours.

Figures 2.3 and 2.4 show zonally and temporally averaged temperature and specific humidity tendencies over the testing period of the final six years from the CAM6 physics, along with the RF and NN anomalies in the mean fields. The differences calculated in all plots are truth (CAM) subtracted from the ML predictions, meaning that positive and negative values correspond to over- and underestimations by the ML scheme, respectively. The magnitude of the RF differences (middle column) is insignificant relative to the tendencies for all three cases, which is especially true for the dry configuration as seen in Figure 2.3b. It is also worth noting that the NN predictions show an order-of-magnitude increase in relevant range on the mean anomalies over the RF predictions in Figures 2.3 and 2.4. The NN predictions in both moist tendencies (Figures 2.3e & 2.4c) show large regions of relatively large magnitude differences in the tropical regions, something that is not apparent for the corresponding RF results. Furthermore, there are symmetric error patterns in the RF case in Figures 2.3d and 2.3g, showing peaks near the equator and the poles, as well as large overshooting regions

in the midlatitude upper atmosphere, tapering off towards the poles and lower atmosphere. This pattern also seems to be amplified in the convection case with regard to the spatial extent and magnitude of the error pattern. Aside from the largest differences occurring closer to the equatorial region near the surface, the RF specific humidity difference plots in Figures 2.4b,d do not show the same discernible pattern.

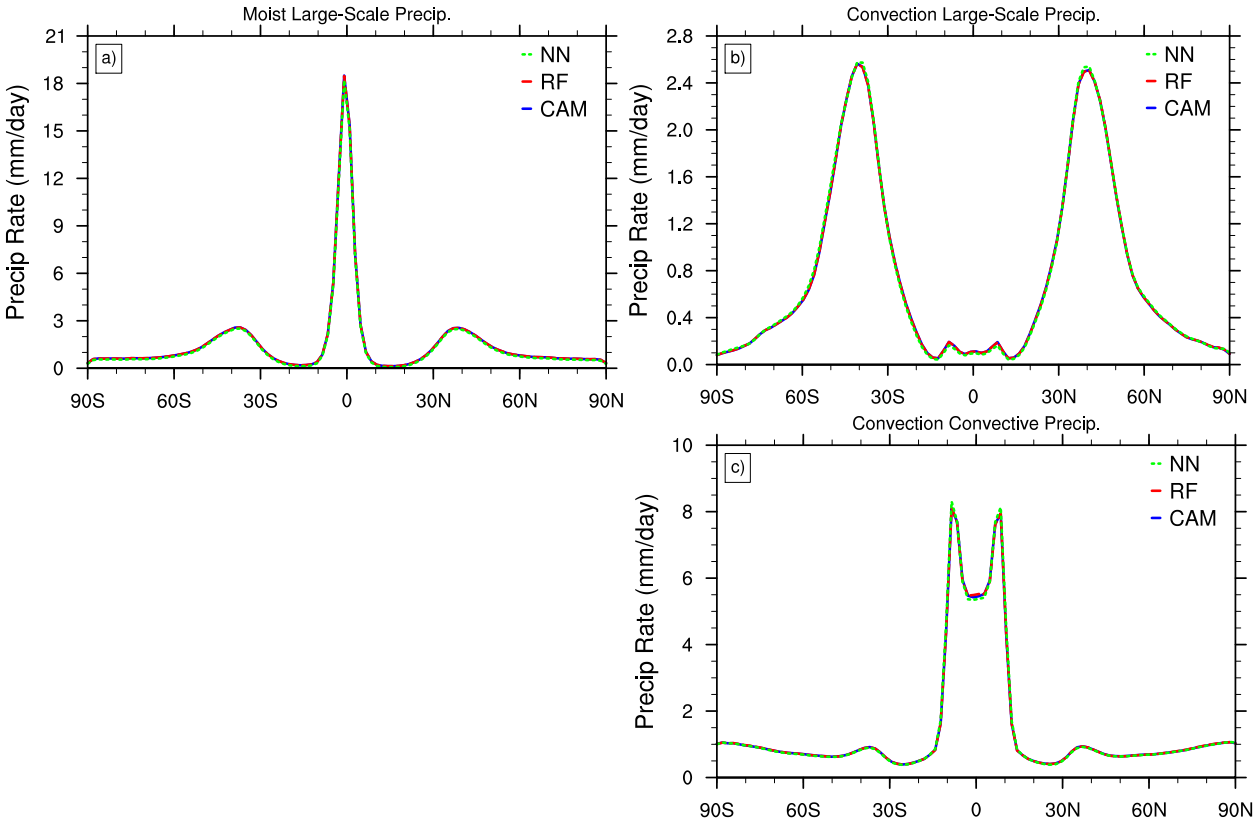



Figure 2.5: Zonal-mean time-mean precipitation rates of CAM6 (blue), RF prediction (red), and NN prediction (green) over the full testing data set for the (top) large-scale precipitation (Equation 2.5) and (bottom) convective precipitation; (left) moist case, (right) convective case.

Figure 2.5 displays the same averaged field for the precipitation rates. The CAM6 output (blue) and both of the ML predictions (green and red) overlay each other almost perfectly. The top row shows the large-scale precipitation rate and the bottom row the convective precipitation rate, while the left column corresponds to the moist case and the right to the convection case. The precipitation rate patterns mirror the same physical characteristics that are displayed in the time snapshots in Figures 2.1 and 2.2 and, even more pronounced, in the climatologies in Figures 2.3 and 2.4. For example, the temperature frontal zones and their moisture tendencies in the midlatitudes lead to heating bands around 40°N and 40°S in Figures 2.3c and 2.3f. These regions correspond to the large-scale midlatitudinal

precipitation peaks in Figures 2.5a 2.5b. In addition, the intense precipitation regions near the equator (moist case) and the tropics-subtropics (convection case) are emulated well by the RFs as displayed in Figures 2.5a and 2.5c. These precipitation patterns are correlated with the intense tropical and subtropical heating peaks in Figures 2.3c,f and the negative moisture tendencies in Figures 2.4a,d.

The minor differences between the ML predictions and the CAM6 output in the snapshot figures (Figures 2.1,2.2) somewhat mirror minor artifacts that could arise through other common numerical changes to a GCM, such as dynamical core grid choices or diffusion settings. Further, when we incorporate the zonal-mean time-means in Figures 2.3, 2.4, and 2.5 these subtle discrepancies disappear, as we would expect. We also begin to see a hint that as we increase the complexity of the schemes, the RF's skill begins to decrease. As noted before, the similar temperature tendency error pattern in Figure 2.3d for the moist case is significantly more pronounced for the convection case in Figure 2.3g. This effect is not as apparent in the RF specific humidity error patterns in Figures 2.4b and 2.4e.

In Figure 2.5, the emulated precipitation rates are even less distinguishable in the mean fields. The various peaks in the zonal-mean time-mean plots in Figure 2.5 align closely with the areas of 'drying' in Figure 2.4. This is in particular true for the equatorial region in both cases, dominant in the moist case, as well as in the midlatitudes in the convection case. We also notice that there is not a noticeable difference in performance between the moist and convection cases' large-scale precipitation emulator in this metric. This is due to the fact that by adding the BM convection scheme to the moist physics, we do not impact the calculation of the large-scale precipitation. Instead, the resulting large-scale precipitation rate in the convection case is impacted only by the fact that the convection scheme, which is called first, has already removed a significant amount of moisture from the atmosphere. Therefore the overall amount of precipitation that accumulates from the large-scale scheme is less and more concentrated in the regions that did not meet the criteria for convection as described in the BM scheme. Mathematically, the large-scale precipitation scheme has not changed and we can see that the RF maintains its skill across the two schemes.

### 2.3.2 Point-wise Comparison

Next, we show one-to-one scatter plots of the results from CAM and the RF emulator in Figures 2.6 and 2.7. They depict the temperature and specific humidity tendencies at the model level closest to 850 hPa, and the precipitation rates, respectively. This is a metric that allows for an effective visualization of the spread of the predictions. If the emulator were to produce the exact results as the CAM model, the points on these plots would follow the

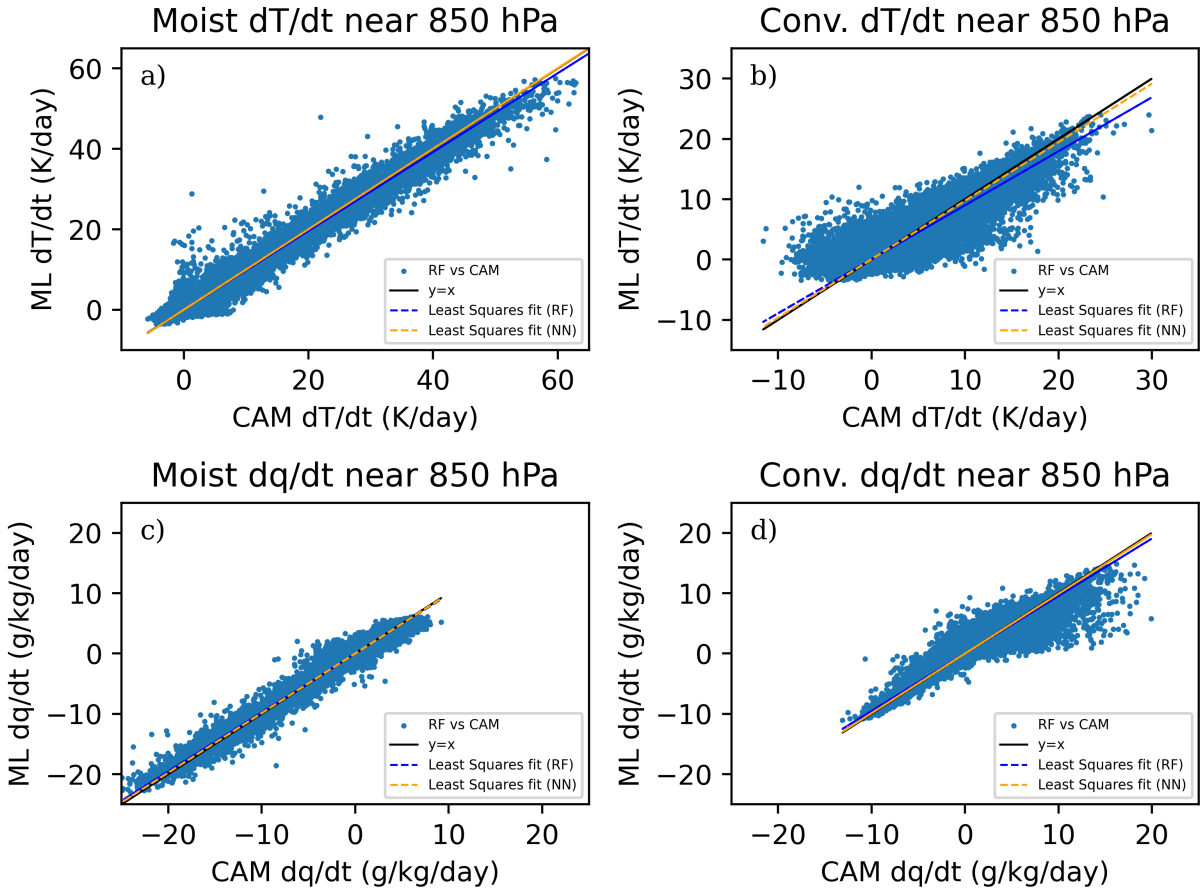



Figure 2.6: Scatter plots for RF predicted values (y-axis) against CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for (a) moist-case temperature tendency, (b) convection-case temperature tendency, (c) moist-case moisture tendency, and (d) convection-case moisture tendency.

one-to-one line $y = x$, shown in black. One-to-one scatter plots have been shown in related papers, such as O'Gorman and Dwyer [2018], Rasp et al. [2018], and Han et al. [2020] for various metrics and fields. Figure 2.6 contains the temperature tendencies in the top row and the moisture in the bottom row for both the moist case (left column) and convection case (right column). Figure 2.7 shows the scatter plots for each precipitation rate, oriented in the same configuration as Figure 2.5. Each scatter plot also contains the $y = x$ (one-to-one) line (solid black) along with least squares linear fits for RF (blue dashed) and NN (orange dashed). The least squares fit is calculated via the Python library NumPy and is used here to illustrate how closely the predictions align with, or deviate from, the $y = x$ line. An additional scatter plot is shown for the moist specific humidity case in Figure 2.8, which is identical to Figure 2.6c but with the NN results (y-axis) shown on the scatter plot rather

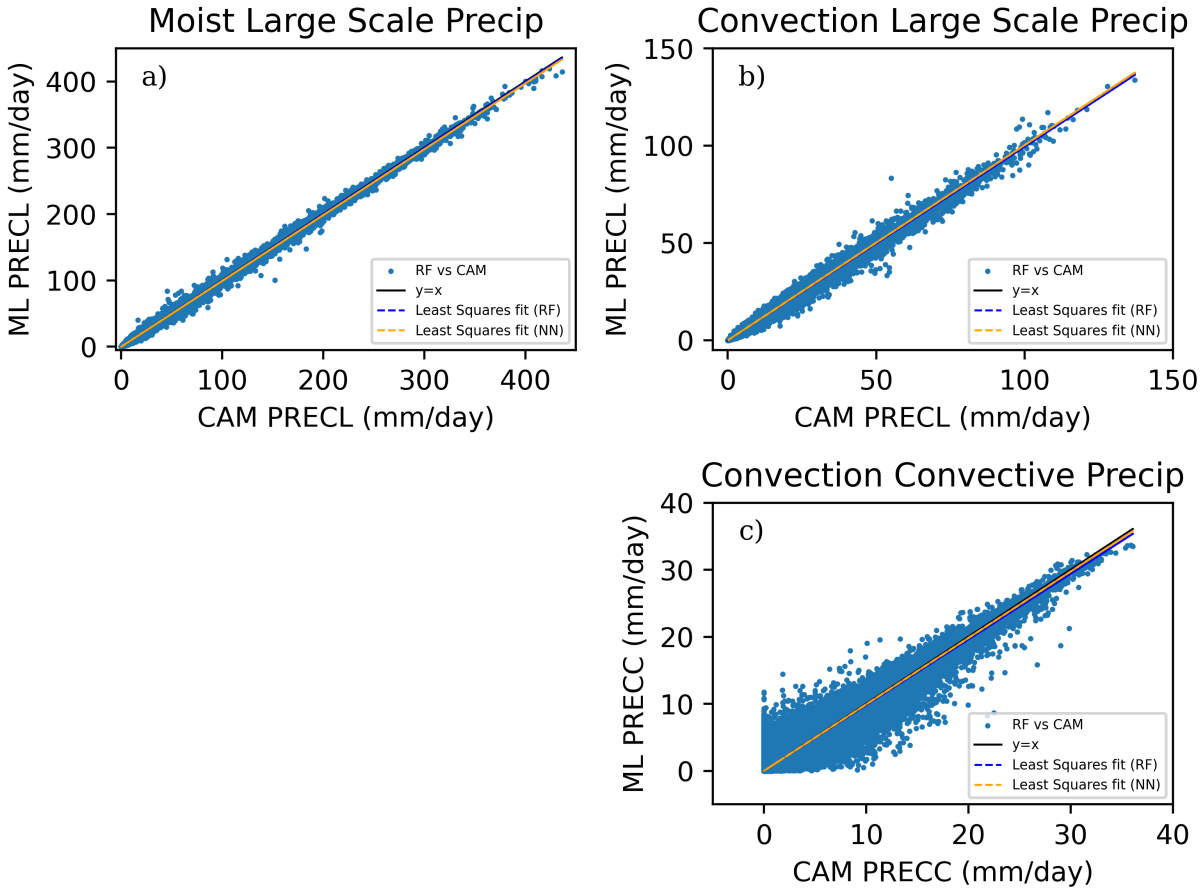



Figure 2.7: Scatter plots for RF predicted values (y-axis) against CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for the (a) moist-case large-scale precipitation rate, (b) convection-case large-scale precipitation rate, and (c) convection-case convective precipitation rate.

than the RF results. We show this for completeness and as an example of how the spread in the distribution is improved when using NNs rather than RFs, something that is also depicted in each plot's least squares fits for the level near 850 hPa. Across all cases the NN least squares fit at 850 hPa is closer aligned to the $y = x$ line. It is worth noting that had this analysis been for a level closer to 500 hPa, the spread in Figure 2.8 is more significant, as we see more frequent anomalies in these model levels near the equator as shown in Figure 2.4.

We also include a panel of histograms in Figures 2.9 and 2.10 corresponding to the same case orientation as Figures 2.6 and 2.7, respectively. In the histograms $N$ denotes the total number of test data points at the model level closest to 850 hPa or the surface (precipitation rates). These are plotted on a log-scale in order to better visualize the histograms, since the

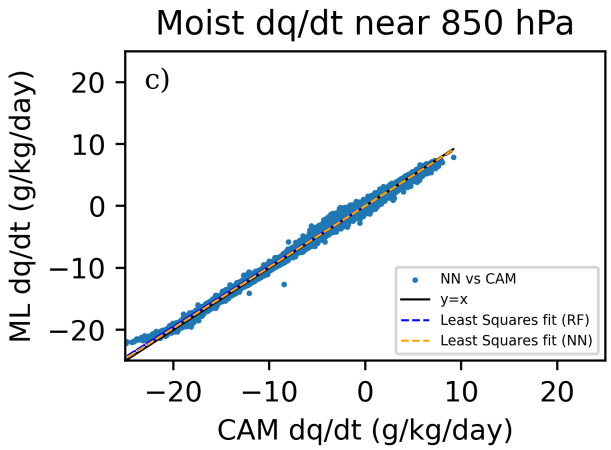



Figure 2.8: Scatter plot for NN predicted values (y-axis) against CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for moist-case moisture tendency

data are saturated around the central bin (minimal error), corresponding to the $y = x$ lines in the scatter plots. The histograms were inspired by the findings in Han et al. [2020] and help to illustrate how our scatter plots are dominated by points that fall along the $y = x$ line. Taking into account the difference between the displayed metrics and model configurations, our results with the one-to-one scatter plots show highly skillful ML emulators, in line with, if not superior to, what is reported in the literature for similar work.

For both of the large-scale precipitation rate emulators in Figures 2.7a,b, the $y = x$ and least-squares fit lines overlap almost completely with the one-to-one line. The plot of the convective precipitation rate 2.7c shows the most visual spread among the precipitation rate scatter plots. Along these same lines, both tendencies in Figures 2.6 and 2.9 display significantly more spread in the convection case over the moist case. This again shows that the enhanced complexity and nonlinearity of the convection process challenges the RF emulation and allows enhanced spread and biases as displayed by the scatter plots in Figures 2.6b,d and 2.7c. In addition, the specific humidity histogram in Figure 2.9d clearly indicates that the magnitude of the outliers increases in the convection case in comparison to the moist case (2.9c). The distribution gets wider in the convection case. However, all of the histograms in Figures 2.9 and 2.10 also highlight that the overwhelming majority of the point-wise differences fall within the first few bins close to the zero center point. The black dashed lines convey the percentage of instances contained within them. Each case indicates at least 95% of the data within the black dashed lines, and in some cases over 97%, as indicated in the legends. This shows that while outliers occur, they are extremely rare. We cannot judge from this study whether these rare occurrences will have a significant impact on emulator performance if coupled to a climate model in an online mode. However, this is

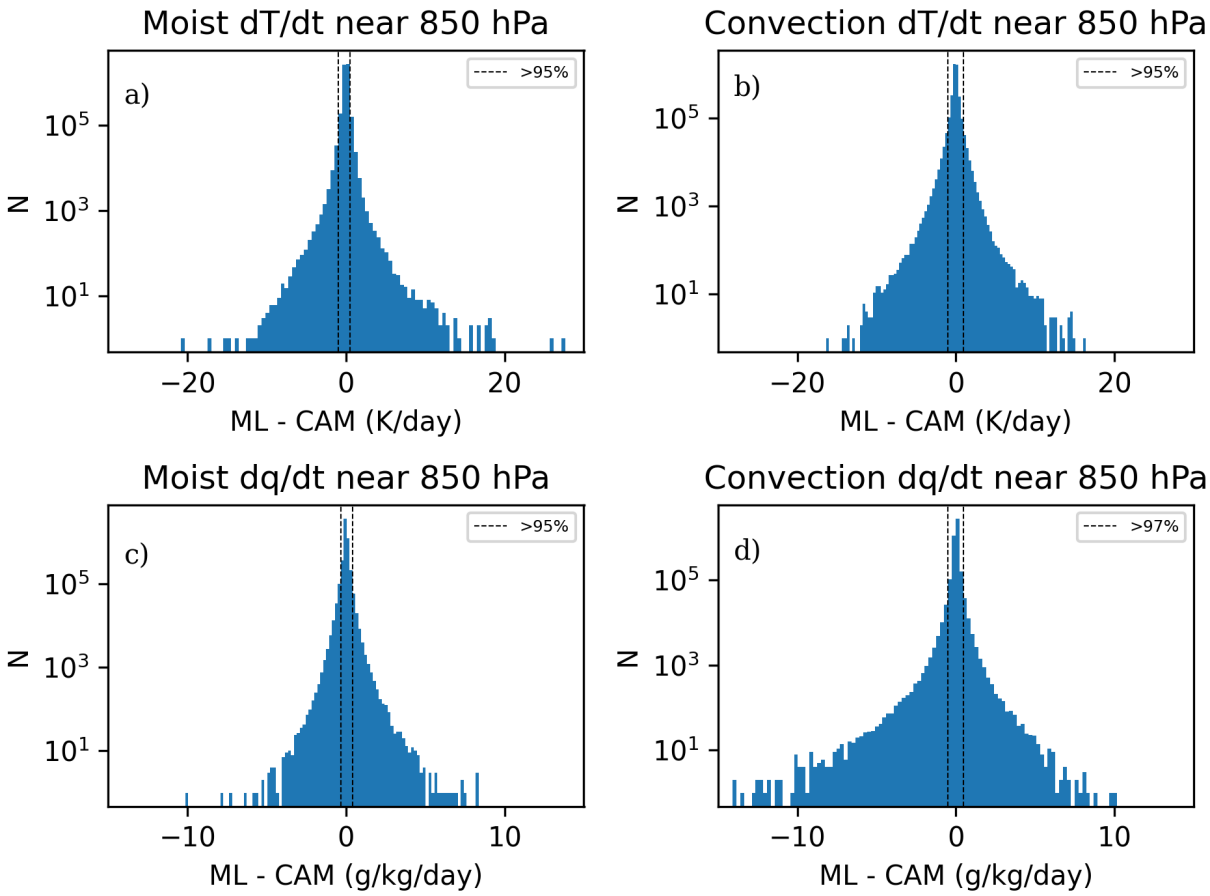



Figure 2.9: Histograms of the point-wise difference (RF - CAM6) for the temperature (top) and specific humidity (bottom) tendencies, corresponding to the scatter plots in Figure 2.6 on a log scale using 100 bins. Percentage of data contained within the black dashed lines are indicated in individual legends.

an aspect will need to be assessed in the future. The plots that show a deviation in the fit from the $y = x$ line appear to have a slight bias to underestimate the extreme precipitation. This is due to the inability for an RF to predict a value that is not within the range of its training data set, as discussed in Section 2.2.2 and is a significantly rare, albeit expected, occurrence.

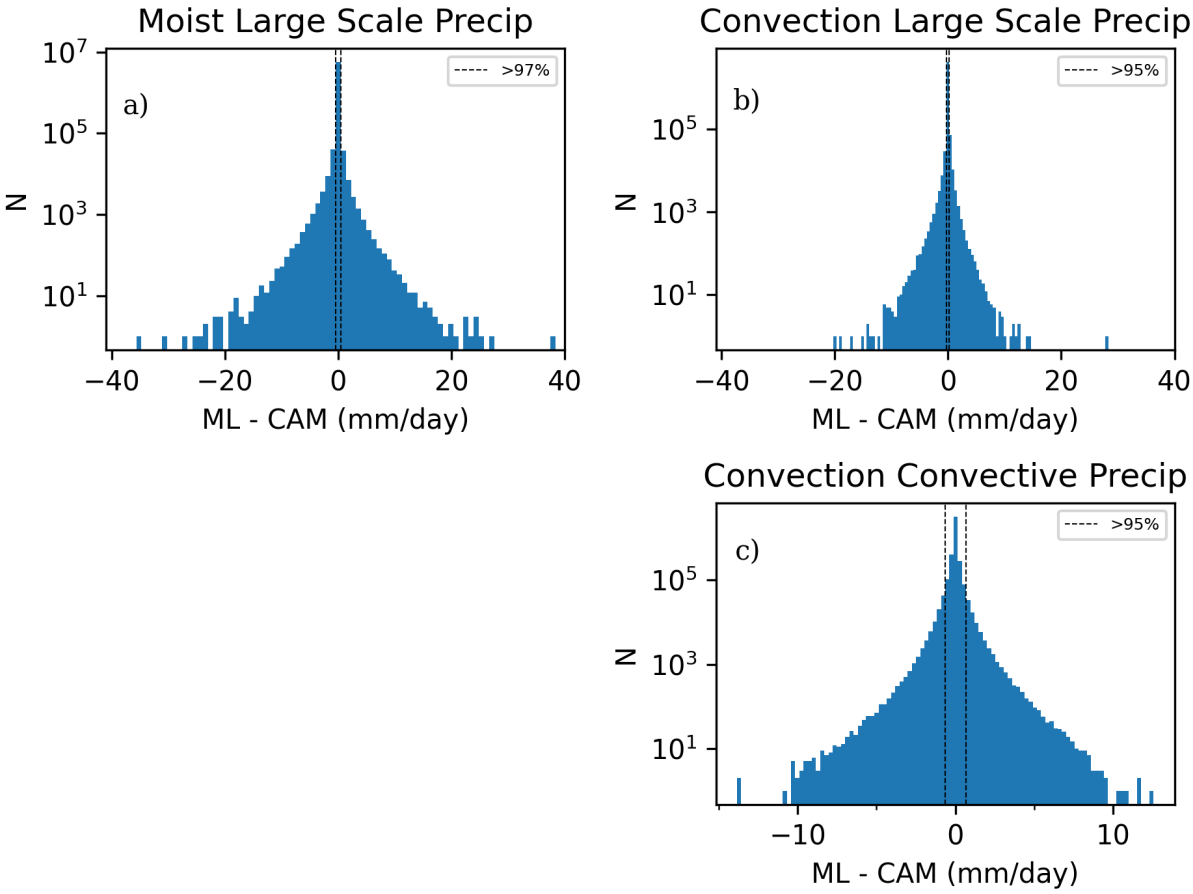



Figure 2.10: Histograms of the point-wise difference (RF - CAM6) for the precipitation rates corresponding to the scatter plots in Figure 2.7 on a log scale using 100 bins. Percentage of data contained within the black dashed lines are indicated in individual legends.

