MACHINE LEARNING APPLICATIONS FOR WEATHER AND CLIMATE MODELING

A Dissertation

by

TROY JOSEPH ARCOMANO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Istvan Szunyogh
Committee Members,	Ramalingam Saravanan
	Craig Epifanio
	Ping Chang
Head of Department,	Ramalingam Saravanan

December 2022

Major Subject: Atmospheric Sciences

ABSTRACT

This study investigates the applications of machine learning (ML) to weather and climate modeling. We first show the potential for data-driven weather prediction by creating a low resolution, ML-based global atmospheric model that predicts the 3-dimensional atmosphere in the same format as a physic-based numerical model. The ML-only atmospheric model is stable during 21-day forecasts and can reproduce large-scale atmospheric dynamics (e.g. Rossby waves). The ML-only model is able to outperform persistence and climatology for the first three forecast days in the midlatitudes. When compared to a simplified atmospheric general circulation model (AGCM), the ML-only model performs best for variables most heavily influenced by parameterizations in the AGCM (e.g. low level specific humidity).

Next, we combine a parallel, machine learning algorithm with a coarse resolution AGCM (SPEEDY) to create a hybrid atmospheric model. The hybrid model produces more accurate forecasts for all variables for at least the first 7 forecast days when compared to the host AGCM. Applications of the hybrid model for climate research are explored with a 11-year free run. The hybrid model is free of instability and can simulate the past climate with substantially smaller systematic errors and more realistic variability than the host AGCM.

Lastly, we show potential of ML for Earth System modeling by dynamically coupling a hybrid atmospheric model and a ML-based ocean model trained to predict the sea surface temperature (SST). The ML-only ocean model is able to reproduce SST dynamics with minimal biases for the past and present climate. The coupled model can simulate long-term variability in both the atmosphere and ocean (e.g. El Niño–Southern Oscillation). During a 70-year free run, we find that the coupled model does not exhibit climate drift and able to conserve total atmospheric mass and water vapor mass.

DEDICATION

I dedicate this dissertation to my loving wife and parents. Without your support I would not be where I am today.

ACKNOWLEDGEMENTS

I would like to first thank my committee chair and advisor, Dr. Istvan Szunyogh, for all of his support, guidance, and help during my graduate studies. He challenged me to be a better scientist and without him I would not be where I am today.

I would also like to thank my committee members: Dr. Ping Chang, Dr. Craig Epifanio, and Dr. Ramalingam Saravanan, for providing valuable and insightful suggestions on my research and the dissertation.

Lastly, I want to thank my friends and family. Rachel, my wife, I am forever grateful for all of your support. Whether it was taking care of the dogs when I had to work late nights at the office or helping me proofread papers, you were always there for me thank you. I thank my parents for always believing me and encouraging me to pursue a career in atmospheric science. I would also like to thank my friends I've met at the Texas A&M; Dr. Kyle Wodzicki, Dr. Kevin Smalley, Judy Dickey, and many more.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Istvan Szunyogh, Ramalingam Saravanan, and Craig Epifanio of the Department of Atmospheric Sciences and Professor Ping Chang of the Department of Oceanography.

The studies shown in Section 2, 3, and 4 were conducted in collaboration with professors and graduate students from the University of Maryland (UMD): Professors Edward Ott and Brian Hunt, Graduate Students: Alexander Wikner and Jaideep Pathak.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by the following grants:

- DARPA contract DARPA-PA-18-01 (HR111890044)
- Office of Naval Research Grant N00014-18-2509
- Office of Naval Research Grant N00014-22-1-2319
- Defense Advanced Research Projects Agency contract HR00112290035

NOMENCLATURE

AGCM	atmospheric general circulation model
CESM	Community Earth System Model
CHyPP	combined hybrid-parallel prediction
CMIP6	Coupled Model Intercomparison Project version 6
CPU	central processing unit
E3SM	Energy Exascale Earth System Model
ECMWF	European Centre for Medium-Range Weather Forecasts
ENSO	El Niño Southern Oscillation
ESM	earth system model
GCM	global climate model
GPU	graphics processing unit
HadISST	Hadley Centre Sea Ice and Sea Surface Temperature
IPCC	Intergovernmental Panel on Climate Change
ITCZ	Intertropical Convergence Zone
KS	Kuramoto-Sivashinsky
LETKF	Local Ensemble Transform Kalman Filter
LLR	local linear regression
ML	machine learning
NH	Northern Hemisphere
NN	neural network
NWP	numerical weather prediction

ONI	oceanic nino index
QBO	quasi-biennial oscillation
RC	reservoir computing
RMSE	root-mean-square error
SH	Southern Hemisphere
SOI	southern oscillation index
SPEEDY	Simplified Parameterizations privitivE-Equation DYnamics
SST	sea surface temperature
SSW	sudden stratospheric warming

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

For the last several decades, numerical weather prediction (NWP) has been backbone of operational weather prediction (e.g., Lynch 2006; Harper 2008). NWP models rely on numerically solving the discretized physic-based governing equations to evolve a finite-resolution prediction of the atmospheric state in space and time. Many important atmospheric processes occur at scales (e.g. cloud microphysics) too small to be resolved directly by the dynamical core of the model. The cumulative effects of these processes have to be parameterized. The *parameterization schemes* are imperfect and typically make theoretical or empirical considerations (e.g., Stensrud 2007; Haupt et al. 2008). They also are computationally expensive and take up a significant fraction of the computations in a modern NWP model. The initial conditions of the numerical model solutions are observation-based estimates (analyses) of the state of the atmosphere, and the computational process that produces these estimates is called *data assimilation* (e.g., Szunyogh 2014). The advances in modeling and data assimilation techniques, alongside with the increase of computing power and the number of observations available for assimilation, led to a *"quiet revolution of NWP"* (Bauer et al. 2015).

During this same period, there was a significant amount of research into improved numerical modeling of the general circulation of the atmosphere. *Atmospheric general circulation models* (AGCMs) quickly became an important tool to study climate change and anticipate likely changes to the climate in the future (e.g., Lynch 2008). AGCMs are now just one component of the state-of-the-art climate models and are fully coupled to other Earth system components such as the ocean, sea-ice, and the land surface (e.g., Golaz et al. 2019; Danabasoglu et al. 2020). These climate models provide crucial information for the Intergovernmental Panel on Climate Change (IPCC) to make projections and recommendations in their Assessment Reports (Arias et al. 2021). However, state-of-the-art climate models still have large systematic biases when compared to observations of the present climate (Flato et al. 2014). Reducing these biases remains a great challenge for researchers and leads to uncertainty in climates projections. Zhu et al. (2020) found that the latest

version of the Coupled Model Intercomparison Project version 6 (CMIP6) has a cold bias in the North Pacific ocean and Li et al. (2019) found a relationship between an equatorial cold bias in the ocean and the underestimation of precipitation in the Northwest Pacific for the climate models in CIMP5. Bias correction methods and post processing (e.g., Ivanov et al. 2018; Vaittinada Ayar et al. 2021) are tools aimed to reduce these biases in climate model output before being used, however, even the practice of bias correction remains controversial (Chen et al. 2021).

Machine learning (ML) is a powerful tool with the ability to perform a wide variety of tasks including natural language processing, image classification, computer vision, and time-series prediction (e.g., LeCun et al. 2015; Sarker 2021). Recently, there has been strong interest to leverage the power of ML for Earth sciences, with data-driven approaches having been successfully applied to a number of problems ranging from satellite estimation of tropical cyclone intensity to severe weather prediction (Pradhan et al. 2018; Lagerquist et al. 2020; Flora et al. 2021). A particularly promising application of machine learning is improving weather forecasts and climate simulations. Machine learning, specifically neural networks (NNs), have been applied to climate model downscaling and postprocessing (e.g., Rasp and Lerch 2018), nonlinear weighting of ensembles (e.g., Campos et al. 2019), and quantifying forecast uncertainty (e.g., Scher and Messori 2018). However, these applications of ML are done outside of the numerical model. The incorporation of ML techniques into the numerical model is a potential avenue to further forecast accuracy gains and bias reduction in climate simulations by extracting additional information from observations.

The earliest applications of ML to atmospheric modeling focused on improving the computational efficiency of the physics-based numerical models (e.g., Krasnopolsky and Chevallier 2003; Krasnopolsky et al. 2005; Krasnopolsky and Fox-Rabinovitz 2006; Krasnopolsky 2013). These applications employed neural networks to emulate the computationally most expensive physics-based parameterization schemes at a reduced computational cost. Many of these first emulators focused on radiative transfer schemes (Chevallier et al. 2000; Krasnopolsky and Chevallier 2003). The neural network based emulator of Chevallier et al. (2000) called ”NeuralFlux” was so successful that it was quickly implemented into operational data assimilation (Haupt et al. 2008). The term *hybrid*

model was first used in reference to models using this technique. One approach employed by this type of hybrid models was to use a single neural network to emulate the combined effect of multiple parameterized processes, such as cumulus convection, radiation, boundary layer transport, etc. (e.g., Krasnopolsky et al. 2010; Krasnopolsky 2013; Brenowitz and Bretherton 2018, 2019; Rasp et al. 2018). In these applications, the ML model components were often trained on data produced by model simulations at higher resolutions, or with more sophisticated physical parameterization schemes.

Another type of ML-based parameterization scheme (e.g., Gentine et al. 2018; Rasp et al. 2018; Chattopadhyay et al. 2020), is trained on observations or observations-based reanalyses. Such a scheme has the potential to learn about the effects of processes that the higher resolution and more sophisticated model simulations are still unable to capture. ML techniques have also been considered for the estimation of the free parameters of physics-based parameterization schemes (Schneider et al. 2017). This approach takes advantage of the knowledge built into the parameterization schemes, but may suffer from the assumptions and approximations made by the schemes.

Numerical stability and climate drift is one of the biggest concerns with the hybrid models, because typically the training of the ML-based parameterization schemes are done in a manner that does not guarantee stability (trained "offline" to only to predict one-time-step into the future) when the hybrid model is run for a long periods of time (Rasp 2020). To reduce instability and climate drift, Brenowitz and Bretherton (2018) used a multi-time-step cost function and Yuval and O'Gorman (2020) used a ML architecture that conserved energy. However, both of those studies used an idealized atmospheric model on an aqua-planet and integrating the methods they proposed into complex GCMs is still a challenge. Another interesting approach that may help tackle climate drift and allow for hybrid models to be used for nonstationary climate simulations, is the scaling of input variables in such a way so they are climate invariant (Beucler et al. 2021).

There has also been research into purely data-driven methods for weather prediction. One of the first studies to investigate this problem was Dueben and Bauer (2018), who trained NNs using reanalyses to predict a single variable (500-hPa geopotential height) at very coarse resolution, but

with limited success. They, however, correctly outlined some of the difficulties that would be faced in making a more complex data-driven weather model, including model/algorithm design, length and quality of training data, and whether to use global or local ML operations. Subsequent studies (e.g., Weyn et al. 2019, 2020; Rasp and Thuerey 2021) quickly increased the complexity and resolution of the ML-only models to predict a select number of 2-dimensional atmospheric state variables. Scher and Messori (2018, 2019) trained their models to predict the three-dimensional multivariate atmosphere, but they did the training on low resolution GCM simulation data instead of observation-based reanalysis. More recently, Pathak et al. (2022) used state-of-art ML techniques with a vast amount of reanalyses training data and computational resources to train a data-driven model to predict the three-dimensional multivariate atmosphere at $0.25^\circ \times 0.25^\circ$ horizontal resolution. Their preliminary results suggest that data-driven models are approaching the resolution and forecast skill of operational NWP models in the day 3 forecast range.

One important feature of a purely data-driven model is that once it has been trained, the computational efficiency of the model can be orders of magnitude faster than that of a traditional numerical model. Taking advantage of this feature, Weyn et al. (2021) developed a ML-only based ensemble system with 360 members for sub-seasonal to seasonal weather prediction. Using computer hardware developed specifically for deep learning (GPU), their 360 member ensemble system was able to make a 6-week forecast in just three minutes. Scher and Messori (2021) explored different methods for creating individual ensemble members and optimal methods to perturb the initial conditions for a neural network based ensemble system.

The hybrid modeling approach in this study belongs to a class of techniques that are different from those mentioned thus far. Techniques of this class use ML for the *frequent periodic interactive correction of the spatiotemporally evolving physics-based numerical model solution* after training on observational analyses. The specific approach used in this study was originally developed by Pathak et al. (2018a) and later adapted to large dynamical systems by Wikner et al. (2020), who named it *Combined Hybrid-Parallel Prediction* (CHyPP). It evolves the hybrid forecasts iteratively, combining a short-term (e.g., 6 h) numerical forecast with a state-dependent ML correction in

each “time step” of the “hybrid model integration”. CHyPP is not a postprocessing technique, because each “time step” of the evolving hybrid model solution starts from the ML-corrected state of the preceding step, whereas a postprocessing technique does not interact with the evolving model solution. The ML component of CHyPP uses the computationally highly efficient parallel *reservoir computing (RC)* algorithm of Pathak et al. (2018b). Wikner et al. (2020) demonstrated the potential of CHyPP for predicting the evolution of a spatiotemporally chaotic system by experiments with the Kuramoto-Sivashinsky (KS) model (Sivashinsky 1977), a model that has a single state variable that depends only on a single space dimension in addition to time.

Farchi et al. (2021) applied a similar hybrid modeling approach using a deep learning neural network and a two-layer quasi-geostrophic channel model. In their model, a short-term prediction from their imperfect numerical model was combined with a the machine learning model to produce a hybrid model prediction. Their hybrid model performed better than either just the imperfect numerical model or the machine learning only model. Watt-Meyer et al. (2021) also used similar hybrid modeling approach to the CHyPP, but instead of RC, they used a random forest-based ML architecture. Their numerical model was FV3GFS, a finite-volume model on a cubed sphere. The random forests were trained to nudge certain model variables based on 6-hour forecast errors of the numerical model when compared to observational-based reanalysis. They found their hybrid model to improve the forecast skill and to stay free of instability in a long (one year) simulation run. Bretherton et al. (2022) furthered this work by correcting a coarse resolution numerical model using a deep learning NN trained on a cloud resolving (3-km resolution) model to learn the nudging for certain model variables.

Machine learning has been demonstrated to be a promising approach to improve weather forecasting and reducing biases in climate models. The parallel machine learning algorithm of Pathak et al. (2018a) and the hybrid modeling approach of Pathak et al. (2018b) and Wikner et al. (2020) were originally tested on relatively small spatio-temporal chaotic systems. The atmosphere has multiple state variables with a wide range of values that depend on all three spatial dimensions. Even at coarse horizontal and vertical resolution, the number of variables needed to be predicted

is $O(10^5)$. Implementing these approaches to the atmosphere also raises a number of challenges related to algorithm design and scaling. The training of such a system requires significant computational resources. The main goal of the present study is to demonstrate the improvement in forecast skill and reduction of biases for climate simulations by implementing CHyPP on the *Simplified Parameterization, primitive-Equation Dynamics (SPEEDY)* (Molteni 2003; Kucharski et al. 2006), a reduced resolution AGCM.

In what follows, Chapter 2 describes performance of an ML-only global atmospheric model based on the parallel RC algorithm of Pathak et al. (2018a). In Chapter 3, we describe the hybrid approach of CHyPP and its implementation on SPEEDY in detail as well as discuss the results of the forecast and climate simulation experiments. In Chapter 4, we describe an atmospheric hybrid model that is coupled with a machine learning only based ocean model. We evaluate the performance of this coupled model for both atmospheric and oceanic variability (e.g. annual cycle and the El Niño-Southern Oscillation). Finally, in Chapter 5 we summarize our key findings and draw our conclusions.

2. A MACHINE LEARNING-BASED GLOBAL ATMOSPHERIC FORECAST MODEL*

2.1 Introduction

The ultimate goal of our research is to develop a *hybrid* (numerical-machine-learning) *weather prediction (HWP)* model. We hope to achieve this goal by implementing algorithms developed by Pathak et al. (2018a,b) and Wikner et al. (2020): the first paper introduced an efficient ML algorithm for numerical-model-free prediction of large, spatiotemporal dynamical systems, based solely on the knowledge of past states of the system; the second paper showed how to combine a machine learning (ML) algorithm with an imperfect numerical model of a dynamical system to obtain a hybrid model that predicts the system more accurately than either component alone; while the third paper combined the techniques of the first two into a computationally efficient hybrid modeling approach. The present paper implements the parallel ML technique of Pathak et al. (2018a) to build a model that predicts the weather in the same format as a global numerical model. We train and verify the model on hourly ERA5 reanalysis data from the European Centre for Medium-Range Weather Forecasts (ECMWF) (Hans et al. 2019).

The work presented here can also be considered an attempt to develop a ML model that can predict the evolution of the three-dimensional, multivariate, global atmospheric state. To the best of our knowledge, the only similar prior attempts were those by Scher and Messori (2018) and Scher and Messori (2019), but they trained their three-dimensional multivariate ML model on data that was produced by low-resolution numerical model simulations. In addition, Dueben and Bauer (2018) and Weyn et al. (2019, 2020) designed ML models to predict two-dimensional, horizontal fields of select atmospheric state variables. Similar to our verification strategy, they also verified the ML forecasts against reanalysis data. Compared to all of the aforementioned studies, an important new aspect of our work is that we employ *reservoir computing (RC)* (Jaeger 2001; Maass et al. 2002; Lukoševičius and Jaeger 2009; Lukoševičius 2012) rather than *deep-learning* (e.g. Goodfellow

*Reprinted with permission from "A Machine Learning-Based Global Atmospheric Forecast Model" by Troy J. Arcomano,I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott, 2020. Geophysical Research Letters, 47, © Copyright (07 May 2020) American Geophysical Union.

et al. 2016), which is primarily motivated by the significantly lower computer wall-clock time required to train an RC-based model. This difference in training efficiency would allow for a larger number of experiments to tune the ML model at higher resolutions.

The structure of the paper is as follows. Section 2.2 describes the ML model, while section 2.3 presents the results of the forecast experiments, using as benchmarks persistence of the atmospheric state, climatology, as well as numerical forecasts from a physics-based model of identical prognostic state variables and resolution. Section 2.4 summarizes our conclusions.

2.2 The ML model

The N components of the state vector $\mathbf{v}^m(t)$ of the ML model are the grid-point values associated with the spatially discretized fields of the Eulerian dependent variables of the model. Training the model requires the availability of a discrete time series of past *observation-based estimates* (analyses) $\mathbf{v}^a(k\Delta t)$ ($k = -K, -K+1, \ldots, 0$) of the atmospheric states that use the same N-dimensional representation of the state as the model. Beyond the training period, the analyses $\mathbf{v}^a(k\Delta t)$ ($k = 1, 2, \ldots$) are used only to maintain the synchronization of the model state with the observed atmospheric state. An ML forecast can potentially be started at any analysis time $k\Delta t$ ($k = 0, 1, \ldots$): the forecast is a discrete time series of model states $\mathbf{v}^m_k(k'\Delta t)$ ($k' = k+1, k+2, \ldots$), where $k\Delta t$ is the *initial time*, $\mathbf{v}^a(k\Delta t)$ is the *initial state*, Δt is the *time-step*, and $(k'-k)\Delta t$ is the *forecast time*. The computational algorithm of the model is designed to take advantage of a massively parallel computer architecture.

2.2.1 Representation of the Model State

2.2.1.1 The Global State Vector

We define $\mathbf{v}^m(t)$ by the grid-based state vector of the physics-based numerical model SPEEDY (Molteni 2003; Kucharski et al. 2013). While SPEEDY is a spectral transform model, it uses the grid-based state vector to represent the input and output state of the model, and to compute the nonlinear and parameterized terms of the physics-based prognostic equations. The horizontal grid spacing is $3.75^\circ \times 3.75^\circ$ and the model has $n_v = 8$ vertical σ-levels (at σ equals 0.025, 0.095, 0.20,

0.34, 0.51, 0.685, 0.835, and 0.95), where σ is the ratio of pressure to the pressure at the surface. The model has four three-dimensional dependent variables (the two horizontal coordinates of the wind vector, temperature, and specific humidity) and one two-dimensional dependent variable (the logarithm of surface pressure). Thus the number of variables per horizontal location is $n_t = 4 \times n_v + 1$. Because there are $n_h = 96 \times 48 = 36,864$ horizontal grid points, the total number of model variables is $N = n_t \times n_h$=1.52064$\times 10^5$. Before forming the state vector $\mathbf{v}^m(t)$, we standardize each state variable by subtracting its climatological mean and dividing by its standard deviation at the particular model level in the local region.

2.2.1.2 Local State Vectors

The global model domain is partitioned into $L = 1,152$ local regions. We use a Mercator (cylindrical) map projection to define the local regions, partitioning the three-dimensional model domain only in the two horizontal directions: each local region has the shape of a rectangular prism with a 7.5°$\times$7.5° base (Fig. 2.1). The model state in local region ℓ ($\ell = 1, 2, \ldots, L$) is represented by the *local state vector* $\mathbf{v}_\ell^m(t)$, whose components are defined by the D_v=4$\times n_t = 132$ components of the global state vector in the local region. The model computes the L evolved local state vectors $\mathbf{v}_\ell^m(t + \Delta t)$ from $\mathbf{v}^m(t)$ in parallel, and the evolved global state vector $\mathbf{v}^m(t + \Delta t)$ is obtained by piecing the L evolved local state vectors together.

2.2.2 The Computational Algorithm

2.2.2.1 RC

The computation of $\mathbf{v}_\ell^m(t+\Delta t)$ from $\mathbf{v}^m(t)$ requires the evaluation of a composite (chain) function for each local state vector. Because we use an RC algorithm, this composite function has only three layers: the *input layer, the reservoir,* and the *output layer.* A key feature of RC is that the trainable parameters of the model appear only in the output layer, which greatly simplifies the training process.

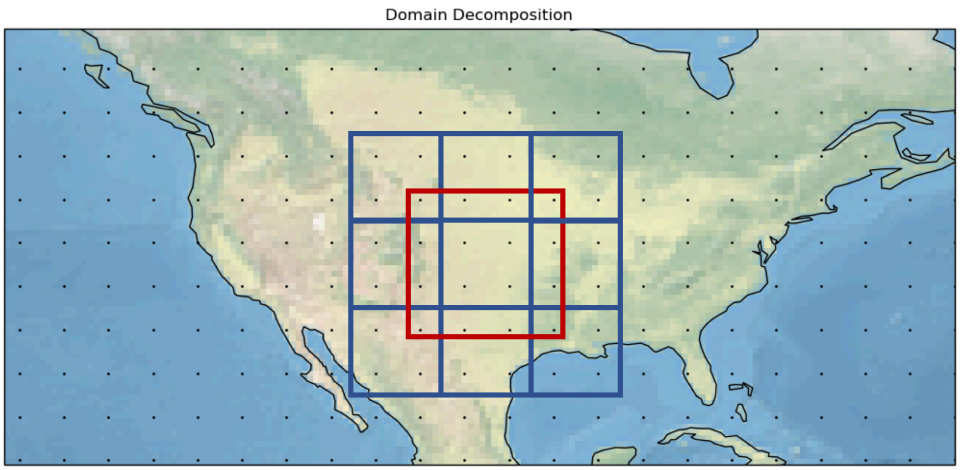

Figure 2.1: Illustration of the local regions. The local regions are defined on a Mercator map projection, where the black dots indicate the horizontal location of the grid-points of the model. The blue rectangles mark the boundaries of nine adjacent local regions. The red rectangle indicates the boundaries of the extended local region for the local region in the center. Reprinted with permission from Arcomano et al. (2020).

2.2.2.2 The Input Layer and Reservoir

The composite of the input layer and the reservoir is

$$\mathbf{r}_\ell(t+\Delta t) = \mathbf{G}_\ell\{\mathbf{r}_\ell(t), \mathbf{W}_{in,\ell}[\hat{\mathbf{v}}_\ell^m(t)]\}, \tag{2.1}$$

where the function $\mathbf{W}_{in,\ell}[\cdot]$ is the input layer. The dimension D_r of the *reservoir state vector* $\mathbf{r}_\ell(t) = (r_{\ell,1}, r_{\ell,2}, \ldots, r_{\ell,D_r})$ is much higher than the dimension $D_{\hat{v}}$ of the input vector $\hat{\mathbf{v}}_\ell^m(t)$. (The reservoir is a high-dimensional dynamical system.) The input vector $\hat{\mathbf{v}}_\ell^m(t)$ is an *extended local state vector* that represents the model state in an extended local region. In the present paper, we define $\hat{\mathbf{v}}_\ell^m(t)$ by the grid points of local region ℓ plus the closest grid points from the neighboring local regions (see Fig. 2.1 for an illustration). In the terminology of Pathak et al. (2018a), the *locality parameter* of our model is 1. Using a nonzero value of the locality parameter is essential, because otherwise no information can flow between the local regions. The dimension of the extended local state vectors is $D_{\hat{v}}$=16$\times n_t$ = 528 for most ℓ. The exceptions are the local regions nearest to the two poles, because for those, we add no extra grid points in the poleward direction. The dimension of the

input vectors in these local regions is $D_{\hat{v}} = 12 \times n_t = 396$.

The 'local approach' of Dueben and Bauer (2018), which was introduced independently of the parallel technique of Pathak et al. (2018a), employs a localization strategy that is formally similar to the one described here. There is, however, an important difference between the two localization techniques: Dueben and Bauer (2018) trained a single common neural network for the different local regions, while we train a different reservoir for each local region.

The input layer is implemented as $\mathbf{W}_{in,\ell}[\hat{\mathbf{v}}_\ell^m(t)] = \mathbf{W}_{\hat{v},\ell}\hat{\mathbf{v}}_\ell^m(t)$, where $\mathbf{W}_{\hat{v},\ell}$ is a sparse $D_r \times D_{\hat{v}}$ random matrix, whose entries are drawn from a uniform probability distribution in the interval $[-0.5, 0.5]$. The reservoir dynamics is defined by

$$\mathbf{G}_\ell\{\mathbf{r}_\ell(t), \mathbf{W}_{in,\ell}[\hat{\mathbf{v}}_\ell^m(t)]\} = \tanh\left[\mathbf{A}_\ell \mathbf{r}_\ell(t) + \mathbf{W}_{\hat{v},\ell}\hat{\mathbf{v}}_\ell^m(t)\right], \tag{2.2}$$

where $\tanh[\cdot]$ is the component-wise hyperbolic tangent function and $\mathbf{A}_\ell$ is a $D_r \times D_r$ *weighted adjacency matrix* that represents a low-degree, directed, random graph (Gilbert 1959). Each entry of $\mathbf{A}_\ell$ has a probability κ/D_r of being nonzero, so that the expected degree of each vertex is a prescribed number κ. Thus, κ is the average number of incoming connections (edges) per vertex. The nonzero entries of $\mathbf{A}_\ell$ are randomly drawn from a uniform distribution in the interval $(0, 1]$ and scaled so that the largest eigenvalue of $\mathbf{A}_\ell$ is a prescribed number ρ. The parameter ρ, which controls the length of the memory of the ML model dynamics, is called the *spectral radius*.

2.2.2.3 The Output Layer

The evolved local state vector is obtained by

$$\mathbf{v}_\ell^m(t + \Delta t) = \mathbf{W}_{out,\ell}[\mathbf{r}_\ell(t + \Delta t), \mathbf{P}_\ell], \tag{2.3}$$

where the function $\mathbf{W}_{out,\ell}[\cdot, \cdot]$ is the output layer. This function is chosen such that it is linear in the $D_v \times D_r$ *matrix of trainable parameters* $\mathbf{P}_\ell$. To be precise,

$$\mathbf{W}_{out,\ell}[\mathbf{r}_\ell(t + \Delta t), \mathbf{P}_\ell] = \mathbf{P}_\ell \tilde{\mathbf{r}}_\ell(t + \Delta t), \tag{2.4}$$

where $\tilde{\mathbf{r}}_\ell(t+\Delta t) = (r_{\ell,1}, r^2_{\ell,2}, r_{\ell,3}, r^2_{\ell,4}, \ldots, r_{\ell,D_r-1}, r^2_{\ell,D_r})(t+\Delta t)$.

