Geophysical Research Letters

RESEARCH LETTER
10.1029/2020GL091363

Use of Neural Networks for Stable, Accurate and Physically Consistent Parameterization of Subgrid Atmospheric Processes With Good Performance at Reduced Precision

Janni Yuval[1], **Paul A. O'Gorman**[1], **and Chris N. Hill**[1]

[1]Department of Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA

Key Points:

- Neural-network parameterization gives stable simulations that replicate climate of idealized simulation of atmosphere at high resolution
- Separate predictions of the effect of each subgrid process allows physical constraints to be incorporated into the parameterization
- Parameterization with reduced numerical precision can decrease computational demands without affecting the simulated climate

Supporting Information:

- Supporting Information S1

Correspondence to:

J. Yuval,
yaniyuval@gmail.com

Citation:

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. *Geophysical Research Letters*, *48*, e2020GL091363. https://doi.org/10.1029/2020GL091363

Received 22 OCT 2020
Accepted 23 JAN 2021

Abstract

A promising approach to improve climate-model simulations is to replace traditional subgrid parameterizations based on simplified physical models by machine learning algorithms that are data-driven. However, neural networks (NNs) often lead to instabilities and climate drift when coupled to an atmospheric model. Here, we learn an NN parameterization from a high-resolution atmospheric simulation in an idealized domain by accurately calculating subgrid terms through coarse graining. The NN parameterization has a structure that ensures physical constraints are respected, such as by predicting subgrid fluxes instead of tendencies. The NN parameterization leads to stable simulations that replicate the climate of the high-resolution simulation with similar accuracy to a successful random-forest parameterization while needing far less memory. We find that the simulations are stable for different horizontal resolutions and a variety of NN architectures, and that an NN with substantially reduced numerical precision could decrease computational costs without affecting the quality of simulations.

Plain Language Summary

Due to computational resource limits, small-scale processes, such as clouds and convection, are not explicitly included in state-of-the-art global climate models. Yet, these processes are crucial for accurate predictions of climate, and therefore the effect of these processes on the climate system needs to be estimated using parameterizations. Traditional parameterizations usually rely on simplified models and suffer from inaccuracies that lead to large uncertainties in climate projections. One alternative to traditional parameterizations is to use machine learning to learn new parameterizations which are data driven. However, neural-network parameterizations often lead to instability when implemented in an atmospheric model. Here, we learn a neural-network parameterization that incorporates physical constraints and show that when this parameterization is coupled to an atmospheric model it leads to stable simulations that can mimic the climate of a high-resolution simulation. Furthermore, we show that a neural-network parameterization with reduced numerical precision has the potential to reduce the computational cost of running simulations without having a large impact on the quality of simulations. Overall, our results provide a step forward toward a stable, accurate, and computationally affordable neural-network parameterization of atmospheric subgrid processes for use in simulations of climate.

1. Introduction

Traditional parameterizations in general circulation models (GCMs) rely on simplified physical models and suffer from inaccuracies which lead to model biases and large uncertainties in climate projections (Bony & Dufresne, 2005; Oueslati & Bellon, 2015; O'Gorman, 2012; Schneider et al., 2017; Wilcox & Donner, 2007). One alternative to traditional parameterizations is to use machine learning (ML) algorithms to create new subgrid parameterizations (Bolton & Zanna, 2019; Brenowitz & Bretherton, 2018, 2019; Gentine et al., 2018; Han et al., 2020; Krasnopolsky et al., 2013; O'Gorman & Dwyer, 2018; Rasp et al., 2018; Yuval & O'Gorman, 2020; Zanna & Bolton, 2020).

Two ML algorithms that have been used for climate-model parameterizations are neural networks (NNs) and random forests (RFs). An RF is an ensemble learning algorithm that is composed of several decision trees (Breiman, 2001). O'Gorman and Dwyer (2018) trained an RF to emulate a conventional convection scheme of an atmospheric GCM and showed that when this RF parameterization is implemented in the GCM it leads to stable simulations that reproduce its climate, and the use of an RF allowed physical constraints, such as energy conservation and non-negative surface precipitation, to be respected. More recently, Yuval and O'Gorman (2020) (hereinafter referred to as YOG20) learned an RF parameterization from output of a three-dimensional high-resolution idealized atmospheric model. The parameterization led to stable simulations that replicate the climate of the high-resolution simulations at several coarse resolutions.

Despite the success of RFs in these cases, NNs have some computational advantages over RFs such as the possibility of greater accuracy and needing substantially less memory when implemented. Furthermore, an NN parameterization could potentially be implemented at reduced precision in graphics processing units (GPUs), tensor processing units (TPUs), and even in central processing units (CPUs; Vanhoucke et al., 2011) leading to very efficient parameterizations. For example, modern hardware developments that use half precision (16-bit) arithmetic on recent generation NVidia A100 GPU devices (Nvidia, 2020) allow up to 64 times the peak compute performance, compared to 64-bit arithmetic. Although this is based on theoretical peak numbers, a first exascale application in Earth science has already been realized for image classification based on GPU technology with 16-bit arithmethic (Kurth et al., 2018). Google TPUv3 devices (Jouppi et al., 2020) have similarly optimized capability, for reduced precision floating point arithmetic to support deep learning applications. Furthermore, previous studies have found that using deep NNs with half-precision multipliers has little effect on accuracy when used for classification (Courbariaux et al., 2014; Das et al., 2018; Micikevicius et al., 2017; Miyashita et al., 2016). For regression tasks, which are needed for parameterizations, a reduced-precision NN might not be as accurate as a full-precision NN, but it has been argued that the dynamics at small horizontal length scales represented by parameterizations does not need the same level of precision as for large-scale dynamics because of the unpredictable and stochastic nature of the small-scale flow (Palmer, 2014). Indeed, use of reduced precision in parts of weather or climate models has already been proposed as a way to increase the speed of simulations and reduce their energy cost (Düben & Palmer, 2014; Düben et al., 2014, 2017; Hatfield et al., 2019; Saffin et al., 2020).

