UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Scaling Across Multiple Dimensions to Accelerate Subgrid Machine Learning Parameterization Development

Permalink

https://escholarship.org/uc/item/8279t561

ISBN

9798291546208

Author

Lin, Jerry

Publication Date

2025-08-22

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Scaling Across Multiple Dimensions to Accelerate Subgrid Machine Learning Parameterization Development

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Earth System Science

by

Jerry Lin

Dissertation Committee:
Professor Mike Pritchard, Co-Chair
Professor Francois Primeau, Co-Chair
Professor Gudrun Magnusdottir

2025

Chapter 2 © 2025 American Geophysical Union
All other materials © 2025 Jerry Lin

DEDICATION

To my mom and my oldest sister, whose endless encouragement helped me persist during periods of great personal uncertainty.

And to my late father, who would have been immensely proud to see me finish a PhD.

TABLE OF CONTENTS

Page

LIST OF FIGURES v

LIST OF TABLES vii

ACKNOWLEDGEMENTS viii

VITA x

ABSTRACT OF THE DISSERTATION xiv

1 Introduction 1
1.1 Breaking Deadlock and Superparameterization 1
1.2 The Promise of Full-Physics Emulation for Subgrid Parameterization 2
1.2.1 A Host of Competing Approaches 3
1.3 Scope and Organization of the Dissertation 6

2 Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles 8
2.1 Introduction 10
2.2 Methods 12
2.2.1 Reference Climate Simulation 12
2.2.2 Training, Validation, and Offline Test Data 13
2.2.3 End-to-End Pipeline and Analysis 14
2.2.4 NN Configurations 15
2.3 Results (in-distribution) 21
2.3.1 Offline Results 21
2.3.2 Online Results 23
2.3.3 Persistent Online Zonal Mean Biases 28
2.4 Estimating necessary sample size for robust detection 30
2.5 Conclusion 31
2.6 Open Research 35
2.7 Acknowledgments 36

3 Crowdsourcing ML Parameterization Innovation with a Seminal Dataset, Benchmark, and Competition 37
3.1 Introduction 38
3.2 Methods 40
3.2.1 ClimSim NeurIPS Datasets and Benchmarks Paper 40
3.2.2 LEAP - Atmospheric Physics Using AI (ClimSim) Kaggle competition 44
3.3 Impact 48
3.3.1 Reception 48

3.3.2 Follow-Up Work and The Road Ahead 48
3.4 Conclusion 49

4 Harvesting ideas for hybrid physics-ML climate simulation from a $50,000 Kaggle Competition 50
4.1 Introduction 52
4.2 Methods 54
4.2.1 Architectures 55
4.2.2 Architecture-Agnostic Configurations (AAC) 61
4.3 Results 63
4.3.1 Offline R^2 Evaluation 63
4.3.2 A new state-of-the-art in online zonal mean biases 65
4.3.3 Common online pathologies 65
4.3.4 Universal offline biases 68
4.4 Discussion and Conclusion 70

5 Conclusion 72
5.1 Summary of the work 72
5.2 Discussion and Outlook for the Future 73

Supplementary Information 75

References 104

LIST OF FIGURES

Page

Figure 2.1 This shows a diagram of ClimScale, the semi-automated, end-to-end pipeline for training, online coupling, and evaluation. 12

Figure 2.2 This shows offline test Root Mean Squared Error (RMSE) for subgrid heating (1a,1c,1e) and moistening (1b,1d,1f) tendencies across configurations are plotted against validation error rank. 1a and 1b show RMSE for average, multiple linear regression (MLR), and martingale baselines. The MLR baseline makes use of inputs and outputs from the standard configuration. 1e and 1f show configurations with statistically distinct average RMSE, and 1c and 1d show the others. Figure S2 in the SI presents an analogous figure using a multiclimate test set. 22

Figure 2.3 This shows histograms of online temperature RMSE in K for each configuration compared against that of the standard configuration are shown here. Only models that integrated for the entire simulation duration (i.e., did not crash while integrating a full simulation year) are shown, and number of surviving models per configuration is shown in the legend. Vertical lines correspond to ensemble-median online temperature RMSE. Bin-width = $\frac{10}{49}K \approx .204$ K. 24

Figure 2.4 This shows histograms of online moisture RMSE in g/kg for each configuration compared against that of the standard configuration are shown here. Only models that integrated for the entire simulation duration (i.e., did not crash while integrating a full simulation year) are shown, and number of surviving models per configuration is shown in the legend. Vertical lines correspond to ensemble-median online moisture RMSE. Bin-width = $\frac{2}{49} \approx .0408$ g/kg. 25

Figure 2.5 This shows shows zonal mean temperature biases for the top five hybrid models across the standard, specific, no dropout, and multiclimate configurations. Models to the left have lower online error. 29

Figure 4.1 This figure depicts a simplified diagram for the U-Net architecture used in Hu et al. (2024) 56

Figure 4.2 This figure depicts a simplified diagram for the Squeezeformer architecture used in the 2024 ClimSim Kaggle Competition 56

Figure 4.3 This figure depicts a simplified diagram for the Pure ResLSTM architecture used in the 2024 ClimSim Kaggle Competition 58

Figure 4.4 This figure depicts a simplified diagram for the Pao Model architecture used in the 2024 ClimSim Kaggle Competition 58

Figure 4.5 This figure depicts a simplified diagram for the ConvNeXt architecture used in the 2024 ClimSim Kaggle Competition 59

Figure 4.6 This figure depicts a simplified diagram for the Encoder-Decoder LSTM architecture used in the 2024 ClimSim Kaggle Competition 60

Figure 4.7 This shows average R^2 values for each variable for the U-Net and winning architectures using the expanded variable list. 63

Figure 4.8 This shows SOTA online five year zonal mean biases for Encoder-Decoder LSTM trained with confidence loss (seed 7). 65

Figure 4.9 This shows precipitation histograms for Encoder-Decoder LSTM trained with confidence loss (seed 7). The purple box highlights areas where precipitation is underestimated. The U-Net refers to the results shown in Hu et al. 2025. 65

Figure 4.10 This figure shows online zonal mean temperature biases across architectures in the standard configuration. 66

Figure 4.11 This figure shows online zonal mean moisture biases across architectures in the standard configuration. 66

Figure 4.12 This figure shows online five year global means for models in the standard configuration for surface temperature (T_{59}), total precipitable water, and total cloud path. All models consistently underestimate total precipitable water, particularly in the tropics. 67

Figure 4.13 This figure shows offline zonal mean moistening biases for models in the standard configuration. 68

Figure 4.14 This figure shows offline zonal mean liquid cloud tendency biases for models in the standard configuration. 68

Figure 4.15 This figure shows offline zonal mean liquid cloud tendency biases for models in the standard configuration. 69

Figure 4.16 This figure shows offline zonal mean liquid and ice cloud tendency biases for the top two Kaggle teams. 69

LIST OF TABLES

Page

Table 2.1 This table details the common NN hyperparameter search space shared by all tested parameterization design configurations (excluding the no dropout configuration). All NNs tested here are fully-connected NNs. RAdam and QHAdam stand for "Rectified Adam" and "Quasi-hyperbolic Adam", respectively. Each hyperparameter is sampled uniformly at random, logarithmically in the case of learning rates. 17

Table 2.2 This table shows input and output variables, their units, and their number of vertical levels N_z for the standard configuration. $(t-1)$ refers to the value corresponding to the previous timestep. Variables indicated with an asterisk are removed or transformed in configurations corresponding to backward ablations. Stratospheric moisture (corresponding to the top five levels) is nearly always zero and excluded from training. 17

Table 2.3 This table shows summary offline and online results. The columns titled Survival, Online Error (K), Offline Error (K/day), and Spearman's ρ in Table 2.3 refer to online survival rate (i.e., percentage of runs that integrated the full simulation year without crashing), ensemble-median online temperature RMSE in K, ensemble-average offline heating RMSE in K/day, and Spearman correlation between offline heating and online temperature RMSE. Online error statistics and offline-online relationships exclude runs that crash. Excluding the column showing correlations, all rows following the first denote anomalies relative to the standard configuration's statistics. Significant anomaly values are bolded, and color indicates direction of difference (with red indicating worse and teal indicating better performance). Statistics are deemed significant at $\alpha = 0.05$ after applying a Benjamini-Hochberg correction, and Figure S1 in the SI shows this correction being applied. The analogous table for online moisture and online moistening error is similar and can be found in Table S1 in the SI. Table S2 in the SI shows similar statistics but on a multiclimate test set. 26

Table 4.1 This table details the list of input features used in the standard, confidence loss, and difference loss configurations. 55

Table 4.1 This table details the list of output features used by all models. 57

Table 4.3 This table details the list of additional input features used in the Expanded Inputs configuration. 64

ACKNOWLEDGEMENTS

This dissertation marks the culmination of several years of dedicated research, a journey that was made possible by a host of mentors and peers whose support, encouragement, and advice have fundamentally shaped the trajectory of my PhD.

First and foremost, I would like to express my deepest gratitude to my PhD advisor, Mike Pritchard. I consider myself extremely lucky that he decided to take a chance on me five years ago. I have grown immensely as his student, becoming a stronger writer, speaker, and scientist under his mentorship. He has opened doors that would have otherwise been out-of-reach and fostered collaborations with incredible mentors that I am learning from to this day.

These mentors include Sungduk Yu, Tom Beucler, Zeyuan Hu, and Liran Peng. Sungduk and Tom's level of involvement in my research effectively made them unofficial PhD co-advisors. Sungduk's technical mentorship and contributions were essential to the success of my first chapter, and Tom has generously continued to informally mentor me even after leaving the Pritchard group for a professorship at the University of Lausanne. While never have been formally part of our research group, Zeyuan has also been an incredible collaborator, setting a new bar for hybrid physics-ML climate modeling during his NVIDIA internship, helping manage a $50,000 Kaggle Competition with me, and providing continued technical guidance for my final chapter. Finally, I wanted to extend my thanks to Liran for showing me how to navigate huge Fortran codebases, something that was a frequent pain point for me in my research.

I would also like to thank the Learning the Earth with Artificial Intelligence and Physics (LEAP) NSF Science and Technology Center (STC) for funding my PhD, providing a community of likeminded researchers working at the frontier of machine learning for climate modeling, and hosting annual events at the Edith Macy Conference Center, where I got to interact with and learn from legendary scientists like Laure Zanna, Pierre Gentine, and Dave Lawrence. LEAP made ClimSim, the Kaggle competition, and my third chapter possible, and being a part of it was transformational for my PhD.

Outside of connections I have made as a direct consequence of being my advisor's student, I am also thankful for the friends I made in grad school and for the friends who stayed in contact with me from undergrad. These friendships were essential to not only helping me preserve my sanity during trying times but also fostering professional opportunities in their own right. I owe a special thank you to Margarita Geleta, who connected me with Ritwik Gupta and Bjorn Lutjens when the outcome of ClimSim was very uncertain, created the website for ICML workshop we co-organized, and helped me find a summer internship at an AI weather startup partnering with Lawrence Berkeley National Laboratory.

Finally, I owe a tremendous debt of gratitude to my mom and oldest sister, whose unwavering support carried me through my PhD journey. My mom has never stopped believing in me, even when I suggest to her that her faith is misplaced. And my oldest sister, who managed multiple family crises during this time, was instrumental in helping me reach the finish line.

The text of Chapter 1 is an adaptation of the material as it appears in Lin et al. (2025), used with permission from the Journal of Advances in Modeling Earth System (*JAMES*). The coauthors listed in this publication are Sungduk Yu, Liran Peng, Tom Beucler, Eliot Wong-Toi, Zeyuan Hu, Pierre Gentine, and Margarita Geleta. The full citation for this publication is:

> Lin, Jerry, Sungduk Yu, Liran Peng, Tom Beucler, Eliot Wong-Toi, Zeyuan Hu, Pierre Gentine, Margarita Geleta, and Mike Pritchard. 2025. "Navigating the Noise: Bringing Clarity to ML Parameterization Design with O(100) Ensembles." Journal of Advances in Modeling Earth Systems 17 (4): e2024MS004551.

