PNAS
www.pnas.org

Supplementary Information for

Deep learning to represent sub-grid processes in climate models

Stephan Rasp, Michael S. Pritchard and Pierre Gentine

Stephan Rasp.
E-mail: s.rasp@lmu.de

This PDF file includes:

Supplementary text
Figs. S1 to S6
Table S1
References for SI reference citations

www.pnas.org/cgi/doi/10.1073/pnas.1810286115

Supporting Information Text

SPCAM Setup. The SPCAM model source code along with our modifications, including the neural network implementation, is available at https://gitlab.com/mspritch/spcam3.0-neural-net (branch: `nn_fbp_engy_ess`).

We use the Community Atmosphere Model 3.0 (1) with super-parameterization (2) as our training and reference model. The model has an approximately two-degree horizontal resolution with 30 vertical levels and a 30 minute time step. The embedded two-dimensional cloud resolving models consist of eight 4 km-wide columns oriented meriodinally, as in Ref. (3). The CRM time step is 20 seconds. Sub-grid turbulence in the CRM is parameterized with a local 1.5-order closure. Each GCM time step the CRM tendencies are applied to the resolved grid. Note that our SPCAM setup does not feed back momentum tendencies from the CRM to the global grid. While these might be important (4), our neural network also cannot capture momentum fluxes. Using global CRM data or augmented SP that includes 3D CRM domains with interactive momentum (or 2D SP equipped with a downgradient momentum parameterization after Ref. (5)) would prove beneficial for this purpose, especially towards ocean-coupled simulations in which cumulus friction is known to be important to the equatorial cold tongue/ITCZ nexus (6). After the SP update, the radiation scheme is called which uses sub-grid cloud information from the CRM. This is followed by a computation of the surface fluxes with a simple bulk scheme and the dynamical core. CTRLCAM uses the default parameterizations which includes the Zhang-McFarlane convection scheme (7) and a simple vertical turbulent diffusion scheme.

The physical parameterization of NNCAM is 20 times faster than SPCAM and 8 times faster than CTRLCAM. This results in a total model speed-up of factor 10 compared to SPCAM and factor 4 compared to CTRLCAM. To generate the best possible training data for the neural network we run the radiation scheme every GCM time step for SPCAM and CTRLCAM. In CTRLCAM, therefore, the radiation scheme is much more computationally expensive than in the standard setup where the radiation scheme is only called every few GCM time steps.

The sea surface temperatures (SSTs) are prescribed in our aquaplanet setup that follows Ref. (8). The reference state is zonally symmetric with a maximum shifted five degrees to the North of the equator to avoid unstable behaviors observed for equatorially symmetric aquaplanet setups:

$$\mathrm{SST}(\phi) = 2 + \frac{27}{2}(2 - \zeta - \zeta^2), \qquad [1]$$

where the SST is given in Celcius, ϕ is the latitude in degrees and

$$\zeta = \begin{cases} \sin^2\left(\pi \frac{\phi - 5}{110}\right) & 5 < \phi \leq 60 \\ \sin^2\left(\pi \frac{\phi - 5}{130}\right) & -60 \leq \phi < 5 \\ 1 & \text{if} |\phi| < 60 \end{cases} \qquad [2]$$

Additionally, we run simulations with a globally increased SSTs up to 4K in increments of 1K and a zonally asymmetric run with a wavenumber one perturbation added to the reference SSTs:

$$\mathrm{SST}'(\lambda, \phi) = 3 \cos\left(\frac{\lambda \pi}{180}\right) \cos\left(0.5\pi \frac{(\frac{\phi \pi}{180} - 5)}{30}\right)^2 \quad \text{if} \quad -25 \leq \phi \leq 35, \qquad [3]$$

where λ is longitude in degrees. The sun is in perpetual equinox with a full diurnal cycle. All experiments were started with the same initial conditions and allowed to spin up for a year. The subsequent five years were used for analysis. Training data for the neural network was taken from the second year of the SPCAM simulations.