2.2.2.4 Synchronization and Training

We define the *local analysis* $\mathbf{v}^a_\ell(k\Delta t)$ by the components of the global analysis $\mathbf{v}^a(k\Delta t)$ ($k = -K, -K+1, \ldots$) that describe the state in local region ℓ. In other words, $\mathbf{v}^a_\ell(k\Delta t)$ is the observation-based estimate of the desired value of the model state $\mathbf{v}^m_\ell(k\Delta t)$. Likewise, we define the *extended local analysis* $\hat{\mathbf{v}}^a_\ell(k\Delta t)$ as the observation-based estimate of the extended local state vector $\hat{\mathbf{v}}^m_\ell(k\Delta t)$ ($k = -K, -K+1, \ldots$).

The synchronization and training of the ML model starts with feeding the past analyses to the reservoir, or more precisely, by substituting $\hat{\mathbf{v}}^a_\ell(k\Delta t)$ ($k = -K, -K+1, \ldots, -1$) for $\hat{\mathbf{v}}^m_\ell(k\Delta t)$ in Eq. (2.1). Thus the output layer, Eq. (2.3), is not needed to compute $\mathbf{r}_\ell(k\Delta t)$ for $k = -K+1, -K+2, \ldots, 0$: we generate $\mathbf{r}_\ell(-K\Delta t)$ randomly, discard the transient sequence $\mathbf{r}_\ell(k\Delta t)$, $k = -K, -K+1, \ldots, -K_t$, and define $\mathbf{v}^m_\ell(k\Delta t)$ for $k = -K_t+1, -K_t+2, \ldots, 0$ according to Eq. (2.1), with $\mathbf{P}_\ell$ as yet undetermined.

The goal of the training is to find the $\mathbf{P}_\ell$ that minimizes the cost function

$$J_\ell(\mathbf{P}_\ell) = \left[\sum_{k=-K_t+1}^{0} \|\mathbf{v}^a_\ell(k\Delta t) - \mathbf{v}^m_\ell(k\Delta t)\|^2 \right] + \beta \|\mathbf{W}_{out,\ell}\|), \qquad \ell = 1, 2, \ldots, L, \tag{2.5}$$

where $\|\cdot\|$ is the Frobenius norm. The purpose of the Tikhonov regularization term $\beta\|\mathbf{W}_{out,\ell}\|$) (Tikhonov and Arsenin 1977) of $J_\ell(\mathbf{P}_\ell)$ is to improve the numerical stability of the computations and prevent overfitting to the training data by choosing large values of the components of $\mathbf{W}_{out,\ell}$. Because $\mathbf{W}_{out,\ell}$ depends linearly on $\mathbf{P}_\ell$, the solutions of the L minimization problems can be obtained by a linear *ridge regression*. That is, $\mathbf{P}_\ell$ is computed by solving the linear problem

$$\mathbf{P}_\ell \left(\tilde{\mathbf{R}}_\ell \tilde{\mathbf{R}}^T_\ell + \beta \mathbf{I} \right) = \mathbf{V}^a_\ell \tilde{\mathbf{R}}^T_\ell, \qquad \ell = 1, 2, \ldots, L, \tag{2.6}$$

where the columns of $\tilde{\mathbf{R}}_\ell$ are $\tilde{\mathbf{r}}_\ell(k\Delta t)$ ($k = -K_t+1, -K_t+2, \ldots, 0$) and the columns of $\mathbf{V}^a_\ell$ are $\mathbf{v}^a_\ell(k\Delta t)$ ($k = -K_t+1, -K_t+2, \ldots, 0$). Notice that the dimension of the linear problem of Eq. (2.6)

does not depend on the length K_t of the training period. To conserve memory, the $D_r \times K_t$ matrix $\mathbf{R}_\ell$ need not be stored; the $D_r \times D_r$ matrix $\tilde{\mathbf{R}}_\ell \tilde{\mathbf{R}}_\ell^T$ and the $D_v \times D_r$ matrix $\mathbf{V}_\ell^a \tilde{\mathbf{R}}_\ell^T$ can be built incrementally, passing the training data through the reservoir time-step by time-step (e.g., Lukoševičius and Jaeger 2009; Lukoševičius 2012).

2.2.3 Implementation on ERA5 Reanalysis Data

2.2.3.1 Training

The global analyses $\mathbf{v}^a(k\Delta t)$ ($k = -K, -K+1, \dots$) are hourly ERA5 reanalyses interpolated to the computational grid and adjusted to the topography of SPEEDY. The training starts at 0000 UTC, 1 January, 1981 and ends at 2000 UTC January 24, 2000 ($K \approx 1.66 \times 10^5$). We add a small-magnitude random noise $\varepsilon(t)$ to $\hat{\mathbf{v}}_\ell^a(k\Delta t)$ ($k = -K, -K+1, \dots, -1$) before we substitute it for $\hat{\mathbf{v}}_\ell^m(t)$ in Eq. (2.1) in order to improve the robustness of the ML model to noise (Jaeger 2001). The transient sequence of $K - K_t$ discarded reservoir states corresponds to the first 43 days of training.

2.2.3.2 Code Implementation and Performance

The current computer code of the ML model is written in Fortran, using both MPI and OpenMP for parallelization and the LAPACK routine DGESV to solve the linear problem of Eq. (2.6). The computations of both the training and forecast phase are carried out on 1,152 Intel Xeon E5-2670 v2 processors. Training the model takes 67 minutes wall-clock time and requires 2.2 Gb of distributed memory per processor. Our current code is designed to minimize the wall-clock execution time given the available memory on a particular supercomputer, but the memory usage could be reduced (e.g., by not keeping all training data in memory simultaneously, or using single- rather than double-precision arithmetic).

2.2.4 The Forecast Cycle

Beyond the training period, the analyses are used only to maintain the synchronization between the reservoirs and the atmosphere. We use the hourly reanalyses for synchronization, but start a new 20-day forecast only once every 48 hours. (Preparing a 20-day forecast takes about 1 minute

of wall-clock time.) We prepare a total of 171 forecasts for the period from January 25, 2000 to 28 December, 2000. The forecast error statistics reported below are calculated based on these forecasts.

2.2.4.1 Selection of the Hyperparameters

The dimension D_r of the reservoir, rank κ of the random network, spectral radius ρ, random noise ε, and regularization parameter β are the *hyperparameters* of the RC algorithm. We found suitable combinations of these parameters by numerical experimentation, monitoring the accuracy and stability of the forecasts. All results reported in this paper are for D_r=9,000, κ=6, $\beta = 10^{-5}$, while ρ monotonically increases from 0.3 at the equator to 0.7 at 45° and beyond. The components of ε are uncorrelated, normally distributed, random numbers with mean zero and standard deviation 0.28. For this combination of the hyperparameters, the ML model predicts realistic values of all state variables for the entire globe and 20-day forecast period.

2.3 Forecast Verification Results

2.3.1 Benchmark Forecasts

We use daily climatology, persistence, and numerical forecasts for the evaluation of the ML model forecasts. Persistence is based on the assumption that the initial atmospheric state will persist for the entire time of the forecast. The numerical forecasts are prepared by Version 42 of the SPEEDY model. While SPEEDY has been developed for research applications rather than weather prediction, it can be considered a low-resolution version of today's NWP models. Most importantly, similar to all operational models, it solves the system of atmospheric primitive equations and has a realistic climate. It provides a good benchmark in the current stage of our research, in which the primary goal is to prove a concept rather than improve operational forecasts.

2.3.2 Results

We verify all forecasts against ERA5 reanalyses interpolated to the computational grid and adjusted to the SPEEDY orography. The magnitude of the forecast error is measured by the mean

of the area-weighted root-mean-square difference between the forecasts and the verification data for all forecasts. Results are shown for selected variables in the Northern Hemisphere (NH) mid-latitudes for the first 72 forecast hours (Fig. 2.2). In this region, the ML model outperforms both persistence and climatology by a large margin in the first 48 forecast hours. While the ML model forecasts remain more accurate than persistence in the next 24 forecast hours, their skill, with the exception of the temperature forecasts, degrades to that of climatology. In the tropics (results not shown) the accuracy of the ML model is very similar to that of persistence and climatology

The performance of the ML model compared to SPEEDY is mixed: the ML forecasts are more accurate for the specific humidity near the surface, especially at 24 h and 48 h forecast times, while the SPEEDY forecasts are more accurate for the wind, particularly at the jet level. The ML temperature forecasts are also more accurate in the tropics (results not shown), where the SPEEDY forecasts rapidly develop a large bias in the upper troposphere.

To better understand the behavior of the root-mean-square error, we decomposed it into a (square of) bias and variance component and also investigated the power spectrum of the variance in the NH midlatitudes with respect to the zonal wavenumber (results are not shown). On the positive side, the ML forecasts of the different variables have little or no bias, and the variance of the longer term forecasts saturates at a realistic level for zonal wave numbers larger than 6. On the negative side, the variance saturates at unrealistically high levels at the lower wave numbers, leading to an over-prediction of the spatial variability of the forecast fields at the longer forecast times. The fast growth of the variance at the large scales, especially at wave number 4, is the main deficiency of ML model in the midlatitudes. Fixing this problem could extend the time range of forecast skill by days.

2.3.3 Near-Surface Humidity and Tropical Temperature Profiles

The short-term forecast advantage of the ML model over SPEEDY has two sources. First, while the SPEEDY forecasts rapidly develop a near-surface humidity bias, the ML model forecasts are free of such bias. Second, the variance of the ML model forecast errors is also lower initially. As

forecast time increases, the advantage of the ML model remains in terms of the bias, but vanishes in terms of the variance. Because the variance becomes the dominant component at the later forecast times, climatology breaks even with the ML model forecasts by 72 h forecast time (bottom right panel of Fig. 2.2). The spatial distribution of the difference of the errors (Fig. 2.3) suggests that the ML model performs better in regions where parameterized atmosphere-surface interactions play an important role in the moist processes in SPEEDY (e.g., regions of the ocean boundary currents). Likewise, the advantage of the ML model in predicting the tropical temperature profiles (not shown) is the result of large biases that are present only in the SPEEDY forecasts in the main regions of parameterized deep convection. Finally, it should be noted that while the current version of the ML model learns about atmosphere-surface interactions strictly from the atmospheric training data, SPEEDY uses a number of prescribed fields to describe the surface conditions (e.g., a spatio-temporally evolved sea-surface temperature analysis.)

2.3.4 Rossby Wave Propagation

The forecast variable for which SPEEDY clearly outperforms the ML model is the meridional component of the wind: while the accuracy of the wind forecasts by the two models is similar at 24 h, the error of the ML model forecasts grows more rapidly beyond that time. The difference between the errors of the two models grows the fastest in the layer around the jet streams of the Northern Hemisphere (NH) midlatitudes (between 400 hPa and 200 hPa). Because the variability of the meridional wind in this layer is dominated by dispersive synoptic-scale Rossby waves, the aforementioned result suggests that the ML model may be inferior to the numerical model in describing the Rossby wave dynamics. To investigate this possibility, we plot Hovmöller diagrams of the meridional wind for both forecasts and the verification data (Figure 2.4).

A pattern of negative (positive) values followed by a pattern of positive (negative) values indicate a trough (ridge). Because the eastward group velocity of the dispersive Rossby waves at the synoptic scales is larger than their eastward phase velocity, new troughs and ridges can develop downstream of the original wave. Such developments are marked by oriented dashed black lines

in the figure. In the first three days, the ML model captures the dispersive dynamics of the wave packets accurately, but because the wave packets are composed of wave number 4-11 waves (e.g. Zimin et al. 2003), the over-intensification of the wave number 4-6 components at the later forecast times leads to a gradual shift of the carrier wave number toward lower values and a deceleration of the group velocity.

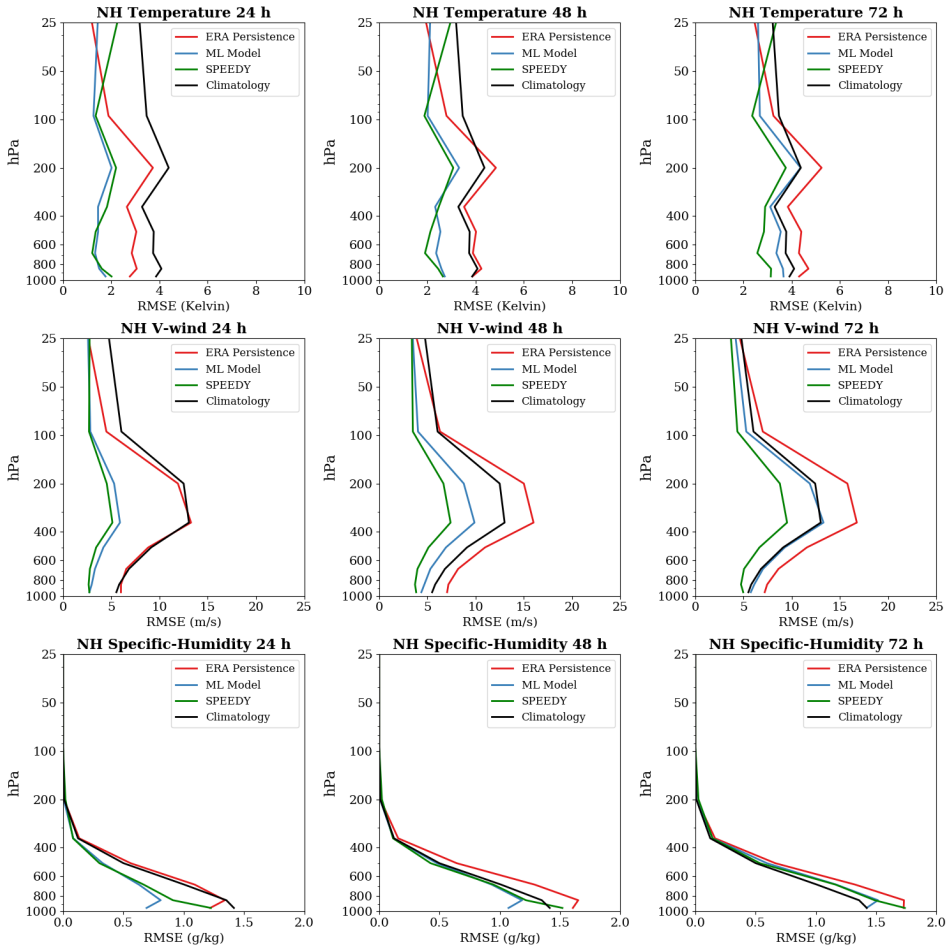

Figure 2.2: Forecast verification results for the NH midlatitudes (30°N and 70°N). Results are shown for (blue) the ML model, (green) SPEEDY, and (red) persistence. Shown is the area-weighted root-mean-square error at the different atmospheric levels for (top row) the temperature, (middle row) meridional wind, and (bottom row) specific humidity at (left column) 24 h forecast time, (middle column) 48 hour forecast time, and (right column) 72 h forecast time. Reprinted with permission from Arcomano et al. (2020).

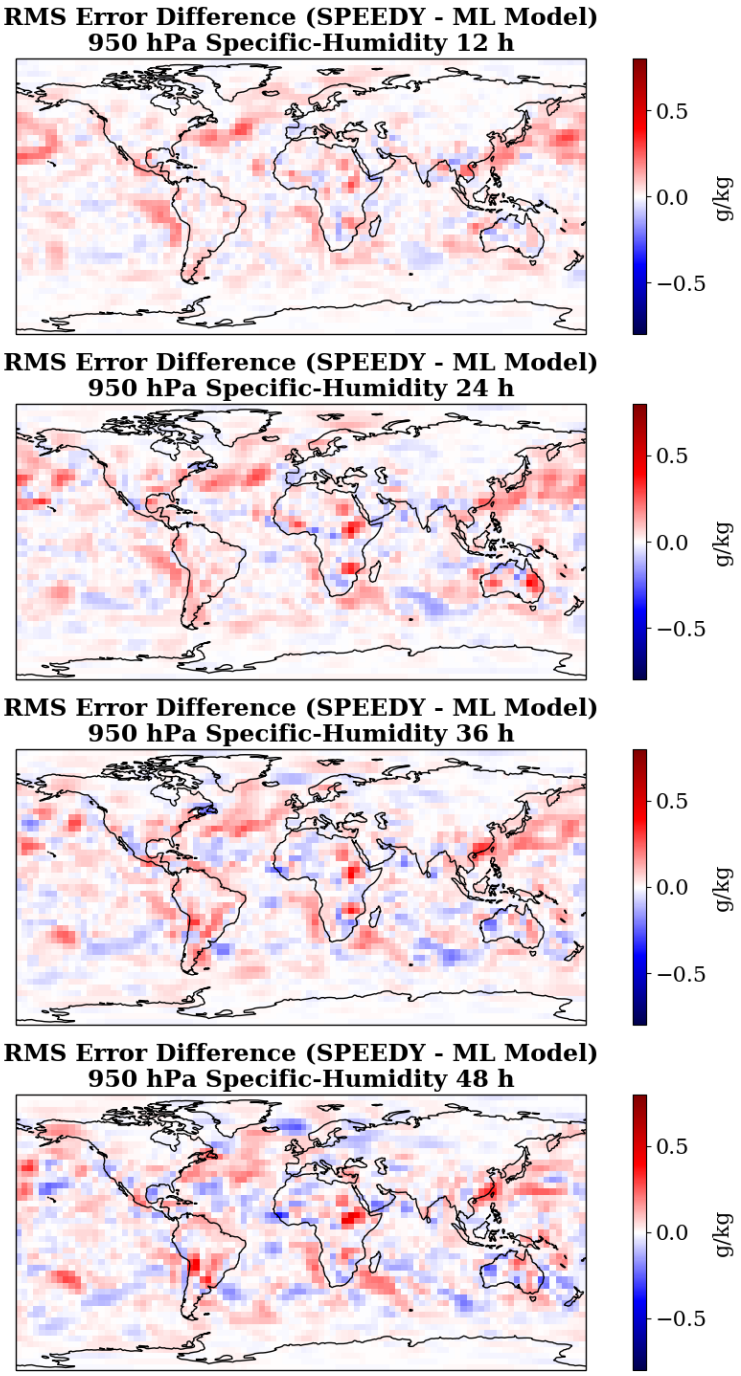

Figure 2.3: Comparison of the near-surface humidity forecast errors between the SPEEDY and ML model forecasts. Shown by color shades is the difference of the 925 hPa relative humidity root-mean-square errors between the SPEEDY and ML model forecasts at forecast times (top) 12 h, (second from top) 24 h, (second from bottom) 36 h, and (bottom) 48 h. Here, the mean is taken over all 171 forecasts. Positive (negative) values indicate locations where the ML model forecasts are more (less) accurate than the SPEEDY forecasts.Reprinted with permission from Arcomano et al. (2020).

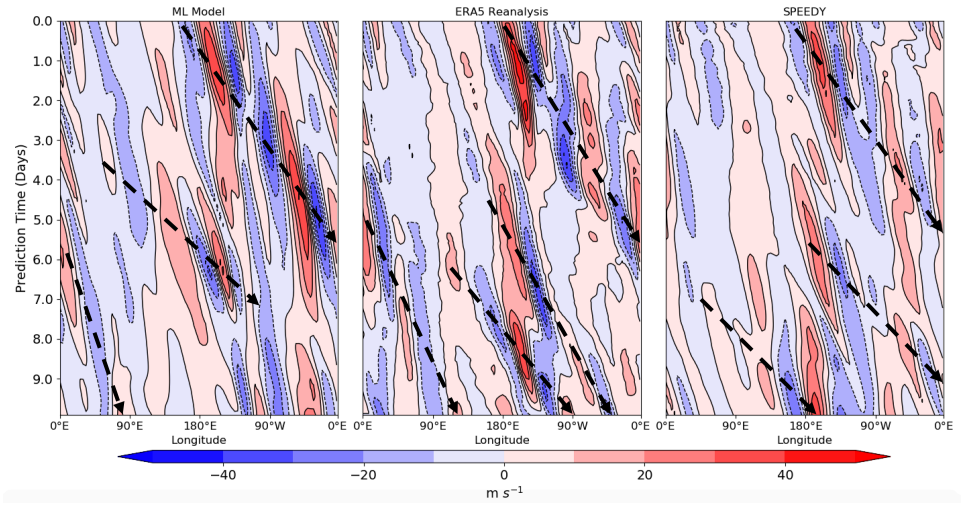

Figure 2.4: Rossby wave packets in the model forecasts and verification data. The Hovmöller diagrams show the propagation of waves packets in (left) a 10-day ML model forecast, (middle) related verification data, and (right) related 10-day SPEEDY forecast. Shown by color shades is the latitude-weighted meridional mean of the meridional coordinate of the wind for latitude band 30°N-60°N at 200 hPa. The forecasts start at 0000 UTC 2 December, 2000. The propagation of the wave packets are marked by the directed straight dashed lines. Reprinted with permission from Arcomano et al. (2020).

2.4 Conclusions

We demonstrated that a RC-based parallel ML model can predict the global atmospheric state in the same gridded format as a numerical (physics-based) global weather prediction model. We found that the 20-day ML model forecasts predicted realistic values of all state variables at all forecast times for the entire globe. The ML model predicted the weather in the midlatitudes more accurately than either persistence or climatology for most of the first three forecast days. This time range could be significantly extended by eliminating, or at least reducing, the over-prediction of atmospheric spatial variability at the large scales (wave numbers lower than 7). The forecast variables for which the ML model performed best compared to a numerical (physics-based) model of identical prognostic state variables and resolution were the ones most affected by parameterized processes in the numerical model.

The results suggests that the current version of our ML model have potential in short-term weather forecasting. Because the parallel computational algorithm is highly scalable, it could be

easily adapted to higher spatial resolutions on a larger supercomputer. As the algorithm is highly efficient in terms of wall-clock time, it could be used for rapid forecast applications and could also be implemented in a limited-area rather than a global setting. The ML modeling technique described here could also be applied to other geophysical fluid dynamical systems.

3. A HYBRID APPROACH TO ATMOSPHERIC MODELING THAT COMBINES MACHINE LEARNING WITH A PHYSICS BASED NUMERICAL MODEL*

3.1 Introduction

Numerical weather prediction (NWP) models have been the backbone of operational weather prediction for several decades now (e.g., Lynch 2006; Harper 2008). A particular model implements a numerical solution algorithm for the physics-based set of coupled partial differential equations that govern atmospheric motion (e.g., Szunyogh 2014). The resulting numerical equations form the *dynamical core* of the model. The effects of processes not resolved explicitly by the dynamical core are taken into account by *parameterization schemes* that contribute to the forcing terms of the equations. These schemes are based on some combination of theoretical and empirical considerations (e.g., Stensrud 2007). The initial conditions of the numerical model solutions are observation-based estimates (analyses) of the state of the atmosphere, and the process that produces these estimates is called *data assimilation* (e.g., Szunyogh 2014). The advances in modeling and data assimilation techniques, alongside with the increase of computing power and the number of observations available for assimilation, led to a *"quiet revolution of NWP"* (Bauer et al. 2015). The incorporation of *machine learning* (ML) techniques into the NWP process promises to lead to further forecast accuracy gains by extracting additional information from the observations.

The earliest applications of machine learning (ML) to atmospheric modeling focused on improving the computational efficiency of the physics-based numerical models (e.g., Krasnopolsky et al. 2005; Krasnopolsky and Fox-Rabinovitz 2006; Krasnopolsky 2013). These applications employed neural networks to emulate the computationally most expensive physics-based parameterization schemes at a reduced computational cost. The term *hybrid model* was first used in reference to models using this technique. One approach employed by this type of hybrid models is to use

*Reprinted with permission from "A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model" by Troy J. Arcomano, I. Szunyogh, A. Wikner, J. Pathak, B. R. Hunt, and E. Ott, 2022. Journal of Advances in Modeling Earth Systems, 14, © Copyright (16 February 2022) American Geophysical Union.

a single neural network to emulate the combined effect of multiple parameterized processes, such as cumulus convection, radiation, boundary layer transport, etc. (e.g., Krasnopolsky et al. 2010; Krasnopolsky 2013; Brenowitz and Bretherton 2018, 2019; Rasp et al. 2018). For this purpose, the ML systems are often trained on data produced by model simulations at higher resolutions, or with more sophisticated physical parameterization schemes.

Another type of ML-based parameterization scheme (e.g., Gentine et al. 2018; Rasp et al. 2018; Chattopadhyay et al. 2020), is trained on observations or observations-based reanalyses. Such a scheme has the potential to learn about the effects of processes that the higher resolution and more sophisticated model simulations are still unable to capture. ML techniques have also been considered for the estimation of the free parameters of physics-based parameterization schemes (Schneider et al. 2017). This approach takes advantage of the knowledge built into the parameterization schemes, but may suffer from the assumptions and approximations made by the schemes.

The hybrid approach we propose belongs to a class of techniques that are different from those mentioned thus far. Techniques of this class use ML for the *frequent periodic interactive correction of the spatiotemporally evolving physics-based numerical model solution* after training on observational analyses. The specific approach we propose was originally developed by (Pathak et al. 2018a) and later adapted to large dynamical systems by (Wikner et al. 2020), who named it *Combined Hybrid-Parallel Prediction* (CHyPP). It evolves the hybrid forecasts iteratively, combining a short-term (e.g., 6 h) numerical forecast with a state-dependent ML correction in each "time step" of the "hybrid model integration". CHyPP is not a postprocessing technique, because each "time step" of the evolving hybrid model solution starts from the ML-corrected state of the preceding step, whereas a postprocessing technique does not interact with the evolving model solution. The ML component of CHyPP uses the computationally highly efficient parallel *reservoir computing (RC)* algorithm of (Pathak et al. 2018b). The other hybrid approaches of the same class use either a random forest (Watt-Meyer et al. 2021) or use a deep learning ML component (Farchi et al. 2021), rather than one based on RC.

Wikner et al. (2020) demonstrated the potential of CHyPP for predicting the evolution of a spa-

tiotemporally chaotic system by experiments with the Kuramoto-Sivashinsky (KS) model (Sivashinsky 1977), a model that has a single state variable that depends only on a single space dimension in addition to time. We implement CHyPP on the *Simplified Parameterization, primitive-Equation Dynamics (SPEEDY)* (Molteni 2003; Kucharski et al. 2006) atmospheric global circulation model (AGCM). Ours is the first implementation of the approach on a model that has multiple state variables with a wide range of values and depend on all three spatial dimensions. Because SPEEDY has a substantially lower resolution than a state-of-the-art NWP or climate model, our primary goal is to demonstrate the feasibility and potentials of CHyPP for an atmospheric application, rather than to propose our current model as a potential replacement for a state-of-the-art numerical model. The results of our forecast experiments show that the performance of the hybrid model is superior to that of either SPEEDY, a model based only on ML, or a model that uses linear regression rather than ML for the correction of the short term ("one time step") numerical forecasts.

In what follows, we first describe the hybrid approach and its implementation on SPEEDY in detail (section 3.2). Then, we discuss the results of the forecast experiments (section 3.3), and then the climate simulation (section 4). Finally, we summarize our key findings and draw our conclusions (section 3.5).

3.2 The Hybrid Model

In CHyPP, the physics-based numerical model state is evolved globally, while the ML correction is done in parallel, in small local domains (Pathak et al. 2018b). The model state of a local domain is represented by a local state vector composed of the relevant components of the global state vector. The global hybrid prediction is obtained by piecing together the local hybrid predictions at the end of each Δt-long "time step" of the "hybrid model integration". This approach can be implemented on any numerical model by adjusting the definition of the local state vectors to the spatial discretization strategy of the model. We note that the localization strategy of CHyPP is similar to that employed by the Local Ensemble Transform Kalman Filter (LETKF) data assimilation scheme (Ott et al. 2004; Hunt et al. 2007; Szunyogh et al. 2008), which has been found to scale

efficiently even for very high (kilometer) resolution operational weather prediction models (e.g., Schraff et al. 2016).

