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 NCAR's CAM6 atmosphere model
- Hierarchical approach probes the limitations of emulators as complexity increases

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 physical parameterization schemes being emulated within the climate model. Utilizing a hierarchy of model configurations, we explore the limitations of random forest emulator skill using simplified model frameworks within NCAR's Community Atmosphere Model, version 6 (CAM6). These include a dry model configuration, a moist extension of the dry model, and an extension of the moist case that includes an additional convection scheme. With unique random forests being optimized for each tendency or precipitation rate across the hierarchy, we create a variety of emulators. Each model configuration is run with identical resolution and over the same time period. The models are then evaluated against the CAM6 output. All models show significant skill across each random forest emulator, 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 that limit 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 known as parameterizations, which account for processes like cloud interactions and rain. However, the limitations of machine learning techniques are not yet fully understood. We explore these limits using a specific machine learning method and simplified climate model frameworks. The machine learning models are then carefully analyzed against the original climate model results. All of our machine learning 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 still indicates a limit on the feasibility of machine learning techniques to substitute for the complicated parameterizations within state-of-the-art climate models. Further investigation is needed to understand the viability of these methods being adopted into the simulation 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, which estimate subgrid-scale processes that the dynamical core does not resolve. The latter 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). Beginning with the work of Krasnopolsky and Fox-Rabinovitz (2006) applying neural networks 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) were able to develop ML emulators for radiative transfer processes, O'Gorman and Dwyer (2018) and Gentine et al. (2018) used ML to emulate moist convection processes, Gettelman et al. (2021) utilized neural networks 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 neural networks. 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 complexities, each with a well-defined increase in non-linearity associated with its mathematical expressions. The parameterization schemes begin with a dry model setup, referred to as HS hereon and described in Held and Suarez (1994). 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 be referred to throughout the papers as dry, moist, and convection, respectively. None of these physics schemes include topography or seasonal and diurnal cycles. We also keep our ML technique consistent, using random forests (Breiman, 1996). This allows for an investigation into the fundamental relationship between the degree of non-linearity within the parameterization scheme and the corresponding complexity of the random forest to effectively emulate the forcing.

In this work, we show that various physical forcing tendencies and the precipitation rate can be emulated by random forest models in an offline mode. We investigate how their skill depends on model complexity, as well as whether we can identify indications of limits on the feasibility of random forests for use in more complex model setups. 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 and the random forest 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 scheme is based on two 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),

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

and radiation is mimicked by a Newtonian temperature relaxation described by

$$\left(\frac{\partial T}{\partial t}\right)_{\text{HS}} = -\frac{1}{k_T(\phi, p)}[T - T_{\text{eq}}(\phi, p)] \tag{2}$$

Here, $\partial/\partial t$ represents a sub-grid physics tendency (forcing) of a variable over a physics time step, $\vec{v}_h$ is the horizontal velocity vector, T is the temperature, k_v and k_T are the dissipation coefficient and the relaxation coefficient, respectively, T_{eq} is a pre-defined equilibrium temperature profile, p is the pressure, and ϕ is the latitude. 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 mid-latitudes. 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 physics scheme is similarly forced by Rayleigh friction and Newtonian temperature relaxation, this time with a modified equilibrium temperature profile and additional forcing mechanisms. These include heating and cooling due to large-scale condensation, surface fluxes of latent and sensible heat, and a planetary boundary layer (PBL) mixing scheme via second order diffusion (Thatcher & Jablonowski, 2016), abbreviated as TJ16 later. The temperature forcing then takes the form

$$\left(\frac{\partial T}{\partial t}\right)_{\text{TJ}} = -\frac{1}{k_T(\phi, p)}[T - \tilde{T}_{\text{eq}}(\phi, p)] + \frac{L}{c_p}C + \frac{C_H|\vec{v}_a|(T_s - T_a)}{z_a} + \text{PBL Diffusion} \tag{3}$$

where $\tilde{T}_{\text{eq}}$ is a modified equilibrium profile, 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 mathematical details of the PBL diffusion of T are left out here and can be found in TJ16 and Reed and Jablonowski (2012). This model setup is similarly implemented within the Simpler Model options in CAM6 via the 'FTJ16' compset, which assumes an ocean-covered lower boundary with a prescribed sea surface temperature.

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

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

Here, q refers to the specific humidity in the atmosphere, 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 TJ16. Additionally we chose to emulate the large-scale precipitation rate given by

$$P_{\text{ls}} = \frac{1}{\rho_{\text{water}} g} \int_0^{p_s} C dp \tag{5}$$

where ρ_{water} is the density of water, g is gravity and p_s is the surface pressure.

2.1.3 Convection Scheme

The final step in our 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 Model framework and required some minor modifications to the TJ setup. The simplified BM technique follows the description by Frierson (2007) and we recommend their 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. 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. When we are 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 for the convection scheme to ensure a conservation of enthalpy in the deep convection regime. The author also describes three techniques to handling shallow convection; in our work we use the "shallower" scheme, in which the reference temperature is further modified in order to lower the depth at which shallow convection occurs, for the shallow convection calculations. This is considered the simplest technique within the BM scheme that still allows for 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 the CAM6 aquaplanet model (Williamson et al., 2012; Medeiros et al., 2016). Details on the aquaplanet and how it was used to identify our choices of $\mathrm{RH}_{\mathrm{BM}}$ and τ can be found in 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-to-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, referred to as the label or what we are predicting, and one or more independent variables, referred to as features or input variables. 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 neural networks, support vector machines, and random forests 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 makes up the vector $\vec{X}$ and our trained ML model will be $\hat{g}(\vec{X})$.

We utilize random forests to emulate the parameterization schemes. A random forest 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.

A random forest approach was chosen due to both its relative simplicity as an application of non-linear regression, along with its ability to inherently preserve underlying physical properties of the 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 random forests cannot extrapolate 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 restriction because it can avoid potential artifacts that could be inconsistent with the physics at play. For example, a random forest 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 neural networks, which historically have trouble 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 and scikitlearn (Pedregosa et al., 2011; Hoyer & Hamman, 2017). Xarray allows for easily manipulating data in the netCDF format, while scikitLearn is a well-maintained ML library that includes user-friendly random forest implementations for Python.

