Interpreting and Stabilizing Machine-learning Parametrizations of Convection

Noah D. Brenowitz*

Vulcan, Inc, Seattle, WA

Tom Beucler

Department of Earth System Science, University of California, Irvine, CA, USA
Department of Earth and Environmental Engineering, Columbia University, New York, NY, USA

Michael Pritchard

Department of Earth System Science, University of California, Irvine, CA, USA

Christopher S. Bretherton

Vulcan, Inc, Seattle, WA
Depts of Applied Mathematics and Atmospheric sciences, Univ. of Washington, Seattle, WA

ABSTRACT

Neural networks are a promising technique for parameterizing sub-grid-scale physics (e.g. moist atmospheric convection) in coarse-resolution climate models, but their lack of interpretability and reliability prevents widespread adoption. For instance, it is not fully understood why neural network parameterizations often cause dramatic instability when coupled to atmospheric fluid dynamics. This paper introduces tools for interpreting their behavior that are customized to the parameterization task. First, we assess the nonlinear sensitivity of a neural network to lower-tropospheric stability and the mid-tropospheric moisture, two widely-studied controls of moist convection. Second, we couple the linearized response functions of these neural networks to simplified gravity-wave dynamics, and analytically diagnose the corresponding phase speeds, growth rates, wavelengths, and spatial structures. To demonstrate their versatility, these techniques are tested on two sets of neural networks, one trained with a super-parametrized version of the Community Atmosphere Model (SPCAM) and the second with a near-global cloud-resolving model (GCRM). Even though the SPCAM simulation has a warmer climate than the cloud-resolving model, both neural networks predict stronger heating/drying in moist and unstable environments, which is consistent with observations. Moreover, the spectral analysis can predict that instability occurs when GCMs are coupled to networks that support gravity waves that are unstable and have phase speeds larger than $5\,\mathrm{m\,s^{-1}}$. In contrast, standing unstable modes do not cause catastrophic instability. Using these tools, differences between the SPCAM- vs. GCRM- trained neural networks are analyzed, and strategies to incrementally improve both of their coupled online performance unveiled.

1. Introduction

Global climate models (GCMs) still cannot both explicitly resolve convective-scale motions and perform decadal or longer simulations (Intergovernmental Panel on Climate Change 2014). To permit grid spacings of 25 km or larger, important physical processes operating at smaller spatial scales, such as moist atmospheric convection, must be approximated. This task is known as sub-grid-scale parameterization, and is one of the largest sources of uncertainty in estimating the future magnitude and spatial distribution of climate change (Schneider et al. 2017).

Owing to advances in both computing and available datasets, machine learning (ML) is now a viable alternative for traditional parameterization. Viewed from the perspective of ML, parameterization is a straightforward regression problem. A parameterization maps a set of inputs, namely atmospheric profiles of humidity and temperature, to some outputs, profiles of sub-grid heating and moistening. Krasnopolsky et al. (2005) and Chevallier et al. (1998) pioneered this growing sub-field by training emulators of atmospheric radiation parameterizations. O'Gorman and Dwyer (2018) trained a random forest (RF) to emulate the convection scheme of an atmospheric GCM and were able to reproduce its equilibrium climate. More recently, neural networks (NNs) have been trained to predict the to-

* *Corresponding author*: Noah D. Brenowitz, noahb@vulcan.com

tal heating and moistening of more realistic datasets including the super-parametrized community atmosphere model (SPCAM) (Rasp et al. 2018; Gentine et al. 2018) and a near-global cloud-resolving model (GCRM) (Brenowitz and Bretherton 2018, 2019). Most ML parametrizations are deterministic, a potentially harmful approximation (Palmer 2001), but stochastic extensions of these techniques have been proposed (Krasnopolsky et al. 2013).

RFs appear robust to coupling: their output spaces are bounded since their predictions for any given input are averages over actual samples in the training data (O'Gorman and Dwyer 2018). In contrast, NNs are often numerically unstable when coupled to atmospheric fluid mechanics. In the case of coarse-grained GCRM data, Brenowitz and Bretherton (2019) eliminated an instability by ablating (i.e. removing) the upper-atmospheric humidity and temperature, which are slaved to the convective processes below, from the input space. Rasp et al. (2018) also encountered instability problems, which they solved by using deeper architectures and intensive hyperparameter tuning, but instabilities returned when they quadrupled the number of embedded cloud-resolving columns within each coarse-grid cell of SPCAM, albeit without substantial retuning of the NN.

These sensitivities suggest that numerical instability can be related to non-causal correlations in the input data or imperfect choices of network architecture and hyperparameters. Consistent with the former view, Brenowitz and Bretherton (2018) argue that a NN may detect a strong correlation between upper-atmospheric humidity and precipitation, which is used by the parameterization in a causal way (humidity affects precipitation) when the true causality is likely reversed. On the other hand, the instabilities in SPCAM do not appear to be sensitive to this causal ambiguity and are not yet fully understood, but sensitivities to hyperparameter tuning are suggestive. Regardless of its origin, for NNs, the numerical stability problem is catastrophic because current architectures can predict unbounded heating and moistening rates once they depart the envelope of the training data, motivating our first question: *can we unambiguously predict the stability of NN parameterizations of convection before coupling them to GCMs?*

Predicting the behavior of NNs is tied to the difficult problem of interpreting NN emulators of physical processes. While many interpretability techniques can be applied to NNs, such as permutation importance or layer-wise relevance propagation (e.g., McGovern et al. 2019; Toms et al. 2019; Montavon et al. 2018; Samek et al. 2017; Molnar et al. 2018), we need to adapt these techniques to interpret NN parameterizations of convection. This motivates our second question: *How can we tailor ML interpretability techniques, such as partial-dependence plots and saliency maps, for the particular purpose of interpreting NN parameterizations of convection?*

In atmospheric sciences, a common way to analyze convective dynamics utilizes the linearized response of parametrized or explicitly-resolved convection to perturbations from equilibrium (Beucler et al. 2018; Kuang 2018, 2010; Herman and Kuang 2013). These linearized response functions (LRFs) are typically computed by perturbing inputs in some basis and reading the outputs (appendix B of Beucler 2019) or by perturbing the forcing and inverting the corresponding operator (Kuang 2010). If the input/output bases are of finite dimension, then the LRF can be represented by a matrix. LRFs can also be directly computed from data, for instance by fitting a linear regression model between humidity/temperature and heating/moistening, or likewise by automatic differentiation of nonlinear regression models (Brenowitz and Bretherton 2019).

Visualizing the LRF as a matrix does not predict the consequences of coupling the scheme to atmospheric fluid mechanics (i.e., the GCM's "dynamical core"). Kuang (2010) takes this additional step by coupling CRM-derived LRFs with linearized gravity wave dynamics and further developing a vertically-truncated ordinary differential equation model (Kuang 2018). He discovered convectively-coupled wave modes that differ from the linearly unstable eigen-modes of the LRFs. This 2D linearized framework has long been used to study the instability of tropical plane-waves (Hayashi 1971; Majda and Shefter 2001; Khouider and Majda 2006; Kuang 2008), but typically by analyzing a vertically-truncated set of equations for 2–3 vertical modes. Coupling the full LRF generalizes this theoretical plane-wave analysis to a fully-resolved basis of vertical structures. When the LRF is computed from a machine-learned parametrization, the resulting wave-spectra will hint at the stability of a coupled simulation using that parameterization, but at a much lower computational expense.

While LRFs provide a complete perspective on the sensitivity of a given parameterization, they can still be difficult to interpret because they have high dimensional input and output spaces. Each side of the LRF matrix is equal to the number of vertical levels times the number of variables. However, the dominant process parametrized by ML schemes is moist atmospheric convection, which has well-known sensitivities to two environmental variables: the mid-tropospheric moisture and the lower-

tropospheric stability (LTS). On the one hand, the intensity of convection increases exponentially with the former (Bretherton et al. 2004; Rushley et al. 2018), perhaps because the buoyancy of an entraining plume is strongly controlled by the environmental moisture (Ahmed and Neelin 2018). On the other hand, convection will fail to penetrate stable air, so a sufficiently large LTS is a prerequisite for forming stratocumulus cloud layers (Wood and Bretherton 2006). While the motions in shallow turbulently-driven clouds are not resolved at the 1 km to 4 km resolution of SPCAM or GCRM training datasets, we still expect lower stability to increase the depth of convection.

In this study, we probe the behavior of the NN parameterizations from our past work using these interpretability techniques. The main goals of this study are to 1) build confidence that ML parameterizations behave like realistic moist convection and 2) introduce a diagnostic framework that can predict if a NN will cause numerical instability. We will subject two sets of NN parameterizations to this scrutiny. The first set was trained by coarse-graining a GCRM simulation (Brenowitz and Bretherton 2019) while the second set was trained using SPCAM (Rasp et al. 2018). We will use the interpretable ML toolkit to compare the sensitivities of these two schemes and by extension the SPCAM and GCRM models.

The outline of the paper follows. In Section 2, we introduce the GCRM and SPCAM training datasets, and briefly summarize our corresponding ML parameterizations, which are quite similar in form. Then, Section 3 introduces the LTS-moisture sensitivity framework (Section 3a) and the wave-coupling methodology (Section 3c). The latter is used to assess the stability of NN parameterizations, so we need to describe the techniques we use to stabilize such schemes (Section 4). Then, we present the results in Section 5. Section 5a compares how changes in LTS and moisture control the NN parameterizations, while Sections 5b and 5c apply the wave-coupling framework to predict coupled instabilities. We conclude in Section 6.

2. Machine Learning Parameterizations

a. Global Cloud-Resolving Model

Brenowitz and Bretherton (2018, 2019) trained their NNs with a near-global aquaplanet simulation performed with the System for Atmospheric Modeling (SAM) version 6.10 (Khairoutdinov and Randall 2003). This simulation is run in a channel configuration (from 46S to 46N) with a horizontal grid spacing of 4 km and 34 vertical levels of varying thickness, over a zonally symmetric ocean surface with a sea surface temperature of 300.15 K at the equator and 278.15 K at the poleward boundaries. This GCRM training data consists of 80 days of instantaneous three-dimensional fields from this simulation, sampled every three hours.

The NN scheme parametrizes the apparent heating and moistening over 160 km grid boxes, a 40-fold downsampling of the original 4 km data. This resolution is large enough that the precipitation still has significant autocorrelation at a lag of 3 hours, so the data are sampled at a high enough frequency to resolve some of the relevant moist dynamics. The apparent heating Q_1 and moistening Q_2 are defined in terms of SAM's prognostic variables: the total non-precipitating water mixing ratio q_T (kg/kg) and the liquid-ice static energy s_L (J/kg). On the coarse grid, these variables are advected by the large-scale flow and forced by the apparent heating Q_1 (W/kg) and moistening Q_2 (kg/kg/s). These dynamics are described by:

$$\frac{\partial \overline{s_L}}{\partial t} = \left(\frac{\partial \overline{s_L}}{\partial t}\right)_{\text{GCM}} + Q_1, \tag{1}$$

$$\frac{\partial \overline{q_T}}{\partial t} = \left(\frac{\partial \overline{q_T}}{\partial t}\right)_{\text{GCM}} + Q_2, \tag{2}$$

where $\bar{\cdot}$ denotes a coarse grid average.

