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

Garrett C. Limon[1], **Christiane Jablonowski**[1]

[1]Department of Climate and Space Sciences and Engineering, University of Michigan, Ann Arbor, MI

Key Points:

- Random forests skillfully emulate simple physics schemes within the Community Atmosphere Model in an offline state.
- Hierarchical approach shows both qualitative and quantitative decreases in skill of random forests as complexity increases.
- In the case of 2-dimensional precipitation fields, random forest skill is in-line with baseline neural network performance.

Corresponding author: Garrett C. Limon, `glimon@umich.edu`

Abstract

Machine learning approaches, such as random forests, have been used to effectively emulate various aspects of climate and weather models in recent years. The limitations to these approaches are not yet known, particularly with regards to varying complexity of the underlying physical parameterization scheme within the climate model. Utilizing a hierarchy of model configurations, we explore the limits of random forest emulator skill using simplified model frameworks within NCAR's Community Atmosphere Model, version 6 (CAM6). These include a dry CAM6 configuration, a moist extension of the dry model, and an extension of the moist case that includes an additional convection scheme. Each model configuration is run with identical resolution and over the same time period. With unique random forests being optimized for each tendency or precipitation rate across the hierarchy, we create a variety of "best case" emulators. The random forest emulators are then evaluated against the CAM6 output as well as a baseline neural network emulator for completeness. All emulators show significant skill when compared to the "truth" (CAM6), often in line with or exceeding similar approaches within the literature. In addition, as the CAM6 complexity is increased, the random forest skill noticeably decreases, regardless of the extensive tuning and training process each random forest goes through. This indicates a limit on the feasibility of random forests to act as physics emulators in climate models and encourages further exploration in order to identify ideal uses in the context of state-of-the-art climate model configurations.

Plain Language Summary

Machine learning has become an intriguing technique for replacing complicated aspects of climate and weather models, processes such as cloud interactions and rain are examples of this. However, the limitations of various machine learning techniques are not yet fully understood. We explore these limits, focusing on a specific machine learning method and utilizing simplified climate modeling frameworks. The machine learning models are then carefully analyzed against the original climate model results and results from a standard baseline machine learning approach. All of our machine learned models show impressive skill at recreating the original results. However, that skill is shown to noticeably decrease as the complexity of the climate model framework is increased. While this may be expected, it is useful for understanding limits on the feasibility of certain machine learning techniques to be used within state-of-the-art climate models. Further investigation is needed to understand the viability and best use-cases of these methods being adopted into simulating of the Earth system.

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 & 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 & 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 & Morrison, 2015; Gettelman et al., 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., 2019; Yuval et al., 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., 2019; 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 & 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 Methods

2.1 CAM6 Configurations

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

$$\frac{\partial \vec{v}_h}{\partial t} = -k_v(p)\,\vec{v}_h. \tag{1}$$

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

$$\left(\frac{\partial T}{\partial t}\right)_{\mathrm{HS}} = -k_T(\phi, p)\,[T - T_{\mathrm{eq}}(\phi, p)]. \tag{2}$$

Here, $\partial/\partial t$ represents a sub-grid physics tendency (forcing) of a variable over a physics time step, p symbolizes the pressure, ϕ denotes the latitude, $\vec{v}_h$ is the horizontal velocity vector, T stands for the temperature, T_{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 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.

2.1.2 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

$$\left(\frac{\partial T}{\partial t}\right)_{\mathrm{TJ}} = -k_T(\phi, p)\,[T - \tilde{T}_{\mathrm{eq}}(\phi, p)] + \frac{L}{c_p}C + \frac{C_H|\vec{v}_a|(T_s - T_a)}{z_a} + \text{PBL Diffusion} \tag{3}$$

where $\tilde{T}_{\mathrm{eq}}$ 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

$$\left(\frac{\partial q}{\partial t}\right)_{\mathrm{TJ}} = -C + \frac{C_E|\vec{v}_a|(q_{\mathrm{sat},s} - q_a)}{z_a} + \text{PBL diffusion} \tag{4}$$

Here, q refers to the specific humidity, C_E is the bulk transfer coefficient for water vapor, $q_{\mathrm{sat},s}$ 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

$$P_{\mathrm{ls}} = \frac{1}{\rho_{\mathrm{water}} g} \int_{p_{top}}^{p_s} C dp \tag{5}$$

where ρ_{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.

2.1.3 Convection Scheme

The final step in our CAM6 model hierarchy couples the BM convection scheme to the TJ setup (Betts, 1986; Betts & Miller, 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

$$\left(\frac{\partial T}{\partial t}\right)_{\mathrm{BM}} = -\frac{T - T_{\mathrm{ref}}}{\tau} + \left(\frac{\partial T}{\partial t}\right)_{\mathrm{TJ}} \tag{6}$$

$$\left(\frac{\partial q}{\partial t}\right)_{\mathrm{BM}} = -\frac{q - q_{\mathrm{ref}}}{\tau} + \left(\frac{\partial q}{\partial t}\right)_{\mathrm{TJ}} \tag{7}$$

where τ is the convective relaxation time and T_{ref} and q_{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 ($\mathrm{RH}_{\mathrm{BM}}$) and τ, 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 $\mathrm{RH}_{\mathrm{BM}}$ and τ 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 $\mathrm{RH}_{\mathrm{BM}} = 0.7$.

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

$$\hat{g}(\vec{X}) \approx f(\vec{X}) \tag{8}$$

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 & 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., 2019).

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 & 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.

2.3 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_{time}, N_{lev} N_{lat}, and N_{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 $\mathrm{N}_{\text{lev}} = 1$. 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

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

where $N_{\mathrm{output\,fields}} = 1$ 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 et al. (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 $N_{\mathrm{labels}} = N_{\mathrm{lev}} = 1$. This allows us to use closer to $N_{\mathrm{samples}} \approx 20$ 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 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 S1 to S9 in the Supporting Information.

3 Results & Discussion

3.1 Snapshots & Mean Fields

Figures 1 and 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 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 1 and 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 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.

Figures 3 and 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 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 3 and 4. The NN predictions in both moist tendencies (Figures 3e & 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 3d and 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 4b,d do not show the same discernible pattern.

Figure 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 1 and 2 and, even more pronounced, in the climatologies in Figures 3 and 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 3c and 3f. These regions correspond to the large-scale midlatitudinal precipitation peaks in Figures 5a 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 5a and 5c. These precipitation patterns are correlated with the intense tropical and subtropical heating peaks in Figures 3c,f and the negative moisture tendencies in Figures 4a,d.

The minor differences between the ML predictions and the CAM6 output in the snapshot figures (Figures 1,2) 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 3, 4, and 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 3d for the moist case is significantly more pronounced for the convection case in Figure 3g. This effect is not as apparent in the RF specific humidity error patterns in Figures 4b and 4e.

In Figure 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 5 align closely with the areas of 'drying' in Figure 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.