Neural network. All neural network code is available at https://github.com/raspstephan/CBRAIN-CAM

We use the Python library Keras (9) with the Tensorflow (10) backend for all neural network experiments. Our neural network architecture consists of nine fully-connected layers with 256 nodes each. This adds up to a total of 567,361 learnable parameters. The LeakyReLU activation function $\max(0.3x, x)$ resulted in the lowest training losses. The neural network was trained for 18 epochs with a batch size of 1024. The optimizer used was Adam (11) with a mean squared error loss function. We started with a learning rate of 1×10^{-3} which was divided by five every three epochs. The total training time was on the order of 8 hours on a single Nvidia GTX 1080 graphics processing unit (GPU).

The input variables for the neural network were chosen to mirror the information received by the CRM and radiation scheme but lack the condensed water species and the dynamical tendencies. The latter are applied as a constant forcing during the CRM integration. We found, however, that they did not improve the neural network performance and trimmed the input variables for the sake of simplicity. Another option would be to include the surface flux computation in the network as well. In this option the fluxes are removed from the input and the surface temperature is added. This option yielded similar results but did not allow us to investigate column energy conservation.

The input values are normalized by subtracting each element of the stacked input vector (Table S1) by its mean across samples and then dividing it by the maximum of its range and the standard deviation computed across all levels of the respective physical variable. This is done to avoid dividing by very small values, e.g. for humidity in the upper levels, which can cause the input values to become very large if the neural network predicts noisy tendencies. For the outputs, the heating and moistening rates are brought to the same order of magnitude by converting them to W kg^{-1} . The radiative fluxes and precipitation were normalized to be on the same order of magnitude as the heating and moistening rates (see Table S1 for multiplication factors). The magnitude of the output values determines their importance in the loss function. In our quadratic loss function differences are highlighted even further. Making sure that no single value dominates the loss is important to get

a consistent prediction quality. For a reasonable range (factor five) around our normalization values the results are largely unaffected, however.

Deep neural networks appear to be essential to achieve a stable and realistic prognostic implementation. Similar to other studies which used shallow neural networks (12, 13) we encountered unstable modes and unrealistic artifacts for networks with two or one hidden layers (Fig. S1). A four layer network was the minimal complexity to provide good results for our configuration. Adding further layers shows little correlation between training skill and prognostic performance. We chose our network design to lie well withing the range of stable network configurations.

Table S1. Table showing input and output variables and their number of vertical levels N_z. For the output variables the normalization factors are also listed. C_p is the specific heat of air. L_v is the latent heat of vaporization.

Input variables	Unit	$\mathbf{N_z}$	Output variables	Unit	$\mathbf{N_z}$	Normalization
Temperature	K	30	Heating rate ΔT_{phy}	K s^{-1}	30	C_p
Humidity	kg kg^{-1}	30	Moistening rate ΔQ_{phy}	kg kg^{-1} s^{-1}	30	L_v
Meridional wind	m s^{-1}	30	Shortwave flux at TOA	W m^{-2}	1	10^{-3}
Surface pressure	Pa	1	Shortwave flux at surface	W m^{-2}	1	10^{-3}
Incoming solar radiation	W m^{-2}	1	Longwave flux at TOA	W m^{-2}	1	10^{-3}
Sensible heat flux	W m^{-2}	1	Longwave flux at surface	W m^{-2}	1	10^{-3}
Latent heat flux	W m^{-2}	1	Precipitation	kg m^{-2} d^{-1}	1	2×10^{-2}
Size of stacked vectors		94			65	