Mike Pritchard directed and supervised the research detailed in this dissertation. My work was primarily funded by National Science Foundation (NSF) Science and Technology Center (STC) Learning the Earth with Artificial Intelligence and Physics (LEAP), Award # 2019625-STC. I also acknowledge co-funding from the U.S. Department of Energy (DOE) Advanced Scientific Computing Research (ASCR) program (DE-SC0022255). The work in Chapter 1 and part of Chapter 2 was made possible with the use of high-performance computing resources from Bridges2 at the Pittsburgh Supercomputing Center (PSC) through allocation ATM190002 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by NSF Grants 2138259, 2138286, 2138307, 2137603, and 2138296. The work resulting in Chapter 2 and all of Chapter 3 used high-performance computing resources from the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. Additional funding support for these resources came from the DOE Regional and Global Model Analysis (RGMA) program (DE-SC0023368).

VITA

Jerry Lin

Education

University of California, Irvine — Irvine, CA
PhD Earth System Science — Expected: 2025
MS Earth System Science — 2022

University of California, Berkeley — Berkeley, CA
BA Applied Mathematics — 2018

Experience

Computational Clouds and Climate Lab — Irvine, CA
Graduate Student Researcher — September 2020 – Present

- Currently working on implementing winning architectures from a 2024 Kaggle competition online in a hybrid physics-ML version of the E3SM-MMF climate model.
- Created an automated pipeline for running O(100) ensemble hybrid physics-ML climate simulations to navigate the noise of emergent, online performance, whose coupled dynamics are invisible to the optimizer during offline training.
- Core contributor (4th author) to the original ClimSim paper, which won the best paper award in the NeurIPS Datasets and Benchmarks track
- Wrote a proposal for, desgined, and ran a $50,000 Kaggle competition on the award-winning ClimSim dataset on behalf of LEAP.
  https://www.kaggle.com/competitions/leap-atmospheric-physics-ai-climsim

Lawrence Berkeley National Laboratory — June 2024 - September 2024

- As part of a joint partnership between Berkeley Lab and Vayuh.ai, I used gradient boosting and deep learning techniques to improve wind power forecasts for various wind power plants in Texas.

California Public Utilities Commission — Irvine, CA
Public Utilities Regulatory Analyst — January 2020 – August 2020

- Conducted hazard rate analysis for utility infrastructure that formed part of CPUC's counterproposal for Southern California Edison's proposed energy rate hikes

California Public Utilities Commission — Irvine, CA
Intern — May 2017 - August 2017

- Created a flexible machine learning pipeline for predicting which customers would be most at risk of defaulting on utility bills
- Co-led workshops to teach employees how to use R and take over projects I participated in after I left
- Created standards for future data submission from California utilities

National Bureau of Economic Research Berkeley, CA
Undergraduate Research Assistant May 2016 - August 2016

- Created a database of lawyers and law firms specializing in disability insurance by scraping the web

Publications

Lin, Jerry, Sungduk Yu, Liran Peng, Tom Beucler, Eliot Wong-Toi, Pierre Gentine, Margarita Geleta, and Mike Pritchard. 2024. "Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles." Published in *JAMES*.
https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2024MS004551?af=R

Hu, Zeyuan, Akshay Subramaniam, Zhiming Kuang, **Jerry Lin**, Sungduk Yu, Walter Hannah, Noah Brenowitz, Josh Romero, and Mike Pritchard "Stable Machine-Learning Parameterization of Subgrid Processes with Real Geography and Full-physics Emulation." Accepted at *JAMES*.
https://arxiv.org/abs/2407.00124

Yu, Sungduk, Zeyuan Hu, Akshay Subramaniam, Walter Hannah, Liran Peng, **Jerry Lin**, Mohamed Aziz Bhouri, et al. 2024. "ClimSim-Online: A Large Multi-scale Dataset and Framework for Hybrid ML-physics Climate Emulation" Accepted at *JMLR*.
https://arxiv.org/abs/2306.08754

Beucler, Tom, Pierre Gentine, Janni Yuval, Ankitesh Gupta, Liran Peng, **Jerry Lin**, Sungduk Yu, et al. 2024. "Climate-Invariant Machine Learning." Science Advances 10 (6): eadj7250.

Lin, Jerry, Mohamed Aziz Bhouri, Tom G. Beucler, Sungduk Yu, and Michael Pritchard. 2023. "Stress-Testing the Coupled Behavior of Hybrid Physics-Machine Learning Climate Simulations on an Unseen, Warmer Climate." In *Proceedings of the NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning,* edited by Climate Change AI.
https://www.climatechange.ai/papers/neurips2023/62/paper.pdf.

Yu, Sungduk, Walter Hannah, Liran Peng, **Jerry Lin**, Mohamed Aziz Bhouri, Ritwik Gupta, Björn Lütjens, et al. 2023. "ClimSim: A Large Multi-Scale Dataset for Hybrid Physics-ML Climate Emulation." In *Proceedings of the NeurIPS 2023 Datasets and Benchmarks Track.* https://openreview.net/pdf?id=W5If9P1xqO.

Talks and posters

Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles. Talk and poster presentation at Learning the Earth with Artificial Intelligence and Physics (LEAP) Year 4 Annual Meeting, October 18, 2024, Briarcliff Manor, New York.

Lin, Jerry, Eliot Wong-Toi, Stephan Mandt, Tom Beucler, Sungduk Yu, Mohamed Aziz Bhouri, Mike Pritchard. 2024. "Towards Well-Calibrated Stochastic Neural Network Convective Parameterizations." Oral presentation at the 23rd Conference on Artificial Intelligence for Environmental Science, part of the 104th AMS Annual Meeting, January 30, 2024, Baltimore, MD.

Lin, Jerry, Mohamed Aziz Bhouri, Tom G. Beucler, Sungduk Yu, and Michael Pritchard. 2023. "Stress-Testing the Coupled Behavior of Hybrid Physics-Machine Learning Climate Simulations on an Unseen, Warmer Climate." Poster presentation at the NeurIPS 2023 Climate Change AI Workshop, December 16, 2023, New Orleans, LA.

Lin, Jerry, Sungduk Yu, Tom Beucler, Pierre Gentine, David Walling, and Mike Pritchard. 2023. "Systematic Sampling and Validation of Machine Learning-Parameterizations in Climate Models." Poster presentation at the AGU23 meeting, December 14, 2023, San Francisco, CA.

Lin, Jerry, Sungduk Yu, Mike Pritchard, Tom Beucler, David Walling. 2023. "Testing learned parameterizations of subgrid climate model physics at industrial scale to confront a prognostic error dilemma." Poster presentation at Scientific Machine Learning Symposium, March 17, 2023, San Diego, CA.

Lin, Jerry, Sungduk Yu, Mike Pritchard, Tom Beucler, David Walling, Eliot Wong-Toi, Azar Umudov. "Two paths toward more principled neural network parameterizations in climate models." Lightning Talk at Learning the Earth with Artificial Intelligence and Physics (LEAP) Year 2 Annual Meeting, January 17, 2023, Briarcliff Manor, New York.

Lin, Jerry, Sungduk Yu, Mike Pritchard, Tom Beucler, David Walling. "Confronting the offline vs. online skill dilemma via prognostic testing of neural network convection parameterizations at a computationally ambitious scale" Virtual poster at the AGU22 meeting, December 17, 2022.

Leadership and service

UCI Earth System Science Professional Development Lead September 2023 - Present

- I co-host alumni speakers to talk about opportunities post-PhD and develop workshops for in-demand skills that typically are not taught during the course of a PhD (e.g. becoming comfortable with Git/GitHub)

2024 LEAP Summer Research Program Mentor May 2024 - August 2024

- Mentored a LEAP Momentum Fellow PhD student and two undergrads on creating stochastic neural network parameterizations for ClimSim.

2024 Machine Learning for Earth System Modeling (ML43SM) ICML Workshop Co-organizer July 26, 2024

- Conceived of and organized the first Machine Learning for Earth System Modeling (ML4ESM) Workshop at ICML.
- 33 submissions, 20 acceptances, 38 reviewers, roughly 40-50 in-person attendees in Vienna, Austria.

LEAP Parameterization Focus Group Lead January 2023 - May 2023

- I hosted speakers and led discussions on improving machine learning parameterizations of subgrid processes in climate models

Data Science Society at Berkeley DeCal Lead August 2017 - December 2018

- I managed a student-run data science course for 3 semesters at UC Berkeley.

Awards

Excellence in Leadership Award (UCI Earth System Science Department 2023)

Skills and interests

Technical: Python, Git/Github, xarray, numpy, pandas, Keras, PyTorch, bash
Language: English (native), Chinese (conversational), Russian (conversational)
Interests: e-biking, hiking, cooking, learning languages, playing with my sister's woolly Shiba Inu

ABSTRACT OF THE DISSERTATION

Scaling Across Multiple Dimensions to Accelerate Subgrid Machine Learning Parameterization Development

by

Jerry Lin

Doctor of Philosophy in Earth System Science

University of California, Irvine, 2025

Professor Mike Pritchard, Co-Chair

Professor Francois Primeau, Co-Chair

Subgrid processes in climate models are a dominant source of uncertainty in long-term climate projections. Due to the prohibitive computational expense of direct simulation, these processes are instead parameterized using hand-tuned approximations and assumptions that lead to systematic biases and uncertainty. In theory, Machine-Learning (ML) parameterizations offer a few attractive advantages, namely being able to emulate higher resolution simulations at a fraction of the cost and scaling in performance with more data. However, they also introduce substantial challenges of their own, ranging from the highly empirical nature of their design and evaluation to the emergent behavior they can exhibit when coupled to the dynamical core of a climate model (i.e. tested online). Left unsolved, these challenges threaten the prospect of breaking "deadlock" in convective parameterization development on a timescale in which humanity has not already experienced the worst effects of climate change.

The goal of this dissertation is to seed a fundamental step change in the progress of ML subgrid parameterization development by embracing scale on two seemingly orthogonal dimensions: online sampling and people. While these may seem like two unrelated directions, they confront the same central, existential problem: the solution space of possible ML

parameterizations is too vast, too chaotic, and too poorly understood for any one graduate student or research lab to efficiently explore alone. However, with large scale $\mathcal{O}(100)$ online sampling, signal can be detected from emergent noise with statistical confidence. And by opening up the ML parameterization problem to seasoned machine learning researchers and data scientists around the world with a standardized dataset, benchmark, software suite, and competition, identifying the neural network architectures and paradigms with the best chance of solving this problem suddenly becomes plausible.

Our results conclusively show that offline error, online error, and online stability do not necessarily improve together. We present reproducible and statistically robust cases in which online stability and online error trade off against each other (i.e. in the cases of changing the loss function or removing dropout). More encouragingly, we also identify design decisions that improve ML parameterizations on all fronts (e.g. when training on multiple climates or using batch normalization). After being exposed to multiple state-of-the-art architectures through a three-month competition open to the world, we achieve state-of-the-art online zonal mean biases and identify offline biases that transcend the choice of architecture, helping narrow down the root cause of online biases that have persisted through multiple independent research papers. Altogether, these results present necessary but not sufficient steps towards operational reliable hybrid physics-ML climate models. We hope that our methods can also serve as a blueprint for continued rapid progress.

Chapter 1

1 Introduction

1.1 Breaking Deadlock and Superparameterization

For decades, error bars in long-term climate projections for different emissions scenarios have been dominated by uncertainties in the representation of subgrid processes like convection, radiation, turbulence, and aerosol-cloud microphysics Sherwood et al. (2020); IPCC (2021). In particular, our inability to explicitly resolve the updrafts that carry water upward from the planetary boundary layer to form cumulus and stratocumulus clouds over tropical oceans is responsible for much the spread in our estimates for how much the global mean surface temperature will respond to anthropogenic greenhouse gas emissions. A greater composition of low, thick stratiform clouds will reflect and emit more radiation back into space, dampening global warming and giving civilization more time to mend its ways. If these clouds shrink, however, that would result in the opposite effect, accelerating the rate at which the planet warms and punishing humanity for its sins Schneider et al. (2017); Meehl et al. (2020).