3.2 Point-wise Comparison

Next, we show one-to-one scatter plots of the results from CAM and the RF emulator in Figures 6 and 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 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 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 7 shows the scatter plots for each precipitation rate, oriented in the same configuration as Figure 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 8, which is identical to Figure 6c but with the NN results (y-axis) shown on the scatter plot rather 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 8 is more significant, as we see more frequent anomalies in these model levels near the equator as shown in Figure 4.

We also include a panel of histograms in Figures 9 and 10 corresponding to the same case orientation as Figures 6 and 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 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 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 7c shows the most visual spread among the precipitation rate scatter plots. Along these same lines, both tendencies in Figures 6 and 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 6b,d and 7c. In addition, the specific humidity histogram in Figure 9d clearly indicates that the magnitude of the outliers increases in the convection case in comparison to the moist case (9c). The distribution gets wider in the convection case. However, all of the histograms in Figures 9 and 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 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 and is a significantly rare, albeit expected, occurrence.

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

$$R^2(:,:) = 1 - \frac{\sum_t \sum_\lambda [\mathrm{CAM}(t,:,:,\lambda) - \mathrm{ML}(t,:,:,\lambda)]^2}{\sum_t \sum_\lambda [\mathrm{CAM}(t,:,:,\lambda) - \overline{\mathrm{CAM}}(:,:)]^2} \tag{9}$$

where λ 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.

We display a panel of R^2 plots for all of our tendencies in Figures 11 and 12 and precipitation rates in Figure 13. All of the predicted fields and tendencies show large 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 et al. (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 11, 12 and 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 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 11c there are broader regions of $\mathrm{R}^2 \leq 0.5$ in the upper atmosphere than in Figure 11b. Similarly, two pockets of $\mathrm{R}^2 \approx 0.3$ form around the tropics in Figure 11e, which were not nearly as pronounced in Figure 11d with $\mathrm{R}^2 > 0.7$ in these regions. This region is associated with tropical convection as shown in Figure 5c and also is present in the dips in R^2 for convective precipitation (blue lines) in Figure 13. For all precipitation 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 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 13 are impressive when compared with R^2 values from the physics tendency results (Figures 11 & 12). This is likely due both to the fact that these are surface fields, as well as their having less complex mathematical representations.

Figure 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 11a,c,d,e. In these regions the variance in the denominator and the sum of squares in the numerator (see Equation 9) are both functionally zero. However, they are still seen as floating point numbers of extremely small order and Equation 9 can lead to various misleading results such as

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

or

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

For the dry case in Figure 11a, this occurs in the tropics in the mid-atmosphere. Similarly, this occurs in the upper atmosphere for the moisture tendencies in Figures 11d and 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 3 and 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.

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 2.1. 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 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.

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 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 15, as expected. It is also worth noting that the rate at which the skill 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 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 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.

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 GPUs, 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 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.

Open Research Section

Software - All machine learning related scripts are available at Limon (2023). Figures were generated using both Matplolib (Hunter, 2007) and *The NCAR Command Language* (2019), while various maching learning-related libraries were used, including Scikit-Learn, Xarray, and Keras (Pedregosa et al., 2011; Hoyer & Hamman, 2017; Chollet, 2017).

Data - The CAM6 output data used for all three cases of machine learning in the study were generated in-house and are available at Limon (2022).

Acknowledgments

This work was made possible by the National Science Foundation's Graduate Research Fellowship Program and the NOAA grants NA17OAR4320152(127) and NA22OAR4320150. We would like to acknowledge high-performance computing support from NCAR and their

Computational and Information Systems Laboratory in our use of the Cheyenne and Casper systems. Lastly, we would like to thank the reviewers and editor who offered their time, along with thoughtful and insightful feedback to our original submission, which assisted in significantly improving this manuscript.

References

Baldi, P. (2021). *Deep Learning in Science: Theory, Algorithms, and Applications.* Cambridge, England: Cambridge University Press. doi: 10.1017/9781108955652

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

Betts, A. K., & Miller, M. J. (1986). 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.

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing analytic constraints in neural-networks emulating physical systems. *Phys. Rev. Lett.*, *126*, 098302. doi: 10.1103/PhysRevLett.126.098302

Beucler, T., Rasp, S., Pritchard, M., & Gentine, P. (2019). Achieving Conservation of Energy in Neural Network Emulators for Climate Modeling. *arXiv*. Retrieved from `http://arxiv.org/abs/1906.06622` (arXiv:1906.06622v1)

Bogenschutz, P. A., Gettelman, A., Morrison, H., Larson, V. E., Craig, C., & Schanen, D. P. (2013, December). Higher-order turbulence closure and its impact on climate simulations in the Community Atmosphere Model. *J. Climate*, *26*, 9655–9676.

Boukabara, S. A., Krasnopolsky, V., Penny, S. G., Stewart, J. Q., McGovern, A., Hall, D., … Hoffman, R. N. (2021). Outlook for exploiting artificial intelligence in the Earth and environmental sciences. *Bulletin of the American Meteorological Society*, *102*(5), E1016–E1023. doi: 10.1175/BAMS-D-20-0031.1

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, *24*(2), 123–140. Retrieved from `https://doi.org/10.1007/BF00058655` doi: 10.1007/BF00058655

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

Bretherton, C. S., Henn, B., Kwa, A., Brenowitz, N. D., Watt-Meyer, O., McGibbon, J., … Harris, L. (2022). Correcting Coarse-Grid Weather and Climate Models by Machine Learning From Global Storm-Resolving Simulations. *J. Adv. Model. Earth Syst.*, *14*(2). doi: 10.1029/2021MS002794

Chantry, M., Christensen, H., Düben, P., & Palmer, T. (2021). Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft AI. *Phil. Trans. R. Soc. A*, *379*(2194), 20200083. doi: 10.1098/rsta.2020.0083

Chapman, W. E., Subramanian, A. C., Delle Monache, L., Xie, S. P., & Ralph, F. M. (2019). Improving atmospheric river forecasts with machine learning. *Geophys. Res. Lett.*, *46*(17–18), 10627–10635. doi: 10.1029/2019GL083662

Chollet, F. (2017). *Deep Learning with Python* (1st ed.). USA: Manning Publications Co., 384 pages.

Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., … Strand, W. G. (2020). The Community Earth System Model Version 2 (CESM2). *J. Adv. Model. Earth Syst.*, *12*(2), e2019MS001916. doi: 10.1029/2019MS001916

Foster, D., Gagne, D. J., & Whitt, D. B. (2021). 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. doi:

10.1029/2021MS002474

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

Gagne, D. J., McGovern, A., Haupt, S. E., Sobash, R. A., Williams, J. K., & Xue, M. (2017). Storm-Based Probabilistic Hail Forecasting with Machine Learning Applied to Convection-Allowing Ensembles. *Weather and Forecasting*, *32*(5), 1819–1840. doi: 10.1175/WAF-D-17-0010.1

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.*, *45*(11), 5742–5751. doi: 10.1029/2018GL078202

Gettelman, A., Gagne, D. J., Chen, C.-C., Christensen, M. W., Lebo, Z. J., Morrison, H., & Gantos, G. (2021). Machine Learning the Warm Rain Process. *J. Adv. Model. Earth Syst.*, *13*(2), e2020MS002268. doi: https://doi.org/10.1029/2020MS002268

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

Gettelman, A., Morrison, H., Santos, S., Bogenschutz, P., & Caldwell, P. (2015). Advanced Two-Moment Bulk Microphysics for Global Models. Part II: Global model solutions and Aerosol-Cloud Interactions. *J. Climate*, *28*(3), 1288–1307.

Han, Y., Zhang, G. J., Huang, X., & Wang, Y. (2020). A moist physics parameterization based on deep learning. *J. Adv. Model. Earth Syst.*, *12*(9), e2020MS002076. doi: 10.1029/2020MS002076

Held, I. M. (2005, November). The gap between simulation and understanding in climate modeling. *Bull. Amer.. Meteor, Soc,*, *86*, 1609–1614.

Held, I. M., & Suarez, M. J. (1994, October). A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models. *Bull. Amer. Meteor. Soc.*, *75*(10), 1825–1830.

Hertel, L., Collado, J., Sadowski, P., Ott, J., & Baldi, P. (2020). Sherpa: Robust Hyperparameter Optimization for Machine Learning. *SoftwareX*, *24*. Retrieved from `10.1016/j.softx.2020.100591`

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.-C., Balaji, V., Duan, Q., ... Williamson, D. (2017). The art and science of climate model tuning. *Bull. Ameri. Meteor. Soc.*, *98*(3), 589–602. doi: 10.1175/BAMS-D-15-00135.1

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

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

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

Limon, G. C. (2022). *Simple Physics CAM6 Dataset for Training Machine Learning Algorithms* (Tech. Rep.). [Dataset] University of Michigan - Deep Blue Data. Retrieved from `https://doi.org/10.7302/r6v3-s064`

Limon, G. C. (2023). *Simple Physics CAM6 Codebase for Training Machine Learning Algorithms* (Tech. Rep.). [Software] University of Michigan - Deep Blue Data. Retrieved from `https://doi.org/10.7302/kxrz-9k87`

Lin, S.-J. (2004, October). A “vertically Lagrangian” finite-volume dynamical core for global models. *Mon. Wea. Rev.*, *132*, 2293–2307.

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

The NCAR Command Language (Tech. Rep.). (2019). [Software] Boulder, Colorado: UCAR/NCAR/CISL/TDD. doi: dx.doi.org/10.5065/D6WD3XH5

O'Gorman, P. A., & Dwyer, J. G. (2018). Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events. *J. Adv. Model. Earth Syst.*, *10*(10), 2548–2563. doi: 10.1029/2018MS001351

Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

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

Reed, K. A., & Jablonowski, C. (2012). Idealized tropical cyclone simulations of intermediate complexity: A test case for AGCMs. *J. Adv. Model. Earth Syst.*, *4*, M04001.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, *566*, 196 – 204. doi: 10.1038/s41586-019-0912-1

Stevens, B., & Bony, S. (2013). What Are Climate Models Missing? *Science*, *340*(6136), 1053–1054. doi: 10.1126/science.1237554

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

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

Watt-Meyer, O., Brenowitz, N. D., Clark, S. K., Henn, B., Kwa, A., McGibbon, J., ... Bretherton, C. S. (2021). Correcting Weather and Climate Models by Machine Learning Nudged Historical Simulations. *Geophys. Res. Lett.*, *48*(15), 1–10. doi: 10.1029/2021GL092555

Williamson, D. L., Blackburn, M., Hoskins, B. J., Nakajima, K., Ohfuchi, W., Takahashi, Y. O., ... Stratton, R. (2012). *The APE Atlas* (NCAR Technical Note Nos. NCAR/TN-484+ STR). Boulder, Colorado: National Center for Atmospheric Research. doi: 10.5065/D6FF3QBR

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

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nat. Commun.*, *11*, 3295. doi: 10.1038/s41467-020-17142-3

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021). 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), 1–11. doi: 10.1029/2020gl091363

Figures

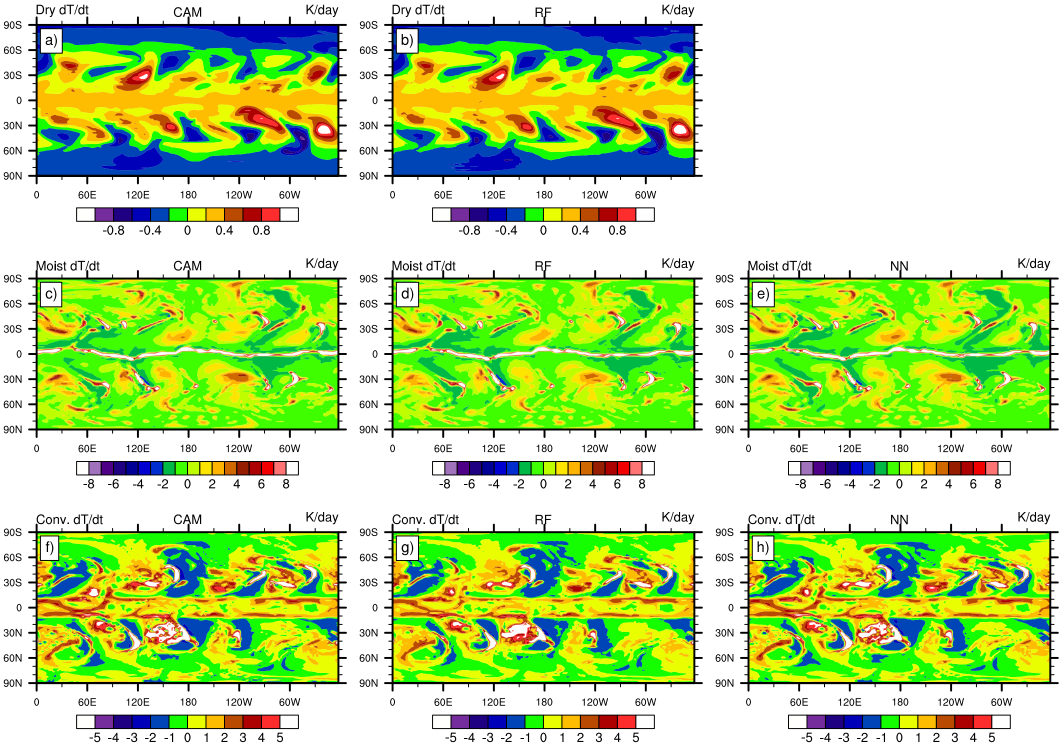

Figure 1. Snapshots of the predicted temperature tendencies near 850 hPa for the (top) dry, (middle) moist, and (bottom) convective cases: (left) CAM6 output, (middle column) RF predictions, (right) NN predictions. The magnitude of the extremes in (c), (d), and (e) is around 50 − 60 K/day and close to 20 K/day in (f), (g) and (h), but were left out in order to avoid over-saturating the contours.

