RESEARCH ARTICLE
10.1029/2020MS002076

Key Points:
• An advanced deep residual convolutional neural network is used for moist physics parameterization.
• The neural network is trained with a 1-year-long simulation from superparameterized CAM5 with actual global land-ocean distribution.
• It reproduces accurately the target simulation in independent test data evaluation and in single-column model prognostic validations.

Correspondence to:
G. J. Zhang and X. Huang.
gzhang@ucsd.edu;
hxm@tsinghua.edu.cn

Citation:
Han, Y., Zhang, G. J., Huang, X., & Wang, Y. (2020). A moist physics parameterization based on deep learning. Journal of Advances in Modeling Earth Systems, 12, e2020MS002076. https://doi.org/10.1029/2020MS002076

Abstract Current moist physics parameterization schemes in general circulation models (GCMs) are the main source of biases in simulated precipitation and atmospheric circulation. Recent advances in machine learning make it possible to explore data-driven approaches to developing parameterization for moist physics processes such as convection and clouds. This study aims to develop a new moist physics parameterization scheme based on deep learning. We use a residual convolutional neural network (ResNet) for this purpose. It is trained with 1-year simulation from a superparameterized GCM, SPCAM. An independent year of SPCAM simulation is used for evaluation. In the design of the neural network, referred to as ResCu, the moist static energy conservation during moist processes is considered. In addition, the past history of the atmospheric states, convection, and clouds is also considered. The predicted variables from the neural network are GCM grid-scale heating and drying rates by convection and clouds, and cloud liquid and ice water contents. Precipitation is derived from predicted moisture tendency. In the independent data test, ResCu can accurately reproduce the SPCAM simulation in both time mean and temporal variance. Comparison with other neural networks demonstrates the superior performance of ResNet architecture. ResCu is further tested in a single-column model for both continental midlatitude warm season convection and tropical monsoonal convection. In both cases, it simulates the timing and intensity of convective events well. In the prognostic test of tropical convection case, the simulated temperature and moisture biases with ResCu are smaller than those using conventional convection and cloud parameterizations.

Plain Language Summary Moist physics parameterization, an algorithm for clouds and convection based on empirical relationships and limited observations, is the main source of biases in rainfall and atmospheric circulation in global climate model simulations. A sophisticated deep learning algorithm with refined inherent architecture is introduced as a new parameterization in this study. It is trained with a year-long simulation from a global climate model that has a two-dimensional high-resolution cloud-scale model embedded in each climate model grid box, where clouds and convection are explicitly simulated. In addition, we consider energy conservation and past history of atmospheric states, clouds, and convection. It is designed to predict heating and moistening rates from convection and clouds, as well as cloud water and ice amount. This new parameterization accurately reproduces target simulations in 1-year independent testing and also performs well in predicting convective events in both midlatitude summer continental land convection and tropical monsoonal convection.

1. Introduction

Clouds play a fundamental role in the radiation budget and hydrological cycle of the Earth system (Tao et al., 1996; Wielicki et al., 1995). The latent heat release from clouds and convection interacts with the atmospheric circulation to affect the global mass and energy transport and distribution (Bony et al., 2015). However, clouds and convection are among the most difficult processes to represent in climate models (Bony et al., 2015; Randall et al., 2003). The spatial scales of cloud-related processes vary from micron scale in cloud droplet formation to thousands of kilometers in tropical disturbances. The cloud microphysical processes must be parameterized. The current general circulation models (GCMs) cannot resolve convection either, due to their coarse resolutions, and as such convection needs to be parameterized as well. The parameterization of convection and clouds is at the heart of many problems in GCMs, including the simulation of the global mean precipitation distribution such as Intertropical Convergence Zone (ITCZ) (G. J. Zhang et al., 2019), climate variability such as Madden-Julian Oscillation (MJO) and associated intraseasonal
variability (Jiang et al., 2015), and intermodal spread in future climate projections (Bony et al., 2015; Stevens & Bony, 2013).

Convective parameterization schemes (e.g., Arakawa & Schubert, 1974; Tiedtke, 1989; G. J. Zhang & McFarlane, 1995, and many more) are based on limited observations and empirical relationships between convection and some observable variables. In these schemes, convective clouds are idealized as individual kilometer-sized cells. Although convection schemes in general can represent the convective transport of heat and moisture, as well as the condensational heating qualitatively, many important characteristics of convection are not represented. For example, most convective parameterization schemes do not represent convection cell self-aggregation (Bony et al., 2015) or mesoscale organization (Moncrieff & Liu, 2006). They do not represent cold pools generated by rain reevaporation in downdraft (Torri & Kuang, 2016) or the influence of wind shear (Anber et al., 2016). Neither do they have any memory of the immediate past history of convection, which is important for simulating the lifespan of organized convective systems (Davies et al., 2009).

Cloud-resolving models (CRMs) have been used to simulate convection since 1980s (Tao et al., 1987). The advancement of CRMs in the last four decades enables it to simulate convection much more realistically today, including organized mesoscale convective systems (Feng et al., 2018). CRMs have also been used to represent convection in GCMs as so-called superparameterization (Arakawa, 2004; Khairoutdinov et al., 2005). Khairoutdinov et al. (2005) developed a superparameterized version of the National Center for Atmospheric Research Community Atmosphere Model (NCAR CAM, hereafter referred to as SPCAM). SPCAM performs much better in capturing the characteristics of convection in the GCM such as mesoscale convection system, diurnal cycle of convection, monsoon, and MJO (Jiang et al., 2015; Khairoutdinov et al., 2005; Pritchard & Somerville, 2009; Randall et al., 2003; Wyant et al., 2006). However, it is too expensive (Randall et al., 2003) for future climate projections, which require century-long and ensemble simulations.

In the last several years, data-driven machine learning (ML) has been explored for applications in climate sciences. Krasnopolsky et al. (2013) used an ensemble of neural networks (NNs) to develop a stochastic convective parameterization for numerical weather prediction and climate models, with training data from CRM simulations of convection during TOGA-COARE in tropical western Pacific. The NN-based parameterization was able to simulate the main features of diabatic heating and cloud distribution in the NCAR CAM4. Gentine et al. (2018) developed a deep NN to predict cloud and radiation processes trained by data from an aquaplanet configured SPCAM. Rasp et al. (2018) successfully coupled an NN-based parameterization into an aquaplanet GCM and performed multiyear prognostic simulations, which reproduced well the SPCAM climatology and higher-order statistics such as variability. Brenowitz and Bretherton (2018) developed an NN-based parameterization by training it with coarse-grained data from a near-global cloud-system resolving model simulation of tropical convection and circulation and tested it prognostically in a single-column model (SCM). Brenowitz and Bretherton (2019) extended its implementation into a GCM and analyzed its linear response to perturbations of input variables. They found that the heat and moisture tendencies so parameterized are coupled too strongly to the atmospheric states in the upper troposphere above 200 hPa and suggested remedies for it. O’Gorman and Dwyer (2018) used random forest decision trees to develop an ML-based convection parameterization. The training data were directly from a GCM using a conventional convective parameterization scheme. Recently, Yuval and O’Gorman (2020) replaced the training data with output from a high-resolution 3-D model run on idealized equatorial beta plane. By integrating this parameterization into a coarse-resolution atmospheric model, they achieved stable simulations that replicated the climatology of the high-resolution model.