NN parameterizations have shown considerable potential, but they have also suffered from instability and climate drift when used in GCMs. For example, Rasp et al. (2018) developed NN parameterizations based on the super parametrized community atmosphere model (SPCAM) and found they are often unstable when coupled dynamically to a GCM. Rasp et al. (2018) were able to acheive a stable simulation with an NN parameterization, but they found that small changes to the configuration led to blow ups in the simulations (Rasp, 2020). Furthermore, when they quadrupled the number of embedded cloud-resolving columns (from 8 to 32) within each coarse-grid cell of SPCAM they found that instabilities returned (Brenowitz et al., 2020). Brenowitz and Bretherton (2018, 2019) learned an NN parameterization in a near-global cloud system resolving model (CRM) and were able to deal with instabilities by removing the upper-atmospheric humidity and temperature from the input space and by using a training cost function that takes into account the predictions from several forward time steps. Although these changes in the learning structure led to stable simulations at coarse resolution with the NN parameterization, the climates of these simulations drifted on longer time scales and were not accurate. Brenowitz et al. (2020) found using linear stability analysis of NN predictions coupled to simplified dynamics that instability occurs when GCMs are coupled to NNs that support unstable gravity waves with certain phase speeds. A study by Ott et al. (2020) tested the stability of simulations coupled to more than a hundred different NNs and found a correlation between offline performance (i.e., the quality of predictions from NNs when they are not coupled to a GCM) and how long simulations run before they blow up, with some accurate NNs leading to fully stable simulations. These results suggest that an exhaustive hyperparameter tuning might be necessary in order to achieve stability in GCM simulations that are coupled to NNs.

RFs might be more stable than NNs since their predictions for any given input are averages over samples in the training data set, and thus they can automatically satisfy physical constraints such as linear energy conservation (O'Gorman & Dwyer, 2018, YOG20) and make conservative predictions for samples outside of the training data (NNs can also be forced to satisfy analytic constraints [Beucler et al., 2019], but such NNs have not yet been coupled to a GCM). However, the RFs and NNs in the studies mentioned above used different training data sets and processed the high-resolution data differently to calculate subgrid terms. Therefore, it is difficult to determine if the stability arises from the different processing of the high-resolution model output or due to the different properties of RFs compared to NNs. The two main differences in the data processing between YOG20 and the NNs studies mentioned above are that (1) the subgrid corrections were calculated accurately for the instantaneous atmospheric state (YOG20) rather than approximating them based on differences over 3-h periods (Brenowitz & Bretherton, 2018, 2019) or predicting the integrated effect of subgrid processes over a 30 min time step (Rasp et al., 2018), and that (2) the subgrid corrections were calculated independently for each physical process (YOG20) rather than for the combined effect of several processes (Brenowitz & Bretherton, 2018, 2019; Rasp et al., 2018).

Here, we learn an NN parameterization with a physically consistent structure using the same accurate data processing that was used to learn an RF parameterization in YOG20. We show that implementing this parameterization in the same model at coarse resolution leads to stable simulations with a climatology that resembles the one obtained from a high-resolution simulation, and we compare the performance of an NN parameterization to the performance of an RF parameterization. We also test how the simulated climate is affected when reducing the precision of the inputs and outputs of the NN parameterization.

We first describe the high-resolution simulation from which the training data was calculated and how this data was coarse-grained (Section 2). We then describe the structure of the NN parameterization and explain how this structure ensures that NN parameterization is consistent with several physical properties (Section 3). We compare simulations using the NN parameterization to the high-resolution simulation and to a simulation with an RF parameterization, and we investigate the dependence of climate on the numerical precision of the NN parameterization (Section 4). Finally, we give our conclusions (Section 5).

2. Methods

2.1. Simulations

All simulations were run on a quasi-global aquaplanet configured as an equatorial beta-plane using the System for Atmospheric Modeling (SAM) version 6.3 (Khairoutdinov & Randall, 2003). The domain has zonal width of 6, 912 km and meridional extent of 17, 280 km. The distribution of sea surface temperature (SST) is specified to be the qobs SST distribution (Neale & Hoskins, 2000), which is zonally and hemispherically symmetric and reaches its maximal value of 300.15 K at the equator and decreases to 273.15 K at the poleward boundaries. There are 48 vertical levels. We use hypohydrostatic rescaling of the equations of motion with a scaling factor of 4, which increases the horizontal length scale of convection and allows us to use a horizontal grid spacing of 12 km for the high-resolution simulation (referred to as hi-res) while resolving deep convection and simulating planetary scale circulations in the same quasi global simulation (Boos et al., 2016; Fedorov et al., 2019; Garner et al., 2007; Kuang et al., 2005; Pauluis & Garner, 2006). The hi-res simulation is the same simulation that was used for training in YOG20. Further details of the model configuration are given in YOG20.

In addition to hi-res, we also consider simulations at horizontal grid spacings of 96 and 192 km, which will be referred to as x8 and x16, respectively, since they correspond to coarser grid spacings by factors of 8 and 16, respectively. We ran a simulation at 96 km horizontal grid spacing without an NN parameterization (x8), several simulations at 96 km horizontal grid spacing with an NN parameterization (x8-NN, variants of this simulation are described in the text below), and simulations at 192 km horizontal grid spacing with (x16-NN) and without (x16) an NN parameterization. All simulations were run for 600 days. The first 100 days are considered as spin up, and the presented results are taken from the last 500 days of each simulation. Simulations with the NN parameterization start with initial conditions taken from the last time step of the simulations without the NN parameterization at the same resolution.

2.2. Coarse-Graining and Calculation of Subgrid Terms

The NN parameterization aims to account for unresolved processes that act in the vertical and affect thermodynamic and moisture prognostic variables. There are three thermodynamic and moisture prognostic variables in SAM (Khairoutdinov & Randall, 2003): liquid-ice static energy h_L, total nonprecipitating water mixing ratio q_T, and precipitating water mixing ratio q_p. Since q_p is a variable that varies on short time scales that are not typically resolved in climate models, we do not include q_p in the NN parameterization scheme following the "alternative" parameterization approach in YOG20. Consequently, we reformulate the equations of motion and define a different thermodynamic energy variable (H_L) that does not include the energy associated with precipitating water (Text S1).