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 a finite volume dynamical core 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 $1.9° \times 2.5°$ - 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 the 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

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

again, 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 random forest for each predicted tendency or precipitation rate. This was a conscious decision that allows for a robust investigation into the effectiveness of random forests 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, which represents the majority of the available data from the 50 years for training. This number depends primarily on the complexity of the random forest parameters and the size and shape of the variable. For example, the moisture tendency is zero above roughly 250 hPa, which means we have six levels between 250 hPa and the model top that can be omitted from the process, resulting in significantly less 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 these emulators, which is just below the upper limit of our generated data. In contrast, the moist and convective temperature tendencies use 15 and 12 million samples, respectively. This was the largest we could use for these cases while remaining within our computational resource limits. The discrepancy between these two cases is a result of the size and complexity of each individually-optimized random forest. 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 our model's ability to emulate the parameterization. The testing data were not available during training and come from the final six years of the 60-year model run. It is important to evaluate model performance on data that the random forest has not seen while training in order to ensure our 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 is 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 random forest architecture for the problem. This is an important part of the ML workflow and we currently utilize the SHERPA hyperparameterization library to accomplish it (Hertel et al., 2020). Further details about the process of hyperparameter tuning and the final choices of hyperparameters can be found in tables S1 to S8 in the Supporting Information.

3 Results & Discussion

3.1 Snapshots & Mean Fields

Figures (1) and (2) show horizontal snapshots of the CAM6 (left column) output, our random forest ML predictions (middle column) and their differences (right column) for temperature and moisture tendencies, respectively. From top to bottom, the figures show each of the various physics schemes used: dry (Fig. (1) only), moist, and convection. We chose a snapshot from a randomly chosen time step at the model level closest to 850 hPa. The differences calculated in all plots are truth (CAM) subtracted from the random forest prediction. This means that positive and negative values correspond to over- and underestimations by the random forest. The snapshots in Figs. (1) and (2) immediately 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 random forest predictions, while still being able to identify areas of discrepancy. These snapshots also show how at a given time step, the ML prediction can reproduce the flow properties associated with baroclinic waves in the mid-latitudes. This is apparent in the heating tendency (1e, 1h) along the frontal zones, as well as decreasing moisture levels (2b, 2e) in these areas, corresponding to precipitation bands.

Figures (3) and (4) are zonally and temporally averaged temperature and specific humidity tendencies over the testing period of the final six years. The CAM6 physics and the random forest results are visually indistinguishable. The order of magnitude of the difference plots (right columns) is quite small in both figures. Additionally, Fig. (5) shows the same averaged field for the precipitation rates. The CAM6 output (blue) and the random forest ML predictions (dashed orange) 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. We can also identify some of the same physical behaviors that were obvious in the snapshots from Figs. (1) and (2), such as the heating bands in the mid-latitudes corresponding to the peaks around 40N & 40S in Figs. (3e) and (3h). In addition, the intense precipitation regions in the tropics are emulated well by the random forests as displayed in Figs. (5a, c). This precipitation is correlated with the intense tropical heating peaks in Figs. (3e, h).

The minor differences between ML predictions and the CAM6 output in Figs. (1 - 5) somewhat mirror minor artifacts that could arise through other common numerical changes to a GCM, such as dynamical core grid choices. Further, when we incorporate the zonal-mean time-means in Figs. (3), (4), and (5) these subtle discrepancies are averaged out and we obtain plots that become virtually indistinguishable for most cases. We also begin to see a hint that as we increase the complexity of the schemes, the random forest's skill begins to decrease. Figure (3) shows the difference plots between the predicted result and the original CAM6 output in the right column for all three cases of the temperature tendencies. The regions of larger magnitude differences for the convection case in plots (3f) and (3i) appear throughout a majority of the domain, while the dominant regions in the dry case in plot (3c) are more-so bound to the poles and equatorial regions. There is also an increase in magnitude throughout the difference plots between Figs. (3f) and (3i), as well as (4c) and (4f). This is another indication of a noticeable decrease in skill as complexity increases.

In Fig. (5), the emulated precipitation rates are even less distinguishable in the mean fields. The various peaks in the zonal-mean time-mean plots in Fig. (5) align closely with the areas of 'drying' in Fig. (4). This is in particular true for the equatorial region in both cases, dominant in the moist case, as well as in the mid-latitudes in the convection case. We also notice that there is a negligible difference in performance between the moist and convection cases' large-scale precipitation emulator. 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 random forest maintains virtually identical skill in this case across the two schemes.

3.2 Point-wise Comparison

Next, we show one-to-one scatter plots of the results from CAM and the random forests in Figs. (6) and (7) for the temperature and specific humidity tendencies at a level near 850 hPa and precipitation rates, respectively. This is a metric that allows for an effective visualization of the spread of our predictions. If the emulator were to produce the exact results as the original scheme, the points on these plots would follow the one-to-one line $y = x$, shown in orange. 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 Fig. (5) above. Each plot in Figs. (6) and (7) contains the $y = x$ (one-to-one) line in orange along with a least squares linear fit in blue. The least squares 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.

We also include a panel of histograms in Figs. (8) and (9) corresponding to Figs. (6) and (7), respectively' where N denotes the total number of test data points of 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 is saturated around the central bin, 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 shown metrics and model configurations, our results with the one-to-one scatter plots show exceedingly skillful ML emulators, in line with, if not superior to, what we find in the literature for similar work. This is something to keep in mind while we discuss potential limitations arising from increasing complexity.

For both of the large-scale precipitation rate emulators in Figs. (7a, b), the $y = x$ and least-squares fit lines overlap almost completely, and correspondingly, we have minimal spread in the underlying scatter plot. The convective precipitation (7c) plot shows the most visual spread among the precipitation rate scatter plots. Along these same lines, both tendencies in Figs. (6) and (8) show significantly more spread in the convection case over the moist case. However, we note that the all of the histograms in Figs. (8) and (9) show the overwhelming majority of point-wise differences fall within the first few bins and that while outliers occur, they are exceedingly rare. 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. All plots appear to have a slight bias to underestimate the extreme precipitation, as indicated by the direction of the deviation of the regression fit from the one-to-one line along with the consistent under estimated tails in the upper right region of all panels in Fig. 7). This is likely due to the inability for a random forest to predict a value that is not within the range of its training data set, as discussed in Section 2.2.

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

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

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 Figs. (1) and (7) in O'Gorman and Dwyer (2018), where in the author shows a panel of R^2 contours for temperature tendencies for various training scenarios also using random forests to emulate the tendencies.