Addressing this issue by increasing resolution would necessitate a massive increase in the horizontal resolution of contemporary climate models—from an effective grid spacing of 50 kilometers down to just 10-100 meters. This task lies orders of magnitude beyond the current and foreseeable capabilities of computing hardware, rendering it infeasible for decades to come Schneider et al. (2017). Instead, progress has been defined by slow, painstaking improvements to parameterizations that have been described as "rough caricatures" of reality. This progress is far too slow given the urgency and speed of climate change, leading experts like David Randall to coin the term "deadlock" for the state of affairs regarding convective parameterization development Randall et al. (2003); Randall (2013). Although Global Cloud-Resolving Models (GCRMs) with resolutions approaching O(1) kilometer are being developed and resolve many of the issues associated with conventional convective parame-

terizations (despite still being unable to resolve low clouds), they remain far too expensive for actionable long-term climate projections, which require ensembles of multidecadal simulations with different levels of anthropogenic forcing to establish appropriate counterfactuals Taylor et al. (2023); Hohenegger et al. (2023).

An alternative approach pursued in the United States uses superparameterization (also known as Multiscale Modeling Framework or MMF), a process in which a Cloud-Resolving Model (CRM) with doubly periodic boundary conditions is embedded inside each grid cell of a climate model. These CRMs run at a much finer spatiotemporal resolution than their host climate model and more explicitly resolve the subgrid physics, leaving the dynamics for the dynamical core. Thanks to the clean scale separation between these CRMs and the host climate model, higher-resolution subgrid physics can be simulated in parallel, resulting in a climate simulation that is both faster and more cost-effective than one from a GCRM Khairoutdinov et al. (2005); Randall (2013). This approach is not perfect, as the scale-separation and often 2D nature of the CRM are unnatural artifacts designed for computational efficiency. However, the resolved convective processes in this approach provide several benefits, including improvements to the representation of precipitation extremes, equatorial sea surface temperature (SST) cold tongue structure, and periodicity of the El Niño-Southern Oscillation Stan et al. (2010); Li et al. (2012); Randall (2013); Woelfle et al. (2018). Unfortunately, MMF is still too expensive to be used for long term climate projections Gentine et al. (2018).

1.2 The Promise of Full-Physics Emulation for Subgrid Parameterization

Nevertheless, the clean-scale separation of MMF yields another attractive (albeit unintended) property—the input-output pairs from the embedded CRMs can be used to train machine learning emulators that, in principle, are capable of emulating the coarsened effects of CRMs at a fraction of the cost Gentine et al. (2018); Rasp et al. (2018); Mooers et al. (2021).

In particular, deep learning methods have demonstrated incredible ability to model highly nonlinear patterns in data, starting with image classification in the 2010s and now language, code, and reasoning in the 2020s Krizhevsky et al. (2012); Vaswani et al. (2017); Brown et al. (2020). Although training neural networks on Graphical Processing Units (GPUs) can be computationally expensive for modern, industrial applications like Large Language Models and Video Diffusion Models, the extremely low-dimensional nature of the parameterization problem (technically 1D $\rightarrow$ 1D regression) makes training a trivial computational expense compared to MMF simulation. In addition, once trained, these neural networks are incredibly fast to inference (i.e. use). Instead of integrating partial differential equations across many timesteps at high resolution in a CRM domain, the equivalent operation for a neural network would be a single "forward pass." In a dense, feedforward neural network (NN), a forward pass simply consists of a combination of matrix multiplications and nonlinear "activation functions," something that can be done extremely quickly and cheaply on GPUs. For more modern neural network architectures, a forward pass would entail more complex mechanisms, but the inference times remain low in comparison to MMF simulation.

1.2.1 A Host of Competing Approaches

Early successful results in reduced-complexity aquaplanet configurations like Rasp et al. (2018) and Ott et al. (2020) were able to make use of simple dense, feedforward neural network architectures with minimal modifications beyond hyperparameter tuning. However, further progress was stymied by inconsistencies in the relationship between how these ML parameterization perform offline (i.e. on a fixed test set) and how they perform online (i.e. when coupled inside a climate model). ML parameterizations with superior offline fits could encounter emergent failure modes online, like unphysical drifts in error or numerical instabilities that prematurely terminated the hybrid simulation. There are many plausible and non mutually exclusive hypotheses explaining the gap between offline and online performance, and these hypotheses inspired a host of competing approaches.

For Rasp (2020), the missing ingredient preventing reliably stable and accurate hybrid simulations was online coupled learning. Because online errors can cause a distribution shift away from the inputs seen in the training data that compounds across timesteps, Rasp (2020) hypothesized that the only way to prevent them would be to train the neural network while it was coupled to the host climate model. Eventually, this line of thinking led to the development of NeuralGCM, a fully differentiable hybrid physics-ML atmospheric model created by Google Research that allowed errors from the subgrid ML parameterization to be backpropagated through the dynamical core such that the ML parameterization could be optimized across multiple timesteps Kochkov et al. (2024). This landmark achievement could indeed foretell the ultimate future of climate models. However, completely rewriting existing climate models composed of millions of lines of Fortran code in Python to enable end-to-end differentiability would be a monumental software engineering challenge that is currently beyond the resources of existing climate modeling labs. For ML parameterizations to have near-term impact on the next generation of climate models, a different approach may ultimately be necessary.

For Yuval et al. (2021), proper ML parameterization meant carefully separating the effects of different subgrid processes, emulating them separately, and diagnosing intermediate and prognostic variables from upstream emulated outputs whenever possible. Their strategy results in a stable and accurate hybrid aquaplanet simulation. However, it also requires meticulous engineering, and, more importantly, sacrifices the ability to coarse grain in time and space, negating some of the efficiency benefits of emulating the coarsened effects of a CRM in a single forward pass.

Others pointed to more fundamental issues with machine learning models. Because neural networks do not intrinsically conserve mass and energy, Beucler et al. (2021) developed methods to enforce analytic constraints to within machine precision. However, evaluating the impact of analytically constrained neural networks online has yet to be tried (at least at the time of this writing). Another potential weakness is that the skill of these data-

driven methods can rapidly deteriorate when extrapolating outside of the distribution of the data they were trained on. To mitigate this issue, Beucler et al. (2024) created "climate-invariant" feature transformations that transformed the inputs to stay "in-distribution" even in an unseen +4K climate.

Other hypotheses for ways to bridge the offline-online gap or improve ML parameterizations in general emerged in the years following the earliest successful prognostic runs, ranging from introducing stochasticity in order to account for variability in the subgrid scales to adopting techniques inspired by reinforcement learning (RL) like experience replay Gagne et al. (2020); Behrens et al. (2024); Christensen et al. (2024); Chen et al. (2025). However, some of the biggest breakthroughs in stable and accurate online simulation with real geography came from adopting a combination of strategies, including the use of more sophisticated architectures, ensemble-averaging, and integrating a microphysics constraint to diagnose liquid and ice cloud tendencies Wang et al. (2022); Han et al. (2023); Hu et al. (2024). While describing every proposed approach in detail is out-of-scope for this dissertation and better suited for a dedicated review paper, it is clear that a diverse set of strategies have been applied toward solving this problem. Nevertheless, prior to the work outlined in this dissertation, several critical impediments to progress remained. The emergent online behavior of these ML parameterizations was rarely sampled at scale in prior work, making reproducible progress and generalizable insights difficult. Additionally, machine learning researchers and data scientists in other fields had little exposure to this problem, sharply limiting the degree to which the blistering pace of machine learning research could lead to breakthroughs in ML parameterizations for climate models. In the end, these obstacles highlight a deficiency in scale across two distinct dimensions: models (specifically, prognostic sampling) and people (specifically, machine learning researchers).

1.3 Scope and Organization of the Dissertation

The work detailed in this dissertation represents a series of steps toward rectifying deficiencies in scale in both dimensions. In Chapter 2, I develop a modular and semi-automated end-to-end pipeline for running $\mathcal{O}(100)$ size ensembles of hybrid simulations. Using this tool, I detect signal from emergent online noise, allowing me to reach conclusions about what is necessary for stable and accurate hybrid simulation with statistical confidence. I find that certain decisions, like using memory, provide unambiguous benefits offline and online. Others, like training on multiple climates using different prescribed Sea Surface Temperatures (SSTs), can have little effect offline but improve both online error and online stability. Finally, offline error, online error, and online stability do not necessarily improve together since higher online stability can trade off against higher online stability. This chapter is published in *Journal of Advances in Modeling Earth Systems (JAMES)*:

Lin, Jerry, Sungduk Yu, Liran Peng, Tom Beucler, Eliot Wong-Toi, Zeyuan Hu, Pierre Gentine, Margarita Geleta, and Mike Pritchard. 2025. "Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles." Journal of Advances in Modeling Earth Systems 17 (4): e2024MS004551.

I published related work in the NeurIPS Climate Change AI Workshop Lin et al. (2024a). However, that work is omitted from this dissertation because it contains a known coupling bug that was fixed in Lin et al. (2025).

In Chapter 3, I summarize results of ClimSim, an award-winning NeurIPS Datasets and Benchmarks paper I made major contributions to, and the associated Kaggle competition, which I organized in the summer of 2024. My contributions to ClimSim included standardizing and streamlining preprocessing, training, and evaluation with a `data_utils.py` script, creating a "quickstart" notebook that allows newcomers to easily test new models offline, and fixing a critical bug in the Continuous Ranked Probability Score (CRPS) metric. The changes I made to the training pipeline (ditching the data loader and training on data that

fit on GPU) shortened training time from days to hours without sacrificing the number of training samples. It is almost certain that the paper would not have been completed if this change did not take place. In order to maximize the benefit for the interested reader, this chapter highlights lessons learned that are not apparent from simply reading the paper or going through the competition website.

In Chapter 4, I discuss results from implementing competition-winning architectures online inside E3SM-MMF. I reveal state-of-the-art results for zonal mean biases, but improvements across the board remain to be seen. Similarities in the offline and online pathologies give strong hints for what work is likely necessary to mitigate remaining online biases. This work is still in progress, and the plan is to submit and publish a more refined version of the listed results in the near future.

Chapter 5 presents a summary of the dissertation's findings along with discussion about the direction of future research. The dissertation concludes with a complete reference list and an appendix containing supplementary figures.

Chapter 2

2 Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles

As appears in:

Lin, Jerry, Sungduk Yu, Liran Peng, Tom Beucler, Eliot Wong-Toi, Zeyuan Hu, Pierre Gentine, Margarita Geleta, and Mike Pritchard. 2025. "Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles." Journal of Advances in Modeling Earth Systems 17 (4): e2024MS004551.

Keypoints

- Reducing offline error can have differing effects on online error and online stability, which do not necessarily improve together.
- Using memory, batch normalization, and training on multiple climates improve both online error and stability.
- Ensemble sizes of $\mathcal{O}(100)$ may be necessary to detect causally relevant differences in the online performance of ML parameterizations.

Abstract

Machine-learning (ML) parameterizations of subgrid processes (here of turbulence, convection, and radiation) may one day replace conventional parameterizations by emulating high-resolution physics without the cost of explicit simulation. However, uncertainty about the relationship between offline and online performance (i.e., when integrated with a large-scale general circulation model (GCM)) hinders their development. Much of this uncertainty

stems from limited sampling of the noisy, emergent effects of upstream ML design decisions on downstream online hybrid simulation. Our work rectifies the sampling issue via the construction of a semi-automated, end-to-end pipeline for $\mathcal{O}(100)$ size ensembles of hybrid simulations, revealing important nuances in how systematic reductions in offline error manifest in changes to online error and online stability. For example, removing dropout and switching from a Mean Squared Error (MSE) to a Mean Absolute Error (MAE) loss both reduce offline error, but they have opposite effects on online error and online stability. Other design decisions, like incorporating memory, converting moisture input from specific humidity to relative humidity, using batch normalization, and training on multiple climates do not come with any such compromises. Finally, we show that ensemble sizes of $\mathcal{O}(100)$ may be necessary to reliably detect causally relevant differences online. By enabling rapid online experimentation at scale, we can empirically settle debates regarding subgrid ML parameterization design that would have otherwise remained unresolved in the noise.