All of these ML-based parameterization studies are based on aquaplanet setups; thus, continental convection is not considered or used in the training. In moist physics processes such as convection and large-scale condensation, moist static energy (h) is conserved if freezing is not considered. Such energy conservation requirement is not explicitly built into the ML-based parameterization although some schemes appear to do well in conserving mean squared error (MSE) judged a posteriori (Rasp et al., 2018). An exception is the recent work by Beucler et al. (2019), who considered mass, enthalpy, and radiation conservations in the design of their NN. Furthermore, none of these parameterization schemes considers the history or memory of convection. The NNs in previous studies are all fully connected deep NNs. They are not the most sophisticated in terms of depth (less than 10 layers) and accuracy and could have many potential
problems such as overfitting (Gardner & Dorling, 1998). More advanced networks have been developed in computer vision (LeCun et al., 2015) that can be applied to climate science.

In this study we use a deep convolutional residual neural network (ResNet; He et al., 2016) to emulate moist physics processes in a realistically configured (as opposed to aquaplanet) SPCAM with actual global land-ocean distribution. We explicitly replace the fully connected layers in NNs with convolution layers and reformulate the layers by learning through residual functions via identity shortcuts. These techniques make ResNet highly accurate and capable of going much deeper with many hidden layers. ResNet has applications in many fields, such as image classification (He et al., 2016) and game playing (Silver et al., 2017). However, it has not been explored in applications to moist physics parameterization in GCMs. This study is the first attempt to use ResNet architecture for this purpose.

The organization of the paper is as follows. Section 2 presents the details of the NN design. Section 3 shows the results of semipropagostic test of the NN. Section 4 performs a set of sensitivity tests on different NN architectures and convective memory. SCM tests using field observational data are performed in section 5. A summary and discussion is given in section 6.

2. Construction of an NN-Based Parameterization

2.1. Base Model Configuration and Training Data

We choose SPCAM (Khairoutdinov et al., 2005) to provide the training data. The SPCAM implements a 2-D CRM (the System for Atmospheric Modeling SAM, Version 6.8.2) in CAM5.2 to replace its conventional parameterization for moist convection (G. J. Zhang & McFarlane, 1995) and large-scale condensation (Morrison & Gettelman, 2008). The 2-D CRM has 32 grid points in the zonal direction and 30 vertical levels. The dynamic framework of CAM5 has a horizontal resolution of 1.9° × 2.5° and 30 levels that are shared with the embedded CRM. The SPCAM used in this study has a coupled land surface model Community Land Model 4.0 (CLM4.0; Oleson et al., 2010). It uses a prescribed climatological sea surface temperature field that comes with the CAM5 model (Hurrell et al., 2008). It is run for 3 years and 3 months from 1 January 1998 to 31 March 2001 with a time step of 20 min. The first year and 3 months are for spin up, the second year is used for training, and the third year is used for testing and evaluation. The training data from SPCAM are output every time step. The 20-min time step for model run and output is set to match the default time step of the SCM version of CAM5 (see section 5).

2.2. NN Setup

With the advent of big data era, deep learning is beginning to have a great impact on weather and climate research. A deep learning architecture is a multilayer stack of simple modules (LeCun et al., 2015). It has evolved into various complex structures over years of development, making it capable of implementing highly intricate functions. Newly developed deep learning methods can provide new solutions for weather and climate forecasting and obtain better results in many areas, such as precipitation nowcasting, seasonal forecasting, and model downscaling (Reichstein et al., 2019). Previous studies have shown superiority of deep NNs over their shallow counterparts: higher accuracy (Gentine et al., 2018), faster to converge during training, and better neuron utilization efficiency (Rolnick & Tegmark, 2017). In particular, convolutional neural network (CNN) performs outstandingly in mining the relationship between the predictor fields and the predictand index with far fewer parameters than in fully connected deep NNs, because by design it can automatically obtain local correlations and translational equivariance by using groups of shared weights and sparse connections instead of using global connections with independent weighted links (LeCun et al., 2015).

In addition to using convolutions, we also use identity shortcuts in the NN, thus making it a residual network (ResNet). ResNet was first introduced by He et al. (2016) to solve network degradation problems; that is, as the depth of a NN increases, its performance becomes saturated and begins to degrade when the depth further increases. ResNet uses an identity shortcut to connect the input layer and the output layer (see inset in Figure 1). The stacked layers combined with the shortcut form a residual unit (Resunit). Many residual units are then stacked together to transform a deep fully connected NN or CNN into a deep residual NN. The main idea of a deep residual network is to let each Resunit handle some part of the errors between inputs
and outputs. The top few Resunits fit the major errors, and the ones below fit the remaining, higher-order errors. With more Resunits, less nonlinear errors will remain.

Here we use ResNet to develop our ML-based subgrid-scale parameterization. We implement one-dimensional convolutions to extract vertical profile features such as systematic dynamic advection or unstable vertical temperature and moisture structures when there is convection and use them to predict subgrid-scale temperature and moisture tendencies and cloud hydrometeor from moist physics processes.

Considering the computational efficiency, the ResNet used in this study is in moderate depth and width with 10 Resunits, making it a 22-layer deep CNN with approximately 1 million parameters in total. As shown in Figure 1, we use 1-D convolutional layers with 128 feature vectors (1-D feature maps) and 128 corresponding filter banks with a kernel size of 3. The feature vector length of all layers is kept constant at 33 by padding with zeros at both ends. The activation algorithms inside each Resunit are Rectified Linear Activations (ReLUs), and the last one in the output layer is hyperbolic tangent (tanh). The ReLU activation function is used to resolve the problem of vanishing gradients when training a deep NN (Glorot et al., 2011). The learning target is the moist physics variables predicted by the 2-D CRMs in SPCAM. We refer to the ResNet used in this study as ResNet for convection, or ResCu for short.