3.2.1 The Global State Vector

SPEEDY is a spectral transform AGCM that was developed to produce rapid climate simulations, using simplified, but modern physical parameterization schemes (Molteni 2003). We implement CHyPP on the standard configuration of Version 41 of the model: the spectral horizontal resolution is T30, while the grid used for the computation of the nonlinear terms and parameterizations has a nominal horizontal spatial resolution of $3.75^\circ \times 3.75^\circ$ with state variables defined at eight vertical σ-levels (0.025, 0.095, 0.20, 0.34, 0.51, 0.685, 0.835, and 0.95), where σ is the ratio of pressure to the surface pressure. The three-dimensionally varying state variables of the model are the two components of the horizontal wind vector, temperature, and specific humidity, while the single two-dimensionally varying state variable is the natural logarithm of surface pressure. The global computational grid and the state variables of the hybrid model are the same as those of SPEEDY.

3.2.2 The Local State Vectors

In our implementation of CHyPP on SPEEDY, each local state vector represents the atmospheric state in a three-dimensional local domain that has the shape of a rectangular box with a $7.5^\circ \times 7.5^\circ$ (2×2 horizontal grid points) base and extends vertically from ground level to $\sigma = 0.025$. (The boundaries of the horizontal footprint of a local domain are marked by a blue rectangle in Fig. 3.1 .) In what follows, we describe the computations carried out in parallel for each of the $L = 1,152$ local domains to evolve the hybrid model state from time t to $t + \Delta t$.

Let $\mathbf{v}(t)$ be the local state vector for an arbitrary local domain at time t. The dimension of this state vector is $4 \times (8 \times 4 + 1) = 132$ (resulting from the 4 grid points of a local domain, the 8 σ-levels, the 4 volume distributed state variables, and the natural logarithm of surface pressure state variable). Because the different state variables have different units and ranges of values, where the ranges also depend on the geographical location and vertical level, each grid-point value of each

state variable is standardized to have a mean of 0 and a standard deviation of 1 before forming $\mathbf{v}(t)$. The standardization is done by using ERA5 reanalysis data (Hersbach et al. 2020) for the computation of the climatological mean and standard deviation of each grid-point variable. We introduce the notation $\mathbf{v}^p(t)$, $\mathbf{v}^h(t)$, and $\mathbf{v}^a(t)$ for the local state vector of SPEEDY, the hybrid model, and the reanalysis, respectively. We also introduce the notations $\mathbf{v}^{gp}(t)$, $\mathbf{v}^{gh}(t)$, and $\mathbf{v}^{ga}(t)$ for the related global state vectors. For instance, the components of $\mathbf{v}^{ga}(t)$ in an arbitrary local domain are the components of $\mathbf{v}^a(t)$. In what follows, we explain the steps of the computation of $\mathbf{v}^{gh}(t+\Delta t)$ from $\mathbf{v}^{gh}(t)$. A flowchart of these steps is shown in Fig. 3.2.a.

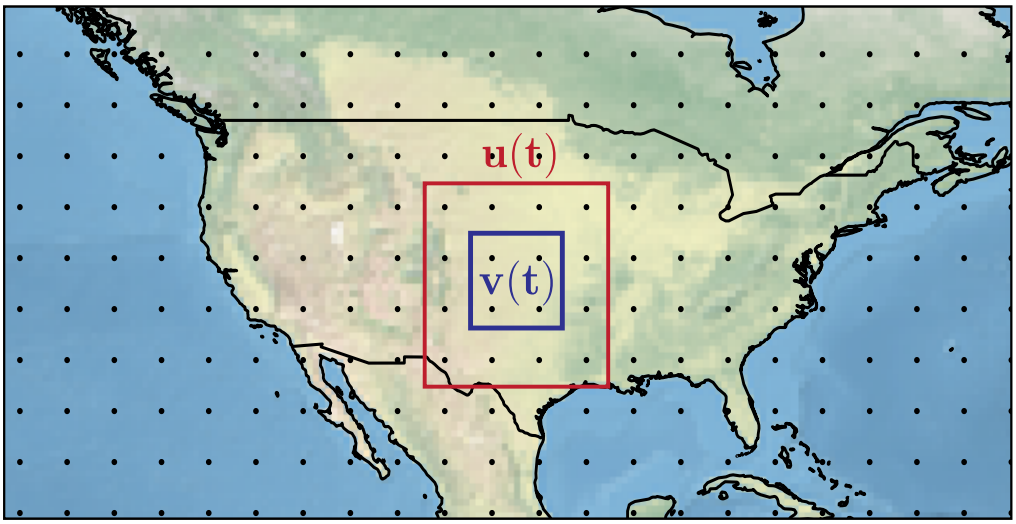

Figure 3.1: Illustration of the localization strategy. The black dots indicate the horizontal locations of the grid-points of the model. The blue rectangle marks the horizontal boundaries of a particular local domain. The red rectangle indicates the horizontal boundaries of the associated extended local domain. Reprinted with permission from Arcomano et al. (2022).

3.2.3 Reservoir Dynamics

The ML model uses (RC) (Jaeger 2001; Lukoševičius and Jaeger 2009; Lukoševičius 2012) to evolve the ML model component from time t to $t+\Delta t$. In RC, the ML model state is evolved by a high-dimensional dynamical system which, for our RC implementation, is defined by the discrete

time map

$$\mathbf{r}(t+\Delta t) = \tanh\left[\mathbf{A}\mathbf{r}(t) + \mathbf{B}\mathbf{u}^{\mathbf{h}}(t)\right]. \tag{3.1}$$

This dynamical system is the *reservoir*, $\mathbf{r}(t)$ is the *reservoir state vector*, and $\mathbf{u}^h(t)$ is the local input state.

During the training, the input term $\mathbf{u}^h(t)$ in Eq. (1) is replaced by $\mathbf{u}^a(t)$. The local input $\mathbf{u}^h(t)$ in our case is a m-dimensional *extended local state vector*, composed of the components of the local state vector $\mathbf{v}^h(t)$ plus additional components of the global state vector $\mathbf{v}^{gh}(t)$ from the neighboring local domains (see Fig. 3.1 for illustration), plus the prescribed incoming solar radiation at the top of the atmosphere for the extended local domain. The latter component is included to help the hybrid model to learn the diurnal cycle from the input data. (SPEEDY uses the daily average value of the incoming solar radiation at the top of the atmosphere at all times of the day.) For all of the local domains, $m = 16 \times (8 \times 4 + 1 + 1)$, except at the local domains adjacent to the poles where $m = 12 \times (8 \times 4 + 1 + 1)$.

Referring to Eq. (1), the dimension D_r of the vector $\mathbf{r}(t)$ is much higher than that of a local state vector $\mathbf{v}^h(t)$ (e.g., 6,000 vs. 132 in the present article). The *activation function* with a vector argument, $\tanh[\cdot]$, is a vector of the same dimension (D_r) as its argument, and a component of this vector is the hyperbolic tangent of the corresponding component of the argument vector. The matrix $\mathbf{A}$ is a sparse $D_r \times D_r$ *weighted adjacency matrix* that represents a low-degree, directed, random graph (Gilbert 1959). Each entry of $\mathbf{A}$ is randomly chosen with a probability κ/D_r of being nonzero, where κ is the degree of the graph (the average number of incoming connections per node), and with the nonzero entries of $\mathbf{A}$ randomly drawn from a zero-mean uniform distribution. (The ratio κ/D_r is a measure of the *sparsity* of $\mathbf{A}$.) After randomization, the entries of $\mathbf{A}$ are scaled such that the largest eigenvalue of $\mathbf{A}$ is a prescribed number ρ ($0 < \rho < 1$), which is called the *spectral radius*. The spectral radius controls the length of the memory of the ML reservoir, and a value $\rho < 1$ typically makes the reservoir state $\mathbf{r}(t)$ depend only on the past states of the modeled system (the atmosphere in our case), and not on the initial reservoir state, when t is sufficiently large. This property of the reservoir is called the *echo state property* (Jaeger 2001).

The matrix-vector product $\mathbf{B}\mathbf{u}^h(t)$ is called the *input layer* in RC. In our model, $\mathbf{B}$ is a $m \times D_r$ sparse random matrix with an equal number of nonzero entries in each row. These nonzero entries, which are chosen randomly from a uniform distribution on the interval $[-\alpha, \alpha]$, couple the components of $\mathbf{u}^h(t)$ to the reservoir nodes. The *input strength* α is an adjustable parameter that controls the degree of non-linearity experienced by the input signal $\mathbf{u}^h(t)$ from the activation function.

3.2.4 The Hybrid Model

In addition to providing the input for Eq. (1), the global state $\mathbf{v}^{gh}(t)$ is used as the initial condition for a SPEEDY model forecast $\mathbf{v}^{gh}(t+\Delta t)$. The next local hybrid model prediction is then obtained by

$$\mathbf{v}^h(t+\Delta t) = \mathbf{W} \begin{bmatrix} \mathbf{v}^p(t+\Delta t) \\ \widetilde{\mathbf{r}}(t+\Delta t) \end{bmatrix}, \tag{3.2}$$

where the components $\widetilde{r}_i(t+\Delta t)$ of the column vector $\widetilde{\mathbf{r}}(t+\Delta t)$, $i = 1, 2, ...D_r$ are defined by $\widetilde{r}_i(t+\Delta t) = r_i(t+\Delta t)$, if i is odd, and $\widetilde{r}_i(t+\Delta t) = r_i^2(t+\Delta t)$, if i is even, and the column vector $\mathbf{v}^p(t+\Delta t)$ represents the local state corresponding to the global SPEEDY forecast $\mathbf{v}^{gp}(t+\Delta t)$. The matrix-vector product on the right-hand side of Eq. (3.2) is the RC *output layer*. The matrix $\mathbf{W}$ is a *matrix of parameters* to be determined by the training procedure described in Sec. 2.4.1. The local vectors $\mathbf{v}^h(t+\Delta t)$ for each local domain are combined to form the next global hybrid model prediction $\mathbf{v}^{gh}(t+\Delta t)$.

Equation (3.2) can be written in the equivalent form

$$\mathbf{v}^h(t+\Delta t) = \mathbf{W}_{mod}\mathbf{v}^p(t+\Delta t) + \mathbf{W}_{res}\tilde{\mathbf{r}}(t+\Delta t), \tag{3.3}$$

which corresponds to $\mathbf{W} = [\mathbf{W}_{mod}\ \mathbf{W}_{res}]$. In the extreme case that $\mathbf{W}_{mod} = 0$, which should be the result of training when the numerical model has no skill according to the training data, the hybrid prediction completely ignores the numerical model forecast $\mathbf{v}^p(t+\Delta t)$. The other extreme case is

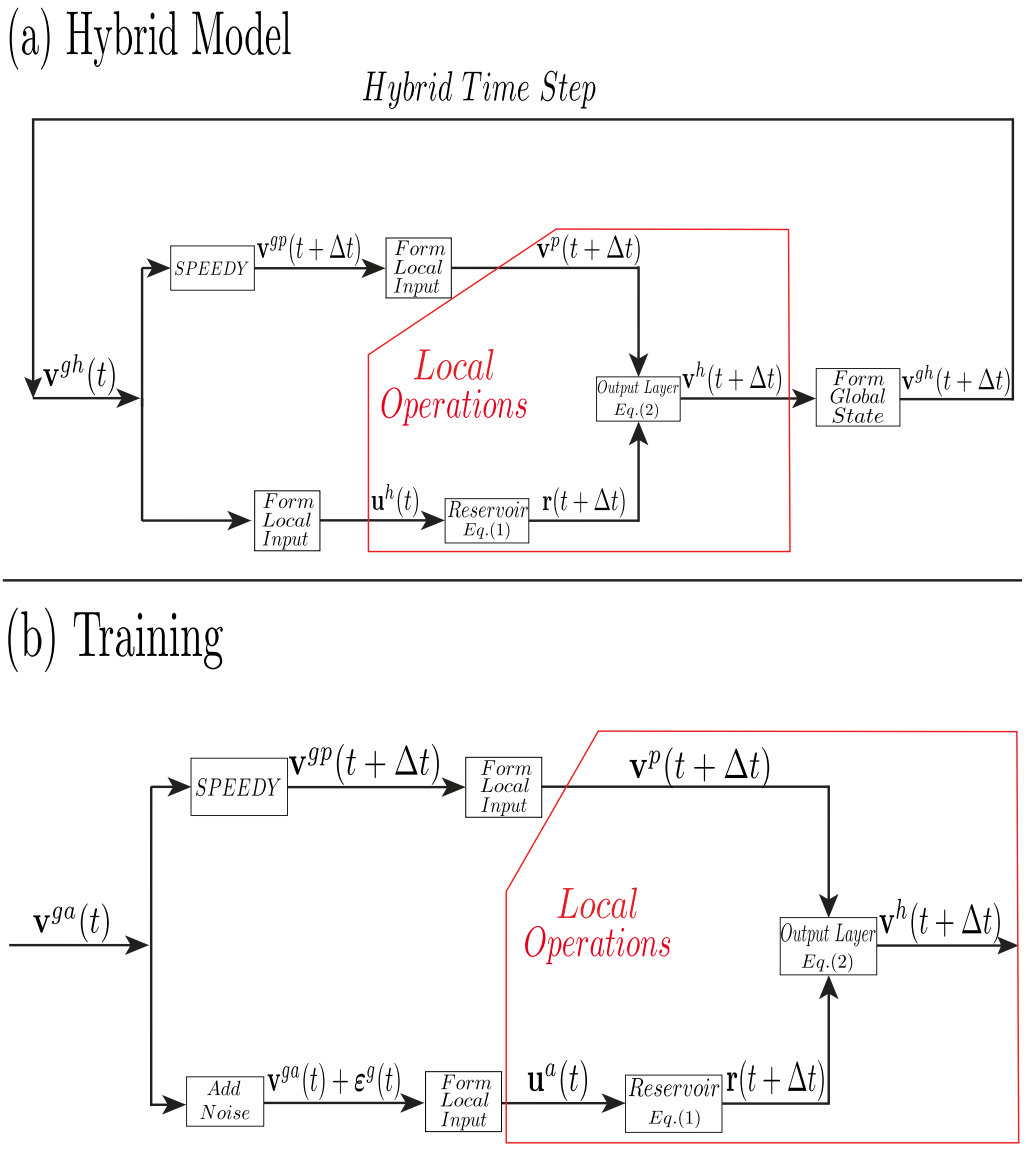

Figure 3.2: A flow chart of (a) the hybrid model and (b) the training operation of the hybrid model. The notation is defined in Secs. 2.2 and 2.3. The steps inside the red boxes are carried out in parallel for each of the $L = 1,152$ local domains. The training finds the $\mathbf{W}$ that minimizes the cost function of Eq. (4) by solving Eq. (5). Reprinted with permission from Arcomano et al. (2022)

when $\mathbf{W}_{mod} = \mathbf{I}$ and $\mathbf{W}_{res} = 0$, which should occur when the numerical model is perfect according to the training data. In a typical case, which falls between the two extremes, the ML output and the Δt-long numerical prediction are combined to maximize agreement with the training data.

3.2.4.1 Training

Figure 2.b shows the flow of operations during training. First, we generate a sequence of perturbed global analyses $\mathbf{v}^{ga}(k\Delta t) + \boldsymbol{\varepsilon}^{g}(k\Delta t)$, $k = -K - K_t, -K - K_t + 1, ..., -1$, where $\boldsymbol{\varepsilon}^{g}(k\Delta t)$ is a small-magnitude, zero-mean, normally distributed random noise vector, uncorrelated in time and uncorrelated between components of the noise vector. The role of this noise is to help the ML model learn to return to the bounded set of realistic atmospheric states (the "attractor") in the presence of perturbations that may arise in future forecasts (e.g., Jaeger 2001; Wikner et al. 2020). The addition of noise to the global analyses during training is essential for the hybrid model to produce stable, realistic predictions; predictions rapidly become unstable without it. Similar behavior has been observed in RC applications involving the prediction of other spatio-temporal systems (e.g., Patel et al. 2021).

The local input state $\mathbf{u}^{a}(k\Delta t)$ is the extended local state vector associated with $\mathbf{v}^{ga}(k\Delta t) + \boldsymbol{\varepsilon}^{g}(k\Delta t)$, for $k = -K - K_t, -K - K_t + 1, ..., -1$ for the particular local domain. The initial state $\mathbf{r}[(-K-K_t)\Delta t]$ of the reservoir can be chosen arbitrarily, because only the evolved reservoir states $\mathbf{r}[(k+1)\Delta t], k = -K, -K+1, \ldots, -1$, are used for training. The purpose of discarding the reservoir state of the first K_t ($K_t \ll K$) iterations is to ensure that the reservoir state $\mathbf{r}(t)$ has sufficient time to settle on its attractor. The unperturbed global analyses $\mathbf{v}^{ga}(k\Delta t)$ are also used as the initial conditions for SPEEDY to obtain $\mathbf{v}^{gp}[(k+1)\Delta t]$ for $k = -K, -K+1, ..., -1$.

Formally, the training is carried out by computing the weight matrix $\mathbf{W} = [\mathbf{W}_{mod}\ \mathbf{W}_{res}]$ that minimizes the cost-function

$$J(\mathbf{W}) = \sum_{k=-K+1}^{0} \|\mathbf{v}^{h}(k\Delta t, \mathbf{W}) - \mathbf{v}^{a}(k\Delta t)\|^2 + \beta_{mod}\|\mathbf{W}_{mod} - \mathbf{W}_{prior}\|^2 + \beta_{res}\|\mathbf{W}_{res}\|^2. \qquad (3.4)$$

The local hybrid states $\mathbf{v}^{h}(k\Delta t, \mathbf{W}), k = -K+1, -K+2, ..., 0$, represent the results of Eq. (3.2) at those times for a particular $\mathbf{W}$, and $\mathbf{v}^{a}(k\Delta t)$ is the local state vector for the unperturbed global analysis $\mathbf{v}^{ga}(k\Delta t)$. (Notice that we use the notation $\mathbf{W}$ for both the variable and the solution of

the minimization problem.) The last two terms of the cost function, in which $\|\cdot\|^2$ denotes the sum of the squares of the entries of a matrix (the Frobenius norm), are regularization terms meant to prevent overfitting, with β_{mod} and β_{res} being the regularization parameters for the numerical model and reservoir component, respectively. With these terms, the direct solution of the least-square problem is a *ridge regression* (Tikhonov and Arsenin 1977). The inclusion of the *prior matrix* $\mathbf{W}_{prior}$, which was not part of Wikner et al. (2020), allows for a choice like $\mathbf{W}_{prior} = \mathbf{I}$, which dictates that in the absence of training data that demonstrates imperfections in the numerical model, the hybrid model should be equivalent to the numerical model. In our experiments, we tried both $\mathbf{W}_{prior} = \mathbf{I}$ and $\mathbf{W}_{prior} = 0$, and found that the latter yielded better stability. Thus, we report results with $\mathbf{W}_{prior} = 0$, but think that other choices for nonzero $\mathbf{W}_{prior}$ merit further study.

To obtain the direct solution for the matrix $\mathbf{W}$ that minimizes the cost function J, we define matrix $\widetilde{\mathbf{R}}$ by choosing its column k to be $\widetilde{\mathbf{r}}(k\Delta t)$ (see Eq. (3.2)), and matrix $\mathbf{V}_p$ by choosing its column k to be the $\mathbf{v}^p(k\Delta t)$ local state vector that corresponds to the global SPEEDY forecast from $\mathbf{v}^{ga}((k-1)\Delta t)$. In addition, we define matrix $\mathbf{V}_a$ by selecting its column k to be the local analysis $\mathbf{v}^a(k\Delta t)$. Then, it can be shown that the minimizing $\mathbf{W}$ is the solution of the linear problem

$$\mathbf{W}\begin{bmatrix} \mathbf{V_p}\mathbf{V_p^T} + \beta_{mod}\mathbf{I} & \mathbf{V_p}\widetilde{\mathbf{R}}^{\mathbf{T}} \\ \widetilde{\mathbf{R}}\mathbf{V_p^T} & \widetilde{\mathbf{R}}\widetilde{\mathbf{R}}^{\mathbf{T}} + \beta_{res}\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{V_a}\mathbf{V_p^T} + \beta_{mod}\mathbf{W}_{prior} & \mathbf{V_a}\widetilde{\mathbf{R}}^{\mathbf{T}} \end{bmatrix} \tag{3.5}$$

for $\mathbf{W}$.

Because the dimension of the matrix products in this problem does not depend on the length $K\Delta t$ of the training period, the matrix products can be computed incrementally, without simultaneously storing every column of $\widetilde{\mathbf{R}}$, $\mathbf{V}_p$, or $\mathbf{V}_a$ in memory (e.g., Lukoševičius 2012). That is, in terms of computer memory usage, the resources used by the training do not depend on the length of the training period. This is a highly desirable property for Earth system modeling, in which long training periods are expected to be necessary. In addition, the corresponding columns of $\tilde{\mathbf{R}}$, $\mathbf{V}_p$, and $\mathbf{V}_a$ can be obtained by training on multiple time series of training data. For example, suppose that the global analyses $\mathbf{v}^{ga}(t)$ have a temporal resolution Δt_a that is finer than the Δt temporal

resolution of the hybrid model with $\Delta t = J\Delta t_a$, where J is an integer. Then, the number of time series available for training is J; i.e., the first term in Eq. (3.4) can be replaced by

$$\sum_{j=0}^{J-1} \sum_{k=-K+1}^{0} \|\mathbf{v}^h(k\Delta t - j\Delta t_a, \mathbf{W}) - \mathbf{v}^a(k\Delta t - j\Delta t_a)\|^2. \tag{3.6}$$

3.2.4.2 Synchronization and Prediction

Let $K_f\Delta t$ be the forecast start time. Starting the hybrid forecast requires the availability of the global analysis $\mathbf{v}^{ga}(K_f\Delta t)$ and the reservoir state $\mathbf{r}(K_f\Delta t)$ for each local domain. Because according to the "echo state property" $\mathbf{r}(K_f\Delta t)$ is determined by the past states of the atmosphere, it can be obtained by synchronizing the evolution of the reservoir states with the analyses for a sufficiently long time period that ends at $K_f\Delta t$. Let $K_s\Delta t$ be the start time of the synchronization. Synchronization is achieved by evolving the reservoir equation using $\mathbf{u}^h(k\Delta t) = \mathbf{u}^a(k\Delta t)$ in Eq. (3.1) for $k = K_s, K_{s+1}, \ldots, K_f$.

Piecing together the local hybrid forecasts for all local domains yields the global "one-step" hybrid forecast $\mathbf{v}^{gh}[(K_f + 1)\Delta t]$ (Fig. 3.2.a). The forecast can be extended arbitrarily far into the future by using an iterative process for $k = K_f{+}1, K_f{+}2, \ldots$, in which the extended local state vector $\mathbf{u}^h(k\Delta t)$ extracted from $\mathbf{v}^{gh}(k\Delta t)$ is used as $\mathbf{u}^h(k\Delta t)$ in the Eq. (3.1) to compute $\mathbf{r}[(k+1)\Delta t]$. The global "one-step" hybrid forecast $\mathbf{v}^{gh}(k\Delta t)$ is also used as the initial condition of the $\mathbf{v}^{gh}[(k+1)\Delta t]$ SPEEDY component of the hybrid forecast. In a cycled forecast system of an operational NWP center, in which analyses are prepared and forecasts are started with a regular frequency (e.g., 6 h), the reservoir state can be kept continuously synchronized with the real-time evolution of the atmosphere.

3.2.5 Implementation with ERA5 Reanalysis Data

We use interpolated hourly global ERA5 reanalyses to train and synchronize the hybrid model. We do the horizontal interpolation of the reanalysis fields onto the computational grid of SPEEDY by a 2-dimensional quadratic B-spline interpolation. We then compute the value of σ at each hori-

zontal grid point and use a 1-dimensional cubic B-spline for the vertical interpolation of the model state variables to the eight prescribed constant σ levels of SPEEDY. The training starts at 0000 UTC on January 1, 1990 and ends at 2300 UTC on June 26, 2011 ($K \approx 3.14 \times 10^4$), with the data discarded for the first 6.25 days ($K = 31355$ and $K_t = 25$).

3.2.6 Selection of the Hyperparameters

Hyperparameters are adjustable parameters (e.g. κ, ρ, α, D_r,β_{res}, β_{mod}, ε, and Δt) that control overall characteristics of the hybrid model and require "tuning" to produce desirable results. There exists "tricks of the trade" practical rules for the selection of the hyperparameters of an RC model (Lukoševičius 2012). These general rules also work for the hyperparameters of the hybrid model. First, the hybrid model is only weakly sensitive to κ and ρ. While we use $\kappa = 6$, other small values of κ (e.g., $\kappa = 3$) work similarly well. We use a value of ρ that monotonically increases toward the poles from 0.3 at the equator to 0.7 at 45°, so that the reservoir mimics the general property of the atmospheric dynamics that its memory is shorter in the tropics than the extratropics. Changing these values by $\pm 0.1 - 0.2$ has little effect on the model performance. We choose $D_r = 6,000$, because we find that further increasing the reservoir size does not lead to substantial further improvement of the model performance. We find the hybrid model performance to be somewhat sensitive to the value of α, which controls the amount of nonlinearity of the reservoir dynamics. Setting $\alpha \leq 0.3$ or $\alpha \geq 0.7$ yields noticeable degradation of the errors compared to the value we use, $\alpha = 0.5$. For each of the options $\mathbf{W}_{prior} = \mathbf{I}$ and $\mathbf{W}_{prior} = 0$, we tried various powers of 10 for the regularization parameters β_{res} and β_{mod}; we found that $\mathbf{W}_{prior} = 0$ yielded better stability, and found that $\beta_{res} = 10^{-4}$ and $\beta_{mod} = 10^0$ led to good model performance. Among the several values we tried, in increments of 0.05, for the standard deviation of the components of the random noise ε added to the training data, we chose the smallest value (0.20) for which all hybrid forecasts were stable. The time step Δt is another important hyperparameter to tune; we chose $\Delta t = 6$ h, because using $\Delta t = 1$ h or $\Delta t = 3$ h (with other hyperparameters tuned accordingly) led to clearly poorer model performance. Moreover, we use a time step of $\Delta t/24 = 0.25$ h for the numerical integration of SPEEDY, because

longer time steps degraded the 6 h forecast performance of SPEEDY. Since the temporal resolution of the ERA5 reanalyses is 1 h ($\Delta t_a = 1$), the training is done on $\Delta t/\Delta t_a = 6$ time series of data.

3.3 Forecast Experiments

We compute forecast error statistics based on 100 21-day forecasts, with start times equally spaced every 4 days between 0000 UTC, June 27, 2011 and 0000 UTC, July 28, 2012. We evaluate the forecast performance of the hybrid model by comparing it to that of a variety of benchmark forecasts started from interpolated ERA5 reanalyses.

3.3.1 Benchmark Forecasts

The set of benchmark forecasts includes numerical forecasts produced by SPEEDY, a model based only on ML, and a model in which the 6 h SPEEDY forecasts are corrected by linear regression rather than by ML. We call the latter benchmark SPEEDY-LLR, where LLR stands for *local linear regression*.

Comparing the performance of the hybrid model to that of a model based only on ML is important, because ML-only models (e.g., Arcomano et al. 2020; Rasp and Thuerey 2021; Weyn et al. 2020) are considered a potential alternative to the hybrid approaches for the utilization of ML in Earth system modeling. Our ML model is formally the same as our hybrid model except that we use the constraint $\mathbf{W}_{mod} = 0$ in Eq. (3), with Eqs. (4) and (5) modified accordingly, and the hyperparameters are different: $D_r = 9,000$, $\beta_{res} = 10^{-6}$, $\Delta t = 3$ h, and ε has a standard deviation of 0.28. (The smaller reservoir size necessary to obtain good results from the hybrid as compared to the ML-only model is an important advantage of the hybrid model.) While this ML-only model is formally identical to the one described by Arcomano et al. (2020), its forecast performance is better, thanks mainly to using a time step of Δt =3 h rather than Δt =1 h and the addition of the incoming solar radiation to the input of the reservoir.

