Response to reviewer 1

This paper describes the use of a neural network (NN) to emulate a super-parameterization, and the implementation of the NN parameterization in CAM with realistic boundary conditions. The authors develop a coupler between python and Fortran which allows them to easily use python-trained NNs during online CAM runs. The authors show that the architecture they use (fully connected with skip connection) combined with the idea of separate the prediction of different outputs to different networks improve the performance of the ML parameterization. The author test their trained NNs in an online CAM setting and find that some of their NN-parameterizations lead to unstable simulations, but some lead to stable simulation (although the one simulation that the results are show for has a climate drift).

We thank the reviewer for his/her throught review of the manuscript. We appreciate many of the constructive comments and suggestions. Below is our point-by-point response. The reviewer's comments are in italic and our response is in normal and blue font. We have considered them carefully in the coming revised manuscript. Please note that the Figures 1-14 are in the revised manuscript, the Figures S1-S4 are in the supplementary, the Figures R1-R5 in the last part of this response, and the Movies S1 and S2 are in the attached files. For your convenience, we also put all related figures in the last part of the response.

I generally think that it is impressive that the authors have succeeded to run a global GCM with realistic boundary conditions with NN-parameteriztions, however I have several general comments on the manuscript that need to be addressed (and more detailed comments below):

- *It seems to me very disappointing that the authors do not provide any idea/hypothesis why some of their networks are stable and others are unstable. From what I understood, the strategy of the authors is to conduct an exhaustive random search for an accurate and stable parameterization. Personally, I think that from a scientific point of view this is not satisfactory. Furthermore, a previous study by Brenowitz et al. (2020) suggested a method to understand why certain parameterizations are unstable (they suggested that when the ML parameterizations are coupled to dynamics it may lead to unstable gravity waves) and certain actions that can be tested in order to remove such instability (e.g., input ablation), so I think it would be crucial to understand if there is an underlying physical reason why some parameterization go unstable (for example by testing if similar framework like Brenowitz et al. (2020) could really help in this case). Furthermore, unlike Mooers et al. (2020) that showed that stability of simulations is correlated with accuracy of the parameterization, other papers (e.g., Rasp et al. (2018). Yuval and O'Gorman (2020)) showed that even inaccurate parameterization can run stably for long periods. So I think it is still undetermined whether it is really necessary to have a very accurate parameterization scheme in*

> *order for it to run stably in a more realistic case. Therefore, I think it would be great if the authors could add their input on this since it seems that they did train also less accurate networks than the resMLP that they use. Namely, did all the non-resMLP networks were unstable? Overall, The achievement of running CAM with orography and a with a neural network parameterization is an important achievement that should be documented in the literature. However, from a scientific point of view, the contribution of this paper is a bit limited, and also it is not clear to me whether the NN parameterization performs better than traditional CAM parameterizations (in a few aspects it does, but in many other it does not).*

Reply:

Thank you for your insightful comments. We agree with the reviewer's point on providing possible ideas/hypotheses on stabilizing the NN parameterization, which would greatly benefit the scientific community. Since there are several questions in Subsection (a), we respond to them in the following order: neural network introduction, prognostic stability versus offline validation loss, gravity waves related to Brenowitz et al. (2020), and climate evaluation.

To determine the necessity of very accurate NN parameterizations and the architecture of ResMLP/ResDNN, we plan to show the prognostic performance versus offline validation accuracy of both ResMLP/ResDNN and fully connected neural networks which represent the non-ResMLP networks. All neural networks are trained for epochs with the same dataset. Small training epoch numbers are for less accurate NNs and large epochs for well-trained and accurate NNs.

First, we briefly list the clarifications of NN design, terminology and training. The ResMLP proposed by our study is a deep residual fully connected neural network. It is 512-node wide and 14-layer deep with an identity shortcut between every 2 layers. For the non-ResMLP networks, we introduce a 7-layer deep fully connected neural network with the same width. To avoid confusion, we apply the terms in Han et al. (2020) and respectfully change ResMLP and MLP in the manuscript to ResDNN and DNN. We think 7-layer is a proper depth for a plain DNN since the degradation problem will occur when stacking more layers (He et al., 2016). However, with the help of identity shortcuts, ResDNN can go much deeper and gain higher accuracy (Han et al., 2020). For other types of ML architectures, random forest is less likely to perform as accurately as neural networks and has not yet been well implemented in GPUs (Yuval et al., 2021). 1D convolution neural network (CNN) seemingly performs accurately offline (Han et al., 2020), but their prognostic performance is still unknown. A recent article by Chantry et al. (2021) found the local connection of CNN layers may not be suitable for emulating deep vertical connections. Therefore, we only compare the detail performance of our ResDNN with that of the DNN with dense and global connections between adjacent layers.

The training dataset used by all considered NNs is 40% temporally random sampled from the 2-year SPCAM simulation from 1997-01-01 to 1998-12-31. Notably, the random sampling is only done in the time dimension but not in latitude and longitude. As a result, the training dataset includes 13,824 samples across global grid points for each selected time step. Finally we have 97 million samples in total. To avoid any mix or temporal connection between the training and offline validation set, we random sample 40% timesteps from the SPCAM simulation in 2000 as the offline validation set. All networks begin with randomly initialized weights and biases and are trained from 5 epoch to 50 epochs with a learning rate of 0.001 and an optimizer Adam. The training loss function is the mean squared error of the selected variables (moisntening rate, heating rate, and 8 radiation fluxes) as $MSE = \left\| y_{NN} - y_{target} \right\|_2$, where y_{NN} is the NN prediction and y_{target} is the target SPCAM simulation.

The prognostic tests of NN Parameterization begin at 1998-01-01 as startup. The half-a-year spin of SPCAM is not needed but calling the CRM in SPCAM at the first step is required to generate the correct large scale forcings as the NN Parameterization input.

Our work proposes a ResDNN set, where each neural network is responsible for the prediction of a class of variables as described in section 2.2.2 of the revised manuscript. The validation R^2 of all 8 radiation fluxes is above 0.98 for ResDNN. We believe the neural network of radiation prediction is highly accurate and well trained. To simplify the neural network choices, we do not change it in the following experiments. Different from the easily and accurately trained radiation fluxes, the tendencies of both temperature and moisture are rather difficult to train and, if not trained well or not with the right NN architecture, can seriously affect the prognostic performance. Multiple ResDNN pairs (dqv and dT) and 3 DNN pairs are brought in this discussion with different offline validation accuracy. To evaluate dqv and dT in one metric, we introduce the MSE of moist static energy changing rate ($dh = C_p dT + L_v dq_v$) as:

$$MSE_h = \left\| \frac{1}{g} (dh_{NN} - dh_{SPCAM}) \Delta p \right\|_2,$$

Where g is the gravity constant, C_p refers to the heat capacity of air, L_v is the latent heat of water vapor, and Δp is the layer thickness.

Figure 4 shows the offline validation MSE_h versus the number of prognostic steps that our NNCAM can run. First, the DNN parameterizations are less accurate than the ResDNN ones in terms of offline validation accuracy. As a results, all the DNN parameterizations cannot run stably longer than half a year in prognostic tests. For the ResDNNs (blue dots and black inverted triangles), the less well-trained ones with high MSE crash within half a year simulation. However, when the offline MSE of ResDNN decreases to a certain level (e.g., $290 W^2/m^4$), the ResDNN parameterization may run stably for long periods. In Figure 4, we observed 4 ResDNNs can run stably.

Further analysis is conducted following the reviewer's suggestion. We included the framework of coupling a NN scheme into large scale dynamics introduced by Brenowitz et al. (2020). The stability is examined by the time evolution of the average energy of several stable and unstable runs (Figure 5). The system energy grows exponentially and then blows up with unstable ResDNN parameterizations (the red and orange lines), while the stable ones can maintain the total energy at a certain level and reproduce the annual cycle fluctuations like SPCAM. Among the stable ResDNN schemes, some can get nearly a perfect reproduction of the total energy evolution of SPCAM (the blue line), while some inaccurately simulate the climate state with a deviation (the green line).

The fast-growing energy of the unstable runs indicates that certain unrealistic perturbations get amplified by NN coupled dynamic system, thereby can be directly identified by the wave spectra analysis. Following Brenowitz et al. (2020), we choose the large scale profiles of temperature and moisture of the earliest breaking point as the background profiles for the 2D gravity wave. We get such a point by picking the first unstable wave initial position (Movie S1), which is around step 2270 step in the maritime continent in an unstable simulation with an offline accurate ResDNN (the red line in Figure 5). After an unreal perturbation is triggered, it rapidly propagates to nearby regions with an intense mid-level moistening/drying. Due to complex background winds and the topography below, the wave is not perfectly round. Unlike the mid-latitude breaking points in the aquaplanet experiment in Brenowitz et al. (2020), we find the tropics are of importance where most unstable waves get trigged and propagate in a realistic configurated GCM. The barotropic atmosphere in the tropics, rather than the baroclinic mid-latitudes, is more similar to the ideal 2D gravity wave model. Based on the background profiles of the breaking point, the spectra analysis for the unstable but offline accurate ResDNN indicates an unstable mode with the growth rate of vast waves between 1/day and 10/day (Figure S1b). In our study, we find ablation of the top input levels cannot stabilize the SPCAM trained NN parameterizations, which is the same as Brenowitz et al. (2020).

On the contrary, the stable ResDNNs are free from the obvious breaking points. Therefore, we have to choose the background condition of a typical convective point in the tropics. The spectra analysis of a stable ResDNN shows a stable mode with the growth rate of nearly all wave numbers with phase speed above 5m/s below zero, indicating the stable ResDNNs attenuate/damp unreal initial perturbation energy (Figure S1a). Related analysis has been added in the revised manuscript (Section 3).

As for whether the NN parameterization performs better than CAM parameterization, we show from aspects. First, in the one-year offline validation (section 4), ResDNN is very closer to both the CRM predicted moistening rate and heating rate with a high coefficient of determination R^2 in most regions in the latitude-pressure cross-section plot (Figure 7), while the CAM parameterization hardly predicts any similarity. This is also true in the global distribution of precipitation R^2 (Figure 8). Second, for the prognostic 5-year simulation (from 1999 to 2003), NNCAM simulates reasonable climate mean states and especially get both better extreme precipitation and better MJO than CAM5, because the NN parameterization inherits the high variability from the resolved cloud and convection in the SP. However, we do observed NNCAM has larger humidity and temperature biases than CAM5 in mean sates. All these results have been added in Section 5 of the revised manuscript.