The input data for NN training vary in different studies. For instance, in Rasp et al. (2018), the input consists of model state profiles including temperature, specific humidity and meridional wind, and scalar variables including surface sensible and latent heat fluxes, surface pressure, and insolation. For the input data into ResCu, we include variables that are used to force the CRM. These are vertical profiles of temperature ($T$) and moisture (specific humidity $q_v$) and large-scale temperature and moisture tendencies $\frac{\partial T}{\partial t}$ and $\frac{\partial q_v}{\partial t}$.
Table 1: Input and Output Variables and Their Normalization Factors

| Inputs | Units for normalization | Outputs |
|--------|-------------------------|---------|
| dq (t0, l_1, l_2, l_3, l_4, z) | 1.5 × 10^{-6} (kg/kg/s) | |
| dT (t0, l_1, l_2, l_3, l_4, z) | 2.5 × 10^{-3} (K/s) | |
| q_v (t0, l_1, l_2, l_3, l_4, z) | 2.2 × 10^{-4} (kg/kg) | |
| T (t0, l_1, l_2, l_3, l_4, z) | 323 (K) | |
| SLHF/Lv (t0, l_1, l_2, l_3, l_4) | 3.1 × 10^{-4} (kg/kg/s) | |
| SS2HF (t0, l_1, l_2, l_3, l_4) | 6.5 × 10^{-1} (K/s) | |

Note: The variables with “z” are 30-layer vertical profiles, and those without are scalars stacked below each profile according to its time step. This results in 36 vectors or channels with 33 elements in each vector for all input variables and 4 channels for output. All data are normalized by dividing the original values by the normalization factors in the middle column of the table.

The training procedure of ResCu is a supervised learning, which means ResCu is fed with input and target output pairs from SPCAM’s simulation. During this learning process, these data are fed repeatedly to minimize the loss function via backward propagation. In order to improve the training speed, we selected a total of 800 grid points randomly distributed in three latitude zones in proportion to their relative surface area. The three zones consist of the tropics (30°S to 30°N), midlatitudes (60°S to 30°S and 30°N to 60°N), and high latitudes (90°S to 60°S and 60°S to 90°N). There are 21 million training samples in total. The NN is trained over 100 epochs (an epoch is one complete cycle through the full training data set to be learned by the NN) to ensure that all the training samples have been fully learned without overfitting. There is no overfitting during our 100-epoch training either since our training samples are sufficiently large compared to the number of parameters in the NN (21 million samples vs. 1 million parameters), similar to what Gentile et al. (2018) found in their NN training.

2.3. Moist Static Energy Conservation in the NN

In moist physics processes (convection and cloud condensation), the atmospheric moist static energy is conserved in the absence of freezing. This can be seen from the tendencies of temperature and moisture from convection and condensation:

\[
\left( \frac{\partial T}{\partial t} \right)_{mp} = -\left( \frac{\partial q}{\partial p} \right)_{mp} + L_v (c - e) + L_f c_f
\]

\[
\left( \frac{\partial q}{\partial t} \right)_{mp} = -L_v \frac{\partial q}{\partial p} - L_v (c - e)
\]

from the dynamic core of SPCAM’s host model CAM5 and planetary boundary layer (PBL) diffusion. In addition, we include surface sensible and latent heat fluxes (SSHF/Lv, normalized to be in the units of temperature and moisture tendencies by dividing by \(c_p\) and \(L_v\), heat capacity of air and latent heat of vaporization, respectively) and surface pressure \(p_s\). To consider the past history or memory of convection, we also include variables in prior GCM time steps as input to ResCu. We consider up to four time steps back. For SPCAM with a 20-min time step, this means that the atmospheric states up to about 1.5 hr prior can affect convection and clouds at the current time step. In addition, the temperature and moisture tendencies from moist physics \(\left( \frac{\partial T}{\partial t} \right)_{mp}\) and \(\left( \frac{\partial q}{\partial t} \right)_{mp}\), cloud water \(q_v\), and cloud ice \(q_i\) as predicted by the CRM from prior four time steps are included as well. Typical convective cells have a lifetime of 30 min to an hour and organized convection can last several hours or longer. Considering the computational resources, we choose to include up to 1.5 hr of convection history. This is admittedly arbitrary and will be examined in sensitivity tests in section 4.1. For the output, ResCu predicts temperature and moisture tendencies from moist physics \(\left( \frac{\partial T}{\partial t} \right)_{mp}\) and \(\left( \frac{\partial q}{\partial t} \right)_{mp}\), cloud water \(q_v\), and cloud ice \(q_i\). Note that for our input variables, we did not use meridional wind and insolation that were used in Rasch et al. (2018). We believe meridional winds are of secondary importance to convection compared to temperature and moisture. While solar insolation is important for convection and the climate system in general, its effects through radiation processes are already accounted for in temperature.

Table 1 lists the input and output variables. The input layer consists of 36 separate vectors with a length of 33 and the output layer consists of four vectors. All input and output variables are normalized to ensure that they are in the same magnitude before they are put into ResCu for training and testing. The normalization factor for each variable is determined by the maximum of its absolute values in the atmospheric column (for profiles) or on the surface (for single-layer variables).

The training procedure of ResCu is a supervised learning, which means ResCu is fed with input and target output pairs from SPCAM’s simulation. During this learning process, these data are fed repeatedly to minimize the loss function via backward propagation. In order to improve the training speed, we selected a total of 800 grid points randomly distributed in three latitude zones in proportion to their relative surface area. The three zones consist of the tropics (30°S to 30°N), midlatitudes (60°S to 30°S and 30°N to 60°N), and high latitudes (90°S to 60°S and 60°S to 90°N). There are 21 million training samples in total. The NN is trained over 100 epochs (an epoch is one complete cycle through the full training data set to be learned by the NN) to ensure that all the training samples have been fully learned without overfitting. There is no overfitting during our 100-epoch training either since our training samples are sufficiently large compared to the number of parameters in the NN (21 million samples vs. 1 million parameters), similar to what Gentile et al. (2018) found in their NN training.

2.3. Moist Static Energy Conservation in the NN

In moist physics processes (convection and cloud condensation), the atmospheric moist static energy is conserved in the absence of freezing. This can be seen from the tendencies of temperature and moisture from convection and condensation:

\[
\left( \frac{\partial T}{\partial t} \right)_{mp} = -\left( \frac{\partial q}{\partial p} \right)_{mp} + L_v (c - e) + L_f c_f
\]

\[
\left( \frac{\partial q}{\partial t} \right)_{mp} = -L_v \frac{\partial q}{\partial p} - L_v (c - e)
\]
where subscript $mp$ stands for moist physics. The terms with primes represent eddy transport (by convec-
tion) of heat and moisture, $c - e$ is condensation minus evaporation from cloud and precipitation pro-
cesses, and $cf$ is the portion of the condensate that freezes. $cp$ is heat capacity of air. $Lv$ is latent heat of
condensation, and $Lf$ is latent heat of freezing. Adding the above two equations, integrating by mass from
the surface to the top of the model and noting that surface turbulent
fluxes are not part of moist physics,
we have the following equation for moist static energy
$h = cpT + LV(q - e) + gz$:

$$
\frac{1}{g} \int \left[ \rho \frac{\partial h}{\partial t} \right] dp = \frac{1}{g} \int \left[ \rho \frac{\partial q}{\partial t} \right] dp + Lv \frac{\partial e}{\partial t} + \frac{Cp}{g} \frac{\partial T}{\partial t} \frac{\partial h}{\partial t} dp
$$

In other words, column-integrated $h$ is conserved during moist processes if freezing is not considered.