Unlike with SPCAM (see below), the apparent sources for the GCRM are defined as budget residuals by estimating the terms in (1) and (2). The storage terms on the left hand side are estimated using a finite difference rule in time with the 3-hourly data. The tendencies due to the coarse-resolution GCM are given by $(\partial \overline{s_L}/\partial t)_{\text{GCM}}$ and $(\partial \overline{q_T}/\partial t)_{\text{GCM}}$. They are estimated by initializing our 'GCM', the coarse-resolution SAM (cSAM) at a grid spacing of 160 km with the coarse-grained data, running it forward ten 120 s time steps without any parametrized physics, and computing the time derivative by finite differences. cSAM is run with a resolution of 160 km as the coarse-graining time-scale. For more details about this complex workflow, we refer the interested reader to Brenowitz and Bretherton (2019).

b. Super-Parametrized Climate Model

To complement the NN parameterization trained on the SAM global cloud-resolving model, we analyze NNs trained on the Super-parametrized Community Atmosphere Model v3.0 (SPCAM) (Khairoutdinov and Randall 2001; Khairoutdinov et al. 2005). SPCAM embeds eight columns of SAM (spatiotemporal resolution of 4 km $\times$ 20s) in each grid column of the Community Atmosphere Model (CAM, spatiotemporal resolution of $2^\circ \times 30$min) in place of its usual deep

convection and boundary layer parameterizations to improve the representation of convection, turbulence and microphysics. In essence, SPCAM is a compromise between the numerically-expensive global SAM and the overly-coarse CAM, which struggles to represent convection (e.g., Oueslati and Bellon 2015). The goal of the NN parameterization (Rasp et al. 2018; Gentine et al. 2018) is to emulate how the embedded SAM models vertically redistribute temperature (approximately Q_1) and water vapor (approximately Q_2) in response to given coarse-grained conditions from the host model's primitive equation dynamical predictions (i.e., temperature profile, water vapor profile, meridional velocity profile, surface pressure, insolation, surface sensible heat flux, and surface latent heat flux; all prior to convective adjustment). The NN also includes the effects of cloud-radiative feedback by predicting the total diabatic tendency (convective plus radiative). Notable differences from the NN parameterization of SAM (section 2a) include training on higher frequency data (30-minute instead of 3-hourly) of a different form, in which a unambiguous separation between grid-scale drivers and subgrid-scale responses can be exploited. This facilitates the NN parameterization's definition by avoiding the challenges of coarse-graining. The lower computational cost of SPCAM, through its strategic undersampling of horizontal space, also allows a longer duration training dataset (2 years instead of 80 days for SAM) and the spherical dynamics of its host model permit fully global (including extratropical) effects. We took advantage of this longer training set by using its first year to train the NN, amounting to approximately 140M samples, and the second year to cross-validate the NN. The main disadvantage of SPCAM-based NNs is that superparameterization by definition draws an artificial scale separation that inhibits some modes of dynamics, and the idealizations of its embedded CRMs (e.g. 2D dynamics and limited extent of each embedded array) compromise aspects of the emergent dynamics (Pritchard et al. 2014). We refer the curious reader to section 2 of Rasp (2019) for an extensive comparison of various ML parameterizations of convection.

c. Neural Network Parameterization

The parameterizations for both SPCAM and the GCRM take a similar form. A neural network predicts Q_1 and Q_2 as functions of the thermodynamic state within the same atmospheric column. The parameterizations therefore have the following functional form

$$\mathbf{Q} = \mathbf{f}(\mathbf{x}, \mathbf{y}; \varphi); \tag{3}$$

where $\mathbf{Q} = [Q_1(z_1), \ldots, Q_1(z_n), Q_2(z_1), \ldots, Q_2(z_n)]^T$ is a vector of the heating and moistening for a given atmospheric column; $\mathbf{x}$ is similarly concatenated vector of the thermodynamic profiles—q_T and s_L in the case of GCRM, or humidity and temperature for SPCAM; $\mathbf{y}$ are auxiliary variables such as sea-surface temperature, the insolation at the top of atmosphere for the GCRM, or surface enthalpy fluxes for SPCAM. The ML will not prognose the source of these auxiliary variables.

Both Rasp et al. (2018) and Brenowitz and Bretherton (2019) represent f as a simple multi-layer feed-forward NN. The hyperparameters and structures of their respective networks differ slightly (e.g. number of layers, activation functions), but in this article, we will only rely on the fact that NNs are almost-everywhere differentiable, a key advantage of NNs over other techniques (e.g. tree-based models). NNs are almost-everywhere differentiable because they compose several affine transformations with nonlinear activations in between. One "layer" of such an NN transforms the hidden values x^n at the n^{th} layer into the next layer's "activations" using the following functional form;

$$\mathbf{x}^{n+1} = \sigma(A^n \mathbf{x}^n + \mathbf{b}^n), \tag{4}$$

where A^n is "weight" matrix, $\mathbf{b}^n$ is a "bias" with the same size as $\mathbf{x}^n$, and σ is a nonlinear activation function that is applied elementwise on its input vector. The inputs feed into the first layer—$\mathbf{x}^0 = \mathbf{x}$—and the outputs read out from the final layer $\mathbf{x}^m = \mathbf{Q}$. The parameters of the NN are the collection of weight matrices and bias vectors that we mathematically denote using a single vector $\varphi = \{A_1, \ldots, A_m, b_1, \ldots, b_m\}$ where m is the total number of layers.

The parameters φ are tuned by minimizing a cost function $J(\varphi)$, typically a weighted mean-squared error, using stochastic gradient descent. Modern NN libraries such as Tensorflow (Abadi et al. 2015) or PyTorch (Paszke et al. 2019) enable such a training procedure by automatically computing derivatives of functions like (3) with respect to their parameters. In this paper, we will also use this capability to explore the linearized sensitivity of a NN parameterization to its inputs across a wide array of base states.

For the GCRM, we analyze a NN that initially includes inputs from all vertical levels of humidity and temperature, a configuration that causes a prognostic simulation to crash after 6 days (Brenowitz and Bretherton 2019). The network has 3 hidden layers of 256 nodes each and uses ReLU activation, and is trained for 20 epochs using the Adam optimizer to minimize the mass-weighted mean-squared-error of predicting the Q_1 and Q_2 estimated by budget

residuals of (1) and (2). The training loss includes a regularization term ensuring that the NN converges to a stable equilibrium. The PyTorch library is used, the input data scaled by the mean and variance, and we used identical hyper-parameters as Brenowitz and Bretherton (2019). The interested reader should refer to that paper for more details.

To avoid over-fitting the GCRM NN, the western half of the data is used for training and the eastern half is reserved for testing. After 20 epochs of training, the mass-weighted root-mean-squared error (MSE) of Q_1 over the training and testing regions are $3.46\,\mathrm{K\,d^{-1}}$ and $3.47\,\mathrm{K\,d^{-1}}$, respectively; for Q_2 the scores are $1.51\,\mathrm{g\,kg^{-1}\,d}$ and $1.48\,\mathrm{g\,kg^{-1}\,d}$, respectively. The test and training errors are nearly identical, suggesting that over-fitting is not occuring. This is not surprising given the large number of samples (millions) compared to free parameters in this network configuration (100 000s). Having ruled-out overfitting, the analyses below are based on the full dataset.

For SPCAM, we analyze two NNs with identical architectures and training conditions (9 fully-connected layers of 256 nodes each trained for 20 epochs using the Adam optimizer): "NN-stable" and "NN-unstable". The tensorflow library is used. Although both NNs were trained using $\sim$ 140M samples from aquaplanet simulations, the training simulation for NN-stable used 8 SAM columns per grid cell and underwent manual hyperparameter tuning (see SI of Rasp et al. 2018) while the training simulation for NN-unstable used 32 SAM columns per grid cell and the NN was less intensively tuned. Helpfully for our purposes, NN-stable led to successful multi-year climate simulations once prognostically coupled to CAM (Rasp et al. 2018), but NN-unstable proved prone to producing moist mid-latitude instabilities that led all prognostic simulations to crash within 2-15 days (see Movie S1). While the time to crash was sensitive to initial condition details, no simulations with NN-unstable proved capable of running more than 2 weeks. Unlike for the GCRM simulation, the SPCAM interpretability analyses below are based only on data from the validation period.

In the following sections, we show how physically-motivated diagnostic tools help anticipate, explain, and begin to resolve these problematic instabilities.

3. Interpreting ML parameterizations

a. Variation of Inputs

A few important parameters control the strength and height of moist atmospheric convection. Any parameterization, including a machine learning parameterization, should capture the dependence of convection to these parameters. One such parameter is the lower tropospheric stability (LTS)

$$LTS = \theta(700\ \mathrm{hPa}) - SST$$

where θ is the potential temperature and SST is the sea-surface temperature. Low LTS indicates the lower troposphere is conditionally unstable, favoring deep convection.

A second controlling parameter is the mid-tropospheric moisture, defined by

$$Q = \int_{850}^{550} q_T \frac{dp}{g}.$$

Cumulus updrafts entrain surrounding air as they rise through the lower troposphere. If that air is dry (low Q), the entrained air induces considerable evaporative cooling as it mixes into the cloudy updraft, impeding deep precipitating convection. Hence moist columns tend to precipitate exponentially more than dry ones (Bretherton et al. 2004; Neelin et al. 2009; Ahmed and Neelin 2018; Rushley et al. 2018).

To see how the NNs depend on these important control parameters, both the GCRM and SPCAM training datasets are partitioned into bins of LTS and Q. For the GCRM training dataset, we partition the points in the tropics and subtropics (23S – 23N); the humidity bins are 2 mm wide, and the LTS bins are 0.5 K wide. For SPCAM, we use 20 bins evenly spaced between 0 mm and 40 mm for mid-tropospheric moisture and from 7K to 23K for LTS. These ranges are chosen to roughly span the observed ranges of samples within the tropics (see Figure 1a,b). In both cases, the NN's inputs x are averaged over these bins. Denote this average by $E[\bar{\mathbf{x}}|Q, \mathrm{LTS}]$. In the SP-CAM case, these averages are performed over the validation set. A parameterization's sensitivity to the variables Q and LTS is given by

$$f(Q, S) = f(E[\bar{\mathbf{x}}|Q, \mathrm{LTS}]; \varphi), \tag{5}$$

where φ are the parameters of the neural network. Because f is nonlinear, this is not equivalent to taking the average of the NN's outputs over the bins. Rather, it tests the nonlinear sensitivity to systematic changes in its inputs. In the sections below, we will also plot the bin-averages of the "true" apparent heating $E[Q_1|Q, \mathrm{LTS}]$ and moistening $E[Q_2|Q, \mathrm{LTS}]$ to indicate the fraction the true variability across bins the NN is able to predict successfully.

b. Linear Response Functions

The method above shows how a ML parameterization depends nonlinearly on a few inputs, but it is

difficult to extend to the full input space of a parameterization. To do this, we instead use the LRF or saliency map. The LRFs in this study will be computed from the output of a neural network (NN). By using a nonlinear continuous (and almost everywhere differentiable) model such as a NN, we can compute the local linear response of convection for a variety of base-states.

LRFs have already been employed to develop machine learning parameterizations. For instance Brenowitz and Bretherton (2019) computed LRFs to analyze what was causing their neural network parameterizations to produce unstable dynamics when coupled to a GCM. For most of this analysis, we linearize the GCRM-trained NN about the tropical mean profile since this is a region where the coupled GCM-NN instabilities of this scheme develop. This state is not a radiative-convective equilibrium (RCE), so some positive modes of the linearized response function likely represent a decay to a true RCE state. However, we assume this equilibration process is slower than the coupled GCM-NN blow-ups, so that LRFs can still reveal mechanisms behind the latter. Developing machine learning parameterizations with a stable radiative-convective equilibrium is a task for future research.

c. Coupling to Two-dimensional Linear Dynamics

While LRFs provide insights into how a parameterization affects a single atmospheric column in radiative convective equilibrium, they cannot alone predict the feedbacks that will occur when coupled to the environment. This coupling is thought to play a critical role in the catastrophic instability because NNs that produce accurate single column model simulations (Brenowitz and Bretherton 2018) can still blow up when coupled to the wind field simulated in a GCM (Brenowitz and Bretherton 2019). While we ultimately suspect that nonlinearity causes coupled simulations to catastrophically blow up once the NNs are forced to make predictions outside of the training set, we hypothesize that the initial movement towards the edge of the training manifold is inherently linear. In particular, we suspect it arises from interactions between the parameterization and small-scale gravity waves, which are the fastest modes present in large-scale atmospheric dynamics. This interaction has been extensively studied in the literature (Hayashi 1971; Majda and Shefter 2001), and is known to cause deleterious effects such as grid-scale storms when using traditional parameterizations based on a moisture convergence closures (Emanuel 1994).

We now derive the linearized dynamics of a ML parameterization coupled to gravity waves. Like many past works in convectively coupled waves, we assume that the flow is two-dimensional (vertical and horizontal) and neglect the influence of rotation. We further assume the dynamics are anelastic and hydrostatic. Further assuming that the mean winds are zero, the linearized anelastic equations in terms of humidity q, static energy s, and vertical velocity w perturbations are written as

$$\begin{aligned} q_t + \bar{q}_z w &= Q_2', \\ s_t + \bar{s}_z w &= Q_1', \\ w_t &= -\left(A^{-1}B\right)_{xx} - dw. \end{aligned} \tag{6}$$

The final equation is obtained by taking the divergence of the horizontal momentum equation and eliminating the pressure gradient term using hydrostatic balance and the anelastic nondivergence condition (see Appendix (a) for more details).

Here, A is a continuous elliptical vertical differential operator defined by $Aw = \frac{\partial}{\partial z}\left(\frac{1}{\rho_0}\frac{\partial}{\partial z}(\rho_0 w)\right)$. When endowed with rigid lid boundary conditions, $w(0) = w(H) = 0$, this linear operator can be inverted to give A^{-1}. In practice, we discretize these continuous operators using finite differences so that these operations can be performed with matrix algebra.