Plain Language Summary

Running high-resolution simulations of convection for climate forecasts takes a lot of computing power, which is why we use simpler and less accurate approximations for processes inside each grid cell of a climate model. Neural networks could mimic more accurate "smaller-scale" subgrid simulations at a fraction of the computational cost, but they can also behave unpredictably when they are actually used inside the "larger-scale" climate model (i.e., in an "online" setting). To account for this unpredictability and understand how to make improvements with statistical confidence, we conduct thousands of "online" experiments. We would like to reduce the online error of these models and improve their stability (i.e., preventing situations in which they crash and stop running). We found that using data from different climates and a previous time step helps with both goals, converting from specific humidity to relative humidity improves stability, and removing dropout reduces error at

the cost of stability. With all this testing, we're helping figure out how to develop neural networks that can work more reliably inside climate models, moving us closer to using them in actual climate projections.

2.1 Introduction

Despite the fact that they occur at scales smaller than the grid cell resolution of a climate model, subgrid processes (e.g. turbulence, deep convection, microphysics) can have major effects on model accuracy and fidelity. Unfortunately, resolving small-scale processes explicitly is too computationally costly for long-term climate projections. This necessitates the use of sub-grid parameterizations, i.e., empirical representation of the impact of subgrid processes on the coarse grid, which are inevitably error-prone due to their phenomenological nature. For over two decades, neural networks (NNs) have held the promise of circumventing the large amounts of compute required to resolve fine scale processes in climate simulations by learning the coarse representation of subgrid processes from high-resolution data Chevallier et al. (1998); Krasnopolsky et al. (2013); Gentine et al. (2018); Rasp et al. (2018); Yu et al. (2023b). Of particular interest is the potential to replace parameterizations that crudely approximate highly nonlinear subgrid scale processes like ocean momentum transport Guillaumin and Zanna (2021), atmospheric momentum transport via small-scale gravity waves Mansfield and Sheshadri (2024), radiative transfer Cachay et al. (2021), and moist convection Gentine et al. (2018); Rasp et al. (2018); Brenowitz et al. (2020); Yuval et al. (2021); Wang et al. (2022); Iglesias-Suarez et al. (2024) to create hybrid physics-ML (i.e., physical model with embedded ML parameterizations) climate models. If NNs could be trained to reliably emulate the behavior of more explicit simulation of subgrid processes, climate models could be improved compared to traditional parameterizations without incurring the associated computational cost of high-resolution simulations. In the context of convection, this could mean breaking the parameterization "deadlock" as coined by Randall et al. (2003)—a situation in which slowing progress on traditional convective parameterizations has not kept

pace with the growing societal need to accurately represent the underlying subgrid physics in climate models. Randall (2013); Shepherd (2014); Gentine et al. (2018); IPCC (2021).

The vision of breaking deadlock with NN parameterizations has been challenged by difficulties ensuring their reliability when they are dynamically coupled to a host climate model (i.e., used online). NNs are typically first fit offline on some training data (typically high-resolution simulations) and then plugged online, where emergent failure modes are revealed. NNs have been shown to outperform conventional convective parameterizations when evaluated offline i.e., on test data from the cloud-resolving models (CRMs) they are trained to emulate Gentine et al. (2018); Han et al. (2020); Mooers et al. (2021). When coupled online, however, small offline imperfections can compound across timesteps, resulting in dramatic error growth and simulation crashes (i.e. cases where the coupled simulations prematurely terminate because of numerical instabilities) Brenowitz et al. (2020); Wang et al. (2022). Previous work has argued that skillful offline fits—which can be found via ample ML tools that exist to optimize it—are no guarantee for stable online performance Ott et al. (2020); Wang et al. (2022).

An overarching problem is that the coupled dynamics between the GCM and NN parameterization are invisible to the optimizer during offline training, and the downstream online effects are rarely sampled at the scale required to distinguish emergent signal from noise. We hypothesize that the main obstacle is the associated technical work: training thousands of ML parameterizations, adapting each to run within a climate model, and running multi-hundred member ensembles of hybrid physics-ML climate simulations can be technically challenging and computationally intensive. As a result, basic questions such as "what differences in online error can be statistically shown to originate from offline ML architecture and optimization choices?" are not empirically addressed with statistical confidence.

To answer these questions, we designed a modular, semi-automated, and end-to-end pipeline for rapidly sampling the coupled behavior of fully-connected NN emulators of moist convection using multi-hundred member ensembles. We use this capability to conduct an

ablation study of different design decisions for full-physics emulators of subgrid atmospheric processes (see Section 2.3 for details).

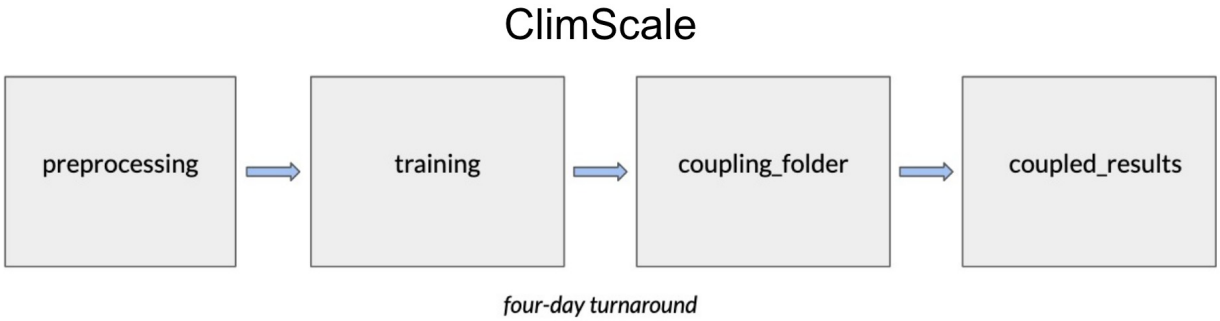

Fig. 2.1. This is a diagram showcasing ClimScale, our end-to-end pipeline for going from preprocessing to online results. Associated code can be found in our GitHub repository: https://github.com/SciPritchardLab/ClimScale.

Our level of online sampling is extensive and at least two orders of magnitude higher than what is traditionally shown in the literature Wang et al. (2022); Han et al. (2023); Iglesias-Suarez et al. (2024). We will show in the following sections that sampling online at a scale of hundreds of architecture trials leads to reproducible improvements to hybrid model performance that become statistically identifiable.

2.2 Methods

2.2.1 Reference Climate Simulation

The data used to train the NNs comes from the Super-Parameterized Community Atmosphere Model v3 (SPCAM 3) in an aquaplanet setting of intermediate complexity that has proved popular for exploring trade-offs in hybrid-ML simulation Gentine et al. (2018); Rasp et al. (2018); Ott et al. (2020); Behrens et al. (2022); Beucler et al. (2024). In superparameterization, each grid cell of a coarse-resolution general circulation model (GCM) contains an idealized two-dimensional CRM that explicitly resolves convection while making use of parameterizations for small-scale turbulence and cloud microphysics Khairoutdinov et al. (2005); Pritchard and Bretherton (2014); Jones et al. (2019). The NNs are trained to emulate subgrid heating (ΔT_{phy}) in K/s and moistening (ΔQ_{phy}) in kg/kg/s, where ΔT_{phy}

represents the combined effect of convective processes, radiation, and subgrid turbulence on heating. This is analogous to a traditional albeit multi-process parameterization. The GCM is a simplified, zonally-symmetric aquaplanet with a full diurnal cycle, fixed season (perpetual austral summer), prescribed sea surface temperatures (SSTs), and a nearly uniformly spaced grid with 64 points along the latitude and 128 points along the longitude dimensions. There are 30 vertical levels within each grid cell, and the GCM has a 30 minute timestep while embedded CRMs have a 20 second timestep. The embedded CRMs have a 4-km horizontal resolution and are each made up of 32 columns. Cloud condensate coupling is omitted from the framework for simplicity. Additional details can be found in the Supplementary Information (SI).

2.2.2 Training, Validation, and Offline Test Data

All NNs were trained on 12 months of longitudinally subsampled simulation data with 30-minute temporal resolution from SPCAM3 and prognostically evaluated for the same time period. Validation data consists of 2 months of simulation data temporally disjoint from the training data. Offline test data consists of 1 week of simulation data with no longitudinal subsampling that is temporally disjoint from both training and validation data. Extending test data volume by a factor of 2 or 4 changes the offline test RMSE by roughly 1 percent, as shown in Figure S28 in the SI. For completeness, we have also included offline results using a multiclimate test set, as shown in Table S2 and Figure S2 in the SI. However, to avoid confusion, we defer discussion of our offline multiclimate results to the Conclusion. Training and validation data retain every 8th and 7th longitude, respectively. This yields 17,927,168 samples for the training data, 3,556,800 samples for the validation data, and 2,752,512 samples for the offline test data. Training on multiple climates includes one extra sample in the data, but its impact is considered negligible. Inputs are normalized by subtracting the mean and dividing by the standard deviation. For vertically resolved variables, means and standard deviations are calculated at each vertical level. Outputted heating and moistening

tendencies are divided by the maximum of the standard deviation and 1×10^{-12} at each level. For training, all NNs were trained for a maximum of 180 epochs with a batch size of 4,096 and early stopping set at 5 epochs. Training, validation, and test data were saved to 32 bit precision so that data could fit entirely on GPU RAM, minimizing I/O induced slowdown.

2.2.3 End-to-End Pipeline and Analysis

Our end-to-end pipeline, shown in Figure 2.1, standardizes preprocessing, training, coupling, and analysis across all configurations and is available at `https://github.com/SciPritchardLab/ClimScale` Lin and Yu (2024). Training is parallelized across multiple GPUs using KerasTuner O'Malley et al. (2019), allowing for the efficient training of hundreds of fully-connected NNs in the span of days using a random search strategy. Coupling to SPCAM 3 is enabled by the Fortran-Keras Bridge (FKB) developed by Ott et al. (2020). All hybrid runs are initialized with the same initial condition file used for the reference SPCAM 3 simulation and are each programmed to run for one simulation-year. To mitigate storage demands, every 16th longitude in simulation output is automatically subsampled using GNU parallel Tange (2018) and NCO Zender (2008). Better offline performance is characterized by lower offline heating and moistening error on the held-out test set, and better online performance is characterized by lower online temperature and moisture error as well as higher online stability. Online error is only computed for runs that integrate for a full simulation-year without crashing, and hybrid simulations that do not crash within that timeframe are deemed as "stable." We detect differences in offline and online performance compared to the standard configuration with an array of statistical tests. For comparing average ensemble offline heating and moistening test root mean squared error (RMSE) between configurations, we use two-tailed, two-sample Welch's t-tests (using the `scipy.stats` library) Virtanen et al. (2020). For online ensemble survival rates (i.e., the percentage of hybrid model runs from a configuration that do not crash within one simulation-year) and ensemble-

median online temperature and moisture RMSE, we use two-tailed proportion tests and two-tailed permutation tests (using 10,000 permutations), respectively. Our decision to compare ensemble-median RMSE (and not ensemble-mean RMSE) for average ensemble online error is informed by our lack of meaningful online error statistics for crashed runs. Although not formally part of our ablation study, we also test the impact of sources of variation in our search space like dropout (among configurations that vary it), leak (in Leaky ReLu), learning rate, parameter count, choice of optimizer, and batch normalization on offline error, online error, and online stability. Details regarding these tests can be found in Figures S26 through S32 in the SI. Finally, to control for the false discovery rate, we apply a Benjamini-Hochberg correction to the unadjusted p-values before declaring statistical significance at a significance level of 5% Benjamini and Hochberg (1995). Additional details can be found in the SI while further details regarding ClimScale can be found in the Data Availability Statement.

2.2.4 NN Configurations

To empirically measure the online effects of various design decisions hypothesized to be critical in the literature Han et al. (2020, 2023); Clark et al. (2022); Bhouri et al. (2023); Beucler et al. (2024); Behrens et al. (2024), we conduct a modified ablation study for a "standard" configuration that bundles several claims regarding what would be necessary to improve the online behavior of NN convective parameterizations (detailed in Section 2.4.1). In our ablation study, we test two different kinds of modifications, which we will call "forward" and "backward" ablations. As with a traditional ablation study in ML, we test the downstream effect of removing a component of the standard (aka baseline) configuration to reveal the importance of the removed component. We call these "backward ablations". Our "forward ablations" correspond to configurations that add a feature we suspect could improve online performance.

Our naive hypothesis is that offline error, online error, and online stability improve together. Correspondingly, we expect the backward ablations to worsen offline and online

performance and vice versa.