In conclusion, adequate training with enough accuracy is necessary for a stable NN-parameterization in realistic configurated GCM. To get "enough" accuracy, the architecture of ResDNN rather than DNN is essential. A DNN in Rasp et al. (2018) can accurately emulate convection in aqua-planet SPCAM and brought stable muti-year prognostic simulations. Even an inaccurate DNN is stable in aqua-planet configuration but simulates biased climate. However, the real-geography data can significantly decrease the emulation skill of a DNN model (Mooers et al., 2021). The convection backgrounds are much more complex with meridional and zonal asymmetric and seasonal variated circulations, not to mention the orograph and various types of the underlying land surface. We believe that the implementation of a stable NN parameterization in a realistic configurated climate model should be much more difficult than that in an aqua-planet model. Errors caused by inaccurate emulations will broadcast to the entire model through more complex waves and circulations. Therefore, it is understandable that plain DNNs and other inaccurate NNs would not work well in the realistic configurated models. Speaking of Yuval and O'Gorman (2020) and Yuval et al. (2021), their machine learning parameterizations are trained from aquaplanet GCRM and are designed to separate vertical fluxes from convection and moistening and heating rate in microphysics and sedimentation, which inherits physical conservations and probably not that sensitive to offline accuracy. The NN-parameterization in this study is tendency-based, so it should not be directly compared with the above two works. After getting enough offline accuracy, we use the 2D gravity wave analysis framework to test stability of the NN parameterizations. The unstable ones will simulate a breaking point and with an unstable wave amplified mode, while the stable ones can attenuate the perturbation energy all the time and show a stable mode in the wave spectra. This framework explains why unstable NN parameterizations crash the model but is not an a priori way to design NN models. We still have to use the trial-and-error to filter out unstable ones and then select the best NN parameterization that can reduplicate the total energy time evolution of SPCAM with the least deviation. This selected NN-parameterization is undoubtedly closer to the SPCAM's CRM than the CAM parameterization in the offline test. For the prognostic simulations, the current NNCAM still suffers from some deviated climatology but can inherit the CRM's convective variability and do better in extreme precipitation and MJO than CAM5.

- *I think that more work on the text is needed to bring it to a level that is appropriate for the journal (see comments below). There are multiple repetitions, and sometimes sentences with little context*

Reply: Thanks for the reminder. We have tried our best to fix the problems in the text.

- *There are multiple times that the authors cite works that are not relevant to what they claim. For me especially alarming since I am not familiar with many of the references in the paper, and I found such mistakes on a small subset of references that I am familiar with. Therefore, I am worried that also other references are not relevant to what they claim.*

Reply: Thank you for the comment. We have checked and corrected all citations in the manuscript.

- *I think that some of the claims that are made in the manuscript are not supported by the results that are shown.*

Reply: We have dropped all the unsupported claims. We think this answered in the last part of comment (a) of *"and also it is not clear to me whether the NN parameterization performs better than traditional CAM parameterizations (in a few aspects it does, but in many other it does not)"* and the comments related to SPCAM, NNCAM and CAM5 comparisons below.

Comments:

Abstract - I think that the authors overstate some of their outcomes (or at least do not show explicitly in the text these results; see more comments below). Some examples: I am not sure that in their parameterization outperforms CAM in terms of spatial distribution of precipitation and MJO propagation. Of the author want to state this, I think it is necessary to show the RMSE for the spatial distribution of precipitation and to show a quantitative comparison between the MJO spectrum/propagation in CAM, SPCAM and NNCAM.

Reply: Thank you for the comment. We agreed that there is some over statements. Following your suggestion, we have rephrased them. NN-Parameterization can emulate the CRM in SPCAM much better than the CAM5 parameterizations in the offline tests. In prognostic tests, the NNCAM is still an experimental climate model. It can simulate the basic climatology with realistic configurations. However, the simulated climatology by NNCAM still cannot match the well-tuned CAM5, except better boreal summer precipitation and less biased winter distribution. Importantly, NNCAM can inherit the convection variability predicted by SPCAM, improving climate variabilities such as rainfall intensity distribution and MJO spectrum and propagation with higher pattern correlations (Figure 12, 13& 14).

- ***Line 15*** *(and also through the abstract): the authors write that they learn from a cloud resolving model. This is confusing. I think that it is important to explain that they use SPCAM, and the task at hand is an emulation of a super-parameterization rather learning from a cloud-resolving output.*

Reply: Thank you for the comment. What we do mean here is to learn from the cloud resolving model in SPCAM. We agree with the comment and have changed that part to *“to emulate a learn a parameterization scheme from different types of outputs from superparameterization (SP) with different types of outputs”* in our revised version.

• ***Line 38***: *I think that one important reference regarding the biasses that are caused by different parameterization is a paper by Wilcox and Donner (2007). I might be mistaken, but this is the only case that I know where they show how two different parameterization schemes in the same model lead to a very different frequency precipitation distribution.*

Reply: Thank you for the comment. Wilcox and Donner (2007) talked about their improvements on a RAS to simulate better extreme rainfall events in GCMs. What we tried to show here is how convection parameterizations affect the simulations of ITCZ and MJO in GCMs. We have checked the citations and changed the first sentence into *“Many studies have attributed most of these biases to the imperfection of the parameterization schemes for atmospheric moist convection and cloud processes in current GCMs (Zhang and Song, 2010; Cao and Zhang, 2017; Song and Zhang, 2018; Zhang and Song, 2019)”.*

• ***Line 44***: *Is there a citation the authors can provide for supporting the sentence “Their interaction with the atmospheric circulation affects the transport and distribution of energy and is the largest source of precipitation biases”. If not please remove.*

Reply: Thanks. Actually, this is from Emanual et al. (1994) which discussed the interaction between convection and large-scale circulation. “*largest source of precipitation biases*” is not stated in that paper. So we have deleted that part of the sentence.

• ***Line 56***: *“The CRMs have been applied to low-resolution GCMs to replace conventional cumulus convection and cloud microphysical parameterization schemes” - the sentence is unclear*

Reply: Thanks for pointing out this. We have changed it to *“In recent years, CRMs have been used as superparameterization (SP) in low-resolution GCMs to replace conventional cumulus convection and cloud parameterization schemes”.*

• ***Line 57***: *Is there any other cases where CRMs were nested in GCM except super-parameteriztation (SP)? if not, please don’t use "such as" as it implies there are other examples*

Reply: As far as we know, there is no other cases than SP. We thank the reviewer’s comment and remove *"such as"* in the revised manuscript.

- ***Line 61:*** *"an order of magnitude larger compared with that for CAM." -this really depends on the CRM configuration, right? Even in the authors simulation it seems that SPCAM uses much more than factor 10 compared to CAM.*

Reply: When using 192 Intel CPU Cores of our commondity cluseter computer for calculations, CAM runs 70 times faster than SPCAM. However, the computation speed for different for subdomain resolution. We have added *"SPCAM requires far more computing resources than the same resolution CAM in 1 to 2 order of magnitude according to the resolution of the subdomain"* at lines 60-62 in the revised manuscript.

- *Line 66: Strange sentence, and I think it has some grammatical errors (e.g., which "the data-driven parameterization scheme", from the sentence it seems you refer to a specific scheme, and I do not think it is the case)*

Reply: Thank you for the comment. We have removed that sentence in the revised manuscript.

- *Line 68: "More recently," - more recently than what?*

Reply: Thank you for this comment. We have changed *"More recently"* to *"In the last five years"* in line 65.

- *Line 72: I cannot see why the citations of Schneider et al. (2017) or of Duben and Bauer are related here. There is no convection scheme learned in these papers.*

Reply: Thank you for this comment. After checking the two papers, we acknowledged the mistake and have removed the two citation in the revised manuscript.

- *Line 74: "trained ones" - unclear*

Reply: Thank you for this comment. The description here is indeed unclear. We changed the *"machine learning algorithm"* to *"machine learning models"*, and also changed *"trained ones"* to *"trained machine learning models"*. Please see line 70 in revised manuscript.

- *Line 76: I do not see how Rasp (2020) citation is relevant here. This citation describes a method for online learning. Nothing related to an implementation of NN in GCM.*

Reply: Thanks for the comment. We have deleted such citation in the revised manuscript.

- *Line 85: "They found that minor changes, either to the training dataset or in the input/output vectors, can lead to model integration instabilities." - can you give a citation. I could not find such statement in the manuscript.*

Reply: Thanks for the comment. We cited the wrong paper. This statement is from Rasp (2020) in a section where it describes the work of Rasp et al. (2018). We have corrected this citation in line 81.

- *Line 89: There is a very relevant paper that the authors do not refer to. Brenowitz et al. (2020) have investigated both for SPCAM and for SAM why they can lead to numerical instabilities. Please add a discussion about it - and more importantly, I think that if the community could benefit from the this work it would be if the authors could identify why some of their networks are unstable (as was done by Brenowitz et al. (2020)).*

Reply: Brenowitz et al. (2020) is indeed important work. We thank the reviewer for the advice. We have added the discussion about numerical instabilities in new section 3.2 in the revised version with the 2D gravity analysis used that work and have confirmed the positive growing rate in unstable NN parameterizations.

- *Line 94-95: maybe instability and drift prevents the application some models, but other studies (that you cite) and also (Yuval et al., 2020) did not have any problem of unstable simulations. It is not fully determined why this is the case but the two possiblities that are raised in these papers are (a) because subgrid terms were calculated more accurately in these works compared to Brenowitz and Bretherton (2019), or (b) because these works succeeded to implement physical-constraints in the ML parameterizations).*

Reply: Thanks for the comment. The machine learning parameterizations in Yuval and O'Gorman (2020) and Yuval et al. (2021) are trained with inherent physical-constraints. Specifically, they are designed to emulate separated convection, microphysics, sedimentation, and vertical diffusion. The first two rely on eddy fluxes that are inherently conserved in mass and energy, and heating rate can be diagnosed from moistening with a factor Lv/Cp in the third and fourth. With this "smart" NN design, the NN prognostic simulations become stable. However, such design is possible in GCRM related studies but not in SPCAM. In our study, the NN parameterizations are tendency-based trained with realistic configurated SPCAM simulation without any physical-constrain, where stability is indeed a problem to face.

- *Line 102: What is "auto-learning technique".*

Reply: Thanks for the question. *"auto-learning technique"* is Automated Machine Learning. This is a program-driven technology that automatically searches for hyperparameters in ML algorithms (He at al. 2021). For better understanding, we change *"auto-learning technique"* to *"automated machine learning technique"*

- *Line 104: Unclear what is "group of NNs" - do you mean different NNs for predicting different outputs? if yes, you should mention that this was already done by Yuval et al. (2020) (although in a different model).*

Reply: Yes, we now refer them as a set of neural network models. In this set of neural networks, each NN is in charge of different class of outputs, including heating moistening and radiation fluxes. The work of Yuval et al. (2020) have been added as reference related statements in the revised manuscript.

• Line 105: The authors write: "We apply two innovative methods in neural network models: multi-target training to achieve balanced results across diverse neural network outputs and multilayer perceptron with residual blocks (ResMLP) to enhance nonlinear fitting ability." To me it is unclear if what the authors write here is really innovative: (a) multi-target training was already done by Yuval et al. (2020), although in a different model, so I could understand that it is a slightly different context. (b) If I understand correctly, Han et al. (2020) also used a DNN with shortcuts (called ResDNN in Han et al. (2020)) and showed it performs better than a standard DNN, which if I understand correctly is exactly what the authors call ResMLP.

As a side comment, I think that the term multi-target training is unclear and should be modified.

Reply: We thank you for mentioning the two papers. We have acknowledged the limited similarity between our work and theirs. We have revised that part of the manuscript and have changed the term of *"muti-target training"* to *"train a set of neural networks"*. Since the *"ResDNN"* in Han et al. (2020) is the same as our *"ResMLP"*, we have changed the term *"ResMLP"* to *"ResDNN"* in the manuscript. Importantly, the *"ResDNN"* in Han et al. (2020) was only shown in their sensivity test to prove that the shortcut can be implemented in fully connected neural network to gain better accuracy. It was not evaluated in details, not to mention be formly validated in prognostic simulations. It is our work to firstly use such network to achieve long-term stable simulations.