For each 3-hourly snapshot from hi-res, we coarse grain the prognostic variables (u, v, w, H_L, q_T, q_p, where u, v, w are the zonal, meridional, and vertical wind, respectively), vertical advective fluxes, sedimentation fluxes, surface turbulent fluxes, tendency of nonprecipitating water mixing ratio due to cloud microphysics, turbulent diffusivity, radiative heating and temperature. Coarse-graining is performed by a simple spatial averaging as in YOG20 to horizontal grid spacings of 96 km (x8) and 192 km (x16).

We define the resolved flux of a given variable as the flux calculated using the dynamics and physics of SAM with the coarse-grained prognostic variables as inputs. The flux due to unresolved (subgrid) physical processes is calculated as the difference between the coarse-grained flux and the resolved flux. For example, the subgrid flux of H_L due to vertical advection is calculated as

$$\left(H_L\right)_{\text{adv}}^{\text{subg-flux}} = \rho_0\left(\overline{wH_L} - \overline{w}\overline{H}_L\right), \tag{1}$$

where overbars denoted coarse-grained variables and $\rho_0(z)$ is the reference density profile. For each high-resolution snapshot, the coarse-grained prognostic fields are used to calculate the resolved vertical advective, sedimentation, and surface turbulent fluxes. The subgrid fluxes are then calculated by taking the difference between the coarse-grained and resolved fluxes.

3. Neural Network Parameterization

3.1. Parameterization Structure

The structure of the NN parameterization is broadly similar to the RF parameterization used in YOG20 except that where possible we predict fluxes and sources and sinks (rather than net tendencies in YOG20) in order to incorporate physical constraints into the NN parameterization (Section 3.2). By contrast, for the RF parameterization in YOG20, it was sufficient to just predict net tendencies because the RF predicted averages over subsamples of the training data set and thus automatically respected physical constraints such as energy conservation.

The NN parameterization is composed of two different NNs (Figure 1). The first NN, referred to as NN1, predicts the vertical profiles of the subgrid vertical advective fluxes of H_L ($(H_L)_{\text{adv}}^{\text{subg-flux}}$) and q_T ($(q_T)_{\text{adv}}^{\text{subg-flux}}$), the subgrid flux due to cloud ice sedimentation ($(q_T)_{\text{sed}}^{\text{subg-flux}}$), the coarse-grained tendency of q_T due to cloud microphysics ($(q_T)_{\text{mic}}^{\text{tend}}$), and the sum of the total radiative heating and the heating from phase changes of precipitation ($(H_L)_{\text{rad-phase}}^{\text{tend}}$, Text S1). Thus the outputs of NN1 are

$$Y_{\text{NN1}} = ((H_L)_{\text{adv}}^{\text{subg-flux}}, (q_T)_{\text{adv}}^{\text{subg-flux}}, (q_T)_{\text{sed}}^{\text{subg-flux}}, (q_T)_{\text{mic}}^{\text{tend}}, (H_L)_{\text{rad-phase}}^{\text{tend}}), \tag{2}$$

where the superscript subg-flux refers to subgrid fluxes and the superscript tend refers to the total tendency due to a process. The tendencies due to cloud microphysics, radiative heating and heating from phase changes of precipitation are treated as entirely subgrid. In the case of cloud microphysics and phase changes of precipitation, this is because it is not possible to calculate the resolved values of these processes when q_p is not used as a prognostic variable in the simulations (Text S1). Tendencies are predicted at the lowest 30 "full" model levels (below $z = 13.9$ km), while subgrid ice sedimentation fluxes are predicted at the lowest 30 "half" model levels, and vertical advective fluxes are predicted at the 29 "half" model levels above the surface (since advective fluxes are zero at the surface over ocean). Thus, NN1 has $29 \times 2 + 30 \times 3 = 148$ outputs. We do not use NN1 to predict outputs for levels above 13.9 km (≈134 hPa) since the NN parameterization is not accurate at these levels, we want to avoid predicting near the sponge layer which is active at heights above 20 km, and the predicted tendencies and subgrid fluxes are small above 13.9 km with the

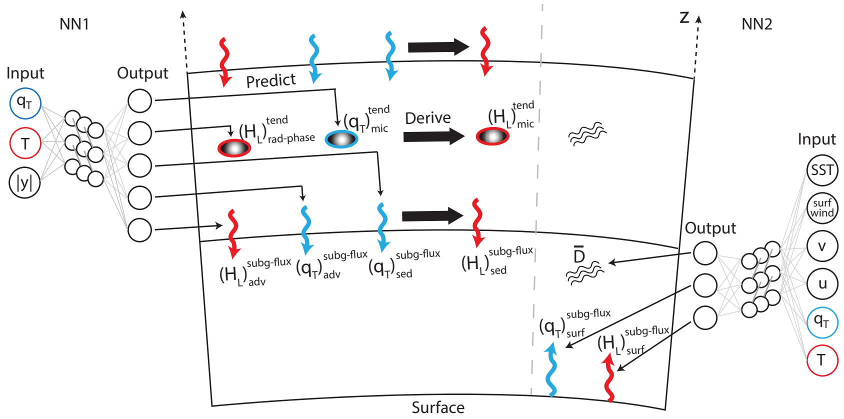

NN1
Input
Output
q_T
T
|y|
Predict
$(H_L)^{tend}_{rad\text{-}phase}$
$(q_T)^{tend}_{mic}$
Derive
$(H_L)^{tend}_{mic}$
$(H_L)^{subg\text{-}flux}_{adv}$
$(q_T)^{subg\text{-}flux}_{adv}$
$(q_T)^{subg\text{-}flux}_{sed}$
$(H_L)^{subg\text{-}flux}_{sed}$
z
NN2
Input
SST
surf wind
v
u
q_T
T
Output
$\bar{D}$
$(q_T)^{subg\text{-}flux}_{surf}$
$(H_L)^{subg\text{-}flux}_{surf}$
Surface

Figure 1. Illustration of the parameterization structure. There are two different NNs included in the parameterization, each having its own inputs and outputs (full description in Section 3.1). Arrows indicate subgrid fluxes due to vertical advection and sedimentation, ellipses indicate tendencies associated with sources/sinks due to cloud microphysics, radiation and phase changes of precipitating water, and the wavy pattern indicates the coarse-grained diffusivity ($\bar{D}$) which is predicted only at the lower 15 levels of the model. Blue (red) color indicates a variable, tendency or flux associated with moisture (energy). NN, neural network.

exception of radiative heating. Above 13.9 km the subgrid fluxes and microphysical tendency are set to zero and the radiative heating tendencies calculated at coarse resolution from SAM are used.