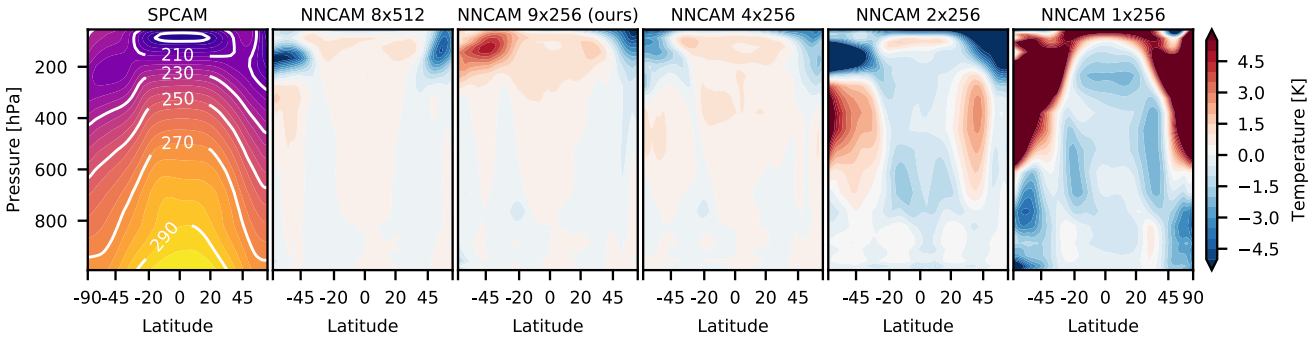

Fig. S1. All figures show longitudinal and five year-temporal averages as in Fig. 1. Zonally and temporally averaged temperature relative to SPCAM for different network configurations (Number of hidden layers x Nodes per hidden layer). 8x512 corresponds to the network in Ref. (14).

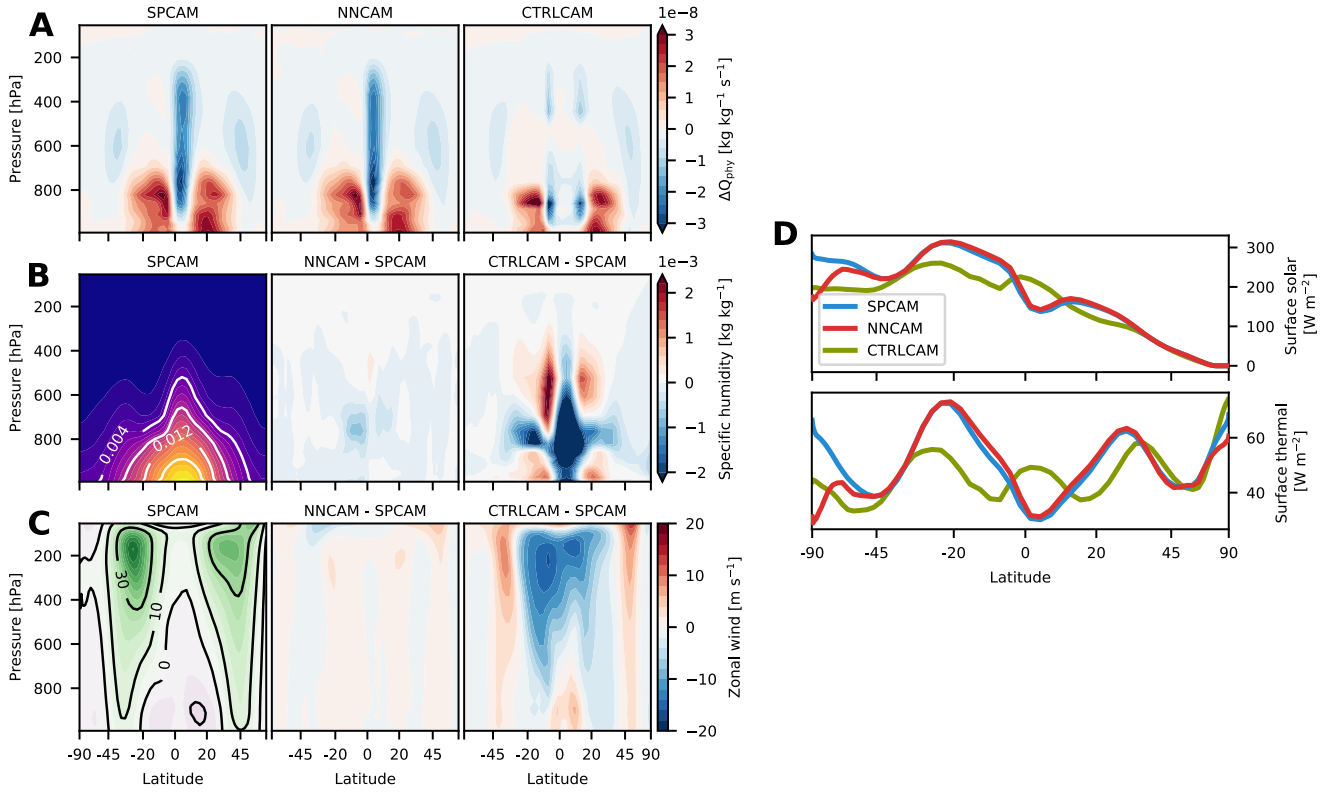

Fig. S2. (A) Mean convective sub-grid moistening rates ΔQ_{phy}. (B) Mean specific humidity Q and (C) zonal wind V of SPCAM and biases of NNCAM and CTRLCAM relative to SPCAM. (D) Mean shortwave (solar) and longwave (thermal) net fluxes at the surface. The latitude axis is area-weighted.