We display a panel of R^2 plots for all of our tendencies in Fig. (10) and precipitation rates in Fig. (11). The plots show areas of significant skill, ranging from 0.8 to 0.99, for all cases. All fields also show regions of low R^2 values, around 0.4 and under, however, the overwhelming majority of the plots show at least $R^2 > 0.7$. The work in this paper is not meant to be a direct comparison to the work in O'Gorman and Dwyer (2018), as we use different model configurations and an entirely different GCM. It is still worth noting how this works as an effective reference due to the many similarities in both ML approach and aspects being emulated. We note again that all of our trained emulators show skill in line with various other examples of similar work within the literature, such as O'Gorman and Dwyer (2018), as well as Yuval et al. (2020).

The R^2 panels in Figs. (10) and (11) reveal a wide variety of aspects. For example, as we increase the complexity of our system, the random forest's global effectiveness decreases with regards to the R^2 skill. Excluding (10a), 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 (10c) there are more regions of $\mathrm{R}^2 \leq 0$ in the upper atmosphere, which was not seen in (10b). Along the same line, two pockets of $\mathrm{R}^2 \approx 0.3$ form around the tropics in (10e), while (10d) shows $\mathrm{R}^2 > 0.7$. These all support the expected conclusion that the effectiveness of random forests for emulating complex functions in climate models are impacted by the increasing complexity of those schemes.

We also note that the R^2 calculation can be an unreliable metric in regimes where there is minimal activity, which does occur in the white regime of Figs. (10a), (10d), and (10e). This is because when the values in the variance and the sum of squares are both functionally zero, they are still seen as floating point numbers of extremely small order and the equation above can lead to the result

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

For the dry case in plot (10a), this occurs around the equator in the mid-atmosphere. Similarly, this occurs in the upper atmosphere for the moisture tendencies in (10d) and (10e). For the dry case, there is very little heating or cooling occurring on average in this area of the atmosphere and for the moist case there is in fact no forcing occurring at upper levels. 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.

There is virtually no noticeable drop-off in skill between the moist and the convection cases for the large-scale precipitations rates (Fig. 11a,b). This is again because their functional relationships are identical in each case. However, we see that the convective precipitation scheme suffers a minor loss in skill when compared to the large-scale precipitation emulators. The emulator is still among the highest skill model, with $\mathrm{R}^2 > 0.78$ globally. However, this is in-line with what we observe with the tendencies in Fig. (10) and the expected result: that as we increase complexity, the effectiveness of the random forest decreases, even if only marginally for some cases inspected in this work.

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. It is important that we do not use every possible quantity as inputs as this increases the computational demand of training these emulators. We know what input fields are used to calculate the functions that we emulate, as discussed in section 2.1. These tend to include the temperature, pressure, specific heat, and heat fluxes for example. One input field that we investigated more closely was 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 (12) shows the R^2 comparison of explicitly including the RH (left) and not including it (right), using identical random forest setups, trained independently, for the moist specific humidity tendency. The random forest 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. In turn, this impacts the make up of moisture in the atmosphere which indirectly influences all predicted fields. This is an example of how important it is to approach ML problems in the physical sciences with our physical intuition in mind.

We also assessed how the model is dependent on the amount of training data points provided. Figure (13) shows the impact that training data can have on the moisture tendency, as we decrease the amount of training data used for otherwise identical random forest emulators. Our current models all used around 12 to 20 million samples in which to train on, as outlined in Supporting Information tables S1 to S8. In this specific test, 20 million samples led to the most skillful R^2 field as well as the highest overall R^2 at 0.876. When decreasing the number of samples we see a decrease in skill in Fig. (13), as expected. It does appear with a minimum of 500,000 samples, we are able to effectively emulate the moisture tendency with $\mathrm{R}^2 > 0.8$ globally. However, once we decrease that number by another order of magnitude we lose much of our skill, including a genuinely poor emulator for only 5,000 training samples. We also find that as we increase above five million samples, the R^2 begins to level out close to 0.9, indicating that roughly five or 10 million samples is a sufficient minimum for our work. Investigations like these can help to guide us as we attempt to find the balance between skillful ML models and computationally feasible approaches.

4 Concluding Thoughts & Applications to Future Work

Individual random forests are developed and trained for emulating temperature tendencies, specific humidity tendencies, large-scale precipitation rates, and convective precipitation rates from physical parameterization schemes within three simple physics model configurations within the CAM6 model framework. The configurations built upon one another in a hierarchy of complexity, beginning with the dry case, before moving to the moist and culminating with the moist case coupled with a convection scheme. We utilized a broad scope of training data over a 60-year model run for each configuration and allowed the SHERPA hyperparameter optimization tool to freely optimize each individual random forest. This allowed us to create robust emulators in order to probe the limits of skill achievable via random forest emulators of sub-grid physics schemes in an offline mode.

All of our emulators showed significant skill when tested on the test data over the final six years of model output. With the given metrics, our emulators showed results at least as skillful as other similar examples within the literature, while in many cases outperforming similar work. This statement bares reminding that in a majority of cases our model configurations were less-complex than we find in the current literature, so direct comparisons are not possible. We also note the benefits of using random forests rather than other methods, such as deep learning, since random forests inherently preserve many physical laws. An example being the non-negativity of precipitation due to the inability for random forests to make a prediction outside of the range of the data it was trained on.

Further, we identify indications of limitations in using random forests for emulating physical parameterization schemes, even within our highly simplified hierarchy of configurations. In many cases, there were noticeable decreases in skill as the complexity of the physics scheme was increased. This raises interesting questions about where random forests fit within the overall approach to the community's interest in using ML to emulate these sub-grid processes and otherwise augment aspects of climate and weather models. Balancing the trade-offs between physical realism, computational efficiency, and model complexity should inform the choice of ML technique, especially when looking forward towards state-of-the-art model complexity. Random forests are unlikely to remain as skillful, particularly in full-tendency emulators like ours, in cases with such a substantial increase in overall complexity. In regards to this work, our next intriguing step will be to couple the random forests to the CAM6 implementation and analyze how they perform in an online mode. Investigating whether the rare, yet present, outliers impact the stability of the coupled model will provide further insight into where random forests may fit into the future of data science-augmented climate and weather models. Based solely on this work, we expect it to likely fall within the emulation of individual components that can take advantage of the inherent conservation of physical laws, such as convection schemes or the radiation parameterization.

Acknowledgments

This work was made possible by the National Science Foundation's Graduate Research Fellowship Program and the NOAA grant NA17OAR4320152(127). We would like to acknowledge high-performance computing support from Cheyenne (doi:10.5065/D6RX99HX) and the Casper data analysis server. These resources were provided by the Computational and Information Systems Laboratory of the National Center for Atmospheric Research (NCAR), sponsored by the National Science Foundation. The CAM6 output data used for all three cases of machine learning in the study are available at Limon (2022) and all machine learning related scripts at https://github.com/gclimon/ml_climate.