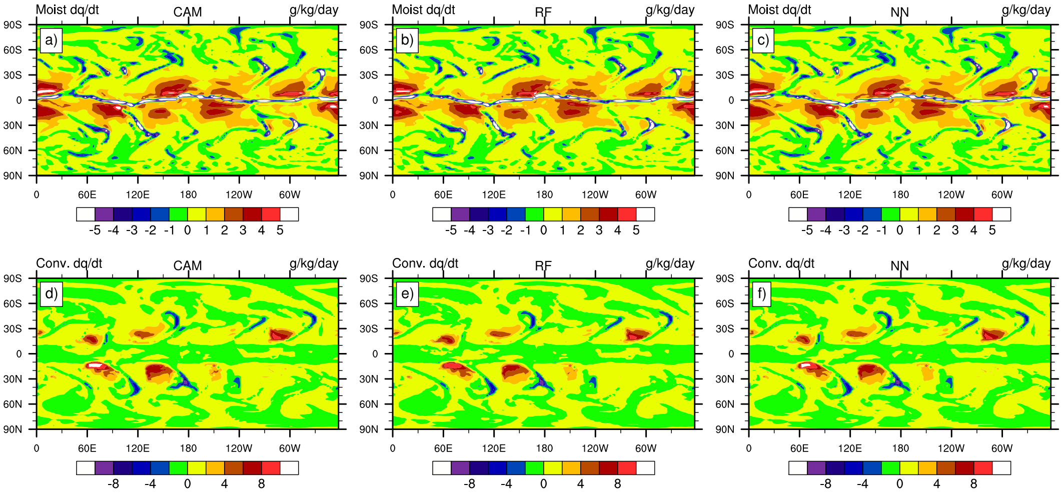

Figure 2. Snapshots of the predicted specific humidity tendencies near 850 hPa for the (top) moist and (bottom) convective cases: (left) CAM6 output, (middle column) RF predictions, and (right) NN predictions. The minima in (a), (b), and (c) are around −20 g/kg/day, but were left out in order to avoid over-saturating the contours.

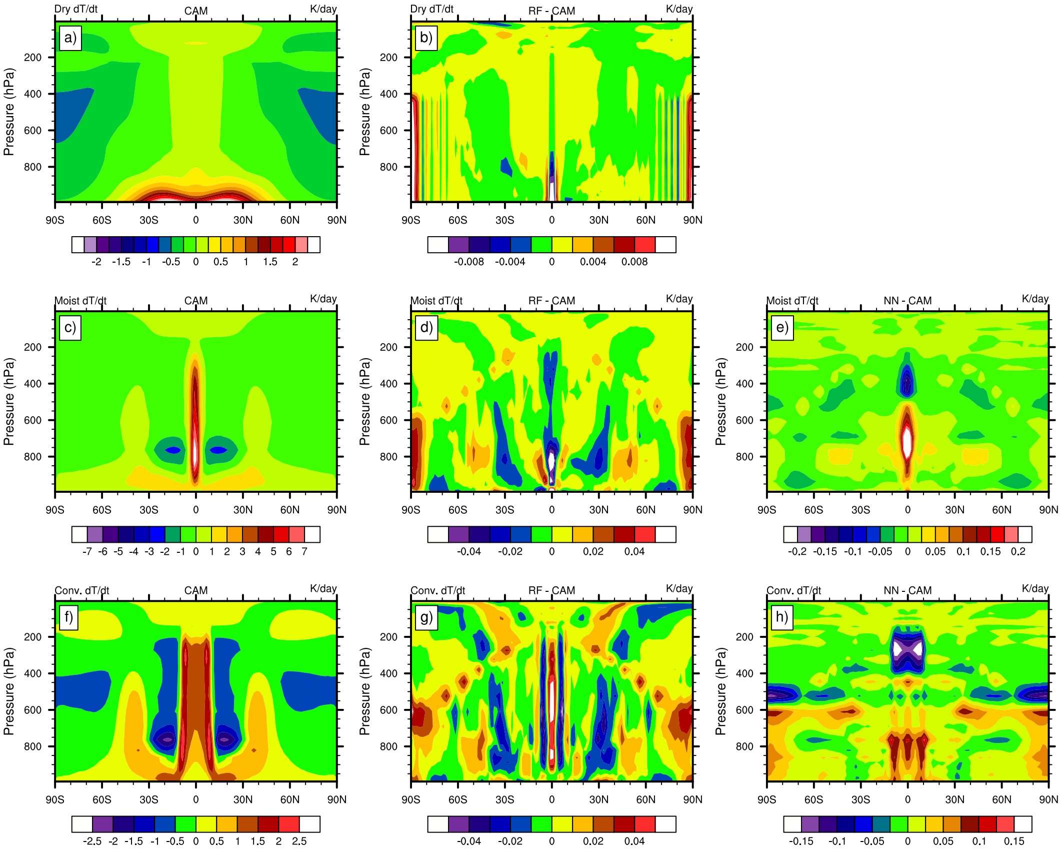

Figure 3. Zonal-mean time-mean temperature tendency output from CAM6 and the ML anomalies over the full testing data set. Ordered by dry (top), moist (middle), and convection (bottom) cases; left column is CAM6 output, middle column is RF difference, and right column is NN differences. The maxima in (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.

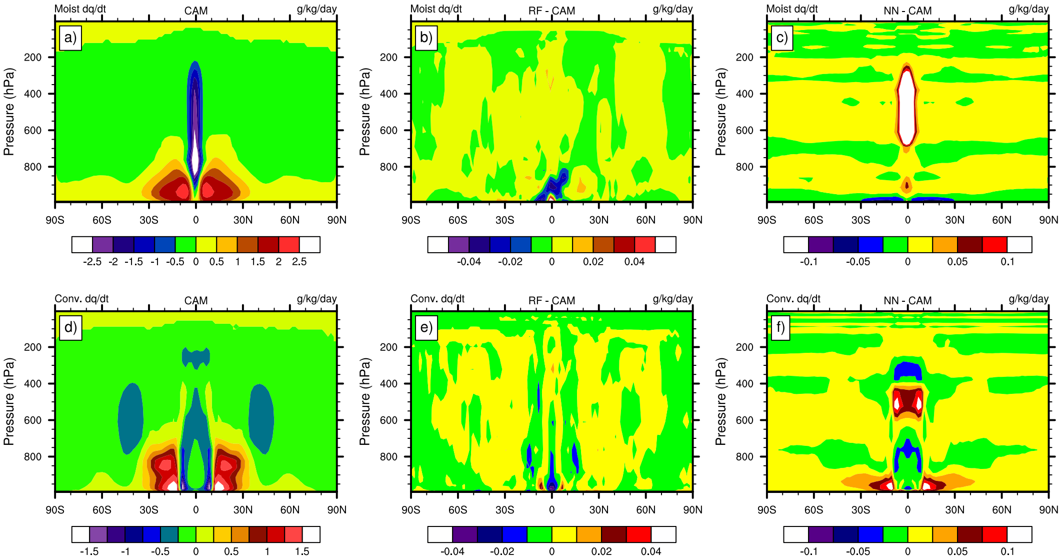

Figure 4. Zonal-mean time-mean moisture tendencies over the full testing data set for the (top) moist and (bottom) convective cases: (left) CAM6 output, (middle column) RF ML predictions, (right) their differences. The minimum in (a) is around −3.6 g/kg/day and the maximum in (c) is around 0.46 g/kg/day, but were left out in order to avoid over-saturating the contours.