Since small sample sizes and high variability in online error stemming from stochastic hyperparameter selection can mask differences between configurations, we train 330 NNs with randomly sampled hyperparameters for each of the nine configurations for a total of 2,970 NNs, which is $> 27.5\times$ that of Ott et al. (2020) and at least two orders of magnitude larger than what is traditionally shown in the literature Rasp et al. (2018); Wang et al. (2022); Iglesias-Suarez et al. (2024). Our hyperparameter search space, shown in Table 2.1, is modified from Ott et al. (2020) to reduce variability resulting from an excessively large, sparse, or suboptimal search space. We found that using a learning rate range that spanned multiple orders of magnitude was a dominant source of variation in validation error, so we narrowed the learning rate range to just one order of magnitude (1e-4 to 1e-3) and sampled it logarithmically. For similar reasons, we decided to sample all possible layer widths within tighter bounds instead of sampling a few possibilities separated by factors of two. Finally, since the Adam optimizer was shown to be dominant in Ott et al. (2020), we replaced the stochastic gradient descent (SGD) and RMSprop optimizers with Rectified Adam and Quasi-Hyperbolic Adam, which performed similarly to Adam in preliminary experiments and have been shown to be performant in other contexts Ma and Yarats (2018); Liu et al. (2019); Geleta et al. (2023). Put together, these interventions result in distributions of online temperature and moisture error that are concentrated within one order of magnitude (instead of two as in Ott et al. (2020)).

All nine configurations share a set of inputs similar to those used by Rasp et al. (2018) and Ott et al. (2020): vertically resolved coarse-scale temperature and humidity variables as well as scalars for surface pressure, top-of-atmosphere insolation, surface sensible heat flux, and surface latent heat flux. Output variables across all configurations are identical, consisting of vertically resolved heating and moistening tendencies. In all cases, the top five vertical levels (corresponding to anything approximately above 55 hPa) of moisture and moistening are not included during training and set to zero during coupling, in line with

Hyperparameter	Range
Hidden layers	⟦ 4, 11 ⟧
Nodes per layer	⟦ 200, 480 ⟧
Batch normalization	{On, Off}
Dropout	[0.0, 0.25]
Optimizer	{Adam, RAdam, QHAdam}
Leaky ReLu slope	[0.0, 0.4]
Learning rate	[1e-4, 1e-3]

Table 2.1: Common NN hyperparameter search space shared by all tested parameterization design configurations (excluding the no dropout configuration). All NNs tested here are fully-connected NNs. RAdam and QHAdam stand for "Rectified Adam" and "Quasi-hyperbolic Adam", respectively. Each hyperparameter is sampled uniformly at random, logarithmically in the case of learning rates.

findings from Clark et al. (2022) and Brenowitz et al. (2020) that negligible magnitudes and causatively irrelevant variability at these altitudes can inhibit the machine learning. NNs are not trained to directly output radiative fluxes and precipitation as those variables are diagnostic (uncoupled) in an aquaplanet setting (i.e., without land). The complete list of inputs and outputs for the standard configuration is shown in Table 2.2.

Input variables			**Output variables**		
Variable	**Unit**	N_z	**Variable**	**Unit**	N_z
Temperature	K	30	Heating rate ΔT_{phy}	K s^{-1}	30
Relative Humidity*	%	25	Moistening rate ΔQ_{phy}	kg kg^{-1} s^{-1}	25
$(t-1)$ Heating rate ΔT_{phy}*	K s^{-1}	30			
$(t-1)$ Moistening rate ΔQ_{phy}*	kg kg^{-1} s^{-1}	25			
Surface pressure	Pa	1			
Incoming solar radiation	W m^{-2}	1			
Sensible heat flux	W m^{-2}	1			
Latent heat flux	W m^{-2}	1			
Meridional wind*	m s^{-1}	30			
Ozone volume mixing ratio*	m^3 m^{-3}	30			
Cosine of zenith angle*		1			
Size of stacked vectors		175			55

Table 2.2: Table showing input and output variables, their units, and their number of vertical levels N_z for the standard configuration. $(t-1)$ refers to the value corresponding to the previous timestep. Variables indicated with an asterisk are removed or transformed in configurations corresponding to backward ablations. Stratospheric moisture (corresponding to the top five levels) is nearly always zero and excluded from training.

2.2.4.1 Standard Configuration [Baseline]

The **standard** configuration is our baseline against which other configurations are compared. Although there is some design similarity to early work of Rasp et al. (2018), additional heuristics like the use of a relative humidity transformation for moisture input Beucler et al. (2024), removing stratospheric moisture and moistening Brenowitz et al. (2020); Clark et al. (2022), including convective memory Han et al. (2020, 2023), expanding the input vector to include the mixing ratio of the volume of ozone and the cosine of the zenith angle, and normalizing the outputs by their standard deviation (per vertical level) are also included to provide for a skillful preliminary baseline. Several of these heuristics are empirically tested in the other configurations listed below. The complete list of inputs and outputs for this configuration is shown in Table 2.2.

2.2.4.2 Specific Humidity Configuration [Backward ablation]

The **specific humidity** configuration reverses the relative humidity transformation for moisture input in the standard configuration in order to see if the offline benefits of a "climate-invariant" transformation discovered in Beucler et al. (2024) also manifest in improved performance online. Beucler et al. (2024) identified multiple feature transformations that improve generalization across climates in an offline setting by decreasing the number of situations in which NNs extrapolate out-of-distribution. In the context of moisture, the marginal distribution of relative humidity changes very little in warmer climates (when measuring change using Hellinger distance) and is bounded by design, except in cases of supersaturation Beucler et al. (2024). While the NNs are tested online in the same climate, the implicit hypothesis is that online errors may include pathologies in which the ML-coupled fluid dynamics lead the input state vector out-of-distribution from the training set, such that climate-invariant feature transformations may be beneficial. Consequently, we expect a configuration that removes such a transformation to worsen offline and online performance relative to our baseline.

2.2.4.3 No Memory Configuration [Backward ablation]

The **no memory** configuration omits convective memory (i.e., heating and moistening tendencies from the previous

timestep) from the input. Including such memory was argued to be beneficial online in Han et al. (2023) and offline in Han et al. (2020) while similar benefits were shown using precipitation from a previous timestep in Behrens et al. (2024). In other work, Colin et al. (2019) and Shamekh et al. (2023) formally reveal the existence of a persistent "microstate memory" due to convective organization in the CRMs that our NNs with memory are trained to emulate. Thus we expect the configuration without memory to worsen offline and online performance relative to our baseline.

2.2.4.4 No Wind Configuration [Backward ablation] The **no wind** configuration excludes meridional wind speeds from the input, which was included in previous work from Rasp et al. (2018) and Ott et al. (2020) but argued to be of secondary importance by Han et al. (2020) and Mooers et al. (2021). Since this dimension of the horizontal wind field theoretically has the capacity to constrain the convection in SPCAM due to organizing effects of wind shear, we expect this configuration to worsen offline and online performance relative to our baseline.

2.2.4.5 No Ozone Configuration [Backward ablation] The **no ozone** configuration omits ozone mixing ratio from the input vector similar to previous work Rasp et al. (2018); Ott et al. (2020). Because ozone is causally relevant to radiative heating in the stratosphere and is prescribed as a function of latitude in the climate simulator, its omission in Ott et al. (2020) and Rasp et al. (2018) was a mistake. Consequently, we expect this configuration to worsen offline and online performance relative to our baseline.

2.2.4.6 No Zenith Angle Configuration [Backward ablation] The **no zenith angle** configuration omits the cosine of the zenith angle from the input vector similar to previous work Rasp et al. (2018); Ott et al. (2020). Calculating optical depth is a crucial part of radiative transfer models. Given that the zenith angle is a key factor in optical depth calculations, we expect omitting it to worsen online and offline performance.

2.2.4.7 Mean Absolute Error (MAE) Configuration [Forward ablation] The **MAE** configuration swaps out the Mean Squared Error (MSE) loss used for training in all other configurations with a Mean Absolute Error (MAE) loss. Compared to the MSE loss, the MAE loss is designed to learn the conditional median rather than the conditional mean, making it potentially less sensitive to outliers in the data as well as asymmetric distributions. In our context, our experimentation with the MAE loss is primarily motivated by the fact that the marginal distributions of different tendency variables appear to be more Laplacian than Gaussian. For Laplacian-distributed errors, an MAE loss is more appropriate Hodson (2022). Although we do not have access to the conditional distributions of the outputs since we do not have multiple realizations of the CRM given a single input, we suspect that they are similar to the marginal distributions. If this is the case, we expect a MAE loss to improve offline and online performance. The marginal distributions of the heating and moistening tendencies are shown in Figures S34 and S35 in the SI.

2.2.4.8 No Dropout Configuration [Forward ablation] The **no dropout** configuration removes dropout from the search space shown in Table 2.1. In many ML contexts, dropout is typically used as a regularization tool to prevent overfitting Hinton et al. (2012); Srivastava et al. (2014); Molina et al. (2021) but also has seen application as a computationally-efficient tool for representing model uncertainty Gal and Ghahramani (2015); Behrens et al. (2024). While Ott et al. (2020) did not find a strong relationship between dropout and online performance, Behrens et al. (2024) found that using even a modest amount of dropout (as small as .05) in the last hidden layer of a NN resulted in noticeable deterioration in offline skill. Given recent experience from Behrens et al. (2024), we expect eliminating dropout completely to improve both offline and online performance.

2.2.4.9 Multiclimate Configuration [Forward ablation] The **multiclimate** configuration is trained on three different climates, one of which is identical to that used for other configurations. The other two are created by prescribing -4K colder and +4K warmer SSTs

and waiting for the atmosphere to equilibrate. The number of training data samples used per climate is roughly 1/3 of the total number of training data samples used in the other configurations. Training on multiple climates is argued as helpful to reduce extrapolation error in prior work Clark et al. (2022); Bhouri et al. (2023); Lin et al. (2024a), and, in general, diversifying training data is a well-known and intuitive heuristic for improving the generalization performance of deep learning models. Intuitively, we expect training on multiple climates to improve offline and online performance.

2.3 Results (in-distribution)

2.3.1 Offline Results

To avoid confusion, we present our offline results in this section within the context of *in-distribution* offline error. Although differences emerge when using an out-of-distribution (i.e. multiclimate) offline test set, we defer the discussion of out-of-distribution differences to the Conclusion. Figure 2.2 tests our expectations for in-distribution offline error. The comprehensive sampling (330 NNs per configuration) shows offline test error values ranked by validation error for each configuration. We measure the offline effects of each backward and forward ablation via two-tailed Welch's t-tests on the differences in ensemble-average in-distribution offline heating and moistening errors for each ablation when compared to the standard configuration. For reference, we also show the offline test errors for a simple average, a multiple linear regression (MLR), and using the previous tendency values to predict the following one (i.e. making use of a martingale assumption).