Since we are focused on free-tropospheric dynamics, we have neglected the virtual effect of water vapor and approximated the buoyancy by $B = gs/\bar{s}$. The momentum damping rate is fixed at $d = 1/(5\,\mathrm{d})$. We discretize this equation using centered differences (more details in Appendix (b), and assume rigid lid ($w = 0$) boundary conditions at the surface and top of atmosphere like Kuang (2018).

The perturbation heating Q_1' and moistening Q_2' are a linear transformation of the perturbed state which could be non-local in the vertical direction. In particular,

$$Q_1'(z) = \frac{1}{H}\int_0^H \left[\frac{\partial Q_1(z)}{\partial q(z')}q(z') + \frac{\partial Q_1(z)}{\partial s(z')}s(z')\right] dz',$$

and similarly for Q_2'. Upon discretizing this integral onto a fixed height grid and concatenating the s and q fields into vectors $\boldsymbol{s} = [s(z_1), \ldots, s(z_n)]^T$, and similarly for q, Q_1, and Q_2, this continuous formula can be written as

$$\begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{pmatrix} = \begin{pmatrix} M_{ss} & M_{sq} \\ M_{qs} & M_{qq} \end{pmatrix} \begin{pmatrix} \mathbf{s} \\ \mathbf{q} \end{pmatrix}.$$

The subblocks of the matrix (e.g. M_{qq}) encode the linearized response function for the fixed height grid

$z_1, \ldots, z_n$. Then, assuming the solution is a plane wave in the horizontal direction with a wavenumber k, (6) can be encoded using the following matrix form

$$\frac{\partial}{\partial t}\begin{pmatrix}\mathbf{q}\\ \mathbf{s}\\ \mathbf{w}\end{pmatrix} = \mathcal{T}\begin{pmatrix}\mathbf{q}\\ \mathbf{s}\\ \mathbf{w}\end{pmatrix}, \tag{7}$$

where the linear response operator $\mathcal{T}$ for total water, dry static energy and vertical velocity is given by:

$$\mathcal{T} = \begin{pmatrix} M_{qq} & M_{qs} & \mathrm{diag}(\bar{\mathbf{q}}_z) \\ M_{sq} & M_{ss} & \mathrm{diag}(\bar{\mathbf{s}}_z) \\ 0 & -gk^2A^{-1}\mathrm{diag}(\bar{\mathbf{s}})^{-1} & -dI \end{pmatrix}. \tag{8}$$

Here, I is the identity matrix, and diag creates a matrix with a diagonal given by its vector argument.

The spectrum of (8) can be computed numerically for each wavenumber k to obtain a dispersion relationship. Appendix (b) derives the discretization we use for the elliptic operator A. The real component of any eigenvalue λ of $\mathcal{T}$ is the growth rate, and the phase speed can be recovered using the formula $c_p = -\Im\lambda/k$. The eigenvectors of $\mathcal{T}$ describe the vertical structure of the wave mode in terms of s, q, and w. Let $\mathbf{v}$ be such an eigenvector, then the wave's structure over a complete phase of oscillation can be conveniently plotted in real numbers by showing $\Re\{\mathbf{v}\exp i\phi\}$ for $0 \le \phi < 2\pi$. In the sections below, we will show the phase speed and growth rates for every single eigenmode over a range of wavenumbers. Then, we can visualize the vertical structures of a few particularly interesting modes, such as unstable propagating modes or standing waves.

4. Regularization

As we have seen, NNs are often numerically unstable when coupled to atmospheric fluid dynamics, and much of our recent work has focused on solving this central challenge. One reason in the case of training data from coarse-grained simulations may be causality issues. In the GCRM, there is a strong correlation between an input variable—upper tropospheric total water—and an output variable—precipitation. This correlation would be expected because deep convection lofts moisture high into the atmosphere and total water includes cloud water and ice. However, using it as a closure assumption would violate a physical causality argument that moist atmospheric convection is triggered by lower atmospheric thermodynamic properties.

Brenowitz and Bretherton (2019) found that reducing the potential for spurious causality by ablating both the temperature above level 15 (375 mb) and humidity above level 19 (226 mb) from the input features of an NN parameterization results in a stable scheme. We will refer to these ablated inputs as "upper atmospheric data". It is ad-hoc, but works consistently. This was discovered using a LRF analysis (see Sec. 3b), which demonstrates that ML interpretability techniques have already significantly aided the development of ML parameterizations.

For SPCAM trained NNs, stability has also been a lingering challenge. Unfortunately, removing upper atmospheric inputs as prescribed by Brenowitz and Bretherton (2019) did not stabilize these NNs (see e.g. Movie S2). We speculate that the instabilities from SPCAM are linked to inevitable imperfections of NN fit, exacerbated by limited hyperparameter tuning. Nonetheless, using some of the same interpretability techniques described above, we have developed an "input regularization" technique for stabilizing SP-trained NNs.

Just as with the upper atmospheric ablation for the GCRM NNs, this technique was discovered using a LRF analysis. We noticed that directly calculating the LRF of SPCAM-trained NNs via automatic differentiation results in noisy, hard-to-interpret LRFs (top line of Figure S2). When coupled to 2D dynamics, these LRFs produce unphysical stability diagrams, with unstable modes with phase speeds greater than $300\,\mathrm{m\,s^{-1}}$ even for the "NN-stable" network (bottom line of Figure S2).

The LRF's noisiness indicates that the NN responds nonlinearly to small changes in its inputs. While this behavior could be a desirable aspect of the NN convective parameterization (Palmer 2001), it prevents a clean interpretation of the NN parameterization through automatic differentiation about an individual basic state alone. On the other hand, the SAM-trained NNs have a much smoother response, a discrepancy that could be due to differences in the network architecture, training strategy, or the underlying training data. Understanding this discrepancy between two networks with the same function trained on datasets with obvious similarities is an important future challenge.

The "NN-stable" SPCAM network performs stably when interactively coupled to GCM dynamics, suggesting it is not overly sensitive to larger perturbations. This motivates feeding an ensemble ($\mathcal{X}$) of randomly perturbed inputs to the SPCAM NN parameterization, which then outputs an ensemble ($\mathcal{Y}$) of predictions that we may average to more cleanly understand the NN's behavior.

For concreteness, we now apply "input regularization" to our initial problem, i.e. the unstable behavior of "NN-unstable". This requires 5 steps:

1. We track the (longitude, latitude) coordinates of the instability (see Movie S1) back in time to

identify the base state $\boldsymbol{x}_{\mathbf{unstable}}$ leading to the instability. For simplicity, we choose the earliest timestep for which the perturbation responsible for the crash produces a maximum in the convective moistening field (Q_2).

2. We construct an input ensemble $\{\mathbf{x}^{(i)}{}_{i=1,2,\ldots,n}\}$ of n members by perturbing the input $\boldsymbol{x}_{\mathbf{unstable}}$ producing the instability using a normal distribution $\mathcal{N}$ of mean 0 and standard deviation σ_{reg}:

$$x_j^{(i)} = (1 + z_j^{(i)})\mathbf{x}_j^{\mathrm{unstable}}, \quad z_j^{(i)} \sim \mathcal{N}(0, \sigma_{\mathrm{reg}}).$$

We refer to σ_{reg} as "regularization amplitude" (in %): the larger σ_{reg} is, the broader the ensemble of exact inputs $\boldsymbol{x}$ will be in the input ensemble.

3. We feed each member $\mathbf{x}^{(i)}$ of the input ensemble into the NN parameterization, producing an ensemble of outputs $\{\mathbf{y}^{(i)}\}_{i=1,\ldots,n}$.

4. We calculate the LRF about the input ensemble by automatically differentiating each input-output pair before taking the ensemble-mean of the resulting LRFs:

$$\mathrm{LRF}_{\mathrm{reg}} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial y^{(i)}}{\partial x^{(i)}}$$

5. We couple the "regularized" LRF to our two-dimensional wave coupler to calculate a stability diagram representative of the NN's behavior near the "regularized" set of inputs $\{\mathbf{x}^{(i)}{}_{i=1,2,\ldots,n}\}$.

Adding significant spread to the input vector may push the NN outside of its training set where large biases are expected, so we do not advocate for such an approach as a general strategy to stabilize SPCAM-based NNs. That being said, we will exploit this regularization technique to investigate the robustness of our interpretability framework by comparing offline predictions to online prognostic failure modes. That said, adding Gaussian noise to inputs is commonly used to regularize NNs during training.

5. Results

a. The Onset of Deep Convection in ML parameterizations

Figure 1 shows the two dimensional binning of the combined tropical and subtropical data in mid-tropospheric humidity Q and lower tropospheric stability (LTS) space for all latitudes equatorward of 22.5 degrees. For the GCRM simulation, the distribution is bi-modal, with many high-moisture low-stability samples—presumably from the tropics—and another peak for lower moistures of about 7 mm from the sub-tropics. The SPCAM simulation is moister, with a modal Q around 30 mm compared to less than 20 mm in the GCRM simulation. This is not surprising because the peak SST of the SPCAM simulation is 2-3 K warmer.

We use net precipitation—surface precipitation (P) minus evaporation (E)—as a proxy for deep convection, because it is predicted by both NNs as the column integral of the apparent drying $-Q_2$, and because it clearly distinguishes between regimes of little precipitation ($P < E$) and substantial precipitation ($P > E$). The GCRM results are for an unablated NN. Both training datasets, and the ML parameterizations trained from them, predict that the net precipitation increases dramatically with mid-tropospheric humidity. A similarly nonlinear dependence has been documented in numerous studies (Bretherton et al. 2004; Neelin et al. 2009; Rushley et al. 2018). The net precipitation depends less strongly on LTS, but for intermediate values of moisture around 15-20 mm, the stability is an important control. The difference between the bin-averaged net precipitation and the NNs predictions with the bin-averages are relatively small ($\sim$ 10 mm/day). We conclude that the machine learning parameterizations depend smoothly and realistically on Q and LTS.

How does the vertical structure of the input variables and the predicted heating and moistening vary with Q and LTS? Figure 2 shows vertical profiles of these quantities for varying LTS binned over tropical grid columns with $20 \leq Q < 22$ mm for the GCRM simulation while Figure 3 shows the LTS dependence for $33.7 \leq Q < 35.7$ mm for the moister SPCAM simulation. The humidity reaches much higher in the atmosphere for low stabilities, and the lower tropospheric must offset these gains to maintain a constant Q. For the GCRM, the predicted heating and moistening switch from shallow to deep profiles as LTS decreases. However, the overall magnitude of moistening is relatively unchanged, consistent with Figure 1c. Thus, LTS controls the height of the predicted convection more than its overall strength. That said, it is unclear whether these changes are a direct response to the lower-tropospheric temperature structure or controlled by the simultaneous changes in the humidity profile (Figures 2a and 3a). On the other hand, the SPCAM NNs fail to predict such a clear deepening of convection with decreased stability. This could owe to the Q-bin we selected for this analysis, a more fundamental difference in moist

convection between the SPCAM and GCRM simulations, or the substantially warmer SSTs in the former. Nonetheless, each NN faithfully represents the convective sensitivity of its own training dataset.

The predicted heating and moistening vary in a similar way to the bin-averaged Q_1 and Q_2 profiles, but the latter are more sensitive to the LTS than the NN. Note that the NNs make their predictions with a single input profile whereas the bin-averaged Q_1 and Q_2 are statistical averages of many individual heating and moistening profiles. Thus, these figures demonstrate how the NN's prediction are sensitive to systematic changes in the input variables, whereas the bin-averaged heating and moistening show a statistical, but potentially non-causal link between heating, moistening, mid-tropospheric humidity, and LTS in the data.

Figures 4 and 5 show how the mid-tropospheric humidity controls the vertical structure of the input and response variables, for the GCRM and SPCAM data respectively. The LTS is fixed between 9 K and 10 K for the GCRM run and 11 K and 12 K for SPCAM, both of which are unstable ranges. The overall amount of water changes dramatically for this change in both simulations because Q is highly correlated with the total precipitable water in an atmospheric column. Both NNs predict cooling and moistening near the top of the boundary layer for the drier profiles ($z = 2000$ km) due to shallow clouds. Once a threshold Q is reached, the sign flips and the machine learning parameterization predicts increasingly deep heating and moistening. For the three moistest profiles, the heating and moistening dramatically strengthen with little change in vertical structure. The predicting tendencies are again similar to their bin averages.

b. Stabilizing via Ablation of Inputs

Brenowitz and Bretherton (2019) obtained a stable scheme by training their NN without input from the upper atmospheric temperature or humidity. However, they did not examine whether it was ablating the atmospheric temperature, humidity or both that prevented numerical instability. The term "ablate" is used as an analogy to neuroscience research on how removing (i.e. ablating) brain tissue affects animal behavior. In this section, we use the wave-coupling framework introduced in Section 3c to explore the independent effects of ablating temperature and humidity. Because this wave coupling is performed for one wavenumber at a time it is much more computationally affordable than a full non-linear simulation, but still hints at how the NN will perform in coupled simulations.