• Line 111: I understand that it would be difficult to integrate complicated NNs into Fortran, but DNN (or DNN with shortcuts) should be a very simple procedure in Fortran.

Reply: Thank you for the comment. As far as we know, python is the base in most famous NN frameworks and is widely-used in AI community. As a result, Python-based NN design has the best flexibility and productivity. Obviously, Fortran is not friendly to NN design but it is the de facto standard of scientific community, especialy for climate modeling . In our work, we want to utilize the two systems (Fortran-based climate modeling and Python-based NN design) as much as possible, and try to use the coupler technique to bridge them. We believe the idea is right and important and we will continuously improve our implementation.

• Line 115: I think it is very problematic not to mention here that the simulations are stable without mentioning that there is a climate drift (although relatively very slow one).

Reply: Thank you for the comment. We observed that the climatology simulated by NNCAM is biased but not constantly drifting away. The global distribution of the mean (Figure S4) precipitation is reasonable in global average with patterns close to those in the first 5 years (1999-01-01 to 2003-12-31) and pressure-latitude cross-section of temperature and humidity (Figure S3) for the last 5 years shows similar structure.

- *Line 145: Why is there a distinction between 2D CRM and CRM radiation? What does it mean CRM radiation?*

Reply: Thank you for the comment. The 2D CRM refers to the 2-dimensional cloud reserving model implemented in SPCAM to replace conventional subgrid parameterization, while CRM radiation refers to the revised RRTMG radiation module to let cloud-adiation interaction take place in the CRM domain. We have revised this part and added proper explanation in the revised version.

- *Line 148-151: this is a repetition of things that were already written in the intro.*

Reply: Thanks. We have deleted that part in the revised manuscript.

- *Figure 1: I might be missing something but for me these figures and the text that describe this figure is confusing. Aren't you just replacing SP with NN (like what was done in several other papers?). If yes I think that a very short and concise description will be much clearer.*

Reply: Thanks. Here we want to use NN parameterization to replace the CRM as well as the following cloud-radiation interaction in the CRM domain. Your suggestion is helpful. We have replaced the old Figure 1 in the revised manuscript with a short description in lines 197-203.

- *Line 152-154: I am not sure on what the statement is based on. I can agree that the prediction is more difficult, but why it has more numerical sensitivities (I am not even sure what it means)*

Reply: We appreciate this comment. We replace it with *"The convection background is much more complicated added with asymmetrical zonal circulation and various underlying surface."* for better clarification.

- *Line 161: the authors use the large-scale forcing as inputs. These were not used in previous papers trying to emulate SPCAM. It would be good to mention why they introduce these inputs, and if whether the inclusion of these inputs substantially improves the offline (or online) performance.*

Reply: Thanks for pointing out this. The large-scale forcings are firstly used in Gentine et al., (2018) and later replaced with meridional wind V in Rasp et al (2018). In SPCAM, the SP is continuously forced by large scale tendencies which includes all large scale processes after the SP call at previous CAM timestep (Khairoutdinov et al., 2005). Most importantly, the large scale forcings in our realistic configurated SPCAM contain PBL process and orographic gravity wave drag beside dynamics, which help our NN parameterizations identify the complicated surface conditions. We have added such description in lines 157-161.

• *Line 164: The authors predict also different outputs compared to previous studies that used ML to emulate a SP. e.g., they do not predict the vertical structure of radiative heating, but do predict other quantities. It would be helpful to understand what is the reason that they use different outputs (and inputs), and explain to the reader the motivation for these choices (and highlight differences from previous attempts to learn ML parameterization from SPCAM).*

Reply: Thank you for the suggestion. Firstly, the vertical profiles of shortwave and longwave radiation heating are included in the emulated total SP heating rate, which is the similar to those in Rasp et al. (2018). The net radiation fluxes (FSNS, FSNT, FLNS, and FSNT) are commonly used in previous studies but we find them just diagnostic terms which do not affect model states in prognostic runs. By reading Mooers et al. (2021) and the source codes of SPCAM, we later find out the solar radiation fluxes down to surface, including solar downward visible direct to surface (SOLS), solar downward near infrared direct to surface (SOLL), solar downward visible diffuse to surface (SOLSD), and solar downward near infrared diffuse to surface (SOLLD), are critical for the land surface model. If not included, the land surface is not heated up by the sun, therefore, seriously weaking the sea and land breeze. By adding those fluxes, we succeeded in getting reasonable land precipitation in the prognostic runs.We have added this description in lines 161-163 of the revised version.

• *Line 170: How does the authors deal with a negative precipitation (both offline and online)? Can you give information what is the percentage of (both online and offline samples) with negative precip?*

Reply: Thanks for the question. In fact, we set all negative precipitation to zero. It is indeed a rough way to derive precipitation from column integration of drying rate. Even for the SPCAM target data, derived negative precipitation accounts for 29% of all samples but only 14% of total precipitation intensity since most of the negative rainfalls stay above -1mm/day. For offline validation, there are 27% negative precipitation predictions in number and 17% in intensity with 93% staying above -1mm/day. For online validation, there are 22% negative precipitation in number and 17% in intensity with 93% above -1mm/day. Generally, the negative precipitation ratio in online tests is close to SPCAM target data and the offline test.

• *Line 180: Random split for the train,validation and test set is not ideal (and not the common practice) due to the time correlation between samples. In order get a reliable results for the test set what usually is done is to take the samples for each of the data* *sets from different time intervals. Please do that and report offline performances when the test set is taken from a time interval that was used during training. Alternatively, make a justification for this choice.*

Reply: Thank you for the comment. The following is how data set is made. We use SPCAM running data from January 1, 1997 to December 31, 1998 to make the training sets; we use SPCAM running data from 2000 to make test sets for all offline validation. First, divide all the original data according to GCM time steps, that is, each GCM step contains 13,824 training samples composed of global grid points. Second, we randomly sample the time dimension to ensure that their time intervals are random and uniform. This ensures that the training and validation data are not continuous in time and come from different time intervals.

- *Line 185: Is there some citation that can backup the statement that MLP can generalize better than other types of networks? If not please remove statement.*

Reply: Thank you for the suggestion. Compared with some generative models, fully connected DNN does not have an advantage in generalization performance. But it is worthwhile and necessary to try DNN first in NNCAM. First of all, the input and output of NN-Parametrization are 1-dimensional vectors composed of multiple variables. It is different from the existing mainstream machine learning problems, such as image recognition and text-speech recognition, so it is impossible to directly apply most of the existing neural networks. Taking the convolutional neural network CNN as an example, the study of Albawi et al. (2017) shows that CNN has more advantages than DNN in the learning of large-scale images. The problem we face is that the input is a 122-dimensional vector stitched by multiple different physical quantities. The dimension of a single physical quantity is only 30 dimensions, which cannot meet the requirements of large-scale (generally at least 32×32 two-dimensional images). So there is no need to use CNN. Hornik et al. (1989) proved that a single-layer neural network can approximate any function. Although the physical process problem that NN-Parametrization needs to deal with is complex, from the point of view of machine learning, it is essentially a mapping problem from a 1D-vector with a length of 122 to a 1D-vector with a length of 65. According to the universal approximation theorem, DNN is feasible. Therefore, when constructing NN-Parametrization, we first tried to use DNN for fitting, and further introduced Residual blocks to extend DNN to ResDNN.

- *General comment: I might not understand what is the NN that you used. But if your multilayer perception is just a fully connected NN please say that, and preferably change the terminology to the common one (fully connected NN)*

Reply: Thank you for the comment. A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training (Rosenblatt et al., 1961; Rumelhart et al., 1986). In this paper, MLP does refer to the Fully Connected Neural Network . Since the definition of MLP may be ambiguous, we will replace all *"MLP"* in the text with *"DNN"* and *"ResMLP"* with *"ResDNN"* in the revised manuscript.

- *Line 186-192: Please add graphs supporting the statements (these graphs can also go in the supplementary). Also the last sentence is unclear to me and I am not sure how it is related to the rest of the text.*

Reply: Thanks. In the new Figure 2, the R^2 of moist static energy changing rate is significantly lower than those of radiation fluxes under the same NN and training data, indicating different training difficulties for different variables. Also, we have removed the last sentence.

- *Line 193: The first sentence in the paragraph is unclear to me.*

Reply: Thanks. We have also removed that sentence and rewritten that paragraph (lines 171-184) in revised manuscript.

- *Lines 193-194: I disagree with this sentence about the independence - see Yuval et al. (2020) where they use the dependencies between the tendencies of moisture and thermodynamic variable to predict only one of them, and diagnose the other (because of a 1 to 1 mapping between the two for parameterizations like microphysics and sedimentation).*

Reply: Thank you for the comment. Both microphysics and sedimentation are only parts of the entire SP process (convection, microphysics, sedimentation, and diffusion). Therefore, our study's SP heating and moistening are related to each other but not the same relationship in Yuval et al (2020). We have revised this paragraph and no longer claim they are muti-target NNs. What we claim is a set of neural networks with each NN taking care of one type of output variables. So the networks can be best trained for their responding targets. The most important is to separate the NN used to predict heating and moistening with that used to predict radiation fluxes. The description has been added in lines 187-189 of the revised manuscript.

- *Line 201: I am not sure why this statement is necessary, and how it fits in the manuscript.*

Reply: Thanks for the question. In ther revised manuscript, this paragraph has been rewritten and this statement is changed to *"Since gradient descending is applied to optimize the network in training, mutual interference between different targets is expected to cause the cancel out of gradient directions used for descending (Crawshaw et al., 2020; Zhang & Yang, 2021) and ultimately affect the convergence of the network".*

• Figure 3: To me it seems that the fully connected NN performs almost identically to the ResMLP (roughly a difference of 0.002 in R2). Did you test several fully connected NNs in an online setup and verified that using skip connections is really what makes the difference in terms of stable simulations? If not, please confirm that fully connected NNs do not lead to similar results (since if I understand correctly, the authors argue that this is one of the important aspects of their work). Furthermore, I think that the R2 results that presented are confusing and not the relevant ones for the reader. R2 should be calculated over different samples without any average before since such calculation gives the idea of the real performance of the network. So please change the way R2 is evaluated.

Reply: We guess you may be referring to Figure 4. We replotted this figure as the new Figure2 with the validation dataset from the SPCAM simulation in 2000 as described in Section 2.2.1. In the revised figure, the R^2 difference is siginicant between ResDNN and DNN.

• Line 213-214: On what are you basing your statement that if the NN is underfitting NNCAM crashes quickly? Did you test 100s of NNs? 1000s NNs? 1 NN? Please show a graph supporting this evidence (similar to the one that Ott did) if you want to keep this part of the text

Reply: Thank you for your suggestion. We have plotted the Figure 4 showing offline validation accuracy versus online prognostic steps. We have implemented about 20 ResDNNs and several DNNs in NNCAM and found out most inadequately trained with low offline accuracy NNs cannot achieve long term stable online runs. We have addressed your concerns in Section 3 of the revised version.

• Line 218-219: To me, this result (more accurate NNs that crushed earlier) indicates that the accuracy is not the important part of the stability. Previous work (e.g., Rasp et al. (2018) showed that also very shallow and less accurate NNs lead to stable NNCAM simulations in an aquaplanet setting). This raises a question why the resMLP is necessary and whether fully connected network could work as well.