The inputs for NN1 are the resolved vertical profiles of q_T and temperature (*T; the temperature is diagnosed from the coarse-grained prognostic variables*) at the lowest 30 full levels, as well as the distance to the equator (|*y*|, which is a proxy for insolation and surface albedo as both are only a function of |*y*| in our simulations), giving $30 \times 2 + 1 = 61$ inputs. We verified that using top of atmosphere insolation as input instead of the distance to equator does not change the results presented in this study (Figure S1). We found that using all 48 levels of T and q_T as inputs in NN1 leads to an instability, possibly related to instabilities found in previous studies when q_p is not used a prognostic variable (Brenowitz & Bretherton, 2019; Brenowitz et al., 2020, YOG20). However, while Brenowitz and Bretherton (2019) and Brenowitz et al. (2020) removed some inputs above the mid-troposphere to achieve stability, we find it is sufficient to not use inputs in the stratosphere.

The second NN, referred to as NN2, predicts subgrid surface turbulent fluxes of H_L and q_T ($(H_L)^{subg-flux}_{surf}$ and $(q_T)^{subg-flux}_{surf}$, respectively) and the coarse-grained vertical turbulent diffusivity ($\bar{D}$) that is used for H_L and q_T. We only predict $\bar{D}$ in the lower troposphere (the 15 model levels below 5.7 km) because it decreases in magnitude with height (YOG20), and the diffusivity calculated at coarse resolution from SAM is used above 5.7 km. Hence the outputs of NN2 are

$$Y_{NN2} = (\bar{D}, (H_L)^{subg-flux}_{surf}, (q_T)^{subg-flux}_{surf}), \tag{3}$$

giving $15 + 1 + 1 = 17$ outputs. The inputs of NN2 are chosen to be the lower tropospheric vertical profiles of T, q_T, u, v, surface wind speed (wind_{surf}) and SST, so that $X_{NN2} = (T, q_T, u, v, \text{wind}_{surf}, SST)$, giving $4 \times 15 + 1 + 1 = 62$ features, where v in the southern hemisphere is multiplied by -1 when used as an input (see further discussion in YOG20).

The tendencies due to subgrid vertical advection, sedimentation and surface turbulence are calculated online (i.e., when running SAM with the parameterization) from the predicted subgrid fluxes. For physical consistency, the subgrid energy flux due to ice sedimentation is also calculated online as

$$(H_L)^{subg-flux}_{sed} = -L_s (q_T)^{subg-flux}_{sed}, \tag{4}$$

where L_s is the latent heat of sublimation. Similarly, the tendency of energy due to cloud microphysics is calculated online as

$$(H_{\mathrm{L}})^{\mathrm{tend}}_{\mathrm{mic}} = -L_{\mathrm{p}}(q_{\mathrm{T}})^{\mathrm{tend}}_{\mathrm{mic}}, \tag{5}$$

where $L_{\mathrm{p}} = L_{\mathrm{c}} + L_{\mathrm{f}}(1 - \omega_{\mathrm{p}})$ is the effective latent heat associated with precipitation (L_{c} and L_{f} are the latent heat of condensation and fusion, respectively, and ω_{p} is the precipitation partition function determining the ratio between ice and liquid phases).

The results presented in the main paper are for NNs (both NN1 and NN2) with five layers (four hidden layers) with 128 nodes and rectified linear unit (ReLu) activation functions. Results for different NN architectures are shown in the supplementary information (Figure S2). More details about the train and test data sets, the NNs training protocol and how inputs and outputs are rescaled can be found in the supplementary information (Text S2)

When running simulations with the NN parameterization, we limit the predictions of q_{T} fluxes and tendencies such that we prevent q_{T} from being assigned with negative values. More specifically, if the NN parameterization predicts any flux or any tendency at any vertical level that would lead to negative q_{T} values if it acted on its own, we reduce this tendency or flux magnitude such that the minimal value that can be assigned to q_{T} is zero. Removing this constraint (and allowing q_{T} to be assigned with negative values) leads to a substantially different climate when the NN parameterization is used, although the simulations remain stable.

3.2. Physical Consistency of the Parameterization

Previous studies that used NN parameterizations usually predicted the sum of tendencies due to several different processes as a single output (Brenowitz & Bretherton, 2018, 2019; Gentine et al., 2018; Rasp et al., 2018). The coarse-graining and calculation of subgrid terms that is used in this study (Section 2.2) enables us to predict the effect of each process on the prognostic variables separately (Section 3.1), and where possible, the effect is diagnosed from other predicted outputs (Equations 4 and 5). Furthermore, the NN parameterization predicts fluxes instead of tendencies where possible. These differences make the NN parameterization presented in this study physically consistent in the following ways:

1. Predicting the subgrid fluxes due to vertical advection instead of the subgrid tendencies guarantees energy and water are conserved by these fluxes. Similarly, predicting the flux due to sedimentation guarantees that changes in the energy of the atmospheric column due to sedimentation are only due to sedimentation that reaches the surface
2. Changes in energy due to cloud microphysics and ice sedimentation are not predicted by the NN, but are instead diagnosed from Equations 4 and 5 (Figure 1). Diagnosing these changes guarantees that the sources or sinks of energy at each grid point due to cloud microphysics and sedimentation are consistent with the amount of moisture that was subtracted or added at that grid point
3. The NN predicts the coarse-grained vertical diffusivity (instead of predicting subgrid diffusive tendencies or fluxes) which ensures that diffusive fluxes are downgradient and conserve energy and water in each atmospheric column
4. The precipitation is diagnosed from the NN outputs (Text S1). Diagnosing the precipitation was not done in some previous studies that used NN parameterization in which the NN was used to predict precipitation directly (Rasp et al., 2018) or the NN outputs could only be used to predict the difference between precipitation minus evaporation (Brenowitz & Bretherton, 2019)

Properties a, b, and c ensure that the NN parameterization exactly conserves energy in the sense that the column integrated frozen moist static energy is conserved in the absence of radiative heating, heating from phase changes of precipitation, surface turbulent fluxes, and conversion of condensate to graupel or snow (see Text S3).