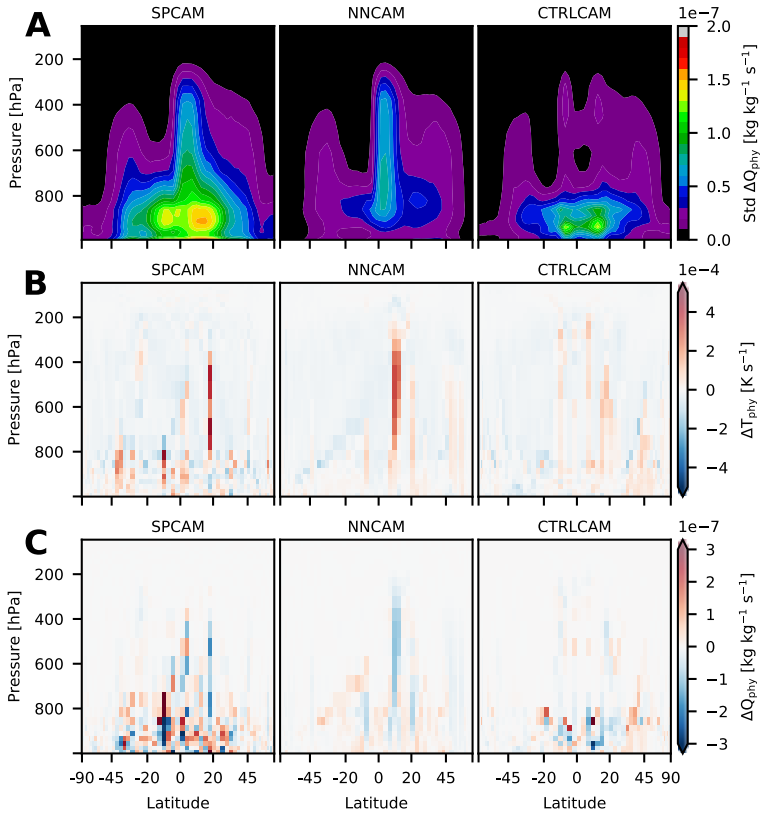

Fig. S3. (A) Zonally averaged temporal standard deviation of the convective sub-grid moistening rate ΔQ_{phy}. (B, C) Snapshots of heating ΔT_{phy} and moistening rate ΔQ_{phy}. Note that these are taken from the free model simulations and should, therefore, not correspond one-to-one between the experiments.

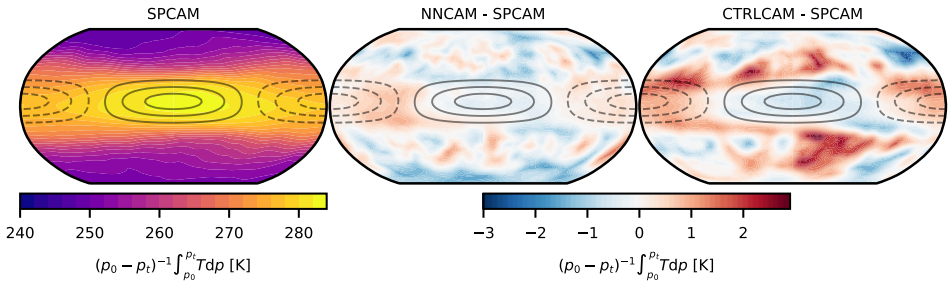

Fig. S4. Mass-weighted temperature integrated over the troposphere from $p_0 = 1000$ hPa to $p_t = 380$ hPa for SPCAM reference and differences of NNCAM and CTRLCAM with respect to reference for zonally perturbed simulations.