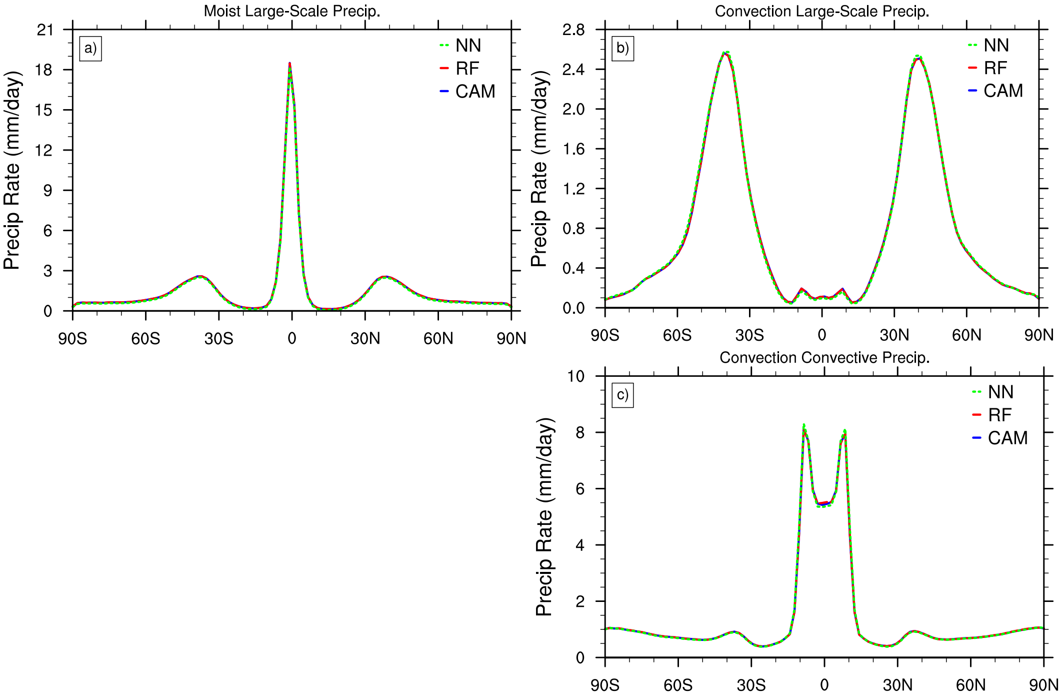

Figure 5. Zonal-mean time-mean precipitation rates of CAM6 (blue), RF prediction (red), and NN prediction (green) over the full testing data set for the (top) large-scale precipitation (Equation 5) and (bottom) convective precipitation; (left) moist case, (right) convective case.

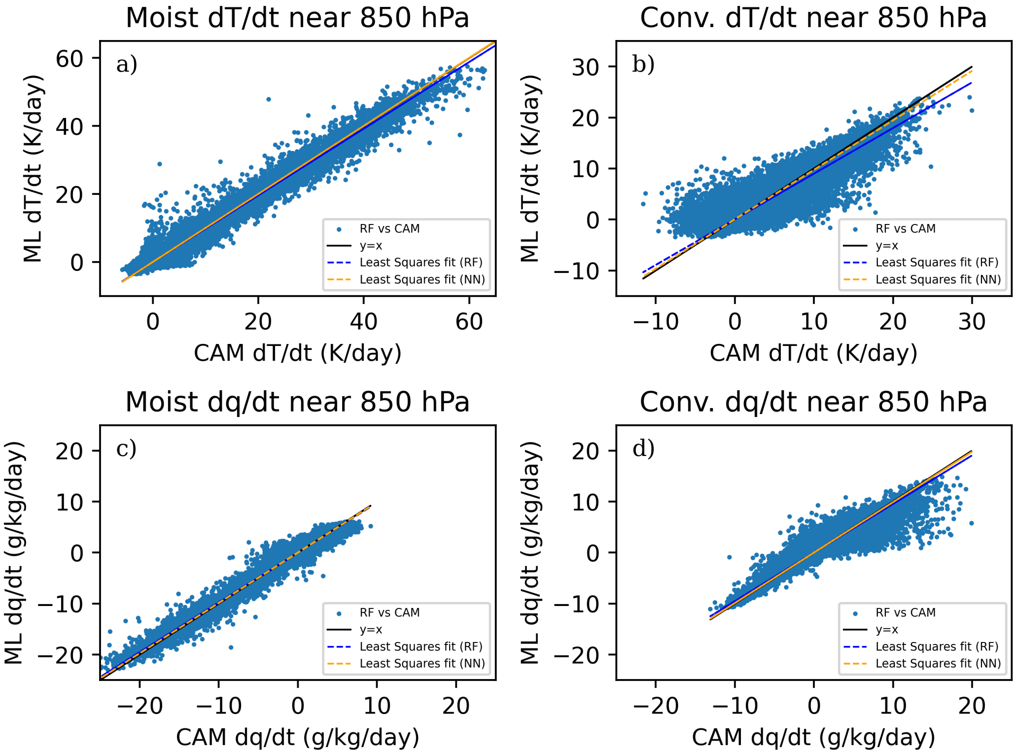

Figure 6. Scatter plots for RF predicted values (y-axis) against CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for (a) moist-case temperature tendency, (b) convection-case temperature tendency, (c) moist-case moisture tendency, and (d) convection-case moisture tendency.