4. Results

4.1. Simulation with Neural-Network Parameterization

To assess the NN parameterization, we compare the climates of x8, x8-NN (using NNs with five fully connected layers, Text S2) and hi-res. We focus on the zonal- and time-mean precipitation distribution (Figure 2a) and the frequency distribution of the 3-h precipitation rate (Figure 2b). The frequency distribution is known to be sensitive to subgrid parameterization of moist convection (Wilcox & Donner, 2007), and the latitudinal distribution of mean precipitation is especially sensitive to subgrid parameterizations in the zonally and hemispherically symmetric aquaplanet configuration used here (Möbis & Stevens, 2012). The frequency distribution is estimated using 34 bins that are equally spaced in the logarithm of precipitation rate, where the lowest bin starts at 1 mm day^{-1}, and the distribution is normalized such that it integrates to one when considering the whole distribution (including values lower than 1 mm day^{-1}). The hi-res and x8-NN simulations exhibit a similar double ITCZ structure, both in amplitude and in the latitudinal structure (Figure 2a). Note that the presence of a double ITCZ in hi-res is likely to be dependent on the exact domain and SST distribution. In contrast, the x8 simulation exhibits a single ITCZ (Figure 2a), highlighting the sensitivity of the tropical circulation to changes in the horizontal resolution and to the inclusion of the NN parameterization. Also the frequency distributions of precipitation in x8-NN closely matches that of hi-res ($R^2 = 0.99$ as measured across bins), although x8-NN overestimates the extreme events, while the frequency distribution in x8 differ substantially from the distribution of hi-res ($R^2 = 0.35$) especially for weak and extreme precipitation events (Figure 2b). The NN parameterization also brings the zonal- and time-means of the zonal wind, meridional wind, eddy kinetic, and nonprecipitating water closer to the hi-res climatology, and outperforms the x8 simulation for these variables (Table S1).

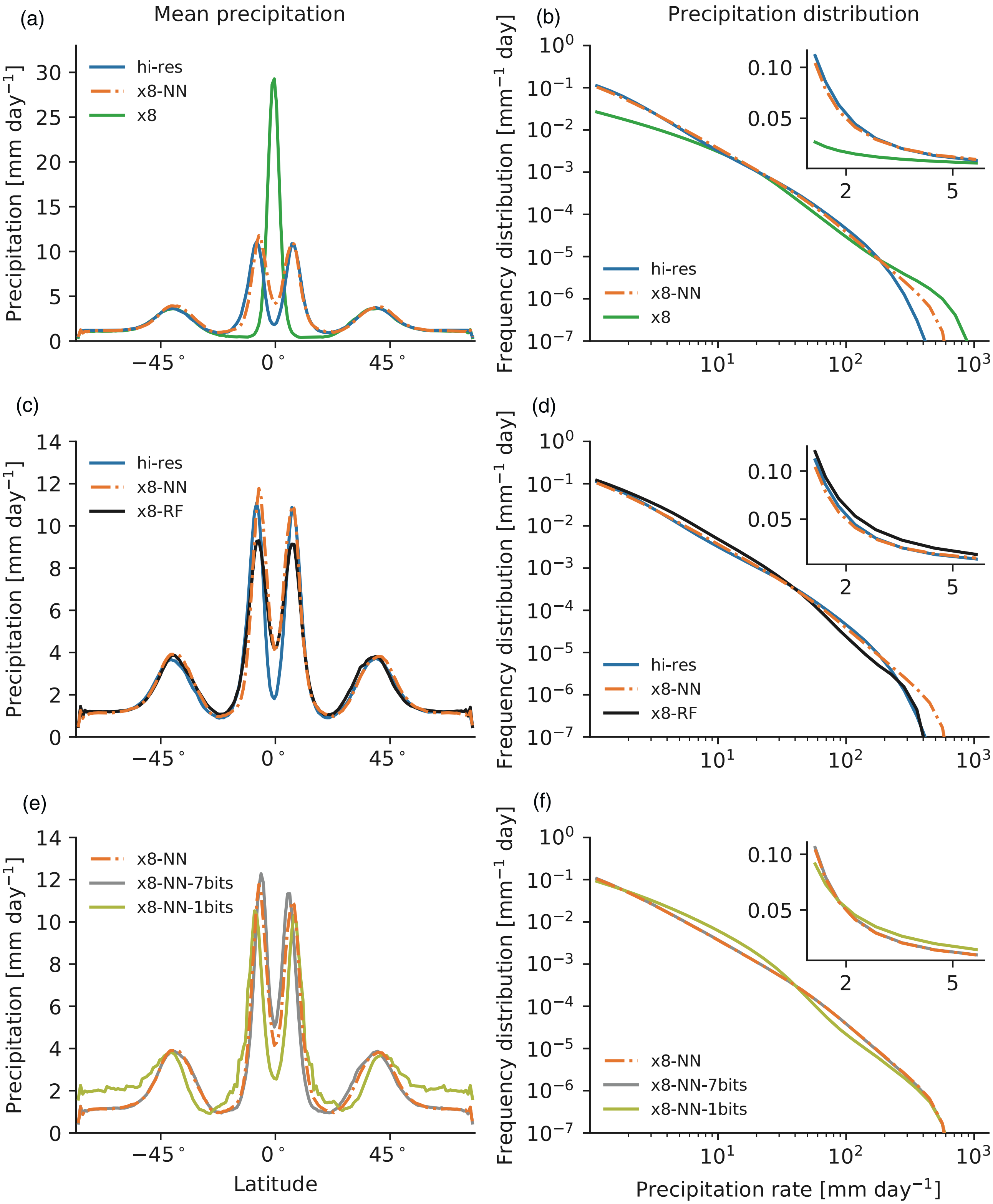
We find that these results are reproduced when the NN1 architecture is changed to have only three or four layers (Figure S2), although the amplitude of the equatorial minimum of precipitation slightly varies between simulations. All the simulations we ran were stable and without climate drift, and even a substantially less accurate NN parameterization with two layers for NN1 (Table S2) leads to stable simulations, though it does not capture the precipitation distributions (Figure S2). When training an NN parameterizations for a coarse-graining factor of x16 and coupling it to a simulation with the corresponding grid spacing, similar results are obtained and precipitation distributions are captured correctly (Figure S3). Overall, we find that coupling an NN parameterization to a simulation at coarse-resolution leads to a climate that is similar to the hi-res climate, and that the stability of simulations is not sensitive to the NN architecture, hyperparameter tuning or the horizontal resolution of the simulations.