In Rasp et al. (2018), while $h$ conservation is not explicitly considered, it is achieved approximately in their
aquaplanet simulation. Here, we customize our loss function to include $h$ conservation in moist physics pro-
cesses by adding the two norm of the difference between column-integrated $h$ change from ResCu and that
from SPCAM in the form of

$$
\| \frac{1}{g} \int \left[ \rho \frac{\partial h}{\partial t} \right] dp - \frac{1}{g} \int \left[ \rho \frac{\partial h}{\partial t} \right] dp \|_2^2
$$

as a penalty term in our loss function to make the
integrated $h$ tendencies from ResCu match those from SPCAM. The loss function is written as

$$
loss = \| \tilde{y} - y \|_2 + \lambda \left\| \frac{1}{g} \int \left[ \rho \frac{\partial h_{SW}}{\partial t} \right] dp - \frac{1}{g} \int \left[ \rho \frac{\partial h_{NN}}{\partial t} \right] dp \right\|_2^2
$$

where $y$ is the target fields from SPCAM, $\tilde{y}$ is the output of ResCu, and $\lambda$ is a Lagrangian multiplier to
enforce $h$ conservation and accuracy simultaneously. Based on a large number of experiments during
the development process of ResCu, we set $\lambda$ to $5 \times 10^{-7}$ for an optimal balance between $h$ conservation.
and prediction accuracy. As will be seen in section 4.2, this makes the penalty term of $h$ conservation account for about 20% of the total loss.

3. Semiprognostic Test of ResCu

In this section, we compare results from ResCu with those from the independent test data of SPCAM simulation. ResCu predicts temperature and moisture tendencies from moist physics processes, cloud water, and ice mixing ratio. Surface precipitation is diagnosed from the vertically integrated moisture tendency, assuming that all condensed water minus evaporation in an atmospheric column rains out to the surface as precipitation. To check on the degree to which $h$ is conserved in ResCu, Figure 2 compares the scatter plots of vertically integrated moisture tendency versus vertically integrated temperature tendency from moist
physics between SPCAM and ResCu (in energy units). The scatter plot of the test results (Figure 2b) is very close to the targets (Figure 2a). The deviation from the diagonal line indicates freezing and melting effects. In both SPCAM and ResCu, when there is large column-integrated condensational heating, the heating exceeds the column-integrated drying, implying additional heating from freezing. Similarly, when there is column-integrated cooling, the cooling exceeds moistening, implying additional cooling from melting. Since radiative energy transfer and PBL turbulent fluxes are calculated outside the CRM part of SPCAM, we should have \( L_n \frac{\partial \theta q}{\partial t} dp + c_p \frac{\partial \theta T}{\partial t} dp = 0 \) when there is no freezing heating if \( h \) is conserved during the moist physics process. The probability density distribution (PDF) of \( L_n \frac{\partial \theta q}{\partial t} dp + c_p \frac{\partial \theta T}{\partial t} dp \) is centered near 0 (0.99 W/m²) in SPCAM, with a spread (standard deviation) of ~10.66 W/m², indicating the contribution of freezing heating and melting cooling to column-integrated \( h \) change. The PDF of ResCu-predicted \( L_n \frac{\partial \theta q}{\partial t} dp + c_p \frac{\partial \theta T}{\partial t} dp \) is very close to that of SPCAM, with a mean of 1.91 W/m² and a standard deviation of 10.15 W/m² (Figures 2c and 2d). However, there is a systematic shift to the positive side by 0.916 W/m². The accuracy of the predicted column-integrated \( h \) can be further seen by the PDF of the instantaneous differences between the target and the prediction (Figure 2e). A root-mean-square error (RMSE) (standard deviation of the difference) of 5.71 W/m² is only about half of the column-integrated \( h \) change from freezing or melting, which indicates a high degree of \( h \) conservation in ResCu.

The predicted annual mean precipitation by ResCu (Figure 3b) agrees remarkably well globally with the SPCAM simulation (Figure 3a), with major precipitation features such as ITCZ, South Pacific Convergence Zone (SPCZ), and the monsoon systems in the tropics and the storm tracks in the
midlatitudes well captured, although there is a slight underestimation over ITCZ, SPCZ, in southeast United States, and over South America and an overestimation over Tibetan Plateau (Figure 3c, note that the color scale is about 10% of that for the mean). In terms of the global mean, there is a small negative bias because negative biases in tropics offset the positive biases in high latitudes.

The diabatic heating and drying produced by convection and large-scale condensation in SPCAM are also reproduced well by ResCu. As an example, Figures 4a and 4b show the latitude-height cross sections of the annual mean heating and drying due to moist physics. The SPCAM shows a deep layer of heating and drying in the tropics, corresponding to deep convection. In the subtropics there is heating and moistening in the lower troposphere, corresponding to shallow convection, stratus, and stratocumulus dominant in the subtropics. In the midlatitudes, there is heating and drying below 400 hPa due to midlatitude cyclones. All these features are very well captured by the ResCu (Figures 4c and 4d). Note that there is strong cooling and drying near the surface. This is an artifact of the SPCAM configuration. In this version of the SPCAM, surface turbulent fluxes and boundary layer mixing are calculated outside the CRM part of the SPCAM. As a result, there is strong PBL forcing as input into the CRM, and in response, the CRM produces strong cooling and drying near the surface to balance the forcing. The differences between ResCu and SPCAM are generally less than 10% in most regions (Figures 4e and 4f) with slightly underestimated heating in tropical midtroposphere and slightly more moistening in tropical and subtropical lower troposphere above the PBL.