References

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., 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)
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
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
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

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.). University of Michigan - Deep Blue Data. Retrieved from `https://doi.org/10.7302/r6v3-s064` doi: 10.7302/r6v3-s064

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.

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

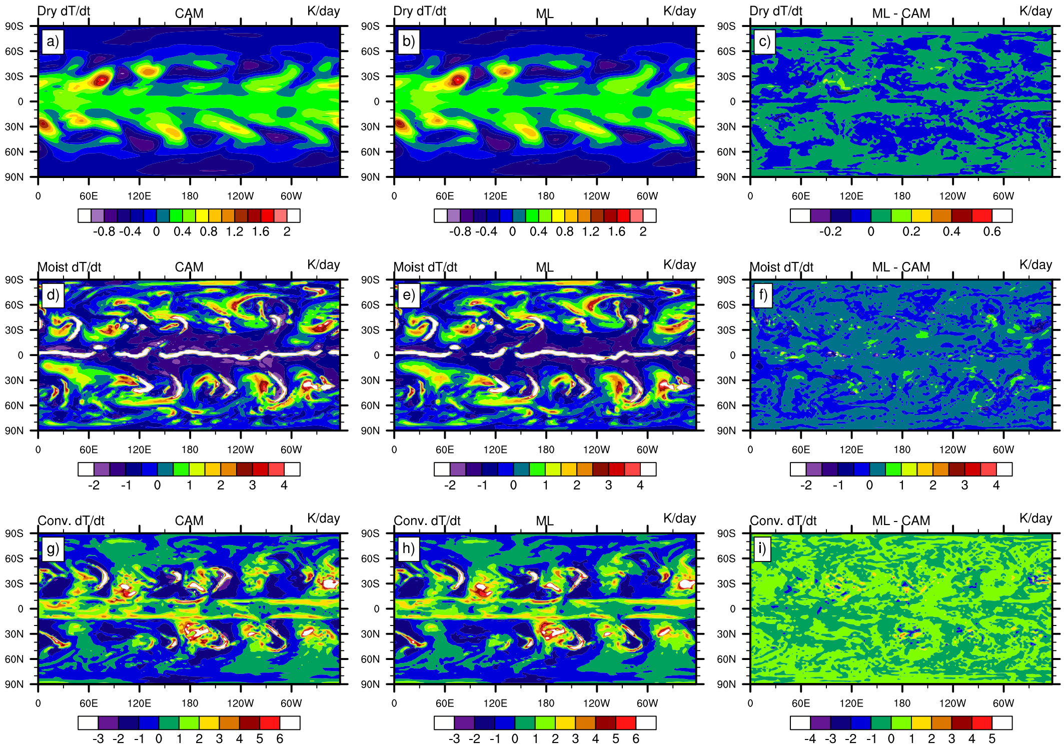

Figure 1: Time step snapshot plots of CAM6 and ML predicted temperature tendencies and their differences near 850 hPa. Top row corresponds to the dry case, middle row to the moist case, and bottom to the convective case. Left column corresponds to CAM6 output, middle the random forest ML predictions, and right depicts their respective differences.

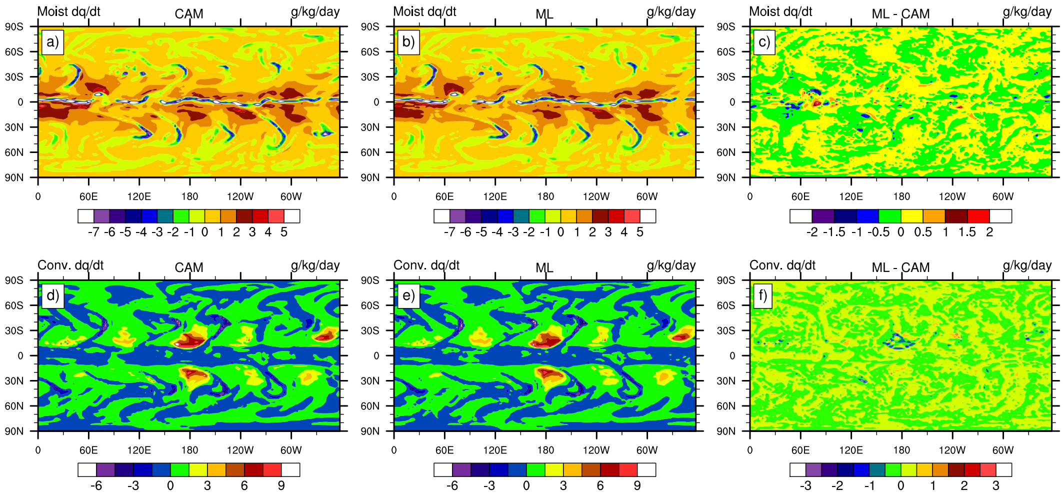

Figure 2: Time step snapshot plots of CAM6 and ML predicted specific humidity tendencies and their differences near 850 hPa. Top row corresponds to the moist case and bottom to the convective case. Left column corresponds to CAM6 output, middle the random forest ML predictions, and right depicts their respective differences.