The SPEEDY-LLR is the same as the hybrid model except that $\mathbf{W}_{res} = 0$. In this model, a larger regularization parameter is necessary to produce stable forecasts for at least 10 days. We use $\beta_{mod} = 1600$, which provides the most accurate short and medium range (1-5 days) forecasts

that also remain stable for at least 10 days. The stability of the SPEEDY-LLR forecasts can be improved by further increasing β_{mod}, but only at the price of degrading the short and medium range forecast accuracy. (For $\beta_{mod} \rightarrow \infty$, SPEEDY-LLR becomes SPEEDY, which produces stable forecasts for indefinitely long lead times). Since, SPEEDY-LLR does not include the nonlinear ML correction of the hybrid model (the second term on the right side of Eq. (3)), training is a simple linear regression of the numerical model forecast. With the help of this benchmark, we can assess the relative importance of making periodic corrections to the numerical forecasts based on linear regression of the model state alone versus making those corrections by the proposed hybrid technique.

To assess whether a model forecast has skill, the figures also include comparisons to forecasts based on persistence and daily climatology. The persistence forecasts are based on the assumption that the state of the atmosphere at the beginning of the forecast persists for the entire duration of the forecast, while the climatological forecasts are based on the daily climatological mean for the calendar day at the particular geographical location and pressure level for years 1990-2010.

3.3.2 The Measure of the Forecast Error

The error of each forecast is measured by the area-weighted *root-mean-square error*,

$$RMSE = \sqrt{\frac{1}{N_{lon}N_{lat}} \sum_{i=1}^{N_{lon}} \sum_{j=1}^{N_{lat}} a(j)(V^f_{i,j} - V^a_{i,j})^2}, \tag{3.7}$$

where,

$$a(j) = \frac{\cos(\varphi(j))}{\frac{1}{N_{lat}} \sum_{j=1}^{N_{lat}} \cos(\varphi(j))}. \tag{3.8}$$

Here the subscript i,j refers to the value of a scalar state variable V for a specific forecast lead time at a particular pressure level at grid point i,j of the verification region defined by N_{lon} discrete longitudes and N_{lat} discrete latitudes. The RMSE is averaged over the 100 forecasts to obtain a single scalar measure of the forecast error for each state variable, pressure level, and forecast lead

time. In what follows, the term *forecast error* refers to this scalar measure. We call a forecast more accurate than another, if the forecast error is lower for the former than the latter forecast. In addition, we say that a model forecast has *forecast value*, if its forecast error is lower than that of both persistence and climatology (the latter two are available without the substantial cost of preparing model forecasts). The qualitative behavior of the errors of the model forecasts with respect to the errors of these two references is well understood. In particular, if the model has realistic climatology, in the sense that it represents the atmospheric variability (the variability of the atmospheric state) correctly, the error of the model forecasts and the error of persistence saturate at the same level. While the error is initially lower for persistence than climatology, its saturation value is higher by a factor of $\sqrt{2}$ (e.g., section 3.8 of Szunyogh (2014)).

3.3.3 Comparisons of the Forecast Accuracy

3.3.3.1 Synopsis of the Forecast Verification Results

Figures 3.3 and 3.4 illustrate the temporal evolution of the forecast errors for the first five forecast days in the NH midlatitudes and Tropics, respectively. The errors are shown for the temperature (top row), meridional component of the wind vector (middle row) and specific humidity (bottom row) at forecast lead times day 1 (left column), day 3 (middle column), and day 5 (right column). In general, the hybrid forecasts (blue curves) have forecast value, except for the specific humidity at day 5 in the NH midlatitudes, for which they are only about as accurate as the forecasts based on climatology. In addition, the hybrid forecasts are either more accurate than all benchmark forecasts, or similarly accurate to the most accurate benchmark forecast. The hybrid model performance in the SH midlatitudes (not shown) is similar to that in the NH midlatitudes. The advantage of the hybrid model compared to the different benchmarks, however, strongly depends on the forecast variable and lead time. Next, we discuss this dependence, as it provides important insight into the mechanisms by which CHyPP improves the numerical forecasts.

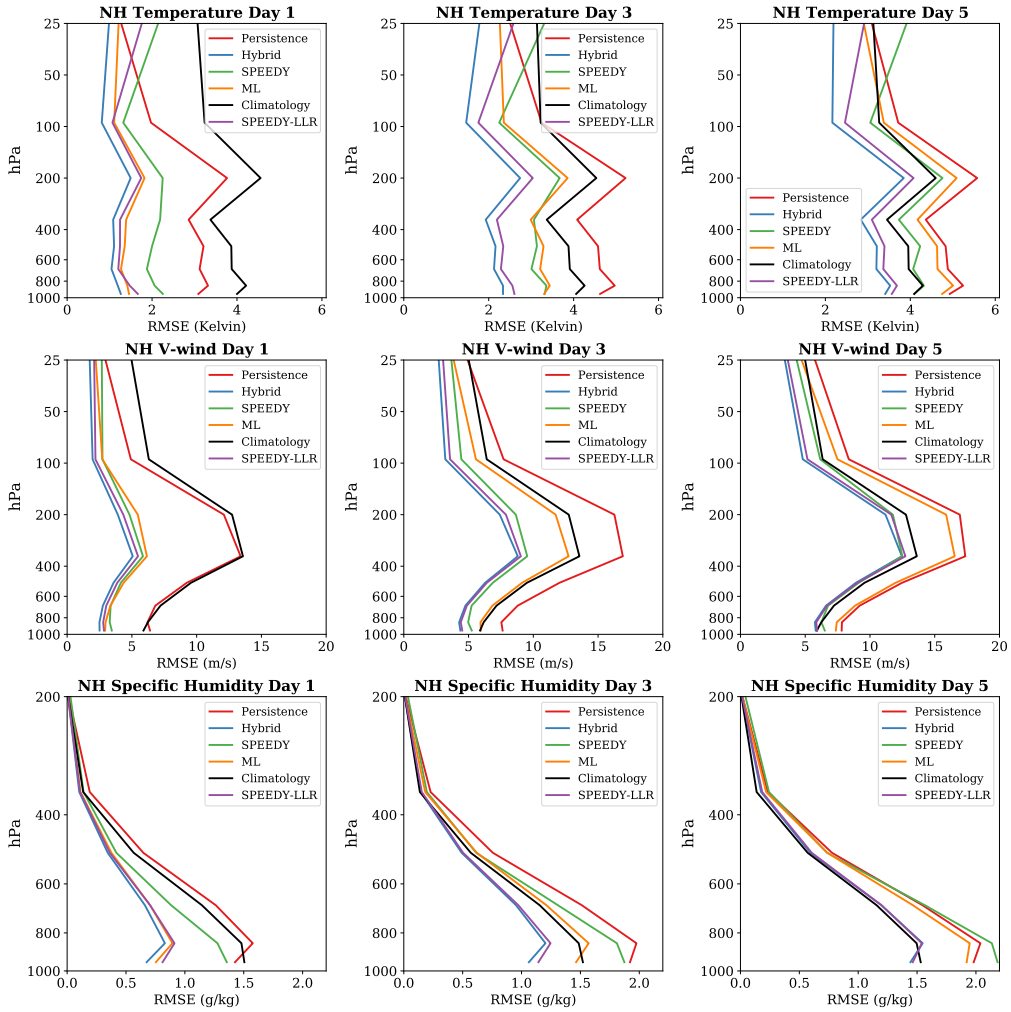

Figure 3.3: Northern Hemisphere midlatitudes (between 30°N and 70°N) forecast verification results. Results are shown for the (blue) hybrid model, (green) SPEEDY, (orange) ML-only model, (purple) SPEEDY-LLR model, (red) persistence, and (black) climatology. Shown is the area□weighted root□mean□square error at the different atmospheric levels for (top row) the temperature, (middle row) meridional wind, and (bottom row) specific humidity at (left column) day 1, (middle column) day 3, and (right column) day 5 forecast time. Reprinted with permission from Arcomano et al. (2022).

3.3.3.2 Hybrid Versus SPEEDY Forecasts

Compared to SPEEDY, the advantage of the hybrid model is the largest for the temperature. While all hybrid temperature forecasts have substantial forecast value for the first 5 forecast days, the SPEEDY day 5 temperature forecasts have no forecast value in the Tropics and in the stratosphere in the NH midlatitudes. In addition, the SPEEDY forecasts have little forecast value at day

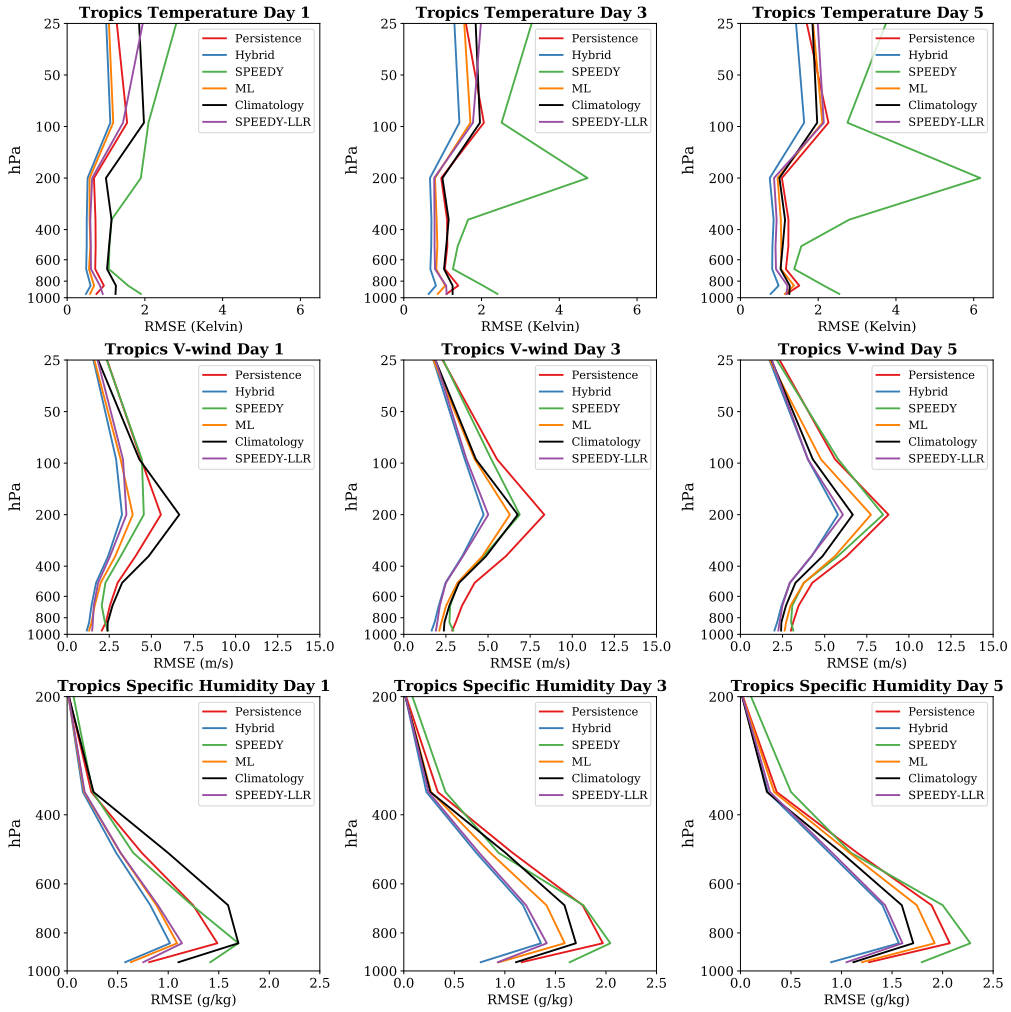

Figure 3.4: As in Fig. 3.3 for the Tropics (between 30°S and 30°N). Reprinted with permission from Arcomano et al. (2022).

5 in the midlatitudes. The benefit of the ML correction is particularly striking in the tropical upper troposphere, where the SPEEDY forecasts have a large error with a maximum of 6 K at 200 hPa, while the error of the hybrid forecasts remains below 1 K.

In addition to the temperature, the hybrid forecasts are also substantially more accurate than the SPEEDY forecasts for the specific humidity, especially, in the lower troposphere, where parameterizations play an important role in modeling the effects of moist atmospheric processes. While in the NH midlatitudes the hybrid forecasts degrade only to the level of the forecasts based on climatology by day 5, the error of the SPEEDY forecasts reaches saturation by that time.

In the two midlatitudes, the state variable for which the advantage of the hybrid model is the smallest compared to SPEEDY is the meridional component of the wind vector. This result is not surprising, as numerical models are known to capture synoptic-scale Rossby wave dynamics, which dominate the variability of weather in the midlatitudes. In contrast, in the Tropics, where wave dynamics is coupled to the parameterized process of deep convection, the advantage of the hybrid model for the meridional wind component is more substantial.

To explore the scale-dependence of the performance of the hybrid and benchmark forecasts, we examine the spectrum of the errors for the meridional component of the wind at 500 hPa with respect to the zonal wave number (Figure 3.5). (This figure also shows results for day 10, in addition to the results for forecast days 1, 3, and 5.) The left panel shows the results for the hybrid and the SPEEDY model. Because SPEEDY is a spectral transform model with cut-off wave number 30, the spectrum for SPEEDY has no power at all beyond that wave number, and it is heavily dampened at wave numbers larger than about 20. Therefore, the errors of the hybrid forecasts, which have realistic power at all wave numbers, are expected to saturate at a level that is higher than that for SPEEDY at the tail-end of the spectrum. At day 1, the hybrid forecasts have a clear advantage over the SPEEDY forecasts at the synoptic and large scales (zonal wave numbers lower than about 20). A smaller, but spectrally similar advantage still exists at day 3, while the advantage of the hybrid forecasts disappears, except at wave numbers 5 and 6, by about day 5.

3.3.3.3 Hybrid Versus ML-only Forecasts

While the errors of the ML-only forecasts (orange curves in Figs. 3.3- 3.5) are only slightly larger than that of the hybrid forecasts at day 1, they grow much faster in the next four days and the ML forecasts typically have no value by day 3. This result suggests that while the RC-based ML technique can produce accurate forecasts in the short range (day 1-2), it is more effective in assisting SPEEDY than directly predicting the weather beyond that range. A comparison of the left and middle panels of Fig. 3.5 suggests that the information provided by SPEEDY to the hybrid is particularly beneficial at the large scales (wave numbers lower than about 6).

3.3.3.4 Hybrid Versus SPEEDY-LLR Forecasts

Next to the hybrid model, the benchmark that performs the best in the medium (day 2-5) forecast range is the SPEEDY-LLR (purple curves). While the hybrid forecasts are more accurate than the SPEEDY-LLR forecasts, the forecast error differences between the two models are modest, except for those in the stratosphere. The fact that the forecast error differences are smaller for the hybrid model versus SPEEDY-LLR than for the hybrid model versus SPEEDY indicates that the periodic interactive correction of the SPEEDY forecasts itself makes an important contribution to the good performance of the hybrid model. The additional forecast improvement, however, is not the only benefit of using ML rather than local linear regression for the forecast correction: while the hybrid forecasts remain stable indefinitely (see section 3.4), some of the SPEEDY-LLR forecasts fail as early as day 11 lead time, with about 60% of the forecasts reaching the intended 21 days.

It should be noted that the fact that local linear regression can efficiently correct the errors of a 6 h forecast is not completely surprising, considering that linear regression can be used to model the short-term forecast error dynamics for even a state-of-the-art NWP model (Bishop et al. 2017), in which nonlinear effects are expected to play a more important role even at short lead times. It is a nontrivial result, however, that the information provided by such a linear approach can be used for the periodic, interactive correction of an evolving numerical forecast. It is also a nontrivial result that an RC-based ML technique stabilizes the resulting hybrid model indefinitely, and leads to further forecast improvement in the short and medium (day 1-5) range.

3.3.4 Global Mean and Spatially Varying Errors

To gain further insight into the ways the hybrid approach improves forecast performance, we decompose the global RMSE into a bias and a standard deviation component. (The sum of the squares of the two components is equal to the square of the root-mean-square error.) The bias measures the global mean error, while the standard deviation measures the spatially varying part of the forecast error. The time evolution of the two error components, averaged over the 100 forecasts is shown for three representative state variables in Fig. 3.6.

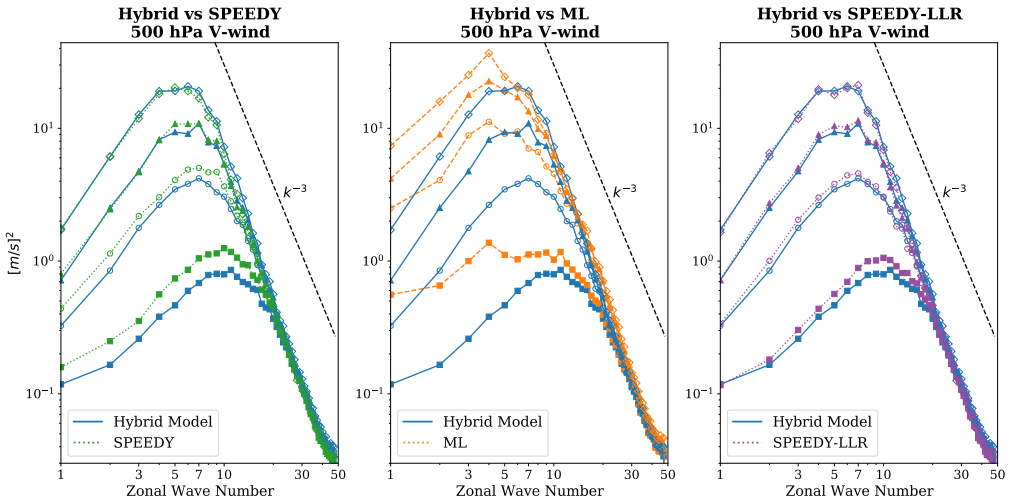

Figure 3.5: Spectral distribution of the 500 hPa meridional wind forecast error in the NH midlatitudes (between 30°N and 70°N) with respect to the zonal wave number. The power spectra of the forecast errors are shown (left) for the the hybrid model (blue) vs SPEEDY (green), (middle) the hybrid model (blue) vs the ML-only model (orange), and (right) hybrid model (blue) vs SPEEDY-LLR (purple) at day 1 (solid square), day 3 (open circle), day 5 (solid triangle), and day 10 (open diamond). Reprinted with permission from Arcomano et al. (2022).

For the temperature near the surface (at 950 hPa, top panel), SPEEDY rapidly develops a warm bias that oscillates around a mean of 0.75 K with the diurnal cycle. This bias is the result of SPEEDY using a single daily average value of the incoming solar radiation at the top of the atmosphere at all times of the day. The hybrid model greatly reduces the magnitude of the bias and also removes its diurnal oscillation. The biases of the ML model and SPEEDY-LLR are comparable to that of the hybrid model in magnitude, but the SPEEDY-LLR bias exhibits diurnal variability.

The spatially variable component of the low-level temperature error remains lower for the hybrid model than for SPEEDY throughout the 14-day period shown in the figure. The same component is initially similarly low for the hybrid and ML-only model, but it increases much more rapidly for the ML-only model. (Even with this rapid increase, the ML-only forecasts remain more accurate than the SPEEDY forecasts until about day 4). This component is initially lower for the hybrid model than for SPEEDY-LLR, but their accuracies are essentially the same after about day 8. Also, while the curves for SPEEDY and the hybrid model saturate at the same level as persistence, the curve for the ML-only model saturates at a higher level, indicating that the ML-only model

overestimates the spatial variability of the low-level temperature at the longer forecast times.

SPEEDY rapidly develops a positive specific humidity bias near the surface (950 hPa, middle panel) that saturates at about 1 g/kg at day 7 lead time. Both the hybrid model and the other two benchmarks eliminate most of this bias. The spatially varying component of the error behaves similarly to that for the low level temperature, with the hybrid model outperforming the benchmarks for lead times from 1-7 days.

For the meridional wind component in the upper troposphere (200 hPa, bottom panel) none of the models develop a noteworthy bias. Thus, the differences in forecast performance are solely due to differences in the spatially varying component of the forecast error. This error component is still smaller for the hybrid model than SPEEDY for the first 9 forecast days, and than for the other benchmarks for the the first 6 forecast days.

3.3.5 Atmospheric Balance

Maintaining the delicate balance between the wind (momentum) and mass field in a numerical model, especially at short forecast lead times, has been one of the biggest challenges of atmospheric modeling since the dawn of NWP (e.g., Lynch 2006). In a modern NWP model, a weakened balance is a short-lived transient property and the magnitude of the initial transient can be greatly reduced by *initialization* techniques (e.g., section 8 of Lynch (2006)). In the hybrid model and SPEEDY-LLR, however, no initialization is done before a corrected 6 h forecast is used as the initial condition of the next 6 h numerical forecast. Hence, the corrections inevitably upset the balance in the numerical component of the hybrid forecasts every 6 h. The forecast verification results discussed thus far suggest that these imbalances do not outweigh the positive effects of the corrections on the accuracy of the hybrid forecasts. But, can the hybrid model produce realistic surface pressure tendencies by also correcting the surface pressure field for the effects of gravity waves excited by the imbalances? We investigate this possibility by examining the global root-mean-square of the surface pressure tendency in the forecasts for the hybrid and the benchmark models (Fig. 3.7). We assume that the value computed for ERA5 (red curve), which is about 0.4 hPa/h, provides a realistic estimate of the

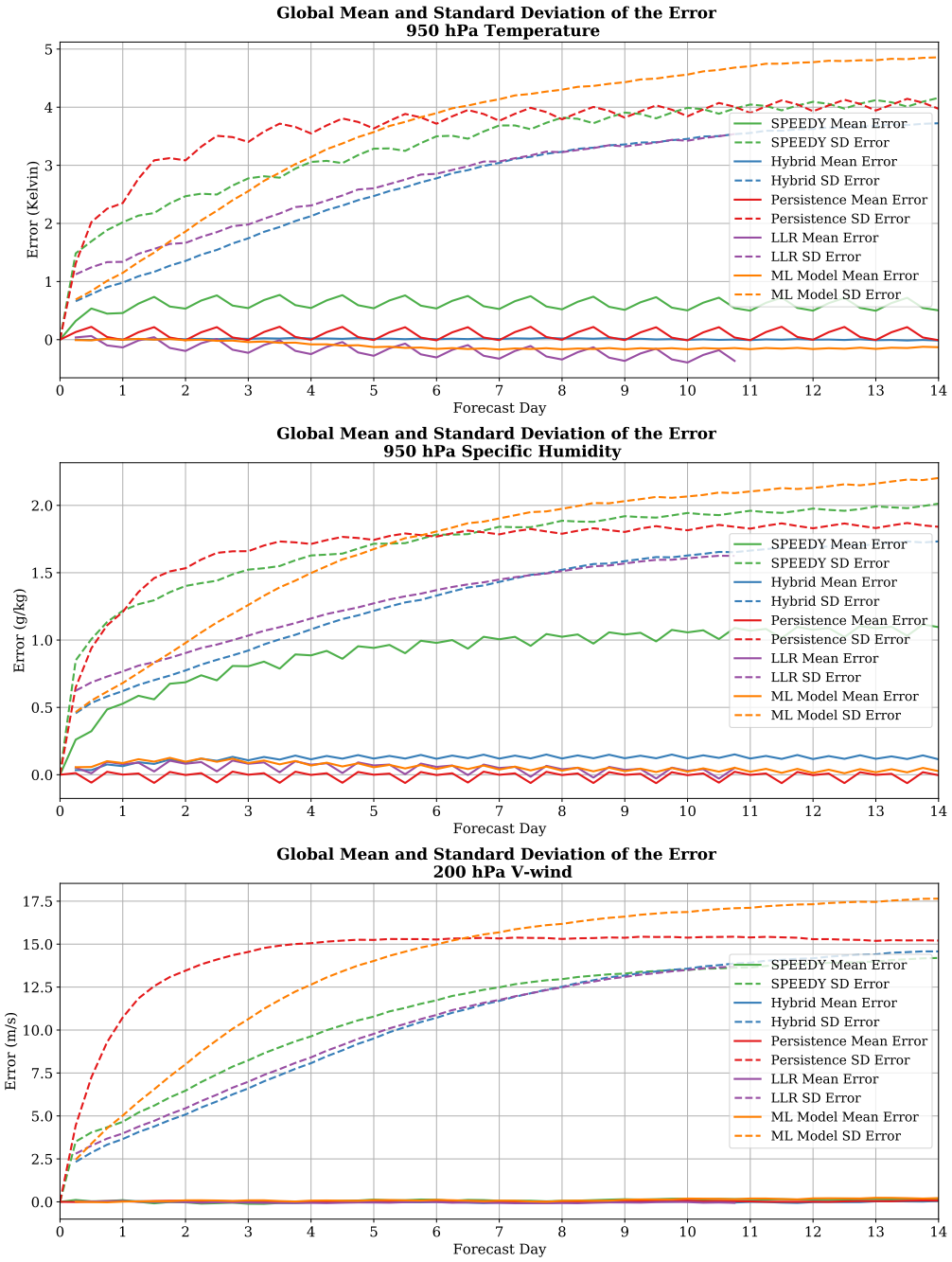

Figure 3.6: The time evolution of the (dashed) standard deviation and (solid) mean of the forecast errors. Each color indicates forecasts by a particular model: (blue) hybrid model, (green) SPEEDY, (purple) SPEEDY-LLR model, (orange) ML model, and (red) persistence. Results are not shown for SPEEDY-LLR beyond day 11, at which time some of the the forecasts for that model fail. Reprinted with permission from Arcomano et al. (2022).

global root-mean-square of surface pressure tendency in the atmosphere.

As can be expected from a numerical model started from an uninitialized initial condition, the initial tendency for SPEEDY (about 1 hPa/h) is higher than desired. As forecast time increases, the the magnitude of the mean tendency drops, first rapidly, and then at a decreasing rate until it settles below the natural level, at about 0.28 hPa/h. The latter behavior suggests that the diffusion built into the model to combat imbalances over-smooths the temporal variability of the forecasts beyond day 1. While the magnitude of the mean tendency for the hybrid forecasts (about 0.38 hPa/h) is initially slightly smaller than the natural value, and further decreases in the first 72-84 h (to about 0.36 hPa/h), it is closer to the natural value than those for the benchmark forecasts. The SPEEDY-LLR is less effective than the hybrid model in eliminating the initial transient and it also produces an average tendency at the later forecast times (about 0.30 hPa/h) that is further below the natural level. The ML-only model behaves similarly to the hybrid model for the first two forecast days, but the saturation value is clearly lower (about 0.33 hPa/h) than for the hybrid model.

3.3.6 Sensitivity to Training Length

To test the sensitivity of the performance and stability of the hybrid model to the training length, we carry out a series of experiments with the same hyperparameters as before, but for shorter training periods. In particular, we train the model on 2 years, 5 years, or 10 years of reanalysis data, with the training always ending at 2300 UTC, June 26, 2011, as for the original forecast experiments. (We recall that the length of the training for the original experiments is 20.5 years.) The results of these experiments for the usual 100 21-day forecast cases for select variables are summarized in Fig. 3.8.