### 2.3.3 $R^2$ Investigation

Another performance metric is the coefficient of determination, or, $R^2$. We calculate $R^2$ contours over the time and zonal dimensions, given by the formula

$$R^2(:,:) = 1 - \frac{\sum_t \sum_\lambda [\mathrm{CAM}(t,:,:,\lambda) - \mathrm{ML}(t,:,:,\lambda)]^2}{\sum_t \sum_\lambda [\mathrm{CAM}(t,:,:,\lambda) - \overline{\mathrm{CAM}}(:,:)]^2} \tag{2.9}$$

where $\lambda$ is the longitudinal dimension, the numerator is referred to as the residual sum of squares and the denominator is the variance of the CAM6 output. The average in the calculation, indicated by $\overline{\mathrm{CAM}}$, is a zonal-mean time-mean over the testing data set. $R^2$ can simply be understood as a measurement of how well a regression model has learned the functional relationship between the input and the predicted output based on the true

output. The closer to one, the better the $R^2$. It should be noted here that the $R^2$ can take negative values whenever the errors in the predictions are larger than the variance in the original data. In general, this may be interpreted as a model that cannot identify, or has not 'learned', the functional relationships at play. This approach was inspired by Figure 1 and 7 in O'Gorman and Dwyer [2018], wherein the author shows a panel of $R^2$ contours for temperature tendencies for various training scenarios also using RFs to emulate the tendencies.

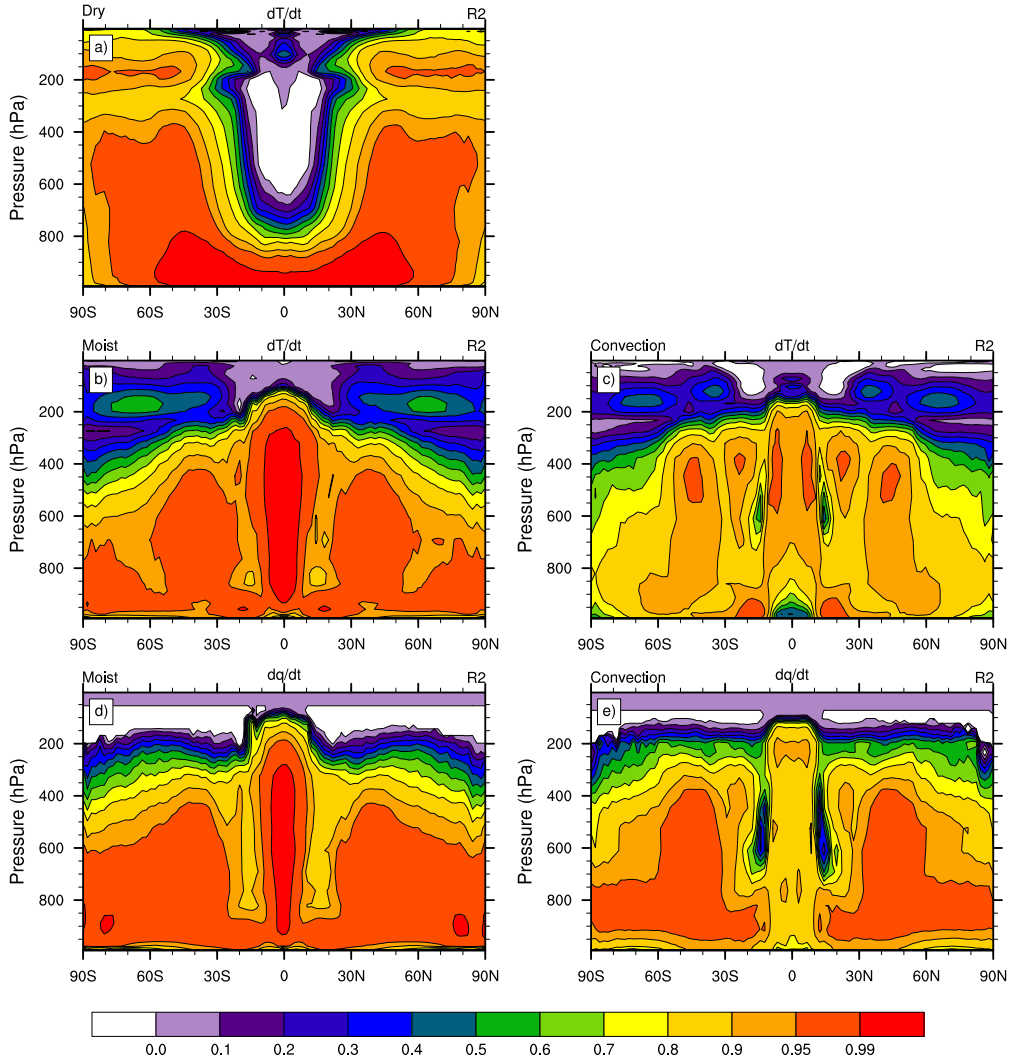



Figure 2.11: $R^2$ calculations over the zonal and temporal dimensions for RF emulators of (a) dry temperature tendency, (b) moist temperature tendency, (c) convection temperature tendency, (d) moist moisture tendency, and (e) convection moisture tendency via Equation 2.9.

We display a panel of $R^2$ plots for all of our tendencies in Figures 2.11 and 2.12 and precipitation rates in Figure 2.13. All of the predicted fields and tendencies show large

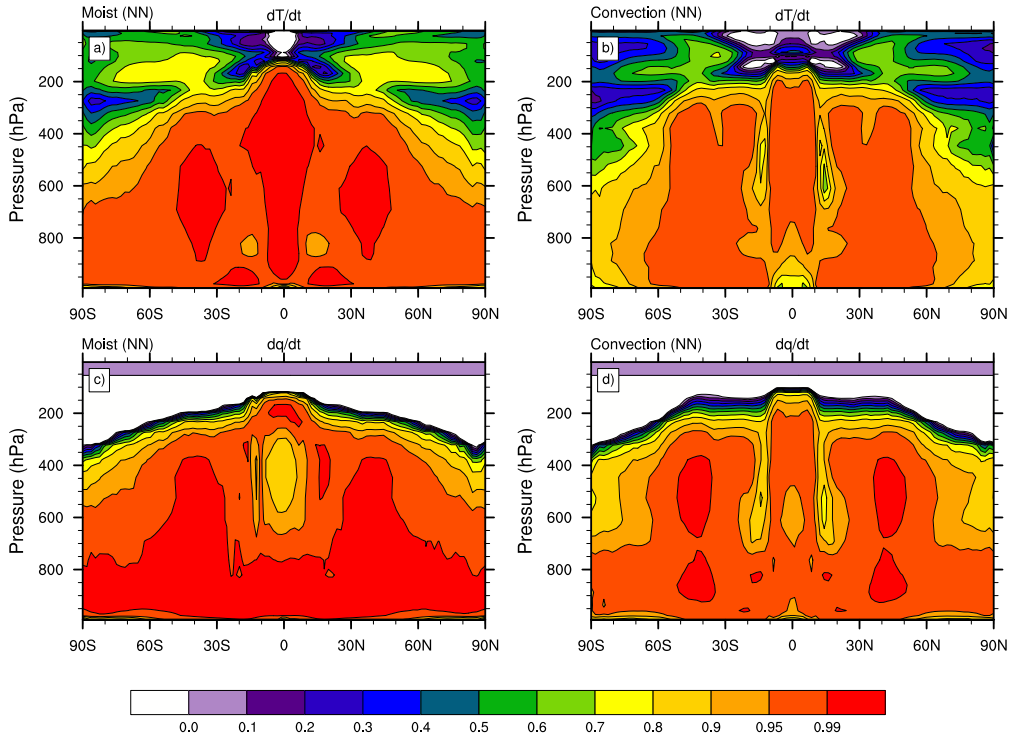



Figure 2.12: $R^2$ calculations over the zonal and temporal dimensions for NN emulators of (a) moist temperature tendency, (b) convection temperature tendency, (c) moist moisture tendency, and (d) convection moisture tendency via Equation 2.9.

regions of highly skilled emulators with at least $R^2 > 0.7$. Our trained emulators show skill in line with various other examples of similar published work. Examples are O'Gorman and Dwyer [2018] and Yuval and O'Gorman [2020] who investigated RF emulators for physical parameterizations via idealized aquaplanet model configurations. While the work in this paper is not meant to be a direct comparison to their findings due to the differences in the atmospheric model designs and RF emulation strategies, it is worth highlighting the similarities of the $R^2$ patterns.

The $R^2$ panels in Figures 2.11, 2.12 and 2.13 reveal a wide variety of aspects. For example, as we increase the complexity of our system, the RF's global effectiveness decreases with regards to the $R^2$ skill. Excluding Figure 2.11a, from left-to-right we increase in complexity from the moist case to the convection case, and in doing so we notice the impact on the $R^2$ skill globally. In Figure 2.11c there are broader regions of $\mathrm{R}^2 \leq 0.5$ in the upper atmosphere than in Figure 2.11b. Similarly, two pockets of $\mathrm{R}^2 \approx 0.3$ form around the tropics in Figure 2.11e, which were not nearly as pronounced in Figure 2.11d with $\mathrm{R}^2 > 0.7$ in these regions. This region is associated with tropical convection as shown in Figure 2.5c and also is present in the dips in $R^2$ for convective precipitation (blue lines) in Figure 2.13. For all precipitation

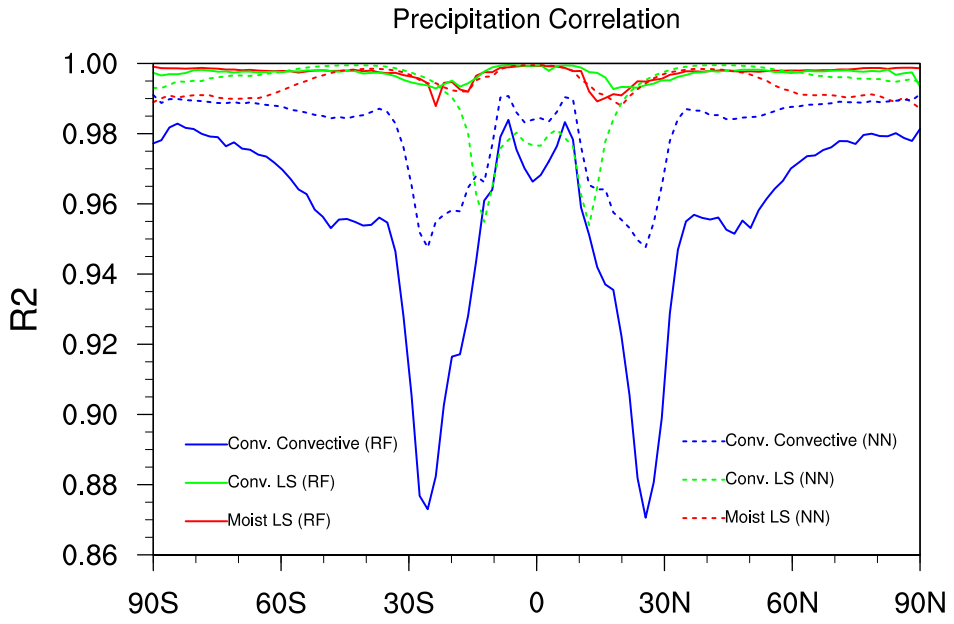



Figure 2.13: $R^2$ calculations over the zonal and temporal dimensions via Equation 2.9 for ML predictions of moist large-scale precipitation (red), convection large-scale precipitation (green), and convection convective precipitation (blue); NN results are dashed lines, RF results are solid.

cases, we see slight dips in $R^2$ in the regions where the majority of the convection occurs, primarily within the tropics or near-tropics. This dipping is most pronounced for the convective precipitation scheme, that accounts for the majority of this region's precipitation and is inherently more complex than the large-scale precipitation scheme. For the moist large scale precipitation (red lines in Figure 2.13), we see almost-overlapping performance around an $R^2 = 0.99$. In the convection case, there is shown to be more variability between the RF and NN approaches. For the large scale precipitation (green), the RF appears to be more skillful, consistently around $R^2 = 0.99$, than the NN, which shows a relatively significant dip in the tropics. The opposite is shown for the convective precipitation, where in there is the most significant dip in performance across all cases for the RF. The NN, however, remains more skilful across the entire domain, even with its own tropical dip in performance. That being said, across both cases and ML emulators, the precipitation results in Figure 2.13 are impressive when compared with $R^2$ values from the physics tendency results (Figures 2.11 & 2.12). This is likely due both to the fact that these are surface fields, as well as their having less complex mathematical representations.

Figure 2.12 shows the $R^2$ panel with regards to our NN emulators, which show a noticeable increase in skill over the RF in almost every case. This is not particularly surprising, since NNs are known to be a more robust ML technique versus RFs. We note here that there

is some evidence of the NNs also noticeably decreasing in skillfulness as we increase in complexity from the moist case to the convection case, however we recall the earlier discussion on the fact that our NNs were not uniquely tuned for each case. It is possible that further turning of hyperparameters/NN architecture might bring the convection results in line with the moist results.

We also note that the $R^2$ calculation can be an unreliable metric in regimes where there is minimal activity. This occurs in the white regime of Figures 2.11a,c,d,e. In these regions the variance in the denominator and the sum of squares in the numerator (see Equation 2.9) are both functionally zero. However, they are still seen as floating point numbers of extremely small order and Equation 2.9 can lead to various misleading results such as

$$R^2(:,:) \approx 1 - \frac{10^{-6}}{10^{-13}} \approx 1 - 10^7 << 0 \tag{2.10}$$

or

$$R^2(:,:) \approx 1 - \frac{10^{-11}}{10^{-11}} \approx 1 - 1 = 0 \tag{2.11}$$

For the dry case in Figure 2.11a, this occurs in the tropics in the mid-atmosphere. Similarly, this occurs in the upper atmosphere for the moisture tendencies in Figures 2.11d and 2.11e. In the dry case there is, on average, very little heating or cooling in the mid-to-upper tropics. Similarly, the moist and convection cases experience very little temperature and moisture forcing at the upper levels as also displayed by the climatologies in Figures 2.3 and 2.4. However, due to the nature of floating point numbers the $R^2$ calculation identifies these regimes as areas of poor skill. This is an example of a weakness in $R^2$ as a metric of regression skill, rather than a reflection of a weakness in the ML model for these particular cases.

### 2.3.4 Skill Variation

Various aspects of the ML training process impact the skill of our emulators. A common example of this is the idea of feature importance. Feature importance is the investigation into the relative importance of various input parameters for the skillfulness of an ML model. In order to maximize the training and inference performance of emulators, it is important to only include useful predictors into our feature set. We know what input fields are used to calculate the parametrizations that we emulate, as discussed in section 2.2.1. These tend to include, for example, the temperature, pressure, latitude, and surface heat fluxes. One input field that we investigate more closely is relative humidity (RH). Since RH is not an explicit variable used in calculating the physics tendencies and precipitation rates, would including

it improve performance? Figure 2.14 shows the $R^2$ comparison of explicitly including the RH (left) and not including it (right). This assessment uses identical RF setups, trained independently, for the moist specific humidity tendency. The RF shows skill without the inclusion of the RH field. However, it is significantly improved upon with the inclusion of the RH.

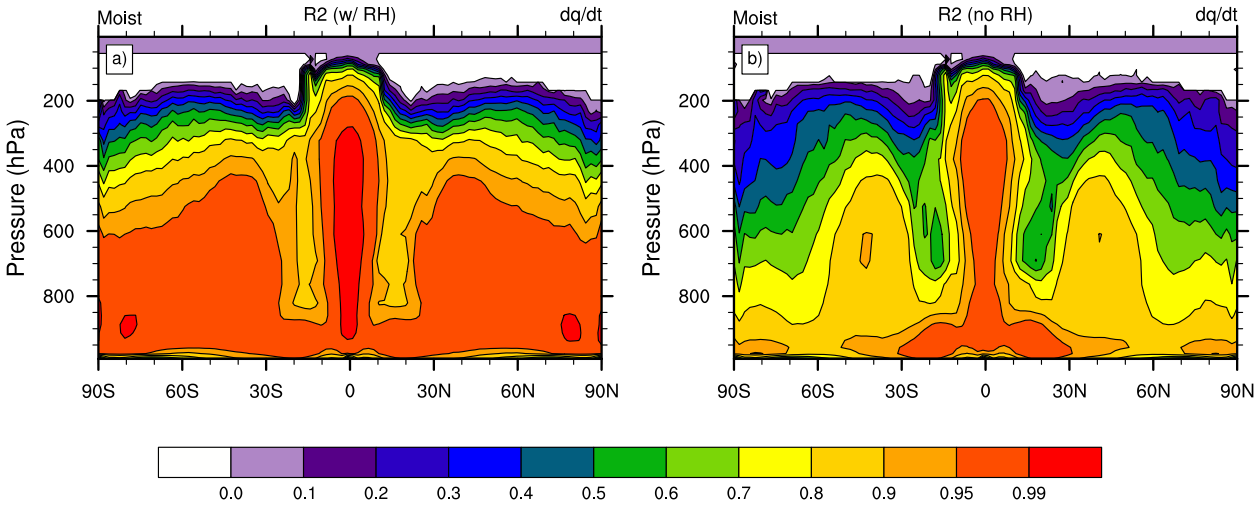



Figure 2.14: Comparison of $R^2$ plot - as defined in Figure 2.11 - (a) with and (b) without relative humidity as a feature for RF prediction of the moisture tendency for the moist case. Figure 2.14a reproduces Figure 2.11d.

From a pure data science perspective, it may not be apparent that the RH field will improve the performance since it is not an explicit variable used in the functional form of the parameterization. From the atmospheric science perspective, this is to be expected since relative humidity is an important indicator of changing moisture levels in the atmosphere. It is also an indicator of supersaturation (RH>100%) in the large-scale precipitation algorithm. The large-scale condensation rate $C$ is only computed in supersaturated regions and then enters the computation of both the temperature and specific humidity tendencies. It thereby acts as a guide for the RF algorithm whether additional forcing mechanisms are present. This illustrates the importance of physical knowledge and intuition when designing ML algorithms.

We also assessed the dependence of the RF emulator on the number of training data. This is displayed in Figure 2.15 which shows the RF skill (as measured by the global-mean $R^2$ value) versus the number of samples (in millions). As we discussed before, our models use around 15 to 20 million training samples which is outlined in more detail in the Supporting Information Tables S1 to S8. When decreasing the number of samples we see a decrease in skill in Figure 2.15, as expected. It is also worth noting that the rate at which the skill

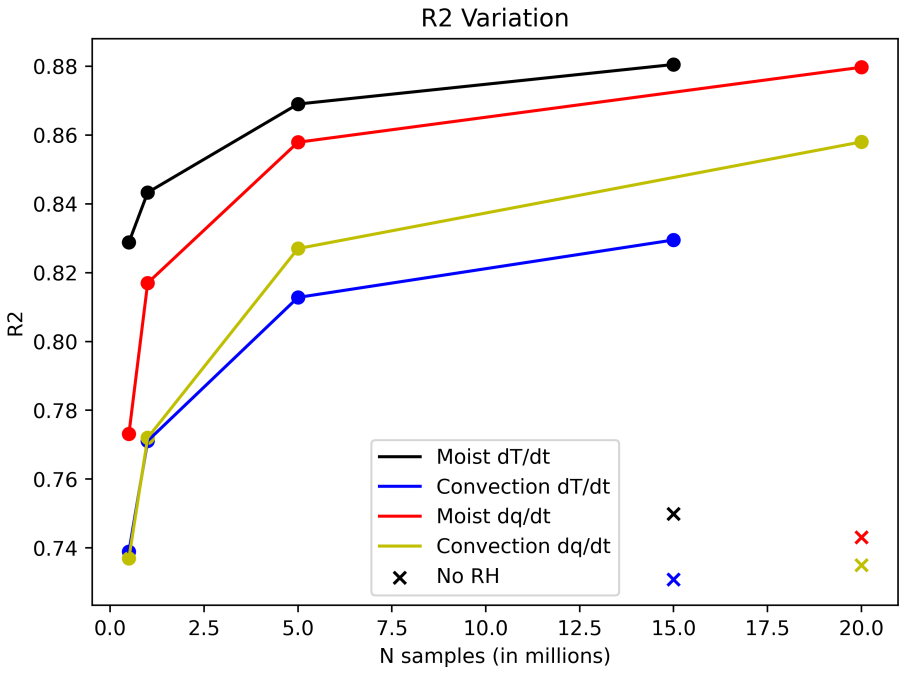



Figure 2.15: Globally-averaged $R^2$ value (y-axis) for RF prediction of the tendencies in the moist and convection cases as the number of data available for training is increased (lines), as well as when RH is removed as an input (crosses) using the maximum amount of training data. Note: to avoid saturation by large negative numbers (discussed in Section 2.3.3), these global $R^2$ values are calculated from the surface up to roughly 175 hPa.

decreases with respect to the number of samples appears fairly consistent across the various tendencies. In addition, there is an upward jump in the emulation skill when the sample size changes from $10^5$ to $10^6$. Figure 2.15 also includes the globally averaged $R^2$ values for selected RF emulators that do not include RH as a predictor. These are marked by the colored crosses. Similar to Figure 2.14, this shows that the emulators lose a significant amount of skill when RH is omitted. Furthermore, the skill of the convection case is always lower than the skill of the moist case without convection. This is true for both the temperature and moisture tendencies and does not depend on the number of samples or the inclusion/omission of RH.

## 2.4 Concluding Thoughts & Applications to Future Work

Individual RFs are configured and trained, along with baseline NNs, to emulate temperature tendencies, specific humidity tendencies, as well as large-scale precipitation and convective

precipitation rates. These tendencies are generated by physical parameterization packages that are based on three 'simple physics' model configurations within NCAR's CAM6 framework. The simple physics configurations are built upon one another and form a model hierarchy with increasing complexity. The hierarchy includes a dry case, a moist case, and the moist case with an added simplified convection scheme. Each CAM6 configuration generated training and test data for the ML emulators and were collected over a 60-year simulation period. In addition, the SHERPA hyperparameter optimization tool was used to optimize each RF configuration. This allowed us to create robust RF emulators in order to probe the characteristics of their skills in an offline configuration. The central question was whether, and how much, ML skill is lost when the complexity of the emulated physical processes is increased.

All of our emulators showed significant skill when tested on the test data over the final six years of the model output. Our RF emulators showed results at least as skillful as other similar examples within the literature, while in many cases outperforming similar work. However, in a majority of cases our climate model configurations were less complex than the examples from the literature. Therefore, direct comparisons are not possible. There are disadvantages to using RFs over other nonlinear regression techniques, like deep learning methods, such as their computational inefficiency, particularly when being ran on graphical processing units (GPU), as well as large memory requirements. This work demonstrated that RFs can be skillful for the prediction of averages but tend to struggle when faced with extremes. Additionally, deep learning methods are known to be more robust and extendable for complex systems. This was apparent in our exploration of a baseline NN emulator for comparison (Figure 2.12) and is an intriguing property since climate modeling includes highly complex physical processes. This demands scalable and computationally efficient approaches to ML emulators.

Our study suggests that there are likely limitations when using RF emulators for physical parameterizations, even within our highly simplified hierarchy of configurations. Clear decreases in the RF skill were exposed as the complexity of the physics scheme was increased, particularly in the case of whole-atmosphere tendency fields ($dT/dt$ & $dq/dt$) when compared to the baseline NN results. In the case of precipitation, however, the skill was in line with the NN approach. This raises interesting insights into when we can take advantage of the useful properties of RFs in the pursuit of data-driven improvements to modeling the Earth system. Balancing the trade-offs between physical realism, computational efficiency, and model complexity must inform the choice of ML technique, especially when looking forward towards state-of-the-art weather or climate model. Random forests are unlikely to remain as skillful as shown here for more complex physics packages. Our next step will be to couple the

emulators to the CAM6 implementation and analyze how they perform in an online mode. A particular interest will be whether the rare, yet present, outliers impact the stability of the coupled model, as well as the degree to which the computational demand of the ML models impact the CAM6 performance. This will continue to shed light on the question of where RFs may fit into the future of data science-augmented climate and weather models.