Reply: Thank you for the question. As our answer in the major comment (a) and the analysis in the section, we believe that high offline accuracy is necessary. The poorly trained and inaccurate NNs are unstable for long term simulations and so do the fully connected DNNs. Among the accurate ResDNNs, there is still no a prior way to determine their stability. However, with the gravity wave analysis introduced by Brenowitz et al. (2020), we can identify differences in the spectrum for stable and unstable NNs.

• Table 2: Please give the full details of the training (e.g., which optimized). Furthermore, I do not think that the data has to be shown in a table.

Reply: Thank you for the suggestion. We accordingly deleted the table 2 and added the the full details of the training stage in lines 205-216 in the revised manuscript.

• Lines 223-229: The division between the stable and unstable group is interesting and I think here would be necessary to use the tools developed by Brenowitz et al. (2020) to check stability. If these tools cannot explain the instabilities in the simulations you are conducting, it is important for the community to know this.

Reply: Thank you for your suggestion. The tools developed by Brenowitz et al. (2020) can help filter out the unstable NN parameterizations and explain why they blow up the model. The unstable ones will simulate a breaking point and have an unstable wave amplified mode, while the stable ones are able to attenuate the perturbation energy all the time and show a stable mode in the wave growth rate plot. This framework explains why unstable NN parameterizations crash the model but is not an a priori way to select stable NN parameterization. The analysis has been added in the new Section 3.2 of the revised manuscript.

• Line 249: The authors claim several times during the manuscript that it is difficult to code in Fortran an NN. However, a fully connected layer is very easy to implement as it involves only matrix multiplications so it would be good to clarify that it might be difficult to use some fancy architecture, but the basic one that the authors use is very simple to code in Fortran.

Reply: Thank you for the comment. Our new coupler excels the hard coding Fortran NNs in the following aspects: 1) It directly use the original python codes, which is more convenient to switch between different NN models since we have to conduct numerous online tests. 2) It supports all advanced NN architectures. 3) It can compute much faster by using GPUs without hindering the computational efficiency. For your concerns, we implemented a ResDNN in Fortran and tested the performance, please see the lines 295-298 in the revised manuscript.

• Line 249: "At runtime..." This is a confusing sentence and it is unclear to me why you cite the papers (as they use a different code infrastructure so I do not understand why these citations are included here)

Reply: Thank you for the comment. We accordingly deleted *"At runtime"* in the revised manuscript.

• Line 251: "outside", you need to explain outside of what.

Reply: Thank you for the comment. It means "outside of fortran program". Deploying NN in a Fortran program requires importing bias and weights of NN from outside of Fortran program. We have clarifed it in the line 236 in the revised manuscript.

- *Line 255: "In this..." this is repeating similar statements so I do not think it is necessary.*

Reply: Thank you for the suggestion.We have deleted *"In this research, we regard NN-Parameterization as a component model coupled to NNCAM."* in the revised manuscript.

- *Line 265-269: Sounds like a great achievement! (I haven't tried it myself though)*

Reply: Thank you for your encouragement.

- *Line 278: Sounds great!*

Reply: Thank you for your encouragement.

- *Line 296 and Figure 7: Please show R2 before zonal averaging! It does not make sense to me first zonal average (and also this was not done in previous work)*

Reply: Thank you for the comment. Mooers et al. (2021) made clear that their R^2 is for the zonal averaged heating and moistening. Han et al. (2020) also calculated temporal standard deviation for zonal averaged fields. So we just follow their examples here.

- *Figure 8: SP is especially important in the tropics, and it seems That the skill in the tropics is very low (many regions have R2 close to 0). Can you give some insight?*

Reply: Thank you for the comment. Our NN parameterization is trained with the loss function of mean squared error, which is not sensitive to incorrect predictions of small values. In Figure R1b, the local variance/std is close to zero for those low skill regions. The MSE in those regions is also low but is still high compared with its variance. Therefore, when calculating R^2 as 1-mse/var, many of those low std regions will have R^2 close to zero.

- *Line 301: I think that it is difficult to determine whether this is a good job or not because there is no baseline to compare to. If you could provide a baseline from CAM (for offline prediction), then it would make sense to give this statement.*

Reply: We appreciate your suggestion here. We have added the R^2 of offline predictions form CAM parameterization in SPCAM. Our NN-Parameterization shows great emulation skill by bring far better offline accuracy than the CAM parameterizations in heating, moistening and precipitation in Figure 7 and Figure 8 of the revised version.

- *Line 306: The word "fitting" should not be there as far as I understand*

Reply: Thanks. We have deleted this word in revised manuscript.

- *Line 306-308: "As suggested..." This sentence is unclear to me. How do you manually tune an NN?*

Reply: Thanks. *"Manually tune an NN"* means that in the process of training the NN, manually adjust the hyperparameters in order to improve the performance of the NN. The study of Mooers et al. (2021) pointed out that manual tuning is difficult to obtain an NN with excellent performance. In practice, we did not use *"manual tuning"* training, but used some automated tools, which use the automated machine learning technology.

- *Line 309-314: I think that these sentences are not well related to each other.*

Reply: Thanks for the comment. We have revised these sentences in Section 4 of the revised manuscript as *"The real-geography data can significantly decrease the emulation skill of a deep learning model (Mooers et al., 2021), where the convection backgrounds are much more complex with meridional and zonal asymmetric and seasonal variated circulations, not to mention the orograph and various types of underlying land surface. This result further strengthens the necessity of the ResDNN architecture."*

- *Line 317-318: "At the same time...." If I remember correctly this is the default choice*

of SPCAM so it is very confusing you are writing this as if this is something special

Reply: Thanks for the comment. We have deleted that statement.

- *Line 321: The fact simulation have a climate drift should be mentioned already in the abstract.*

Reply: Thank you for the comment. We have added *"some biases in climate simulation"* in the abstract.

- *Line 321-325: Please compare NNCAM and CAM to the relevant simulated period of SPCAM. I do not understand why you compare different periods (and different length of time intervals).*

Reply: Thanks for the comment. We have changed the simulation period from 1999-01-01 to 2003-12-31 for NNCAM, CAM5 and SPCAM.

- *Line 324: Why here (CAM) 1 year spinup and for NNCAM - half a year? I guess that this is how it is initialized?*

Reply: Thanks for the question. We were afraid of the incompetence of our NN-Parameterization in model initialization. Therefore, we ran SPCAM with the original SP for half a year to spin up the model. We later find the half a year SP run is not needed. NNCAM can be run as start up. SPCAM, CAM5 and NNCAM all start up at 1998-01-01. We have emphasized this in the revised manuscript.

• Line 324: Is it critical to use SPCAM for initialization for NNCAM? if yes please explain why? Previous studies used a coarse run (in this case CAM) to initialize the model which makes more sense to me.

Reply: The half-a-year spin of SPCAM is not needed but calling the SP in SPCAM at the first step is required to generate the correct large scale forcings for NN Parameterization. We have emphasized this in lines 364-371 in the revised manuscript.

• Figure 9: Please show RMSE in the figure for each of the sublots - so we could compare NN to CAM. Also for other figures and claims made by the the authors - if the authors want to state that NNCAM performs better than CAM, please provide a quantitative metric for the comparison.

Reply: We thanks for the suggestion and have added RMSE for the NNCAM and CAM predictions. Please take a look at the new Figure 9 in the revised manuscript. The mean climate states simulated by NNCAM is more biased than CAM5. We have added this statement the abstract.

• Figure 10: Need to quantify the accuracy - please show RMSE

Reply: Thank you for the suggestion. We have added RMSE in the new Figure 10 in the revised manuscript.

• Figure 11: NNCAM has less skill than CAM so it would be fair to clearly mention it. Furthermore, please use some metric for the comparison if such a comparison is made. The fact that CAM is closer to SPCAM than NNCAM in many of the fields should be also mentioned in the abstract (to me it is a bit misleading to only mention in the abstract that NNCAM performs better than CAM without stating that in many aspects CAM is better than NNCAM)

Reply: Thanks for the comment. We have corrected all the statements about the performance of climate mean states. We have made clear that NNCAM is still an experimental model and simulates reasonable but has some biases in climate mean states. We improved Figure 11 of zonal mean precipitation with spatial distribution differences. We believe that the improved Figure 11 will add more information than the previous one.

• Figure 12: Would be good to avoid the noise in this plot by using bins that have similar distances in a log scale.

Reply: Thank you for the suggestion. We have revised this figure by changing the log scale x axis to a linearized one and used a constant bin as 2mm/day.

• Figure 13: Can the authors quantify the distance between distributions? It is difficult for me to determine if NNCAM or CAM is closer to SPCAM, since NNCAM has a very different MJO structure compared to SPCAM. Furthermore, please add

more contours and make sure that the colorbar isn't saturated (at the moment it seems saturated).

Reply: Yes, we have reploted the Figure 13 and added coefficient of determination R^2 for the distrance between distributions. In the revised Figure 13, we don't think saturation is a problem anymore.

• *Line 383/Figure 14: The comparison should be between SPCAM propagation and NNCAM propagation because SPCAM is the baseline. I do not see why using the value of 5 m/s is relevant. To my eyes it seems that SPCAM propagates faster than 5m/s and has similar propagation as CAM. Namely, I disagree that NNCAM propagation is closer to SPCAM propagation compared to CAM.*

Reply: Thank you for the comment. We no longer use 5 m/s as a reference MJO propagation speed but directly compare the 3 time-lag plots of SPCAM, NNCAM and CAM5. When plotting precipitation and U200 at the same time, SPCAM and NNCAM show eastern propagation and but CAM5 contains a western propagation speed over Indian Ocean and the maritime continent.What's more, NNCAM has a higher R^2 of U200 than that of CAM5, showing more resemblance.

• *Figure 14: Can you describe on which data was that calculated for.*

Reply: We use the total precipitation and zonal wind U at 200mb predicted by SPCAM, NNCAM, and CAM5 respectfully.

• *Line 395: instead of "GCM" use or "a GCM" or "GCMs"*

Reply: Thanks. We applied this change in the revised manuscript.

• *Line 399-404: You cannot not mention the climate drift.*

Reply: Thank you for the comment. We think *"some biases in climate mean states"* is a more proper term than *"climate drift"*. We have discussed the stabilities and climate mean states of NNCAM in Section 3 of the revised manuscript.

• *Line 425-427: Also I think that the relevant comparison would be how fast it would run without the coupler. As far as I understand it should work pretty fast also without the coupler (which uses also additional resources) - as the matrix multiplication operation in Fortran is optimized pretty well.*

Reply: When using 192 CPU Cores of our commondity cluseter computer, the SYPD of Coupler-based NNCAM (with the support of 1 GPU) is 10.0, and the SYPD of Fortran-based NNCAM using Intel Math Kernel Library but witout GPU is only 1.5.

• *General comment: The authors mention that they have a group of NNs that lead to stable simulations. Are all these NNs lead to similar online results? If yes, please* *mention this. I suggest showing the STD for precipitation and a couple of other fields among the different stable online simulations you achieved since it will show the reader that there is no "cherry picking" with the choice of the simulation you end up showing.*

Reply: Thank you for the comment. We so far have 4 stable NN parameterizations during our experiments. They all have reasonable global distribution of precipitation but are different from each other in the heavy precipitation regions. We plot the STD for precipitation across these models as in Figure R2.

• references not ordered alphabetically in some cases which makes it more difficult to find them.

Reply: Thank you for the suggestion. We have reorganized all the references.