Several of our expectations are confirmed. Among configurations corresponding to backward ablations, removing memory stands out as having the largest impact as it increases the ensemble-average offline RMSE substantially by 0.567 K/day (0.256 g/kg/day) for mean heating (moistening) (Figure 2.2 a,b). The offline impact of reformulating the moisture input from relative to specific humidity is less extreme yet nonetheless detectable, increasing ensemble-average offline RMSE by .0288 K/day (.00789 g/kg/day). In terms of forward ab-

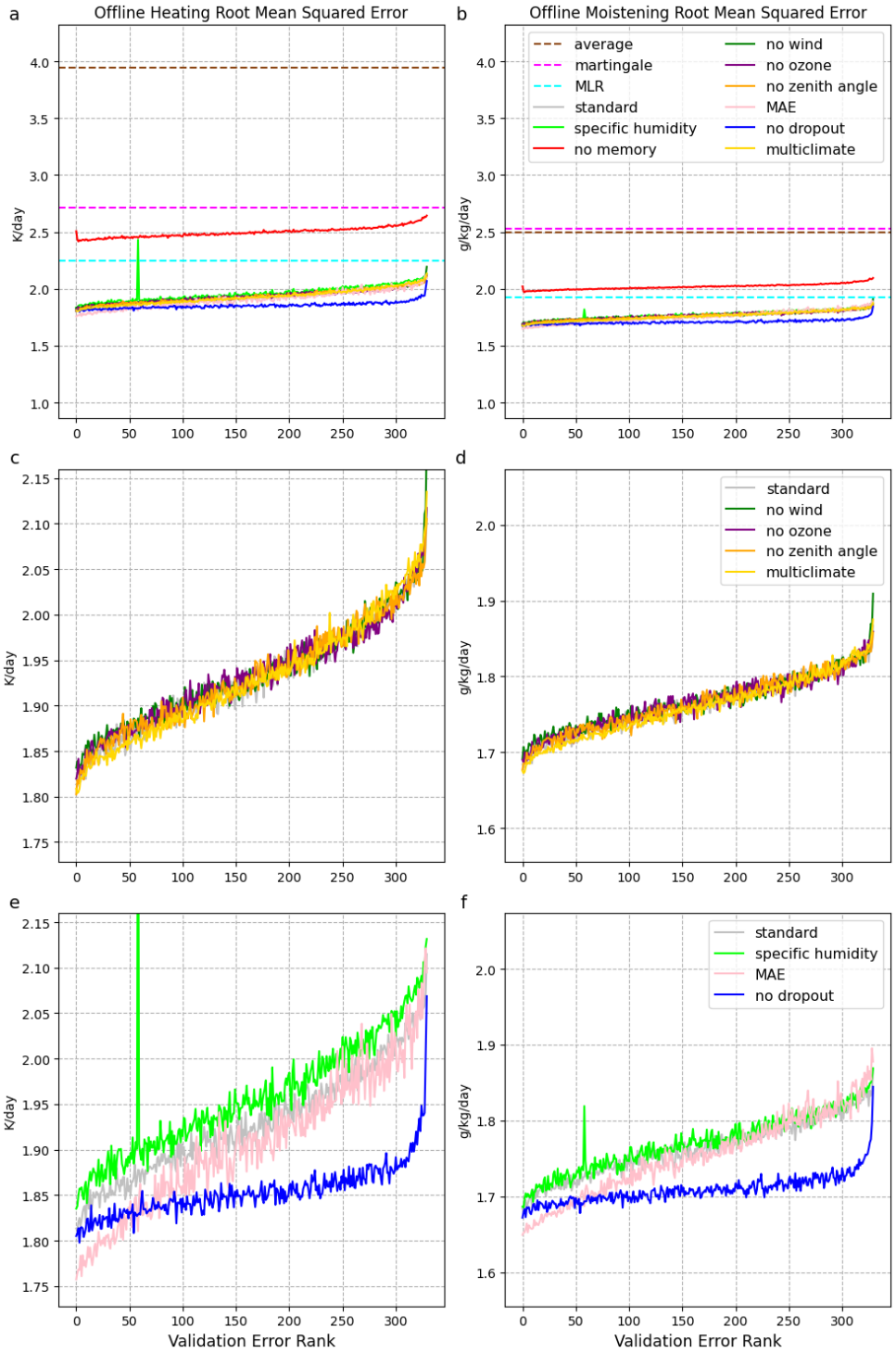

Fig. 2.2. Offline test RMSE for subgrid heating (1a,1c,1e) and moistening (1b,1d,1f) tendencies across configurations are plotted against validation error rank. 1a and 1b show RMSE for average, multiple linear regression (MLR), and martingale baselines. The MLR baseline makes use of inputs and outputs from the standard configuration. 1e and 1f show configurations with statistically distinct average RMSE, and 1c and 1d show the others. Figure S2 in the SI presents an analogous figure using a multiclimate test set.

lations, the MAE and no dropout configurations both show offline improvement, decreasing ensemble-average offline RMSE by .0226 K/day (.00952 g/kg/day) and .0765 K/day (.0545 g/kg/day), respectively. These differences are all statistically significant.

Other expectations are invalidated. Our backward ablations of removing wind, ozone, and zenith angle from the input vector did not statistically affect offline error (Figure 2.2 c,d). Our forward ablation of training on multiple climates also had no statistically detectable effect. Since Ott et al. (2020) and Wang et al. (2022) both showed a relationship between offline error and online performance (albeit solely in the form of online simulation stability), we might expect inter-configuration differences in online error (and online stability) to mirror differences identified offline. However, these relationships, especially with respect to offline and online error specifically, have not been statistically evaluated in previous works (e.g., Ott et al. (2020); Wang et al. (2022)).

If *in-distribution* offline error is predictive of *in-distribution* online performance in general, Figure 2.2 implies we should expect both online stability and online error to be worse for the no memory and specific configurations, statistically the same for the no wind, no ozone, no zenith angle, and multiclimate configurations, and better for the MAE and no dropout configurations. We explore this topic in the next section.

2.3.2 Online Results

We present online survival statistics, offline heating error, online temperature error, and Spearman rank correlations between offline heating and online error for each configuration in Table 2.3 (analogous statistics for offline moistening error and online moisture error can be found in Table S2 in the SI). Histograms of online temperature and moisture RMSE for simulations that do not crash can be found in Figure 2.3 and Figure 2.4, respectively. Our results immediately invalidate our naive hypothesis that offline error, online error, and online stability improve together. More specifically, we can see that the "MAE" and "no dropout" forward ablations have divergent effects on online error and online stability despite both

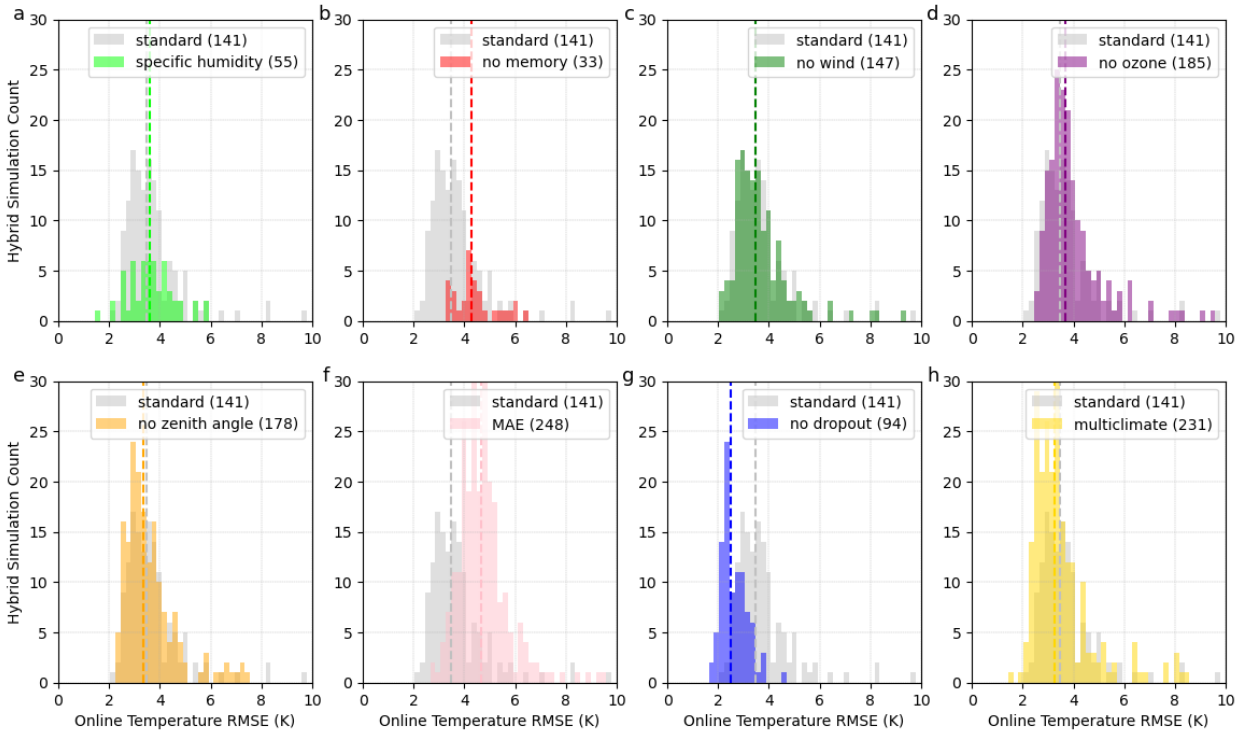

Fig. 2.3. Histograms of online temperature RMSE in K for each configuration compared against that of the standard configuration are shown here. Only models that integrated for the entire simulation duration (i.e., did not crash while integrating a full simulation year) are shown, and number of surviving models per configuration is shown in the legend. Vertical lines correspond to ensemble-median online temperature RMSE. Bin-width = $\frac{10}{49}K \approx .204$ K.

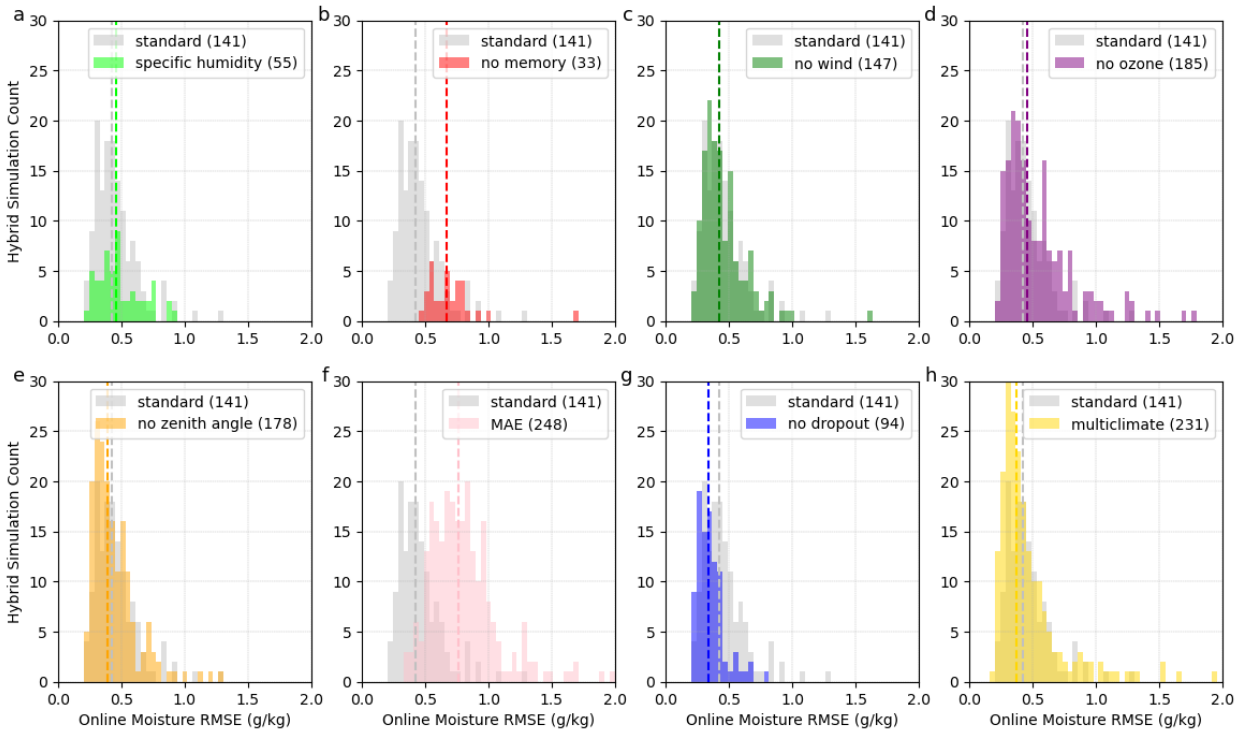

Fig. 2.4. Histograms of online moisture RMSE in g/kg for each configuration compared against that of the standard configuration are shown here. Only models that integrated for the entire simulation duration (i.e., did not crash while integrating a full simulation year) are shown, and number of surviving models per configuration is shown in the legend. Vertical lines correspond to ensemble-median online moisture RMSE. Bin-width = $\frac{2}{49} \approx .0408$ g/kg.