To study this further, we first compute the LRF of an NN trained with all atmospheric inputs (Brenowitz and Bretherton 2019, cf. Figure 1). Brenowitz and Bretherton (2019) chose a lower humidity level because the humidity in the upper troposphere is vanishingly small, while the temperature remains of the same order of magnitude. Then, the upper atmospheric humidity and/or temperature inputs are sequentially knocked out by inserting 0 in the corresponding entries of the LRF. This section studies the following configurations:

- All atmospheric inputs (unablated),
- No humidity above level 15 (375 mb),
- No temperature above level 19 (226 mb),
- No humidity above level 15 nor temperature above level 19 (fully-ablated).

Figure 6 shows the dispersion relationships resulting after each of these ablations, a zoomed-in version of which is shown in Figure 7. With all atmospheric inputs, there are numerous propagating modes with phase speeds between 10 m/s and 25 m/s with positive growth rates . These modes become increasingly unstable for shorter wavelengths. When the upper-atmospheric humidity is ablated, there still remain numerous unstable modes, including a ultra-fast 100 m/s propagating instability. The results when the upper-atmospheric temperatures are ablated, but not the upper moisture, are similar to the full atmospheric input. Finally, ablating both the temperature and humidity inputs from the upper atmosphere removes many unstable propagating modes. The remaining unstable modes are either stationary or have very slow phase speeds. This suggests that ablating both humidity and temperature is necessary for nonlinear simulations to be stable when using an ML trained on coarse-grained GCRM data that is at 3-hourly time resolution. Table 1 summarizes these results.

The phase-space structure of two modes is shown in Figures 8 and 9. Figure 8 shows a standing mode of the LRF-wave system in the fully-ablated case. For a wavelength of 628 km this mode has an e-folding time of $1/.32 \approx 3$ d and zero phase-speed. It primarily couples vertical velocity (panel a) to the total water mixing ratio (panel c), with upward velocity (negative ω) corresponding to anomalous moistening and moisture (panel c). The heating and cooling (panel d) nearly perfectly compensates for the vertical motions.

The free tropospheric heating and temperature are relatively uninvolved, but boundary layer heating

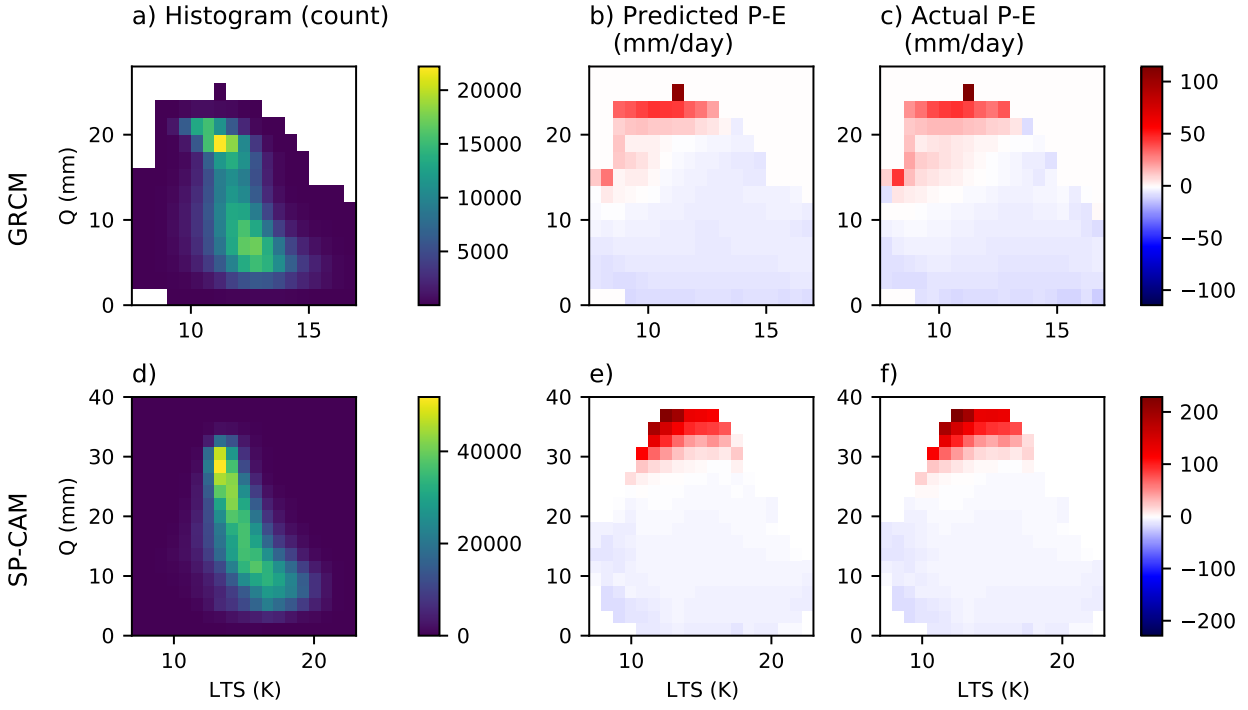

FIG. 1. Binning in LTS and Moisture Space. The first row shows, for the GCRM, (a) population of each LTS/Q bin, predicted net precipitation (b) and (c) the $-\langle Q_2 \rangle \approx P - E$ from the training dataset. $\langle \cdot \rangle$ is the mass-weighted vertical integral. The second row (d-f) are the corresponding results for SPCAM.
t

and temperature anomalies do have a large magnitude. Thus, the standing mode appears to be mostly a moisture mode. Similar modes are responsible for convective self-aggregation in large-domain CRM simulations of radiative-convective equilibrium (Bretherton et al. 2005) and are thought to be important for large-scale organized convection such as the Madden Julian Oscillation (Sobel and Maloney 2013; Adames and Kim 2015). Kuang (2018) found that a similar mode is unstable only when the LRF includes radiative effects. In contrast to Kuang's study, a NN trained to predict $Q_1 - Q_{rad}$, where Q_{rad} is the coarse-grained radiative tendency of the 4 km model, also predicted an unstable standing mode (not shown). However, it is not clear that this method reliably separates the convection from the radiation because of the noisiness inherent in the GCRM budget residuals and training method.

The LRF-wave analysis can also identify spurious wave-like modes which could contribute to numerical instability in coupled simulations. Figure 9 shows a mode of the un-ablated model which has planetary-scale wavelength of 6283 km, phase speed of 44 m/s and fast growth rate of 1.29 d^{-1}. This mode is stable for shorter waves, so we have chosen a longer wavelength. The moisture, humidity, moistening, and drying tendencies are in phase with each other but in quadrature with the ω. The vertical motion tilts upward and away from the direction of propagation. Both humidity and temperature anomalies contribute significantly to the wave's structure, but have strange vertical structures. The upper-atmosphere temperature anomalies are strongly coupled to moisture anomalies in the free troposphere, which have a complex vertical structure. These structure are not reminiscent of known modes of convective organization, and such a mode could explain the instability this scheme causes when coupled to a GCM.

c. *Stabilizing Gravity Wave Modes via Regularization of Inputs*

It is natural to wonder whether the wave-coupling framework successfully applied to the GCRM data can also predict prognostic-mode failures in the SP-CAM simulation. The answer is not obvious since ML climate models trained on vastly different data sets exhibit different forms of instability. For instance, the SPCAM-trained "NN-unstable" model tends to go unstable outside of the tropics, and more

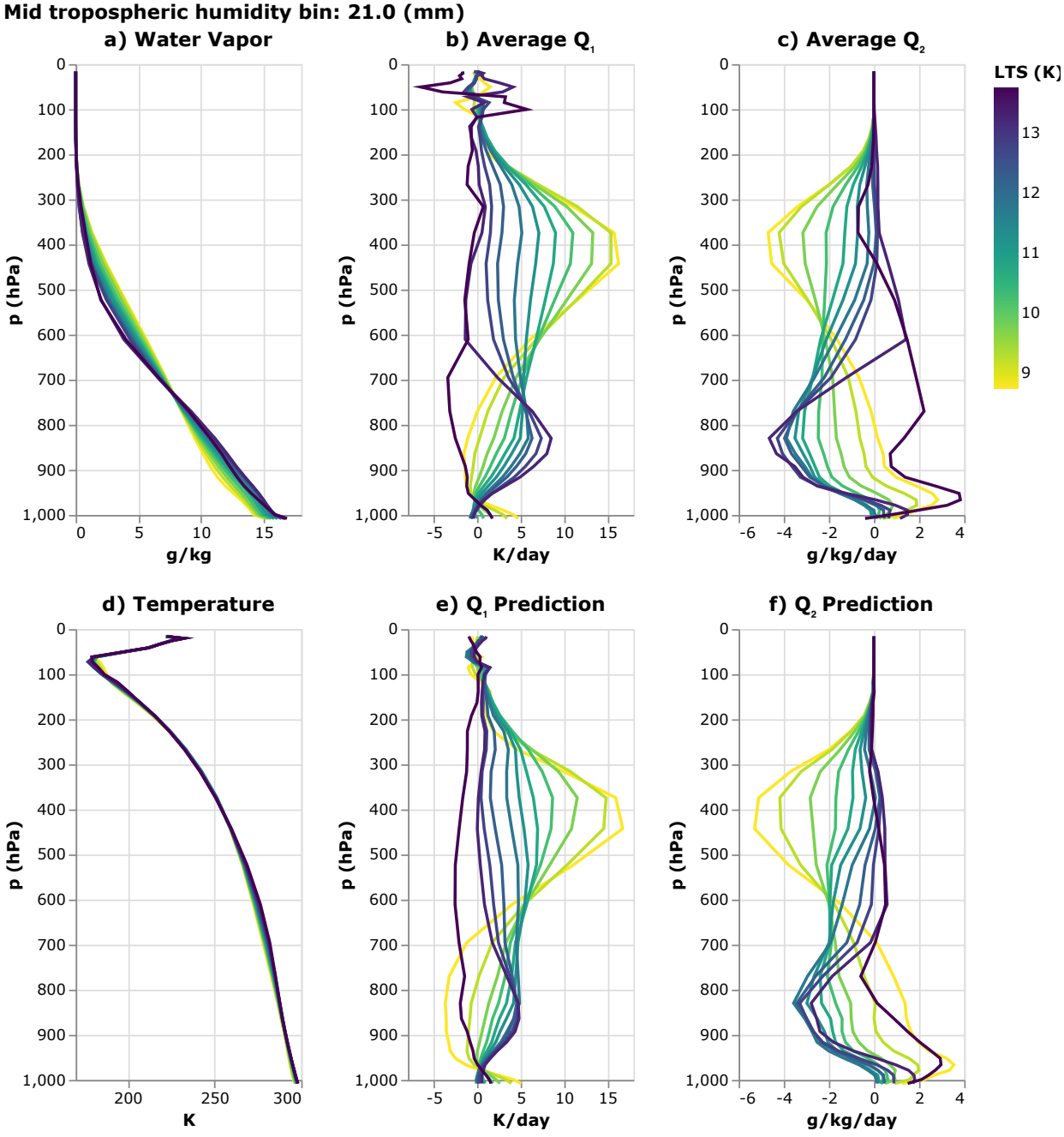

FIG. 2. Deepening of convection for SAM, varying the LTS with mid-tropospheric humidity between 20 and 22 mm. Shows conditional averages $E[\cdot|Q, LTS]$ over the data of specific humidity (a), heating Q_1 (b), moistening Q_2 (c), and temperature (d). The predicted heating (e) and moistening (f) for the conditionally averaged input data are also shown.

gradually than the GCRM-trained climate model diagnosed above. This suggests that different ML-based models go unstable for different root causes. Consistent with this view, despite successfully stabilizing the GCRM-trained NN, input ablation does not stabilize the SPCAM-trained "NN-unstable": that NN still crashes after ~ 3 days when coupled to CAM, even after ablating the input vector's top half components (15 first components of water vapor and temperature profiles, from $p = 0$hPa to $p = 274$hPa, see e.g. Movie S2). As an alternative to ablation, we introduce a new empirical technique, "input regular-