Reference:

Albawi, S., Mohammed, T. A., and Al-Zawi, S.: Understanding of a convolutional neural network, 2017 International Conference on Engineering and Technology (ICET), 21-23 Aug. 1-6, 10.1109/ICEngTechnol.2017.8308186, 2017.

Chantry, M., Hatfield, S., Dueben, P., Polichtchouk, I., and Palmer, T.: Machine Learning Emulation of Gravity Wave Drag in Numerical Weather Forecasting, Journal of Advances in Modeling Earth Systems, 13, e2021MS002477, https://doi.org/10.1029/2021MS002477, 2021.

Emanuel, K. A., David Neelin, J., and Bretherton, C. S.: On large-scale circulations in convecting atmospheres, Quarterly Journal of the Royal Meteorological Society, 120, 1111-1143, 10.1002/qj.49712051902, 1994.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G.: Could machine learning break the convection parameterization deadlock?, Geophysical Research Letters, 45, 5742-5751, 2018.

Han, Y., Zhang, G. J., Huang, X., and Wang, Y.: A Moist Physics Parameterization Based on Deep Learning, Journal of Advances in Modeling Earth Systems, 12, e2020MS002076, 10.1029/2020ms002076, 2020.

He, K., Zhang, X., Ren, S., and Sun, J.: Deep residual learning for image recognition, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, June 26 - July 1, 770-778, 2016.

He, Xin, Kaiyong Zhao, and Xiaowen Chu. "AutoML: A Survey of the State-of-the-Art." Knowledge-Based Systems 212 (2021): 106622.

Hornik, K., Stinchcombe, M., and White, H.: Multilayer feedforward networks are universal approximators, Neural Networks, 2, 359-366, https://doi.org/10.1016/0893-6080(89)90020-8, 1989.

Khairoutdinov, M., Randall, D., and DeMott, C.: Simulations of the Atmospheric General Circulation Using a Cloud-Resolving Model as a Superparameterization of Physical Processes, Journal of the Atmospheric Sciences, 62, 2136-2154, 10.1175/jas3453.1, 2005.

Mooers, G., Pritchard, M., Beucler, T., Ott, J., Yacalis, G., Baldi, P., and Gentine, P.: Assessing the Potential of Deep Learning for Emulating Cloud Superparameterization in Climate Models With Real-Geography Boundary Conditions, Journal of Advances in Modeling Earth Systems, 13, e2020MS002385, https://doi.org/10.1029/2020MS002385, 2021.

Rasp, S.: Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and Lorenz 96 case study (v1.0), Geosci. Model Dev., 13, 2185-2196, 10.5194/gmd-13-2185-2020, 2020.

Rasp, S., Pritchard, M. S., and Gentine, P.: Deep learning to represent subgrid processes in climate models, Proceedings of the National Academy of Sciences, 115, 9684-9689, 10.1073/pnas.1810286115, 2018.

Rosenblatt, F.: Perceptions and the theory of brain mechanisms, Spartan books, 1962.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning internal representations by error propagation, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Yuval, J. and O’Gorman, P. A.: Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions, Nature Communications, 11, 3295, 10.1038/s41467-020-17142-3, 2020.

Yuval, J., O'Gorman, P. A., and Hill, C. N.: 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, 2021.

Figure R1 - R5

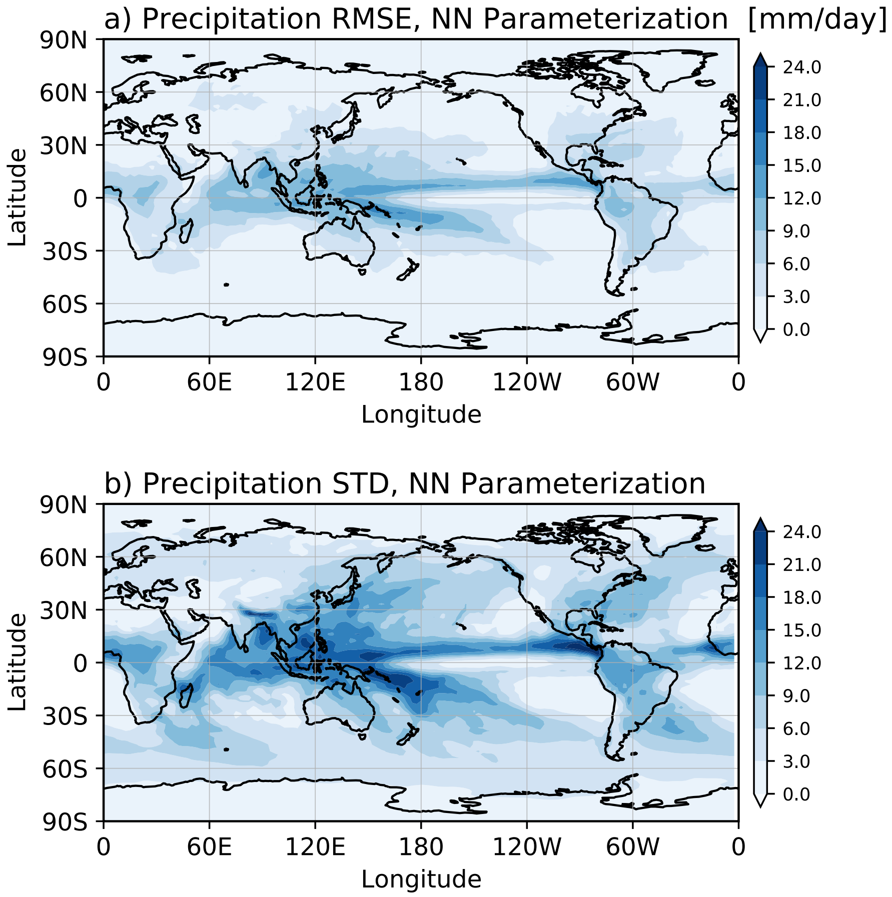

Figure R1. Spatial distribution of a) root mean square error (RMSE) and b) standard deviation (STD) of precipitation prediction.

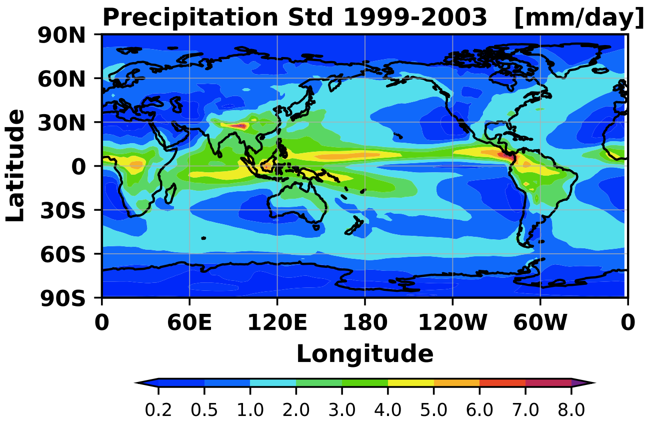

Figure R2. Spatial distribution of precipitation STD accros all 4 stable NN parameterizations for the prognotistic simulation from 1999 to 2003

Figure 1 – 14

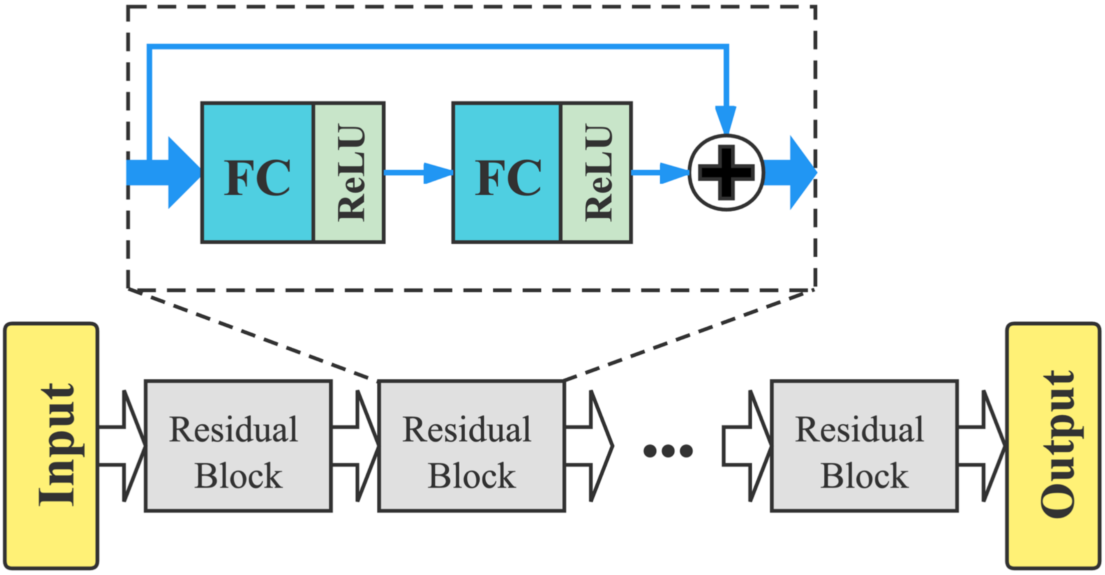

Figure 1. Schematic showing the structure of ResDNN. It consists of 7 residual blocks, each of which (shown in dashed box) contains two 512 node-wide dense (fully-connected) layers with a ReLU as activation, and a layer jump. The input and output are discussed in section 2.2.2.

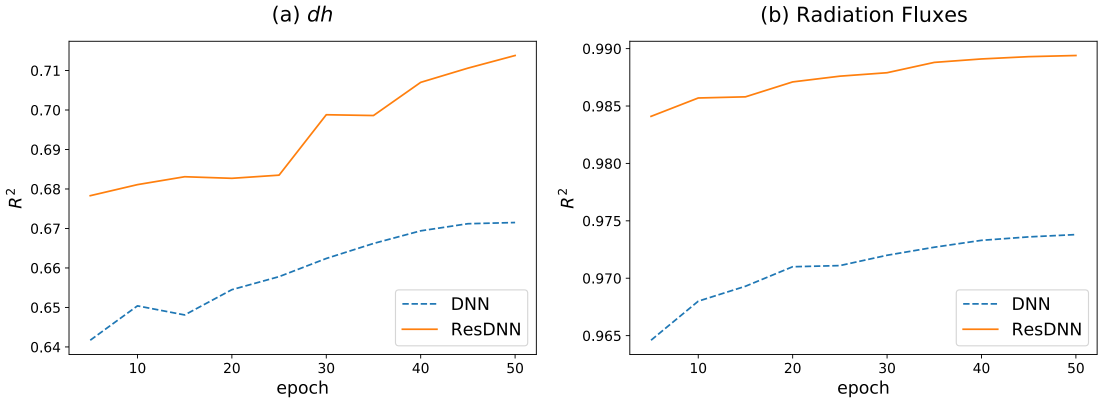

Figure 2. Fitting accuracies (R^2) of both the proposed ResDNN (orange solid lines) and DNN (blue dashed lines) for different targets. (a) shows the R2 of moist static energy changing rate (dh) versus training epochs and (b) shows the fitting accuracy of the average R^2 over the 8 radiation fluxes. Note: Spatial averaging of MSE is performed before calculating R^2.