# CHAPTER 3

# Evaluating the Online Coupling of Machine Learning Emulators for Simple Physical Parameterizations in CAM6

## 3.1 Introduction

The integration of Machine Learning (ML) techniques into climate modeling has gained significant attention in recent years, particularly in the context of improving the efficiency and accuracy of climate simulations (Boukabara et al., 2021; Reichstein et al., 2019). One promising application is the use of ML-based emulators to replace traditional physical parameterization schemes within climate models. These parameterizations represent subgrid-scale processes, such as cloud formation, turbulence, and convection, that are crucial for simulating climate but are computationally expensive and often uncertain due to limited resolution. Using ML models to emulate these processes has the potential to be computationally more efficient, while maintaining the accuracy of the simulations.

However, the online coupling of ML emulators to complex climate models presents a series of significant challenges. First, the accuracy and reliability of ML models are inherently tied to the quality and quantity of training data. Climate models often operate in highly non-linear and chaotic environments, making it difficult to generate sufficient training data that are representative of the wide range of conditions encountered in long-term climate simulations. Moreover, ML emulators must be able to generalize well to unseen conditions, a task complicated by the complex and uncertain nature of many physical processes.

Second, integrating ML emulators with existing climate models introduces computational and operational challenges. Climate models, especially flagship models, are highly integrated computational systems, making them challenging to modify or interface with. Introducing an ML-based emulator as a replacement for traditional parameterizations can introduce additional complexity in terms of model architecture, library linking, memory management,

and maintenance. Ensuring that these emulators perform efficiently within the context of the larger model, without causing excessive computational overhead or introducing instability, is a critical concern. Since ML models, despite often showing high skill in offline emulation [Limon and Jablonowski, 2023], frequently experience instabilities and reduced performance when coupled to the dynamical core [Beucler et al., 2021, Yuval and O'Gorman, 2020].

Furthermore, the interpretability and transparency of ML models remain a significant challenge when applied to scientific domains such as climate modeling. Although ML models may provide more accurate results in some cases, their "black-box" nature makes it difficult to understand the underlying mechanisms and to verify their physical plausibility. This lack of transparency complicates their validation, particularly when it comes to ensuring that the models respect the fundamental principles of physics and climate dynamics.

Lastly, the transferability of ML emulators across different climate models, regions, and timescales adds another layer of complexity. An ML model trained on a particular climate model or a specific geographic region may not perform well when applied to a different climate model configuration or new atmospheric conditions. The generalization of ML emulators across these variations is desired for a widespread implementation in global climate modeling. The two most common findings are that it is a nontrivial process to couple ML models to atmospheric models and that once coupled performance can degrade and stability becomes a concern. For example, Beucler et al. [2021] notes the lack of NNs, in particular, to maintain physical constraints such as mass, energy, and radiation, as well as their inability to generalize beyond the configurations in which they were trained on.

On the contrary, Yuval et al. [2021] showed that for momentum flux transport emulations, the ML results were promising. The authors showed that momentum fluxes were conserved with the NN emulated parameterizations, and that the coupled system gives stable simulations. However, it is noted that while the emulator is skillful, it is not as effective at predicting the momentum flux as it is for the tendencies, such as temperature and moisture. Their results remain promising as they note improvements when implemented in coarse simulations, reducing biases and maintaining a stable simulation of the atmosphere.

Recent work by Connelly and Gerber [2024] explored random and boosted forests to emulate parameterizations of atmospheric gravity wave momentum transport. They evaluated the performance of these models both offline and online in an atmospheric model, comparing them to a neural network (NN) benchmark. The results indicated that boosted forests performed comparably to NNs, while random forests (RFs) showed lower accuracy, which was a similar finding featured in Chapter 2 and Limon and Jablonowski [2023]. Notably, the boosted forest model demonstrated stable coupling with the atmospheric model and exhibited reduced biases in control climate simulations compared to the NN. Furthermore, both

the boosted forest and NN successfully captured key stratospheric variability modes, such as the Quasi-Biennial Oscillation and Sudden Stratospheric Warmings, across various carbon dioxide emissions scenarios.

Despite these challenges, the potential benefits of incorporating ML-based parameterizations in climate models are considerable. The ability to significantly reduce computational costs, improve the resolution of simulations, and improve the understanding of complex climate processes provides a compelling incentive to continue exploring these approaches.

In this chapter, we extend our offline studies from the preceding chapter. Focusing on addressing fundamental questions regarding the feasibility of using simple ML for simplified parameterization emulation within a flagship climate model. Hence, the feasibility of this endeavor is determined by factors such as the ease of building and training the ML model, its offline performance, how easily the ML model can be integrated with the climate model, and the online skill degradation when the hybrid coupled model is used.

## 3.2 Methods

### 3.2.1 CESM and CAM Setup

Analogous to chapter 2, our research is conducted within the framework of the Community Atmosphere Model's (CAM) simple physics packages. CAM constitutes the atmospheric component of the Community Earth System Model (CESM), which is the preeminent Earth system model established at NCAR.

CAM offers an extensive array of physics schemes, each tailored for specific scientific inquiries. Among these is the category known as 'simple physics' packages, predominantly utilized as numerical test cases during the model development phase. These packages are particularly suited for foundational investigations into the application of ML to emulate aspects of CAM, providing an opportunity to utilize a hierarchy of parameterization schemes.

Our aim is to emulate the physical tendencies from the Thatcher-Jablonowski (TJ) parameterization scheme and to couple these into CAM [Thatcher and Jablonowski, 2016]. The TJ scheme is designed as a moisture-inclusive variant of the arid dynamical core test case developed by Held and Suarez [1994] (referred to here as [HS]). In the TJ scheme, the model's wind patterns are subject to Rayleigh friction in the lower troposphere, serving as an analogue for terrestrial friction and the mixing of momentum attributable to the Planetary Boundary Layer (PBL). The Rayleigh friction is formalized as follows:

$$\frac{\partial \vec{v}_h}{\partial t} = -k_v(p)\,\vec{v}_h. \tag{3.1}$$

where $\vec{v}_h$ symbolizes the horizontal wind vector and $k_v$ is the Rayleigh friction coefficient that depends on the pressure, $p$. In addition, radiation is mimicked by a Newtonian temperature relaxation, along with terms associated with heating and cooling via large-scale condensation, latent and sensible heat fluxes, and a PBL mixing scheme for temperature and moisture via a second-order diffusion mechanism. Further details of the TJ moist physics package are provided in Thatcher and Jablonowski [2016]. This culminates in the TJ temperature forcing taking the form

$$\left(\frac{\partial T}{\partial t}\right)_{\text{TJ}} = -k_T(\phi, p)\left[T - T_{\text{eq}}(\phi, p)\right] + \frac{L}{c_p}C + \frac{C_H|\vec{v}_a|(T_s - T_a)}{z_a} + \text{PBL Diffusion} \tag{3.2}$$

Here, $\partial/\partial t$ represents a sub-grid physics tendency (forcing) of a variable over a physics time step, $\phi$ denotes the latitude, $\vec{v}_h$ is the same horizontal velocity vector as equation 3.1, $T$ stands for the temperature, $T_{\text{eq}}$ is a pre-defined equilibrium temperature profile, $k_T$ is the dissipation coefficient, with the inverse time unit $\text{s}^{-1}$, $L$ is the latent heat of vaporization, $C$ is the large-scale condensation rate, $c_p$ is the specific heat at constant pressure, $C_H$ is the transfer coefficient for sensible heat, $|\vec{v}_a|$ is the horizontal wind speed at the lowest model level, $T_s$ is the surface temperature, $T_a$ is the temperature of the lowest model level, and $z_a$ is the height of the lowest model level. The latter five are needed for the computation of the sensible heat flux at the surface. The details of the PBL temperature diffusion algorithm are provided in Thatcher and Jablonowski [2016] and Reed and Jablonowski [2012].

We also account for specific humidity forcing, which is similarly impacted by these physical processes. The forcing takes the form

$$\left(\frac{\partial q}{\partial t}\right)_{\text{TJ}} = -C + \frac{C_E|\vec{v}_a|(q_{\text{sat},s} - q_a)}{z_a} + \text{PBL diffusion} \tag{3.3}$$

where $q$ refers to the specific humidity, $C_E$ is the bulk transfer coefficient for water vapor, $q_{\text{sat},s}$ is the saturation specific humidity at the surface, and $q_a$ is the specific humidity at the lowest model level, as outlined in Thatcher and Jablonowski [2016].

For this project, we utilize a similar model setup as Chapter 2, in that we run the finite-volume dynamical core in CAM6 at the same 2-degree resolution for the TJ setup for 60 years with tendency and state variables output every five days. We use the first 50 years of the simulation to train our ML models, this also includes validation data (subsampled from the training data), then a brief gap of four years, with the testing data being the final six years of simulation. When coupling our ML models, we run the model starting from the end of the 60-year initial simulation, aiming for a six month simulation with the ML forcings. In the case of coupled models, this simulation period varies across testing domains, as issues

with memory management and model stability arose. various results will be explained and discussed in further detail across Section 3.3.

### 3.2.2 Machine Learning and Coupling Techniques

As discussed in Chapter 2, our approach to emulating simplified atmospheric parameterization schemes involves the use of both RFs and NNs. NNs offer the advantage of greater computational efficiency, enabling faster simulations, whereas RFs more robustly maintain the physical realism within their training data. In this chapter, we shift focus to the various coupling techniques we tested within CAM, which are critical for integrating ML models into the existing framework.

Additionally, we made one significant change to how we configure our ML emulators. In Section 2.2.3, we discuss how we configured our emulators for uniquely each tendency for each case. We still utilize the same input features for our emulators, temperature, pressure, moisture, relative humidity, and latent and sensible heat fluxes. In this work, however, we chose to train a single ML emulator (one RF and one NN) to predict both the temperature and moisture tendencies together. We did this by stacking both our input features (state variables) and output labels (tendency fields) in the model's 'level' dimension. Essentially, as described in section 2.2.3, our $N_{outputfields} = 2$ here, for using both tendencies. This allowed for a more seamless implementation of our emulators into the CAM model. The only major change we made to hyperparameters was to reduce the parameter 'max_depth' to 24 for our RFs. This parameter significantly impacts the disk size of our trained RFs. The choice of 24 was our best attempt at balancing the trade-off between RF skill and RF size, since reading in these large files into memory within CAM is a significant concern in this work. While offline skill is not discussed here in depth, it is inherently showcased in section 3.3.1 when the ML results are ran in CAM's Fortran code, while not actively forcing the model's physics (see section 3.3). The chosen hyperparameters for our RFs and NNs used in this chapter are show in Tables 3.1 and 3.2, respectively.

Table 3.1: Random Forest Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 2 Million |
| Number of Trees | 36 |
| Max Depth | 24 |
| Min Samples Split | 12 |
| Min Samples Leaf | 12 |

Table 3.2: Neural Netork Setup/Hyperparameters

| NN Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 12.8 Million |
| Number of Layers | 10 |
| Nodes per Layer | 512 |
| Hidden Layer Activation | ReLU |
| Output Layer Activation | tanh (sigmoid for precip) |
| Loss Function | MSE |
| Batch Size | 256 |
| Epochs | 10 |
| Optimizer | Adam (learningRate= 0.0001) |

Initial efforts to couple ML models into CESM revealed significant limitations with both our attempts at direct integration of RFs and with the use of a deep learning-specific Fortran library, Neural-Fortran [Curcic, 2019]. Directly embedding RFs via NetCDF export led to persistent issues with memory management, grid incompatibility, and array indexing mismatches between Python and Fortran. Alternatively, the Neural-Fortran library, while promising for NNs, lacked support for our RF models and posed substantial challenges during integration with CESM due to complex linking and compilation requirements. These technical and logistical barriers ultimately led us to seek more flexible and maintainable coupling strategies.

#### 3.2.2.1 Possible Alternative Approaches

Several other tools have emerged to facilitate this integration, including the call-py-fort library, the Fortran-Keras-Bridge (FKB), and the Open Neural Network Exchange (ONNX). These libraries attempt to enable seamless interoperability between Fortran and Python, allowing researchers to leverage ML techniques into legacy models. The call-py-fort library provides an interface for calling Python functions directly from Fortran [Brenowitz]. This approach is particularly useful for integrating ML-based parameterization schemes, as it allows Fortran-based climate models to interact with Python-trained models in real-time. By leveraging interprocess communication (IPC) techniques and shared memory, call-py-fort appears to minimize data transfer overhead. This suggests efficient execution even when invoking complex deep learning models.

Similarly, the FKB enables Fortran programs to interface with NNs trained using Keras and TensorFlow [Ott et al., 2020]. FKB allows Fortran simulations to call pre-trained NNs during runtime, facilitating hybrid modeling approaches where ML corrects biases or en-

hances subgrid-scale processes without fully replacing physics-based parameterizations. FKB is limited in its scope as it is specifically implemented to work within the Keras framework and is also no longer maintained as a supported product.

ONNX is an open-source format designed to facilitate interoperability between ML frameworks, allowing models trained in PyTorch, TensorFlow, Keras, and other ML libraries to be efficiently deployed across various environments, including those integrated with Fortran-based numerical models [Bai et al., 2019]. ONNX enables ML models to be converted into a standardized, optimized format that can be executed with high-performance runtimes such as ONNX Runtime (ORT), which provides accelerated inference on CPUs, GPUs, and specialized hardware. This capability is particularly useful for integrating ML-driven parameterizations or post-processing corrections into climate and weather models, as it allows pre-trained models to be embedded within computationally intensive Fortran-based simulations without requiring direct execution in Python. By leveraging ONNX, researchers can achieve low-latency inference and cross-platform compatibility, making it a valuable tool for hybrid AI-physics modeling in Earth system sciences and an intriguing resource moving forward in this field.

These tools all represent a growing trend in the scientific computing community toward hybrid modeling, where ML-based corrections are integrated into traditional numerical models. As the use of AI in weather and climate modeling expands, libraries like call-py-fort, FKB, and ONNX provide critical foundations of much needed infrastructure for bridging the gap between modern ML frameworks and well-established Fortran-based climate models. Other similar products that we are aware of include the SmartSim and Inferno libraries [Zhang et al., 2025, Partee et al., 2022].

#### 3.2.2.2 Forpy and CESM

Our chosen method to streamline the integration of our models into the CAM framework led us to the ForPy library. ForPy is a Fortran-based library designed to facilitate seamless interaction between Python and Fortran, providing an efficient and straightforward interface for calling Python routines from within Fortran programs [Rabel, 2025]. This approach offers a 'best of both worlds' solution for scientific computing and modeling, allowing us to leverage Python's flexibility and ease of use alongside Fortran's computational efficiency. This library and approach can also accommodate both of our ML techniques using their standard Python functionalities.

ForPy simplifies the process of wrapping Python functionality and integrating Python libraries within Fortran subroutines, thus enabling a smoother integration between the two languages. One of its key features is the automatic conversion of data types, such as arrays,

integers, and floating-point numbers, ensuring smooth data exchange between Python and Fortran without the need for manual conversions. This automatic handling of data types significantly reduces the potential for errors and enhances the robustness of the integration process.

By using ForPy, we can maintain Python's user-friendly environment for high-level logic, data manipulation, and model orchestration, while still taking advantage of Fortran's performance benefits for numerically intensive tasks, such as matrix operations and large-scale scientific simulations. Additionally, unlike our previous challenges with the Neural-Fortran library, ForPy's integration was facilitated by ongoing work at NCAR. Specifically, the ForPy library had already been successfully linked and tested within a unique development branch of CESM (Will Chapman, personal communication). This pre-existing integration allowed us to proceed with the coupling tasks in this project more effectively.

## 3.3 Results & Discussion

A note on terminology: This study employs two distinct approaches for coupling machine-learned parameterization schemes with the CAM6 dynamical core. The first approach, referred to as 'online,' involves the standard CAM6 simulation, which operates with its internally calculated parameterized forcing while simultaneously invoking the machine-learned schemes via the ForPy library. This configuration ensures the concurrent execution of both models, allowing for the export of ML-predicted and CAM-calculated results at each time step while maintaining consistency in the input features for statistical analysis and comparison between the original subroutine and the machine learned routine. This online coupling technique is still forced by the CAM physical parameterization. The second approach, termed 'coupled,' integrates the ML-based physics tendencies directly into the CAM6 dynamical core, such that the dynamics are now forced by the output from the ML model. By employing these distinct methodologies, we enable a comprehensive analysis of the coupling process from multiple perspectives.

### 3.3.1 Online Results

#### 3.3.1.1 Zonal-Mean Time-Mean Analysis

In the set of panels in figure 3.1, mean temperature tendencies are visualized across a pressure-latitude cross section. The CAM output (a, d) shows a distinct vertical structure with a concentrated core of warming around the equator extending through the troposphere, representing the model's intended active convective heating tendency. The machine-learned

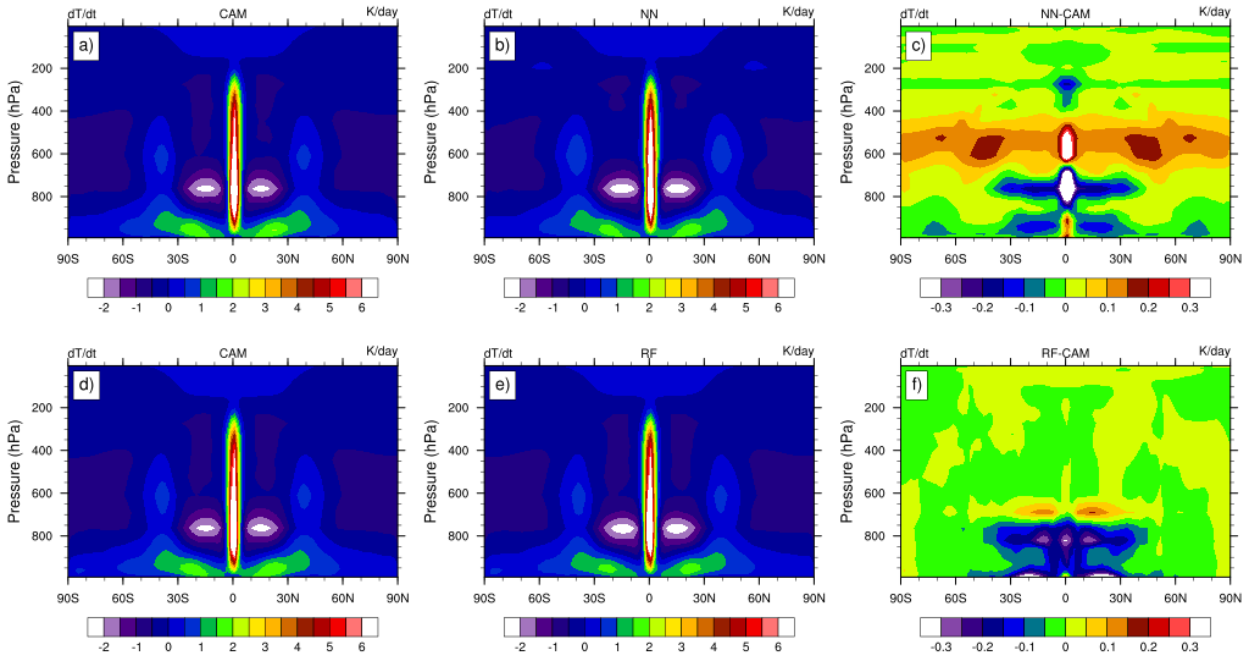



Figure 3.1: Zonal-mean time-mean of temperature tendencies in $K$/day in the online runs. The CAM6 'truth' on the left and the Fortran-calculated ML predictions in the middle; along with their difference on the right. The top row corresponds to the NN algorithm and the bottom is the RF.

emulators (b, e) capture this central heating structure well, though subtle differences are present in the surrounding regions. Panels c and f show mean anomalies between the NN and RF emulated tendencies and the CAM output, respectively. The NN differences showcase vertically alternating cooling and heating biases in patches, primarily observed in the equatorial region. Between the 500-600 hPa level, we see a broad region of overestimated heating, which turns into a cooling region around 800 hPa and 250 hPa. The strongest signals in these areas are at the equator, but error also accumulates into the subtropics and mid-latitudes, as is the case of the overestimated heating around that 600 hPa region. Regardless, these patterns imply that while the NN captures the primary convective heating, it may slightly misrepresent the vertical extent and strength of the warming.

In comparison, the RF emulator shows a similar overall pattern in its mean field, but with a more localized region of disagreement in the lower troposphere around the subtropics. The RF generally overestimates cooling in this region, while maintaining minimal discrepancies in the atmosphere above 600 hPa. This may indicate that the RF has a smoother representation of the tendency as is less prone to overestimation and underestimation.

We note that the errors in our RFs are substantially larger than in the offline experiments of Chapter 2 (Figures 2.3 and 2.4). This arises from two main differences, the first being

that we emulated two tendencies with a single RF rather than training separate models. While the second is due to reducing the 'max_depth' parameter during training. The latter strongly influences both predictive skill and memory footprint of RFs for these tasks, and tuning it was critical to balancing accuracy with computational feasibility.

These emulators effectively replicate the main heating structure but exhibit subtle biases, particularly in capturing finer details of the temperature tendency. The difference plots highlight that while these ML models grasp the dominant physical mechanisms, further refinement could improve the representation of smaller-scale processes and vertical structures. These discrepancies underline the complexity of accurately emulating the dynamic forcings within climate models and suggest that additional training or architectural adjustments may be needed to enhance performance.

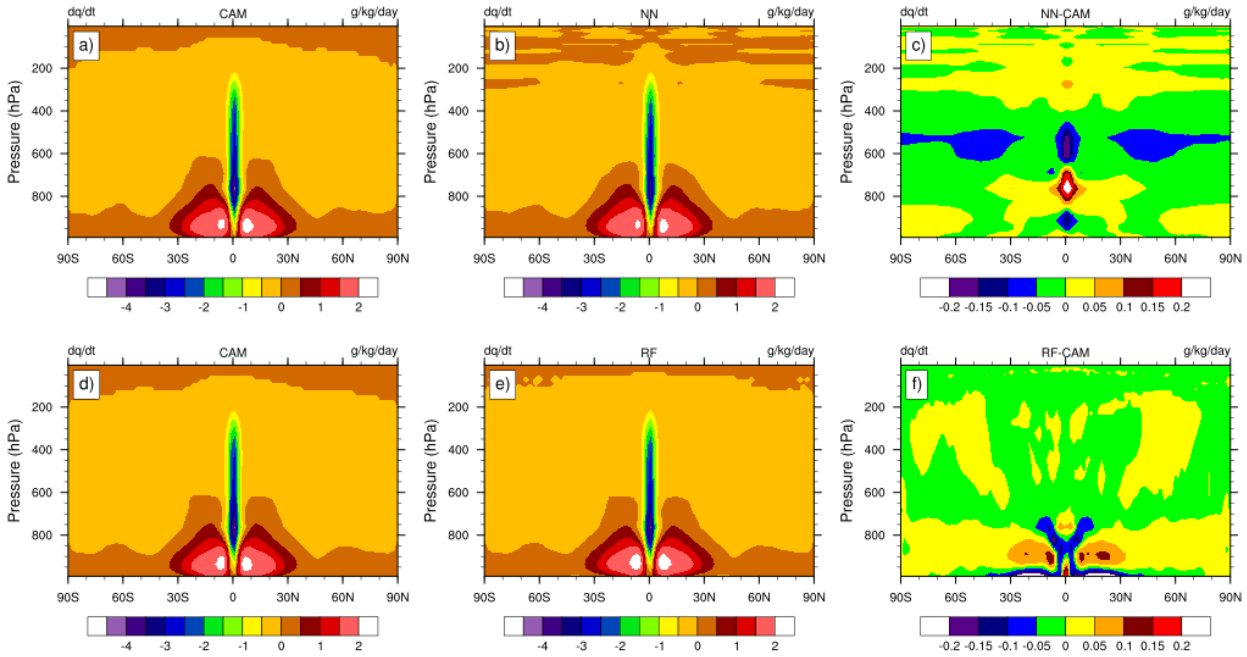



Figure 3.2: Zonal-mean time-mean of moisture tendencies in the online runs. The CAM6 'truth' on the left and the Fortran-calculated ML predictions in the middle; along with their difference on the right. The top row corresponds to our NN and the bottom is the RF.

Similarly, Figure 3.2 shows the online moisture tendency results in the same representation as Figure 3.1. Both models capture the active region of drying centered around the equator, corresponding to convective activity in the tropics, as well as the positive moisture tendencies in the subtropics below about 800 hPa. Panel (c) highlights the NN anomaly, revealing subtle discrepancies. Notably, the NN slightly overestimates moisture removal in the upper mid-latitudes (blue), along with a similar alternating over-and underestimation at the equator (red and blue).