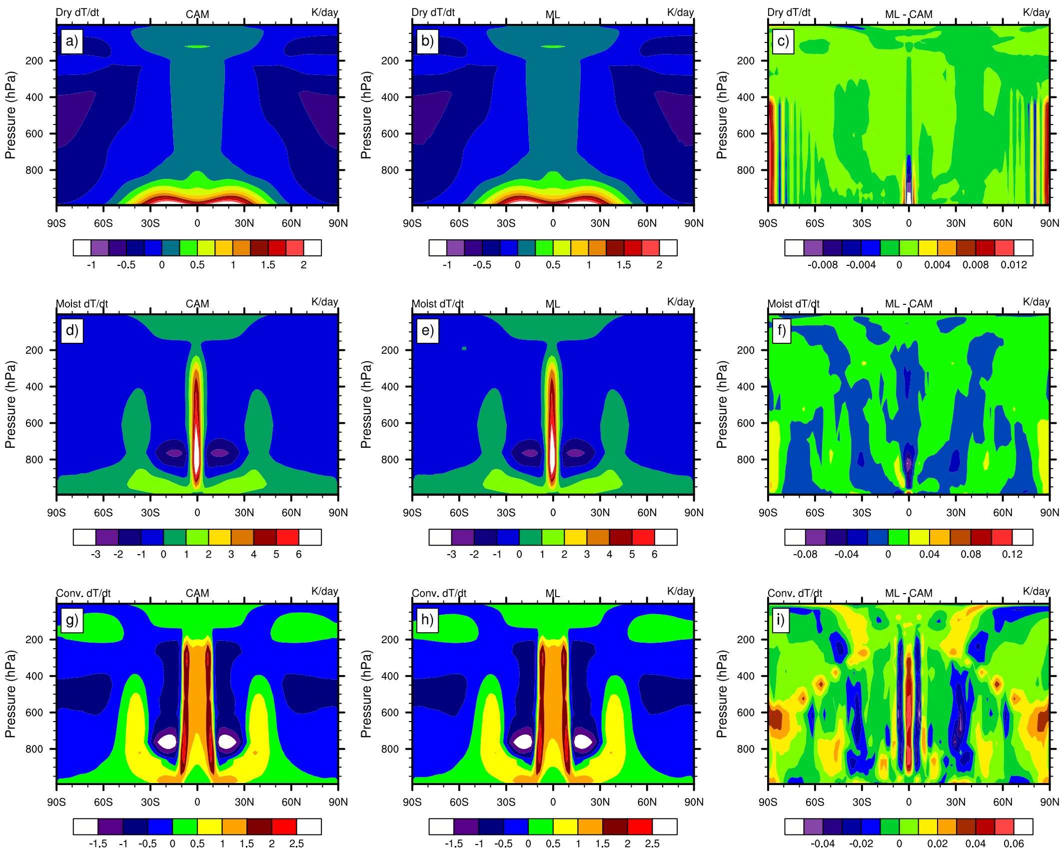

Figure 3: Zonal-mean time-mean plots of original and predicted temperature tendencies and their differences over the full testing data set. Top row corresponds to the dry case, middle row to the moist case, and bottom to the convective case. Left column corresponds to CAM6 output, middle the random forest ML predictions, and right depicts their respective differences.

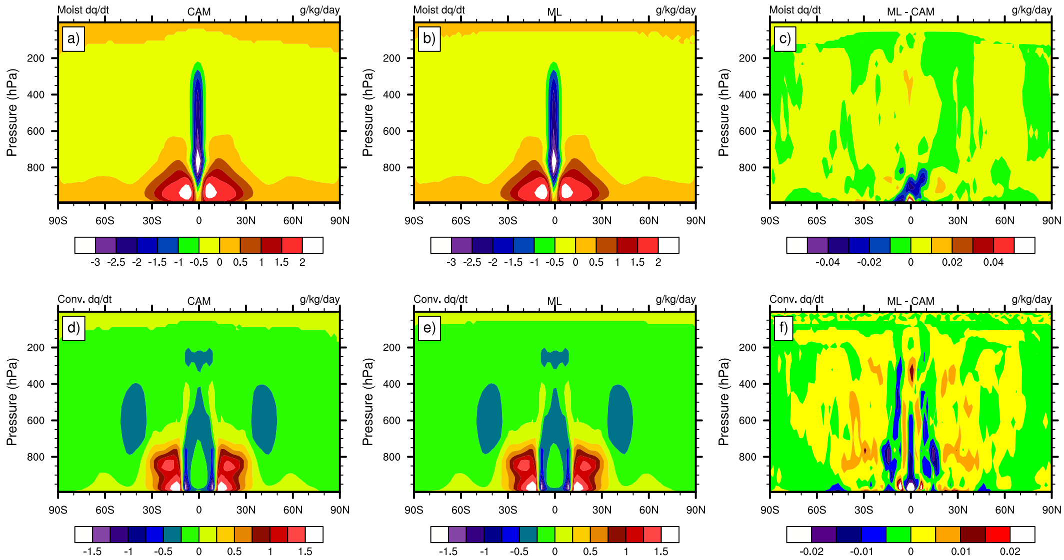

Figure 4: Zonal-mean time-mean plots of original and predicted moisture tendencies and their differences over the full testing data set. Top row corresponds to the moist case, and bottom to the convective case. Left column corresponds to CAM6 output, middle the random forest ML predictions, and right depicts their respective differences.

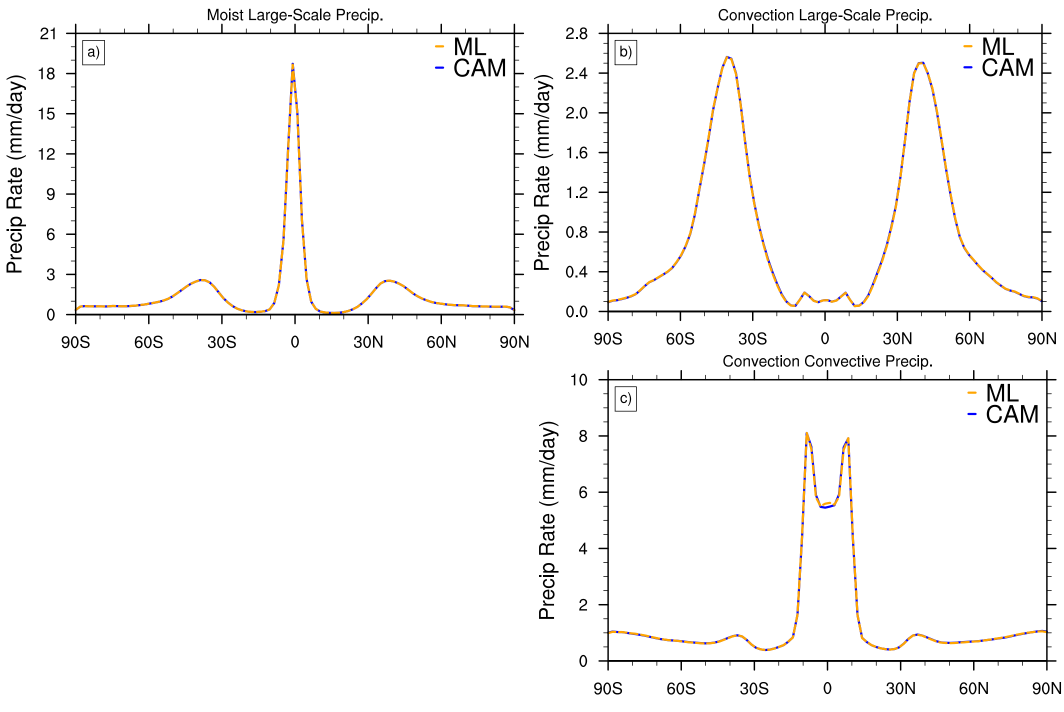

Figure 5: Zonal-Mean Time-Mean plots of CAM6 (blue) and ML predicted (orange) precipitation totals over the full testing data set. Top row corresponds to the large-scale precipitation (eq. 5) and bottom to the convective precipitation. Left column corresponds to moist case, while the right is the convective case.