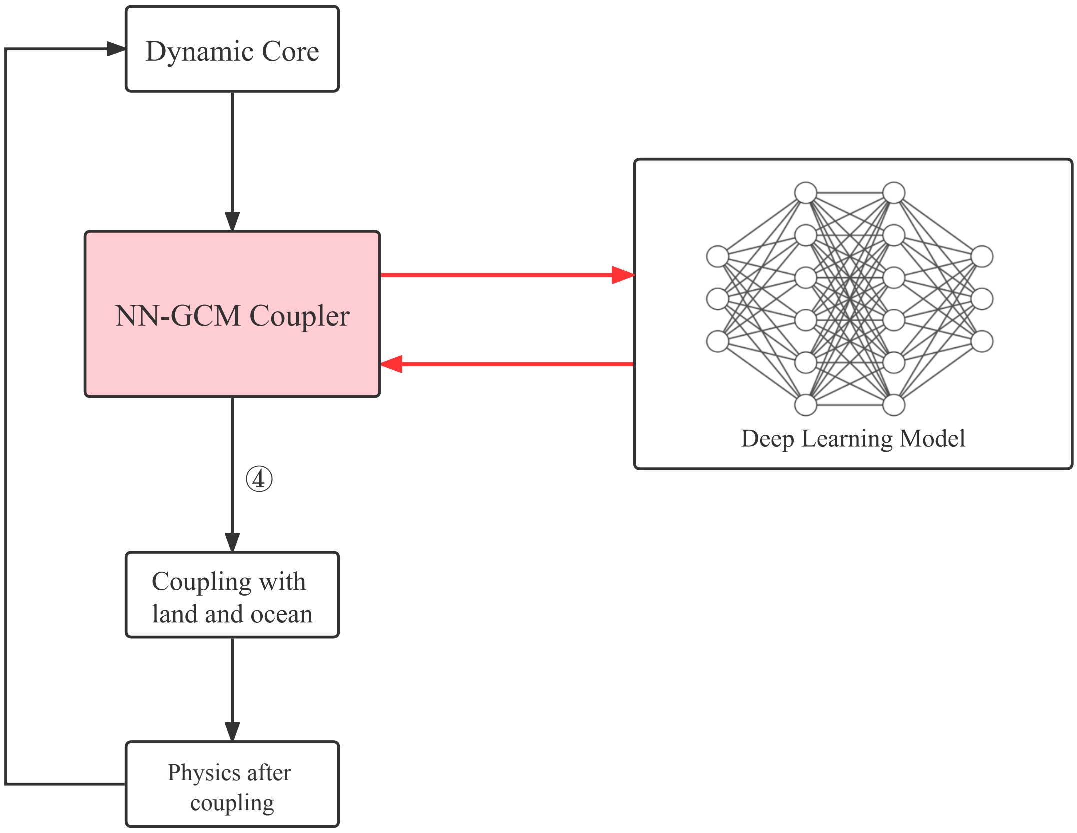

Figure 3. A flow chart of NNCAM including NN-GCM Coupler. NNCAM runs in the direction of the arrow, and each box represents a module. Among them, NN-GCM Coupler is indicated by light red. NN-Parameterization is shown in the sub-figure on the right. Note: ① represents the dynamic core transmits data to NN-GCM Coupler; ② and ③ represent the data communication between DNN-GCM Coupler and NN-Parameterization; ④ represents the host GCM accepts the result from NN-Parameterization.

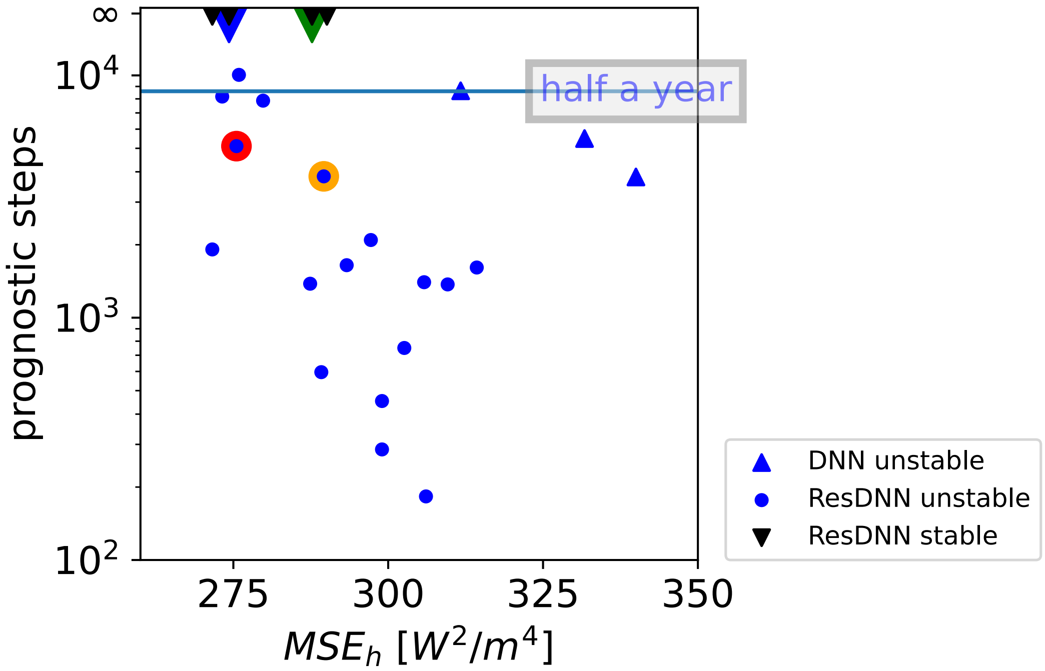

Figure 4. The offline moist static energy mean square error vs. prognostic steps. The black reversed triangles are stable NN coupled prognostic simulations lasting more than 10 years, blue ones are unstable simulations, and the blue triangles are for DNNs. The marked dots with colored outline are later exhibited in Figure 5 for time evolution of global averaged energy.

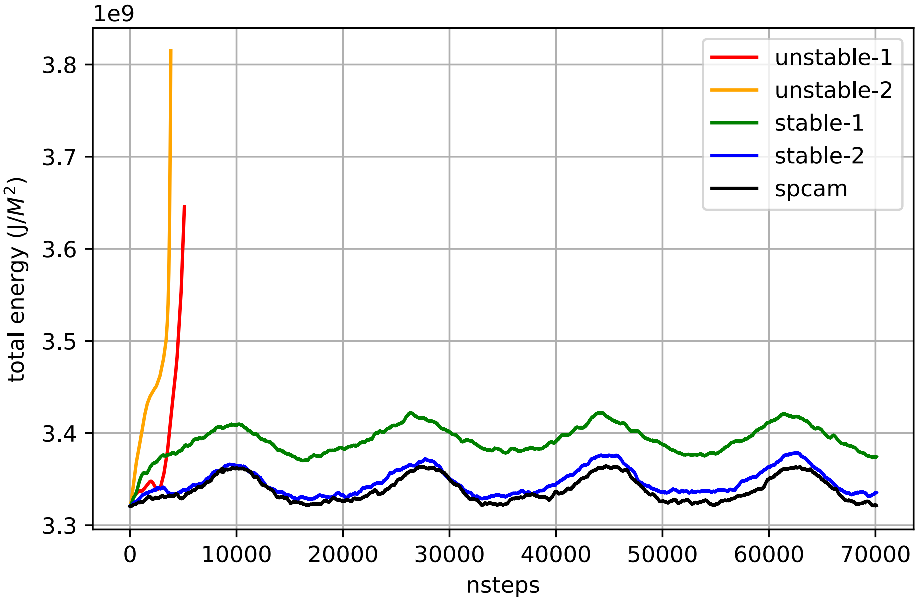

Figure 5. Time evolution of global averaged column integral total energy of NNCAM with different ResDNN parameterizations (marked with the same colors in Figure 4) and SPCAM target (the black line): Blue for stable and accurate ResDNN, green for a stable but deviated ResDNN, orange and red lines for unstable ResDNN.

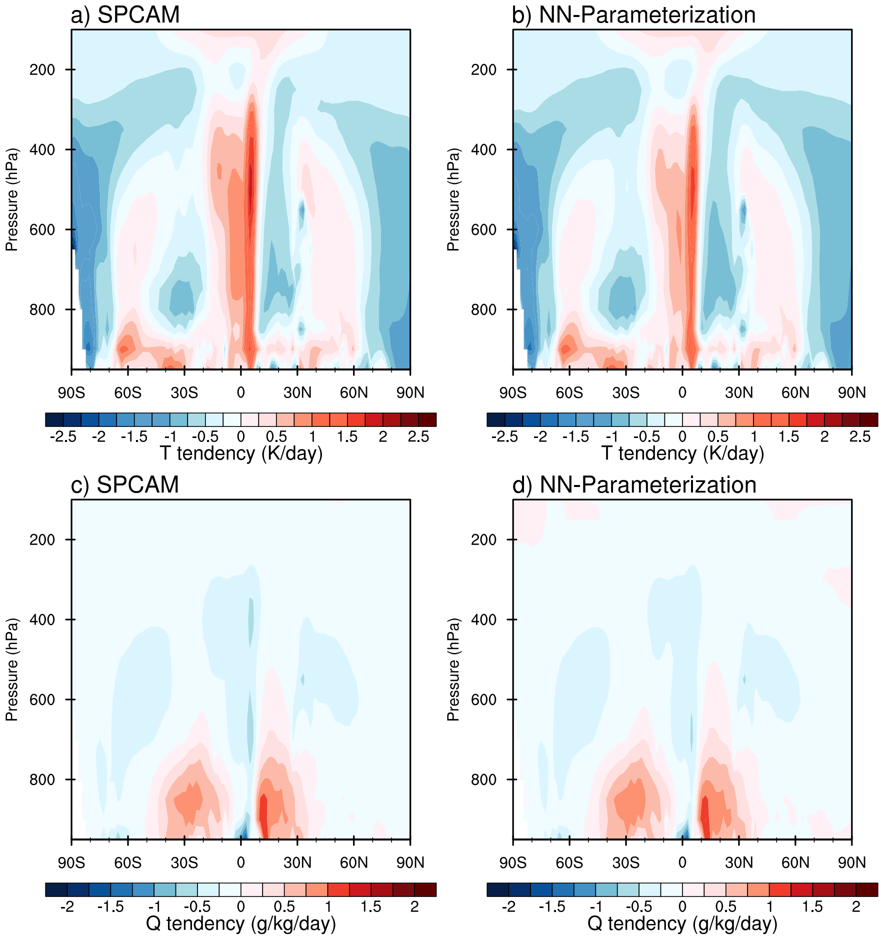

Figure 6. Latitude-pressure cross sections of annual and zonal mean heating (top) and moistening (bottom) from moist physics during the year 2000 for (a, c) SPCAM simulations, and (b, d) offline test by the NN-Parameterizations.