Configuration	Survival	Online Error (K)	Offline Error (K/day)	Spearman's ρ
standard	42.7%	3.48 K	1.93 K/day	**.855**
specific humidity	**-26.1%**	+.130 K	**+.0288 K/day**	**.834**
no memory	**-32.7%**	**+.800 K**	**+.567 K/day**	.311
no wind	+1.82%	-.00430 K	+.00526 K/day	**.789**
no ozone	**+13.3%**	**+.212 K**	+.00443 K/day	**.803**
no zenith angle	**+11.2%**	-.142 K	+.00226 K/day	**.829**
MAE	**+32.4%**	**+1.16 K**	**-.0226 K/day**	**.706**
no dropout	**-14.2%**	**-.967 K**	**-.0765 K/day**	**.353**
multiclimate	**+27.3%**	**-.237 K**	-.000903 K/day	**.858**

Table 2.3: The columns titled Survival, Online Error (K), Offline Error (K/day), and Spearman's ρ in Table 3 refer to online survival rate (i.e., percentage of runs that integrated the full simulation year without crashing), ensemble-median online temperature RMSE in K, ensemble-average offline heating RMSE in K/day, and Spearman correlation between offline heating and online temperature RMSE. Online error statistics and offline-online relationships exclude runs that crash. Excluding the column showing correlations, all rows following the first denote anomalies relative to the standard configuration's statistics. Significant anomaly values are bolded, and color indicates direction of difference (with red indicating worse and teal indicating better performance). Statistics are deemed significant at $\alpha = 0.05$ after applying a Benjamini-Hochberg correction, and Figure S1 in the SI shows this correction being applied. The analogous table for online moisture and online moistening error is similar and can be found in Table S1 in the SI. Table S2 in the SI shows similar statistics but on a multiclimate test set.

reducing offline error. Using an MAE, instead of MSE, loss results in the highest improvement in online stability but the worst deterioration in online error. At the same time, removing dropout grants the highest improvement to online error but comes at the cost of reduced online stability. In addition to disproving our naive hypothesis, this contradicts prior practice of relying on stability as a proxy for general online performance and disproves the notion that improving performance across one online metric implies simultaneous improvement on all others Ott et al. (2020); Gagne et al. (2020); Wang et al. (2022).

Other results are less surprising. The backward ablation of excluding memory unambiguously worsens offline and online performance across all metrics: offline error, online stability, and online error. Conversely, although the multiclimate configuration had no detectable offline impact, it is the only ablation to statistically improve upon the standard configuration on every online metric, as seen in Figures 2.3h and 2.4h. The no wind, no ozone, and no zenith angle configurations also had no detectable offline impact, but, the no ozone

configuration does have statistically higher ensemble-median online temperature RMSE, a reassuring expected finding. Despite the importance of zenith angle in calculating optical depth, we suspect it had less of an impact on the ensemble-median temperature RMSE because of a high R-squared between cosine of zenith angle and solar insolation ($> .99$ when values corresponding to zero solar insolation are masked out). Surprisingly, the no ozone and no zenith angle configurations both have improved stability. We speculate this is because certain input features are redundant and causal pruning may be necessary to avoid learning spurious correlations that fail to generalize outside the training data Iglesias-Suarez et al. (2024). As for the remaining backward ablation, the specific humidity configuration worsens stability but does not have a statistically detectable impact on online error. In Section 4, we provide further details regarding what sample size we believe may be necessary for robustly detecting causally relevant differences online.

When it comes to the relationship between offline and online error, rank correlations in Table 2.3 and Table S1 indicate that such a relationship exists for configurations with convective memory and non-zero dropout. However, when eliminating dropout entirely, the offline-online error relationship is weakened, but still significant, for offline heating and online temperature and no longer present for offline moistening and online moisture. For configurations that vary it, dropout is a dominant source of variation for both offline and online error, as seen in Figure S26. Other aspects of our search space also have notable impacts on offline error, online error, and online stability, shown in Figures S27 to S32. In particular, batch normalization presents benefits on all fronts, as seen in Figure S31 and Figure S32f. In Figures S28, S29, and S32c we also see some evidence that smaller learning rates improve offline heating error, online temperature error, and online stability while higher parameter counts reduce offline heating error, offline moistening error, and online temperature error. Finally, we find that the inclusion of the QHAdam optimizer in our search space was a mistake, as it has a detrimental effect on online stability when compared to the alternatives, as seen in Figure S32e.

To ensure the robustness of our conclusions, we investigate the possibility that our decision to exclude crashed simulations in our comparisons of online error may have biased our results. For the ablation study, this could happen in cases where simulations with high online error crash less frequently in one configuration than another, resulting in a higher ensemble-median online error than would have otherwise been recorded. Under such circumstances, we would expect a pairwise comparison of ensemble-median online errors for the standard configuration and some ablation to be different at the end of the simulation-year than at some point during the simulation where the high online errors of the (not yet) crashed simulations are still being recorded. However, as seen in Figure S13, intersections of the standard configuration's ensemble-median online error growth only occur with those of ablations whose online error we already consider to be statistically identical. We believe this is sufficient evidence to conclude that this potential for bias has not impacted the results of our ablation study. We conduct a similar check for our finding that batch normalization reduces online temperature and moisture error in Figure S33, and conclude that that result is also not impacted. Readers interested in more detail can find additional figures in the SI showing time-series plots of online error growth (S4 to S12), offline and online error scatterplots (S16 to S24), and histograms for the number of months integrated for each configuration (S15).

2.3.3 Persistent Online Zonal Mean Biases

Our online results show the effect of various ablations on aggregate statistics across $\mathcal{O}(100)$ size ensembles. However, a closer look at the best online simulations across different configurations reveals common patterns in zonal mean biases relative to the reference SPCAM simulation, with a cold bias in the stratosphere and a warm bias in the poles (as seen in Figure 2.5). Systematic biases in the stratosphere and near the poles have also been reported by others Wang et al. (2022); Iglesias-Suarez et al. (2024), and our results confirm that the biases experienced in our work are reproducible. However, we also show that certain interventions can change the structure of this bias. For example, training on multiple climates

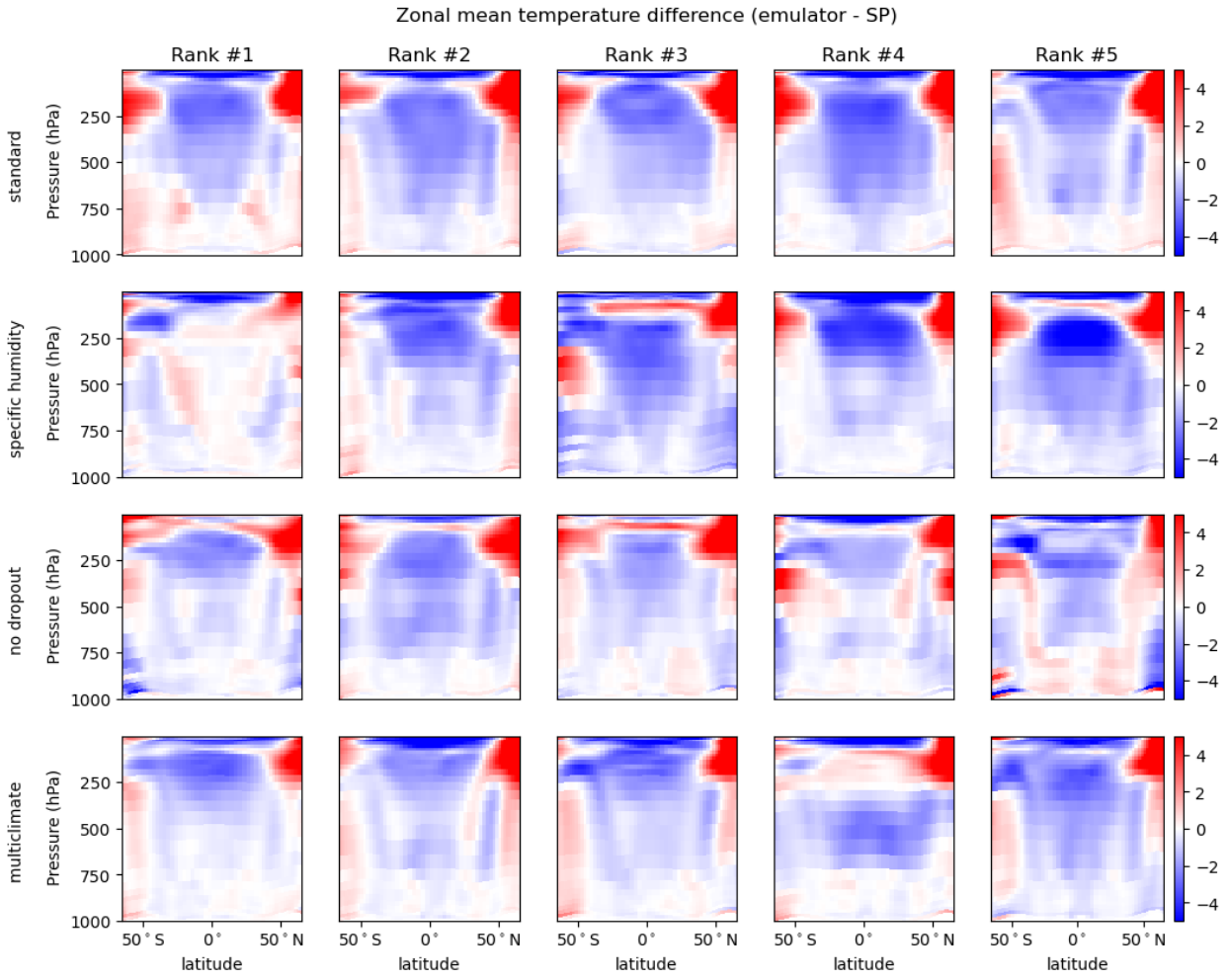

Fig. 2.5. This figure shows zonal mean temperature biases for the top five hybrid models across the standard, specific, no dropout, and multiclimate configurations. Models to the left have lower online error.

results in an improved, asymmetric zonal mean bias, indicating that while persistent, this bias structure can be systematically mitigated. Nevertheless, our bias structure is very sensitive to hyperparameter sampling, as the hybrid simulation with the smallest zonal biases comes from the specific humidity configuration, which is less stable than the standard configuration and consistently has higher offline temperature and moisture RMSE than the no dropout configuration. Sensitivity to hyperparameter tuning and high downstream, online noise may imply that reliably mitigating this bias structure in future work will also require testing additional hypotheses with large ensembles.

2.4 Estimating necessary sample size for robust detection

While the scale of our online sampling is extensive by design, one might naturally wonder whether or not this level of scale is actually necessary. Could our results be reproduced using a smaller sample size and fewer computational resources? We chose a large sample size of 330 NNs because the dispersion of online error seen in previous work like that of Ott et al. (2020) spanned two orders of magnitude. However, as seen in Figures 2.3 and 2.4, our online errors only span one. In this section, we estimate the sample size necessary to robustly detect meaningful differences online.

Selecting a sample size for non-parametric tests—especially for statistics like medians which do not obey the usual Central Limit Theorem—is not firmly established; there exist multiple methods but they are imperfect and make use of simplifying assumptions Noether (1987); Hamilton and Collings (1991). The situation is further complicated by the fact that many hybrid runs may crash, and only those completing the entire simulation are used for online error statistics.

To establish a conservative estimate, we make use of our least detectable (or highest significant p-value) yet scientifically justified finding—that not including ozone in the input increases temperature error online. In line with common practice, we choose a significance level, α, of 5% and a power, $1 - \beta$, of 80%. We also make several simplifying assumptions.

- An equal proportion, γ, of NNs integrate without crashing in both configurations. More specifically, this proportion is equal to the survival rate of the standard configuration.
- If the distributions of online error for both configurations vary, they only vary in location, not shape, skew, or dispersion (i.e., the distribution of online error for one configuration can be approximated by shifting the other).

Under these assumptions, the Mann Whitney U-Test (otherwise known as the Wilcoxon ranked sum test) is equivalent to a test of a difference in medians and provides a closed form for power calculations Noether (1987); Hamilton and Collings (1991). In this case, we apply the Mann Whitney U-Test to the online RMSEs for the standard configuration and a version of those same online RMSEs offset by the difference in ensemble-median RMSE when compared to the no ozone configuration. We arrive at a sample size estimate of 384, demonstrating the value of $\mathcal{O}(100)$ ensembles for detecting causally relevant differences in online error. We hope this sample size estimate serves as a useful reference point, and not a canonical requirement, for future work. Larger effect sizes require far fewer samples for detection. As an example, detecting the shifts in ensemble-median online temperature error for the multiclimate and no dropout configurations would require sample sizes of 312 and 29, respectively. Making use of a baseline with improved stability and a search space with less variation may also reduce the sample size necessary to detect a given effect size. Further details regarding the sample size estimation can be found in Text S3 in the SI.