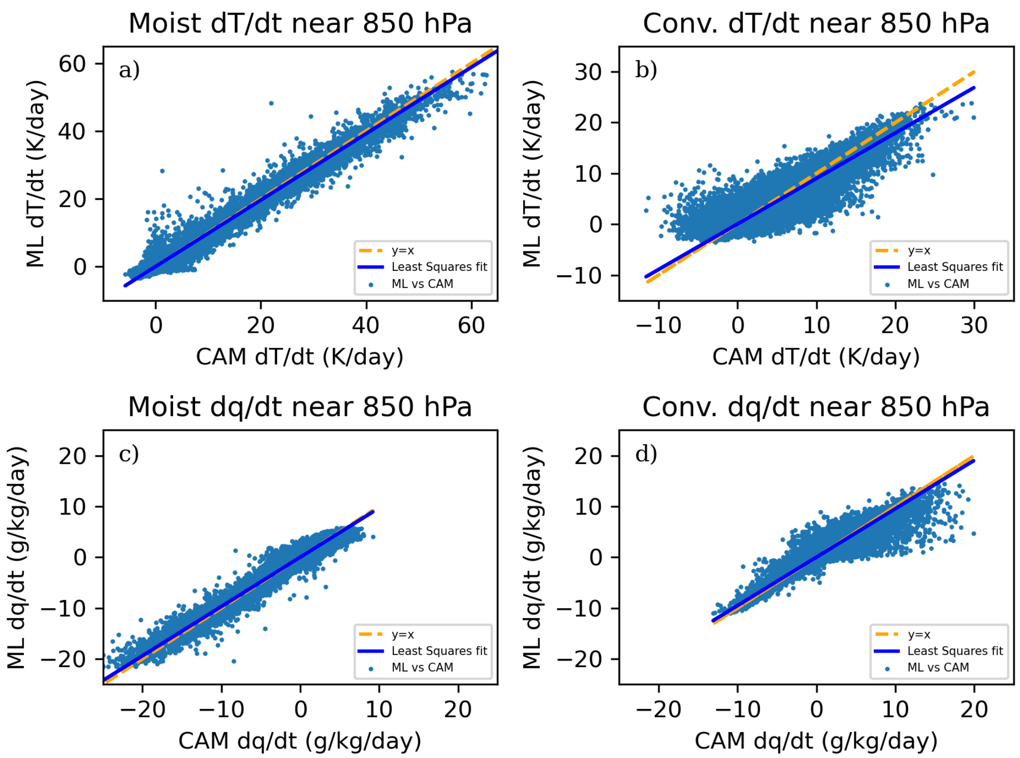

Figure 6: Direct comparison scatter plots for ML predicted values (y-axis) against original CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for (a) moist-case temperature tendency, (b) convection-case temperature tendency, (c) moist-case moisture tendency, and (d)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

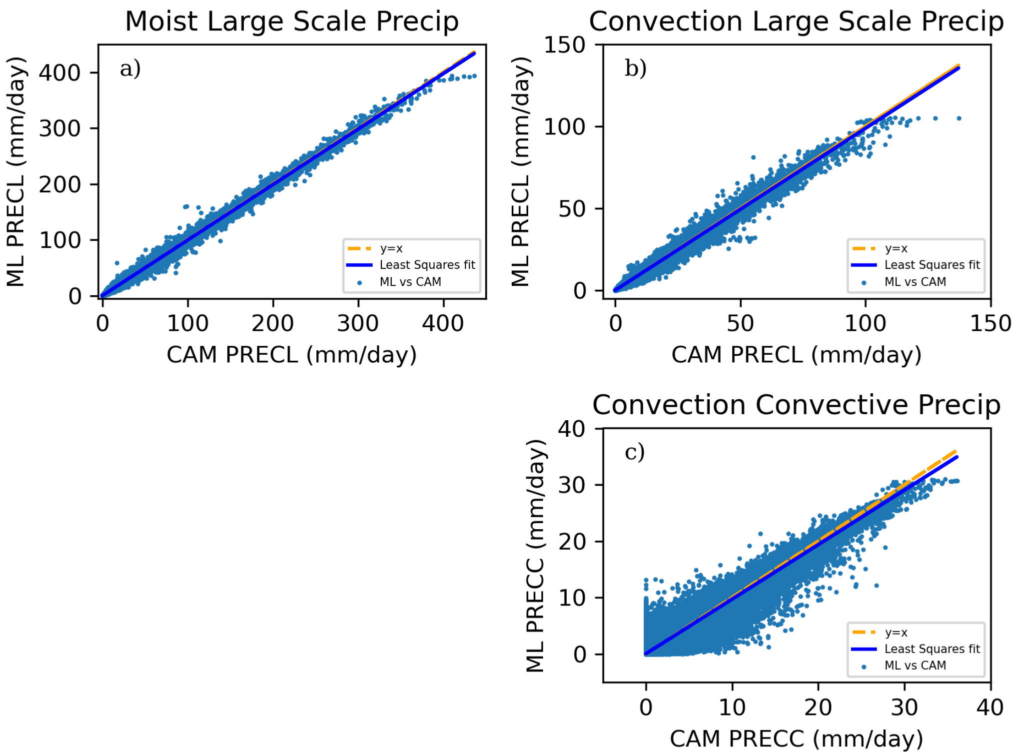

Figure 7: Direct comparison scatter plots for ML predicted values (y-axis) against original CAM6 output (x-axis) for all horizontal grid points near 850 hPa over the testing data for (a) moist-case large-scale precipitation, (b) convection-case large-scale precipitation, and (c) convection-case convective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