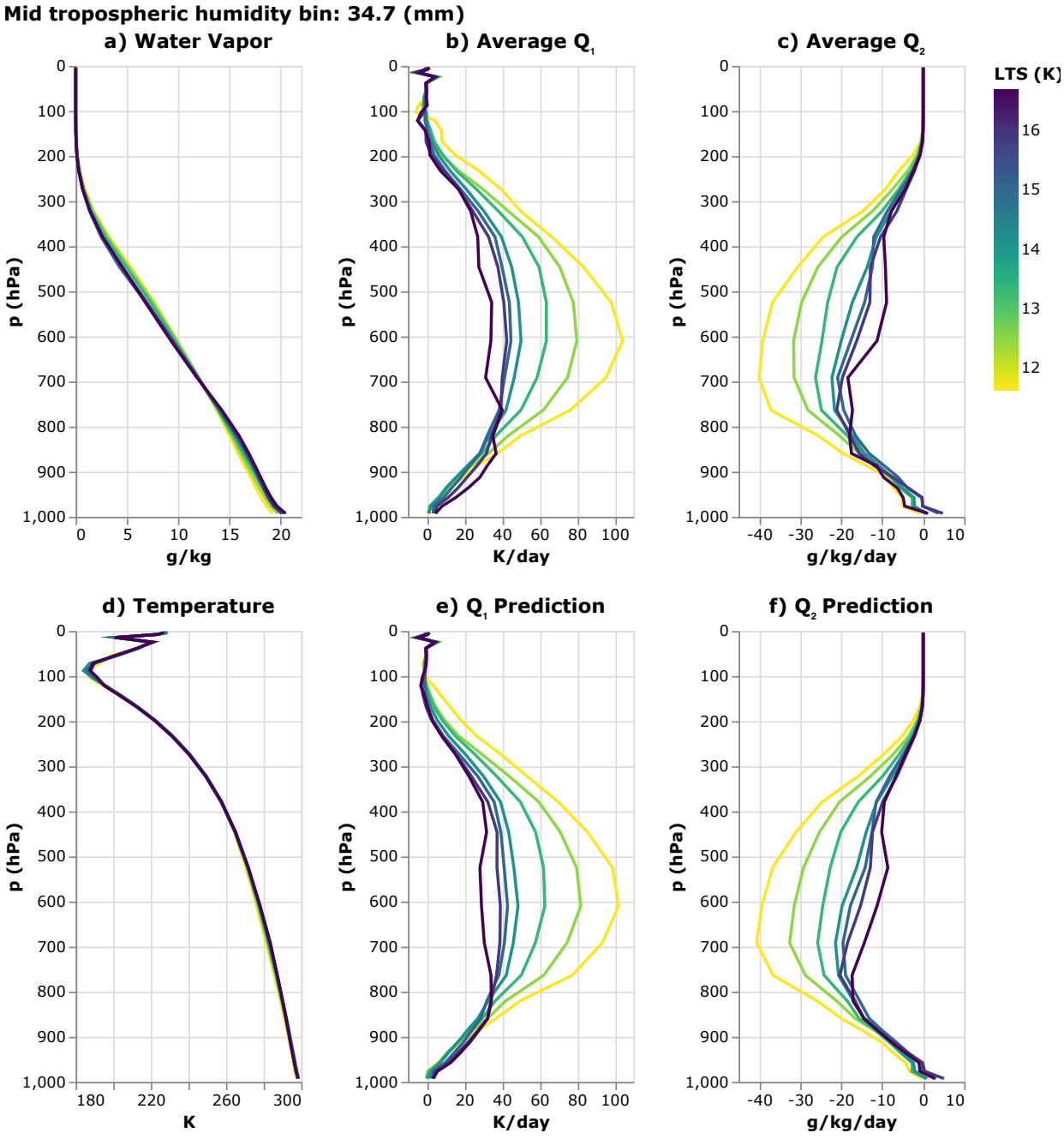

FIG. 3. Deepening of convection for SPCAM, varying the LTS with mid-tropospheric humidity between 33.7 and 35.7 mm. Same panels as Figure 2.

ization", that can improve the SPCAM-trained NN's prognostic performance. We then show that the computationally affordable wave-coupling introduced in Section 3c can predict how much regularization is required to stabilize coupled GCM-NN simulations.

We begin by revisiting the physical credibility of NN-unstable compared to NN-stable using the diagnostics that previously revealed the causal ambiguity endemic to the GCRM-trained NN. From this view, we expect NN-unstable to struggle in prognostic mode, since it exhibits significant positive heating and moistening biases (see Figure S1) for large mid-tropospheric moisture (¿20mm) and low LTS (¡12K), conditions favorable for deep convection. Positive

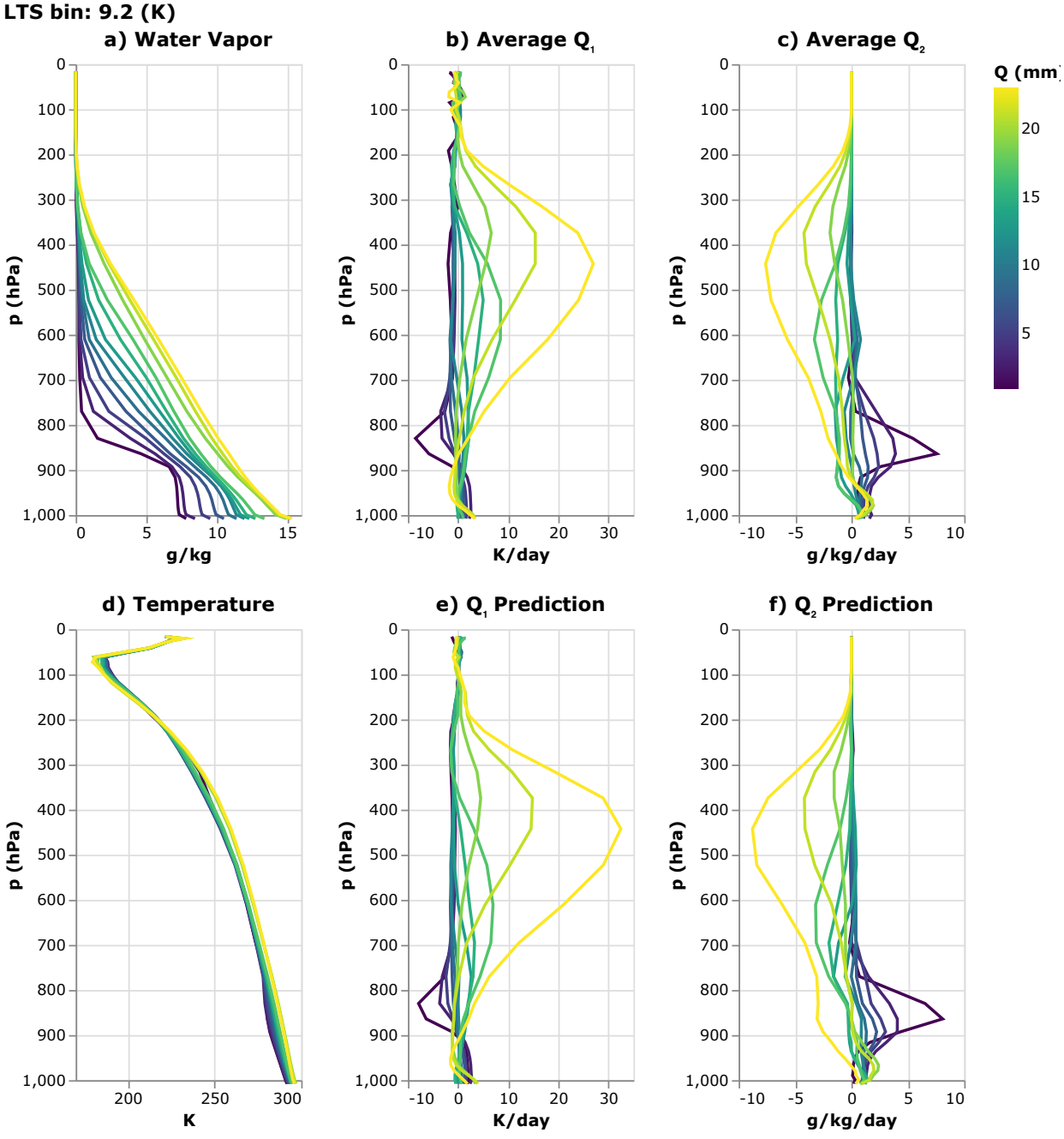

FIG. 4. Strengthening of convection for SAM, varying mid-tropospheric moisture Q for LTS between 9.0 and 9.5 K. Same panels as Figure 2.

convective moistening biases in moist regions may then lead to instability through spurious moistening of growing water vapor perturbations.

Next, we study the effect of increased input regularization as described in Sec 4, focusing on the relation between offline prediction and online performance. Figure 10 shows that input regularization effectively controls stability properties. The top row shows the regularized LRFs calculated about the base state $\boldsymbol{x}_{\mathbf{unstable}}$ for regularized amplitudes of 1%, 10%, and 20%. A preliminary stability analysis, including direct eigenvalue analysis and simple dynamical coupling using the strict weak-temperature gradient approximation (Beucler et al. 2018), indi-

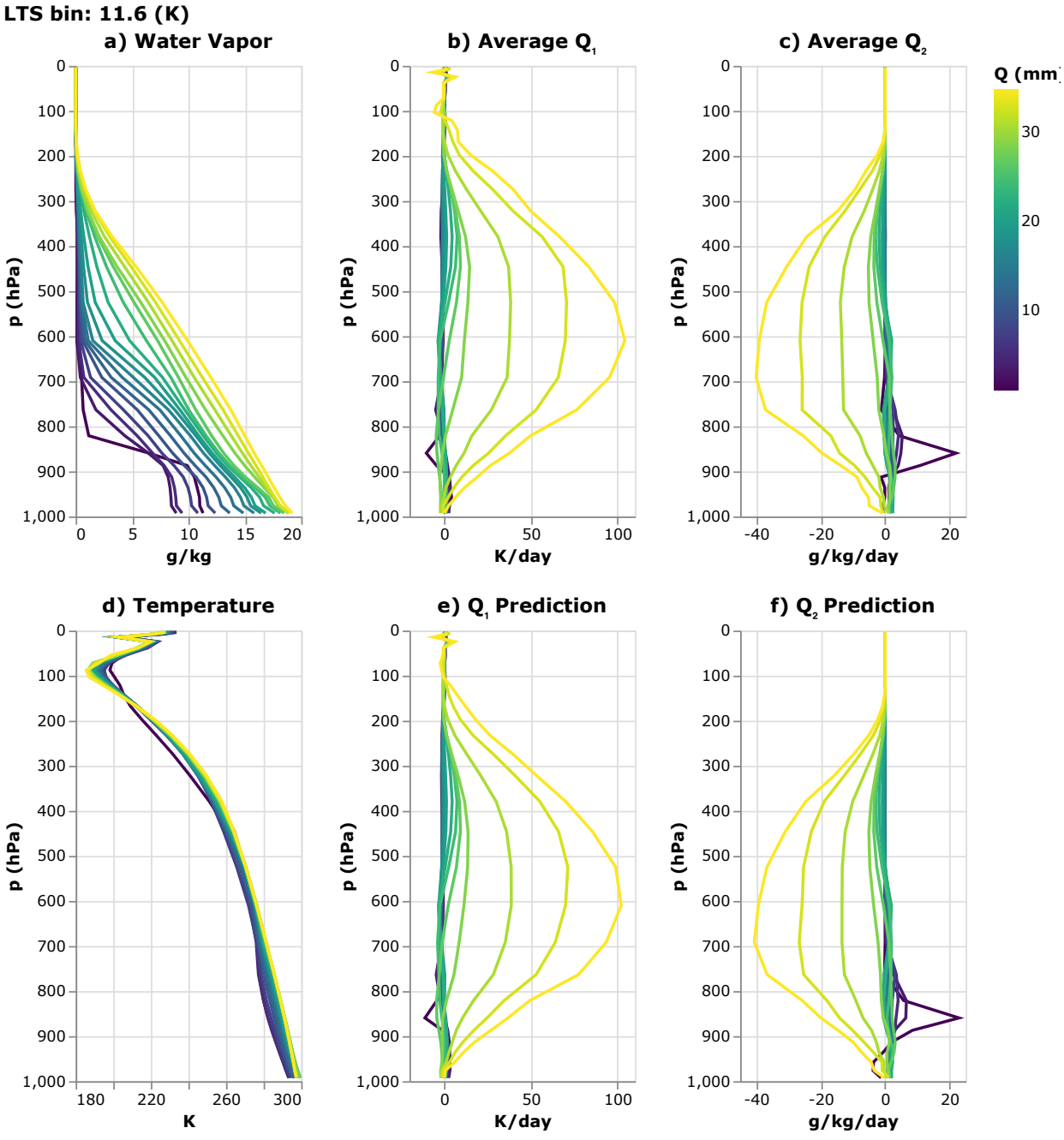

FIG. 5. Strengthening of convection for SPCAM, varying mid-tropospheric moisture Q for LTS between 11 and 12 K. Same panels as Figure 2.