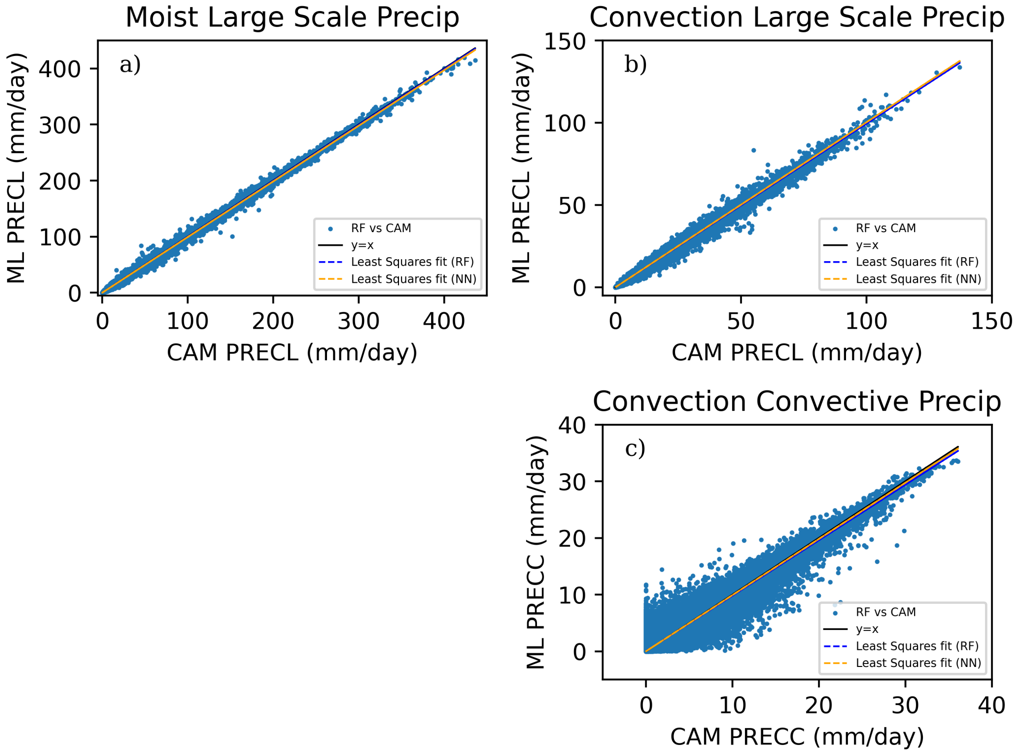

Figure 7. Scatter plots for RF predicted values (y-axis) against CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for the (a) moist-case large-scale precipitation rate, (b) convection-case large-scale precipitation rate, and (c) convection-case convective precipitation rate.

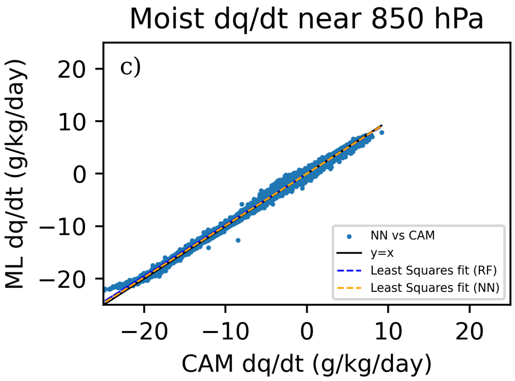

Figure 8. Scatter plot for NN predicted values (y-axis) against CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for moist-case moisture tendency

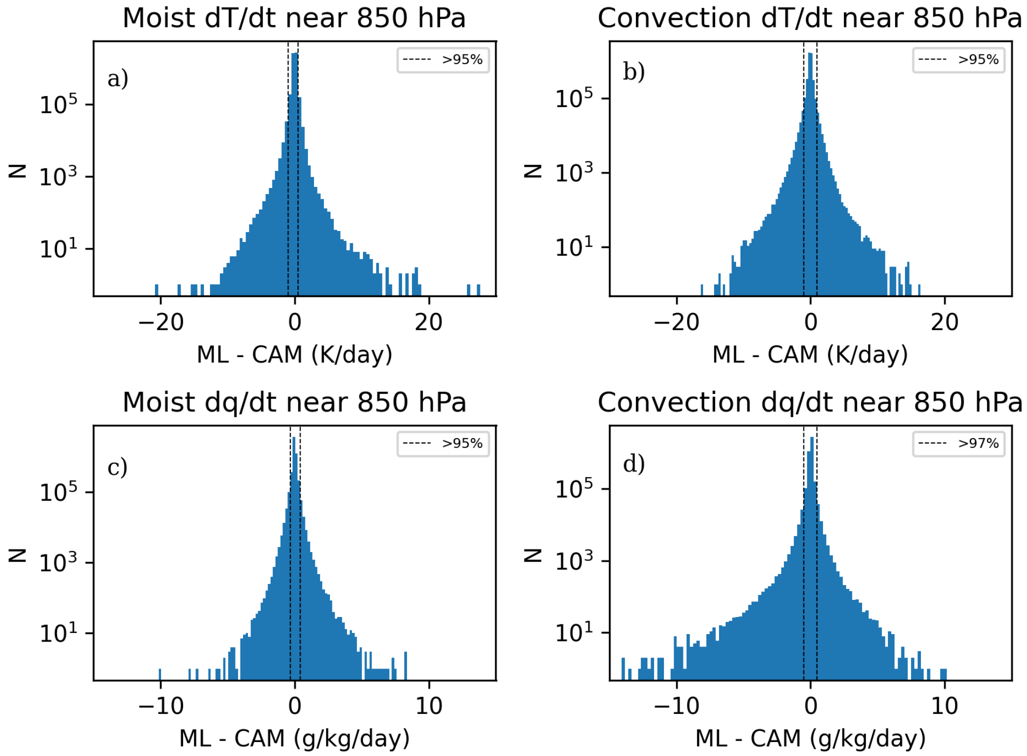

Figure 9. Histograms of the point-wise difference (RF - CAM6) for the temperature (top) and specific humidity (bottom) tendencies, corresponding to the scatter plots in Figure 6 on a log scale using 100 bins. Percentage of data contained within the black dashed lines are indicated in individual legends.