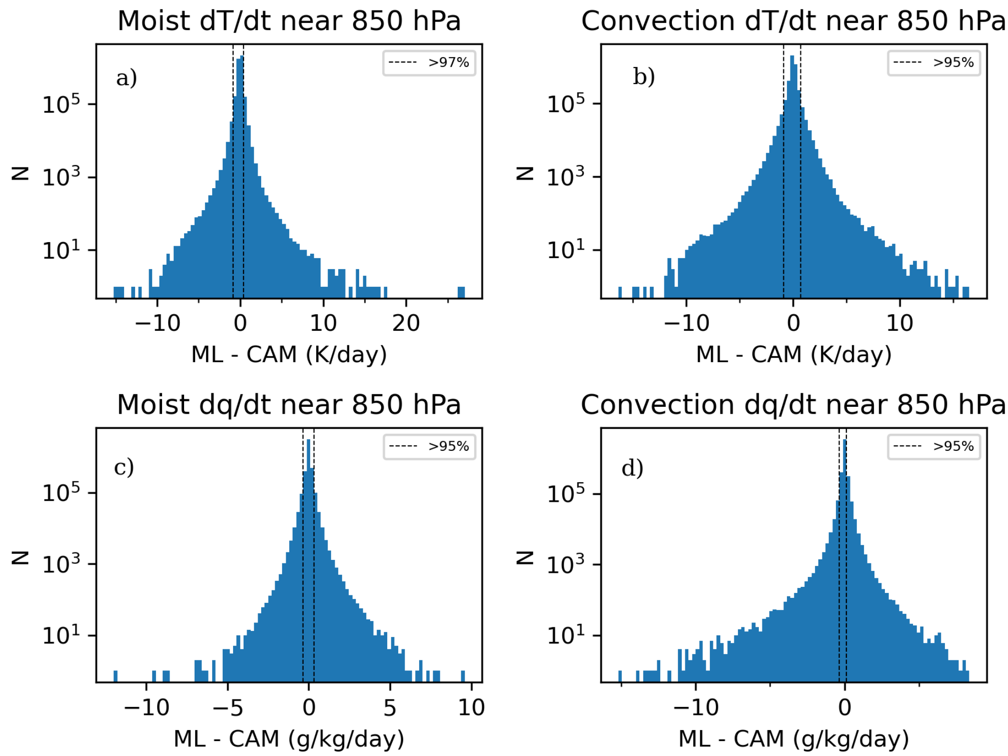

Figure 8: Histograms of the point-wise difference (ML - CAM6) for the temperature (top) and specific humidity (bottom) tendencies, corresponding to the scatter plots in Fig. (6) on a log scale using 100 bins. Percentage of data contained within the black dashed lines are indicated in individual legends and conveys the significance of the log-scale and the overwhelming majority of instances occurring near the peak of the distribution.

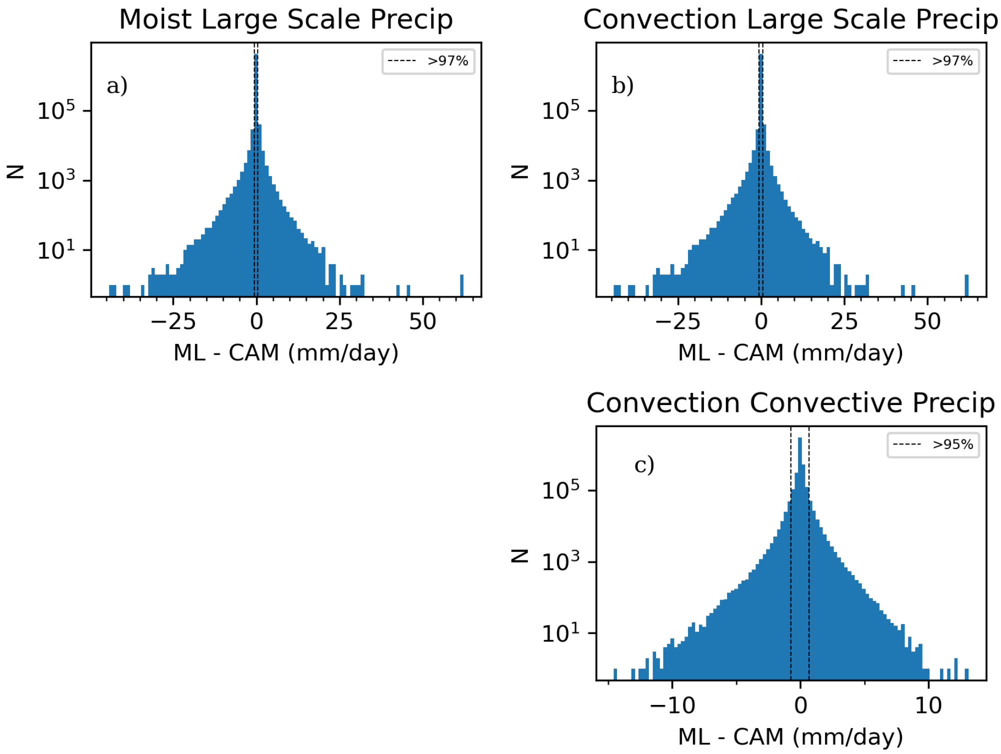

Figure 9: Histograms of the point-wise difference (ML - CAM6) for the precipitation rates corresponding to the scatter plots in Fig. (7) on a log scale using 100 bins. Percentage of data contained within the black dashed lines are indicated in individual legends and conveys the significance of the log-scale and the overwhelming majority of instances occurring near the peak of the distribution.