cates that NN-unstable's damping rates are closer to 0 than NN-stable's damping rates. Although these results suggest that NN-unstable is unable to damp developing instabilities quickly enough, the stability diagram from our wave-coupler (Figure 10f) can provide a more extensive description of developing instabilities if we use an input ensemble tightly regularized (1%) about the base state $\mathbf{x}^{\text{unstable}}$. While NN-stable only has a few slowly-growing modes about the unstable base state (Figure 10e), NN-unstable exhibits a myriad of fast-growing ($\sim$ 1day^{-1}) modes (Figure 10f) propagating at phase speeds between 5m/s and 20m/s, indicated with the light-blue background. Hence our wave-coupling framework can anticipate

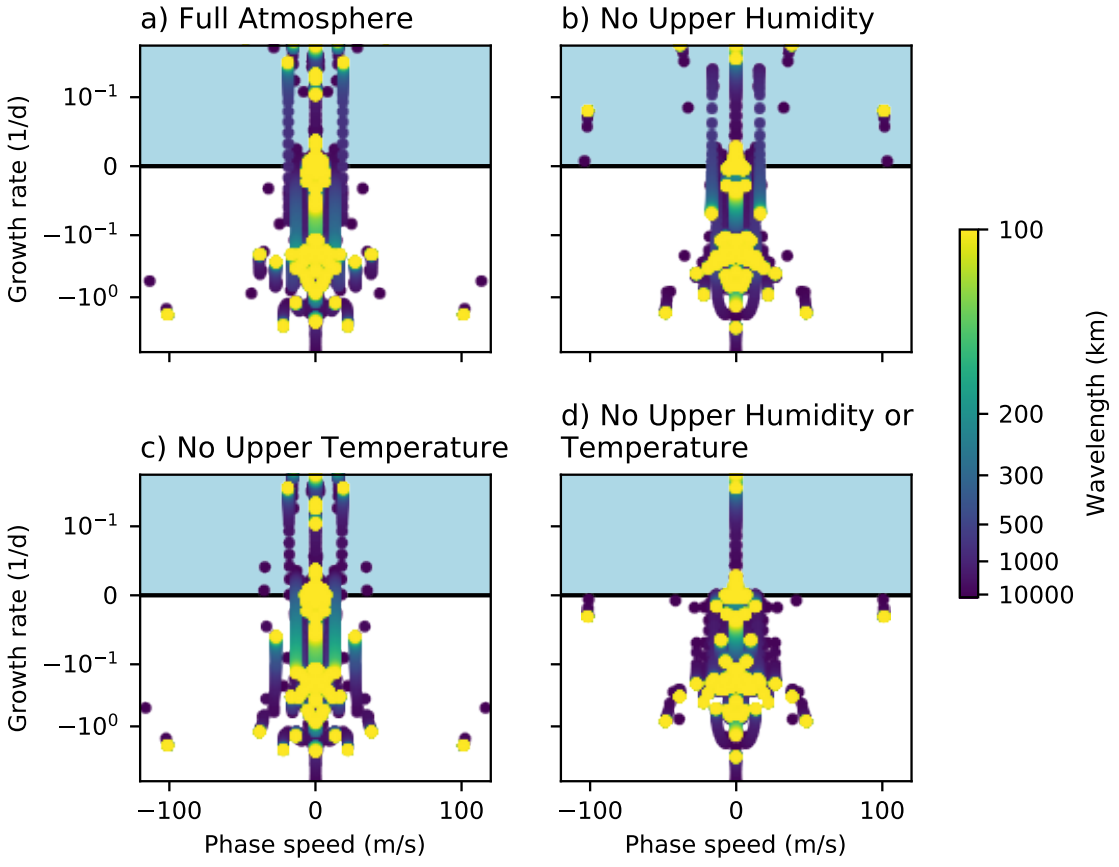

FIG. 6. Wave spectra with and without upper-atmospheric input for the GCRM-trained NN. a) full atmospheric input, b) lower-tropospheric humidity, c) lower-tropospheric temperature, and d) lower-tropospheric humidity and temperature. The light-blue background indicates where the phase speed is greater than $|c_p| > 5\,\mathrm{m/s}$ and the growth rate is positive. The wavelength is defined as $2\pi/k$. This box is more visible in the zoomed-in plot (cf. Figure 7). Waves inside of this region are likely responsible for instability.

Upper atmospheric humidity input (above level 15)	Upper atmosphere temperature input (above level 19)	Coupled blow-up	Maximum phase speed	Stability
Yes	Yes	Yes	50 m/s	yes
Yes	No	Yes	100 m/s	yes
No	Yes	Yes	50 m/s	yes
No	No	No	1 m/s	yes

TABLE 1. Summary of stability issues related to removing upper atmospheric inputs.. The second to last column shows the maximum phase speed over all modes with growth rates larger than $0.05\,\mathrm{d}^{-1}$.

the instability arising from coupling NN-unstable to CAM.

Interestingly, our wave-coupling framework further predicts that more input regularization should stabilize NN-unstable. The largest propagating growth rates decrease from $\sim 1\ \mathrm{day}^{-1}$ for a 1% regularization amplitude to $\sim 10^{-3}\,\mathrm{day}^{-1}$ for a 25% regularization amplitude (Figure 10h). To test this prediction, we run a suite of CAM simulations in which the host climate model's grid columns are each coupled to the mean prediction of a 128-member ensemble of NN predictions, formed via Gaussian-perturbed inputs, instead of the typical single NN prediction per grid cell. The amount of input spread is varied between 1% and 25% standard deviation across five experiments. To provide a measure of internal variability, each experiment is repeated across a four-member mini-ensemble launched from different SPCAM-generated initial conditions spaced five days

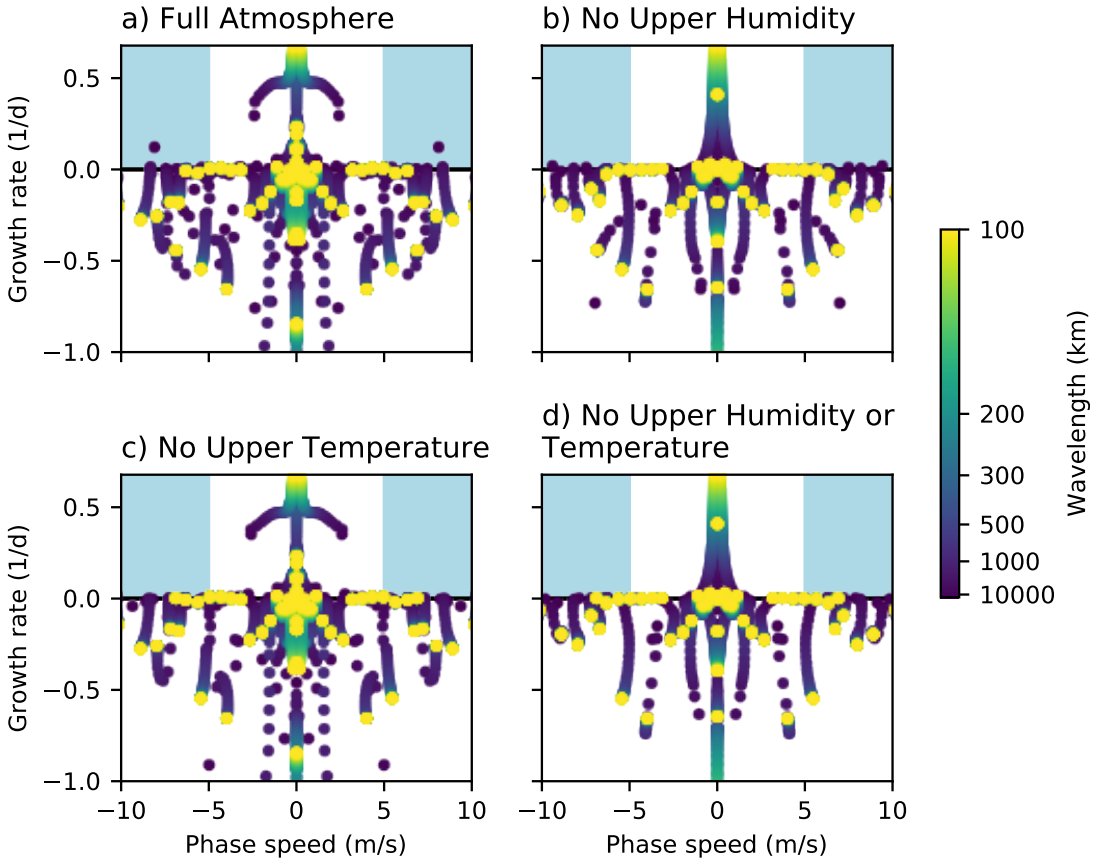

FIG. 7. Same as Figure 6 but only showing modes with a phase speeds less then 10 m/s.

apart. Figure 11 shows the time to failure of these runs for increasing regularization. With 15% or less input regularization, none of these simulations is able to run more than 21 days, consistent with the existence of many unstable modes in the 2D wave-coupler diagnostic; instead, variants of the same extratropical failure mode eventually occur. But when 20% spread is used to seed sufficient diversity in the input conditions, longer term prognostic tests become possible. Interestingly, the most dramatic effect of the input regularization in delaying the time to instability happens between spread magnitudes between 15% and 20% – this is consistent with the fact that the offline 2D wave coupler tests indicate an especially prominent shutdown of unstable modes in the vicinity of a 16% regularization magnitude. While we do not expect a simple linear model neglecting rotation to accurately predict non-linear, mid-latitude instabilities of a full-physics general circulation model, our results suggest that the offline diagnostics tools developed in this study apply to a wide range of instability situations.

6. Conclusions

Machine learning parameterizations promise to resolve many of the structural biases present in current traditional parameterizations by greatly increasing the number of tuning parameters. This massive increase in flexibility has two main drawbacks: 1) ML parameterizations are not interpretable a priori and 2) neural network parameterizations are often unstable when coupled to atmospheric fluid dynamics. This study addresses both of these points by developing an interpretability framework specialized for ML parameterizations of convection, and deepening analysis of the relationship between offline skill vs. online coupled prognostic performance.

By systematically varying input variables in a two-dimensional thermodynamic space, we have demon-

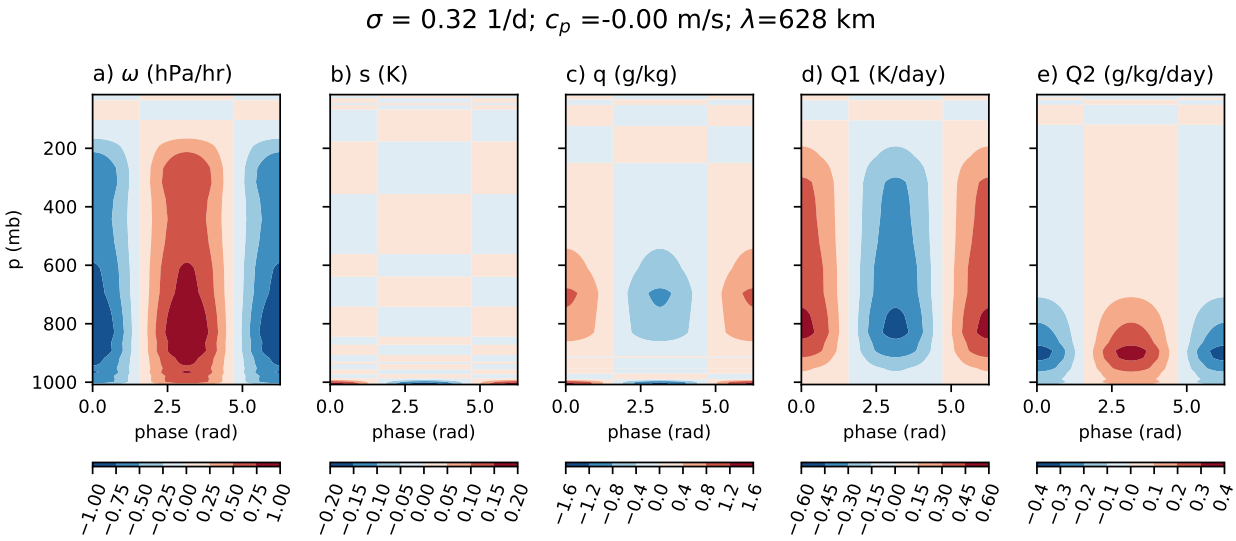

FIG. 8. Slow-moving instability reminiscent of a moisture mode. The vertical-horizontal structure for a single period of oscillation is shown for the pressure-velocity ω (a), the static energy s (b), the humidity (c), the heating (d), and the moistening (e).