The RF emulator also reproduces the overall structure observed in CAM, including the equatorial peak and the drying in the subtropics. However, Panel (f) again shows minimal overall deviations, most of which that do appear are centralized around the tropics and close to the surface.

Overall, both ML-based emulators capture the large-scale structure of moisture tendencies well, with the RFs showcasing less spatial deviations from the CAM forcing.

#### 3.3.1.2 $R^2$ Analysis

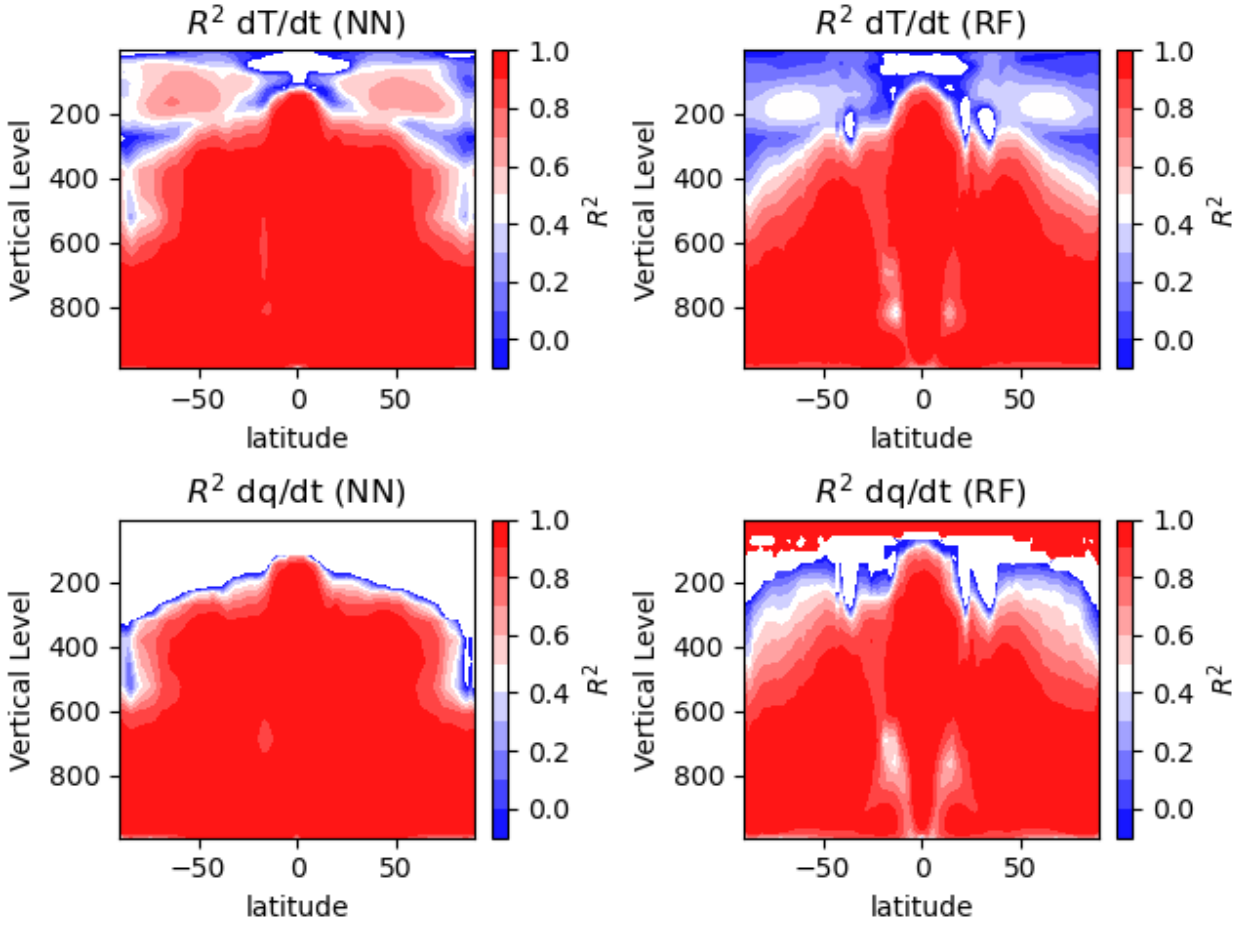



Figure 3.3: Spatial representation of $R^2$ score in a pressure-latitude cross section for the online temperature (top) and moisture (bottom) tendencies. The NN results are on the left while the RFs are on the right.

Figure 3.3 presents $R^2$ scores for temperature and moisture tendencies across latitude and vertical levels, comparing the performance of NN and RF emulators against the CAM run in our online mode. Refering back to Chapter 2 (Section 2.3.3), the $R^2$ score indicates

how well each emulator reproduces the CAM outputs, with values close to 1 (red) reflecting higher correlation (i.e. skill) to the original model and values closer to 0 or negative (blue) showing what we would consider to be poor emulator skill.

In the top row, the $R^2$ scores for temperature tendencies are shown. The NN (top-left) exhibits consistently high $R^2$ values throughout most of the vertical profile and latitudes, suggesting it captures temperature tendencies well, with only minor degradation near the upper atmosphere and polar regions. In contrast, the RF (top-right) shows more pronounced areas of low $R^2$ in the upper atmosphere and even some small areas in the lower troposphere, indicating it struggles more in these regions.

The bottom row presents $R^2$ scores for moisture tendencies and showcases similar findings. The NN (bottom-left) again shows strong performance, maintaining high values across much of the profile, though some degradation appears near the upper troposphere and at high latitudes. The RF (bottom-right), however, displays more substantial discrepancies, especially in the upper atmosphere and near the poles, along with larger areas of lower $R^2$ scores in the subtropics of the lower model levels, indicating challenges in accurately capturing the moisture-related processes.

With respect to $R^2$ analysis, the NN consistently outperforms the RF across both temperature and moisture tendencies with respect to online $R^2$ scores, demonstrating greater skill in replicating CAM outputs. Although, it is important to keep in mind that the anomaly fields for the RFs were consistently less pronounced throughout the atmosphere in Figures 3.1 and 3.2.

### 3.3.2 Coupled RFs

For the 'coupled' results, the ML emulator is now forcing the model, rather than the parameterization scheme. We run this simulation for six months, in order to evaluate our model performance. The temperature tendency in Figure 3.4 panel (a) exhibits the characteristic structure of our expected heating tendencies, with a strong localized maximum near the equator extending vertically through the troposphere. The peak heating occurs around 700 hPa in the mid-troposphere, consistent with deep convection in the tropics. The symmetric structure about the equator, along with weaker heating at higher latitudes, aligns well with the expected distribution of tropical convection and its influence on large-scale circulation patterns. Notably, the maximum equatorial heating in the RF-coupled system is significantly lower, around 5 K/day, compared to the CAM fields shown in Figure 3.1. This suggests that the RFs produce less intense heating in the coupled simulation, even in the most dynamically active regions.

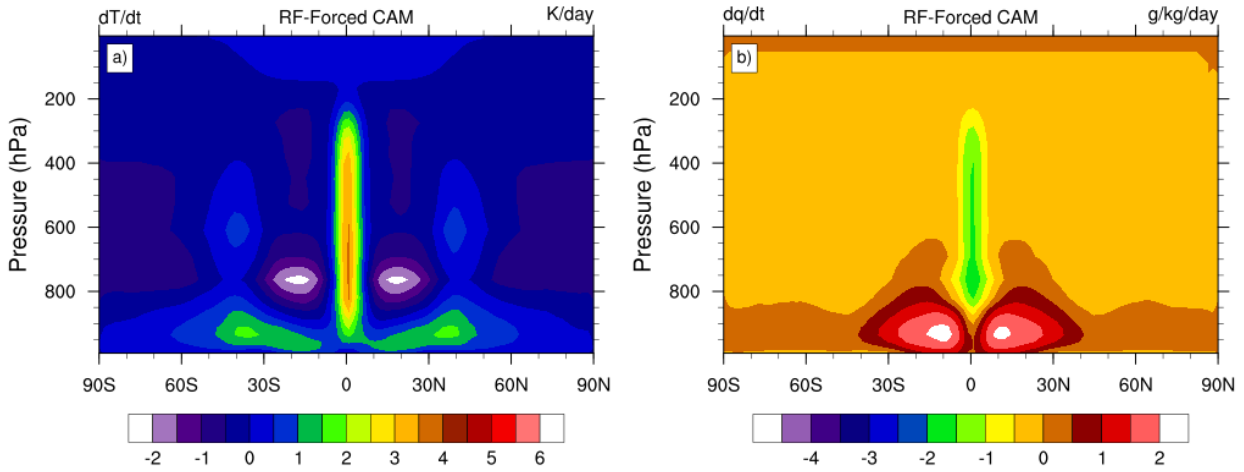



Figure 3.4: Zonal-mean time-mean of temperature (left) and moisture (right) tendencies for the coupled RFs.

The moisture tendency in Figure 3.4 panel (b) similarly reflects the expected moistening and drying processes associated with convection. While the overall structure aligns with the patterns observed in Figure 3.2, the coupled system shows a much less pronounced equatorial minimum in the mid-troposphere, indicating a reduction in the strength of drying at those levels.

Overall, this coupled run demonstrates the emulator's ability to maintain realistic convective patterns when integrated into CAM, with the RF-emulated tendencies successfully driving the model's parameterized physics. Additionally, the RF's tendency to produce more moderate temperature and moisture forcings contributes to a stable coupled system, enabling the model to run indefinitely without numerical instability.

### 3.3.3 Coupled NNs

#### 3.3.3.1 Min-Max Limiter

Initial tests with the forpy-coupled NN runs showed promising performance during the first few days of simulation. Since NNs have been shown to have trouble with out-of-sample prediction, these tests enforced a simple min-max limiter on the predictions. This was done by filtering any predicted value above the maximum at a given pressure level within the training data to be assigned to that maximum. Similarly, we did this for any predicted tendency below the minimum values seen in the training data as well. However, the coupled NN system developed non-physical tendencies that propagated with the general circulation of the model. These instabilities were consistently seeded in the tropics, where precipitation

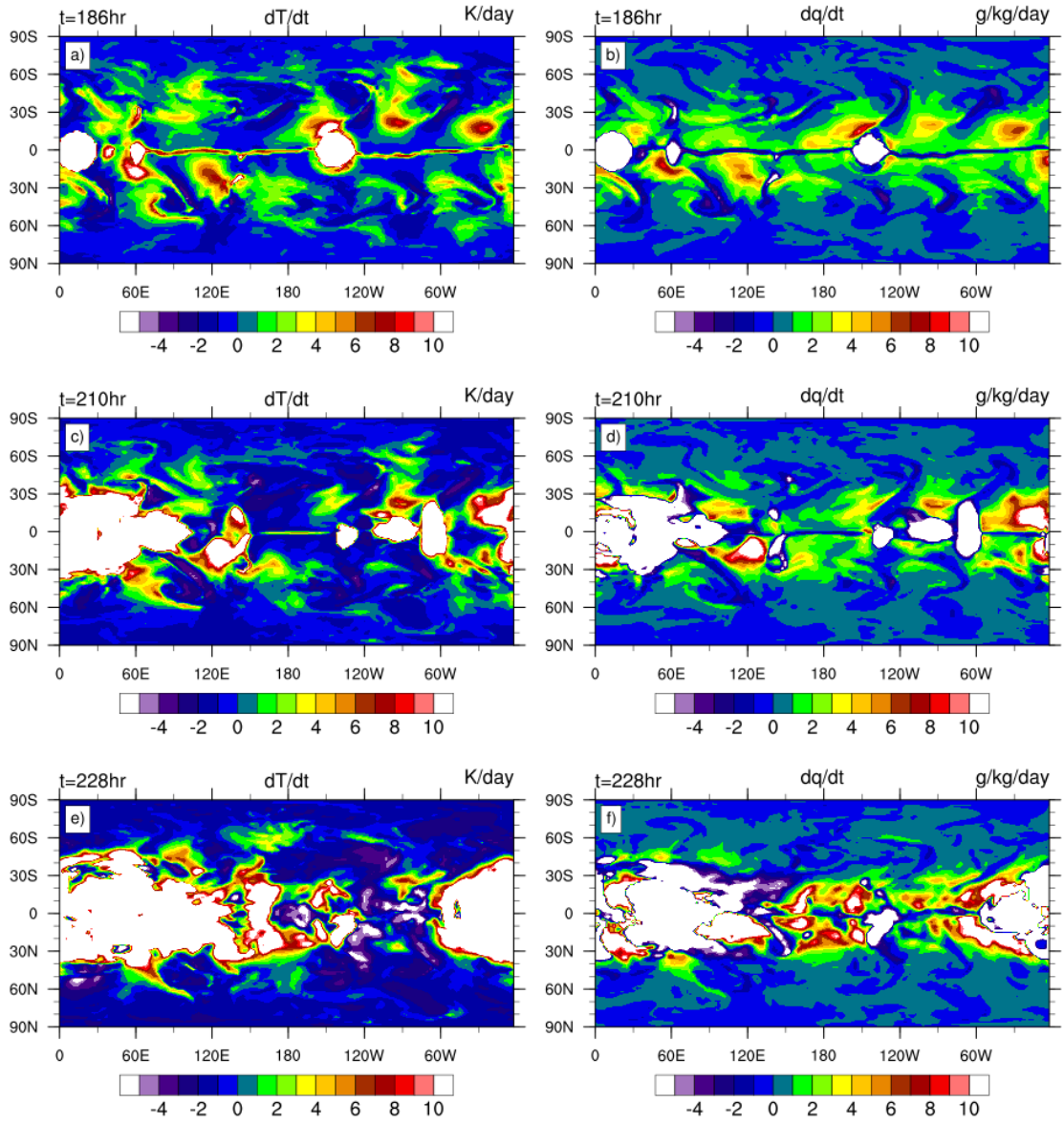



Figure 3.5: Development of an equatorial instability of the min-max limited NN-forced temperature (left) and moisture (right) tendencies by day 8 of simulation run, shown near the 850 hPa horizontal cross-section.

and latent heat processes are most intense. As these values grew outside of the scope of the training data, they fed back into the model, influencing subsequent time steps and amplifying the instability. Figure 3.5 illustrates this escalation, where clusters of extreme values emerged in dynamically active regions.

Ultimately, these instabilities compromised the model's vertical remapping process, a core

component for advancing time steps in CAM, and led to simulation failure after approximately 10 model days. These results underscore a well-documented challenge in ML-for-physics applications: the sensitivity of NNs to extrapolation [Qi and Majda, 2020]. In this context, even simple parameterization schemes can trigger instability when ML components interact with complex atmospheric feedbacks.

#### 3.3.3.2 Scaling the Min-Max Limiter

To address the crashing issues observed with the initial coupled NN runs, we implemented a scaling factor to our limiting scheme designed to constrain the model's predictions by a percentage of the max-min range. The intention was to aggressively prevent extreme values from forcing the model into a non-physical regime, triggering the numerical instability and led to model crashes. The color scales in Figure 3.6 highlight the growth and persistence of the instability, with the most extreme values (in white) corresponding to regions outside the min-max range of the training data.

The scaled limiting scheme extended the model runtime to approximately 18 days, nearly doubling the stability duration of the initial NN runs. This improvement suggests partial success in suppressing runaway instabilities, particularly in the early stages of simulation. However, as illustrated in Figure 3.6, localized non-physical structures, especially in regions of strong dynamical gradients and convective activity, still emerged. These anomalies, while reduced in intensity and frequency compared to the unscaled case, highlight the persistent challenge of numerical error amplification within coupled NN systems.

#### 3.3.3.3 Latitudinally-Dependent Limiter

To create a more adaptive stabilization strategy, we implemented a latitudinally dependent limiting scheme, designed to better match the latitudinal variability of atmospheric processes across regions. This works by checking if the NN generates values outside the scaled (0.7x) training range at a given latitude–pressure point, they were constrained to the corresponding scaled minimum or maximum observed in the training data. While this approach was conceptually appealing, it resulted, qualitatively, in slightly more pronounced nonphysical structures when compared to the non-latitudinally dependent method of Section 3.3.3.2. Figure 3.7 shows that highly localized hot spots (red and white regions) appear along the equatorial region and near the mid-latitude storm tracks in the temperature tendency field, suggesting regions of rapid destabilization. Similarly, the humidity tendency field exhibits strong anomalous drying and moistening bands, particularly in the tropics, where moisture transport and convection processes are dominant.

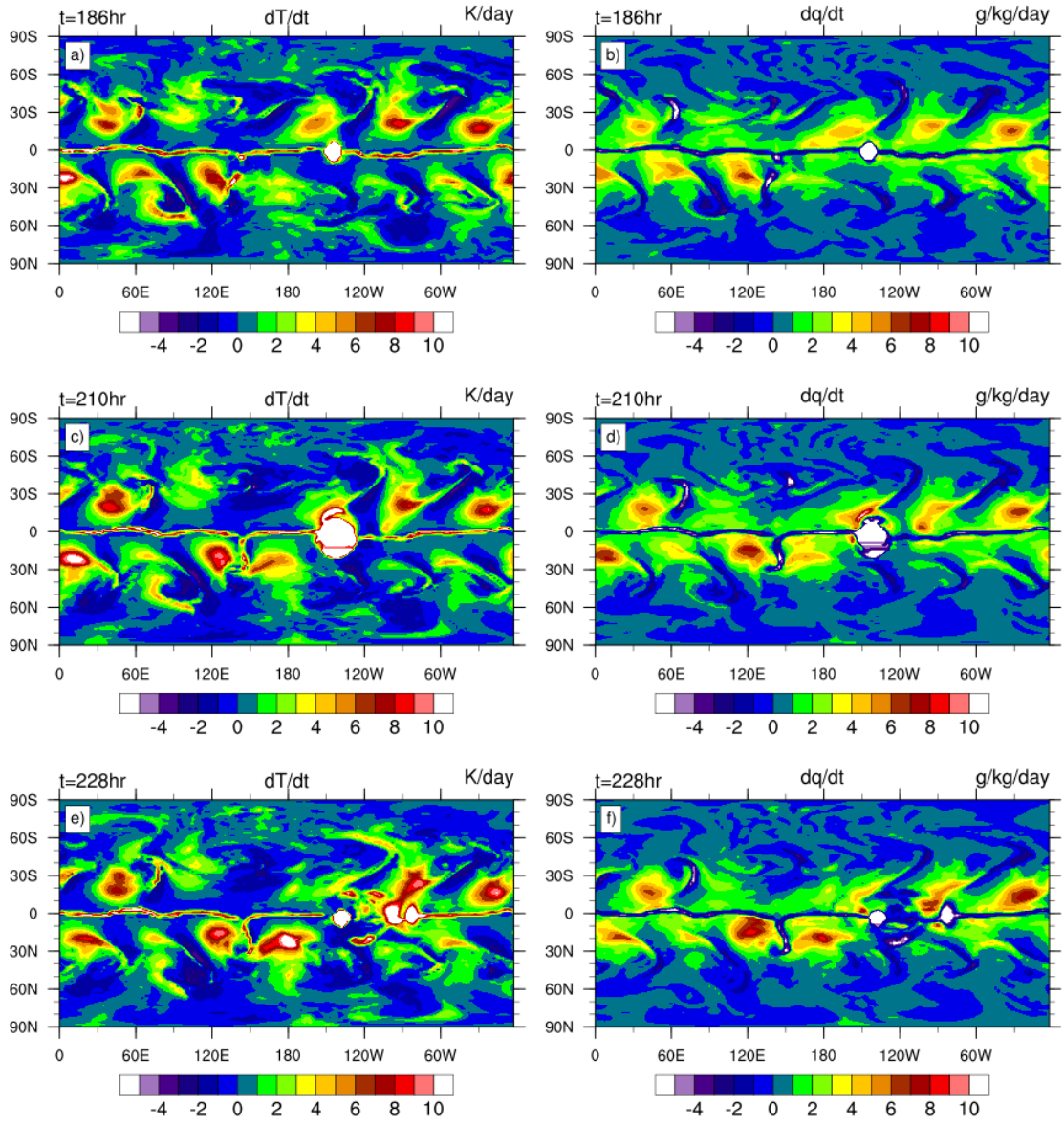



Figure 3.6: Development of an equatorial instability of the scaled (0.7 × min-max range) NN-forced temperature (left) and moisture (right) tendencies by day 8 of simulation run, shown near the 850 hPa horizontal cross-section.

These outcomes suggest that spatially varying constraints must be implemented carefully to avoid introducing artificial gradients that interact nonlinearly with model dynamics. Still, this experiment provides valuable insight into the design of regionally adaptive stabilizers. Future improvements might include smoothing strategies for spatial thresholds, or incorporating physically informed priors that respect large-scale balance constraints.

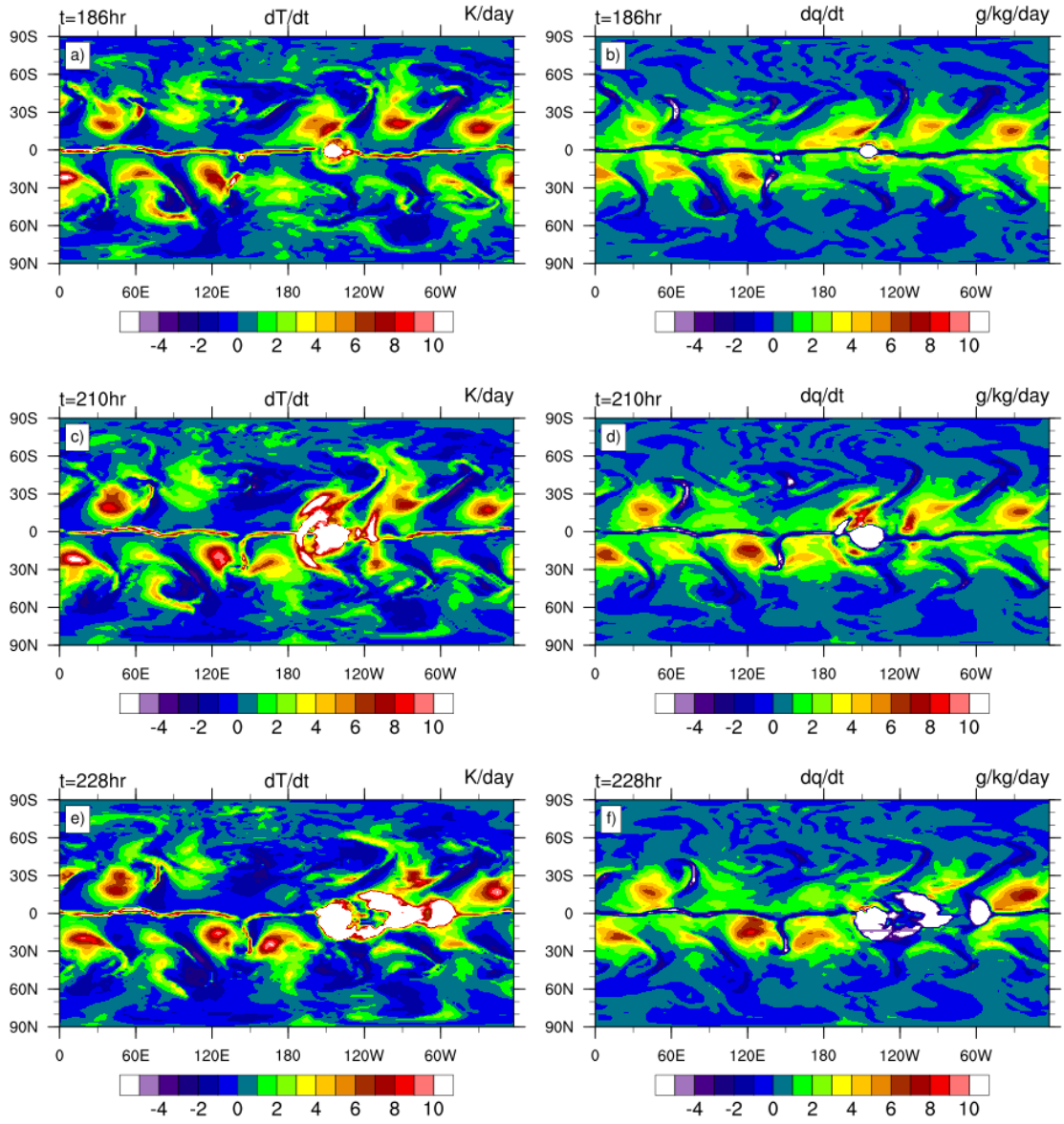



Figure 3.7: Development of an equatorial instability of the latitudinally-dependent scaled (0.7 × min-max range) NN-forced temperature (left) and moisture (right) tendencies by day 8 of simulation run, shown near the 850 hPa horizontal cross-section.

## 3.4 Concluding Thoughts & Applications to Future Work

Our findings extend prior investigations into the role of ML in physical parameterizations, reinforcing both its potential and the practical challenges associated with online deployment

within a flagship climate modeling framework. While ML methods, particularly simple architectures like feed-forward NNs and RFs, can emulate key features of subgrid tendencies offline, their integration into full dynamical systems introduces new complexities that warrant deeper exploration.

A key challenge lies in the engineering interface between Python-based ML models and Fortran-based climate model infrastructure. Our work highlights the non-trivial effort required to build robust, low-latency coupling mechanisms and the sensitivity of model stability to seemingly minor communication inefficiencies. These findings emphasize the importance of not only model architecture but also implementation design when deploying ML in climate models.