While training the hybrid model for only 2 years already significantly improves the forecast performance for the near-surface temperature and specific humidity compared to that of SPEEDY, extending the training length further improves the forecasts. The hybrid model trained for 2 years does not improve the meridional wind component in the upper troposphere, and actually degrades the forecasts beyond 3 days. A longer training makes the hybrid model perform better initially than

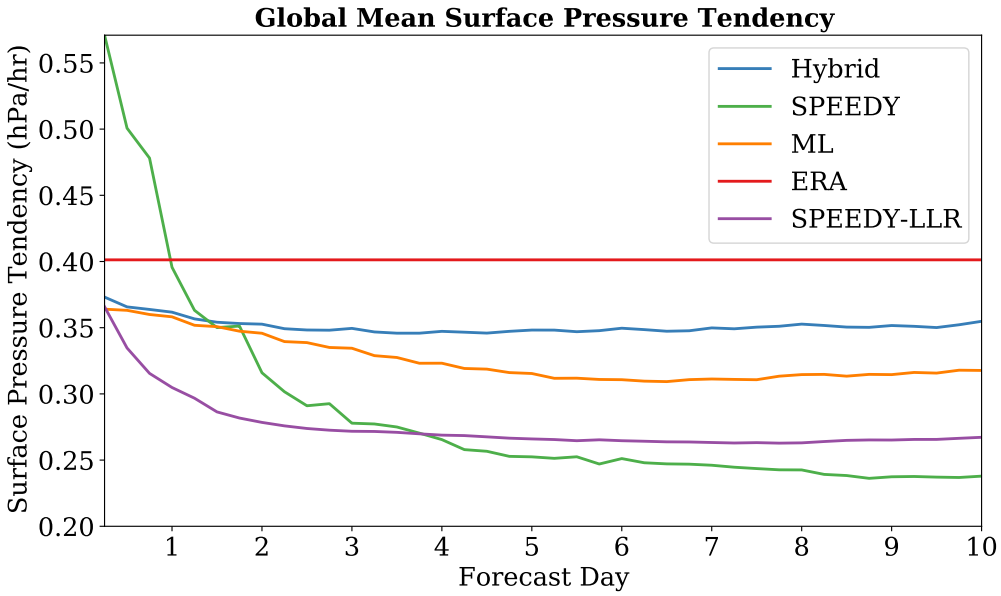

Figure 3.7: Atmospheric balance in the model forecasts. Shown is the global root-mean-square of the approximate surface pressure tendency computed by finite-differences based on 6-hourly data for the (blue) hybrid model, (green) SPEEDY, (orange) ML-only model, and (purple) SPEEDY-LLR model. The (red) value computed for 2011-2012 based on the ERA5 reanalyses is also shown for reference. Reprinted with permission from Arcomano et al. (2022).

SPEEDY. The length of the superior performance of the hybrid model becomes longer as the length of the training period increases. The results shown in Fig. 3.8 also suggest that a further modest improvements of the forecast performance could be achieved by using a training period even longer than 20.5 years.

3.4 Climate Simulation Experiment

To evaluate the long term stability of the hybrid model and its ability to simulate the climate, we compute an 11 year long free run with the model. For this simulation experiment, the hybrid model is trained on ERA5 reanalyses for the 19-year period from January 1, 1981 to December 27, 1999. The simulation starts from the ERA5 reanalysis valid at 0000 UTC, January 1, 2000. To suppress the effects of initial transients and the initial condition on the model diagnostics, we discard the data from the first year of the simulations before computing the diagnostics. To compare the performance of the hybrid model and SPEEDY in simulating the climate, we assume that the two

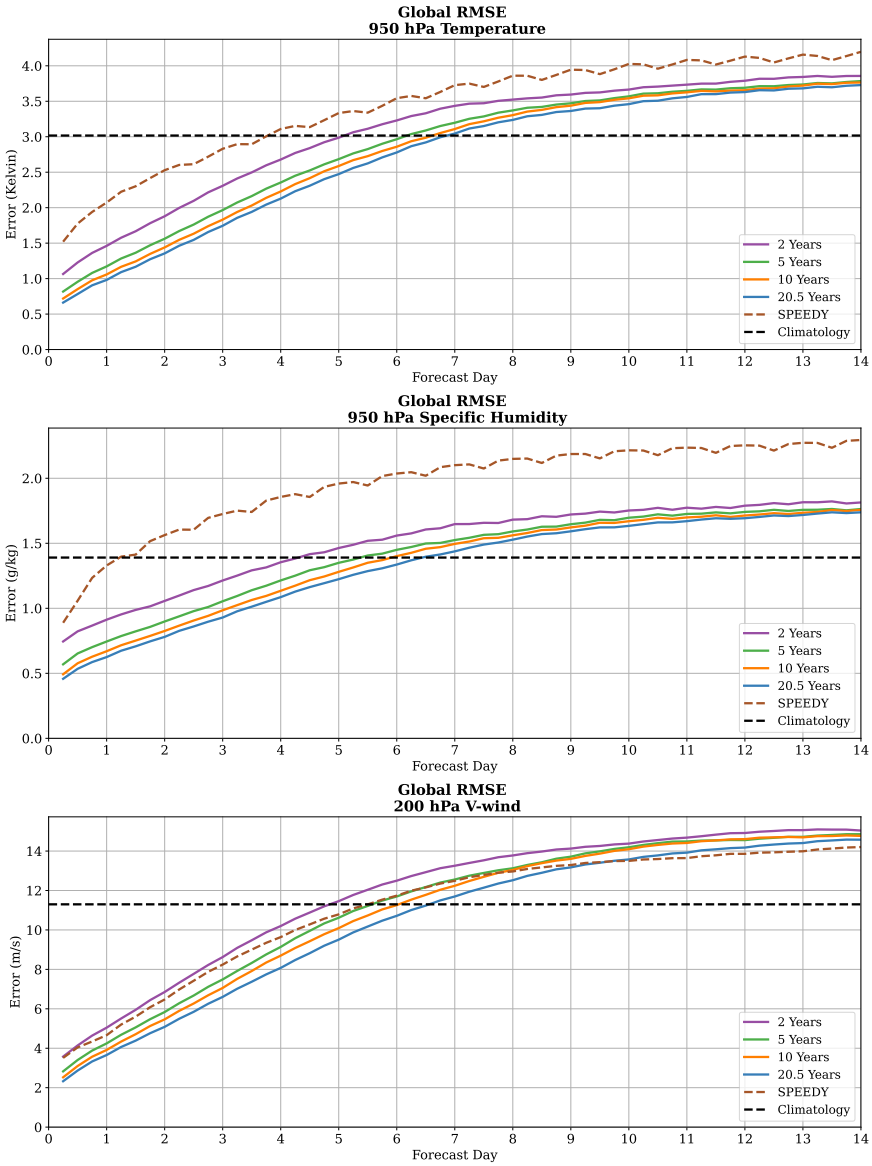

Figure 3.8: Time evolution of the global root-mean-square forecast error for different lengths of the training of the hybrid model. Results are shown for a (purple) 2 years, (green) 5 years, (red) 10 years, and (blue) 20.5 years training period. For reference, the forecast errors are also shown for (brown dashes) SPEEDY and (black dashes) climatology. Reprinted with permission from Arcomano et al. (2022).

simulations attempt to simulate the climate of the 10-year period from 2001-2010 as represented by ERA5.

3.4.1 Zonal Mean Biases

Figures 3.9 and 3.10 show the zonal mean biases of the simulations by SPEEDY (left panels) and the hybrid (right panels) for the boreal winter (December, January, and February) and boreal summer (June, July, and August), respectively. These figures can be used, not only to compare the quality of the two simulations, but also to assess the average magnitude of the corrections made by the ML component of the hybrid model. In particular, the difference between a left panel and the corresponding right panel is the zonal mean of the ML correction for a particular state variable.

The top left panels show that SPEEDY has a large upper tropospheric warm bias for the tropical regions, during both the boreal winter and summer. In both polar regions SPEEDY has a cold bias for the upper troposphere and stratosphere during the boreal winter and a warm (cold) bias in the southern (northern) polar region during the boreal summer. The magnitude of the bias is not surprising given the coarse resolution and simplified parameterizations used in SPEEDY (Molteni 2003). The top right panels show that the hybrid model greatly reduces, but does not completely eliminate, these biases when the model is cycled over a long period of time. The bias reduction is particularly notable in the the tropics and the midlatitudes. The largest remaining biases are in the polar regions. The hybrid model reduces the zonal component of the wind bias, especially in the stratosphere and upper troposphere, and in the lower troposphere in the SH midlatitudes in the boreal summer. The only exception is the introduction of a positive zonal component of the wind bias in the stratosphere in the tropics. The hybrid model also greatly reduces the large positive humidity bias of SPEEDY with maxima in the tropics.

Figure 3.11 shows the mean surface pressure biases for the simulations by SPEEDY (left panels) and hybrid model (right panels) for the boreal winter (top row) and boreal summer (bottom row). The mottled short scale patterning seen in the two left panels of the figure are due to the spectrally truncated topography of SPEEDY, which is much smoother than the topography determining the interpolated ERA5 reanalyses used for the evaluation of the simulations, and for the training of the hybrid model. In combination with the artifacts caused by the spectral truncation in SPEEDY, the large local differences in the mountainous regions lead to substantial surface pressure biases in

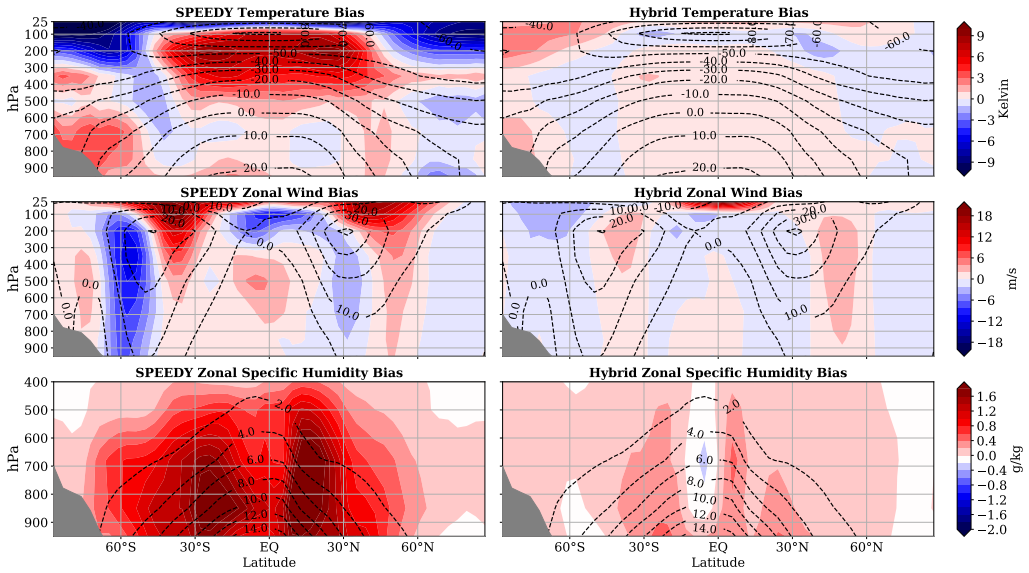

Figure 3.9: Comparison of the zonal mean biases of the SPEEDY and hybrid simulation simulations for the boreal winter (December, January, February). Results are shown for (top) the temperature (middle), zonal wind, and (bottom) specific humidity for (left) SPEEDY and (right) the hybrid model. Reprinted with permission from Arcomano et al. (2022).

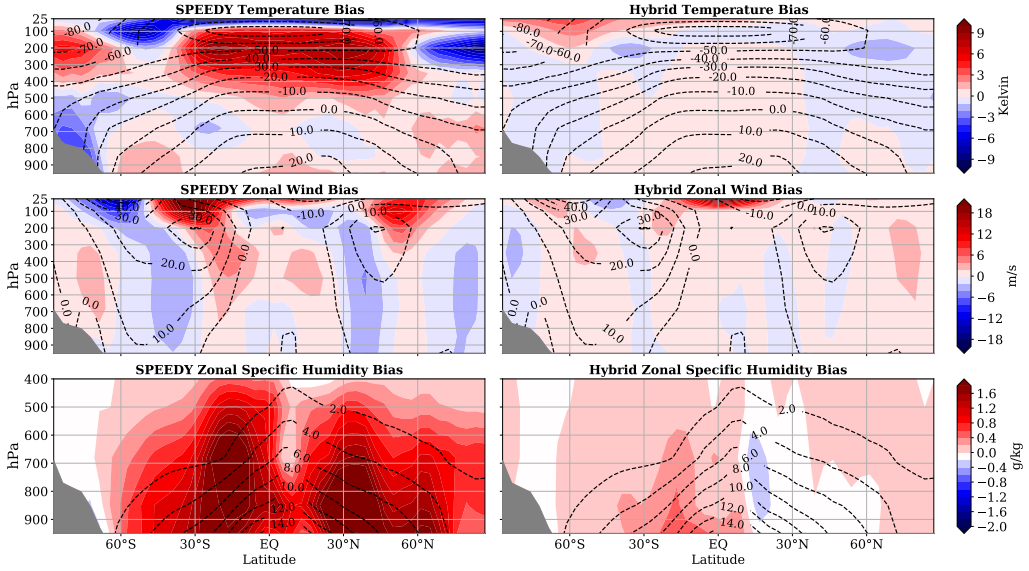

Figure 3.10: Same as Fig. 3.9, except for the boreal summer (June, July, August). Reprinted with permission from Arcomano et al. (2022).

the SPEEDY simulations. The hybrid model corrects the large local biases, but still has smaller magnitude large scale biases. The wave-number-two structure of the large-scale hybrid model bias

in the NH suggests that these biases are related to the low resolution representation of the topography and the land-sea contrasts in the numerical model. The remaining biases are also relatively large in the polar regions, especially in the boreal summer. We speculate that the bias of the hybrid model in the polar regions might be related to our particular strategy to do the localization on a cylindric (Mercator) map projection. On the other hand, the bias is not concentrated at the poles for the variables shown in Figures 9 and 10.

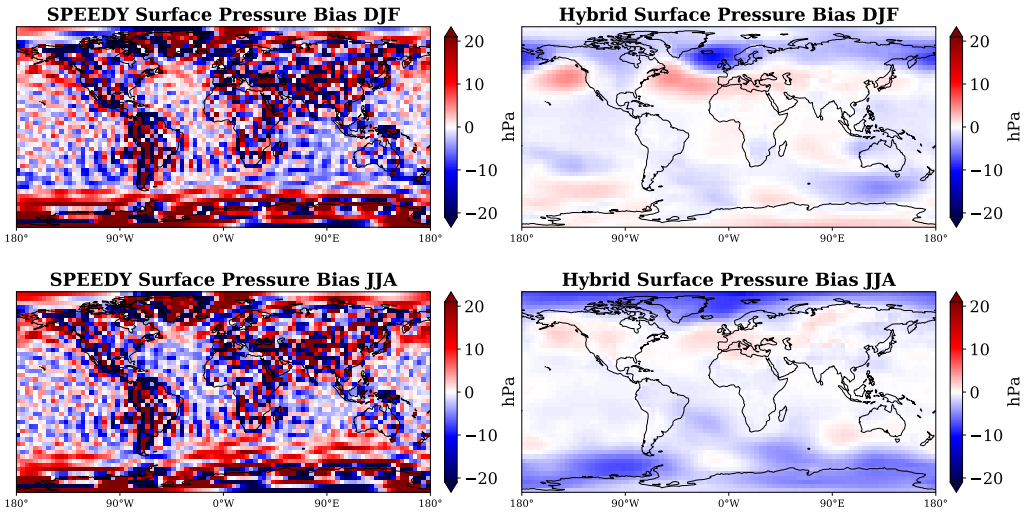

Figure 3.11: The mean surface pressure bias in the SPEEDY and hybrid climate simulations. Shown is the bias for (top) the boreal winter (December, Januar, February) and (bottom) boreal summer (June, July, August) for (left) SPEEDY and (right) the hybrid model. Reprinted with permission from Arcomano et al. (2022).

3.4.2 Temporal variability

To investigate the temporal variability of the atmosphere in the SPEEDY and hybrid climate simulations, we examine the temporal dependence of the 950 hPa temperature at the four model grid points that fall in the Sahara Desert. The top two panels of Fig. 3.12 show the power spectra of the temporal variability for the two models. These power spectra are computed by applying a Hamming filter first, and then a discrete Fourier transform to the 10 years of 6-hourly simulation

data, and finally computing the square of the absolute value of the Fourier coefficients. The results show that both simulations correctly capture the variability at time scales longer than about a week. At the shorter time scales, however, SPEEDY increasingly underestimates the variability. The ML correction greatly reduces, but does not completely eliminate, this problem: the hybrid model underestimates the variability at the scales between one week and one day only slightly, and reduces the underestimation by SPEEDY at the even shorter scales. Most importantly, unlike SPEEDY, the hybrid model has a strong diurnal cycle. It should be noted that an earlier version of the hybrid model, which did not include the incoming solar radiation at the top of the atmosphere as an input to the reservoir, lost the diurnal cycle at around the end of year 4. This motivated us to add the incoming solar radiation as an input parameter, even though it had no significant effect on the forecast accuracy. We find it a noteworthy, nontrivial result that the earlier version of the hybrid model was able to learn the diurnal cycle strictly from the training data.

The fact that a simulation correctly captures the variability at a number of frequencies does not guarantee that the phases of the temporal changes (e.g. the timing of the seasons) are also correct. To exclude the possibility of such a flaw of the simulations, we plot (bottom panel of Fig. 3.12) the time series of the average 950 hPa temperature for the same four Saharan grid points for the last full year of the simulations. The points along these curves should fall within two standard deviations from the mean for the given date and time (the interval marked by gray shading) with a 95% observed frequency. Based on the full ten years of data, the observed frequency is 88.2% for SPEEDY and 98.0% for the hybrid model.

3.5 Conclusions

In this paper, we described results from the first implementation of the hybrid modeling approach CHyPP of Wikner et al. (2020) on a realistic atmospheric model. We used a low-resolution AGCM based on the full set of primitive equations, along with ERA5 reanalysis data for training and verification, to demonstrate the potentials of CHyPP for both NWP and climate modeling. The spatio-temporal structure of the improvements of the forecasts and simulations suggests that

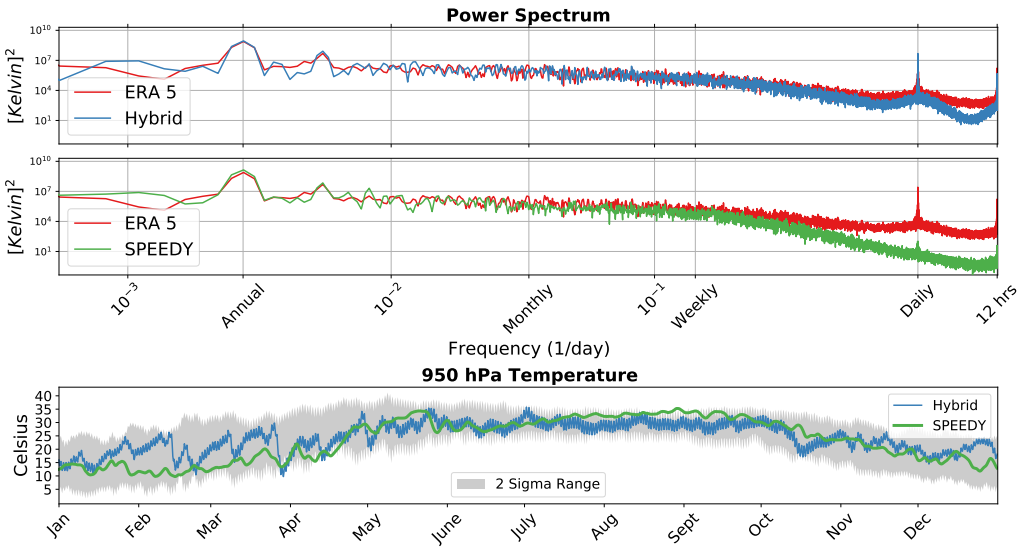

Figure 3.12: Temporal variability of the 950 hPa temperature in the Sahara Desert for the ten years of simulations. Shown are the power spectra for (top) the hybrid model and ERA5 and (middle) SPEEDY and ERA5. The bottom panel shows the time series of simulated temperatures for the last full year of the simulations. The gray shading represents the range of plus/minus two standard deviations from the mean in the ERA5 reanalyses for 2001-2010. Reprinted with permission from Arcomano et al. (2022).

the ML component of the model primarily corrects for errors caused by the limitations of the parameterization schemes of the AGCM. While state-of-the-art numerical models have much higher resolutions and more advanced parameterization schemes than SPEEDY, the weather forecasts and climate simulations they provide still have substantial biases. We expect the hybrid approach to effectively reduce these biases.

Because the ML component of the hybrid model is based on RC, training the model is computationally highly efficient. Specifically, the training described in this paper requires only 30 minutes wall-clock time using 1,152 Intel Xeon E5-2670 v2 processors on a supercomputer that is much less powerful than those at the operational NWP centers. Using the same computational resources, preparing a 21-day forecast takes about 52 seconds, while carrying out a one-year simulation takes about 15 minutes. These numbers are only 25% higher than those for SPEEDY, and the extra time is mainly due to the overhead associated with the frequent restart of SPEEDY.

Due to the parallel nature of the computational algorithm, we expect it to scale well for higher model resolutions and larger number of processors. A modification of the current implementation

of our method that might be helpful for scaling is vertical localization. By "vertical localization" we mean the use of local domains that, as well as being limited in horizontal extent as shown in Fig. 3.1, are also of limited height and are stacked vertically with overlap from ground-level to the top of the atmosphere. Though we do not use vertical localization in this article, we plan to test it soon for potential improvements with SPEEDY.

The ideal size of a local domain still needs to be determined through additional experimentation, both for SPEEDY and for higher-resolution models. Thus, it is hard to make a precise quantitative projection for scaling, but here is a comparison that indicates feasibility for operational models. The current computer of ECMWF has 129,960 processors (about 100 times more than what we used), and their operational model has 6.5×10^6 horizontal grid points (about 180 times more than SPEEDY) (ecm 2020). If the local regions for the ECMWF model would be defined by four horizontal and all vertical grid points, as in our paper, each processor would have to handle less than twice as many local regions at ECMWF than in our model. Also, there is no obvious reason to believe that the computational overhead of the hybrid model would be substantially higher than the 25% we found for SPEEDY. The high computational efficiency of the approach would allow for a large number of experiments to find the optimal configuration of a future operational hybrid model. Developing an efficient systematic approach to find a near optimal combination of the hyperparameters, nevertheless, would be highly desirable and is one of the subjects of our ongoing research efforts. An unknown factor that could have a very favorable impact on future scaling considerations is the ongoing rapid technological developments of alternative, fast, cheap physical implementations of reservoir computing, e.g., implementations based on photonics or on Field Programmable Gate Arrays.

We emphasize that while the ML component of the hybrid model is highly efficient in correcting the biases of the forecasts and simulations prepared by the host model, it is not a ML-based postprocessing technique. While a technique of the latter type corrects the numerical-model-based forecasts of a specific forecast variable or phenomenon (e.g., Rasp and Lerch 2018; Chapman et al. 2019; Kim et al. 2021) without interacting with the numerical model, the ML component of the hy-

brid model makes frequent periodic interactive corrections to the numerical model solution. Hence, it also greatly improves the representation of the spatiotemporal variability of the atmospheric state by the model.

We expect that the performance of the hybrid model can be further improved by investigating the relationship between the parameters of the ML model and the representation of basic atmospheric processes. Such an investigation could lead to further improvements of the model, similar to the way studies of the interactions between numerics and dynamics (e.g., Arakawa and Lamb 1977) led to much improved physic-based numerical models. For instance, one potentially important fundamental question is the optimal relationship between the size of the local domains, the overlap between the local domains in the input of the reservoir, and the length of the time step Δt. The fact that the ML component is more effective in correcting localized errors than errors at the larger scales in the current version of our hybrid model may be partly the result of using local domains and an overlap that are less than optimal for the selected time step. In our experiments, the size of the overlap was primarily dictated by the structure of our code and the available computer resources, but larger local domains and a larger overlap could be used in the future.

An intriguing possibility is to use the hybrid model for data assimilation in addition to forecasting, as data assimilation could greatly benefit from the higher accuracy and smaller biases of the short term hybrid forecasts used as background. Furthermore, integrating ML and data assimilation may allow in the future to do online training of the ML component of the hybrid model on real-time observations rather than canned reanalyses data. The availability of such training procedure would make it possible to extend the hybrid modeling approach to numerical models for which high-quality reanalysis data are not available (e.g., an AGCM that also includes a sophisticated model of the upper atmosphere well beyond the lower stratosphere). It could also allow the ML component of the model to adjust to variability and changes of the climate. We have made a first step toward this ambitious goal, in which we iteratively use the hybrid model to prepare an updated set of analyses, which is then used to train the next iteration of the hybrid model (Wikner et al. 2021). Our plan is to test this approach with the hybrid model of the current paper.

4. COUPLING THE ATMOSPHERIC HYBRID MODEL WITH A MACHINE LEARNING OCEAN MODEL

4.1 Introduction

Over the last several decades there has been significant advancement in global climate models (GCMs) with the development of coupled GCMs that can numerically simulate the interactions between the various Earth system components (e.g. atmosphere, ocean, cryosphere). Earth System Models (ESMs) add the simulation of chemical and biological processes and their interactions with the Earth system. GCMs and ESMs are the main tools for providing insight on possible future climates caused by anthropogenic activities (e.g. Lynch 2008). Simulations of hypothetical, future climates from GCMs are crucial for making recommendations on future climate risk and allow for mitigation to reduce impacts of climate change (e.g. Arias et al. 2021). GCMs have steadily improved at replicating the present and past climate due to advances in high performance computing allowing GCMs to be run at higher spatial resolutions (e.g. Ma et al. 2015; Caldwell et al. 2021). At the same time advancements in parameterization schemes have improved effects of subgrid processes not explicitly resolved by the dynamical core of climate models (e.g. cloud microphysics) (Lynch 2008). Even with these advancements, state-of-the-art climate models continue to have large systematic biases especially for atmospheric variables heavily influenced by parameterization schemes (e.g. precipitation and clouds), sea surface temperatures in the ocean models, and sea ice (e.g. Danabasoglu et al. 2020; Golaz et al. 2019; Zhang et al. 2019).

Recently, the incorporation of machine learning (ML) into the numerical weather prediction (NWP) process has been shown to improve short-term weather forecasts and reduce the biases of atmospheric general circulation models (AGCMs) when compared to the past and present climates. These hybrid models that combine machine learning with a traditional numerical model offer a pathway to reduce forecast error and model bias by using ML to make *frequent periodic interactive correction* to the evolving model solution. Watt-Meyer et al. (2021) used deep learning

and random forest-based ML architectures to learn the nudging tendencies for select variables in a coarse AGCM using reanalysis. Clark et al. (2022) and Bretherton et al. (2022) expanded this work by using ML to learn the nudging produced by cloud resolving (3km) simulations in stationary and nonstationary climates. The hybrid modeling approach described in Section 3 of this dissertation that combines a coarse AGCM (SPEEDY) with a parallel, reservoir-computing-based (RC) algorithm. As already shown in Section 3, the hybrid model improved short to medium range weather forecasts and greatly reduced the biases of SPEEDY during an 11-year free run.

Another promising approach to improving weather forecast and climate models is to use purely data-driven models for a time-series prediction of various Earth System components. Recently, machine learning only models for weather prediction has become an area of active research (e.g. Weyn et al. 2019, 2020; Scher and Messori 2019; Rasp and Thuerey 2021). Data-driven models offer a major advantage to numerical-based models, because once trained, ML models can make predictions significantly faster than traditional numerical models (e.g. Pathak et al. 2022). For climate change projections, having an ensemble of models can help quantify uncertainty and deliver probabilistic forecasts that are crucial for policy makers (Arias et al. 2021). GCMs are run for decades or centuries and typically need to be run on high performance computing cluster. In contrast, once trained, ML models can be run on a small number of CPUs or GPUs. Taking advantage of this computational efficiency, Weyn et al. (2021); Scher and Messori (2021) were able to create a large ensemble system with more ensemble members than what would have been computationally feasible using numerical models, allowing for the possibility to improve subseasonal to seasonal forecasts.