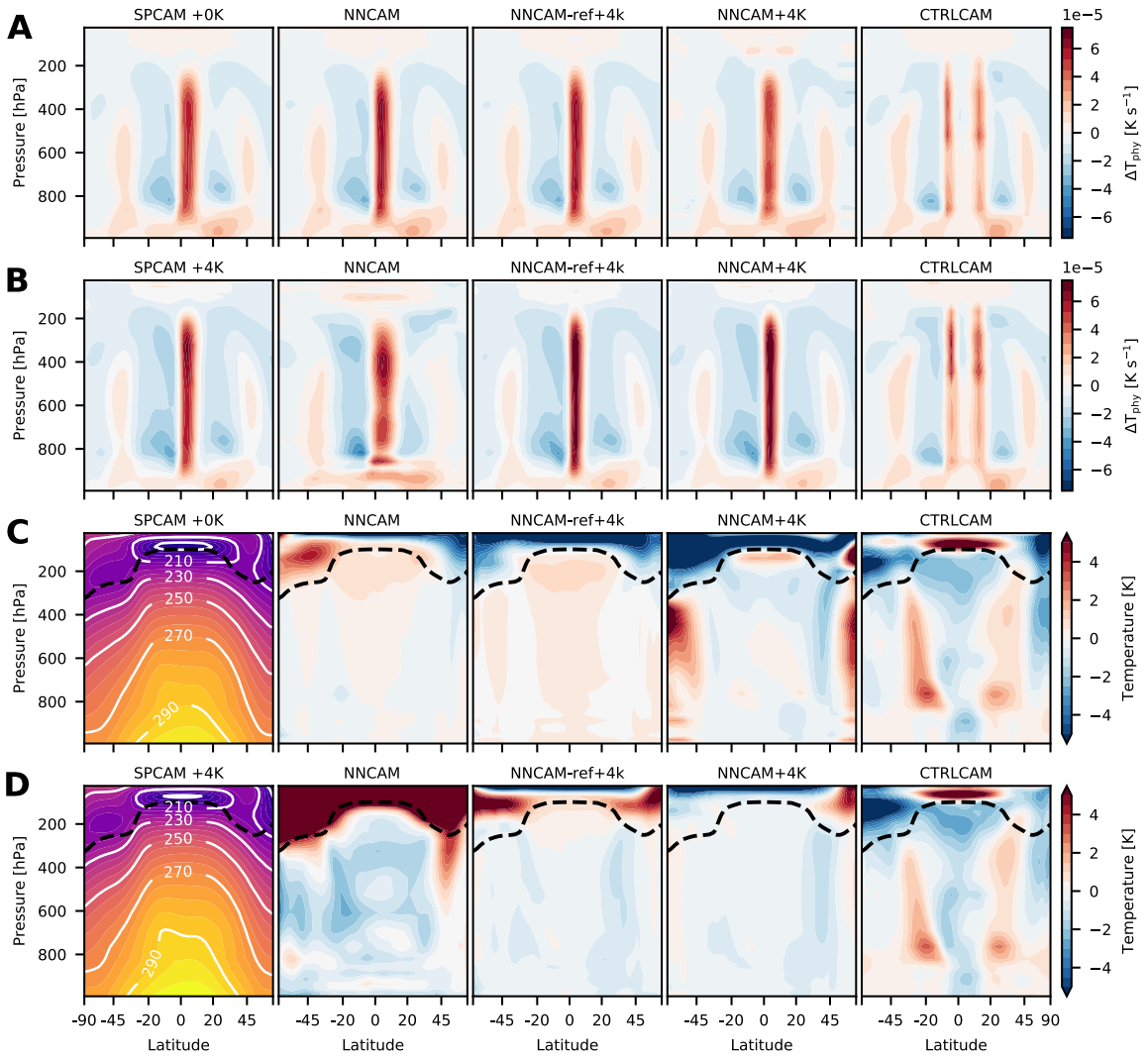

Fig. S5. Zonally and temporally averaged (A, B) heating rate and (C, D) temperature relative to SPCAM. Panels A and C show reference SSTs while panels B and D show global 4 K perturbation. Temperature panels show SPCAM reference and differences to reference for several experiments described in the text.