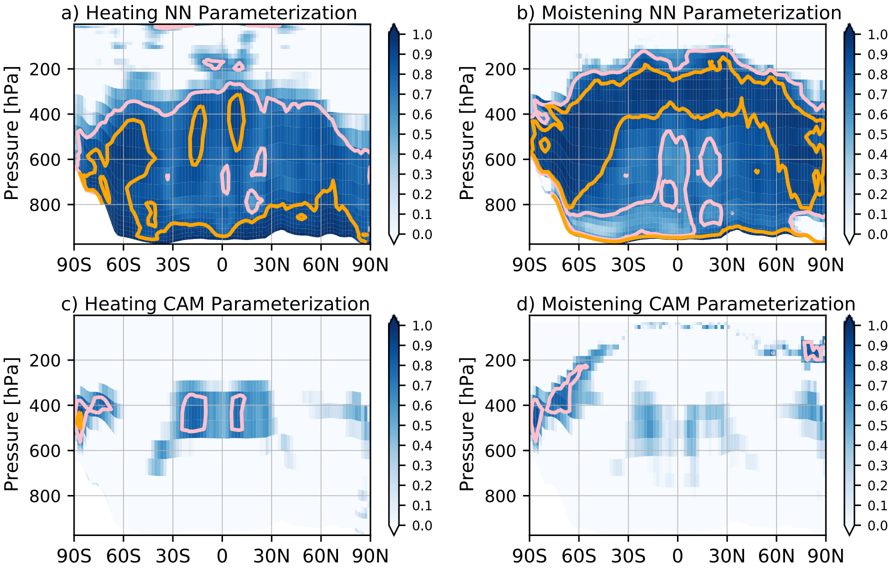

Figure 7. Latitude-pressure cross sections of coefficient of determination (R^2) for zonal averaged heating (left panels) and moistening (right panels). They are predicted by (a & b) NN-Parameterization in the offline one-year SPCAM run, and (c & d) by offline CAM5 parameterizations. Both are evaluated at 30-min timestep interval. Note: areas where R^2 is greater than 0.7 are contoured in pink and those greater than 0.9 are contoured in orange.

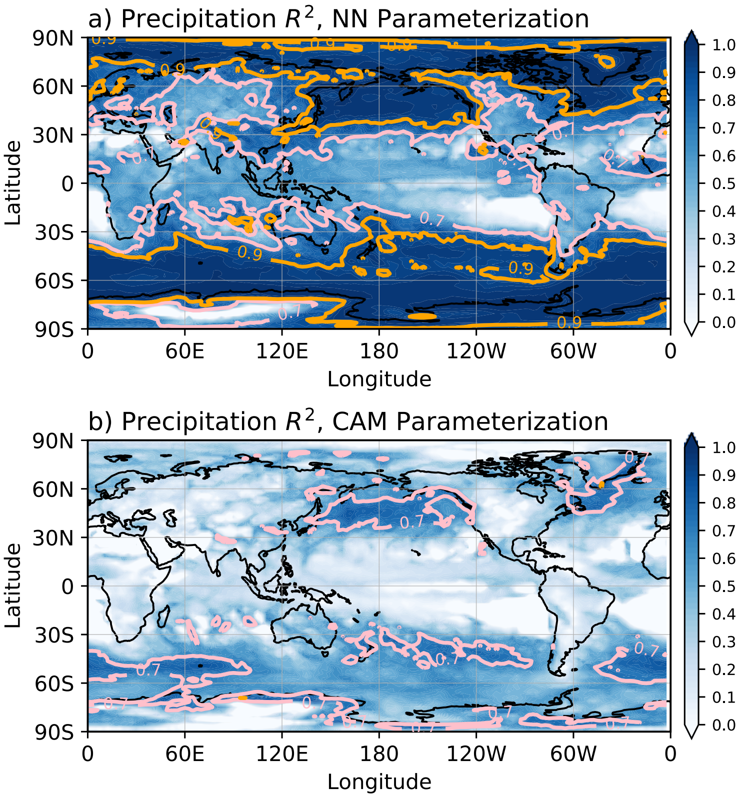
a) Precipitation R^2, NN Parameterization
b) Precipitation R^2, CAM Parameterization
Latitude
90N
60N
30N
0
30S
60S
90S
0 60E 120E 180 120W 60W 0
Longitude
0.9
0.7
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Figure 8. Latitude-pressure cross sections of coefficient of determination (R^2) for the derived precipitation predicted by NN-parameterization (a) and total precipitation from CAM5 parameterization (b) in the offline one-year SPCAM run. The predictions and SPCAM targets are in 30min timestep interval. Note: areas where R^2 is greater than 0.7 are contoured in pink and those greater than 0.9 are contoured in orange.

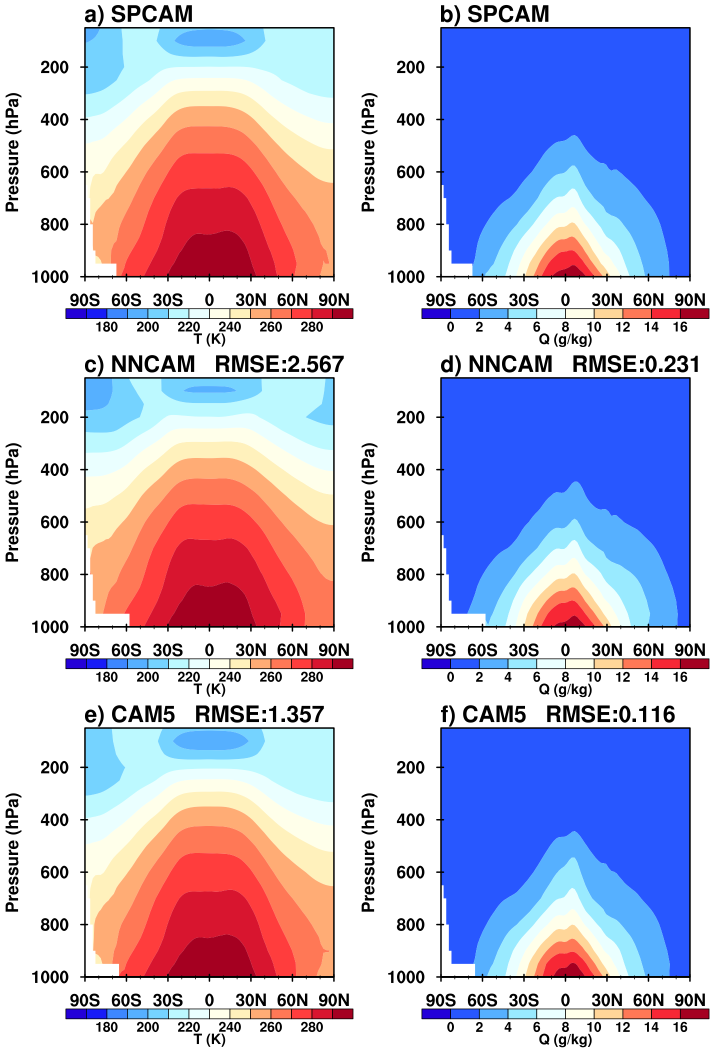

Figure 9. Latitude-pressure cross sections of annual and zonal mean temperature (left panels) and specific humidity (right panels) from (a, b) SPCAM (1999−2003), (c, d) NNCAM (1999−2003), and (e, f) CAM5 (1999−2003).

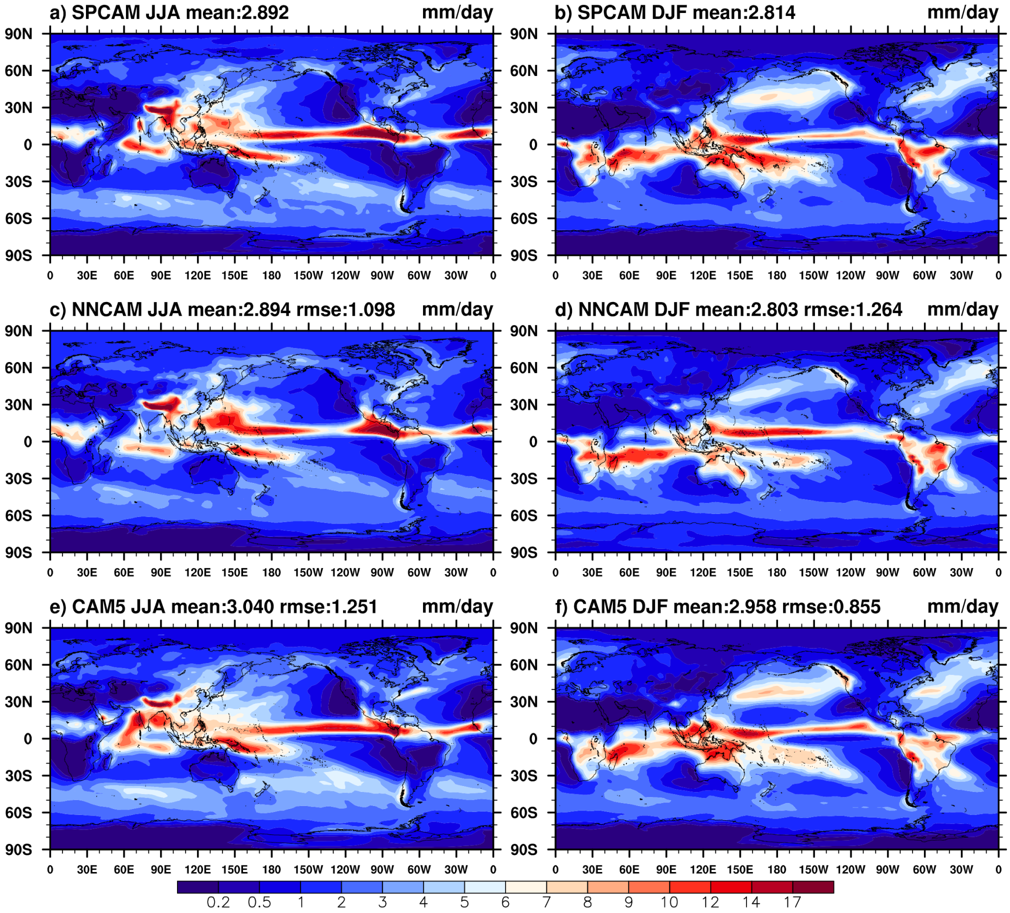

Figure 10. The mean precipitation rate (mm day^{-1}) of June-July-August (left panels) and December-January-February (right panels) for (a, b) SPCAM (1999−2003), (c, d) NNCAM (1999−2003), and (e, f) CAM5 (1999−2003).

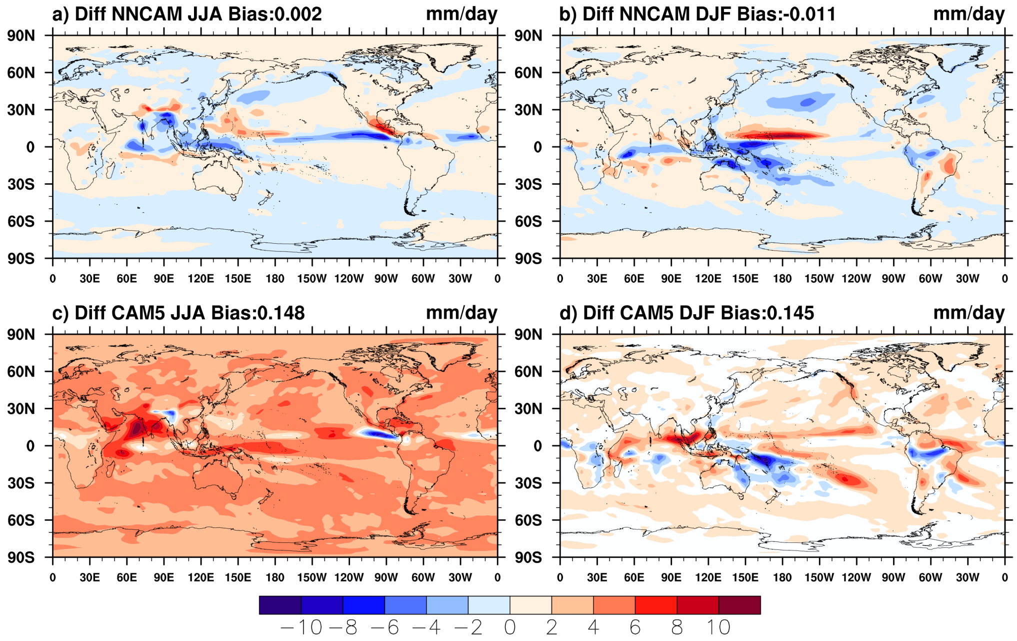
a) Diff NNCAM JJA Bias:0.002
mm/day
b) Diff NNCAM DJF Bias:-0.011
mm/day
c) Diff CAM5 JJA Bias:0.148
mm/day
d) Diff CAM5 DJF Bias:0.145
mm/day
90N
60N
30N
0
30S
60S
90S
0
30E
60E
90E
120E
150E
180
150W
120W
90W
60W
30W
0
−10 −8 −6 −4 −2 0 2 4 6 8 10

Figure 11. Global distribution of precipitation difference averged over boreal summer (left panels) and winter (right panels) between NNCAM and SPCAM (a & b) and between CAM5 and SPCAM (c & d).