The ocean is one of the most important components of the Earth system. The sea surface temperature (SST) is particularly important, because it is the interface between the ocean and atmosphere, playing key roles in the Earth energy budget and water cycle. Ocean modeling also poses unique challenges compared to the atmosphere due to processes such as land-ocean interface and sea ice. The oceanicc components of modern GCMs and ESMs are sophisticated ocean models that are computationally expensive and still suffer large SST biases when compared to past and

present climate (e.g. Golaz et al. 2019; Capotondi et al. 2020; Zhu et al. 2020). Data-driven models have been developed to predict sea surface temperatures (SSTs) and other various ocean phenomena (e.g. El Nino Southern Oscillation) with the hopes to improve predictability and reduce biases. Deep learning methods have been used to extend ENSO predictability and try to break the ”spring predictability barrier”, a challange current numerical models struggle with (e.g. Ham et al. 2019). There has also be development using deep learning to predict local and global SSTs (e.g. Xiao et al. 2019; Sarkar et al. 2020; Taylor and Feng 2022).

Unlike the authors of aforementioned publications, we propose a global ML-ocean model that utilizes the parallel, reservoir computing-based (RC) approach of Pathak et al. (2018a) and Arcomano et al. (2020) rather than deep learning to predict SST from past oceanic and atmospheric states. Walleshauser and Bollt (2022) in which the authors used the same parallel, reservoir computing-based algorithm as us to build a stand-alone ML model for the prediction of the global SST dynamics. In our model, the ML model of the SST dynamics is coupled to the hybrid model described in Section 3.

The work presented here can be considered a first step towards creating a fully coupled GCM that utilizes machine learning to couple and predict various Earth system components (e.g. atmosphere, ocean, cryosphere).We will show that the coupled model is stable during a 70-year free run and it is able to reproduce the present climate with significantly less bias than the AGCM (SPEEDY). The coupled model can simulate long-term climate variability for both the ocean and atmosphere (e.g. El Niño–Southern Oscillation).

In what follows, we first describe the coupled model (Section 4.2). Then, we discuss the results of the 70-year climate free run (Section 4.3). Finally, we summarize our key findings and draw our conclusions (Section 4.4).

4.2 The Coupled Model

The coupled model has two components: a hybrid atmospheric model similar to the one described in Section 3 and, a machine learning ocean model that uses the parallel RC algorithm de-

scribed in Section 2.2 and 3.2.3 to predict the SST. Like numerical physics-based coupled GCMs, the two components of our coupled model evolves at different timesteps (Δt_{atmo} and Δt_{ocean}): $\Delta t_{atmo} = 6$ hours and $\Delta t_{ocean} = 7$ days. In the prediction phase the hybrid atmospheric model is dynamically coupled to the ML-ocean model. Each component of the coupled model evolves at its respective timestep and exchange information every 7 days. A flowchart showing the exchange of information in the prediction phase is provided in Figure 4.2.

4.2.1 Machine Learning Only Ocean Model

4.2.1.1 The Local State Vectors

In our implementation of the parallel RC-based algorithm for predicting the SST, each local state vector represents the SST state in a two-dimensional local domain that has the shape of a rectangular box with dimensions of $7.5^\circ \times 7.5^\circ$ (2×2 horizontal grid points). In what follows, we describe the computations carried out in parallel for each of the L local domains to evolve the ML-ocean model state from time t to $t + \Delta t_{ocean}$.

Let $\mathbf{v}(t)$ be the local state vector for an arbitrary local domain at time t. The dimension of this state vector is 4 (SST value at each grid point). The state vector is standardized to have a mean of 0 and a standard deviation of 1 before forming $\mathbf{v}(t)$. The standardization is done by using ERA5 reanalysis data (Hersbach et al. 2020) for the computation of the climatological mean and standard deviation for each local domain.

The local input $\mathbf{u}(t)$ is an m-dimensional *extended local state vector*, composed of the components of the local state vector $\mathbf{v}(t)$, plus the additional components from the neighboring local domains, plus the rolling mean of the two components of the wind vector, temperature, and specific humidity for Δt_{ocean} at σ=0.95, plus the rolling mean of the natural logarithm of surface pressure and the prescribed incoming solar radiation for Δt_{ocean} at the top of the atmosphere for the extended local domain. For all of the local domains, $m = 16 \times (4 + 1 + 1 + 1)$, except at the local domains adjacent to the poles where $m = 12 \times (4 + 1 + 1 + 1)$.

4.2.1.2 Training

The reservoir evolves by the discrete time mapping

$$\mathbf{r}(t+\Delta t) = \tanh\left[\mathbf{A}\mathbf{r}(t) + \mathbf{B}\mathbf{u}(t)\right], \tag{4.1}$$

where $\mathbf{r}(t)$ is the *reservoir state*, and $\mathbf{u}(t)$ is the input vector. In the training phase $\mathbf{u}(t)$ is ERA5 reanalysis data (Hersbach et al. 2020). The matrix $\mathbf{A}$ is a sparse *weighted adjacency matrix* that represents a low-degree, directed, random graph Gilbert (1959). Each nonzero element of $\mathbf{A}$ is randomly chosen from a zero-mean uniform distribution, with the number of nonzero elements being prescribed by a specified sparsity. To ensure that the reservoir has the *echo state property* (Jaeger 2001), the values of the nonzero elements are scaled such that the largest eigenvalue of $\mathbf{A}$ is a prescribed value that is between 0 and 1. The matrix-vector product $\mathbf{B}\mathbf{u}(t)$ is called the *input layer* of RC. In our model, $\mathbf{B}$ is a sparse random matrix with an equal number of nonzero entries in each row. A flowchart of the different parts of the RC are shown in Fig. 4.1.

Similarly to Section 2, the ML ocean model prediction is obtained by

$$\mathbf{v}(t+\Delta t_{ocean}) = \mathbf{W}[\mathbf{r}(t+\Delta t_{ocean})], \tag{4.2}$$

where $\mathbf{W}$ is the *output layer*. The output layer $\mathbf{W}$ is a matrix of parameters that are determined by training on a time series of SST reanalyses $\mathbf{v}^a(k\Delta t_{ocean})$, $k = -K - K_t, -K - K_t + 1, ..., -1$. The the reservoir states $\mathbf{r}(k\Delta t_{ocean})$ for $k = -K - K_t, -K - K_t + 1, ..., -1$ are obtained by substituting $\mathbf{v}^a(k\Delta t_{ocean})$ for $\mathbf{u}(k\Delta t_{ocean})$ in Equation 2.1. The $\mathbf{W}$ can be computed by minimizing the cost-function

$$J(\mathbf{W}) = \sum_{k=-K+1}^{0} \|\mathbf{v}(k\Delta t_{ocean}, \mathbf{W}) - \mathbf{v}^a(k\Delta t_{ocean})\|^2 + \beta\|\mathbf{W}\|^2. \tag{4.3}$$

The first term on the right hand-side of the equation minimizes the difference between a one-time-step prediction from the reservoir and the reanalysis valid at the same time. The second term of the cost function, in which $\|\cdot\|^2$ denotes the sum of the squares of the entries of a matrix (the Frobenius

norm), is a regularization term to prevent overfitting (Tikhonov and Arsenin 1977).

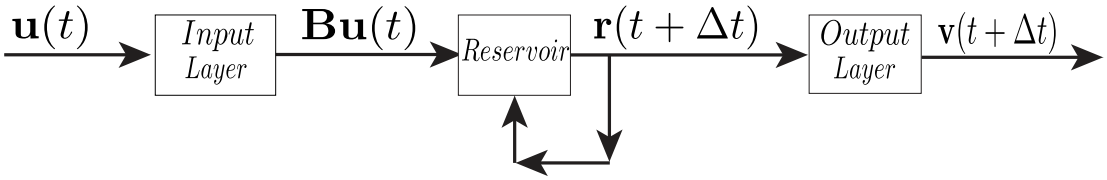

Figure 4.1: A flow chart of our implementation of reservoir computing. The notation is defined in Section 4.2.1. The flow chart highlights the three main components of the algorithm: the input layer, the reservoir, and the output layer.

4.2.1.3 Sea ice and Coastlines

The ML-ocean model has the capability to simulate sea ice dynamics in a simplistic manner. First, any local domain near the poles that are permanent sea ice during the period of 1981-2007 are ignored and are assumed to be sea ice during the entirety of the predictions. We determine if sea ice is present by assuming any SST value less than -1C indicates sea ice. In the transitional regions between permanent sea ice and water that were completely sea ice free during the training period, we treat sea ice as part of the ocean. During prediction the reservoir may predict values less than -1$^{\circ}C$, but to prevent instability any values below -1C are overwritten to be precisely -1$^{\circ}C$ at the end of each ocean time step.

Any local domain that contains only grid points over land is ignored. For reservoirs with local domains that contain grid points over both the ocean and land, the grid points over land are ignored. Thus, the local reservoirs make predictions only for oceanic grid points. Atmospheric input variables are provided for all grid points in the local region, including those over land. If a grid point in the extended local state vector $\mathbf{u}(t)$ is over land, we assign a constant land mask value to the corresponding entry of $\mathbf{u}(t)$. This provides the reservoir with spatial information whether a grid point is over the ocean or land.

4.2.2 Hybrid Atmospheric Model

The atmospheric hybrid model is an updated version of the model described in detail in Section 3. The updates include the addition of the SST field to the local state vector to allow for the dynamically coupling of the hybrid atmospheric model to the ML-ocean model in the prediction phase. In the prediction phase, the SST field from the ML-ocean model provides the boundary condition for SPEEDY. A flowchart of this is process shown in Fig. 4.2. We also add 6-hourly total precipitation (TP) as a prognostic variable to the hybrid atmospheric model. Because SPEEDY does not provide a short-term prediction of TP, the ML component of the hybrid model is solely responsible for the prediction of TP. Because TP is highly skewed towards zero with the occasional nonzero value, we apply a log-transformation to TP: $\log(1 + \text{tp}/\epsilon)$, where ϵ is a tunable *hyperparameter*.

We also found a coding error related to the addition of noise to the training data. In Sections 2 and 3 we reported that a small-magnitude, zero-mean, normally distributed random noise vector ε^g, uncorrelated in time and uncorrelated between components of the noise vector was added to the global analyses to create a sequence of perturbed global analyses $\mathbf{v}^{ga}(k\Delta t) + \varepsilon^g(k\Delta t)$, $k = -K-K_t, -K-K_t+1, ..., -1$. We discovered that because of the coding error, the noise vector was multiplied element by element with the components of the global analyses instead of being added to them. We found (by accident) that this method of perturbing $\mathbf{v}^{ga}$ helps stabilize the hybrid model more than adding the noise vector to the training data. In fact, for the hyperparameters reported in Section 3, we were unable to stabilize the hybrid model by the addition of noise.

4.2.3 Synchronization and Prediction

Synchronization is achieved by evolving the reservoir equation using reanalyses for both the atmosphere and ocean components. In the prediction phase, all of the local predictions for the atmosphere and ocean are evolved time step by time step to produce a global coupled model forecast. Because $\Delta t_{atmo} \ll \Delta t_{ocean}$, the hybrid atmospheric model takes several timesteps before the ocean model takes another step. The two components of the coupled model exchange information only once per oceanic time step. A flowchart of the coupled model for the prediction phase is shown in

Figure 4.2.

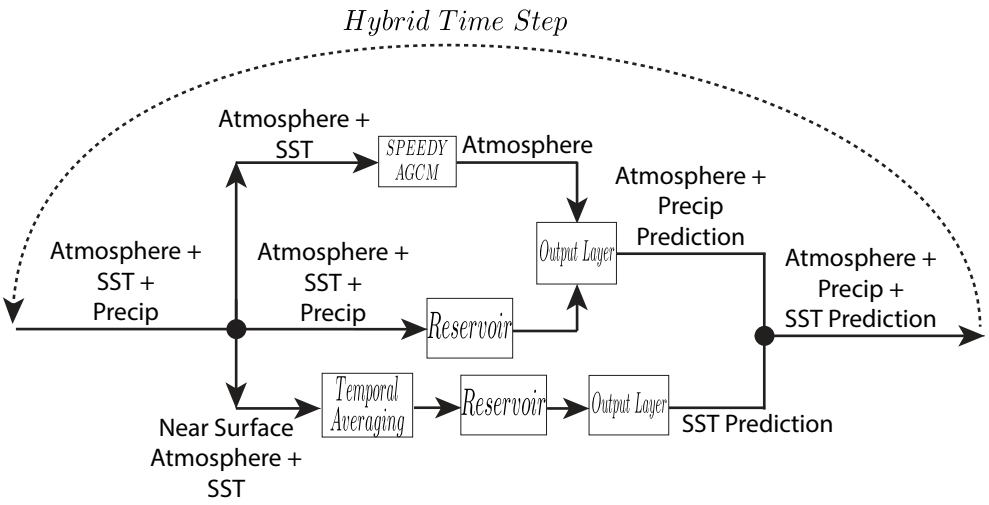

Figure 4.2: A flow chart of the hybrid atmospheric model coupled with a ML-based ocean model. The SST from the ML-only ocean model is used as boundary condition for SPEEDY during the climate free run.

4.2.4 Implementation with ERA5 Reanalysis Data

ERA5 data is used for the training and synchronization of the hybrid atmospheric component and the ML-only ocean component of the model. As in Sections 2 and 3, we first regrid the atmospheric fields to the horizontal grid of SPEEDY by a 2-dimensional quadratic B-spline interpolation. If a variable is 3-dimensional, we calculate the σ value for each grid point first and then interpolate the fields of the atmsopheric variables to the σ-levels of SPEEDY vertically, using a 1-dimensional cubic B-spline. For the SST fields, we use the Climate Data Operators (CDO) software (Schulzweida 2022) to interpolate the ERA5 SST reanalysis to the SPEEDY horizontal grid using a 2-dimensional quadratic B-spline with a land mask.

4.2.5 Selection of the Hyperparameters

As in Sections 2 and 3, there are a number of *hyperparameters* that require tuning to produce stable and realistic results. To reduce the overall number of hyperparameters needed to be tuned, we chose the same values as in Section 3 for the hybrid atmospheric model (see Section 3.2.6 for more detail). The addition of 6-hourly precipitation in the hybrid atmospheric model, however, introduces the new hyperparameter ϵ that requires tuning. Using the values from Rasp and Thuerey (2021) and Pathak et al. (2022) as starting points, we run experiments with $\epsilon = 10^{-3}$, $\epsilon = 10^{-4}$, and $\epsilon = 10^{-5}$ and evaluate each by monitoring the annual precipitation bias and extreme precipitation rates. Using a value of $\epsilon = 10^{-5}$ over-predicted extreme precipitation rates and has a significant wet bias. For $\epsilon = 10^{-4}$ there is still a substantial wet bias, but the extreme rainfall rates match ERA5 well. For $\epsilon = 10^{-3}$ the model tends to under-predict extreme rainfall rates, but it does not have a significant wet or dry bias. We chose $\epsilon = 10^{-3}$, because it offers the best balance between reducing biases and predicting extreme rainfall.

We found that using the same values of the hyperparameters κ and α for the ML-only ocean model as for the hybrid atmospheric model ($\kappa = 6$ and $\alpha = 0.5$) performed well. We varied ρ from 0.3 to 0.9 in increments of 0.1 and found that $\rho = 0.9$ was needed to achieve stability. The ML-only ocean model was found to not be sensitive to β_{res}, and a value of $\beta_{res} = 10^{-3}$ was chosen. After performing several experiments, we found that the smallest value of ε for which the model was stable was 0.1. $D_r = 4,000$ was chosen, because there was little improvement found in performance using larger values. We found that the most important hyperparameter for both stability and performance was Δt_{ocean}. We tested $\Delta t_{ocean} = 1$ day, $\Delta t_{ocean} = 7$ days, and $\Delta t_{ocean} = 14$ days. Using a $\Delta t_{ocean} = 7$ days and $\Delta t_{ocean} = 14$ days both produced stable coupled predictions, but using $\Delta t_{ocean} = 7$ days significantly decreased the SST biases.

4.3 Climate Simulation

To evaluate the long-term stability of the coupled model and its ability to simulate the atmosphere and ocean climate, we compute a 70 -year long free run with the coupled model. For this

simulation, the coupled model is trained on ERA5 reanalyzes for the 26-year period from 1 January 1981 to 1 December 2006. After training both the atmospheric hybrid model and the ML-only ocean model are synchronized with ERA5 reanalyzes for the month of December 2006 until the simulation starts with the ERA5 reanalysis valid at 0000 UTC, 1 January 2007. During the free run the atmospheric hybrid model and ML-only ocean model evolve together.

4.3.1 Zonal Bias

We compare the annual zonal mean biases of our coupled model to SPEEDY using ERA5 as reference (Fig.4.3). Similar to A22, we find the coupled model is able to greatly reduce the zonal mean temperature biases. SPEEDY develops a large warm bias in the Equatorial upper troposphere and is too cold in the polar stratospheres. The machine learning component of the hybrid atmospheric model is able to correct these biases but not completely eliminate them.

SPEEDY has significant biases for zonal wind in the jet stream layer of the midlatitudes and in the stratosphere. Overall, the coupled model greatly reduces these biases but can not completely eliminate them. The coupled model does, however, introduce a large positive bias in the equatorial stratosphere. We found that this was related to the coupled model's inability to simulate the quasi-biennial oscillation (QBO) correctly. Instead of oscillating between the positive and negative phases of the QBO every 12-18 months, the coupled model stays in a near permanent positive phase during the entirety of the 70-year free run. This behavior was also observed in Section 3, suggesting that this is not related to coupling to the ML-only ocean model or the addition of precipitation as a prognostic variable.

The hybrid model greatly reduces the large positive humidity bias of SPEEDY with maxima in the tropics. The reduction of specific humidity biases in our coupled model is slightly worse than the hybrid atmospheric model Section 3 which does not predict 6-hourly precipitation. The addition of precipitation to our hybrid atmospheric model is what leads to this slight degradation in bias reduction. The positive specific humidity bias in the Southern Hemisphere tropics may be related to the wet bias in the precipitation climatology in the same area that we discuss in the next

section.

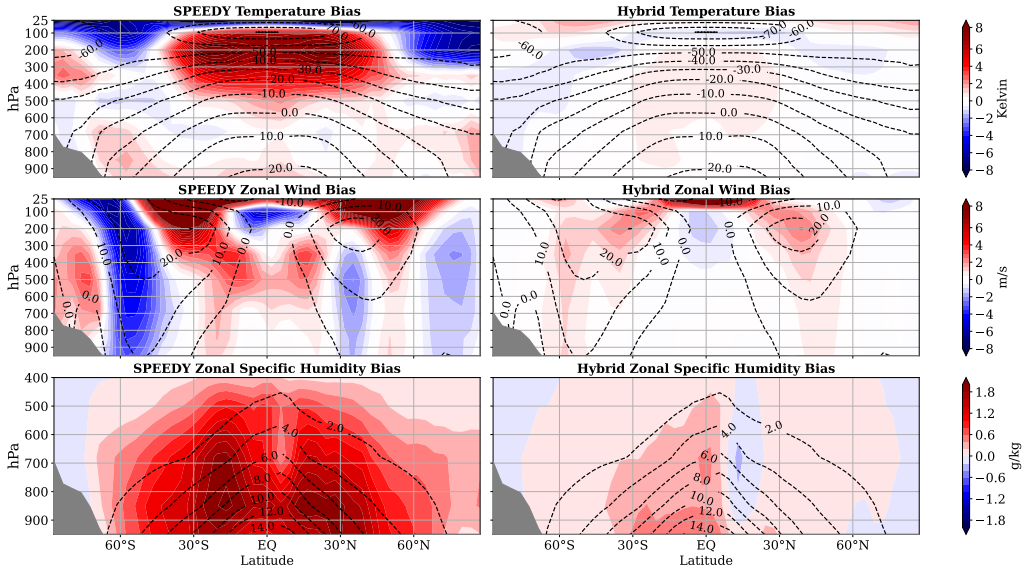

Figure 4.3: Comparison of the annual zonal mean biases of the SPEEDY and coupled model simulations. Results are shown for (top) the temperature (middle), zonal wind, and (bottom) specific humidity for (left) SPEEDY and (right) the coupled model.

4.3.2 Precipitation Climatology

Annual precipitation for SPEEDY and the coupled model are shown in Fig. 4.4, biases are computed by comparing each model to ERA5 for a 40-year period. While SPEEDY can replicate the overall global precipitation patterns, there are significant biases, most notably, large dry biases in the midlatitudes over the ocean. These dry biases are down-wind of major ocean currents (e.g. Gulf Stream and Kurisho Currents), suggesting that SPEEDY may underestimate the precipitation associated with midlatitude cyclones. SPEEDY exhibits a wet biases in the western Indian Ocean and Tropical Pacific. There are also wet biases over land in Central Africa, the Rocky Mountains of North America, and in Eastern Asia.

Overall, the coupled model greatly reduces the annual precipitation bias compared to SPEEDY, reducing the global annual precipitation RMSE of 1.29 mm/day in SPEEDY to 0.63 mm/day in

the coupled model (Table 4.1). The largest magnitudes of biases are also reduced in the coupled model (5.17 mm/day) compared to SPEEDY (-10.50 mm/day). The spatial correlation of the annual precipitation pattern in the coupled model matches better with ERA5 than SPEEDY (Pearson correlation coefficient of 0.96 vs. 0.82). The coupled model nearly eliminates the dry biases in the midlatitudes and improves rain climatology over land, most notably in the Rocky Mountains of the United States and China. The major expectation is over regions of the Tropical Pacific, where the coupled model has a larger biases than SPEEDY. The coupled model overestimates precipitation in that region and suggests the coupled model has too much convection near the equator and just south of the equator. The excessive amount of precipitation near and slightly south of the equator in the Pacific Ocean may be a manifestation of the double Intertropical Convergent Zone (ITCZ) problem commonly seen in coupled numerical models (Zhang et al. 2019). The excessive amount of precipitation may also be a symptom of the coupled model's positive specific humidity bias in the Southern Hemisphere tropical region (Fig.4.3).

	Annual Precipitation (mm/day)		
Model/ERA5	Global Max	Global Mean	Correlation to ERA5
ERA5	20.25	2.95	1.0
Coupled Model	18.32	3.15	0.96
SPEEDY	18.12	2.88	0.82

	Annual Precipitation Bias (mm/day)			
Model	Min Bias	Max Bias	Mean Bias	RMSE
Coupled Model	-3.83	5.17	0.19	0.63
SPEEDY	-10.50	10.29	-0.08	1.29

Table 4.1: Summary of annual precipitation climatology (top table) for ERA5, our coupled model, and SPEEDY. Summary of annual precipitation biases (bottom table) for our coupled model and SPEEDY. Lower biases mean better simulation of annual precipitation.

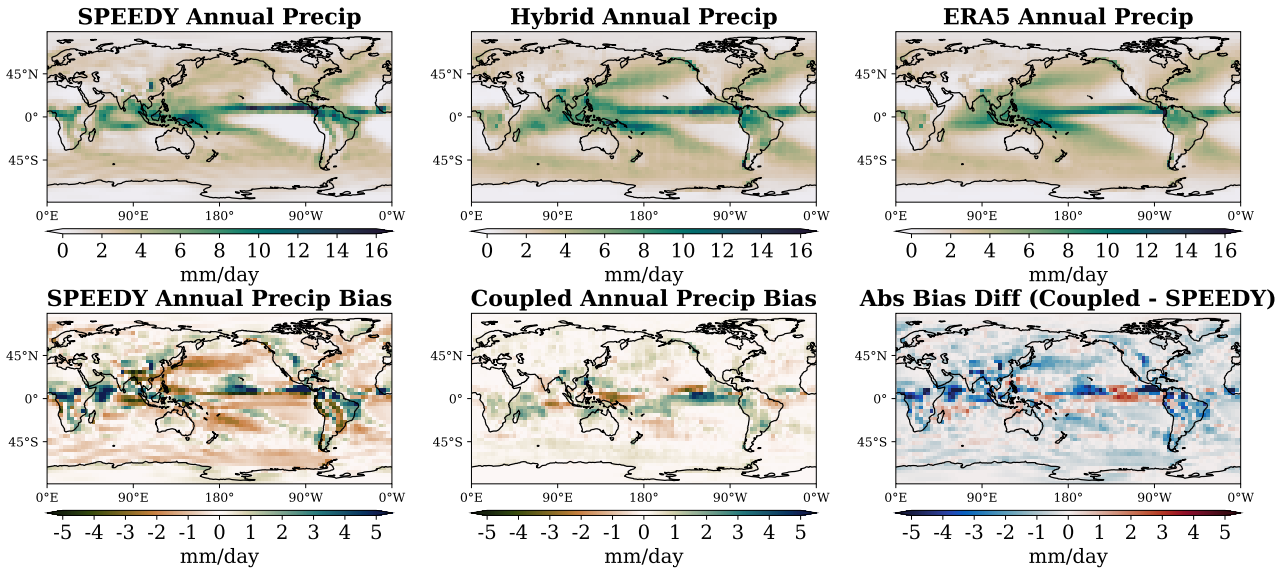

Figure 4.4: The comparison of the total annual precipitation (top row) for SPEEDY (left), the coupled model (middle), and ERA5 (right). The biases for annual precipitation (bottom row) are show for SPEEDY (left) and the coupled model (center). Also shown (bottom right) is the difference between the magnitude of the biases for SPEEDY and the coupled model (blue colors indicates locations where the coupled model has a lower bias than SPEEDY).

4.3.2.1 Precipitation Extremes

Understanding the change to precipitation extremes in future climates and being able to make short-term predictions for weather forecasts are important due to the wide-ranging socioeconomic effects they have from flooding to crop failure to wildfires. As discussed in Section 4.2.5, the prediction of extreme rainfall is heavily influenced by the choice of hyperparameters in the hybrid atmospheric model. For this dissertation, we compromised between annual precipitation bias and extreme rainfall prediction, but with tuning it is possible to better predict extreme rainfall rates at the expense of the model developing larger biases.

To evaluate how well the coupled model captures precipitation extremes, we examine 6-hourly total precipitation percentiles. Like Pathak et al. (2022) and Fildier et al. (2021), we use 100 logarithmically-spaced bins from the 0% to the 99.999%, in order to capture the most extreme values (Fig. 4.5). The coupled model and ERA5 are in general agreement until 99% percentile.

In contrast, SPEEDY does not match with ERA5 closely even at percentiles near 90%. Overall, SPEEDY overestimate the occurrence of low precipitation rates, while it underestimates the extreme high precipitation values. The overestimation of low precipitation rates and being unable to capture the most extreme precipitation rates is not unusual for a coarse resolution AGCM (e.g. Stephens et al. 2010). The coupled model tends to under-predict extreme precipitation events, however, the general slope of precipitation rates as a function of percentile generally matches ERA5. We note, that similar behavior of under-predicting extreme rainfall rates was observed in Pathak et al. (2022).

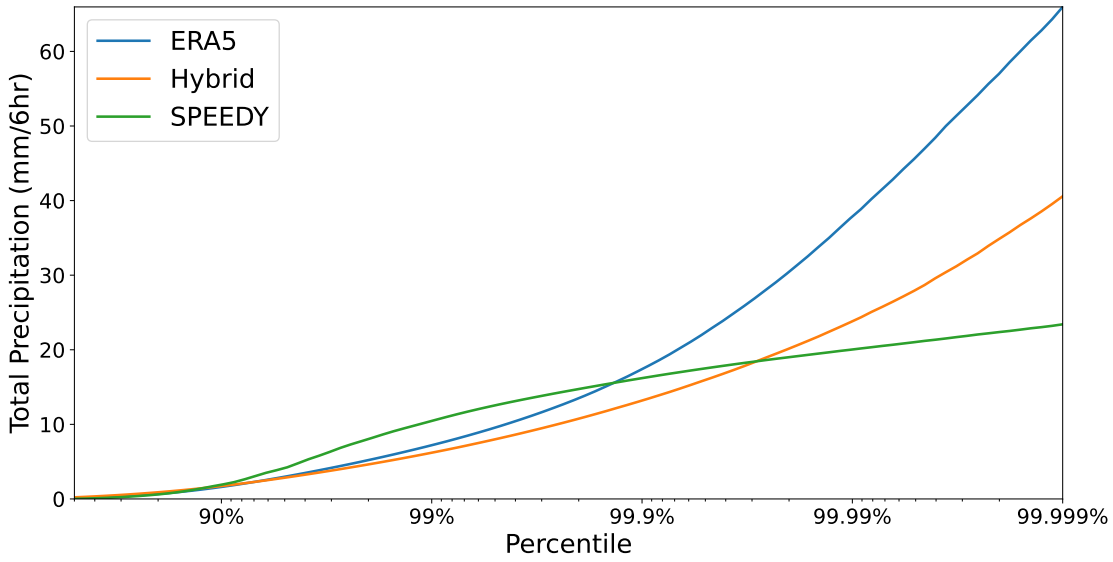

Figure 4.5: Comparison of extreme percentiles of 6-hourly total precipitation for ERA5 (blue), the coupled model (orange), and SPEEDY (green).