Several approximations where made when deriving the instantaneous precipitation rate (text S1). These approximations and inaccuracies in the NN predictions result in negative 3-h surface precipitation in 20% of samples in the x8-NN simulation. Nevertheless, almost all of the negative values are very small, and only 0.03% of samples are less than -1 mm day^{-1}.

4.2. Comparing Neural Network and Random Forest Parameterizations

In this section, we compare the offline and online performance of NN parameterization to the ("alternative") RF parameterization that was developed in YOG20 and also did not include q_p. For offline performance (performance when not coupled to a GCM), we find that the NN parameterization outperforms the RF parameterization across all variables (Figure S4, Text S4), although the comparison is not straightforward since a different number of levels is used for input variables for RF and NN, and different outputs are predicted by the two parameterizations (Text S4). Yet, the online performance of the x8-NN and x8-RF simulations is comparable (Figures 2c and 2d and Table S1). Both x8-NN and x8-RF have a double ITCZ as in hi-res. The x8-NN simulation better captures the frequency distribution at most precipitation rates, though for extreme events x8-RF is more accurate (Figure 2d). Other climatological measures such as mean q_T, meridional and zonal wind and eddy kinetic energy are better captured by x8-RF (Table S1).

One advantage of NN is that it requires less memory compared to RF. For example, for x8, NN1 is 0.3 MB and NN2 is 0.2 MB when stored in netcdf format at single precision, which is ≈1900 times less memory compared to the memory needed to store the RF parameterization. Another advantage of the NN parameterization is that the x8-NN simulation requires less CPU time than the x8-RF simulation by a factor of 1.25, although both require much less CPU time than hi-res (by a factor of 54 in the case of x8-NN). NN parametrizations could be even more efficient when used in climate models that run with GPUs since NN predictions involve matrix multiplications which are highly optimized and fast on GPUs.


(a)
Mean precipitation
Precipitation [mm day^{-1}]
hi-res
x8-NN
x8
0
5
10
15
20
25
30
−45°
0°
45°
(b)
Precipitation distribution
Frequency distribution [mm^{-1} day]
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{1}
10^{2}
10^{3}
0.10
0.05
2
5
hi-res
x8-NN
x8
(c)
Precipitation [mm day^{-1}]
hi-res
x8-NN
x8-RF
0
2
4
6
8
10
12
14
−45°
0°
45°
(d)
Frequency distribution [mm^{-1} day]
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{1}
10^{2}
10^{3}
0.10
0.05
2
5
hi-res
x8-NN
x8-RF
(e)
Precipitation [mm day^{-1}]
x8-NN
x8-NN-7bits
x8-NN-1bits
0
2
4
6
8
10
12
14
−45°
0°
45°
Latitude
(f)
Frequency distribution [mm^{-1} day]
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{1}
10^{2}
10^{3}
0.10
0.05
2
5
x8-NN
x8-NN-7bits
x8-NN-1bits
Precipitation rate [mm day^{-1}]

4.3. Reduced-Precision Parameterization

As discussed in the introduction, NN parameterizations run at reduced precision could bring considerable savings in speed and energy. To test the effect of reduced precision NNs, we run simulations in which the scaled inputs and outputs of the NNs are reduced in precision. Such simulations aim to check how precise the outputs and inputs have to be to give a similar climate to the full precision. In our default configuration, SAM and the NNs are evaluated in single precision which corresponds to 23 bits in the mantissa. We reduce the precision by rounding to 7,5, 3, and 1 bits in the mantissa and the resulting simulations are referred to as x8-NN-7bits, x8-NN-5bits, x8-NN-3bits, and x8-NN-1bits, while keeping the same number of bits in the exponent (8 bits). In all simulations we use the default NNs and we do not retrain these networks for different precisions. Note that 7 bits in the mantissa corresponds to the bfloat16 "brain" floating-point (7 bits in the mantissa, 8 in the exponent and 1 to determine the sign) which is used in TPUs.

For x8-NN-7bits and x8-NN-5bits simulations we find that the resulting climate is similar, while reducing the precision to x8-NN-3bits leads to a small degradation and keeping only 1 bit in the mantissa clearly degrades the results (Table S1, Figures 2e and 2f). Overall, we find that it is possible to reduce the precision of inputs and outputs even beyond bfloat16 format without substantially affecting the climate of the simulations, suggesting that NN parameterizations at reduced precision could bring substantial advantages in speed and power requirements. Using reduced numerical precision could also help with RF parameterizations by reducing their memory requirements which can be large.

5. Conclusions

In this study, we develop an NN parameterization with a physically motivated structure learned from accurate coarse graining of the output and equations of a three-dimensional high-resolution simulation of the atmosphere. We show that the NN parameterization can be dynamically coupled to the atmospheric model at coarse resolution to give stable simulations with a climate that is close to that of the high-resolution simulation. In contrast to the approach in Rasp et al. (2018), we find that simulations with the NN parameterization are stable for a variety of configurations, and in contrast to Brenowitz and Bretherton (2018, 2019), they do not exhibit climate drift. Furthermore, in contrast to Ott et al. (2020), we find that achieving stable simulations does not require the NN parameterization to be very accurate in an offline test, and a mediocre performing NN1 with two layers (i.e., one hidden layer) is stable when coupled to SAM.