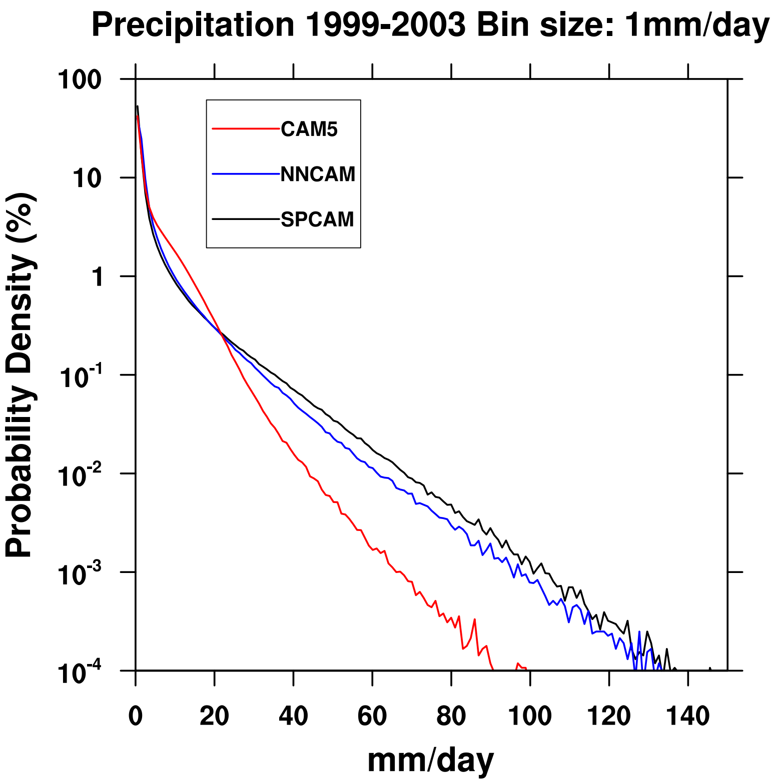

Figure 12. Probability densities of daily mean precipitation in the tropics (30°S−30°N) from the three model simulations. Black, blue and red solid lines denote SPCAM, NNCAM and CAM5, respectively.

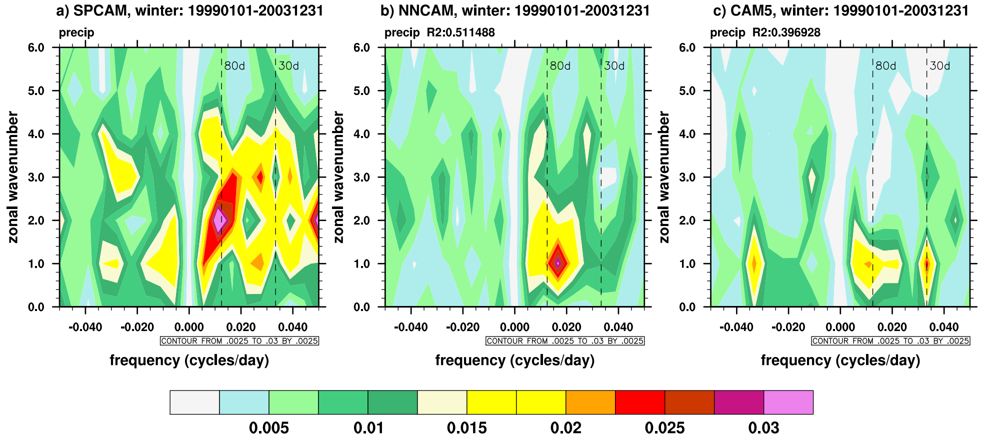

Figure 13. The wavenumber-frequency spectra of 10°S−10°N daily precipitation anomalies for (a, b) SPCAM, (c, d) NNCAM, and (e, f) CAM5 simulations for boreal winter.

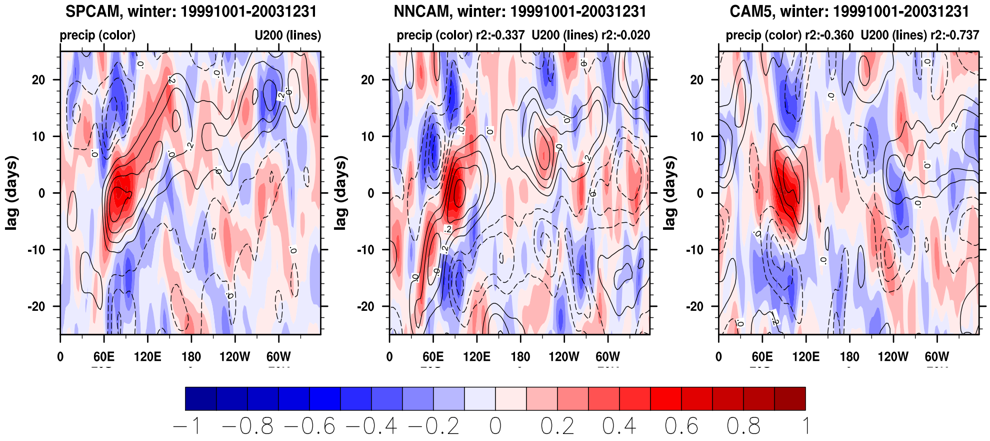

Figure 14. Longitude-time evolution of lagged correlation coefficient for the 20-100 day band-pass-filtered precipitation anomaly (averaged over 10°S−10°N) against regionally averaged precipitation (shaded) and zonal wind at 200hPa (contoured) over the equatorial eastern Indian Ocean (80E−100°E, 10°S−10°N). Dashed lines in each panel denote the 5 m s^{-1} eastward propagation speed.

Figure S1 – S4

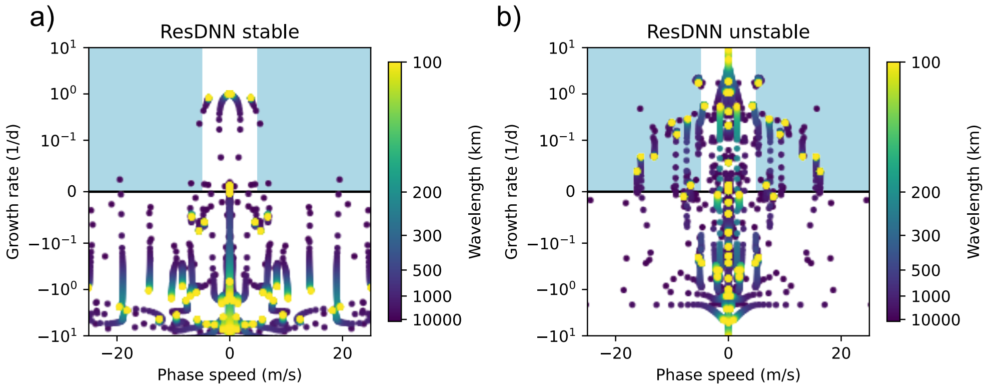

Figure S1. Wave spectra of a) a stable NN parameterization and b) an unstable parameterization. The light blue background indicates where the phase speed is above 5m/s and the growth rate is positive. The stability diagrams are obtained by coupling linear responses of NN parametrizations to simplified 2D dynamics with a chosen base state, which is normal convection background in the long-term prognostic for a) and an initial state for unreal gravity wave in Move S1 for b).

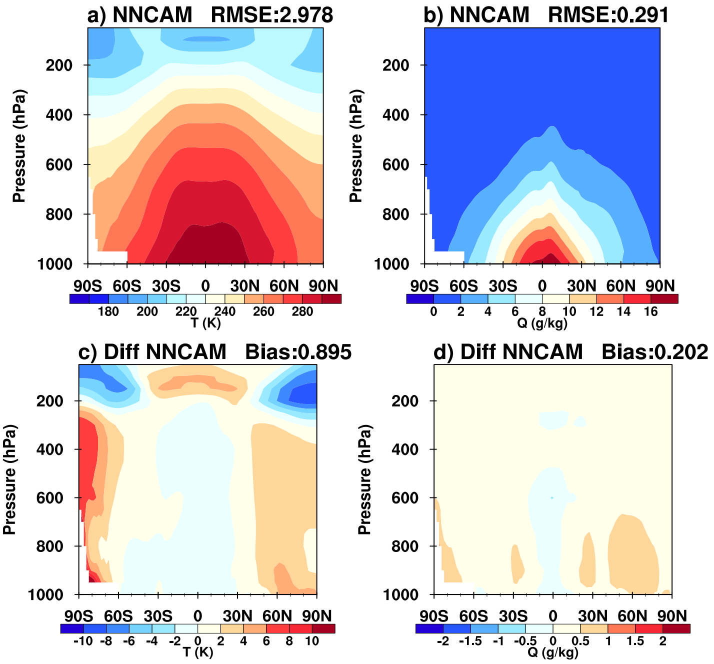

Figure S3. Latitude-pressure cross-section zonal and annual mean for a) temperature and b) specific humidity in NNCAM simulated from 2004 to 2008 with their differences with the SPCAM simulation from 1999 to 2003.

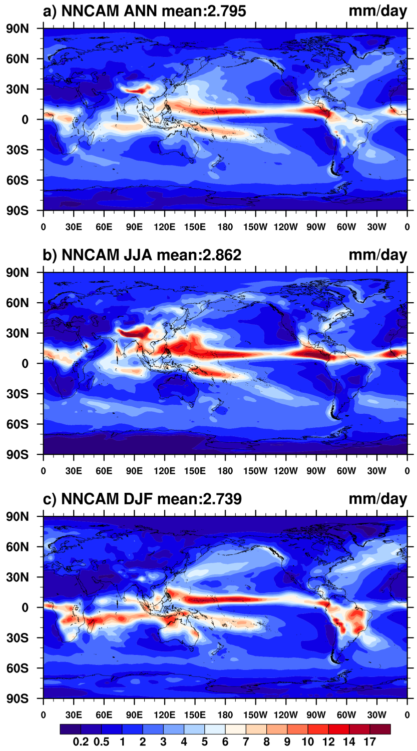

Figure S4. Global distribution of temporal mean precipitation predicted by NNCAM from January 1st 2004 to December 31st 2008 for a) annual, b) boreal summer (JJA), and c) boreal winter (DJF).