4.3.3 Ocean Climatology

The machine learning ocean model when coupled with the hybrid atmospheric model produces realistic ocean dynamics (e.g. annual cycle) and closely matches ERA5. The annual ocean climatology and the biases compared to ERA5 are shown in Figure 4.6. Overall, the coupled model

exhibits small biases with a RMSE of 0.43°C. There are several areas of notable biases, first there is a 1-2°C bias in the North Atlantic, near the Labrador Sea. The Labrador Sea cold bias may be explained by our simplistic handling of sea ice in our coupled model, although we do not see a similar bias in the Southern Ocean. There is also a 1°C warm bias in the Equatorial Pacific near the coast of South America. This bias is attributed to the ML-only ocean model having a slight bias toward the El Niño phase of the El Niño Southern Oscillation (ENSO).

To investigate the temporal variability of the ocean, we computed the average standard deviation of the monthly mean SST at each grid point (Fig. 4.7). The ML-ocean model is able to capture the spatial pattern of ocean variability seen in ERA5. Large variability in the SST fields are found in areas with ocean currents (e.g. Gulf Stream and Kursoshio Extension Region) and the Equatorial pacific (ENSO). The ML-ocean model tends to under-predict variability in areas influenced by ocean currents, however, the spatial characteristics match well with ERA5. Ocean variability near the interface between permanent sea ice and open ocean are also under-predicted, suggesting that the ML-only ocean model may not capture variability in the extent of sea ice, or the timing of the creation/melting of sea ice. We discuss ENSO in our coupled model in detail in the next section, but it should be noted that the ML-ocean model does have too much variability especially, near the coast of South America.

4.3.4 Variability

4.3.4.1 El Niño Southern Oscillation

The most important variability associated with the nonlinear dynamical coupling between the atmosphere and ocean is the El Niño Southern Oscillation (ENSO). ENSO can be described by the relationship between the change of near surface atmospheric winds, SST, and the equatorial Pacific thermocline (Bjerknes 1969). As such, AGCMs cannot replicate ENSO. Typically, traditional physics-based coupled climate models require a sophisticated multi-layer ocean model to reproduce ENSO. Our coupled model consisting of only a hybrid atmospheric model and a ML-based ocean model trained to predict the SSTs, can reproduce an ENSO-like signal in both the atmosphere and

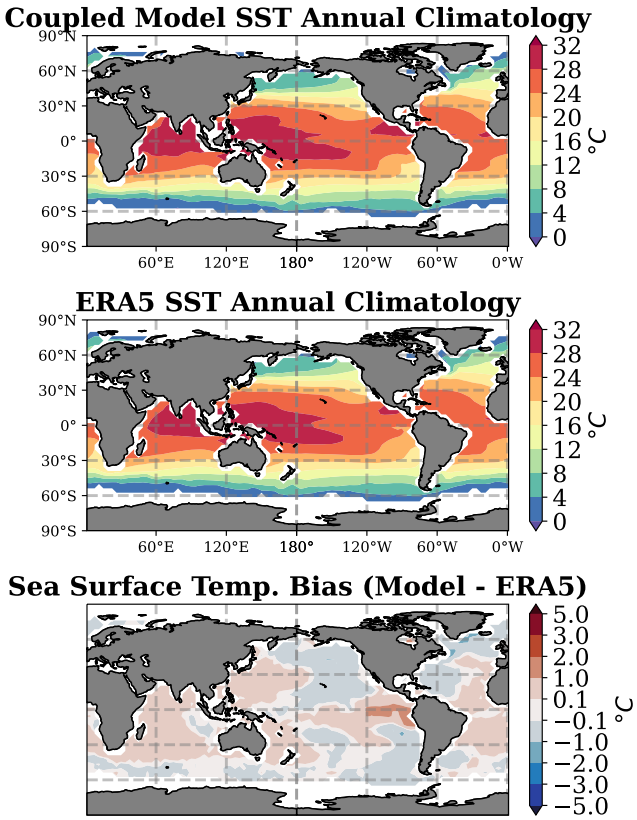

Figure 4.6: Annual Averaged sea surface temperatures for the coupled model (top panel), ERA5 (middle panel), and the model bias (bottom panel). The annual SST climatology for the coupled model is from the first 40 years of the free run and ERA5 is averaged from 1981-2021.

the SST fields. Two of the most common metrics used to diagnose ENSO phase is the Oceanic Nino Index and the Southern Oscillation Index. ONI represents the a 3-month running mean of SST anomalies in the Niño 3.4 region (5°S-5°N, 120°W-170°W). SOI measures the monthly standardized atmospheric pressure difference between Tahiti and Darwin, Australia. Figure 4.8 shows the evolution of the ONI and SOI during the first 55 years of the climate simulation. The coupled model is able to simulate the inverse relationship between the ONI and SOI.

To further examine ENSO in our coupled model, we use wavelet analysis (Torrence and Compo 1998) with HadISST used as reference (Rayner et al. 2003) (Fig 4.9). The coupled model has sharp peak at 5 years, and is in general agreement with HadISST which has a broad peak between 3.5 and 5 years. Generally, the coupled model has too much variability from 1 year onward. State-of-

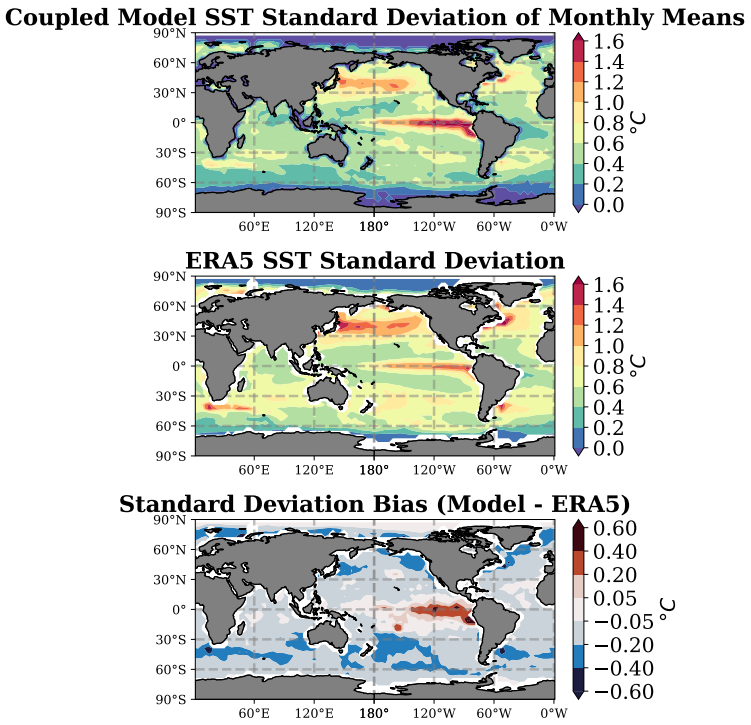

Figure 4.7: Average monthly sea surface temperature standard deviation for the coupled model (top panel), ERA5 (middle panel), and the model bias (bottom panel). The monthly SST standard deviation climatology for the coupled model is based off the first 40 years of the free run and ERA5 is for the period of 1981-2021.

the-art ESMs such as E3SM and CESM2 have similar problems with too much variability in the 2-5 year period (Golaz et al. 2019; Capotondi et al. 2020).

The autocorrelation functions of the Niño 3.4 SST anomalies are shown for ERA5 and the coupled model (Fig. 4.10). The coupled model is in strong agreement with ERA5 for the first 6 months of lag. However, the coupled model fails to capture the correct timing of the cross over into negative correlation occurring at 10 months for ERA5, which occurs at 15 months instead. The coupled model does capture the magnitude of minimum correlation (-0.30), but is delayed, having the minimum at a lag of 34 months rather than 24 months. This result indicates that the coupled model is delayed in transitioning from one phase of ENSO to another.

Some climate models fail to properly capture the western extent of the ENSO variability, producing too much variability in that region (e.g. Menary et al. 2018). The spatial pattern of variability in our coupled model in the western Tropical Pacific, especially west of 180W, matches well

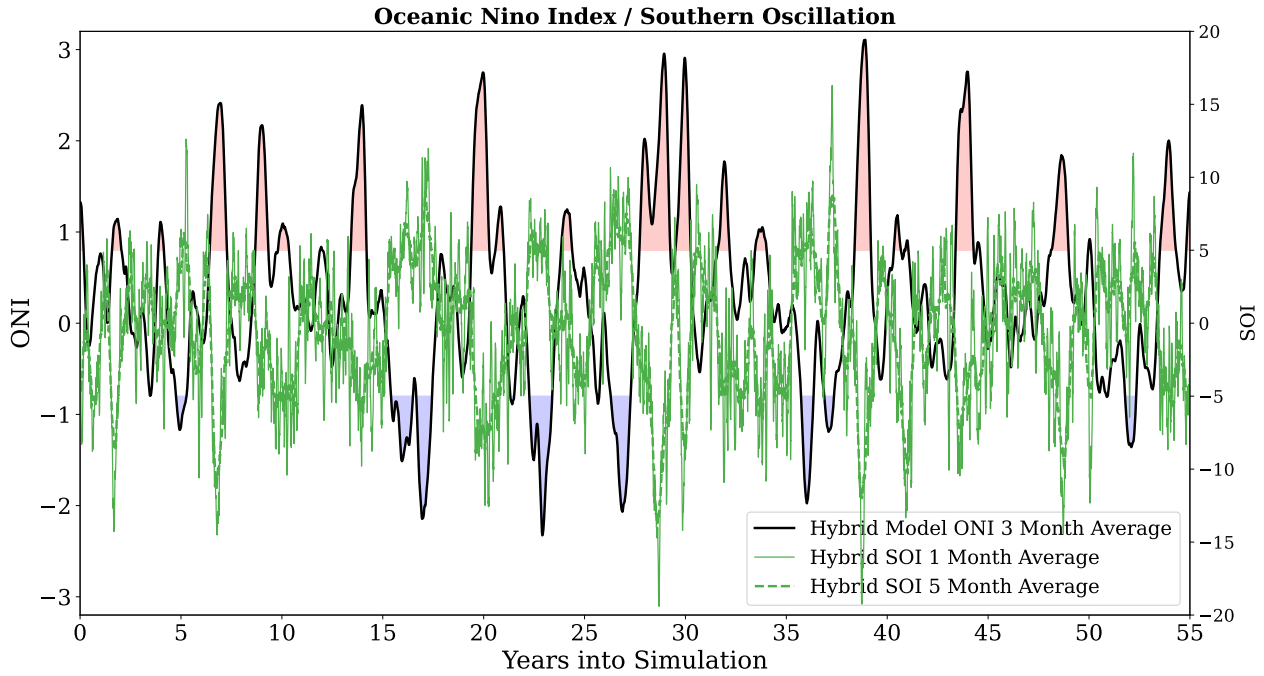

Figure 4.8: A time-series showing the Ocean Nino Index (ONI) and the Southern Oscillation Index (SOI) for the first 55 years of a 70-year free run. The color fill of the ONI indicates when the criteria is met to be classified as an El Niño (red fill) and La Niña (blue fill). The monthly SOI value and trailing 5-month average are show. Negative SOI values typically occur during an El Niño and positive values during an La Niña.

with ERA5 showing minimal bias in this region, indicating that our coupled model can capture the western extent of ENSO variability (Fig. 4.7).

4.3.4.2 Atmosphere Variability

We investigate sudden stratospheric warming (SSW) in our coupled model to evaluate the model's atmospheric variability and the ability for our coupled model to simulate troposphere-stratosphere coupling. SSW events are caused by the disruption of the winter Northern Hemisphere stratospheric polar vortex from upward propagating tropospheric waves (Andrews et al. 1987). In order to simulate stratospheric dynamics, physics-based models require higher vertical resolution and more model layers in the stratosphere than our hybrid atmosphere mode. SPEEDY, which has the same vertical resolution and model levels as the hybrid atmospheric model, is unable to correctly simulate stratospheric dynamics. This can be seen by SPEEDY's large zonal biases in

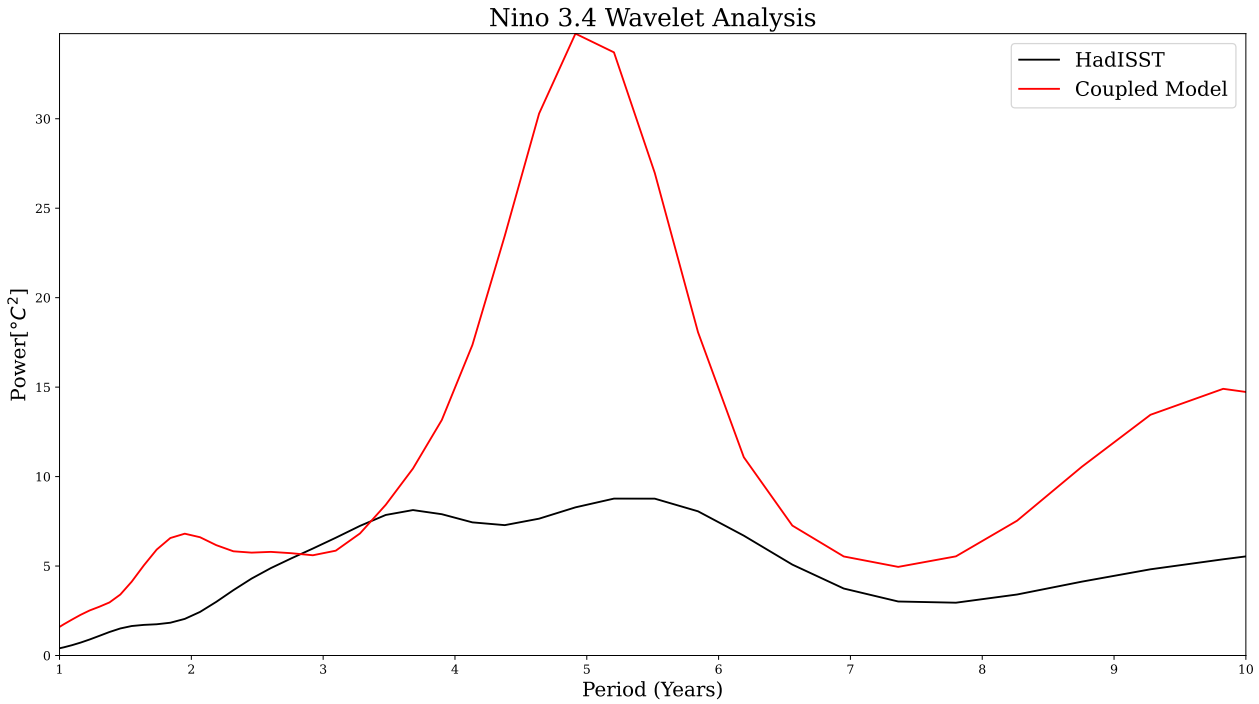

Figure 4.9: Wavelet power spectrum of ENSO (Nino3.4) using a Morlet wavelet of degree 6 for the coupled model (red) and HadISST (black).

the stratospheric for both temperature and zonal wind (Fig. 4.3). The hybrid atmospheric model significantly reduces these biases, while it also better captures the variability in the stratosphere.

Looking at the 20 hPa zonal wind climatology between 55°N-65°S, our coupled model matches well with ERA5 (Fig. 4.11). The coupled model is able to capture the annual cycle with strong westerlies during the winter months, and then the weakening of the winter stratospheric polar vortex during early spring, and finally reversal of the winds to easterlies during the summer months. SPEEDY consistently has a positive bias, with winds too strong in each month of the year and it does not capture the reversal of winds during the summer.

The grey shaded regions of Figure 4.11 represent ± 2 standard deviations from the mean. Using this as proxy for variability, we see that during the winter months ERA5 has large variability, with values ranging from 55 m/s to -10 m/s. This large variability is dominated by the varying year-to-year strength of the winter stratospheric polar vortex and the occasional SSW where the polar vortex is severely disrupted and easterly winds can develop for a short time. SPEEDY being a

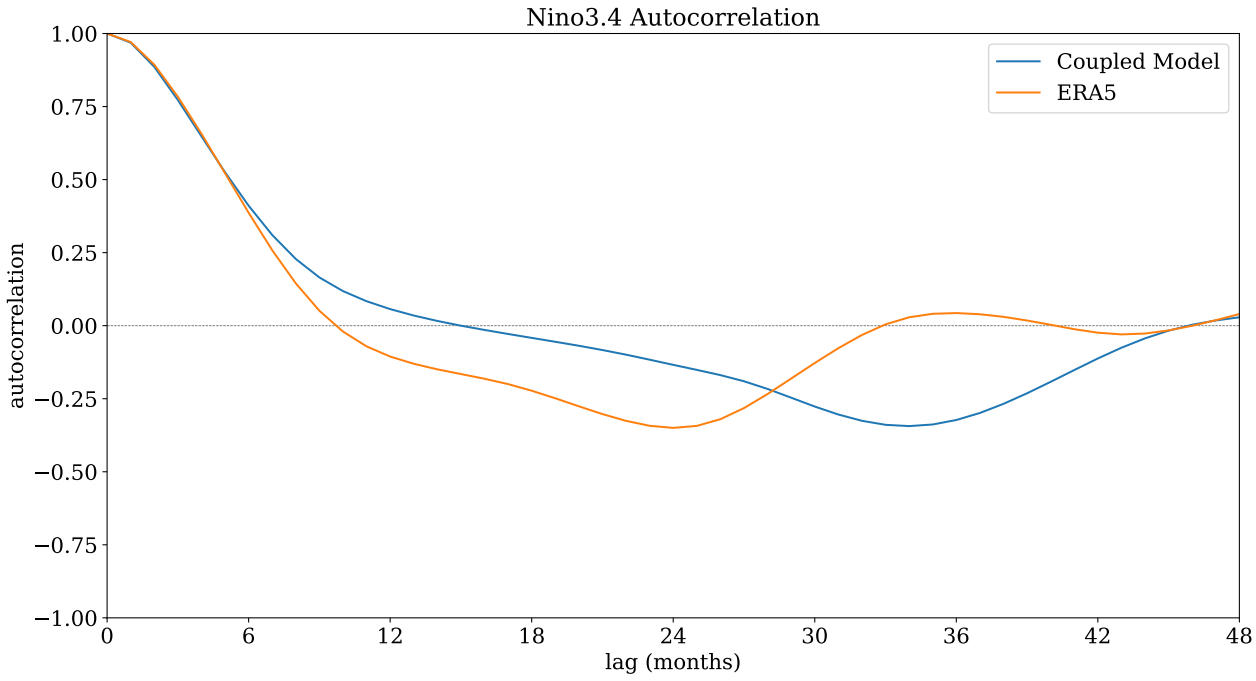

Figure 4.10: The autocorrelation functions of Niño 3.4 for the coupled model (blue) and ERA5 over the period of 1981-2021 (orange).

coarse resolution model with insufficient vertical resolution in the stratosphere is unable to replicate this variability. The coupled model is able capture this variability, with values closely matching ERA5, which indicates that the coupled model can simulate SSW events.

To identify SSW events in our coupled model, we use the same criteria as Charlton and Polvani (2007) except we look at 20 hPa zonal mean winds at 60°N instead of 10 hPa zonal mean winds. Because the top of the coupled model is at $\sigma = 0.025$, we first convert to pressure coordinates and then linearly interpolate to 20 hPa. We found that our coupled model slightly over-predicts the occurrences of SSW events (0.84 per year) during the 70-year free compared to the observed value of ~0.6 per year (Charlton and Polvani 2007). It should be noted, SSW events are almost completely exclusive to the NH and only one has ever been observed in the SH. The coupled model did not produce any SSW event during the 70-year free run. A significantly longer simulation would be necessary to determine if the coupled model would be able to produce infrequent SSW events in the SH.

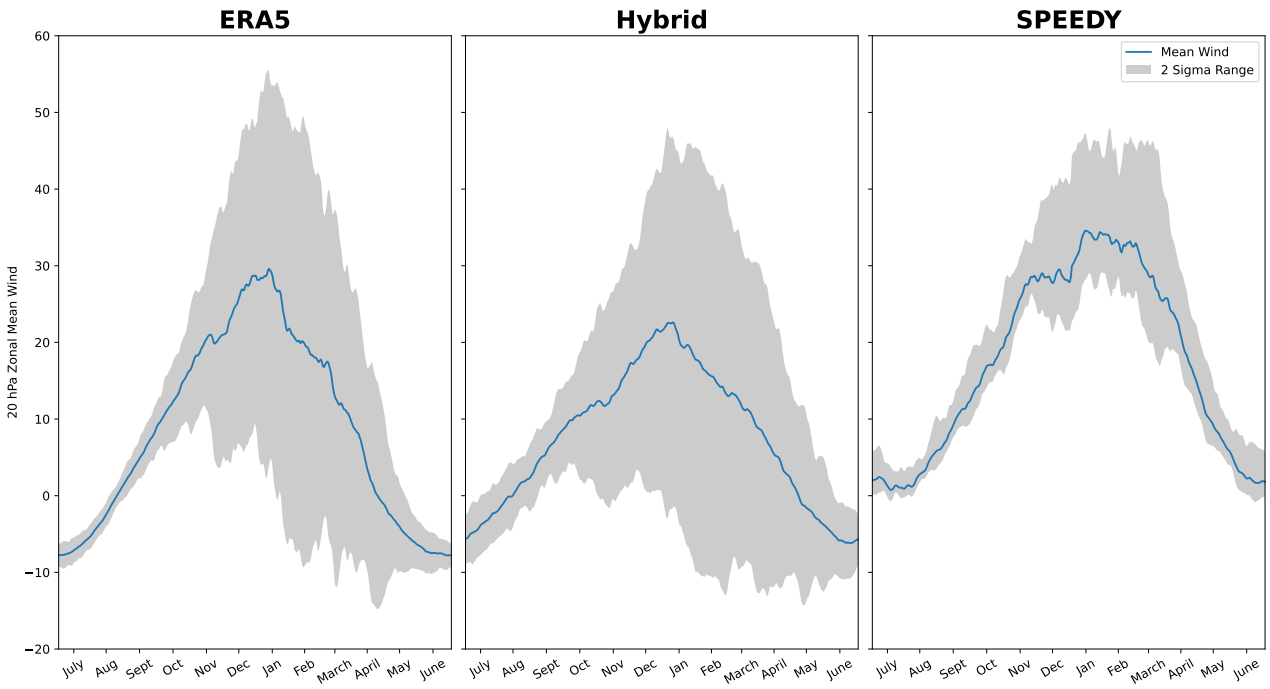

Figure 4.11: 20 hPa mean wind (blue line) and 2 standard deviations (grey shaded region) for ERA5 1981-2018 (left panel), our coupled model (center panel), and SPEEDY (right panel).

4.3.5 Stability and Climate Drift

One of the major challenges with incorporating machine learning into NWP is instability and climate drift. Traditional numerical physics-based climate models enforce the conservation laws (e.g. mass and global water budget). This allows a well-designed climate model to produce long simulations without a significant drift of the mass or energy of the atmosphere. We evaluate the conservation of total atmospheric mass and the total mass of water vapor in our coupled model to determine whether there is an considerable mass or water vapor change over the 70-year free run. Values for the total atmospheric mass and water vapor contribution were calculated similarly to Trenberth and Smith (2005). We also compare our coupled model to ERA5 and SPEEDY.

We found our coupled model does conserve the total mass of the atmosphere well with no significant trend observed (-0.02% per century) (Fig.4.12). However, the total mass of the atmosphere is much more variable for the coupled model than for ERA5 or SPEEDY. The coupled model does well with conserving total atmospheric mass of water vapor (Fig. 4.13). While there is a strong

annual cycle in the total mass of water vapor, there is no trend during the 70-year free run. We also note that the total mass of water vapor and the range of the annual cycle is in good agreement with ERA5 and calculated values from other reanalysis products (Trenberth and Smith 2005).

To evaluate climate drift of our coupled model, we look at the globally averaged lowest model level temperature during the 70-year free run. The lowest model level global mean temperature is stable during the 70-year free run with almost no linear trend (0.077 $^{\circ}C$ per century) (Fig. 4.14).

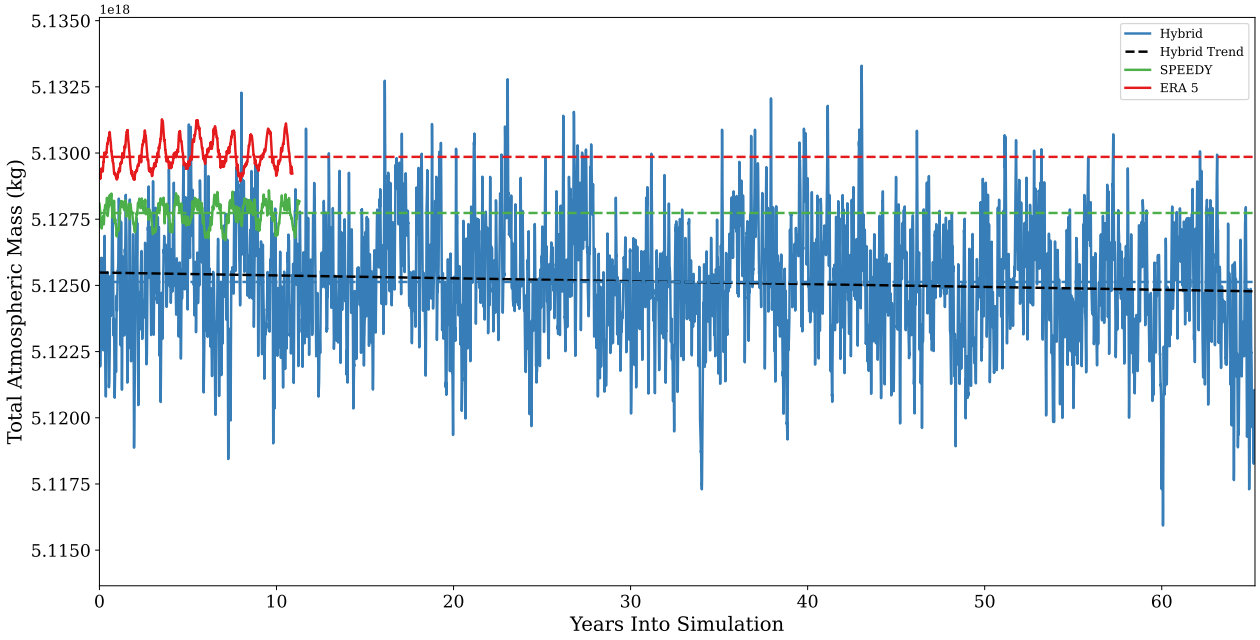

Figure 4.12: Time series of total atmospheric mass for the coupled model (solid blue line) and linear trend during the free run (dashed black line). 10 years of ERA5 (solid red line) and the mean for 1981-2018 (dashed red line) and 10 years of SPEEDY (solid green line) and mean (dashed green line) are shown for reference.