In terms of stability, RFs offered a more robust solution when coupling, though their memory footprint, especially for deeper, highly skilled, trees, proved problematic for operational use. Conversely, NNs demonstrated better scalability but suffered from numerical instability due to their sensitivity to out-of-distribution inputs. This is of particular concern when said inputs are being calculated based on the prior time step's NN prediction. Despite efforts to apply limiting schemes, the NNs in this study exhibited runaway tendencies, spawning in convective regions, and leading to model breakdowns within a few simulation weeks.

One potential avenue for improving our emulators is to modify the spatial sampling strategy during training to better capture and represent tropical extremes. The tropics are the clear source of instability when our NNs are coupled to CAM, and they remain underrepresented in our training dataset relative to other latitudinal bands. At present, we do not apply any subsampling or weighting to increase tropical representation; however, similar studies have adopted such strategies with some success [O'Gorman and Dwyer, 2018]. Incorporating targeted sampling in this region may enhance emulator stability and performance in coupled settings.

Despite these limitations, our results provide valuable insight into the challenges of coupling ML emulators to climate models and point toward concrete paths for improvement. A guiding principle throughout this work was to explore whether simple ML methods could effectively emulate what are considered 'simple' physical parameterizations. By maintaining this approach, we were able to investigate foundational questions about coupling and emulator behavior, an approach that aligns with common practices in the model development community when introducing new numerical tools. That said, we acknowledge that more advanced techniques, including physics-informed NNs, hybrid ML-physics frameworks, and dynamically constrained emulators, are likely to offer a more robust foundation for stable and physically consistent coupling [Kashinath et al., 2021]. In parallel, methods from

numerical weather prediction, including adaptive regularization, error correction loops, or ensemble-based stabilization, could be adapted to help mitigate numerical drift in online learning systems [Zhao et al., 2019].

Our results also underscore the value of cross-disciplinary tools and frameworks. For instance, interfacing tools such as FTORCH may offer more efficient communication strategies, especially if the ML model is natively developed within PyTorch [Atkinson et al., 2025]. Transitioning to a unified software stack could help alleviate some of the memory management bottlenecks observed with RFs, while facilitating the use of GPU-accelerated inference. Work along these lines has already begun at NCAR and with regards to bridging ML into the CESM and CAM frameworks via the CREDIT initiative [Chapman et al., 2025, Schreck et al., 2025].

Overall, this work affirms that coupling ML-based parameterizations with comprehensive Earth system models is a deeply interdisciplinary challenge; one that blends algorithm design, physical modeling, and software engineering. Even in simplified setups, success will likely require a combination of physically grounded architectures, scalable interfacing strategies, and a deep understanding of the model's dynamical sensitivity. As interest in ML-accelerated simulation continues to grow, we expect that advances in architecture, training methodology, and physical interpretability will increasingly enable stable and efficient online emulation.

# CHAPTER 4

# Challenges and Opportunities of Evolving Dynamical Tests for AI-Driven Weather Models

## 4.1 Introduction

### 4.1.1 The Importance of Dynamical Tests for Weather and Climate Models

As discussed in Section 1.4.2.3, our research has shifted focus from the emulation of parameterization schemes to the exploration of the dynamic properties of AI-driven weather forecasting models. This shift began with the idea of incorporating a machine learning (ML) working group into the 2025 Dynamical Core Intercomparison Project (DCMIP), held in June, 2025, in Boulder, CO, USA. A critical aspect of weather and climate model development is the dynamical core, which serves as the foundation for the model's ability to simulate atmospheric motion. The dynamical core is responsible for solving the fundamental equations of motion that govern the behavior of the atmosphere. As such, the choice of dynamical core plays a pivotal role in determining the accuracy and efficiency of the entire model and understanding the dynamical properties of any climate or weather model is essential for the field to progress.

Given the multitude of choices that dictate the variety of dynamical cores that are used in climate and weather models, intercomparison is crucial to understand their differences across standardized physical test cases. This provides insight into developmental choice, advantages and disadvantages, as well as maintaining physical consistency across modeling frameworks. This focus on the performance and behavior of various dynamical cores in a controlled setting forms the basis of DCMIP.

### 4.1.2 AI-Driven Weather Models

Currently, AI-driven models are rapidly transforming the field of weather forecasting. Leveraging advanced ML techniques, particularly generative and transformer-based deep learning architectures, these models are pushing the boundaries of what is possible in weather prediction, achieving impressive speed and accuracy. These AI-driven approaches have the potential to bypass traditional, complex physics-based dynamical models, offering a novel and more efficient means of forecasting.

A few key models have led the way in this transition towards fully data-driven global weather forecasting, including NVIDIA's FourCastNet, Google's GraphCast, and Huawei Cloud's PanguWeather [Kurth et al., 2023, Lam et al., 2023, Bi et al., 2023]. These models share a common foundation in their reliance on large-scale reanalysis datasets, particularly the ERA5 dataset, which has become a key resource for training and validating these AI models. ERA5 is a global, high-resolution reanalysis dataset that combines observations from various sources with model outputs to provide an interpolated, comprehensive representation of the atmosphere and climate over the past 80 years [Hersbach et al., 2020].

Reanalysis data, such as ERA5, are generated through a process that merges numerical weather prediction models with historical observational data, creating a consistent record of past atmospheric conditions. This approach enables the generation of large-scale, high-resolution weather data that can be used to train ML models, thus empowering AI systems to learn from both historical patterns and real-time data. The reliance on ERA5, with its broad temporal coverage and global scope, ensures that AI models can be trained on a rich and diverse dataset, capable of capturing complex climate and weather phenomena across time and space.

It is important to recognize that ERA5 and other reanalysis datasets are not intended to represent an exact record of true atmospheric conditions globally. Despite their impressive spatial and temporal resolution, these datasets remain approximations of the climate system, constrained by model assumptions, data assimilation techniques, and observational limitations. Consequently, any data-driven forecasting model trained to emulate reanalysis output can, at best, achieve predictive skill only up to the extent that the reanalysis itself accurately represents the underlying atmospheric state. Recognizing this limitation is essential to accurately evaluate such models.

In recent years, the landscape of AI-driven weather and climate forecasting models has rapidly expanded, with contributions from a wide range of research groups beyond the three pioneering efforts mentioned earlier. For instance, in 2024, Microsoft introduced its Aurora model, while the European Centre for Medium-Range Weather Forecasts (ECMWF) unveiled their AI Forecasting System (AIFS) model [Bodnar et al., 2025, Lang et al., 2024]. Both

of these additions represent significant advancements in the growing body of data-driven forecasting models, further cementing AI's role in modern meteorology.

In the same year, GenCast was launched by Google's DeepMind, building on the success of their earlier model, GraphCast [Price et al., 2025]. Unlike GraphCast, which employed graph neural networks, GenCast utilizes generative AI techniques as its core framework, seemingly offering a more powerful approach to modeling weather systems. Similarly, NVIDIA released the Spherical Neural Operator (SFNO) model, an extension of their FourCastNet model [Bonev et al., 2023]. SFNO incorporates spherical Fourier neural operators to enhance the model's performance, further improving its capability to predict atmospheric phenomena with high accuracy.

For this project, we focus primarily on the GraphCast model. At the core of GraphCast is its use of graph neural networks. This type of deep learning architecture appears particularly well-suited for handling spatially and temporally correlated data, such as atmospheric variables [Lam et al., 2023]. In GraphCast, the "graph" represents a network of interconnected nodes, each corresponding to a different part of the Earth's atmosphere, like temperature, pressure, and wind. This network structure allows the model to capture the intricate relationships between different atmospheric regions and variables, appearing to learn various dependencies that underlie atmospheric dynamics. The model is capable of quickly generating high-resolution weather forecasts at global scales, extending into multiple days into the future.

These models, along with many others being developed worldwide, represent the forefront of AI-driven advancements in weather forecasting. Each model brings its own unique innovations in terms of architecture and approach, reflecting the growing diversity and sophistication of ML methods in atmospheric science. For readers interested in detailed technical analyses of the underlying architectures and methodologies of these models, we direct them to the relevant literature cited above. Given the complexity of deep learning architectures and their nuanced differences, we acknowledge that these technical aspects are beyond the primary scope of this project and are not the focus of our current research.

### 4.1.3 Using AI-Driven Models as a Proxy for Sophisticated Dynamical Cores

This work builds upon a growing interest in using AI-driven weather forecasting models as proxies for traditional numerical weather prediction (NWP) systems. Our goal is to extend a framework for developing dynamical test cases that assess the learned underlying dynamics of such models.

The work by Hakim and Masanam [2024] highlights the importance of evaluating whether AI-based models exhibit essential dynamical characteristics such as wave response, energy conservation, stability, and error growth. Their study demonstrated that the PanguWeather model was capable of reproducing key large-scale dynamical processes, including baroclinic instability and Rossby wave propagation. These findings motivate the development of systematic test cases to evaluate such behavior across a broader set of AI-driven models.

Inspired by this direction, we designed a suite of well-defined dynamical test cases and applied them to three state-of-the-art models: PanguWeather, SFNO, and GraphCast-Operational. These experiments were conducted in preparation for and during the DCMIP2025 intercomparison project. The results from our DCMIP studies are currently being prepared for publication in collaboration with other researchers; however, the methodological frameworks developed there have been extended in this project to an additional model not included in the original intercomparison.

### 4.1.4 Overview of Models and Their Variants

During DCMIP2025, we enlisted PanguWeather, GraphCast's operational variant, and SFNO. As previously mentioned, SFNO is NVIDIA's successor to FourCastNet and uses spherical Fourier neural operators to improve model stability and performance at longer lead times [Bonev et al., 2023]. GraphCast-Operational, referred to from here on out as GraphCast-OP, is one of several available versions of Google DeepMind's GraphCast model, optimized for real-time forecasting using a reduced vertical resolution and disregarding the TP06 variable for six-hourly accumulated precipitation as an input variable. In contrast to the default 37-level GraphCast model, referred to moving forward as GraphCast-37. All of the models discussed in this chapter run at a horizontal resolution of $0.25^\circ$ and outside of GraphCast-37, each model utilizes 13 vertical levels.

It is important to note for our framework, we standardize the models to use a constant 6-hour inference time step for all models. This was done both in this work and for the DCMIP2025 experiments. This distinction is important when interpreting model intercomparisons of dynamical responses, since the work done by Hakim and Masanam [2024] incorporates a dynamic timestepping method utilizing the various inference versions of PanguWeather available (see the first paragraph of Section 2 in their work for details on their inference framework).

In this project, we primarily investigate the performance of GraphCast-37 in relation to GraphCast-OP in order to probe the impact of the additional vertical resolution on the dynamical response we see. GraphCast-37 is important for our investigation in this thesis,

as there are currently no existing studies within the literature that examine the dynamical response of an AI-driven weather forecasting model with this vertical resolution. We also incorporate the PanguWeather-6hr model for direct comparison to the original methods shown in Hakim and Masanam [2024]. No investigations into SFNO were performed for this work, as that model is more central to our overview paper that is being worked on with the whole DCMIP-AI team.

## 4.2 Methods

### 4.2.1 Earth2MIP Framework and Tendency Reversion

To deploy our test cases, we utilized the Earth2MIP framework developed by NVIDIA [NVIDIA, 2025a]. Earth2MIP constitutes a substantial advancement in the unification of ML techniques with physical modeling, aiming to streamline the evaluation and deployment of ML-based weather models by providing a common framework that hosts a variety of state-of-the-art weather emulators, including PanguWeather, GraphCast, SFNO, and many others. The framework offers a standardized interface for interacting with the heterogeneous datasets commonly used in climate and weather modeling, thereby reducing the complexity associated with data preprocessing, model validation, and most importantly, inference. By promoting accessibility and reproducibility, Earth2MIP provides consistent evaluation metrics and preprocessing routines, enabling more rigorous comparisons across different ML approaches. In doing so, the framework attempted to bridge the gap between the ML and geosciences communities, lowering the barrier to operational adoption of ML techniques in both forecasting and climate simulation contexts.

The key to the test cases we incorporate and extend upon in this work is the technique known here as tendency reversion (TR). We refer readers to the source code repository, primarily developed by Joshua Elms [Elms, 2025], for implementation details, and to Hakim and Masanam [2024] for additional mathematical details of the TR method and initial condition formulations. A simple description of how TR works is given as

$$\mathbf{x}(t+1) = \mathbf{N}[\mathbf{x}(t)] - \mathbf{d}\bar{\mathbf{x}} + \mathbf{f} \tag{4.1}$$

where $\mathbf{x}$ is the model state vector, $\mathbf{N}$ is our AI-driven weather forecast operator which acts on the previous state vector, $t$ is the time stepping index of our models, and $\mathbf{d}\bar{\mathbf{x}}$ is what we refer to as the 'tendency' produced from one-step of an atmospheric mean state and is formulated by

$$\mathbf{d}\bar{\mathbf{x}} = \mathbf{N}[\bar{\mathbf{x}}] - \bar{\mathbf{x}} \tag{4.2}$$

Here, an overbar denotes the December–January–February (DJF) winter mean of a variable.

In this way, the procedure relies on the generation of seasonal mean-state initial conditions (e.g., boreal winter, DJF), which are used to obtain the model's tendency $\mathbf{d\bar{x}}$. We note that the DJF mean must be taken from the same time of day for each day within the averaged period, 20 to 40 years of ERA5 data in our case, in order to preserve the signal of the diurnal cycle that these models expect within their input fields. This means that we take the 0000 UTC time of each day during the DJF winter from ERA5 data. The tendency is defined as the difference between the predicted state after one timestep and the original seasonal mean state. This estimated tendency is then recursively subtracted from the evolving model state at each inference step, as shown in equation 4.2. In the absence of external perturbations to the initial conditions or input fields, this approach is expected to yield a quasi-steady-state atmospheric evolution. Conversely, the introduction of perturbations allows for isolation and analysis of the model's dynamic response, thereby enabling controlled experimentation within a physically constrained framework. To enable model runs to utilize our TR framework, we implemented minor modifications to Earth2MIP's internal handling of model inference.

### 4.2.2 GraphCast: Challenges and Benefits

The core of our analysis focuses on the application of TR to the GraphCast-37 model, which presents both scientific opportunities and technical challenges. Unlike GraphCast-OP, PanguWeather, or SFNO, the GraphCast-37 architecture operates on a 37-level vertical grid. Pressure levels of these models are constant and defined at irregular intervals from 1000 hPa at the surface up to 50 hPa for the 13-level models and 3 hPa for GraphCast-OP. This added vertical resolution enables a more detailed representation of the atmosphere and allows us to investigate features that may be muted or absent in lower-resolution counterparts.

A central question is whether the TR methodology can be robustly applied to the GraphCast models. Should TR perform well, we are also interested in whether the increased vertical fidelity allows detection of additional dynamical features not resolvable in GraphCast-OP.

A key challenge in implementing TR to the GraphCast models, compared to SFNO and PanguWeather, lies in the differences in inference framework. Whereas SFNO and PanguWeather accept a single input state (at a given time step) and produce a prediction for the subsequent state, GraphCast requires two consecutive prior states to generate its forecast. Although this modification may appear relatively straightforward, it necessitates additional changes within the Earth2MIP inference source code in order to incorporate the TR methodology appropriately. Along with the source code modifications, it also introduces choices we make that impact the effectiveness of TR, such as how to apply the initial perturbations.

Another important distinction of GraphCast-37, not shared by the other models, including GraphCast-OP, is the inclusion of the TP06 variable. This is the total accumulated precipitation over a 6-hour period. This variable is not a required input for SFNO, PanguWeather or GraphCast-OP, but it is required for the GraphCast-37 input. In our implementation, we set this variable to zero, especially when using seasonal mean states as initial inputs. However, the appropriateness of this choice remains an open question and may warrant further investigation.

### 4.2.3 Adopting the Test Cases

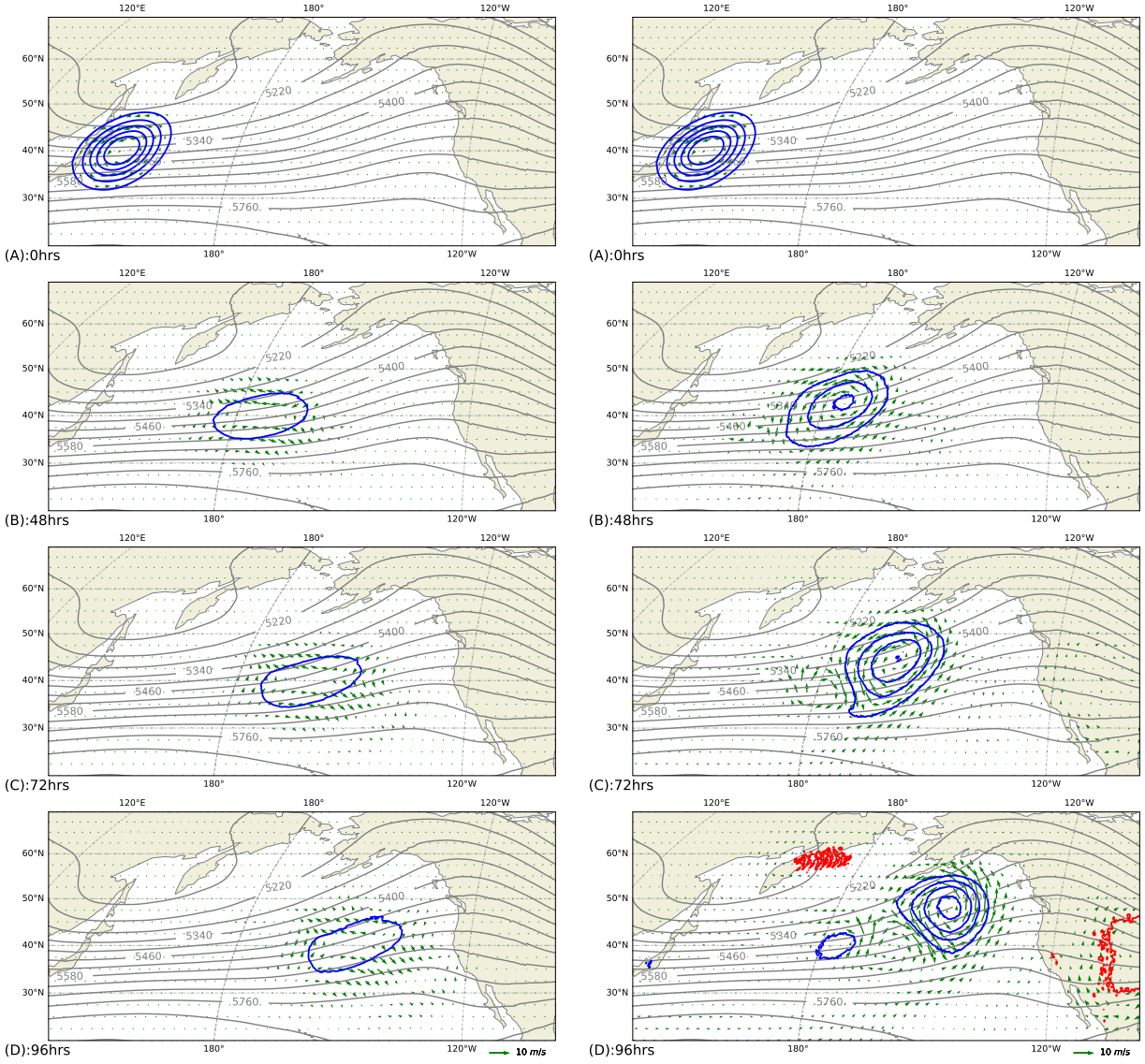
Both of the test cases that we explore with GraphCast-37 use a simple formulation of perturbation of a given atmospheric state. This perturbation is functionally described by a perturbation amplitude multiplied by the spatially varying gret

$$G(r, L) = \begin{cases} -(1/4)r^5 + (1/2)r^4 + (5/8)r^3 - (5/3)r^2 + 1 & 0 \leq d \leq L/2 \\ (1/12)r^5 - (1/2)r^4 + (5/8)r^3 + (5/3)r^2 - 5r + 4 - (2/3)r^{-1} & L/2 \leq d \leq L \\ 0 & L \leq d \end{cases} \tag{4.3}$$

Here, $r = 2d/L$ and $d$ is the distance on the sphere from a central reference point [Gaspari and Cohn, 1999].

The first is an extratropical cyclone (ETC) initial condition, in which we adopt the same framework as experiment 'b' from Hakim and Masanam [2024], using the same linear regressed perturbation field scaled by Equation 4.3 with $L = 2000$ km, centered at 40°N, 150°E (see supplemental information of Hakim and Masanam [2024] for details on the regression field). This test is employed to assess the ability of our GraphCast-37 implementation to support the TR technique, and to facilitate a direct comparison with both PanguWeather and GraphCast-OP regarding the evolution and propagation of the ETC within each simulation.

The second test case involves the application of a constant tropical heating anomaly centered at the equator near 120°E. The anomaly is applied using the same functional form as in Equation 4.3, with control runs using a constant heating rate amplitude of 0.1 K per six hours (the natural time step of our models).




(a) Pangu-Weather (6hr) ETC Simulation

(b) GraphCast-Operational ETC Simulation

Figure 4.1: Positive (red) and negative (blue) 500 hPa Geopotential height anomaly from ETC test case at 0, 48, 72, and 96 hours (A-D, respectively) of simulation time. Anomalous contours shown at 20 m spacing, with the 0 m contour being suppressed. Background 40-year DJF-mean geopotential height (grey) along with anomalous wind vectors (green arrows) are also shown.

## 4.3 Results

### 4.3.1 Intercomparison of Simple Extratropical Cyclone Perturbation

Our first test case examines the ETC–like response to a perturbed 500 hPa geopotential height (m) field in the northern tropics, applied across each model. Figure 4.1 illustrates the resulting daily geopotential height anomalies, overlaid as red and blue contours on the DJF mean state for the six-hour PanguWeather model (panel a) and the GraphCast-OP model (panel b) over the first four days of simulation. Anomalous wind vectors are included to highlight the cyclonic structure of the disturbance.

Both models exhibit an eastward-propagating trough across the Pacific, but notable differences emerge in the structure and amplitude of the response. The 6-hour PanguWeather response is considerably more muted than that reported in Hakim and Masanam [2024], likely due to differences in inference design, specifically, their use of dynamic time-stepping sequences as opposed to our fixed 6-hour inference steps. To ensure comparability with their results, we adopt the same 20 m anomaly contour interval used in their figures for this initial test case.

Despite this, GraphCast-OP results (panel b) display a more pronounced trough in response to the initial disturbance, more closely resembling the one documented in Hakim and Masanam [2024]. However, the surrounding ridges (positive anomalies) are weaker in amplitude and not displayed with the chosen +20 m contour. Also, additional noise emerges by day 4 of the simulation. Specifically, we observe a localized −20 m anomaly over Japan, a strong +20 m signal over the North Pacific, and another +20 m anomaly over the United States. These features do not correspond to the initial perturbation and are likely artifacts arising from error accumulation during the inference process under the TR framework.

These noisy responses suggest that GraphCast models, as implemented, may be more sensitive to TR than the other models tested. This was observed during simulation, as our codebase includes diagnostic checks to ensure the TR method is functioning correctly. These checks consistently revealed that errors associated with the TR technique were noticeably larger in magnitude for the GraphCast models compared to PanguWeather and the other models used during DCMIP2025. We will return to this point in more detail in Section 4.4.1. Nonetheless, it is important to note that such spurious signals persist throughout our experiments using both the GraphCast-OP and GraphCast-37 models, and must be considered when interpreting the results.