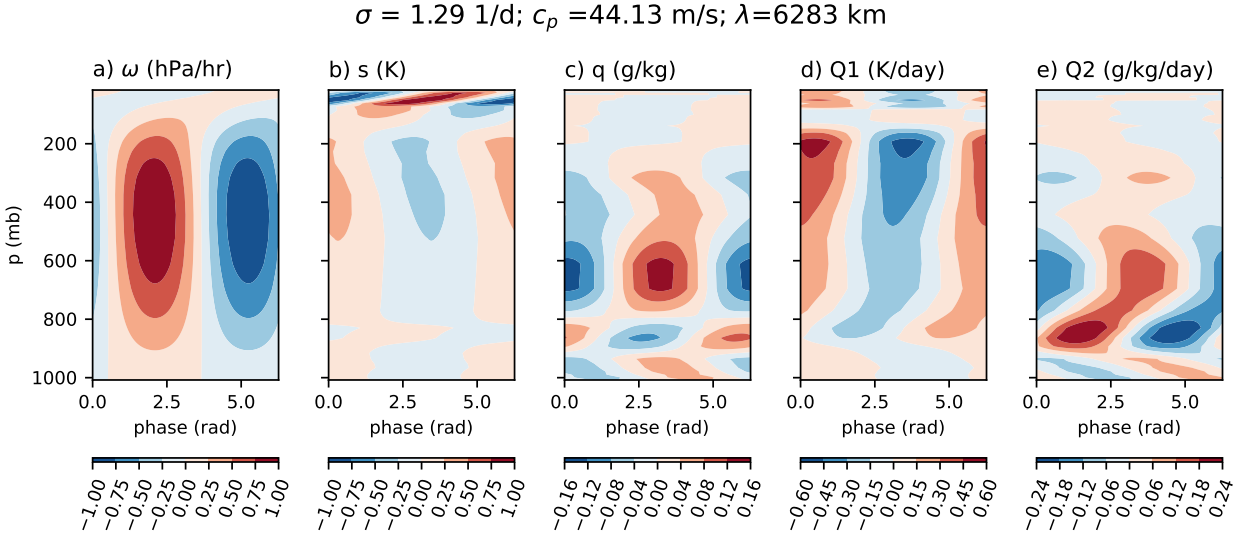

FIG. 9. Structure of a spurious unstable propagating mode. Same panels as Figure 8.

strated that two independent ML parameterizations behave as our intuition would expect. Increases in mid-tropospheric moisture tend to greatly increase the predicting heating and moistening, a widely documented fact (Bretherton et al. 2004), while increasing the lower-tropospheric stability effectively controls the depth of convection. These changes are consistent with the actual sensitivity present in our training dataset, and both the SPCAM and GCRM neural networks behave somewhat consistently, demonstrating the robustness of the machine learning approach to parameterizations. They both predict that net precipitation increases for moist and unstable columns, although the precise vertical structures of the their heating and moistening profiles differ. Future work in a similar spirit could easily build on these methods. For instance, a limitation of our approach here is that lower-tropospheric stability and mid-tropospheric moisture co-vary strongly because stable columns tend to be warmer and therefore carry more moisture. Therefore, the sensitivities we demonstrate are not entirely indepen-

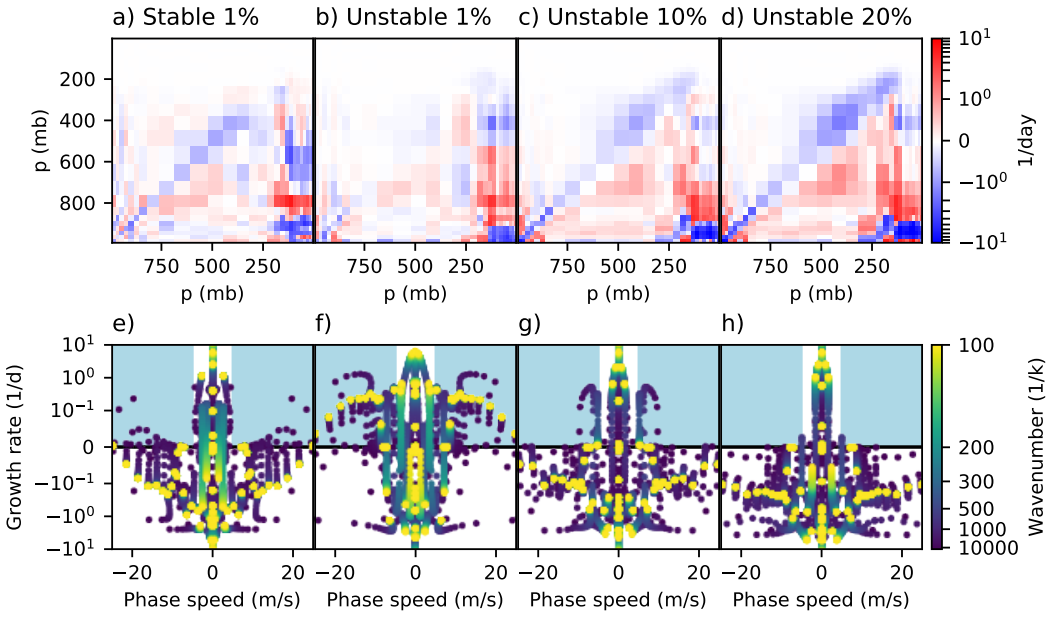

FIG. 10. (Top) Regularized ($\partial \dot{\boldsymbol{q}}_{\boldsymbol{v}}/\partial \boldsymbol{q}_{\boldsymbol{v}}$) linear response functions (in units 1/day) of NN-stable (leftmost column) and NN-unstable (three rightmost columns) for various regularization amplitudes (in %). (Bottom) Corresponding stability diagrams obtained by coupling the linear response functions to simple two-dimensional dynamics. As in Figures 6 and 7 the light-blue background indicates where the phase speed is greater than $|c_p| > 5\,\mathrm{m/s}$ and the growth rate is positive.

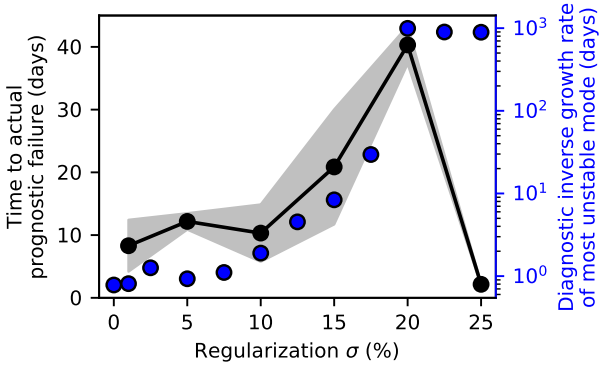

FIG. 11. Time to failure for an ensemble of prognostic tests (ensemble mean in black and standard deviation in grey) with varying "input regularization" spread magnitude (see text) compared to (blue) offline predictions from the most unstable mode derived from the 2D wave-coupler, defined as the maximal growth rate from the stability diagram that propagates with a phase speed of absolute value greater than 5 m/s.

dent. Instead of stability and moisture, the estimated inversion strength (Wood and Bretherton 2006) and lower-tropospheric buoyancy (Ahmed and Neelin 2018) may control convection more orthogonally, which would further ease the interpretation of such results, and is recommended in future diagnostics of this vein.

We have also developed an offline technique that shows some skill in predicting whether a given ML parameterization will be stable online (i.e. when coupled to atmospheric fluid dynamics) in both the GCRM-trained and SPCAM-trained limits, despite their many intrinsic differences. By coupling gravity-wave dynamics to the linearized response functions of an ML parameterization, one can compute the growth rates, phase speeds, and physical structure of the gravity wave modes associated with the parameterizations. We find that, in both SPCAM and SAM, propagating unstable modes are associated with the numerical instability in online coupled runs, and likely one root cause of instability. In stabilized versions of both schemes (using "input ablation" for SAM and "input regularization" for SPCAM), the propagating modes are all stable and prognostic tests run more stably. That said, these stabilization schemes are rather crude and may cause bias in climate modeling applications. That this framework does not include rotation or background shear indicates that interactions between gravity waves and the parametrized heating play a role in numerical instability. We speculate such instability causes coupled simulations to drift towards the boundary of their

training datasets. Once they reach this boundary, the neural networks are forced to extrapolate to unseen input data, which causes the final rapid blow-up.

Since this is also the first study to comprehensively compare NNs trained on coarse-grained GCRM data vs. SPCAM data, some comments on interesting differences worthy of future analysis are appropriate. Fully understanding why the SPCAM LRFs are so much noisier than the SAM LRFs will require further study of the many factors that could be involved, beyond obvious differences in the nature of the training data (e.g. hyperparameter settings and neural network architecture and optimization settings), especially since computing LRFs from traditional parameterizations typically requires some degree of regularization (e.g., Beucler 2019; Kuang 2010). Multiple techniques have been developed to smooth neural-network Jacobians, including averaging Jacobians derived from an ensemble of neural networks (Krasnopolsky 2007), taking the Jacobian of a single mean profile (Chevallier and Mahfouf 2001), or even multiplying the Jacobian by a "weight smoothing" regularization matrix (Aires et al. 1999). Here, we have introduced "input regularization" as an alternative strategy to ensemble-average Jacobians without having to train multiple networks. The regularized SPCAM LRFs are smoother, making them attractive for physical interpretation and easier to compare to GCRM LRFs, as well as easier to analyze using our wave-coupling framework. But we caution that, while incrementally helpful for full prognostic stability, "input regularization" should not be viewed as a solution to the instability problems in SPCAM-trained models. More attention on other strategies like formal hyperparameter tuning to efficiently uncover optimally skillful fits, which may be even more likely to perform stably online, is also warranted.

This wave-coupling analysis also has potentially interesting physical implications, but more work is required to compare and contrast NN-derived LRFs with other approaches (Kuang 2010, e.g.). Unlike Kuang (2010), our analysis is not conducted about radiative-convective equilibrium (RCE) profiles, but rather about base states close to the regions where we saw numerical instabilities. For a fair comparison, we should compute our spectra about an equilibrium state, if such exists for our schemes. Finally, the robustness of our spectra is hard to quantify, especially for phase-speeds and growth rates near zero. This is acceptable for discovering the spurious unstable propagating waves above, but making inferences about true physical modes will require a more rigorous statistical framework.

In summary, this manuscript has presented a pair of techniques that allow peering into the inner workings of two sets machine learning parameterizations. These tools have led to the development of new regularization techniques, and could allow domain experts to assess the physical plausibility of an ML parameterization. Reassuringly, ML parameterizations appear to behave according to our physical intuition, creating the potential to accelerate current parameterizations and develop more accurate data-driven parameterizations. We hope that these interpretability techniques will aid in discovering more elegant solutions to the coupled stability problem and facilitate a more detailed exploration of neural network hyperparameters (e.g. depth) than has been possible in the past.

Data availability statement. The wave-coupling code, bin-averaged data, and serialized linearized response functions are included in a source code repository to be uploaded with a suitable open source license to github and zenodo before final publication. The training dataset for the GCRM simulation have been archived on zenodo.org (Brenowitz 2019). The raw SPCAM outputs amount to several TB and are available from the authors upon request.

Acknowledgments. When starting this work, N.B. was supported as a postdoctoral fellow by the Washington Research Foundation, and by a Data Science Environments project award from the Gordon and Betty Moore Foundation (Award #2013-10-29) and the Alfred P. Sloan Foundation (Award #3835) to the University of Washington eScience Institute. C.B. was initially supported by U. S. Department of Energy grant DE-SC0016433. NB and CB acknowledge support from Vulcan Inc. for completing this work. TB and MP acknowledge support from NSF grants OAC-1835863, OAC-1835769, and AGS-1734164, as well as the Extreme Science and Engineering Discovery Environment supported by NSF grant number ACI-1548562 (charge numbers TG-ATM190002 and TG-ATM170029) for computational resources.