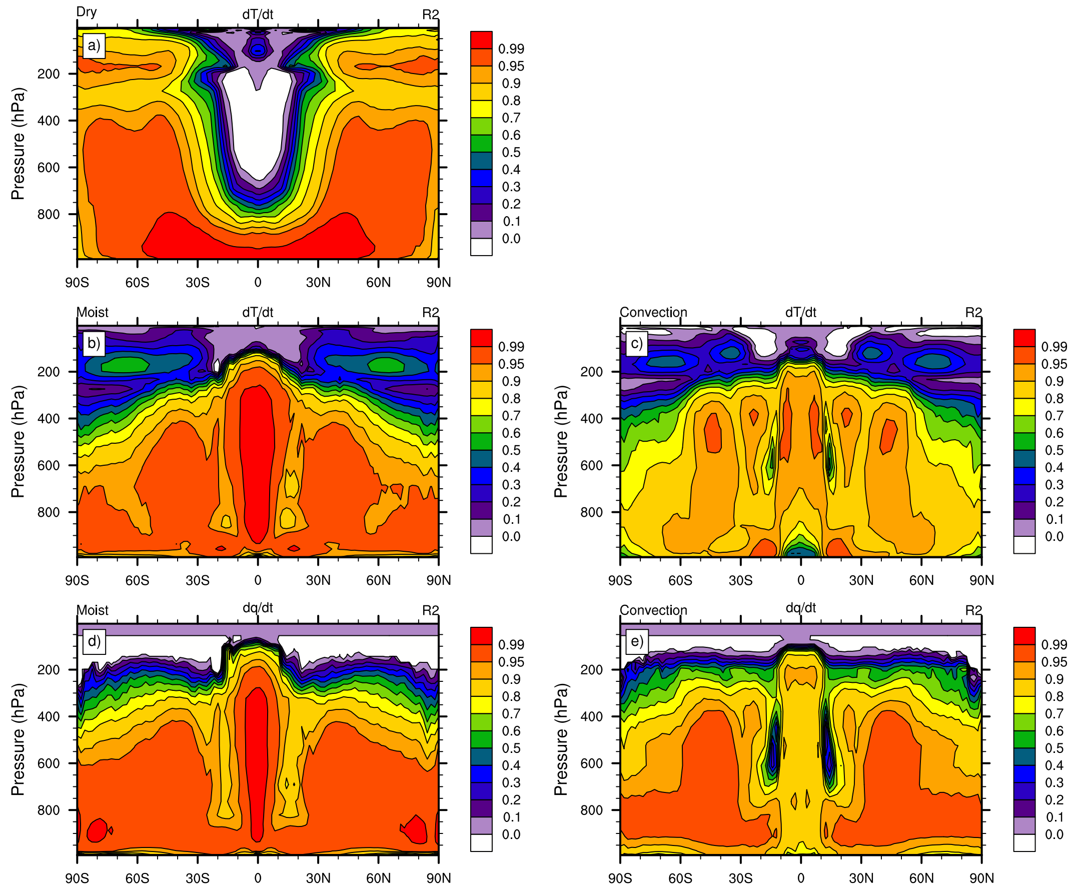

Figure 10: R^2 calculations over the zonal & temporal dimensions via eq. 10. (a) Dry temperature tendency, (b) moist temperature tendency, (c) convection temperature tendency, (d) moist moisture tendency, and (e) convection moisture tendency.

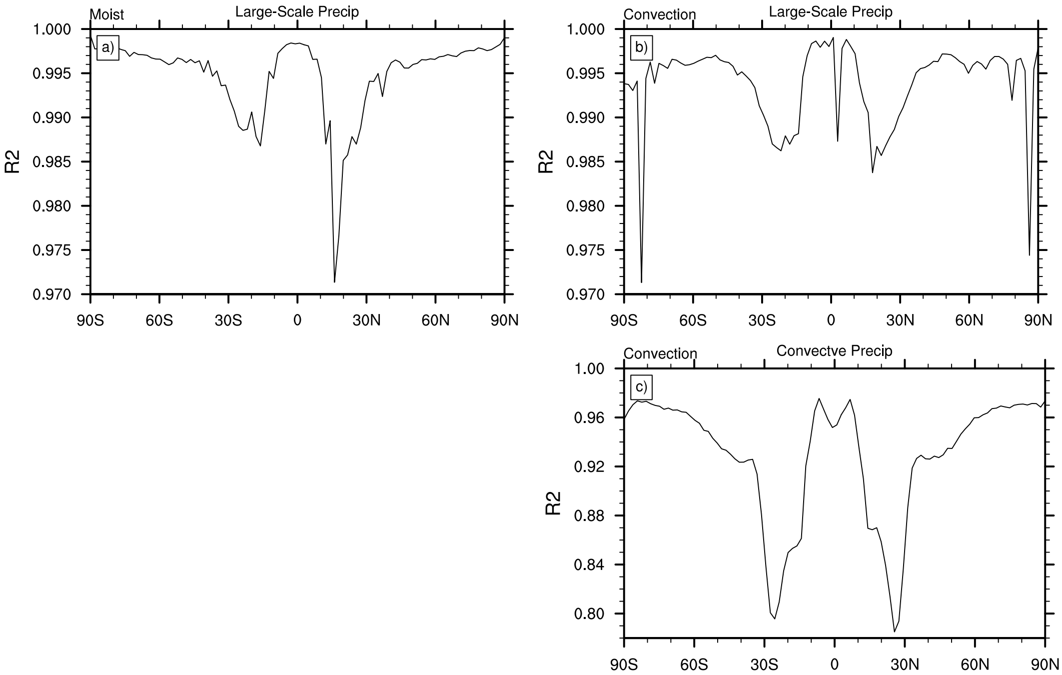

Figure 11: R^2 calculations over the zonal & temporal dimensions via eq. 10. (a) moist large-scale precipitation, (b) convection large-scale precipitation, and (c) convection convective precipitation.

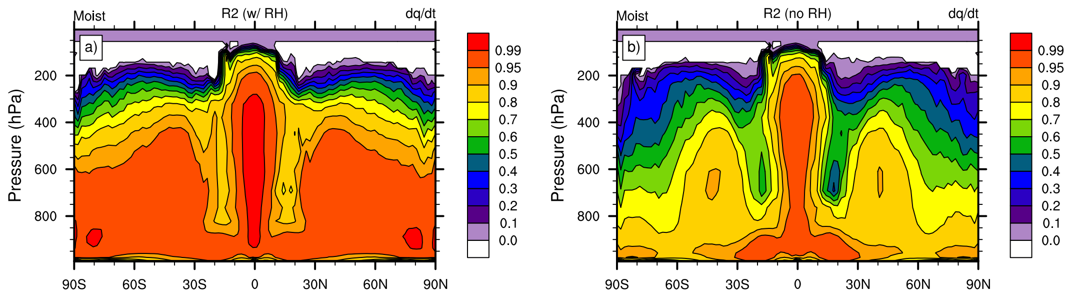

Figure 12: Comparison of R^2 plot - as defined in Fig. (10) - with (a) and without (b) relative humidity as a feature for ML prediction of the moisture tendency for the moist case. Figure (12a) reproduces Fig. (10d).

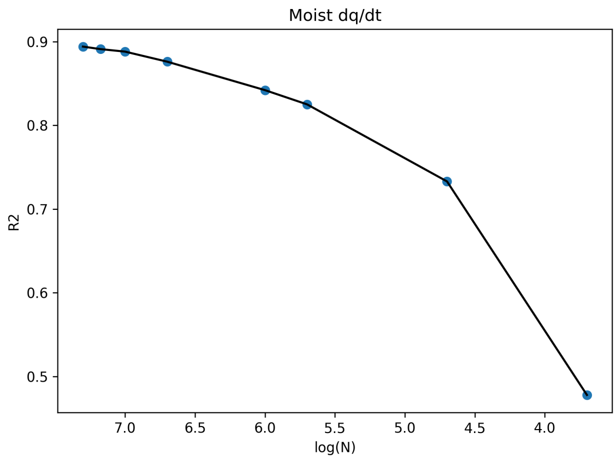

Figure 13: Globally-averaged R^2 value (y-axis) for ML prediction of the moisture tendency for the moist case using varying number (N) of training samples (x-axis) on a log scale.