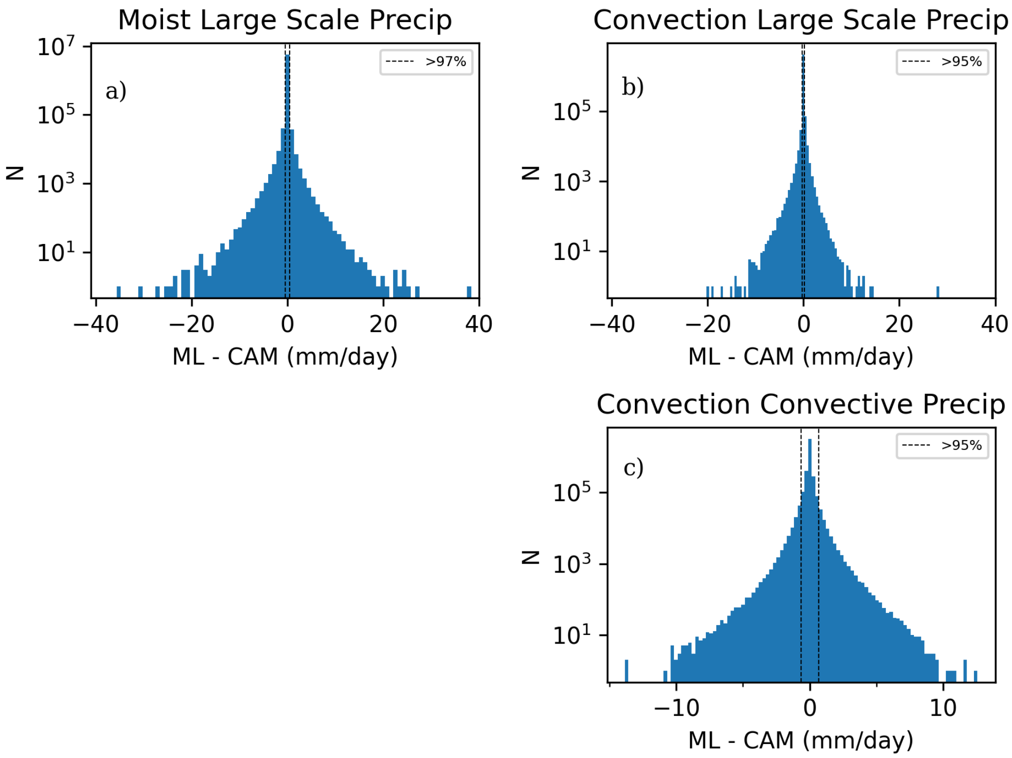

Figure 10. Histograms of the point-wise difference (RF - CAM6) for the precipitation rates corresponding to the scatter plots in Figure 7 on a log scale using 100 bins. Percentage of data contained within the black dashed lines are indicated in individual legends.

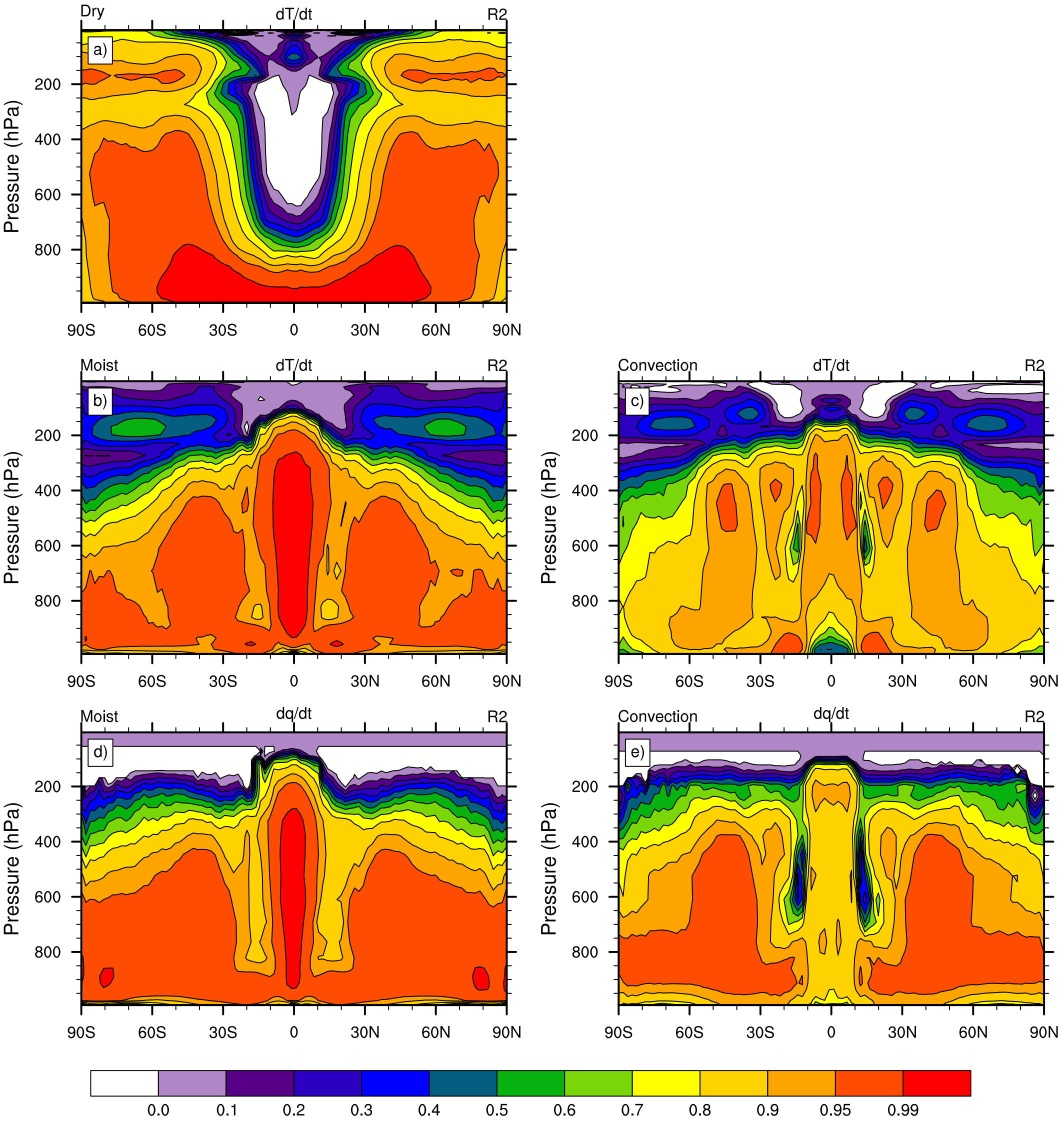

Figure 11. R^2 calculations over the zonal and temporal dimensions for RF emulators of (a) dry temperature tendency, (b) moist temperature tendency, (c) convection temperature tendency, (d) moist moisture tendency, and (e) convection moisture tendency via Equation 9.