4.4 Conclusion

In this chapter, we described results of a coupled hybrid atmospheric model with a machine learning-only ocean model. We trained a parallel, reservoir computing based model using ERA5 to predict the sea surface temperatures from past ocean and atmosphere states. The ML-only based ocean model was then coupled to the hybrid atmospheric model. This coupled model can repro-

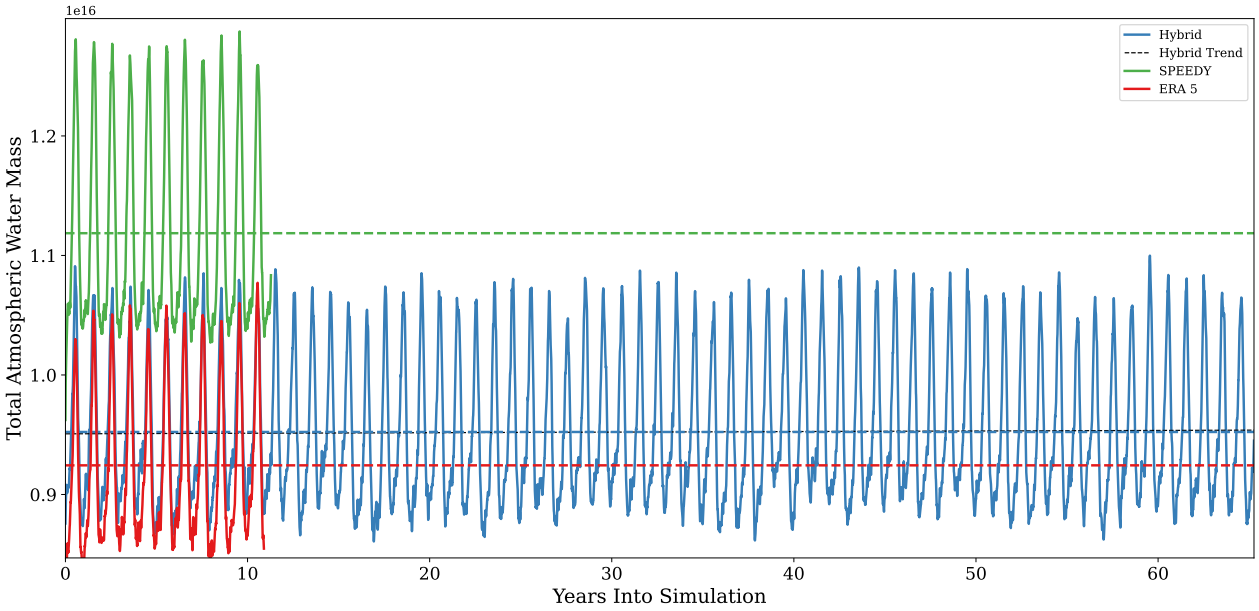

Figure 4.13: Same as Figure 4.12 for total atmospheric water vapor mass.

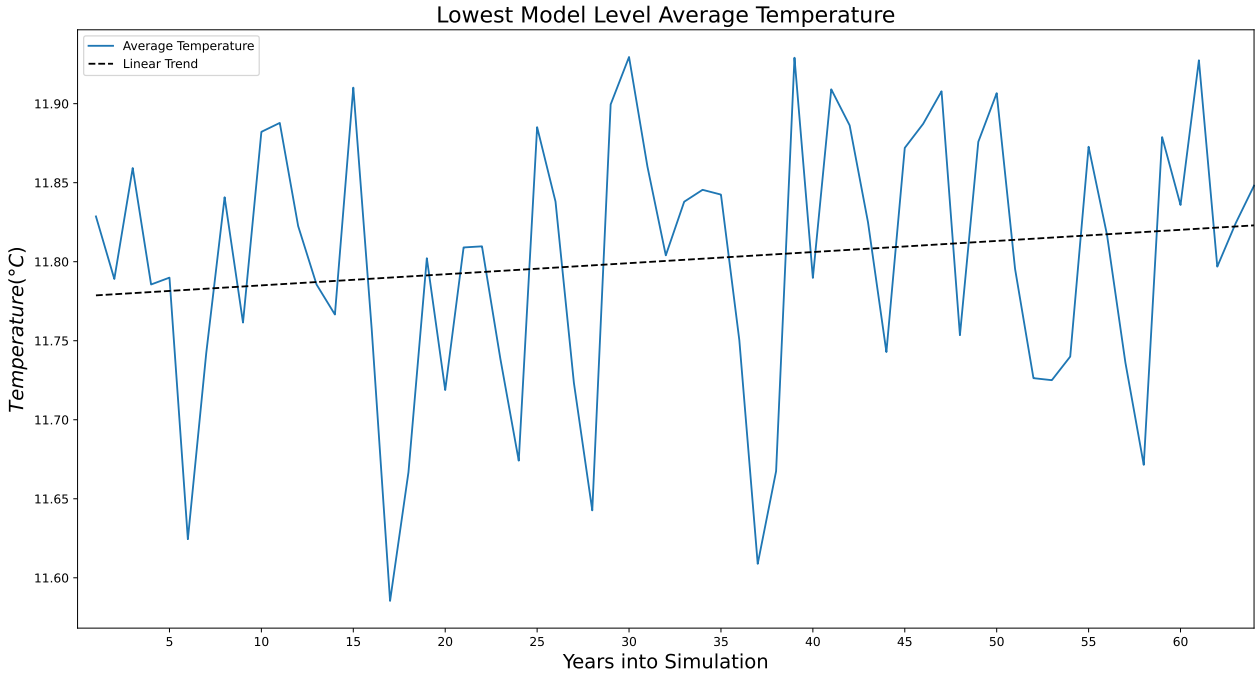

Figure 4.14: Time series of the area averaged annual mean temperature of the lowest model level in our coupled model during the 70-year free run (solid blue) and linear trend (dashed black).

duce long term variability of both the atmosphere and ocean (e.g. ENSO and sudden stratospheric warming).

Compared to state-of-the-art climate models, our coupled model requires significantly less computational resources. Once trained, the coupled model can be run on any desktop or small computer cluster, the only requirement being the availability of 32 Gigabytes of RAM. Modern climate models typically require a high performance computing cluster with thousands or tens of thousands of processors and can only simulate a decade or two per day (e.g. Golaz et al. 2019). For the climate experiment presented in this study, we use 32 Intel Xeon 6248R processors and can simulate 4 years per hour (96 simulated years per day). The addition of the ML-ocean model does not add any significant run time when compared to a stand-alone version of the hybrid atmospheric model. In forecast mode the computational bottlenecks are the IO associated with restarting SPEEDY for each hybrid atmospheric timestep.

The coupling of an ML-ocean model to our hybrid atmospheric model is a major step towards our ultimate goal of using the coupled model for climate change research. While we show that the coupled model is able to replicate the past and current climate, climate change is inherently a nonstationary dynamical problem. Patel et al. (2021) and Patel and Ott (2022) outline a method to incorporate nonstationarity into a hybrid model similar to the one presented in this study. Using this method, their hybrid model was able to anticipate tipping points and simulate post-tipping point climates in toy models. Our plan is to apply this method to the coupled model presented in this study for climate change research.

5. CONCLUSIONS

In this study, we used the parallel, reservoir-computing algorithm of Pathak et al. (2018a) and the hybrid modeling approach of Pathak et al. (2018b) and Wikner et al. (2020) to demonstration the potential of machine learning for improved weather and climate modeling. Our results in Section 2 showed that a machine learning-based global atmospheric model can predict the 3-dimensional atmosphere in the same format as a NWP model. Once trained, the global ML weather model is significantly faster than a traditional numerical physics-based model producing 21-day forecasts in 30 seconds.

Next, we combined this parallel, reservoir-computing algorithm with a simplified AGCM for weather prediction and climate simulation. Using a variety of verification metrics, including RMSE and model bias, we demonstrated that the hybrid model can improve weather forecasts compared to the host AGCM for all variables for at least the first seven forecast days. Forecasts from the hybrid model are well-balanced and had improved variability, highlighting the potential application for using the hybrid model for data assimilation. Future work will utilize the iterative method of Wikner et al. (2021) to produce analyses using the hybrid atmospheric model.

To test the stability and the ability of the hybrid model to simulate the past and present climate, we carried out a 11-year free run. Zonal mean biases of temperature, wind, and specific humidity were greatly reduced compared to the host AGCM. For temperature and zonal wind, the hybrid model produced biases on par with state-of-the-art GCMs, but requiring significantly less computational resources.

Finally, we demonstrated the potential of ML for coupling a computationally inexpensive model of another Earth system component by dynamically coupling an ML-only ocean model of the SST. This coupled model was stable and did not exhibit a climate drift during a 70-year free run. The coupled model could reproduce important components of the atmosphere and ocean variability without significant biases. We investigated the skill of our coupled model in simulating ENSO and found that the coupled model was able to produce an ENSO-like response in both the atmosphere

and ocean. This indicates that our model can capture nonlinear interaction between the atmosphere and ocean. The spatial extent and frequency of ENSO matches well with observations, but, like in many GCMs, in our model the SST has too much variability in the El Nino region in the 2-5 year period range and it is delayed in switching the phase of ENSO. To the best of our knowledge, this is the first time that a ML-based ocean model has been coupled to a hybrid atmospheric model, representing a major step of applying ML for Earth System modeling.

REFERENCES

, 2020: *IFS Documentation CY47R1 - Part III: Dynamics and Numerical Procedures*. IFS Documentation, ECMWF.

Andrews, D. G., J. R. Holton, and C. B. Leovy, 1987: *Middle atmosphere dynamics*, International Geophysics, Vol. 40. Academic Press.

Arakawa, A., and V. R. Lamb, 1977: *Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model*, Vol. 17, 173–265. Elsevier.

Arcomano, T., I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott, 2020: A machine learning-based global atmospheric forecast model. *Geophysical Research Letters*, **47**, e2020GL087 776.

Arcomano, T., I. Szunyogh, A. Wikner, J. Pathak, B. R. Hunt, and E. Ott, 2022: A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model. *Journal of Advances in Modeling Earth Systems*, **14 (3)**, e2021MS002 712, doi:https://doi.org/10.1029/2021MS002712.

Arias, P., and Coauthors, 2021: *Technical Summary*, 33−144. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, doi:10.1017/9781009157896.002.

Bauer, P., A. Thorpe, and G. Brunet, 2015: The quiet revolution of numerical weather prediction. *Nature*, **525**, 47–55.

Beucler, T., and Coauthors, 2021: Climate-invariant machine learning. arXiv, doi:10.48550/ARXIV.2112.08440.

Bishop, C. H., S. Frolov, D. R. Allen, D. D. Kuhl, and K. Hoppel, 2017: The local ensemble tangent linear model: an enabler for coupled model 4d-var. *Quarterly Journal of the Royal Meteorological Society*, **143 (703)**, 1009–1020, doi:https://doi.org/10.1002/qj.2986.

Bjerknes, J., 1969: Atmospheric teleconnections from the equatorial pacific. *Monthly Weather Review*, **97 (3)**, 163–172, doi:10.1175/1520-0493(1969)097<0163:Atftep>2.3.Co;2.

Brenowitz, N. D., and C. S. Bretherton, 2018: Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, **45**, 6289–6298.

Brenowitz, N. D., and C. S. Bretherton, 2019: Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, **11**, 2728–2744.

Bretherton, C. S., and Coauthors, 2022: Correcting coarse-grid weather and climate models by machine learning from global storm-resolving simulations. *Journal of Advances in Modeling Earth Systems*, **14 (2)**, e2021MS002 794, doi:https://doi.org/10.1029/2021MS002794.

Caldwell, P. M., and Coauthors, 2021: Convection-permitting simulations with the e3sm global atmosphere model. *Journal of Advances in Modeling Earth Systems*, **13 (11)**, e2021MS002 544, doi:https://doi.org/10.1029/2021MS002544.

Campos, R. M., V. Krasnopolsky, J.-H. G. M. Alves, and S. G. Penny, 2019: Nonlinear wave ensemble averaging in the gulf of mexico using neural networks. *Journal of Atmospheric and Oceanic Technology*, **36 (1)**, 113–127, doi:10.1175/jtech-d-18-0099.1.

Capotondi, A., C. Deser, A. S. Phillips, Y. Okumura, and S. M. Larson, 2020: Enso and pacific decadal variability in the community earth system model version 2. *Journal of Advances in Modeling Earth Systems*, **12 (12)**, e2019MS002 022, doi:https://doi.org/10.1029/2019MS002022.

Chapman, W. E., A. C. Subramanian, L. Delle Monache, S. P. Xie, and F. M. Ralph, 2019: Improving atmospheric river forecasts with machine learning. *Geophysical Research Letters*, **46 (17-18)**, 10 627–10 635, doi:https://doi.org/10.1029/2019GL083662.

Charlton, A. J., and L. M. Polvani, 2007: A new look at stratospheric sudden warmings. part i: Climatology and modeling benchmarks. *Journal of Climate*, **20 (3)**, 449–469, doi:10.1175/jcli3996.1.

Chattopadhyay, A., A. Subel, and P. Hassanzadeh, 2020: Data-driven super-parameterization using deep learning: Experimentation with multiscale lorenz 96 systems and transfer learning. *Journal of Advances in Modeling Earth Systems*, **12 (11)**, e2020MS002 084, doi:https://doi.org/10.1029/2020MS002084.

Chen, J., R. Arsenault, F. P. Brissette, and S. Zhang, 2021: Climate change impact studies: Should we bias correct climate model outputs or post-process impact model outputs? *Water Resources*

Research, **57 (5)**, e2020WR028 638, doi:https://doi.org/10.1029/2020WR028638.

Chevallier, F., J.-J. Morcrette, F. Chéruy, and N. A. Scott, 2000: Use of a neural-network-based long-wave radiative-transfer scheme in the ecmwf atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, **126 (563)**, 761–776, doi:https://doi.org/10.1002/qj.49712656318.

Clark, S. K., and Coauthors, 2022: Correcting a 200 km resolution climate model in multiple climates by machine learning from 25 km resolution simulations. *Journal of Advances in Modeling Earth Systems*, **14 (9)**, e2022MS003 219, doi:https://doi.org/10.1029/2022MS003219.

Danabasoglu, G., and Coauthors, 2020: The community earth system model version 2 (cesm2). *Journal of Advances in Modeling Earth Systems*, **12 (2)**, e2019MS001 916, doi:https://doi.org/10.1029/2019MS001916.

Dueben, P. D., and P. Bauer, 2018: Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Model Dev.*, **11 (10)**, 3999–4009, doi:10.5194/gmd-11-3999-2018.

Farchi, A., P. Laloyaux, M. Bonavita, and M. Bocquet, 2021: Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, **147 (739)**, 3067–3084, doi:https://doi.org/10.1002/qj.4116.

Fildier, B., W. D. Collins, and C. Muller, 2021: Distortions of the rain distribution with warming, with and without self-aggregation. *Journal of Advances in Modeling Earth Systems*, **13 (2)**, e2020MS002 256, doi:https://doi.org/10.1029/2020MS002256.

Flato, G., and Coauthors, 2014: *Evaluation of climate models*, 741–866. Cambridge University Press.

Flora, M. L., C. K. Potvin, P. S. Skinner, S. Handler, and A. McGovern, 2021: Using machine learning to generate storm-scale probabilistic guidance of severe weather hazards in the warn-on-forecast system. *Monthly Weather Review*, **149 (5)**, 1535–1557, doi:10.1175/mwr-d-20-0194.1.

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

Gilbert, E. N., 1959: Random graphs. *Ann. Math. Statist.*, **30**, 1141–1144.

Golaz, J.-C., and Coauthors, 2019: The doe e3sm coupled model version 1: Overview and evaluation at standard resolution. *Journal of Advances in Modeling Earth Systems*, **11 (7)**, 2089–2129, doi:https://doi.org/10.1029/2018MS001603.

Goodfellow, I. J., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, Cambridge, MA, USA.

Ham, Y.-G., J.-H. Kim, and J.-J. Luo, 2019: Deep learning for multi-year enso forecasts. *Nature*, **573 (7775)**, 568–572, doi:10.1038/s41586-019-1559-7.

Hans, H., and Coauthors, 2019: Global reanalysis: goodbye era-interim, hello era5. *ECMWF Newsletter*, **(159)**.

Harper, K. C., 2008: *Weather by the Numbers The Genesis of Modern Meteorology*. The MIT Press, doi:10.2307/j.ctt5hhddq.

Haupt, S. E., A. Pasini, and C. Marzban, 2008: *Artificial Intelligence Methods in the Environmental Sciences*. Springer Publishing Company, Incorporated.

Hersbach, H., and Coauthors, 2020: The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, **146 (730)**, 1999–2049, doi:https://doi.org/10.1002/qj.3803.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2007: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, **230 (1)**, 112–126, doi:https://doi.org/10.1016/j.physd.2006.11.008.

Ivanov, M. A., J. Luterbacher, and S. Kotlarski, 2018: Climate model biases and modification of the climate change signal by intensity-dependent bias correction. *Journal of Climate*, **31 (16)**, 6591–6610, doi:10.1175/jcli-d-17-0765.1.

Jaeger, H., 2001: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, **148**.

Kim, H., Y. G. Ham, Y. S. Joo, and S. W. Son, 2021: Deep learning for bias correction of mjo prediction. *Nature Communications*, **12 (1)**, 3087, doi:10.1038/s41467-021-23406-3.

Krasnopolsky, V., and M. S. Fox-Rabinovitz, 2006: Complex hybrid models combining determin-

istic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, **19**, 122–134.

Krasnopolsky, V., M. S. Fox-Rabinovitz, and A. A. Belochitski, 2010: Development of neural network convection parameterizations for numerical climate and weather prediction models using cloud resolving model simulations. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Krasnopolsky, V., 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. *Monthly Weather Review*, **133 (5)**, 1370–1383.

Krasnopolsky, V. M., 2013: *Applications of NNs to Developing Hybrid Earth System Numerical Models for Climate and Weather*, 81–143. Springer Netherlands, Dordrecht, doi:10.1007/978-94-007-6073-8_4.

Krasnopolsky, V. M., and F. Chevallier, 2003: Some neural network applications in environmental sciences. part ii: advancing computational efficiency of environmental numerical models. *Neural Networks*, **16 (3)**, 335–348, doi:https://doi.org/10.1016/S0893-6080(03)00026-1.

Kucharski, F., F. Molteni, and A. Bracco, 2006: Decadal interactions between the western tropical pacific and the north atlantic oscillation. *Climate Dynamics*, **26 (1)**, 79–91.

Kucharski, F., F. Molteni, M. P. King, R. Farneti, I.-S. Kang, and L. Feudale, 2013: On the need of intermediate complexity general circulation models: A "speedy" example. *Bulletin of the American Meteorological Society*, **94 (1)**, 25–30, doi:10.1175/bams-d-11-00238.1.

Lagerquist, R., A. McGovern, C. R. Homeyer, D. J. Gagne II, and T. Smith, 2020: Deep learning on three-dimensional multiscale data for next-hour tornado prediction. *Monthly Weather Review*, **148 (7)**, 2837–2861, doi:10.1175/mwr-d-19-0372.1.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *Nature*, **521 (7553)**, 436–444, doi:10.1038/nature14539.

Li, G., and Coauthors, 2019: Effect of excessive equatorial pacific cold tongue bias on the el niño-northwest pacific summer monsoon relationship in cmip5 multi-model ensemble. *Climate*

Dynamics, **52 (9)**, 6195–6212, doi:10.1007/s00382-018-4504-9.

Lukoševičius, M., 2012: *A Practical Guide to Applying Echo State Networks*, 659–686. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lukoševičius, M., and H. Jaeger, 2009: Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, **3 (3)**, 127–149.

Lynch, P., 2006: *The emergence of numerical weather prediction: Richardson's dream*. Cambridge University Press.

Lynch, P., 2008: The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, **227 (7)**, 3431–3444, doi:https://doi.org/10.1016/j.jcp.2007.02.034.

Ma, P.-L., and Coauthors, 2015: How does increasing horizontal resolution in a global climate model improve the simulation of aerosol-cloud interactions? *Geophysical Research Letters*, **42 (12)**, 5058–5065, doi:https://doi.org/10.1002/2015GL064183.

Maass, W., T. Natschläger, and H. Markram, 2002: Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, **14 (11)**, 2531–2560, doi:10.1162/089976602760407955.

Menary, M. B., and Coauthors, 2018: Preindustrial control simulations with hadgem3-gc3.1 for cmip6. *Journal of Advances in Modeling Earth Systems*, **10 (12)**, 3049–3075, doi:https://doi.org/10.1029/2018MS001495.

Molteni, F., 2003: Atmospheric simulations using a GCM with simplified physical parametrizations. I: model climatology and variability in multi-decadal experiments. *Climate Dynamics*, **20 (2)**, 175–191.

Ott, E., and Coauthors, 2004: A local ensemble Kalman filter for atmospheric data assimilation. *Tellus*, **56 (A)**, 415–428.

Patel, D., D. Canaday, M. Girvan, A. Pomerance, and E. Ott, 2021: Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **31 (3)**, 033 149, doi:10.1063/5.0042598.

Patel, D., and E. Ott, 2022: Using machine learning to anticipate tipping points and extrapolate to post-tipping dynamics of non-stationary dynamical systems. *arXiv preprint arXiv:2207.00521*.

Pathak, J., B. Hunt, M. Girvan, Z. Lu, and E. Ott, 2018a: Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, **120 (2)**, 024 102.

Pathak, J., A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, 2018b: Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos*, **28 (4)**, 041 101.

Pathak, J., and Coauthors, 2022: Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. arXiv, doi:10.48550/ARXIV.2202.11214.

Pradhan, R., R. S. Aygun, M. Maskey, R. Ramachandran, and D. J. Cecil, 2018: Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Transactions on Image Processing*, **27 (2)**, 692–702, doi:10.1109/TIP.2017.2766358.

Rasp, S., 2020: Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and lorenz 96 case study (v1.0). *Geosci. Model Dev.*, **13 (5)**, 2185–2196, doi:10.5194/gmd-13-2185-2020.

Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, **146 (11)**, 3885–3900.

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

Rasp, S., and N. Thuerey, 2021: Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for weatherbench. *Journal of Advances in Modeling Earth Systems*, **13 (2)**, e2020MS002 405.

Rayner, N. A., D. E. Parker, E. B. Horton, C. K. Folland, L. V. Alexander, D. P. Rowell, E. C. Kent, and A. Kaplan, 2003: Global analyses of sea surface temperature, sea ice, and night marine air temperature since the late nineteenth century. *Journal of Geophysical Research: Atmospheres*, **108 (D14)**, doi:https://doi.org/10.1029/2002JD002670.

Sarkar, P. P., P. Janardhan, and P. Roy, 2020: Prediction of sea surface temperatures using deep learning neural networks. *SN Applied Sciences*, **2 (8)**, 1458, doi:10.1007/s42452-020-03239-3.

Sarker, I. H., 2021: Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, **2 (3)**, 160, doi:10.1007/s42979-021-00592-x.

Scher, S., and G. Messori, 2018: Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, **144 (717)**, 2830–2841.

Scher, S., and G. Messori, 2019: Weather and climate forecasting with neural networks: using general circulation models (GCMs) with different complexity as a study ground. *Geosci. Model Dev.*, **12 (7)**, 2797–2809.

Scher, S., and G. Messori, 2021: Ensemble methods for neural network-based weather forecasts. *Journal of Advances in Modeling Earth Systems*, **13 (2)**, doi:https://doi.org/10.1029/2020MS002331.

Schneider, T., S. Lan, A. Stuart, and J. Teixeira, 2017: Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, **44 (24)**, 12,396–12,417.

Schraff, C., H. Reich, A. Rhodin, A. Schomburg, K. Stephan, A. Periáñez, and R. Potthast, 2016: Kilometre-scale ensemble data assimilation for the cosmo model (KENDA). *Quarterly Journal of the Royal Meteorological Society*, **142 (696)**, 1453–1472.

Schulzweida, U., 2022: Cdo user guide. Zenodo, doi:10.5281/zenodo.7112925.

Sivashinsky, G. I., 1977: Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations. *Acta Astronautica*, **4 (11)**, 1177–1206, doi:https://doi.org/10.1016/0094-5765(77)90096-0.

Stensrud, D. J., 2007: *Parameterization Schemes: Keys to Understanding Numerical Weather Prediction Models*. Cambridge University Press, Cambridge, UK.

Stephens, G. L., and Coauthors, 2010: Dreary state of precipitation in global models. *Journal of Geophysical Research: Atmospheres*, **115 (D24)**, doi:https://doi.org/10.1029/2010JD014532, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010JD014532.

Szunyogh, I., 2014: *Applicable Atmospheric Dynamics: Techniques for the Exploration of Atmospheric Dynamics*. Applicable Atmospheric Dynamics, doi:10.1142/8047.

Szunyogh, I., E. J. Kostelich, G. Gyarmati, E. Kalnay, B. R. Hunt, E. Ott, E. Satterfield, and J. A. Yorke, 2008: A local ensemble transform Kalman filter data assimilation system for the NCEP global model. *Tellus*, **60 (A)**, 113–130.

Taylor, J., and M. Feng, 2022: A deep learning model for forecasting global monthly mean sea surface temperature anomalies. arXiv, doi:10.48550/ARXIV.2202.09967.

Tikhonov, A. N., and V. I. Arsenin, 1977: Solutions of ill-posed problems.

Torrence, C., and G. P. Compo, 1998: A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, **79 (1)**, 61–78, doi:10.1175/1520-0477(1998)079<0061:Apgtwa>2.0.Co;2.

Trenberth, K. E., and L. Smith, 2005: The mass of the atmosphere: A constraint on global analyses. *Journal of Climate*, **18 (6)**, 864–875, doi:10.1175/jcli-3299.1.

Vaittinada Ayar, P., M. Vrac, and A. Mailhot, 2021: Ensemble bias correction of climate simulations: preserving internal variability. *Scientific Reports*, **11 (1)**, 3098, doi:10.1038/s41598-021-82715-1.

Walleshauser, B., and E. Bollt, 2022: Predicting sea surface temperatures with coupled reservoir computers. *Nonlin. Processes Geophys.*, **29 (3)**, 255–264, doi:10.5194/npg-29-255-2022.

Watt-Meyer, O., N. D. Brenowitz, S. K. Clark, B. Henn, A. Kwa, J. McGibbon, W. A. Perkins, and C. S. Bretherton, 2021: Correcting weather and climate models by machine learning nudged historical simulations. *Geophysical Research Letters*, **48 (15)**, e2021GL092 555, doi:https://doi.org/10.1029/2021GL092555.

Weyn, J. A., D. R. Durran, and R. Caruana, 2019: Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. *Journal of Advances in Modeling Earth Systems*, **11 (8)**, 2680–2693, doi:https://doi.org/10.1029/2019MS001705.

Weyn, J. A., D. R. Durran, and R. Caruana, 2020: Improving data-driven global weather prediction

using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, **12**, e2020MS002 109.

Weyn, J. A., D. R. Durran, R. Caruana, and N. Cresswell-Clay, 2021: Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, **13 (7)**, e2021MS002 502, doi:https://doi.org/10.1029/2021MS002502.

Wikner, A., J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott, 2020: Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems. *Chaos*, **30 (5)**, 053 111.

Wikner, A., J. Pathak, B. R. Hunt, I. Szunyogh, M. Girvan, and E. Ott, 2021: Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **31 (5)**, 053 114.

Xiao, C., and Coauthors, 2019: A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data. *Environmental Modeling & Software*, **120**, 104 502, doi:https://doi.org/10.1016/j.envsoft.2019.104502.

Yuval, J., and P. A. O'Gorman, 2020: Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, **11 (1)**, 3295, doi:10.1038/s41467-020-17142-3.

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

Zhu, Y., R.-H. Zhang, and J. Sun, 2020: North pacific upper-ocean cold temperature biases in cmip6 simulations and the role of regional vertical mixing. *Journal of Climate*, **33 (17)**, 7523–7538, doi:10.1175/jcli-d-19-0654.1.

Zimin, A. V., I. Szunyogh, D. J. Patil, B. R. Hunt, and E. Ott, 2003: Extracting envelopes of rossby wave packets. *Monthly Weather Review*, **131 (5)**, 1011–1017, doi:10.1175/1520-0493(2003)131<1011:Eeorwp>2.0.Co;2.