Different from other NN-based parameterizations (Brenowitz & Bretherton, 2018; Rasp et al., 2018), since ResCu only represents moist physics, that is, it excludes radiative heating, it also predicts cloud condensates (cloud ice and cloud water) that are used to feed into the CAM5 radiative transfer scheme. Figure 5 shows annual mean height-latitude cross sections of cloud water, cloud ice, and their differences between ResCu and SPCAM. Both cloud water and cloud ice from ResCu are very close to those from SPCAM simulations. The cloud water is concentrated in the middle and lower troposphere, and cloud ice is in the upper

Figure 5. As in Figure 4 but for cloud water (a, c, and e) and cloud ice (b, d, and f), except that the color range for the differences is 10% of that for the means.
troposphere. The errors of ResCu prediction mainly occur in regions of large cloud water and cloud ice values. Nonetheless, they are less than 5%.

In addition to the time mean fields, the predicted temporal variance is another measure of NN-based parameterizations. Gentine et al. (2018) and Rasp et al. (2018) noted that in the lower troposphere their NN-based parameterizations tend to produce a lower variance of heating and moistening compared to SPCAM. To check on this with ResCu, Figure 6 shows the height-latitude cross sections of the temporal variance of ResCu-predicted fields. ResCu reproduces the variance of SPCAM simulation very well. Although an underestimation of the maximum variance for heating and drying is visually noticeable, the global average temporal standard deviations of zonal means for the entire 1-year data are 97% or more of those in the SPCAM simulation. In particular, for cloud water and cloud ice mixing ratios, the underestimation of the global average standard deviations is less than 1%.

In summary, ResCu can accurately reproduce the global distribution of diabatic heating and drying from moist physics, and thus precipitation, in SPCAM. In addition, the predicted cloud water and ice are also very accurate. The temporal variance of these fields in SPCAM is also well captured. Table 2 summarizes the overall performance of ResCu using three metrics: the coefficients of determination ($R^2$), biases, and RMSEs. $R^2$ for the five output variables are very high, varying from 0.877 for moisture tendency to 0.998 for cloud ice. The global mean relative biases are a few percent compared with the SPCAM simulation. The RMSE for each predicted field is in general less than 35% of the standard deviation.
4. Sensitivity Tests

4.1. Convective Memory

As discussed in section 2.2, to account for the effect of convective memory, we included both GCM grid scale and convective fields from prior four time steps. Here we test the impact of including history information from different numbers of previous time steps as measured by the loss function. We perform a set of four sensitivity experiments by varying the number of prior time steps to be included in ResCu from 1 (one time step prior to the current time step) to 4 (four time steps prior), everything else remain unchanged. In all the sensitivity tests here and in the following subsection, the loss function, training data, and training epochs are the same as in the original ResCu. However, to save computational resources, tests are performed only using data from June, July, and August (JJA) of the third year SPCAM simulation, with about 92 million independent data samples.

The test loss, together with contributions from the MSE (the first term in Equation 4) and the penalty for violation of column-integrated $h$ conservation (the second term), is shown in Figure 7a. The normalized loss is close to $0.8 \times 10^{-4}$ for all experiments. Including only one time step memory has $\sim5\%$ higher loss compared to the original ResCu, with contributions from both MSE and $h$ conservation penalty. In comparison, the loss for including two and three prior time steps is within $1\%$ of ResCu. Actually, the loss is smaller by including two and three prior time steps. However, while the accuracy is higher, as indicated by a smaller MSE, the penalty from violation of $h$ conservation is higher compared to the original ResCu. As energy conservation is of greater interest for GCMs, we continue to include four time steps of history to represent convective memory.

Table 2

|                  | $d_{qv}$ | $dT$   | $q_c$  | $q_i$  | Precipitation |
|------------------|----------|--------|--------|--------|---------------|
| $R^2$            | 0.877    | 0.928  | 0.959  | 0.998  | 0.912         |
| Bias             | $-0.024$ (5.7%) | $0.0244$ (8.3%) | 0.129 (2.8%) | 0.040 (4.6%) | 0.017 (0.7%) |
| RMSE             | 1.621 (34.9%) | 1.539 (26.8%) | 4.340 (20.2%) | 0.485 (4.9%) | 2.363 (29.7%) |

Note. $R^2$ is calculated as $1 - \frac{\text{MSE}}{\text{Var}}$, where MSE is the mean squared error between predictions by ResCu and SPCAM targets and Var is the variance of SPCAM targets. The units of bias and RMSE are g/kg/day for $d_{qv}$, K/day for $dT$, mg/kg for $q_c$ and $q_i$, and mm/day for precipitation. The numbers in parentheses for bias and RMSE are relative to SPCAM mean and standard deviation, respectively.

Figure 7. Test loss in hyperparameter and network sensitivity experiments. (a) The total loss (blue bar) and contributions from mean squared error (orange bar) and penalty for violations of column-integrated moist static energy conservation (green bar) versus number of history time slices included for convective memory in ResCu. (b) Same as in (a) but versus different neural network architectures. BN is for batch normalization, No-Act is for no activation in the output layer, CNN-22 is for a 22-layer CNN, DNN-10 and DNN-22 are for 10-layer and 22-layer fully connected network, and ResDNN-22 is for DNN-22 with shortcuts. The tests are done using JJA SPCAM data.
memory in the rest of this study. Nevertheless, for practical purposes including two to three prior time steps should work as well because of the very small loss differences.

4.2. Other NN Architectures

ResCu uses a residual network architecture including convolution layers and shortcuts (skip connections). As shown in section 3, it reproduces the SPCAM simulation very well. In previous studies (Brenowitz & Bretherton, 2018, 2019; Gentine et al., 2018; Rasp et al., 2018), fully connected deep NNs without convolutions are used. Therefore, it is of great interest to compare different network architectures in terms of their performance accuracy. In addition, both convolutions and shortcuts are used in ResCu, what are their individual roles? Furthermore, in ResCu we did not apply batch normalization after each convolution layer; does it have any effect? Finally, as mentioned in section 2.2, in the output layer of our network, we used a hyperbolic tangent activation function \( \tanh \). What if no activation is used in the last year? These questions will be addressed in this section.

The other network architectures to be tested are the fully connected deep NNs (referred to simply as DNNs hereafter) and CNNs. In DNNs, all units/neurons are flattened in a line and globally connected to the previous layer with different weights. On the other hand, in CNNs, units/neurons in a convolution layer are organized in feature vectors sharing the same weights and are sparsely and locally connected to the previous layer. Therefore, for the same number of neurons, the parameters used in a convolutional layer are far fewer than in a fully connected layer.

For the plain CNN, we use 22 layers (CNN-22), the same depth as in ResCu, by eliminating all shortcuts in ResCu but keep the rest unchanged. Thus, the only difference between CNN-22 and ResCu is the shortcuts. The loss together with contributions from the MSE and energy conservation penalty is shown in Figure 7b. It turns out that CNN-22 performs as well as ResCu when measured by the loss. We speculate that a 22-layer ResNet is probably not deep enough for it to take full advantage of shortcuts in improving its accuracy since identity shortcuts only provide faster convergence in a “not overly deep” ResNet (He et al., 2016).