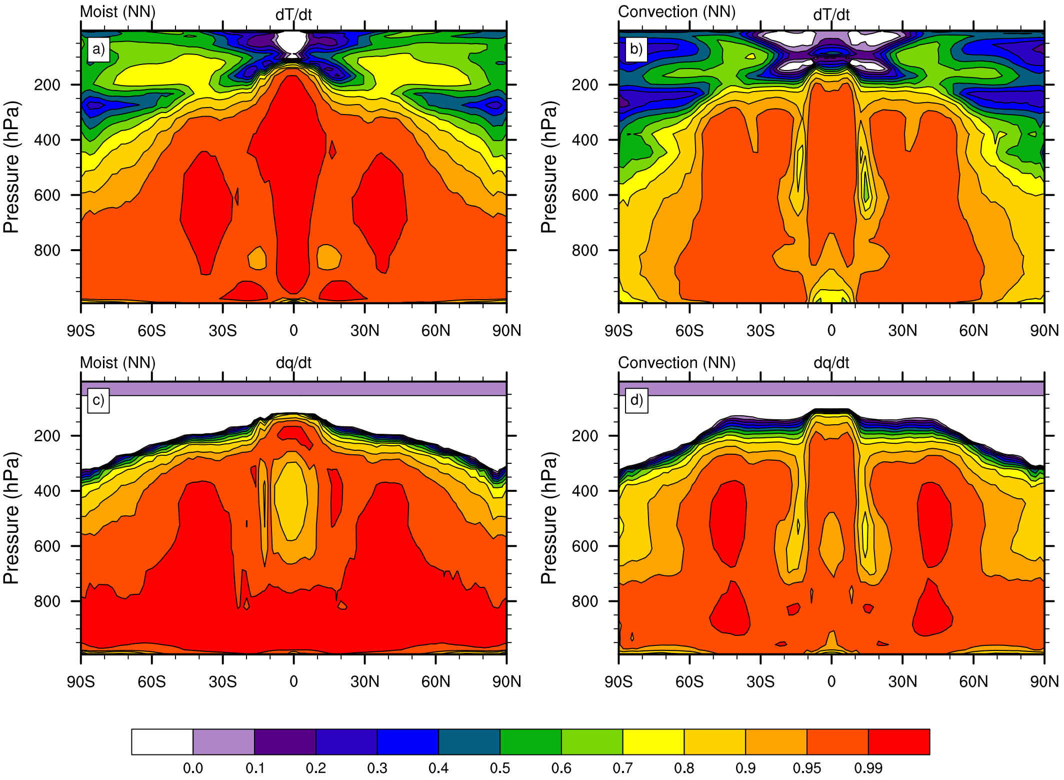

Figure 12. R^2 calculations over the zonal and temporal dimensions for NN emulators of (a) moist temperature tendency, (b) convection temperature tendency, (c) moist moisture tendency, and (d) convection moisture tendency via Equation 9.

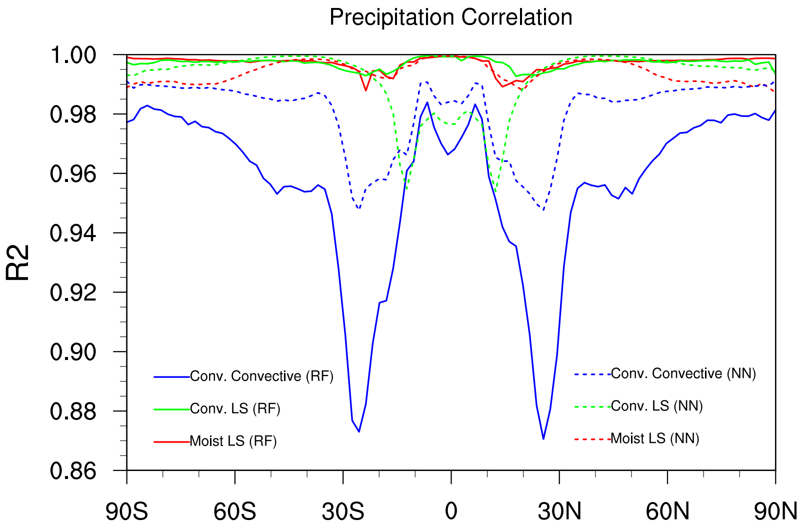

Figure 13. R^2 calculations over the zonal and temporal dimensions via Equation 9 for ML predictions of moist large-scale precipitation (red), convection large-scale precipitation (green), and convection convective precipitation (blue); NN results are dashed lines, RF results are solid.

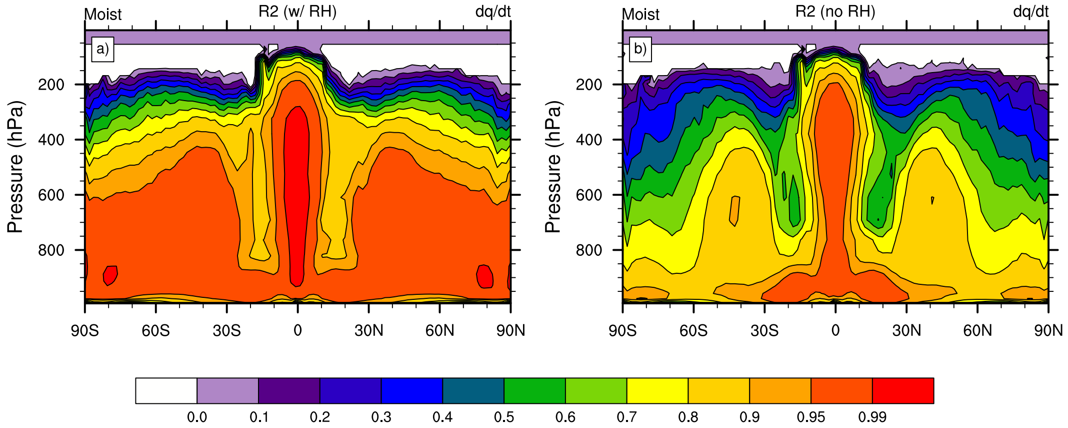

Figure 14. Comparison of R^2 plot - as defined in Figure 11 - (a) with and (b) without relative humidity as a feature for RF prediction of the moisture tendency for the moist case. Figure 14a reproduces Figure 11d.

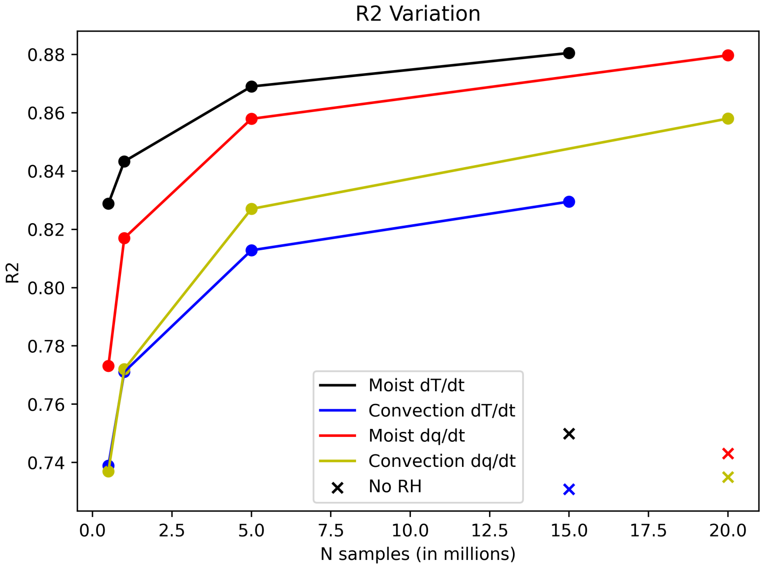

Figure 15. Globally-averaged R^2 value (y-axis) for RF prediction of the tendencies in the moist and convection cases as the number of data available for training is increased (lines), as well as when RH is removed as an input (crosses) using the maximum amount of training data. Note: to avoid saturation by large negative numbers (discussed in Section 3.3), these global R^2 values are calculated from the surface up to roughly 175 hPa.