Figure 4.2 presents the 500 hPa geopotential height anomalies resulting from the ETC ini-

tial condition applied to GraphCast-37. In this configuration, the model produces a response that more closely resembles the PanguWeather results reported in Hakim and Masanam [2024], sustained consistently throughout the four-day simulation period. We observe a well-defined trough progressing across the North Pacific, flanked by ridges and exhibiting clear

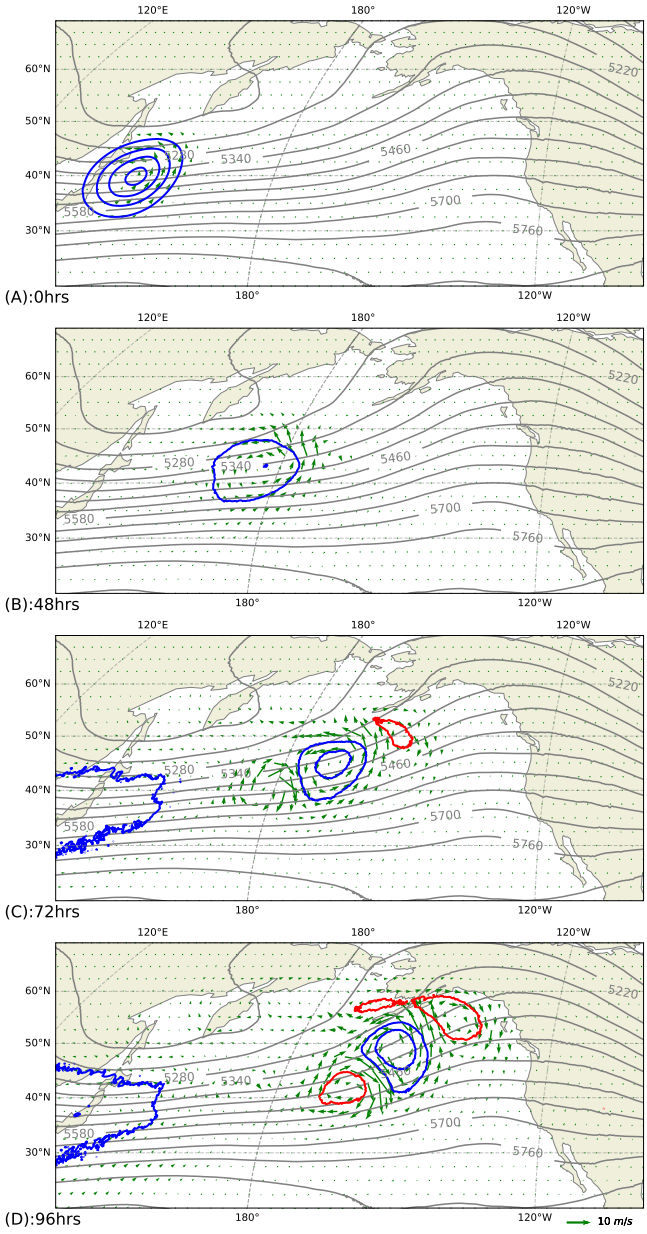



Figure 4.2: Positive (red) and negative (blue) 500 hPa Geopotential height anomaly from the ETC test case at 0, 48, 72, and 96 (A-D, respectively) hours for GraphCast-37. Anomalous contours shown at 20 m spacing, with the 0 m contour being suppressed. Background 20-year DJF-mean geopotential height (grey) along with anomalous wind vectors (green arrows) are also shown.

cyclonic motion. Compared to GraphCast-OP variant, the 37-level model does not reproduce the secondary trough, evident also in the Hakim and Masanam [2024] experiment. However, it exhibits a stronger primary cyclonic response and more pronounced ridge structures.

Additionally, we still detect signs of noisy error accumulation outside the immediate region of interest. A notable example is the early appearance of a $-20$ m anomaly over the western Pacific, which stands out as the only strong artifact to arise prior to day four in any of our experiments. By day four, a $+20$ m anomaly emerges just north of the cyclonic disturbance, similar in character to the spurious signal observed over the eastern edge of Russia in Figure 4.1, panel b. However, in the 37-level case, this artifact appears much closer to the region of interest and may interfere more directly with interpretation of the dynamical response.

### 4.3.2 Extratropical Cyclone: A Closer Look

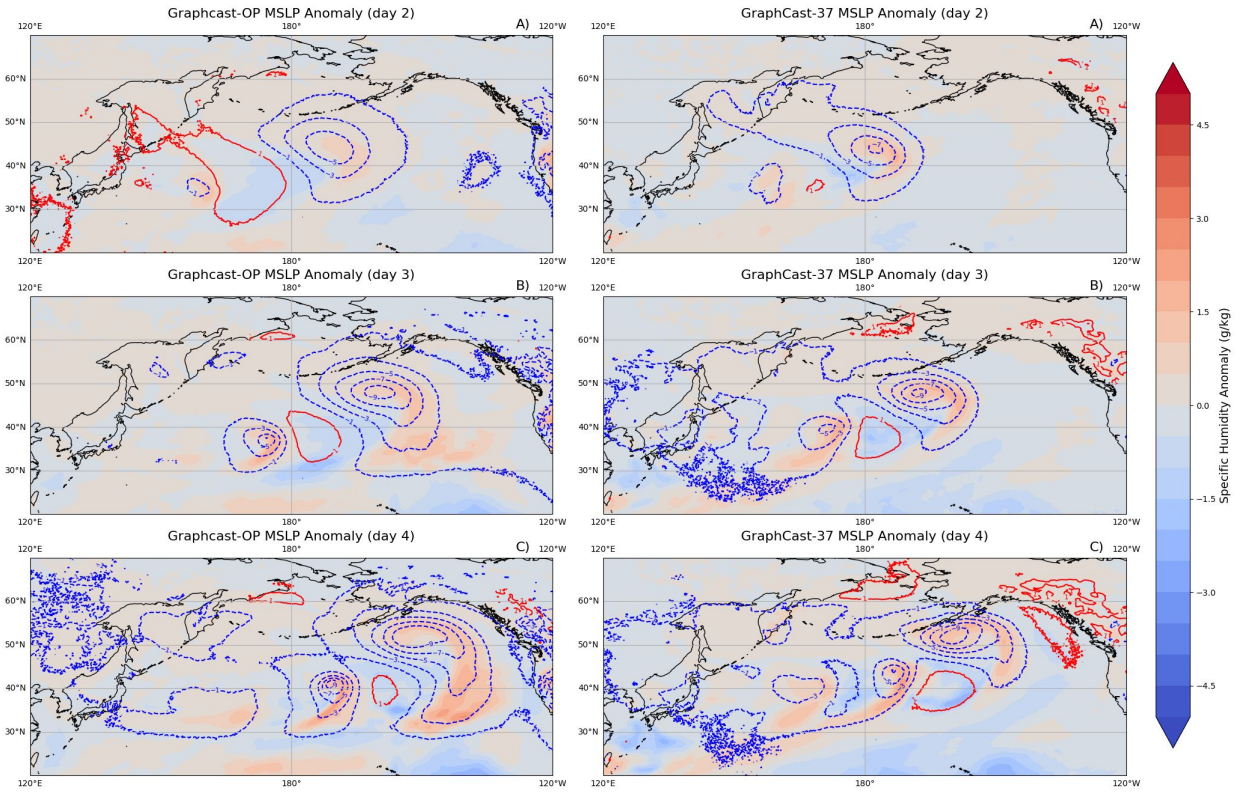



Figure 4.3: Mean sea level pressure (MSLP, red/blue contours, hPa) and 850 hPa specific humidity anomalies for GraphCast-OP (left) and GraphCast-37 (right) at forecast days 2, 3, and 4 from the ETC seed. Red (blue) contours denote positive (negative) MSLP anomalies at 2 hPa intervals, with the zero contour suppressed. Shading shows moisture anomalies.

We further examine the differences between GraphCast-OP and the higher-resolution

GraphCast-37 using the ETC perturbation, focusing on mean sea level pressure (MSLP) and 850 hPa specific humidity anomalies, shown in Figure 4.3. This analysis was motivated by the diagnostics presented in Figure 4 of Hakim and Masanam [2024], with the goal of assessing the realism, structural consistency, and localized impacts of ETC development in both configurations.

By forecast day two, both models develop a coherent low-pressure anomaly from the DJF mean MSLP over the northwestern Pacific, accompanied by expected moist anomalies along the fronts of the low-pressure system, features characteristic of baroclinic wave development. By day three, differences between the two configurations begin to emerge, shown in GraphCast-37 retaining a compact cyclonic core with tightly wrapped MSLP anomalies along the fronts, while GraphCast-OP shows earlier radial expansion of the low-pressure field and more diffuse specific humidity anomalies. One notable feature in GraphCast-OP is the broad positive specific humidity anomaly to the southeast of the MSLP low on day three, extending to 20°N between 160°W and 140°W. This drying is clearly linked to the pressure perturbation and the spatially extended frontal low, indicating more rapid dissipation of the ETC in GraphCast-OP compared to GraphCast-37.

By day four, these contrasts are more pronounced. The 37-level simulation continues to maintain a sharply defined cyclone center and a clear moisture–dryness dipole aligned with the frontal zones, reflecting stronger preservation of baroclinic structure. In contrast, the operational configuration produces a broader low with spatially smeared humidity anomalies elongated towards the south.

GraphCast-OP also advances the low-pressure system eastward more rapidly than GraphCast-37. This is evident from the position of the central negative MSLP contours, with the GraphCast-OP frontal trough located roughly 5°–15° ahead of the GraphCast-37 trough, with the gap widening slightly each simulation day. These differences suggest variations in the learned representation of baroclinic wave phase speed. Overall, these results indicate that the increased vertical resolution in GraphCast-37 supports stronger baroclinic structure, slower dissipation of gradients, and more coherent downstream anomaly patterns, whereas the lower-resolution operational configuration exhibits greater diffusivity and a tendency toward faster eastward propagation.

### 4.3.3 Tropical Heating Response

Figure 4.4 displays the day-five response to the constant tropical heating experiment for both GraphCast-OP and GraphCast-37. The 500 hPa geopotential height DJF mean state is shown as the background via green contours. One of the most apparent features in both

simulations is the presence of widespread noisy accumulations in the geopotential height anomaly fields. These artifacts are more abundant here than in previous test cases, likely due to our choice to emphasize smaller-scale variability by contouring anomalies at 10 m intervals, beginning at ±5 m. This choice increases the visibility of weaker signals but also

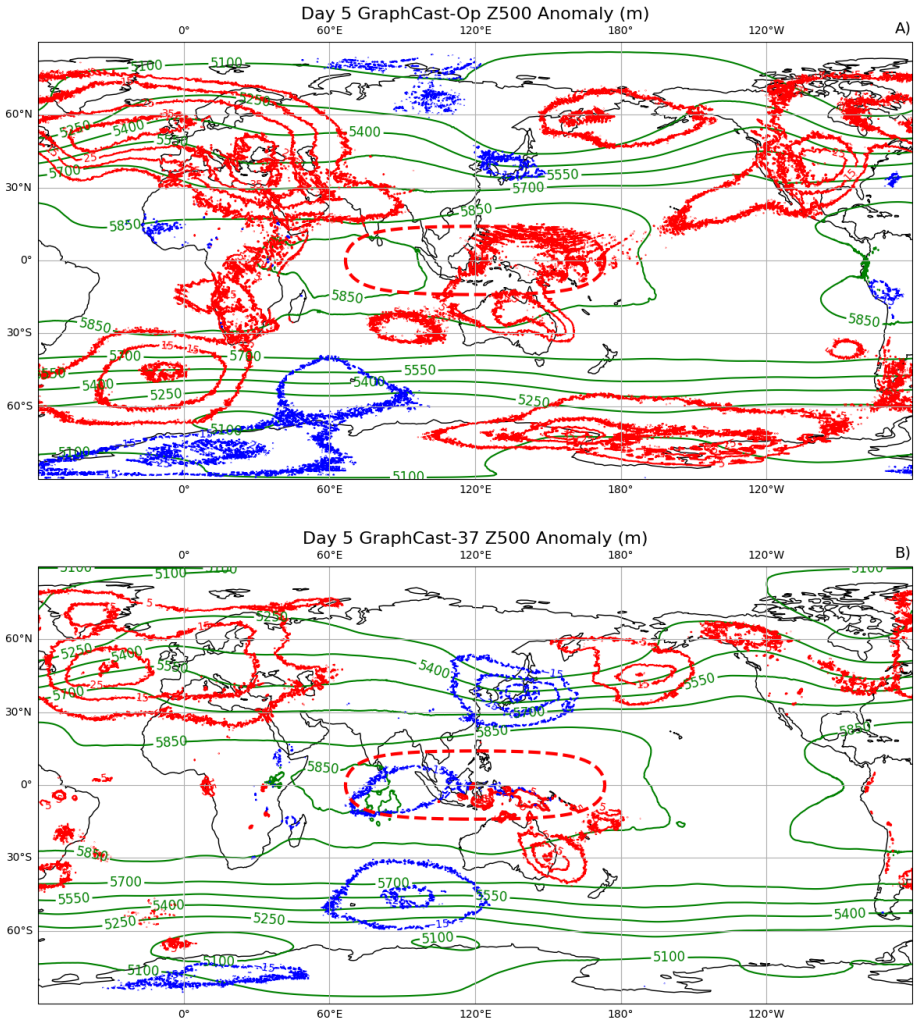



Figure 4.4: 500 hPa geopotential height anomaly field overlaid over DJF mean geopotential height (green contours) after 5 simulation days for GraphCast-OP (A) and GraphCast-37 (B). Positive and negative (red and blue, respectively) anomalies are shown at 10 m intervals with the 0 m contour being suppressed. Heating region is represented by the dashed red line.

highlights numerical noise and model artifacts more prominently.

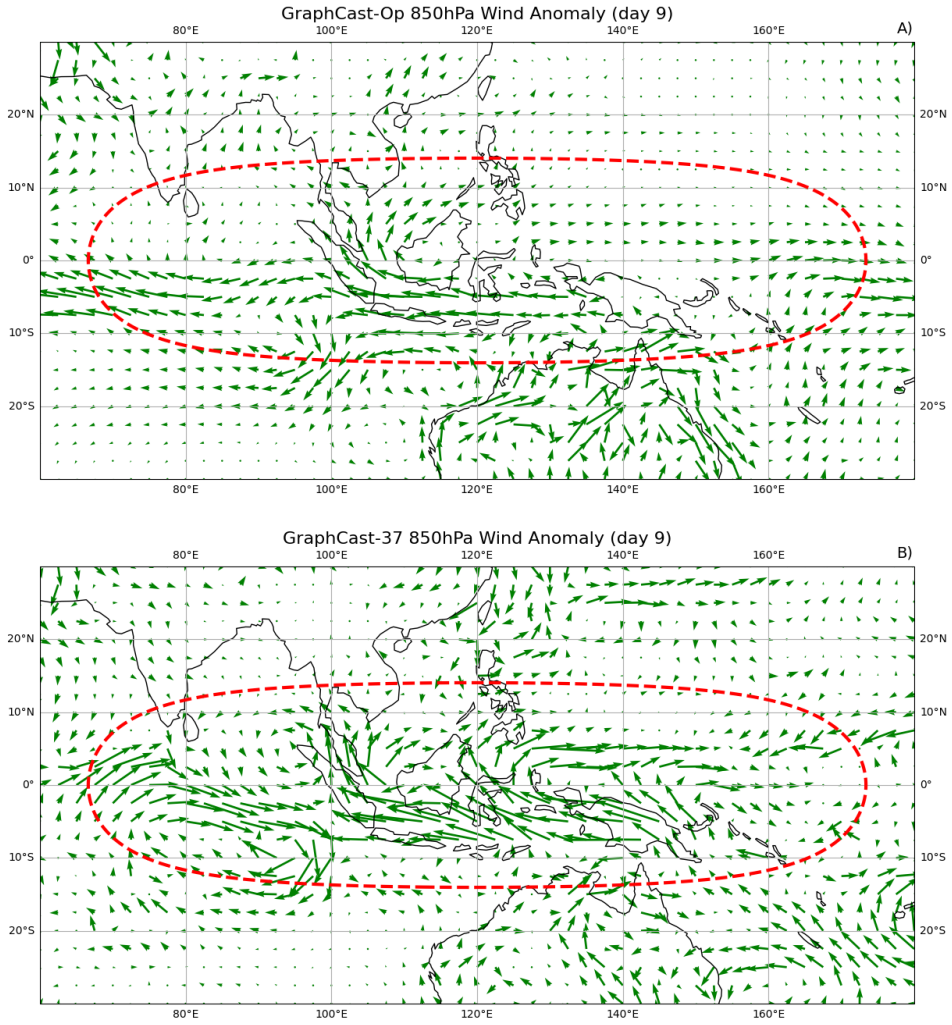



Figure 4.5: 850 hPa wind anomaly vector field near the region of heating (shown as red dashed line) for GraphCast-OP (top) and GraphCast-37 (bottom).

Qualitatively, GraphCast-OP simulation exhibits more pronounced noise accumulation than GraphCast-37, with artifacts appearing earlier and with greater spatial extent. These errors are especially evident over land regions, including substantial anomalies near Southeast Asia, where the tropical heating is applied. Importantly, while GraphCast-OP model does not exhibit any clear signs of an organized dynamical response by day five, GraphCast-37

shows a distinct planetary wave response at upper latitudes between 30°N and 60°N over the pacific ocean. This pattern bears qualitative similarity to the Northern Hemisphere wave train reported in Hakim and Masanam [2024] for their tropical heating experiments. Additionally, a prominent ridge appears in the Southern Hemisphere along the eastern edge of Australia, which may be part of a similar wave train observed in the day-20 results of Hakim and Masanam [2024], due to the applied warming anomaly.

Motivated by the analysis in Hakim and Masanam [2024], we also examine the near-surface wind vector anomalies in response to the constant tropical heating perturbation, as shown in Figure 4.5. This figure compares the day-nine wind anomaly fields for both GraphCast-Operational (top panel) and GraphCast-37 (bottom panel), with a particular focus on evaluating the presence of hemispheric gyres and equatorial wind convergence in the vicinity of the heating source.

By day nine, only the GraphCast-37 simulation exhibits the expected cyclonic gyres forming at the western edge of the heating source. These gyres are the result of the model developing convergent equatorial flow between approximately 80°E and 120°E. In this region, the wind anomaly vectors in the 37-level model demonstrate a clear convergence pattern: westerly anomalies (east-to-west) to the east of the heating and easterly anomalies (west-to-east) to the west, creating a convergent zone centered near 80°E and 120°E.

In contrast, the GraphCast-OP simulation shows consistently zonal flow along the equator in this region, with anomaly vectors largely oriented from east to west across the entire region, lacking evidence of convergence. This absence likely contributes to the failure of either of the cyclonic gyres to form in the operational model. While there are weak signals of possible rotation developing in the region around 100°E in the Southern hemisphere, the cyclonic motion we expect to see in response to this heating source fails to develop fully in operational GraphCast. The presence of both equatorial convergence and coherent gyres in the 37-level simulation indicates a more physically consistent dynamical response to the imposed tropical heating, similar to the findings by Hakim and Masanam [2024] and their dynamically time-stepped experiments with PanguWeather.

## 4.4 Thoughts, Discussion, and Future Steps

### 4.4.1 Difficulties with TR in the GraphCast Models

The application of TR to AI-driven weather forecasting models opens an intriguing set of questions about model sensitivity, initialization strategies, and dynamical fidelity. In particular, GraphCast presents a unique challenge due to its architectural design, raising the

question of whether TR is best suited for probing dynamical response in this model compared to others like SFNO or PanguWeather. While the TR framework offers a well-defined way to evaluate how a model reacts to an imposed perturbation, mimicking controlled numerical experiments from the dynamical core community, its effectiveness may be constrained by the inherent structural assumptions within GraphCast. These include its reliance on the TP06 variable (non-operational GraphCast versions) and its use of two discrete input states to generate a prediction, since TR was developed with a model that utilizes a single input state to predict the next. As a result, the traditional TR implementation, which assumes a single consistent initial condition, may not be ideal when applied to GraphCast without modification.

GraphCast requires two time-adjacent inputs, $x(t_0)$ and $x(t_{-1})$, to make a single prediction. In this work, we handle this by only applying our perturbed state to the $t_0$ state, leaving the $t_{-1}$ state as an unperturbed duplicate of our initial DJF mean state. Consequently, replicating the same initial state across both time inputs may unintentionally impact the initial prediction of the model, which is used consistently through the TR framework. This raises the question: should perturbations be applied to both $x(t_0)$ and $x(t_{-1})$ to more accurately mimic physical imbalances? Alternatively, should perturbations be changed when applied to each initial state?

Questions also remain regarding how the initial two states are implemented within the TR framework for GraphCast and other dual-input models. In this study, we duplicate the same DJF mean state for both initial timesteps prior to applying perturbations, but a sensitivity analysis is needed to determine whether this is the most effective approach. An optimal configuration may involve a different combination of initial states and a strategic choice of which timestep(s) receive a perturbation, potentially yielding a clearer and more physically interpretable dynamical response. Additional refinements could include supplying approximated precipitation fields for perturbed states. These are currently set to zero, but modifications may improve first-step predictions. Additional work should also strive to modify the TR mathematical formulation itself to better account for the dual-input architecture. Such adjustments reflect not only technical fine-tuning, but also broader questions about how AI models can be probed for dynamical signals and how those signals align with traditional physics-based systems.

Ultimately, the potential for TR to reveal meaningful insight in GraphCast is far from exhausted. As one of the most widely adopted and foundational models in the AI weather forecasting landscape, GraphCast presents a critical testbed for methodological innovation [NOAA]. The success of TR as a diagnostic or interpretive tool may depend not on its original formulation, but on our ability to apply it to various modeling frameworks. Further explo-

ration of perturbation design, input conditioning, and temporal sensitivity will be essential for ensuring that TR remains a robust method for evaluating model realism, particularly as the field moves toward more complex and effective AI-driven forecasting models.

### 4.4.2 Ripe with Research Opportunity

The rapid evolution of AI-driven weather prediction models in recent years has created an unprecedented landscape of research opportunity. Since the release of PanguWeather, a landmark moment that demonstrated how transformer-based architectures could rival and in some cases outperform traditional NWP systems, the pace of innovation has only accelerated [Bi et al., 2023]. Numerous models have since followed, each attempting to refine and expand upon the capabilities first demonstrated by early entrants like FourCastNet and GraphCast. These more recent models add to a diverse suite of emulators with varying architectures, enabling experimental and analytical studies of such systems.

Supporting this ecosystem is the emergence of platforms like Earth2Studio, NVIDIA's successor to the Earth2MIP framework [NVIDIA, 2025b]. Earth2Studio is designed as a comprehensive benchmarking and intercomparison environment for data-driven weather models, currently hosting around 20 publicly accessible models. While this framework represents another major step forward in standardizing evaluation practices across the community, it is still incomplete, several foundational models, including the original FourCastNet and the GraphCast-37 model we used in this study, are notably absent from the current Earth2Studio catalog. This gap underscores the fact that while tools for standardized usage are advancing, researchers still need to carefully select and configure models for meaningful intercomparison studies.

The diversity and availability of models, combined with the flexibility of platforms like Earth2MIP and Earth2Studio, open the door to a wide range of scientific inquiries. The DCMIP2025 effort began to scratch the surface of this space, extending the findings of Hakim and Masanam [2024] applying idealized dynamical test cases, to additional models like GraphCast-OP and SFNO. However, the potential for intercomparison runs far deeper. Novel frameworks, like TR, can be adapted to these models to study how they express dynamical responses, particularly in controlled settings. These analyses could be extended beyond wave propagation to interrogate more complex behavior: model sensitivity to moisture perturbations, mass and energy conservation, or general model stability under extreme conditions and forcings.

# CHAPTER 5

# Conclusion and Outlook

This dissertation explored the integration, deployment, and evaluation of Machine Learning (ML) techniques within weather and climate modeling frameworks. Through a sequence of three projects, we investigated both the potential and the limitations of ML-driven approaches across a spectrum of modeling contexts; from investigating the limits of various approaches to offline emulation of subgrid physics, to online coupling within a flagship atmospheric general circulation model, to expanding upon the novel approaches for dynamical testing of fully AI-based forecasting systems. These efforts collectively aimed to understand not only whether ML models can serve in critical roles within geophysical modeling frameworks, but under what conditions they do so effectively. On top of this, we interrogated whether these models behave in physically meaningful ways. In what follows, we synthesize the key findings from these investigations, discuss their broader implications, and outline opportunities for future research.

## 5.1 Emulation and Complexity: Offline Lessons

In the first chapter of this work, we focused on the emulation of simplified parameterization schemes using random forests (RFs) and neural networks (NNs) within the Community Atmosphere Model (CAM6). By building emulators across a hierarchy of CAM configurations, from dry dynamics to moist physics to schemes with simple convection, we demonstrated that RFs can effectively replicate model tendencies and precipitation rates with skill. However, as physical complexity increased, RF performance declined despite extensive tuning. This reveals a fundamental limitation of tree-based emulators in representing the complex and strongly interacting processes within atmospheric physics.

Within this chapter, we also found that incorporating domain knowledge into our feature selection can substantially improve emulator performance. A notable example was the inclusion of relative humidity (RH) as a predictor. Although RH is not an explicit input in

the functional form of the target parameterizations, adding it markedly improved the skill of both the RF and NN emulators. From a purely statistical perspective, this result is not obvious, but from an atmospheric science standpoint, it is intuitive, as RH serves as a proxy for supersaturation and indirectly encodes the presence of large-scale condensation processes that influence both temperature and moisture tendencies. This highlights the value of combining physical intuition with ML experimentation, both to guide feature studies and to better interpret model behavior.

More broadly, the structured, hierarchical benchmarking approach employed here proved to be an effective, low-cost strategy for diagnosing potential limitations of ML techniques. Limitations that can be considered before future investment and deployment of these techniques. The same methodology, gradually increasing process complexity, quantifying offline skill, and identifying scaling bottlenecks, can serve as a transferable framework for evaluating other ML parameterizations and approaches within the physical sciences. Finally, the lessons from this chapter directly shaped the subsequent chapters of this thesis. In particular, the exploration of the complexities of coupling strategies even within these simplified frameworks.

## 5.2 Deployment in CAM: Online Realities

Building on the offline emulator studies, the second project addressed the challenges of coupling ML-based parameterizations to the dynamical core of CAM in an online environment, a setting where even small numerical or physical imbalances can amplify into large-scale instabilities. In this chapter, we moved beyond purely statistical metrics to evaluate how emulators interact dynamically with the host model, comparing feed-forward NNs and RFs in terms of scalability, numerical stability, and implementation burden.