For DNNs, we test two depths: One is 10-layer deep (DNN-10) with 1,024 neurons in each layer, resulting in 9.7 million parameters; and the other is 22-layer deep (DNN-22) with the same number of neurons per layer, resulting in 22 million parameters. We also test DNN-22 by adding shortcuts (ResDNN-22). The inputs to the DNNs (DNN-10, DNN-22, and ResDNN-22) still contain all 36 vectors as in section 2.2 but are flattened into a single vector with \( 36 \times 33 = 1,188 \) elements. When neither convolutions nor shortcuts are used (cf. ResCu with DNN-22 in Figure 7b), the fully connected DNN-22 has extremely large loss (~15 times larger than ResCu, mostly from the MSE). This shows that DNN-22 is too deep for a fully connected NN, a well-known fact in ML. By reducing the depth to 10 layers (DNN-10) as is often used in fully connected NN architecture, the loss is reduced significantly. Even so, the test loss of DNN-10 is still very large compared to ResCu. By adding shortcuts to DNN-22, ResDNN-22 has a much smaller loss than DNN-10 (Figure 7b). Note that while a plain DNN-22 is too deep, adding shortcuts resolves this problem, making it even more accurate than DNN-10. This demonstrates the power of residual NNs when convolutions are not used.

The effect of convolution can be seen by comparing ResCu with ResDNN-22 or comparing CNN-22 with DNN-10. The loss is significantly larger in ResDNN-22 than in ResCu. Without shortcuts, CNN-22 also greatly outperforms DNN-10 (Figure 7b). Thus, with or without shortcuts the use of convolutional layers achieves a much smaller loss, yet with only \((1/10)\) of parameters used in fully connected NNs. These comparisons clearly show that convolutions play an important role in our ResCu.

A possible reason that a CNN works better than a fully connected DNN in emulating the atmospheric moist processes is that the vertical profiles of the thermodynamic fields (temperature and moisture) and the advective forcing by GCM grid-scale circulation often have vertically coherent structures in the atmosphere when moist physical processes occur. For instance, when convection occurs, the atmosphere is often unstable in a deep tropospheric layer. The vertical spatial scale is on the order of a couple of kilometers for relatively shallow convection to over 10 km for deep convection. The coherent local features are well suited for convolutional networks. Another feature of CNNs is translational equivariance, that is, if the input changes the output also changes accordingly. If a local feature in the input is shifted vertically, the output feature of the CNN will also shift in similar ways. This translational equivariance property of CNNs fits our purpose well for emulating the moist physics processes. For example, since convective available potential energy
(CAPE) is a measure of the potential for convection, when CAPE is concentrated in the lower troposphere, the convective heating will also occur in the lower troposphere; if CAPE is concentrated in the middle or upper troposphere, the altitude of convective heating will also move up accordingly. The same applies to advective forcing in temperature and moisture fields.

In addition to using a CNN, the depth of the network also plays an important part in the superior performance of ResCu. We tested networks with depths varying from 4 to 22 layers and noticed that the accuracy increased with depth. This is partly due to the increased ability of the activation function to represent non-linearity with a deeper NN and partly due to the fact that the receptive field of neurons in deeper layers of a CNN is wider than the receptive field of neurons in shallower layers. As the CNN’s depth increases, neurons in a deep layer will have a global receptive field, thereby receiving global information indirectly despite the sparse local connection in a given layer.

We also analyzed the impacts of activation function in the last layer and batch normalization on the performance of ResCu. We removed the activation function tanh in the output layer of ResCu (denoted No activation in Figure 7b). Its performance is comparable (actually slightly better) to ResCu in terms of total loss. It implies the activation function in the output layer in a deep neural network is not essential, probably because there are already enough activations in previous layers to account for the nonlinear representation. In addition, batch normalization (Ioffe & Szegedy, 2015) can be used in deep neural networks, including ResNet, to improve its robustness and accuracy. To test its effect, we add batch normalization after every convolution layer except the last one in ResCu and designate it as BN in Figure 7b. As can be seen, batch normalization can bring the test loss down by about 5% compared to ResCu. However, it increases the training time by up to 50% and makes it more difficult to be ported to Fortran codes for future implementation into GCMs.

In summary, with the above sensitivity tests, we conclude that using convolution layers is an effective way to achieve a better performance in deep learning-based physics parameterization with fewer parameters. Additionally, identity shortcuts are proven essential in super deep neural networks for image classification. Although we did not see a significant benefit of shortcuts in our 22-layer ResCu, we expect it to play a more important role in deeper NNs for moist physics parameterization. Batch normalization can add additional improvements of the performance of ResCu and will be included in the future.

5. SCM Tests

5.1. SCM Prognostic Validation Strategy

Our eventual goal is to implement the trained ResCu into a three-dimensional GCM such as the NCAR CAM5 (Neale et al., 2012). However, before the full implementation, further tests are needed. In section 3, we tested the trained ResCu against independent SPCAM simulation data. In this section, we test it against observations in a SCM framework using the single-column version of CAM5 (SCAM) with a default time step of 20 min. Under the SCM framework, the large-scale dynamics are prescribed by observed advective tendencies of temperature and moisture from field campaign case studies. SCM has been used extensively in the past to test GCM physics parameterizations (Ghan et al., 2000; Song & Zhang, 2011; Xie et al., 2002).

The most straightforward and clean test of model physics parameterization in SCM is by ensuring that the model thermodynamic state is the same as the observed thermodynamic state or as close to it as possible at each time step so that the difference in the thermodynamic state has a minimal impact on the parameterization outcome. This can be achieved by nudging the SCM model temperature and specific humidity to observations. We use a nudging time of 3 hr following Ghan et al. (2000).

In the implementation into SCAM, the forward testing process of ResCu is directly converted to a Fortran module to replace the moist physics processes, which are run before coupling with the land model, including deep convection, shallow convection, cloud macrophysics, and cloud microphysics. A similar validation strategy has also been used in Brenowitz and Bretherton (2018). Since radiative parameterization is excluded in our NN, the CAM5 radiation scheme operates separately after ResCu is called. The cloud water and cloud ice from ResCu are input into the radiation scheme. We selected two field observation cases. One is the summer 1997 Intensive Observation Period (IOP) of the U.S. Department of Energy’s Atmospheric Radiation Measurement (ARM) program (Xie et al., 2002) at the Southern Great Plains (SGP) for midlatitude
continental convection, and the other is the Tropical Warm Pool-International Cloud Experiment (TWP-ICE) in 2006, Darwin, Australia, for tropical maritime summer monsoon convection (May et al., 2008). The ResCu coupled SCAM (RN-SCAM) will be referred to as the experiment run, while the control run is SCAM with conventional parameterizations. We compare the subgrid-scale diabatic heating and drying rates composed of all subgrid physical processes and precipitation from moist physics processes in the control run and the RN-SCAM to the apparent heat source (Q1), apparent moisture sink (Q2; Yanai et al., 1973), and rainfall in observations. Note that some prognostic variables in conventional macrophysics and microphysics parameterization that are important for radiation, such as droplet number concentration and droplet radius, are not provided by ResCu. We use the CAM5 control run values for these as a stopgap.