We use the same high-resolution model output for training that was used by YOG20, which suggests that the stability of simulations with an RF parameterization in previous studies (O'Gorman & Dwyer, 2018, YOG20) is not only possible with RFs since we find NNs to be robustly stable as well. Instead, the stability of simulations with NN parameterizations may require accurate processing of the high-resolution model output to obtain subgrid tendencies and fluxes (in addition to not including stratospheric levels). The main differences in the processing used here compared to previous studies with NN parameterizations of atmospheric subgrid processes are that the contribution of subgrid terms were calculated using the equations of the model for the instantaneous state of the atmosphere rather than approximating them based on differences over 3-h periods (Brenowitz & Bretherton, 2018, 2019) and that subgrid corrections were calculated independently for each physical process. The latter allows us to encapsulate more physics in the parameterization, such as by calculating fluxes and sinks rather than net tendencies. A direct comparison between RF and NN parameterizations shows that although NNs are more accurate in offline tests, when coupling the parameterizations to the atmospheric model at coarse resolution, both parameterizations have comparable results. Overall, our results imply that accurate processing of the high-resolution output that is used for the training and use of a physically based structure may be more important than intensive hyper-

Figure 2. ((a),(c),(e)) Zonal- and time-mean precipitation rate as a function of latitude and ((b),(d),(f)) frequency distribution of 3-hourly precipitation rate. Results are shown for the high-resolution simulation (hi-res in blue; panels (a)–(d)), the coarse-resolution simulation (x8 in green; panels (a)–(b)), the coarse-resolution simulation with the NN parameterization (x8-NN in orange dash-dotted; panels (a)–(f)), the coarse-resolution simulation with the RF parameterization (x8-RF in black; panels (c)–(d)), and simulations with reduced numerical precision of the inputs and outputs of the NN parameterization with 7 significant bits in the mantissa (x8-NN-7bits in gray; panels (e)–(f)), and 1 significant bit in the mantissa (x8-NN-1bits in yellow; panels (e)–(f)) as compared to 23 bits in the mantissa for x8-NN. The frequency distribution is shown for axes with linear scale in the insets of (b),(d),(f). For hi-res, the precipitation is coarse-grained to the grid spacing of x8 prior to calculating the frequency distribution. hi-res, high-resolution simulation; NN, neural network; RF, random forest.

parameter tuning of an ML algorithm. Nevertheless, combining accurate processing of the high-resolution output, a physically based structure and intensive hyperparameter tuning could be necessary to achieve accurate parameterization in more difficult scenarios such as in real-geography settings. The time step in our coarse-resolution simulations cannot be larger than roughly 75 s because of the explicit time stepping of turbulent diffusion in SAM, and future work is needed to extend these results to longer time steps and to simulations with land and topography.

Finally, we show that reducing the numerical precision of the inputs and outputs of the NN parameterization to bfloat16 floating-point format leads to a similar climate compared to using single precision. This implies that NN parameterizations with reduced precision could be used for faster training, and more importantly, for reducing the computational resources and energy needed to run climate simulations. To further investigate the feasibility of NN parameterizations with reduced precision, future work should also test NN parameterizations that were trained with reduced precision, such that the weights, biases and multiplications used during forward propagation of the NN are performed at reduced precision. Our results also suggest that the simulated climate may not be strongly affected by reducing the precision of conventional parameterizations or super parameterizations, but in those cases, it would likely be necessary to check for each particular parameterization which parts of its algorithm can be safely reduced in precision (Düben et al., 2017).

Data Availability Statement

Associated code, processed data from simulations with neural network and random forest parameterizations, trained neural network parameterizations and (a link to) the output of the high-resolution simulation are available at zenodo.org (https://doi.org/10.5281/zenodo.4526521).

Acknowledgments
The authors thank Bill Boos for providing the output from the high-resolution simulation, and Tamar Regev for help in creating Figure 1 in this manuscript. The authors also thank Michael Pritchard and Peter Düben for their helpful comments. The authors acknowledge high-performance computing support from Cheyenne (doi.org/10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. J. Yuval acknowledges support from the EAPS Houghton-Lorenz postdoctoral fellowship. P. O'Gorman acknowledges support from NSF award AGS-1552195. P. O'Gorman and C. N. Hill acknowledge support from NSF award OAC 1835618. C. N. Hill is also supported by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Beucler, T., Rasp, S., Pritchard, M., & Gentine, P. (2019). *Achieving conservation of energy in neural network emulators for climate modeling*. preprint at https://arxiv.org/abs/1906.06622

Bolton, T., & Zanna, L. (2019). Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, *11*, 376–399. https://doi.org/10.1029/2018MS001472

Bony, S., & Dufresne, J.-L. (2005). Marine boundary layer clouds at the heart of tropical cloud feedback uncertainties in climate models. *Geophysical Research Letters*, *32*. L20806. https://doi.org/10.1029/2005GL023851

Boos, W. R., Fedorov, A., & Muir, L. (2016). Convective self-aggregation and tropical cyclogenesis under the hypohydrostatic rescaling. *Journal of the Atmospheric Sciences*, *73*, 525–544. https://doi.org/10.1175/JAS-D-15-0049.1

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Brenowitz, N. D., Beucler, T., Pritchard, M., & Bretherton, C. S. (2020). *Interpreting and stabilizing machine-learning parametrizations of convection*. arXiv preprint arXiv:2003.06549.

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*, 6289–6298. https://doi.org/10.1029/2018GL078510

Brenowitz, N. D., & Bretherton, C. S. (2019). Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, *11*, 2727–2744. https://doi.org/10.1029/2019MS001711

Courbariaux, M., Bengio, Y., & David, J.-P. (2014). *Training deep neural networks with low precision multiplications*. arXiv preprint arXiv:1412.7024.