Our experiments revealed sharp contrasts between the two architectures. NNs, though compact in memory footprint and more amenable to GPU acceleration, proved highly sensitive to out-of-distribution inputs, especially in convectively active tropical regions where extreme heating and moisture tendencies may arise. Such sensitivity often manifested as runaway instabilities, highlighting the need for more robust and complex deep learning approaches, such as input sub-sampling, complex architectures, or physics-informed NNs [O'Gorman and Dwyer, 2018, Zhao et al., 2019, Kashinath et al., 2021]. In contrast, RFs exhibited a natural resilience to instability thanks to their bounded predictions, enabling them to survive longer integrations without unstable feedbacks. However, this stability came at a cost: skillful RFs required substantially more memory, produced higher inference latency, and scaled poorly when the number of features or output variables increased, making them

less attractive for online simulation.

A key outcome of this work was the recognition that the technical interface between Python-based ML models and CAM's Fortran-based infrastructure can be as critical as the choice of model architecture itself. Coupling overhead, from library linking difficulties to I/O concerns, can limit the feasibility of deploying ML emulators into online simulations, particularly when working within the framework of a highly optimized software environment like CAM6. We identified several engineering pathways to mitigate this bottleneck, including the adoption of emerging hybrid toolchains such as FTORCH and CREDIT, which bring ML inference closer to the Fortran execution layer [Atkinson et al., 2025, Chapman et al., 2025, Schreck et al., 2025]. These findings emphasize that successful ML–GCM integration is not merely a matter of achieving high offline skill: it is an inherently interdisciplinary challenge spanning software systems design, high-performance computing, numerical analysis, and atmospheric science.

## 5.3 Probing AI-Driven Weather Prediction Models

The final chapter of this dissertation shifted focus from the use of ML as a replacement for traditional parameterizations toward ML as the core engine of a new class of data-driven global forecasting systems. In this context, we leveraged the TR methodology, originally proposed by Hakim and Masanam [2024], and our implementation of TR within the Earth2MIP inference framework for DCMIP2025. The primary model that we utilized in this chapter was GraphCast, a state-of-the-art AI-based global weather forecasting model, selected both for its demonstrated skill and its publicly available research implementation. We adapted the TR method to GraphCast's 37-level vertical configuration to evaluate its variance in wave response relative to the 13-level operational version, as well as to probe vertical wave propagation and characteristics in the 37-level configuration under an imposed tropical heating anomaly. These experiments represented, to our knowledge, the first application of TR to an AI forecasting system at such high vertical resolutions, providing a unique window into how these models represent dynamical adjustments and energy propagation in pseudo-idealized settings.

The TR framework proved to be a valuable diagnostic tool. Architectural aspects of GraphCast, most notably its requirement for two time-adjacent inputs to produce a forecast, complicated the method of initializing our perturbations, introducing ambiguity when applying our testcases and TR framework. These challenges underscore an important distinction from traditional numerical models: AI forecasting systems often embed temporal context directly into their inference architectures. While this choice can enhance predictive

skill, it may limit the effectiveness of controlled experiments. This observation suggests that dynamical diagnostics like TR may not be fully agnostic to model design, while still yielding insightful responses in our tests. Future improvements of TR, as well as developments of alternative testing frameworks, may benefit from being tailored to the temporal encoding, spatial discretization, and state-variable representation of individual models.

Despite these hurdles, the TR-based experiments demonstrated the feasibility and value of porting idealized dynamical test cases, long a staple of physics-based model evaluation, into the AI modeling domain. The results offer an early template for community-driven frameworks that can assess dynamical properties across AI models. In particular, embedding TR and similar diagnostics within cross-platform toolchains such as Earth2MIP, and its emerging successor Earth2Studio, opens the door to comparative studies spanning the various models being developed worldwide. Such a capability would not only deepen our physical understanding of AI forecasting systems but would also help identify architectural or training-driven biases that might otherwise go unnoticed in conventional skill metrics.

## 5.4 Broader Implications and Future Directions

The work presented in this dissertation underscores that the most impactful applications of ML for emulation in modeling the atmosphere will require targeted emulation of tasks that align with the inherent strengths of ML. Rather than attempting to replicate full model tendencies, future efforts may yield greater benefits by focusing on individual schemes or processes where ML can offer substantial speed-up, improved numerical stability, or enhanced representation of localized phenomena. This targeted approach is particularly promising for computationally intensive physical parameterizations, such as cloud microphysics, radiation, or chemistry, where ML surrogates can deliver efficiency gains without compromising interpretability, stability, or physical realism.

Equally important is the need to design AI and ML solutions with their eventual deployment environment in mind. For models intended to operate within existing operational or research forecast systems, often written in legacy Fortran, the architecture, training, and deployment pipelines should be constructed from the outset to interface cleanly with these environments. For example, using frameworks such as PyTorch, combined with interoperability libraries like FTorch, can significantly streamline the integration of ML components into Fortran-based workflows [Atkinson et al., 2025]. This co-design mindset ensures that promising prototypes can transition into real-world applications without major computational overhead.

At the same time, AI-driven weather and climate models are being developed at a pace

that far exceeds the current ability to systematically assess their dynamical and physical fidelity. Without standardized benchmarks, comparisons between models risk becoming inconsistent or incomplete, limiting scientific understanding and operational trust. The extension of dynamical testing frameworks, such as those utilized and developed in this thesis, offers a path toward reproducible, architecture-aware evaluation tools that can probe the types of dynamics and physics learned by these systems. Such frameworks, when integrated into shared infrastructures like Earth2Studio, have the potential to serve as community-wide standards, fostering comparability, transparency, and physical interpretability.

Looking ahead, this line of work highlights two intertwined frontiers. First, there is a pressing need to co-design AI architectures and diagnostic tools so that evaluation is not an afterthought but an integrated aspect of model development. Second, fostering shared, reproducible benchmarks for AI models will require active collaboration between the AI, NWP, and climate modeling communities. Bridging these worlds offers the potential for a new generation of forecasting systems, ones whose skill is matched by their physical interpretability and whose evaluation frameworks are as rigorous as those used for their numerical predecessors.

## 5.5 Final Thoughts

This dissertation has spanned a continuum, from emulating components of climate models, to embedding them in real-time systems, to testing entirely new architectures that forecast weather without explicit physics. At each stage, we have uncovered not only the promise of ML in advancing climate science, but the pitfalls and puzzles that still lie ahead.

We find ourselves at a pivotal moment. The accelerating pace of ML development offers remarkable new tools for climate modeling, but with these tools comes a responsibility to question, to test, and to interpret. As researchers and practitioners in this space, we must remain critical, collaborative, and grounded in physical understanding. The work presented here is one step toward a future where ML and climate science work in tandem: where the strengths of each can offset the limitations of the other, and where our models are not only fast and accurate, but interpretable, reliable, and physically sound.

# APPENDIX A

# Chapter 2 Supplemental Information

**Text S1. Aquaplanet Details**

The aquaplanet configuration was used to inform parameter choices for the BM convection scheme discussed in section 2.1. An aquaplanet is an ocean-covered model with prescribed sea surface temperatures (SST) in which the exchange of heat and moisture between the ocean and the atmosphere provides additional quasi-realistic atmospheric fluid flow. It is a widely used configuration for simplified physics studies of GCMs. We used the aquaplanet configuration with the older CAM4 physics package with the CONTROL SST profile configuration described in Neale and Hoskins [2000] to guide our choice of $\mathrm{RH_{BM}}$ and $\tau$ in the BM scheme Neale et al. [2010]. Zonal-mean, time-mean fields for various model output fields comparing the aquaplanet and the convection scheme are shown in Figures A.1 and A.2 and were used to inform our decision for the chosen parameters.

While we acknowledge that these two cases are not identical, there are many fields with similar flow characteristics. In particular, the temperature, specific humidity, relative humidity, zonal wind, and precipitation rates share many similarities in their averaged profiles. The physical tendencies in Figures A.1d,e and A.2d,e display greater differences. However, this is expected as the complexity of the physical parameterizations differs. All cases are run at the same $1.9 \times 2.5$ degree spatial resolution with 30 model levels. Since the CONTROL case for the aquaplanet setup in CAM4 is not the default setup, we note here that the compset 'long name' format is
"2000_CAM40_SLND_SICE_DOCN%AQP1_SROF_SGLC_SWAV".
This is needed to reproduce Figure A.1.

**Text S2. Machine Learning Hyperparameter Tuning**

Parameters like the number of trees in an RF, the number of training samples, as well as the choice of activation functions in a neural network are examples of hyperparameters. These impact the effectiveness of the emulators. The majority of the RF parameters for this study were chosen via the SHERPA hyperparameter optimization library. Tables A.1 to A.8

show the hyperparameter choices for the various RF emulators. For further details on the RF parameters and how they work to impact the overall model, we direct the reader to the SciKit-Learn documentation Pedregosa et al. [2011]. We also show choices for the neural network setups in able A.9, all of which were informed by Beucler et al. [2021]. Each field uses an identical setup, however precipitation rates use a *sigmoid* activation (rather than *tanh*) on the final layer in order to enforce positive-definite solutions. Our NNs also use Keras' Normalization layer for our features in order to transform the input to be unitarily invariant, see Keras documentation for further information on this normalization process Chollet [2017]. The symbols RELHUM, LHFLX, and SHFLX stand for the relative humidity, surface latent heat flux, and surface sensible heat flux, respectively. We note that upon review we found that reducing the number of trees in our RFs from the SHERPA suggestion down to 50 trees across each configuration did not noticeably impact our results. Therefore, we kept the number of trees consistent across all RF models at 50 trees.

**Figures**

**Tables**

Table A.1: Dry dT/dt Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $\phi$ |
| Number of Samples | 20 Million |
| Number of Trees | 50 |
| Max Depth | 39 |
| Min Samples Split | 17 |
| Min Samples Leaf | 6 |

Table A.2: Moist dT/dt Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 15 Million |
| Number of Trees | 50 |
| Max Depth | 30 |
| Min Samples Split | 20 |
| Min Samples Leaf | 15 |

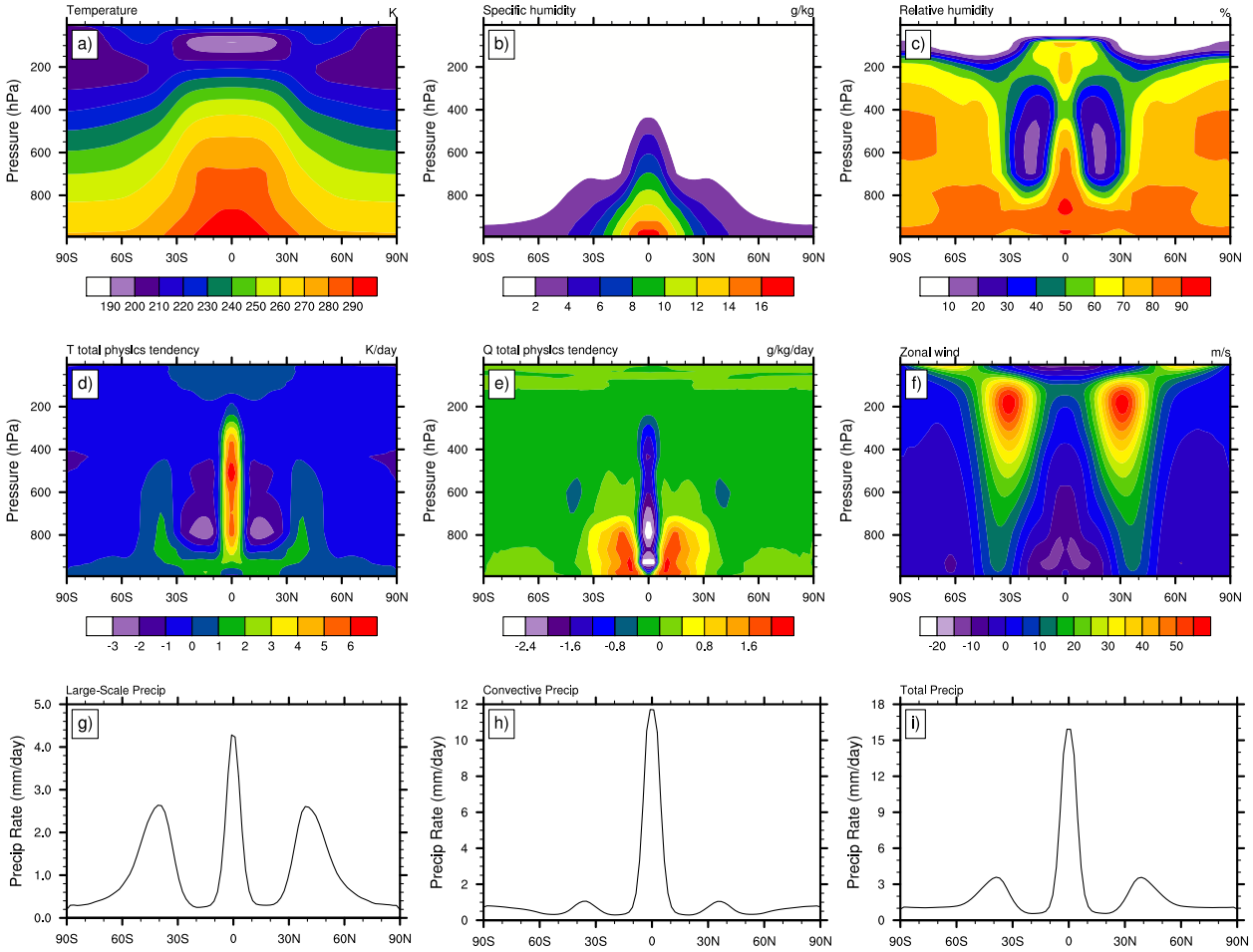



Figure A.1: Zonal-mean time-mean panel of (a) temperature, (b) specific humidity, (c) relative humidity, (d) temperature tendency, (e) moisture tendency, (f) zonal wind, (g) large-scale precipitation, (h) convective precipitation, (i) total precipitation rate for the CAM4 aquaplanet setup with the CONTROL SST profile.

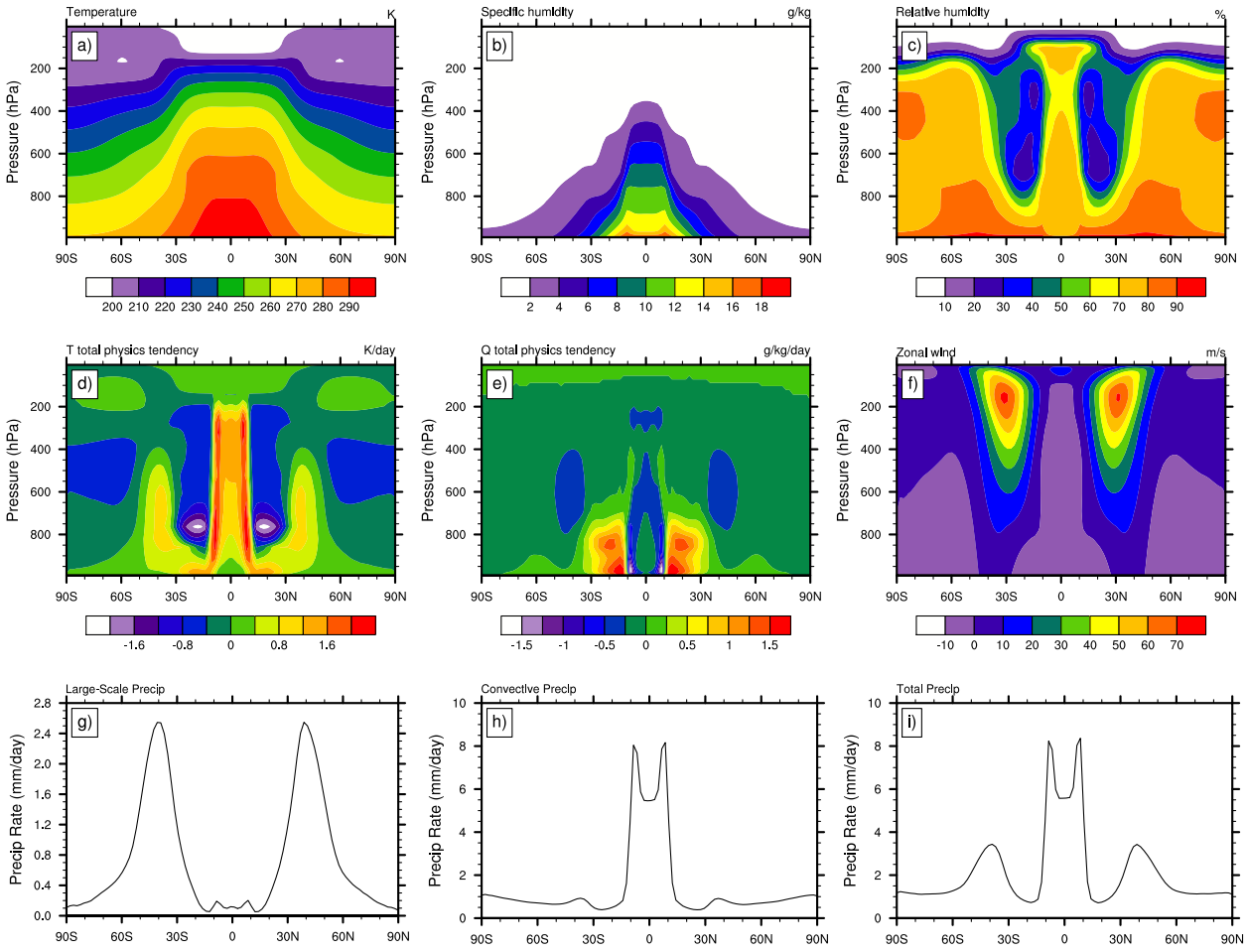



Figure A.2: Zonal-mean time-mean panel of (a) temperature, (b) specific humidity, (c) relative humidity, (d) temperature tendency, (e) moisture tendency, (f) zonal wind, (g) large-scale precipitation, (h) convective precipitation, (i) total precipitation rate for the TJ16 configuration in CAM6 coupled with the BM convection scheme with $\tau = 4$ hr and $\mathrm{RH}_{\mathrm{BM}} = 0.7$.

Table A.3: Convection dT/dt Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 15 Million |
| Number of Trees | 50 |
| Max Depth | 22 |
| Min Samples Split | 23 |
| Min Samples Leaf | 18 |

Table A.4: Moist dq/dt Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 20 Million |
| Number of Trees | 50 |
| Max Depth | 30 |
| Min Samples Split | 45 |
| Min Samples Leaf | 15 |

Table A.5: Convection dq/dt Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 20 Million |
| Number of Trees | 50 |
| Max Depth | 32 |
| Min Samples Split | 19 |
| Min Samples Leaf | 17 |

Table A.6: Moist Large-Scale Precipitation Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 20 Million |
| Number of Trees | 50 |
| Max Depth | 30 |
| Min Samples Split | 30 |
| Min Samples Leaf | 5 |

Table A.7: Convection Large-Scale Precipitation Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 20 Million |
| Number of Trees | 50 |
| Max Depth | 30 |
| Min Samples Split | 30 |
| Min Samples Leaf | 5 |

Table A.8: Convection Convective Precipitation Hyperparameters

| RF Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 20 Million |
| Number of Trees | 50 |
| Max Depth | 37 |
| Min Samples Split | 2 |
| Min Samples Leaf | 11 |

Table A.9: Neural Netork Setup/Hyperparameters

| NN Option | Choice |
|---|---|
| Input Variables | $T$, $p$, $q$, RELHUM, LHFLX, SHFLX |
| Number of Samples | 12.8 Million |
| Number of Layers | 8 |
| Nodes per Layer | 512 |
| Hidden Layer Activation | LeakyReLU ($\alpha = 0.25$) |
| Output Layer Activation | tanh (sigmoid for precip) |
| Dropout Rate | 0.001 |
| Loss Function | MSE |
| Batch Size | 128 |
| Epochs | 15 |
| Optimizer | Adam (learningRate= 0.00001) |

# BIBLIOGRAPHY


Jack Atkinson, Athena Elafrou, Elliott Kasoar, Joseph G. Wallwork, Thomas Meltzer, Simon Clifford, Dominic Orchard, and Chris Edsall. FTorch: a library for coupling PyTorch models to Fortran. *Journal of Open Source Software*, 10(107):7602, March 2025. doi: 10.21105/joss.07602.

Junjie Bai, Fang Lu, Ke Zhang, and others. ONNX: Open Neural Network Exchange, 2019. URL https://github.com/onnx/onnx.

Pierre. Baldi. *Deep Learning in Science: Theory, Algorithms, and Applications*. Cambridge University Press, Cambridge, England, 2021.

Elizabeth A. Barnes, James W. Hurrell, Imme Ebert-Uphoff, Chuck Anderson, and David Anderson. Viewing Forced Climate Patterns Through an AI Lens. *Geophys. Res. Lett.*, 46(22):13389–13398, 2019. doi: 10.1029/2019GL084944.

A. K. Betts. A new convective adjustment scheme. Part I: Observational and theoretical basis. *Quart. J. Roy. Meteor. Soc.*, 112:677–692, 1986.

A. K. Betts and M. J. Miller. A new convective adjustment scheme. Part II: Single column tests using GATE wave, BOMEX, and arctic air-mass data sets. *Quart. J. Roy. Meteor. Soc.*, 112:693–709, 1986.

Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. Enforcing Analytic Constraints in Neural-Networks Emulating Physical Systems. *Phys. Rev. Lett.*, 126:098302, 2021. doi: 10.1103/PhysRevLett.126.098302.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970): 533–538, July 2023. doi: 10.1038/s41586-023-06185-3.

Vilhelm Bjerknes. The problem of weather prediction, considered from the viewpoints of mechanics and physics. *Meteorologische Zeitschrift*, 18(6):663–667, 1904. doi: 10.1127/0941-2948/2009/416.

M. Blackburn, D. L. Williamson, K. Nakajima, W. Ohfuchi, Y. O. Takahashi, Y.-Y. Hayashi, H. Nakamura, M. Ishiwatari, J. McGregor, H. Borth, V. Wirth, H. Frank, P. Bechtold, N. P. Wedi, H. Tomita, M. Satoh, M. Zhao, I. M. Held, M. J. Suarez, M.-I. Lee, M. Watanabe, M. Kimoto, Y. Liu, Z. Wang, A. Molod, K. Rajendran, A. Kitoh, and R. Stratton.

The Aqua-Planet Experiment (APE): Control SST simulation. *J. Meteor. Soc. Japan*, 91A:17–56, 2013.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the Earth system. *Nature*, 641(8065):1180–1187, May 2025. doi: 10.1038/s41586-025-09005-y.

P. A. Bogenschutz, A. Gettelman, H. Morrison, V. E. Larson, C. Craig, and D. P. Schanen. Higher-Order Turbulence Closure and Its Impact on Climate Simulations in the Community Atmosphere Model. *J. Climate*, 26:9655–9676, December 2013.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical Fourier neural operators: learning stable dynamics on the sphere. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 2806–2823. JMLR.org, July 2023.

Ann Bostrom, Julie L. Demuth, Christopher D. Wirz, Mariana G. Cains, Andrea Schumacher, Deianna Madlambayan, Akansha Singh Bansal, Angela Bearth, Randy Chase, Katherine M. Crosman, Imme Ebert-Uphoff, David John Gagne II, Seth Guikema, Robert Hoffman, Branden B. Johnson, Christina Kumler-Bonfanti, John D. Lee, Anna Lowe, Amy McGovern, Vanessa Przybylo, Jacob T. Radford, Emilie Roth, Carly Sutter, Philippe Tissot, Paul Roebber, Jebb Q. Stewart, Miranda White, and John K. Williams. Trust and trustworthy artificial intelligence: A research agenda for AI in the environmental sciences. *Risk Analysis*, 44(n/a):1498–1513, 2024. doi: 10.1111/risa.14245.

Sid Ahmed Boukabara, Vladimir Krasnopolsky, Stephen G. Penny, Jebb Q. Stewart, Amy McGovern, David Hall, John E. Ten Hoeve, Jason Hickey, Hung Lung Allen Huang, John K. Williams, Kayo Ide, Philippe Tissot, Sue Ellen Haupt, Kenneth S. Casey, Nikunj Oza, Alan J. Geer, Eric S. Maddy, and Ross N. Hoffman. Outlook for exploiting artificial intelligence in the Earth and environmental sciences. *Bulletin of the American Meteorological Society*, 102(5):E1016–E1023, 2021. doi: 10.1175/BAMS-D-20-0031.1.

Leo Breiman. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1007/BF00058655.

Noah D. Brenowitz. nbren12/call_py_fort. URL https://github.com/nbren12/call_py_fort.

Noah. D. Brenowitz and Christopher S. Bretherton. Prognostic Validation of a Neural Network Unified Physics Parameterization. *Geophys. Res. Lett.*, 45(12):6289–6298, 2018. doi: 10.1029/2018GL078510.

Noah D. Brenowitz and Christopher S. Bretherton. Spatially Extended Tests of a Neural Network Parametrization Trained by Coarse-Graining. *J. Adv. Model. Earth Syst.*, 11(8):2728–2744, 2019. doi: 10.1029/2019MS001711.

Noah D. Brenowitz, Tom Beucler, Michael Pritchard, and Christopher S. Bretherton. Interpreting and Stabilizing Machine-Learning Parametrizations of Convection. *J. Atmos. Sci.*, 77(12):4357–4375, 2020. doi: 10.1175/JAS-D-20-0082.1.