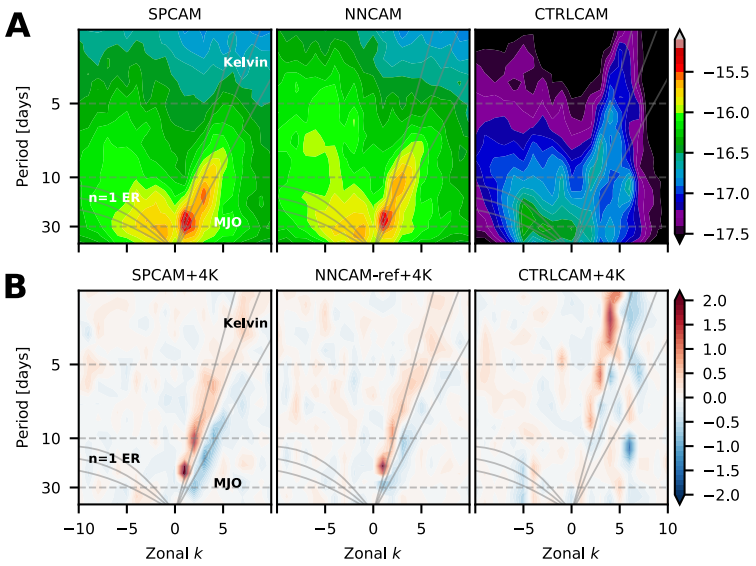

Fig. S6. (A) Space-time spectrum of the equatorially symmetric component of 15S-15N daily precipitation anomalies. As in Fig. 1b of Ref. (15). (B) Space-time spectrum of the equatorially symmetric component of 15S-15N daily precipitation anomalies divided by background spectrum. As in Fig. 3b of Ref. (15). Figure shows +4K SST minus reference SST. Negative (positive) values denote westward (eastward) traveling waves.

References

1. Collins WD, et al. (2006) The Formulation and Atmospheric Simulation of the Community Atmosphere Model Version 3 (CAM3). *Journal of Climate* 19(11):2144–2161.
2. Khairoutdinov MF, Randall DA (2001) A cloud resolving model as a cloud parameterization in the NCAR Community Climate System Model: Preliminary results. *Geophysical Research Letters* 28(18):3617–3620.
3. Pritchard MS, Bretherton CS, DeMott CA (2014) Restricting 32-128 km horizontal scales hardly affects the MJO in the Superparameterized Community Atmosphere Model v.3.0 but the number of cloud-resolving grid columns constrains vertical mixing. *Journal of Advances in Modeling Earth Systems* 6(3):723–739.
4. Moncrieff MW, Liu C, Bogenschutz P (2017) Simulation, Modeling and Dynamically Based Parameterization of Organized Tropical Convection for Global Climate Models. *Journal of the Atmospheric Sciences* pp. 16–0166.
5. Tulich SN (2015) A strategy for representing the effects of convective momentum transport in multiscale models: Evaluation using a new superparameterized version of the Weather Research and Forecast model (SP-WRF). *J. Adv. Model. Earth Syst.* 7(2):938–962.
6. Woelfle MD, Yu S, Bretherton CS, Pritchard MS (2018) Sensitivity of Coupled Tropical Pacific Model Biases to Convective Parameterization in CESM1. *J. Adv. Model. Earth Syst.* 10(1):126–144.
7. Zhang G, McFarlane NA (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.
8. Andersen JA, Kuang Z (2012) Moist Static Energy Budget of MJO-like Disturbances in the Atmosphere of a Zonally Symmetric Aquaplanet. *Journal of Climate* 25(8):2782–2804.
9. Chollet F, Others (2015) Keras. https://keras.io/.
10. Abadi M, et al. (2016) TensorFlow: A system for large-scale machine learning in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. pp. 265–283.
11. Kingma DP, Ba J (2014) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980.
12. Krasnopolsky VM, Fox-Rabinovitz MS, Belochitski AA (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.
13. Brenowitz ND, Bretherton CS (2018) Prognostic Validation of a Neural Network Unified Physics Parameterization. *Geophys. Res. Lett.* 45(12):6289–6298.
14. Gentine P, Pritchard M, Rasp S, Reinaudi G, Yacalis G (2018) Could Machine Learning Break the Convection Parameterization Deadlock? *Geophys. Res. Lett.* 45(11):5742–5751.
15. Wheeler M, Kiladis GN (1999) Convectively Coupled Equatorial Waves: Analysis of Clouds and Temperature in the Wavenumber–Frequency Domain. *Journal of the Atmospheric Sciences* 56(3):374–399.