5.2. ARM Summer 1997 IOP

The ARM summer 1997 IOP was conducted from 19 June to 18 July 1997 by the ARM program. The observations include radiosondes released every 3 hr, measurements from the ground meteorological network, wind profiler, radar, and satellite (Ghan et al., 2000). The advection of temperature and moisture and vertical velocity needed to drive SCAM are derived by constrained variational analysis (M. H. Zhang & Lin, 1997). During the IOP, there are several intense precipitation events and dry and clear days, associated with the large-scale upper-level troughs and ridges over North American continent. The details of observations and convective events are described in Ghan et al. (2000) and Xie et al. (2002). We carried out SCAM simulation for the first 25 days.

Figure 8 shows the observed and simulated precipitation time series. RN-SCAM successfully predicts most precipitation events in the 25-day simulation, which includes continental thunderstorms on 26 June, 29–30 June, and 9 July. While the timing is well captured, the precipitation amount is overestimated. On
the other hand, the control run underestimates these precipitation events and shows large spurious precipitation during nonprecipitation periods after 26 and 30 June. For the corresponding apparent heat source (Q1) and apparent moisture sink (Q2), the RN-SCAM simulation is closer to observations compared to SCAM control run. It shows a deep layer of convective heating and drying during each precipitation event, particularly for events on 26 and 30 June. Q1 from the SCAM simulation is overly concentrated in the upper troposphere (Figure 9) and the simulated Q2 is often underestimated (Figure 10).

5.3. TWP-ICE IOP

TWP-ICE was conducted in Darwin, northern Australia, from 13 January to 13 February 2006. There are four different convective regimes during the TWP-ICE IOP, when an MJO event moved through the region. At the beginning there is a convectively active monsoon period, followed by a suppressed monsoon period, 3 days of clear skies, and then a break period (May et al., 2008). The active monsoon period characterized by intense and systematic upward motion and convection is between 13 and 25 January. During this period, our SCAM simulation is initialized on 17 January and run for 8 days. We conducted two simulations, with and without nudging, and both show similar precipitation time series. Thus, only simulation results without nudging are presented for TWP-ICE to demonstrate the feasibility of incorporating ResCu output into model integration. Technically, the moist physics tendencies for temperature and moisture form ResCu are fed into the model integration the same way as the conventional parameterization schemes. The cloud liquid water and ice content from ResCu are used for model radiative transfer calculations.

RN-SCAM generally captures the heavy monsoon rainfall events well, with some underestimation in intensity except the event on 20 January (Figure 11). The SCAM control run also performs comparably well. The correlation coefficients between RN-SCAM and observations and between RN-SCAM and SCAM control run are 0.73 and 0.69, respectively. Figure 12 shows the time-height cross section of apparent heat source Q1. RN-SCAM reproduces the vertical extent
Figure 12. As in Figure 9 but for TWP-ICE IOP.

Figure 13. As in Figure 10 but for TWP-ICE IOP.
and intensity of most events very well. For example, on 21 January, the observed Q1 is concentrated in the layer between 700 and 300 hPa. This is well captured by RN-SCAM. The time evolution of Q1 for the event on 23/24 January is also well reproduced by RN-SCAM, though with a slightly weaker intensity. Note that the precipitation event on 23/24 January is an organized mesoscale convective system (Xie et al., 2010). Both the deep layer of heating at the beginning and the heating above 600 hPa only in the later stage of the event are captured by RN-SCAM. The control run also performs well, although there is somewhat stronger cooling in the lower troposphere such as in the events on 20 and 23 January. Figure 13 shows the apparent moisture sink Q2. Again, both RN-SCAM and SCAM control runs simulate the evolution of Q2 well. Compared to the ARM SGP97 case, the diurnal cycle in the boundary layer is much weaker, and this difference is captured by both the ML-based parameterization and the conventional parameterization schemes.

As the simulation is not nudged, the temperature and moisture fields will evolve differently from observations. Figure 14 shows the temperature and moisture biases. Large cold biases start to show up beginning on 22 January, especially above 400 hPa, in the RN-SCAM simulation. However, the cold biases start 2 days earlier in SCAM control run. The magnitude of the biases is also larger in the last 2 days. The dry moisture biases are also larger in the SCAM control run in the lower troposphere compared to the RN-SCAM run, although the latter has a slightly larger moist bias in the upper troposphere above 400 hPa.

In summary, ResCu trained using simulations in SPCAM can reproduce the observed precipitation events reasonably well in SCAM for both midlatitude continental convection and tropical monsoon convection. The vertical extent and time evolution of convective heating and drying are also captured by the NN-based parameterization. The performance is comparable to that of conventional parameterization of convection and clouds. In other words, ResCu can successfully capture the physics relationships behind the various types of convection produced in the training data.

6. Conclusions and Discussion

A deep residual convolutional neural network (ResCu) is used for developing a ML-based parameterization scheme for moist physics processes. It is trained with output every 20 min from a 1-year global climate simulation by SPCAM with real-world orography. The input variables to the NN include thermodynamic profiles and large-scale advective forcing of temperature and moisture. In addition, the memory or history of past convection and clouds is also considered by including large-scale thermodynamic states and advective forcing, as well as cloud liquid and ice water content and temperature and moisture tendencies in previous model time steps. The output of the NN is temperature and moisture tendencies due to moist physics,
cloud liquid, and ice water content for use in radiative transfer calculation in the host model. Precipitation is derived from column-integrated moisture tendency. The column-integrated moist static energy conservation is reinforced by including it as a penalty term in the design of loss function.

The parameterization, ResCu, is trained with more than 21 million data points covering the globe over a year. Thus, all dynamic regimes in the current climate are sampled properly in the training data. ResCu is tested against an independent 1-year SPCAM simulation diagnostically. It reproduces the annual mean fields as well as variance of interested climate variables very well. We further implemented it in SCAM and compared its simulations with standard SCAM control runs and observations in ARM SGP summer 1997 and TWP-ICE IOPs. The simulations using ResCu over the two periods in general reproduce the observed time series of precipitation events and diabatic heating and drying from moist physics processes. Furthermore, in the TWP-ICE simulation without nudging, the temperature and moisture biases are smaller than in the control run in which conventional convection and cloud parameterization schemes are used.