Christopher S. Bretherton, Brian Henn, Anna Kwa, Noah D. Brenowitz, Oliver Watt-Meyer, Jeremy McGibbon, W. Andre Perkins, Spencer K. Clark, and Lucas Harris. Correcting Coarse-Grid Weather and Climate Models by Machine Learning From Global Storm-Resolving Simulations. *J. Adv. Model. Earth Syst.*, 14(2), 2022. doi: 10.1029/2021MS002794.

Matthew Chantry, Hannah Christensen, Peter Düben, and Tim Palmer. Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft AI. *Phil. Trans. R. Soc. A*, 379(2194), 2021. doi: 10.1098/rsta.2020.0083.

W. E. Chapman, A. C. Subramanian, L. Delle Monache, S. P. Xie, and F. M. Ralph. Improving Atmospheric River Forecasts With Machine Learning. *Geophys. Res. Lett.*, 46 (17–18):10627–10635, 2019. doi: 10.1029/2019GL083662.

William E. Chapman, John S. Schreck, Yingkai Sha, David John Gagne II, Dhamma Kimpara, Laure Zanna, Kirsten J. Mayer, and Judith Berner. CAMulator: Fast Emulation of the Community Atmosphere Model, April 2025.

Francois Chollet. *Deep Learning with Python*. Manning Publications Co., 384 pages, USA, 1st edition, 2017.

Kamal Choudhary, Brian DeCost, Chi Chen, Anubhav Jain, Francesca Tavazza, Ryan Cohn, Cheol Woo Park, Alok Choudhary, Ankit Agrawal, Simon J. L. Billinge, Elizabeth Holm, Shyue Ping Ong, and Chris Wolverton. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials*, 8(1):1–26, April 2022. doi: 10.1038/s41524-022-00734-6.

David S. Connelly and Edwin P. Gerber. Regression Forest Approaches to Gravity Wave Parameterization for Climate Projection. *Journal of Advances in Modeling Earth Systems*, 16(7), 2024. doi: 10.1029/2023MS004184.

Milan Curcic. A parallel Fortran framework for neural networks and deep learning. *SIGPLAN Fortran Forum*, 38(1):4–21, March 2019. doi: 10.1145/3323057.3323059.

Aiguo Dai and T. M. L. Wigley. Global patterns of ENSO-induced precipitation. *Geophysical Research Letters*, 27(9):1283–1286, 2000. doi: 10.1029/1999GL011140.

G. Danabasoglu, J.-F. Lamarque, J. Bacmeister, D. A. Bailey, A. K. DuVivier, J. Edwards, L. K. Emmons, J. Fasullo, R. Garcia, A. Gettelman, C. Hannay, M. M. Holland, W. G. Large, P. H. Lauritzen, D. M. Lawrence, J. T. M. Lenaerts, K. Lindsay, W. H. Lipscomb, M. J. Mills, R. Neale, K. W. Oleson, B. Otto-Bliesner, A. S. Phillips, W. Sacks, S. Tilmes, L. van Kampenhout, M. Vertenstein, A. Bertini, J. Dennis, C. Deser, C. Fischer, B. Fox-Kemper, J. E. Kay, D. Kinnison, P. J. Kushner, V. E. Larson, M. C. Long, S. Mickelson, J. K. Moore, E. Nienhouse, L. Polvani, P. J. Rasch, and W. G. Strand. The Community

Earth System Model Version 2 (CESM2). *J. Adv. Model. Earth Syst.*, 12(2), 2020. doi: 10.1029/2019MS001916.

Imme Ebert-Uphoff and Kyle Hilburn. The outlook for AI weather prediction. *Nature*, 619 (7970):473–474, July 2023. doi: 10.1038/d41586-023-02084-9.

Joshua Elms. DCMIP2025 Idealized Tests, June 2025. URL https://github.com/Joshua-Elms/dcmip2025_idealized_tests.

Gregory M. Flato. Earth system models: an overview. *WIREs Climate Change*, 2(6):783–800, 2011. doi: 10.1002/wcc.148.

Montgomery L. Flora, Corey K. Potvin, Amy McGovern, and Shawn Handler. A Machine Learning Explainability Tutorial for Atmospheric Sciences. *Artificial Intelligence for the Earth Systems*, 3(1), January 2024. doi: 10.1175/AIES-D-23-0018.1.

Dallas Foster, David John Gagne, and Daniel B. Whitt. Probabilistic Machine Learning Estimation of Ocean Mixed Layer Depth From Dense Satellite and Sparse In Situ Observations. *J. Adv. Model. Earth Syst.*, 13(12):1–33, 2021. doi: 10.1029/2021MS002474.

Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, February 2002. doi: 10.1016/S0167-9473(01)00065-2.

D. M. W. Frierson. The Dynamics of Idealized Convection Schemes and Their Effect on the Zonally Averaged Tropical Circulation. *J. Atmos. Sci.*, 64:1959–1976, 2007.

David John Gagne, Amy McGovern, Sue Ellen Haupt, Ryan A. Sobash, John K. Williams, and Ming Xue. Storm-Based Probabilistic Hail Forecasting with Machine Learning Applied to Convection-Allowing Ensembles. *Weather and Forecasting*, 32(5):1819–1840, 2017. doi: 10.1175/WAF-D-17-0010.1.

Gregory Gaspari and Stephen E. Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, 1999. doi: 10.1002/qj.49712555417.

P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis. Could Machine Learning Break the Convection Parameterization Deadlock? *Geophys. Res. Lett.*, 45(11):5742–5751, 2018. doi: 10.1029/2018GL078202.

A. Gettelman and H. Morrison. Advanced Two-Moment Bulk Microphysics for Global Models. Part I: Off-Line Tests and Comparison with Other Schemes. *J. Climate*, 28(3):1268–1287, 2015. doi: 10.1175/JCLI-D-14-00102.1.

A. Gettelman, D. J. Gagne, C.-C. Chen, M. W. Christensen, Z. J. Lebo, H. Morrison, and G. Gantos. Machine Learning the Warm Rain Process. *J. Adv. Model. Earth Syst.*, 13(2), 2021. doi: 10.1029/2020MS002268.

Gregory J. Hakim and Sanjit Masanam. Dynamical Tests of a Deep Learning Weather Prediction Model. *Artificial Intelligence for the Earth Systems*, 3(3), July 2024. doi: 10.1175/AIES-D-23-0090.1.

Yilun Han, Guang J. Zhang, Xiaomeng Huang, and Yong Wang. A Moist Physics Parameterization Based on Deep Learning. *J. Adv. Model. Earth Syst.*, 12(9), 2020. doi: 10.1029/2020MS002076.

I. M. Held and M. J. Suarez. A proposal for the Intercomparison of the Dynamical Cores of Atmospheric General Circulation Models. *Bull. Amer. Meteor. Soc.*, 75(10):1825–1830, October 1994.

Isaac M. Held. The gap between simulation and understanding in climate modeling. *Bulletin of the American Meteorological Society*, 86(11):1609–1614, 2005. doi: 10.1175/BAMS-86-11-1609.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

Lars Hertel, Julian Collado, Peter Sadowski, Jordan Ott, and Pierre Baldi. Sherpa: Robust Hyperparameter Optimization for Machine Learning. *SoftwareX*, 12:100591, 2020. doi: 10.1016/j.softx.2020.100591.

Helge Heuer, Mierk Schwabe, Pierre Gentine, Marco A. Giorgetta, and Veronika Eyring. Interpretable Multiscale Machine Learning-Based Parameterizations of Convection for ICON. *Journal of Advances in Modeling Earth Systems*, 16(8), 2024. doi: 10.1029/2024MS004398.

F. Hourdin, Thorsten Mauritsen, Andrew Gettelman, Jean-Christophe Golaz, Venkatramani Balaji, Qingyun Duan, Doris Folini, Duoying Ji, Daniel Klocke, Yun Qian, Florian Rauser, Catherine Rio, Lorenzo Tomassini, Masahiro Watanabe, and Daniel Williamson. The Art and Science of Climate Model Tuning. *Bull. Ameri. Meteor. Soc.*, 98(3):589–602, 2017. doi: 10.1175/BAMS-D-15-00135.1.

Fredric Hourdin and A. Armengaud. The Use of Finite-Volume Methods for Atmospheric Advection of Trace Species. Part I: Test of Various Formulations in a General Circulation Model. *Mon. Wea. Rev.*, 127(5):822–837, 1999.

Stephan Hoyer and Joe Hamman. xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*, 5(1):10, 2017. doi: 10.5334/jors.148.

Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances

in the era of big data. *Information Sciences*, 622:178–210, April 2023. doi: 10.1016/j.ins.2022.11.139.

M. Z. Jacobson. *Fundamentals of atmospheric modeling*. Cambridge University Press, second edition, 2005.

Andrew D. Justin, Colin Willingham, Amy McGovern, and John T. Allen. Toward Operational Real-Time Identification of Frontal Boundaries Using Machine Learning. *Artificial Intelligence for the Earth Systems*, 2(3), July 2023. doi: 10.1175/AIES-D-22-0052.1.

Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine Learning for the Geosciences : Challenges and Opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, 2019. doi: 10.1109/TKDE.2018.2861006.

K. Kashinath, M. Mustafa, A. Albert, J-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, and null Prabhat. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194), February 2021. doi: 10.1098/rsta.2020.0093.

Jan D. Keller and Roland Potthast. AI-based data assimilation: Learning the functional of analysis estimation, June 2024.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. Neural General Circulation Models for Weather and Climate. *Nature*, 632(8027):1060–1066, August 2024. doi: 10.1038/s41586-024-07744-y.

Vladimir M. Krasnopolsky and Michael S. Fox-Rabinovitz. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2):122–134, 2006. doi: 10.1016/j.neunet.2006.01.002.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. FourCastNet: Accelerating Global High-Resolution Weather Forecasting Using Adaptive Fourier Neural Operators. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–11, June 2023. doi: 10.1145/3592979.3593412.

Ryan Lagerquist, Amy McGovern, and Travis Smith. Machine Learning for Real-Time Prediction of Damaging Straight-Line Convective Wind. *Weather and Forecasting*, 32(6):2175–2193, December 2017. doi: 10.1175/WAF-D-17-0038.1.

Ryan Lagerquist, Amy McGovern, and David John Gagne Ii. Deep Learning for Spatially Explicit Prediction of Synoptic-Scale Fronts. *Weather and Forecasting*, 34(4):1137–1160, August 2019. doi: 10.1175/WAF-D-18-0183.1.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, November 2023. doi: 10.1126/science.adi2336.

Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallègue, Ana Prieto Nemesio, Peter D. Dueben, Andrew Brown, Florian Pappenberger, and Florence Rabier. AIFS – ECMWF's data-driven forecasting system, August 2024.

Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13), March 2024. doi: 10.1126/sciadv.adk4489.

Garrett C. Limon and Christiane Jablonowski. Probing the Skill of Random Forest Emulators for Physical Parameterizations Via a Hierarchy of Simple CAM6 Configurations. *Journal of Advances in Modeling Earth Systems*, 15(6), 2023. doi: 10.1029/2022MS003395.

S.-J. Lin. A "Vertically Lagrangian" Finite-Volume Dynamical Core for Global Models. *Mon. Wea. Rev.*, 132:2293–2307, October 2004.

Eric D. Loken, Adam J. Clark, Amy McGovern, Montgomery Flora, and Kent Knopfmeier. Postprocessing Next-Day Ensemble Probabilistic Precipitation Forecasts Using Random Forests. *Weather and Forecasting*, 34(6):2017–2044, December 2019. doi: 10.1175/WAF-D-19-0109.1.

Eric D. Loken, Adam J. Clark, and Amy McGovern. Comparing and Interpreting Differently Designed Random Forests for Next-Day Severe Weather Hazard Prediction. *Weather and Forecasting*, 37(6):871–899, June 2022. doi: 10.1175/WAF-D-21-0138.1.

E. N. Lorenz. *Predictability: a problem partly solved*. PhD Thesis, ECMWF, Shinfield Park, Reading, 1995.

Tianjiao Ma, Wen Chen, Xiadong An, Chaim I. Garfinkel, and Qingyu Cai. Nonlinear Effects of the Stratospheric Quasi-Biennial Oscillation and ENSO on the North Atlantic Winter Atmospheric Circulation. *Journal of Geophysical Research: Atmospheres*, 128(17), 2023. doi: 10.1029/2023JD039537.

Zhanshan Ma, Chuanfeng Zhao, Jiandong Gong, Jin Zhang, Zhe Li, Jian Sun, Yongzhu Liu, Jiong Chen, and Qingu Jiang. Spin-up characteristics with three types of initial fields and the restart effects on forecast accuracy in the GRAPES global forecast system. *Geoscientific Model Development*, 14(1):205–221, January 2021. doi: 10.5194/gmd-14-205-2021.

Antonios Mamalakis, Elizabeth A. Barnes, and Imme Ebert-Uphoff. Investigating the Fidelity of Explainable Artificial Intelligence Methods for Applications of Convolutional Neural Networks in Geoscience. *Artificial Intelligence for the Earth Systems*, 1(4), October 2022. doi: 10.1175/AIES-D-22-0012.1.

Amy McGovern, Imme Ebert-Uphoff, David John Gagne, and Ann Bostrom. Why we need to focus on developing ethical, responsible, and trustworthy artificial intelligence approaches for environmental science. *Environmental Data Science*, 1, January 2022. doi: 10.1017/eds.2022.5.

Amy McGovern, Randy J. Chase, Montgomery Flora, David J. Gagne, Ryan Lagerquist, Corey K. Potvin, Nathan Snook, and Eric Loken. A Review of Machine Learning for Convective Weather. *Artificial Intelligence for the Earth Systems*, 2(3), July 2023. doi: 10.1175/AIES-D-22-0077.1.

B. Medeiros, D. L Williamson, and J. G Olson. Reference aquaplanet climate in the Community Atmosphere Model, version 5. *J, Adv. Model. Earth Syst.*, 8(1):406–424, 2016.

G. A. Meehl, T. F. Stocker, W. D. Collins, P. Friedlingstein, A. T. Gaye, J. M. Gregory, A. Kitoh, R. Knutti, J. M. Murphy, A. Noda, S. C. B. Raper, I. G. Watterson, A. J. Weaver, and Z.-C. Zhao. Global Climate Projections. In S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis amd K. B. Averyt, M. Tignor, and H. L. Miller, editors, *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 747–845. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007.

R. B. Neale and B. J. Hoskins. A standard test for AGCMs including their physical parameterizations: I: The proposal. *Atmos. Sci. Lett.*, 1:101–107, 2000.

R. B. Neale, C.-C. Chen, A. Gettelman, P. H. Lauritzen, S. Park, D. L. Williamson, A. J. Conley, R. Garcia, D. Kinnison, J.-F. Lamarque, D. Marsh, M. Mills, A. K. Smith, S. Tilmes, F. Vitt, H. Morrison, P. Cameron-Smith, W. D. Collins, M. J. Iacono, R. C. Easter, S. J. Ghan, X. Liu, P. J. Rasch, and M. A. Taylor. Description of the NCAR Community Atmosphere Model (CAM 5.0). NCAR Technical Note NCAR/TN-486+STR, National Center for Atmospheric Research, Boulder, Colorado, June 2010.

NOAA. (Experimental) NOAA GraphCast Global Forecast System (GFS). URL https://registry.opendata.aws/noaa-nws-graphcastgfs-pds.

NVIDIA. Earth2MIP, June 2025a. URL https://github.com/NVIDIA/earth2mip.

NVIDIA. Earth2Studio, July 2025b. URL https://github.com/NVIDIA/earth2studio.

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. *J. Adv. Model. Earth Syst.*, 2018. doi: 10.1029/2018MS001351.

Karl Otness, Laure Zanna, and Joan Bruna. Data-driven multiscale modeling of subgrid parameterizations in climate models, March 2023. URL http://arxiv.org/abs/2303.17496.

Jordan Ott, Mike Pritchard, Natalie Best, Erik Linstead, Milan Curcic, and Pierre Baldi. A Fortran-Keras Deep Learning Bridge for Scientific Computing. *Scientific Programming*, 2020(1):8888811, 2020. doi: 10.1155/2020/8888811.

Sam Partee, Matthew Ellis, Alessandro Rigazzi, Andrew E. Shao, Scott Bachman, Gustavo Marques, and Benjamin Robbins. Using Machine Learning at scale in numerical simulations with SmartSim: An application to ocean climate modeling. *Journal of Computational Science*, 62:101707, July 2022. doi: 10.1016/j.jocs.2022.101707.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Anders Persson. Early operational Numerical Weather Prediction outside the USA: an historical Introduction. Part 1: Internationalism and engineering NWP in Sweden, 1952–69. *Meteorological Applications*, 12(2):135–159, June 2005. doi: 10.1017/S1350482705001593.

Grant W. Petty. *A first course in atmospheric radiation*. Madison, Wis: Sundog Pub, 2nd ed. edition, 2006.

Norman A. Phillips. The general circulation of the atmosphere: A numerical experiment. *Quarterly Journal of the Royal Meteorological Society*, 82(352):123–164, 1956. doi: 10.1002/qj.49708235202.

George W. Platzman. A Retrospective View of Richardson's Book on Weather Prediction. *Bulletin of the American Meteorological Society*, 48(8):514–551, August 1967. doi: 10.1175/1520-0477-48.8.514.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, January 2025. doi: 10.1038/s41586-024-08252-9.

Di Qi and Andrew J. Majda. Using machine learning to predict extreme events in complex systems. *Proceedings of the National Academy of Sciences*, 117(1):52–59, January 2020. doi: 10.1073/pnas.1917285117.

Elias Rabel. ylikx/forpy, March 2025. URL https://github.com/ylikx/forpy.

D. A. Randall, R. A. Wood, S. Bony, R. Colman, T. Fichefet, J. Fyfe, V. Kattsov, A. Pitman, J. Shukla, J. Srinivasan, R. J. Stouffer, A. Sumi, and K. E. Taylor. Climate Models and Their Evaluation. In S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis amd K. B. Averyt, M. Tignor, and H. L. Miller, editors, *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 589–662. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007.

S. Rasp. Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and Lorenz 96 case study (v1.0). *Geoscientific Model Development*, 13(5):2185–2196, 2020. doi: 10.5194/gmd-13-2185-2020.

Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. Deep learning to represent sub-grid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39): 9684–9689, 2018. doi: 10.1073/pnas.1810286115.

Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11), 2020. doi: 10.1029/2020MS002203.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A Benchmark for the Next Generation of Data-Driven Global Weather Models. *Journal of Advances in Modeling Earth Systems*, 16(6), 2024. doi: 10.1029/2023MS004019.

K. A. Reed and C. Jablonowski. An analytic vortex initialization technique for idealized tropical cyclone studies in AGCMs. *Mon. Wea. Rev.*, 139:689–710, 2011.

K. A. Reed and C. Jablonowski. Idealized tropical cyclone simulations of intermediate complexity: A test case for AGCMs. *J. Adv. Model. Earth Syst.*, 4, 2012.

Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566:196 – 204, 2019. doi: 10.1038/s41586-019-0912-1.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24574-4_28.

Andrew Ross, Ziwei Li, Pavel Perezhogin, Carlos Fernandez-Granda, and Laure Zanna. Benchmarking of Machine Learning Ocean Subgrid Parameterizations in an Idealized Model. *Journal of Advances in Modeling Earth Systems*, 15(1), 2023. doi: 10.1029/2022MS003258.

John S. Schreck, Yingkai Sha, William Chapman, Dhamma Kimpara, Judith Berner, Seth McGinnis, Arnold Kazadi, Negin Sobhani, Ben Kirk, Charlie Becker, Gabrielle Gantos, and David John Gagne II. Community Research Earth Digital Intelligence Twin: a scalable framework for AI-driven Earth System Modeling. *npj Climate and Atmospheric Science*, 8(1):239, June 2025. doi: 10.1038/s41612-025-01125-6.

Gabriel Stachura, Zbigniew Ustrnul, Piotr Sekula, Bogdan Bochenek, Marcin Kolonko, and Malgorzata Szczech-Gajewska. Machine learning based post-processing of model-derived near-surface air temperature – A multimodel approach. *Quarterly Journal of the Royal Meteorological Society*, 150(759):618–631, 2024. doi: 10.1002/qj.4613.

B. Stevens and S. Bony. What Are Climate Models Missing? *Science*, 340:1053–1054, 2013.

Diana R. Thatcher and Christiane Jablonowski. A moist aquaplanet variant of the Held-Suarez test for atmospheric model dynamical cores. *Geoscientific Model Development*, 9 (4):1263–1292, 2016. doi: 10.5194/gmd-9-1263-2016.

Peter Ukkonen. Exploring Pathways to More Accurate Machine Learning Emulation of Atmospheric Radiative Transfer. *J. Adv. Model. Earth Syst.*, 14(4):1–19, 2022. doi: 10.1029/2021ms002875.

W. M. Washington and C. L. Parkinson. *An introduction to three-dimensional climate modeling*. University Science Books, second edition, 2005.

D. Watson-Parris, Y. Rao, D. Olivié, Ø. Seland, P. Nowack, G. Camps-Valls, P. Stier, S. Bouabid, M. Dewey, E. Fons, J. Gonzalez, P. Harder, K. Jeggle, J. Lenhardt, P. Manshausen, M. Novitasari, L. Ricard, and C. Roesch. ClimateBench v1.0: A Benchmark for Data-Driven Climate Projections. *Journal of Advances in Modeling Earth Systems*, 14 (10), 2022. doi: 10.1029/2021MS002954.

Oliver Watt-Meyer, Noah D. Brenowitz, Spencer K. Clark, Brian Henn, Anna Kwa, Jeremy McGibbon, W. Andre Perkins, and Christopher S. Bretherton. Correcting Weather and Climate Models by Machine Learning Nudged Historical Simulations. *Geophys. Res. Lett.*, 48(15):1–10, 2021. doi: 10.1029/2021GL092555.

D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber. A Standard Test Set for Numerical Approximations to the Shallow Water Equations in Spherical Geometry. *J. Comput. Phys.*, 102:211–224, 1992.

D. L. Williamson, M. Blackburn, B. J. Hoskins, K. Nakajima, W. Ohfuchi, Y. O. Takahashi, Y. Y. Hayashi, H. N. M. Ishiwatari, J. McGregor, H. Borth, V. Wirth, H. Frank, P. Bechtold, N. P. Wedi, H. Tomita, M. Satoh, M. Zhao, I. M. Held, M. J. Suarez, M.-I. Lee, M. Watanabe, M. Kimoto, Y. Liu, Z. Wang, A. Molod, K. Rajendran, A. Kitoh, and R. Stratton. The APE Atlas. NCAR Technical Note NCAR/TN-484+ STR, National Center for Atmospheric Research, Boulder, Colorado, 2012.

Ruyi Yang, Jingyu Hu, Zihao Li, Jianli Mu, Tingzhao Yu, Jiangjiang Xia, Xuhong Li, Aritra Dasgupta, and Haoyi Xiong. Interpretable machine learning for weather and climate prediction: A review. *Atmospheric Environment*, 338:120797, December 2024. doi: 10.1016/j.atmosenv.2024.120797.

M. Soner Yorgun and Richard B. Rood. A Decision Tree Algorithm for Investigation of Model Biases Related to Dynamical Cores and Physical Parameterizations. *J. Adv. Model. Earth Syst.*, 8:1769–1785, 2016. doi: 10.1002/2016MS000657.

Janni Yuval and Paul A. O'Gorman. Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, 11(1): 1–10, 2020. doi: 10.1038/s41467-020-17142-3.

Janni Yuval and Paul A. O'Gorman. Neural-Network Parameterization of Subgrid Momentum Transport in the Atmosphere. *Journal of Advances in Modeling Earth Systems*, 15 (4), 2023. doi: 10.1029/2023MS003606.

Janni Yuval, Paul A. O'Gorman, and Chris N. Hill. Use of Neural Networks for Stable, Accurate and Physically Consistent Parameterization of Subgrid Atmospheric Processes With Good Performance at Reduced Precision. *Geophys. Res. Lett.*, 48(6), 2021. doi: 10.1029/2020GL091363.

Cheng Zhang, Pavel Perezhogin, Cem Gultekin, Alistair Adcroft, Carlos Fernandez-Granda, and Laure Zanna. Implementation and Evaluation of a Machine Learned Mesoscale Eddy Parameterization Into a Numerical Ocean Circulation Model. *Journal of Advances in Modeling Earth Systems*, 15(10), 2023. doi: 10.1029/2023MS003697.

Tao Zhang, Cyril Morcrette, Meng Zhang, Wuyin Lin, Shaocheng Xie, Ye Liu, Kwinten Van Weverberg, and Joana Rodrigues. A Fortran–Python interface for integrating machine learning parameterization into earth system models. *Geoscientific Model Development*, 18 (6):1917–1928, March 2025. doi: 10.5194/gmd-18-1917-2025.

Han Zhao, Yao-Hung Hubert Tsai, Ruslan Salakhutdinov, and Geoffrey J. Gordon. Learning neural networks with adaptive regularization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 1022, pages 11393–11404. Curran Associates Inc., Red Hook, NY, USA, December 2019.