Das, D., Mellempudi, N., Mudigere, D., Kalamkar, D., Avancha, S., Banerjee, K., et al. (2018). *Mixed precision training of convolutional neural networks using integer operations*. arXiv preprint arXiv:1802.00930.

Düben, P. D., McNamara, H., & Palmer, T. N. (2014). The use of imprecise processing to improve accuracy in weather & climate prediction. *Journal of Computational Physics*, *271*, 2–18. https://doi.org/10.1016/j.jcp.2013.10.042

Düben, P. D., & Palmer, T. (2014). Benchmark tests for numerical weather forecasts on inexact hardware. *Monthly Weather Review*, *142*, 3809–3829. https://doi.org/10.1175/MWR-D-14-00110.1

Düben, P. D., Subramanian, A., Dawson, A., & Palmer, T. (2017). A study of reduced numerical precision to make superparameterization more competitive using a hardware emulator in the OpenIFS model. *Journal of Advances in Modeling Earth Systems*, *9*, 566–584. https://doi.org/10.1002/2016MS000862

Fedorov, A. V., Muir, L., Boos, W. R., & Studholme, J. (2019). Tropical cyclogenesis in warm climates simulated by a cloud-system resolving model. *Climate Dynamics*, *52*, 107–127. https://doi.org/10.1007/s00382-018-4134-2

Garner, S. T., Frierson, D. M. W., Held, I. M., Pauluis, O., & Vallis, G. K. (2007). Resolving convection in a global hypohydrostatic model. *Journal of the Atmospheric Sciences*, *64*, 2061–2075. https://doi.org/10.1175/JAS3929.1

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, *45*, 5742–5751. https://doi.org/10.1029/2018GL078202

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

Hatfield, S., Chantry, M., Düben, P., & Palmer, T. (2019). Accelerating high-resolution weather models with deep-learning hardware. *Proceedings of the platform for advanced scientific computing conference*, (pp. 1–11).

Jouppi, N. P., Yoon, D. H., Kurian, G., Li, S., Patil, N., Laudon, J., & Patterson, D. (2020). A domain-specific supercomputer for training deep neural networks. *Communications of the ACM*, *63*, 67–78. https://doi.org/10.1145/3360307

Khairoutdinov, M. F., & Randall, D. A. (2003). Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities. *Journal of the Atmospheric Sciences*, *60*, 607–625. https://doi.org/10.1175/1520-0469(2003)060<0607:CRMOTA>2.0.CO;2

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. *Advances in Artificial Neural Systems*, *2013*, 1–13. https://doi.org/10.1155/2013/485913

Kuang, Z., Blossey, P. N., & Bretherton, C. S. (2005). A new approach for 3D cloud-resolving simulations of large-scale atmospheric circulation. *Geophysical Research Letters*, *32*, L02809. https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2004GL021024

Kurth, T., Treichler, S., Romero, J., Mudigonda, M., Luehr, N., Phillips, E., et al. (2018). Exascale deep learning for climate analytics. *Sc18: International conference for high performance computing, networking, storage and analysis*, (pp. 649–660).

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., et al. (2017). *Mixed precision training*. arXiv preprint arXiv:1710.03740.

Miyashita, D., Lee, E. H., & Murmann, B. (2016). *Convolutional neural networks using logarithmic data representation*. arXiv preprint arXiv:1603.01025.

Möbis, B., & Stevens, B. (2012). Factors controlling the position of the Intertropical Convergence Zone on an aquaplanet. *Journal of Advances in Modeling Earth Systems*, *4*. M00A04. https://doi.org/10.1029/2012MS000199

Neale, R. B., & Hoskins, B. J. (2000). A standard test for AGCMs including their physical parametrizations: I: The proposal. *Atmospheric Science Letters*, *1*, 101–107. https://doi.org/10.1006/asle.2000.0022

Nvidia (2020). *Nvidia A100 tensor core GPU architecture. (Technical Report)*. Retrieved from https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., & Baldi, P. (2020). *A Fortran-Keras deep learning bridge for scientific computing*. arXiv preprint arXiv:2004.10652.

Oueslati, B., & Bellon, G. (2015). The double ITCZ bias in CMIP5 models: interaction between SST, large-scale circulation and precipitation. *Climate Dynamics*, *44*, 585–607. https://doi.org/10.1007/s00382-015-2468-6

O'Gorman, P. A. (2012). Sensitivity of tropical precipitation extremes to climate change. *Nature Geoscience*, *5*, 697–700. https://doi.org/10.1038/ngeo1568

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. *Journal of Advances in Modeling Earth Systems*, *10*, 2548–2563. https://doi.org/10.1029/2018MS001351

Palmer, T. N. (2014). More reliable forecasts with less precise computations: A fast-track route to cloud-resolved weather and climate simulators? *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, *372*. 20130391. https://doi.org/10.1098/rsta.2013.0391

Pauluis, O., & Garner, S. (2006). Sensitivity of radiative–convective equilibrium simulations to horizontal resolution. *Journal of the Atmospheric Sciences*, *63*, 1910–1923. https://doi.org/10.1175/JAS3705.1

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

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences of the United States of America*, *115*, 9684–9689. https://doi.org/10.1073/pnas.1810286115

Saffin, L., Hatfield, S., Düben, P., & Palmer, T. (2020). Reduced-precision parametrization: lessons from an intermediate-complexity atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, *146*, 1590–1607. https://doi.org/10.1002/qj.3754

Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, G. K., Schär, C., & Siebesma, A. P. (2017). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*, 3–5. https://doi.org/10.1038/nclimate3190

Vanhoucke, V., Senior, A., & Mao, M. Z. (2011). Improving the speed of neural networks on CPUs. *Proceedings in Deep learning and unsupervised feature learning, neural information processing systems workshop*.

Wilcox, E. M., & Donner, L. J. (2007). The frequency of extreme rain events in satellite rain-rate estimates and an atmospheric general circulation model. *Journal of Climate*, *20*, 53–69. https://doi.org/10.1175/JCLI3987.1

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

Zanna, L., & Bolton, T. (2020). Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, *47*. e2020GL088376. https://doi.org/10.1029/2020GL088376