Most of the previous studies (e.g., O’Gorman & Dwyer, 2018; Rasp et al., 2018) are based on fully connected NNs or random forest and trained on aquaplanet simulation data. Our work is the first attempt to use a ResNet architecture for convection and cloud parameterization. ResNet outperforms fully connected NNs in accuracy and parameter utilization efficiency. This success can be attributed to the use of convolutional layers and identity shortcuts. The sparse connection and parameter sharing gives CNNs an edge in processing model moist physics profiles, which possess strong local features and translational equivariance. In addition, deep CNNs have an advantage of receptive field expansion, which composes lower-level features into higher-level ones in terms of scale and complexity. The comparison of ResNet with other network architectures, that is, CNN without identity shortcuts and fully connected network, shows that the use of convolutional layers plays an important role in achieving the accuracy of ResCu, whereas the potential of identity shortcuts is not fully realized with a 22-layer network. Deeper networks will benefit more from it. We also tested the sensitivity of the network’s performance to inclusion of past history in its input. Including convective memory for up to an hour can reduce the mean squared error the accuracy by about 5%.

Our NN is designed to represent moist physics processes only. Some other NN-based parameterizations (Brenowitz & Bretherton, 2018; Gentile et al., 2018; Rasp et al., 2018) chose to also include radiation and PBL physics. In our opinion, focusing on moist physics parameterization is more appealing for at least two reasons. First, moist physics (convection and clouds) parameterization is the largest source of uncertainty in climate simulation and future projection. Thus, it is desirable to keep the problem simple but not simpler. Second, aerosols are important in atmospheric radiation; without interactive aerosol information in the training data, it will not be possible to develop a NN-based parameterization that can realistically represent aerosol-radiation interaction. Cloud properties from moist physics strongly affect atmospheric radiation. Our ResCu predicts cloud water and ice mixing ratio besides diabatic heating and drying rate. However, other cloud properties such as droplet number concentration and droplet radius needed in radiation calculation are not included. We use the CAM5 control run values for these as a stopgap. More work is needed to address this in the future.

Data Availability Statement
The training and testing data, associated codes, and validations in SCAM are provided at Zenodo (https://doi.org/10.5281/zenodo.3908503). Additional data for the 2 years of SPCAM simulation with output every time step are available at https://doi.org/10.6075/J0CZ35PP and https://doi.org/10.6075/J03J3BGF.

References
Amber, U., Wang, S., & Sobel, A. (2016). Response of atmospheric convection to vertical wind shear: Cloud-system-resolving simulations with parameterized large-scale circulation. Part II: Effect of interactive radiation. Journal of the Atmospheric Sciences, 73(1), 199–209. https://doi.org/10.1175/JAS-D-15-0151.1
Arakawa, A. (2004). The cumulus parameterization problem: Past, present, and future. Journal of Climate, 17(13), 2493–2525. https://doi.org/10.1175/1520-0442%282004%2917%3C2493%3APCP%3E2.0.CO%3B2
Arakawa, A., & Schubert, W. H. (1974). Interaction of a cumulus cloud ensemble with the large-scale environment. Part I. Journal of the Atmospheric Sciences, 31(3), 674–701. https://doi.org/10.1175/1520-0469%281974%2931%3C674%3AOACE%3E2.0.CO%3B2
Beucker, T., Rasp, S., Pritchard, M., & Gentile, P. (2019). Achieving conservation of energy in neural network emulators for climate modeling. arXiv preprint arXiv:1906.06622.
Torri, G., & Kuang, Z. (2016). Rain evaporation and moist patches in tropical boundary layers. *Geophysical Research Letters*, 43, 9895–9902. https://doi.org/10.1002/2016GL070893

Wielicki, B. A., Cess, R. D., King, M. D., Randall, D. A., & Harrison, E. F. (1995). Mission to planet Earth: Role of clouds and radiation in climate. *Bulletin of the American Meteorological Society*, 76(11), 2125–2153. https://doi.org/10.1175/1520-0477%281995%29076%3C2125%3AMTPERO%3E2.0.CO%3B2

Wyant, M. C., Khairoutdinov, M., & Bretherton, C. S. (2006). Climate sensitivity and cloud response of a GCM with a superparameterization. *Geophysical Research Letters*, 33, L06714. https://doi.org/10.1029/2005GL025464

Xie, S., Hume, T., Jakob, C., Klein, S. A., McCoy, R. B., & Zhang, M. (2010). Observed large-scale structures and diabatic heating and drying profiles during TWP-ICE. *Journal of Climate*, 23(1), 55–79. https://doi.org/10.1175/2009JCLI3071.1

Xie, S., Xu, K.-M., Cederwall, R. T., Bechtold, P., Genio, A. D. D., Klein, S. A., et al. (2002). Intercomparison and evaluation of cumulus parametrizations under summertime midlatitude continental conditions. *Quarterly Journal of the Royal Meteorological Society*, 128(582), 1095–1135. https://doi.org/10.1256/003590002320373229

Yanai, M., Esbensen, S., & Chu, J.-H. (1973). Determination of bulk properties of tropical cloud clusters from large-scale heat and moisture budgets. *Journal of the Atmospheric Sciences*, 30(4), 611–627. https://doi.org/10.1175/1520-0469%281973%29076%3C0611%3ADOBPOT%3E2.0.CO%3B2

Yuval, J., & O’Gorman, P. A. (2020). Use of machine learning to improve simulations of climate. arXiv preprint arXiv:2001.03151

Zhang, G. J., & McFarlane, N. A. (1995). Sensitivity of climate simulations to the parameterization of cumulus convection in the Canadian climate centre general circulation model. *Atmosphere-Ocean*, 33(3), 407–446. https://doi.org/10.1080/07055900.1995.9649539

Zhang, G. J., Song, X., & Wang, Y. (2019). The double ITCZ syndrome in GCMs: A coupled feedback problem among convection, clouds, atmospheric and ocean circulations. *Atmospheric Research*, 229, 255–268. https://doi.org/10.1016/j.atmosres.2019.06.023

Zhang, M. H., & Lin, J. L. (1997). Constrained variational analysis of sounding data based on column-integrated budgets of mass, heat, moisture, and momentum: Approach and application to ARM measurements. *Journal of the Atmospheric Sciences*, 54(11), 1503–1524. https://doi.org/10.1175/1520-0469%281997%29054%3C1503%3ACVAOSD%3E2.0.CO%3B2