APPENDIX

Derivation of 2D anelastic wave dynamics

a. Continuous equations

The linearized hydrostatic anelastic equations in the horizontal direction x and height z are given by

$$q_t + \bar{q}_z w = Q_2',$$
$$s_t + \bar{s}_z w = Q_1',$$
$$u_t + \phi_x = -du.$$

The prognostic variables are humidity q, dry static energy $s = T + \frac{g}{c_p} z$, horizontal velocity u, and vertical velocity w. These are assumed to be perturbations from a large scale state denoted by $\bar{\cdot}$. The an-elastic geopotential term is given by $\phi = p'/\rho_0$, where $\rho_0(z)$ is a reference density profile specified for the full non-linear model.

These prognostic equations are completed by assuming hydrostatic balance and mass-conservation. Hydrostatic balance is given by

$$\phi_z = B$$

where the $B = gT/\bar{T}$ is the buouyancy. Mass conservation is defined by

$$u_x + \frac{1}{\rho_0}\partial_z \rho_0 w.$$

We now combine these diagnostic relations and zonal momentum equation into a single prognostic equation for w. For convenience, we define two differentiable operators, $L = \partial_z$ and $\mathcal{H} = \frac{1}{\rho_0}\partial_z \rho_0$. Taking the x derivative of the momentum equation, and applying the divergence-free condition gives

$$\mathcal{H} w_t + d\mathcal{H} w - \phi_{xx} = 0.$$

Then, applying L gives

$$L\mathcal{H}(\partial_t + d)w = B_{xx}.$$

We let $A = L\mathcal{H}$, and manipulate the equations to obtain

$$w_t = -\partial_{xx} A^{-1} B - dw.$$

Because A is an elliptic operator in the vertical direction, it requires two boundary conditions. In this case, we assume these are given by a rigid lid and impenetrable surface (e.g. $w(0) = w(H_T) = 0$) where H_T is the depth of the atmosphere.

b. Vertical Discretization

Solving (6) numerically requires discretizing the elliptic operator A. To do this, we assume that w, s, and q are vertically collocated. Then, in the interior of the domain, the operator A can be discretized as the following tri-diagonal matrix:

$$(Aw)_k = a_k w_{k-1} + b_k w_k + c_k w_{k+1}$$

where

$$\begin{aligned}
a_k &= \frac{\rho_{k-1}}{(z_k - z_{k-1})(z_{k+1/2} - z_{k-1/2})\rho_{k-1/2}} \\
b_k &= -\frac{\rho_k}{(z_{k+1/2} - z_{k-1/2})} \times \\
&\quad \left[\frac{1}{(z_{k+1} - z_k)\rho_{k+1/2}} + \frac{1}{(z_k - z_{k-1})\rho_{k-1/2}}\right], \\
c_k &= \frac{\rho_{k+1}}{(z_{k+1} - z_k)(z_{k+1/2} - z_{k-1/2})\rho_{k+1/2}}.
\end{aligned}$$

The index k ranges from 1 to N, the number of vertical grid cells, and z is the height.

The rigid-lid boundary conditions are satisfied by: $w_0 = -w_1$ and $w_{n+1} = -w_n$. It is not simply w_0 because the vertical velocity should be located at the cell center. These boundary conditions can be implemented by modifying the matrix representation of A to satisfy

$$\begin{aligned}
(Aw)_1 &= -a_1 w_1 + b_1 w_1 + c_1 w_2, \\
(Aw)_n &= a_n w_{n-1} + b_n w_n - c_n w_n
\end{aligned}$$

at the lower and upper boundaries.

References

Abadi, M., and Coauthors, 2015: TensorFlow: Large-scale machine learning on heterogeneous systems. URL http://tensorflow.org/, software available from tensorflow.org.

Adames, Á. F., and D. Kim, 2015: The MJO as a dispersive, convectively coupled moisture wave: Theory and observations. *J. Atmos. Sci.*, **73 (3)**, 913–941, doi:10.1175/JAS-D-15-0170.1.

Ahmed, F., and J. D. Neelin, 2018: Reverse engineering the tropical Precipitation–Buoyancy relationship. *J. Atmos. Sci.*, **75 (5)**, 1587–1608, doi:10.1175/JAS-D-17-0333.1.

Aires, F., M. Schmitt, A. Chedin, and N. Scott, 1999: The" weight smoothing" regularization of mlp for jacobian stabilization. *IEEE Transactions on Neural Networks*, **10 (6)**, 1502–1510.

Beucler, T., T. Cronin, and K. Emanuel, 2018: A linear response framework for Radiative-Convective instability. *J. Adv. Model. Earth Syst.*, **10 (8)**, 1924–1951, doi:10.1029/2018MS001280.

Beucler, T. G., 2019: Interaction between water vapor, radiation and convection in the tropics. Ph.D. thesis, Massachusetts Institute of Technology.

Brenowitz, N., 2019: Coarse-grained Near-global Aqua-planet Simulation with Computed Dynamical Tendencies. Zenodo, URL https://doi.org/10.5281/zenodo.2621638, doi:10.5281/zenodo.2621638.

Brenowitz, N. D., and C. S. Bretherton, 2018: Prognostic validation of a neural network unified physics parameterization. *Geophys. Res. Lett.*, **17**, 2493, doi:10.1029/2018GL078510.

Brenowitz, N. D., and C. S. Bretherton, 2019: Spatially extended tests of a neural network parametrization trained by coarse-graining. *J. Adv. Model. Earth Syst.*, doi:10.1029/2019MS001711.

Bretherton, C. S., P. N. Blossey, and M. Khairoutdinov, 2005: An Energy-Balance analysis of deep convective Self-Aggregation above uniform SST. *J. Atmos. Sci.*, **62 (12)**, 4273–4292, doi:10.1175/JAS3614.1.

Bretherton, C. S., M. E. Peters, and L. E. Back, 2004: Relationships between water vapor path and precipitation over the tropical oceans. *J. Clim.*, **17 (7)**, 1517–1528, doi:10.1175/1520-0442(2004)017⟨1517:RBWVPA⟩2.0.CO;2.

Chevallier, F., F. Chéruy, N. A. Scott, and A. Chédin, 1998: A neural network approach for a fast and accurate computation of a longwave radiative budget. *J. Appl. Meteorol.*, **37 (11)**, 1385–1397, doi:10.1175/1520-0450(1998)037⟨1385:ANNAFA⟩2.0.CO;2.

Chevallier, F., and J.-F. Mahfouf, 2001: Evaluation of the jacobians of infrared radiation models for variational data assimilation. *Journal of Applied Meteorology*, **40 (8)**, 1445–1461.

Emanuel, K. A., 1994: *Atmospheric convection*. Oxford University Press on Demand.

Gentine, P., M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis, 2018: Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.*, doi:10.1029/2018GL078202.

Hayashi, Y., 1971: Instability of Large-Scale equatorial waves with a Frequency-Dependent CISK parameter. *Journal of the Meteorological Society of Japan. Ser. II*, **49 (1)**, 59–62.

Herman, M. J., and Z. Kuang, 2013: Linear response functions of two convective parameterization schemes. *Journal of Advances in Modeling Earth Systems*, **5 (3)**, 510–541.

Intergovernmental Panel on Climate Change, 2014: *Climate Change 2013 - The Physical Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, doi:10.1017/CBO9781107415324.

Khairoutdinov, M., D. Randall, and C. DeMott, 2005: Simulations of the atmospheric general circulation using a Cloud-Resolving model as a superparameterization of physical processes. *J. Atmos. Sci.*, **62 (7)**, 2136–2154, doi:10.1175/JAS3453.1.

Khairoutdinov, M. F., and D. A. Randall, 2001: A cloud resolving model as a cloud parameterization in the NCAR community climate system model: Preliminary results. *Geophys. Res. Lett.*, **28 (18)**, 3617–3620, doi:10.1029/2001GL013552.

Khairoutdinov, M. F., and D. A. Randall, 2003: Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities. *J. Atmos. Sci.*, **60 (4)**, 607–625, doi:10.1175/1520-0469(2003)060⟨0607:CRMOTA⟩2.0.CO;2.

Khouider, B., and A. J. Majda, 2006: A simple multicloud parameterization for convectively coupled tropical waves. part i: Linear analysis. *J. Atmos. Sci.*, **63 (4)**.

Krasnopolsky, V. M., 2007: Reducing uncertainties in neural network jacobians and improving accuracy of neural network emulations with nn ensemble approaches. *Neural Networks*, **20 (4)**, 454–461.

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and A. A. Belochitski, 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**, e485 913, doi:10.1155/2013/485913.

Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and D. V. Chalikov, 2005: New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Mon. Weather Rev.*, **133 (5)**, 1370–1383, doi:10.1175/MWR2923.1.

Kuang, Z., 2008: A Moisture-Stratiform instability for convectively coupled waves. *J. Atmos. Sci.*, **65 (3)**, 834–854, doi:10.1175/2007JAS2444.1.

Kuang, Z., 2010: Linear response functions of a cumulus ensemble to temperature and moisture perturbations and implications for the dynamics of convectively coupled waves. *J. Atmos. Sci.*, **67 (4)**, 941–962, doi:10.1175/2009JAS3260.1.

Kuang, Z., 2018: Linear stability of moist convecting atmospheres part i: from linear response functions to a simple model and applications to convectively coupled waves. *J. Atmos. Sci.*, doi:10.1175/JAS-D-18-0092.1.

Majda, A. J., and M. G. Shefter, 2001: Waves and instabilities for model tropical convective parameterizations. *J. Atmos. Sci.*, **58 (8)**, 896–914.

McGovern, A., R. Lagerquist, D. John Gagne, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith, 2019: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Am. Meteorol. Soc.*, **100 (11)**, 2175–2199, doi:10.1175/BAMS-D-18-0195.1.

Molnar, C., G. Casalicchio, and B. Bischl, 2018: iml: An r package for interpretable machine learning. *Journal of Open Source Software*, **3 (26)**, 786.

Montavon, G., W. Samek, and K.-R. Müller, 2018: Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, **73**, 1–15, doi:10.1016/j.dsp.2017.10.011.

Neelin, J. D., O. Peters, and K. Hales, 2009: The transition to strong convection. *J. Atmos. Sci.*, **66 (8)**, 2367–2384, doi:10.1175/2009JAS2962.1.

O'Gorman, P. A., and J. G. Dwyer, 2018: Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *J. Adv. Model. Earth Syst.*, **10 (10)**, 2548–2563, doi:10.1029/2018MS001351.

Oueslati, B., and G. Bellon, 2015: The double itcz bias in cmip5 models: interaction between sst, large-scale circulation and precipitation. *Climate dynamics*, **44 (3-4)**, 585–607.

Palmer, T. N., 2001: A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models. *Quart. J. Roy. Meteor. Soc.*, **127 (572)**, 279–304, doi:10.1002/qj.49712757202.

Paszke, A., and Coauthors, 2019: Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 8024–8035, URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pritchard, M. S., C. S. Bretherton, and C. A. DeMott, 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.

Rasp, S., 2019: Online learning as a way to tackle instabilities and biases in neural network parameterizations. *arXiv preprint arXiv:1907.01351*.

Rasp, S., M. S. Pritchard, and P. Gentine, 2018: Deep learning to represent subgrid processes in climate models. *Proc. Natl. Acad. Sci. U. S. A.*, **115 (39)**, 9684–9689, doi:10.1073/pnas.1810286115.

Rushley, S. S., D. Kim, C. S. Bretherton, and M.-S. Ahn, 2018: Reexamining the nonlinear Moisture-Precipitation relationship over the tropical oceans. *Geophys. Res. Lett.*, **45 (2)**, 2017GL076 296, doi:10.1002/2017GL076296.

Samek, W., T. Wiegand, and K.-R. Müller, 2017: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.

Schneider, T., J. Teixeira, C. S. Bretherton, F. Brient, K. G. Pressel, C. Schär, and A. P. Siebesma, 2017: Climate goals and computing the future of clouds. *Nat. Clim. Chang.*, **7 (1)**, 3–5, doi:10.1038/nclimate3190.

Sobel, A., and E. Maloney, 2013: Moisture modes and the eastward propagation of the MJO. *J. Atmos. Sci.*, **70 (1)**, 187–192, doi:10.1175/JAS-D-12-0189.1.

Toms, B. A., E. A. Barnes, and I. Ebert-Uphoff, 2019: Physically interpretable neural networks for the geosciences: Applications to earth system variability. *arXiv preprint arXiv:1912.01752*.

Wood, R., and C. S. Bretherton, 2006: On the relationship between stratiform low cloud cover and Lower-Tropospheric stability. *J. Clim.*, **19 (24)**, 6425–6432, doi:10.1175/JCLI3988.1.