2.5 Conclusion

Leveraging an end-to-end pipeline for $\mathcal{O}(100)$ ensembles of hybrid physics-ML simulations, we arrive at empirically robust recommendations for the development of future NN parameterizations of moist convection and identify counterintuitive nuances regarding the relationships between offline error, online error, and online stability. We summarize our findings below, categorized by primary and secondary importance:

Primary Findings:

- Using memory (to implicitly represent convective memory), batch normalization, and training on multiple climates reduce offline and online error and improve online stability.
- Dropout negatively impacts both offline and online error, but setting the dropout rate to zero reduces stability.
- Switching from an MSE to an MAE loss also improves offline error, but online error is worsened while online stability is improved.
- Using a "climate-invariant" feature transformation from specific to relative humidity and removing redundant input variables (i.e. feature pruning) both improve stability.
- Detecting causally relevant differences in online statistics with 80% power may require $\mathcal{O}(100)$ ensembles of hybrid simulations.

Secondary Findings:

- Smaller learning rates are weakly associated with lower offline heating error, lower online temperature error, and higher stability.
- Higher parameter counts are weakly associated with lower offline heating and moistening error as well as lower online temperature error.
- Use of the QHAdam optimizer (as configured in our training code) decreases online stability compared to Adam and RAdam.

When reporting relationships between offline error, online error, and online stability, we use an in-distribution offline test set to avoid confusion and maintain narrative consistency. However, if the root cause of online instability and error is a failure to generalize out-of-distribution, the use of an out-of-distribution offline test set may result in a more reliable

offline proxy for online error. To check whether or not this is the case, we also show rank correlations between multiclimate offline error and "in-distribution" online error in Table S2 in the SI. We find that these rank correlations are actually weaker than those for in-distribution offline and online error (with the exception of the multiclimate configuration, as seen in Table 2.3 and Table S1). That being said, there are a few differences worth noting, e.g. the multiclimate offline error is significantly lower with the multiclimate configuration and higher with the specific humidity configuration, validating results from Beucler et al. (2024). Additionally, the offline moistening error for the no dropout configuration is higher, not lower, than that of the standard configuration with a multiclimate test set. Nevertheless, it is possible that there exists a more reliable offline proxy for online error than the ones evaluated in this paper. Such a proxy would enable faster progress in NN parameterization development by augmenting the utility of more modern optimizers and optimization algorithms. Additionally, lessons learned from open competitions that crowd-source offline error optimization, like the recent 2024 LEAP Atmospheric Physics using AI (ClimSim) Kaggle competition, could become more relevant to the online hybrid climate modeling evaluation task Lin et al. (2024b).

Limitations in our work may qualify the claims we are able to make, but they also point to promising and concrete directions for future research. While our work is empirical, dynamical analyses in follow-up work may unveil the root causes for why using an MAE instead of MSE loss and removing dropout have opposite effects on online error and online stability (despite both reducing offline error). They may also identify physically-informed ways to systematically eliminate the online stratospheric zonal mean biases seen in this and prior work. Such advances may result in scenarios in which high online skill and stability are the norm and not the exception, warranting longer, multi-year integration runs and the use of a more complex GCM (as opposed to a simplified, zonally-symmetric aquaplanet with prescribed SSTs). Given how the online temperature and moisture RMSEs often continue to increase for many hybrid simulations that are deemed "stable" by virtue of having completed

one year of integration without crashing (as seen in Figures S4 through S13), multi-year integrations in future work may also be necessary to verify whether or not one-year integrations are a valid proxy for stability in general. For the purposes of this study, we believe one-year integrations may be sufficient since $> 50\%$ of hybrid runs crash within one simulation-year in our baseline and, in general, simulations that do crash most frequently crash within the first month, as seen in Figure S15. Nevertheless, true step-change improvements in both online error and online stability for closer-to-operational settings will likely require moving beyond fully-connected NNs to more sophisticated architectures, as showcased by Hu et al. (2024). Even further advancements may require the use of coupled ensembles, dynamics-informed curation of training data, additional "climate-invariant" feature transformations, and inherently stochastic models that better represent the coarse-grained effects of convection and turbulent flow Behrens et al. (2024); Balogh et al. (2021); Beucler et al. (2024); Yu et al. (2023b); Kohl et al. (2023).

Nevertheless, our results demonstrate that large-scale sampling is a vital tool for navigating noise and nuance in the online behavior of ML parameterizations. The dispersion in online RMSE or survival rates from hyperparameter tuning can be large enough to obscure the impact of different design choices, and it is possible to train a superior model (i.e., when evaluated online) using seemingly suboptimal design choices offline, as evidenced by a single NN from the specific humidity configuration having the lowest online zonal mean temperature biases across all 2,970 models evaluated (shown in Figure 2.3a). Had we drawn conclusions from this singular, performant NN parameterization, we would have come to a conclusion contrary to that implied by the ensemble at large. Although hundreds of ensemble members may be necessary to detect small yet statistically significant and causally relevant differences, our work suggests that, at a minimum, ensemble sizes of a few dozen should be used to detect larger effect sizes with sufficient power. While there exist multiple sophisticated strategies for mitigating emergent effects of online behavior like online coupled learning as described in Rasp (2020), gradient-free ensemble Kalman methods as in Lopez-Gomez et al. (2022),

and creating a fully differentiable hybrid physics-ML atmospheric model as demonstrated by Kochkov et al. (2024) with NeuralGCM, sampling at sufficient scale may ultimately still be necessary to empirically settle competing hypotheses and make reproducible progress.

Beyond climate simulation, the lessons learned in this paper are relevant to hybrid (ML-physics) simulations across the geosciences. We have demonstrated a standardized process for distinguishing emergent signal from noise in the online behavior of ML parameterizations. Our view is that without infrastructure for such rapid, large-scale downstream online tests, development of hybrid ML-physics models can be muddled. This arises from the difficulty of comparing configurations of ML-parameterizations whose distributions of online error fail to converge due to insufficient sampling and the technical complexity of sampling at sufficient scale. Evidently, the climate model use case is one in which high dispersion owing to stochastic aspects of ML optimization, such as architecture searches, can mask modest but real gains in downstream performance. It can also give a false impression of the benefits of a given design decision. This can ultimately manifest in premature conclusions and missed opportunities. We hope our work motivates a transition from extrapolating from anecdotal experiences or small ensemble sizes to adopting a more systematic and reproducible approach to ML parameterization development.

2.6 Open Research

Version v2 of spcam3.0-neural-net-spreadtesting, the codebase used for creating the reference SPCAM3 simulation and dynamical core for online simulations is preserved at `https://zenodo.org/records/11392025`, available via Apache License 2.0 and developed openly at `https://github.com/SciPritchardLab/spcam3.0-neural-net` Pritchard et al. (2024). v4 of ClimScale used for conducting hybrid physics-ML climate runs is preserved at `https://zenodo.org/records/14252488`, available via Apache License 2.0 and developed openly at `https://github.com/SciPritchardLab/ClimScale` Lin and Yu (2024). All experiments were run on NVIDIA V100 GPUs. Approximately 2,290 GPU-hours were used

to train 2,970 NNs across all nine configurations.

2.7 Acknowledgments

This work is primarily funded by National Science Foundation (NSF) Science and Technology Center (STC) Learning the Earth with Artificial Intelligence and Physics (LEAP), Award # 2019625-STC. High-performance computing was facilitated by Bridges2 at the Pittsburgh Supercomputing Center (PSC) through allocation ATM190002 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296. MP acknowledges co-funding from the US Department of Energy (DE-SC0023368). Tom Beucler acknowledges partial funding from the Swiss State Secretariat for Education, Research and Innovation (SERI) for the Horizon Europe project AI4PEX (Grant agreement ID: 101137682). Eliot Wong-Toi is funded by the Hasso Plattner Research School at UC Irvine. We are thankful to the system administrators at PSC, in particular Tom Maiden and TJ Olesky, as well as David Walling for his assistance with HPC support through the XSEDE Extended Collaborative Support Service program. We extend our gratitude to Dave Lawrence (NCAR), Laure Zanna (NYU), Sara Shamekh (NYU), Tian Zheng (Columbia University), Robert King (Stanford), Akshay Subramaniam (NVIDIA), and Stephan Hoyer (Google) for their insightful discussions during the 2024 LEAP-STC NSF site visit, the 2024 LEAP-STC annual meeting, and AGU24 Annual Meeting. Finally, we would like to express our gratitude to the anonymous reviewers at JAMES for their meticulous, constructive, and thoughtful feedback.

Chapter 3

3 Crowdsourcing ML Parameterization Innovation with a Seminal Dataset, Benchmark, and Competition

The work in this chapter is associated with the following publications / products and distills the lessons learned in their creation:

- Yu et al. (2023a)
- Lin et al. (2024b)

Keypoints

- Given the urgency of climate change and what is technically possible, the cycle time of idea to publication is too slow.
- Benchmarks, datasets, and competitions may accelerate the advent of machine learning enhanced climate models.
- The ClimSim saga and its associated papers provide a reproducible blueprint for other domains in climate science that could benefit from machine learning.

Abstract

There has long been speculation that machine learning enhanced climate models could one day supersede conventional numerical models. By learning to cheaply emulate higher resolution processes, machine learning parameterizations may hold the key to bypassing Moore's Law and doing justice to the small-scale processes that have outsize impacts on our long-term climate. Unfortunately, the pace of development of machine learning parameterizations has been slow relative to the actual pace of climate change and the speed at which

the field of AI has been advancing. While there are many reasons why this is the case, we argue that the situation can be vastly improved provided we streamline the process by which machine learning researchers 1.) become aware of domain science problems, 2.) are provided frictionless open-source software repositories to interface with these problems, and 3.) have public benchmarks by which they can evaluate progress. In this work, we showcase how we have executed on all three points with a seminal dataset, benchmark, and competition, and we detail how this can be a model for future endeavors.

3.1 Introduction

More than a decade ago, the introduction of the ImageNet dataset ignited the deep learning revolution Deng et al. (2009); Krizhevsky et al. (2012). Before 2012, neural networks faced skepticism: they lacked the mathematical elegance and interpretability of dominant methods like support vector machines, and their computational demands seemed unjustified given modest performance gains Cortes and Vapnik (1995). This changed abruptly in 2012, when convolutional neural networks trained on ImageNet's 14 million images—accelerated by GPU-powered parallel processing—achieved unprecedented accuracy in image classification. This breakthrough demonstrated deep learning's transformative potential, triggering an era of rapid AI advancement that continues to redefine technology today LeCun et al. (2015); Vaswani et al. (2017); Song et al. (2021).

However, today's operational climate models do not take advantage of modern machine learning innovations. This, however, is not for a lack of interest or potential. While many machine learning researchers are fully aware that climate change poses potentially existential risks to society, far fewer are aware that subgrid parameterizations are a major source of uncertainty and error for long-term projections of future climates. Furthermore, machine learning researchers who are aware of this issue often lack the means to contribute meaningfully without the close guidance of a domain scientist. Finally, without a common dataset or benchmarks, algorithmic progress becomes impossible to measure, mitigating incentives

for machine learning researchers to dedicate time toward inventing new models with state-of-the-art performance. As a consequence, progress in machine learning parameterizations has largely been driven by a relatively small group of climate scientists adapting machine learning tools at a far slower pace than the field of machine learning itself was advancing.

Closing the gap in research pace may not be as farfetched as one might initially assume. Just like ImageNet spawned an explosion of machine learning advancements, there also exists historical precedent for the strategic use of datasets, benchmarks, and competitions as a catalyst for scientific discovery. A notable example is in the field of biology with the biennial Critical Assessment of protein Structure Prediction (CASP), a community-wide experiment initiated in the 1990s to spur advancements in the "protein folding problem" Moult et al. (1995). This challenge involves predicting the three-dimensional structure of a protein from its amino acid sequence. For years, progress was incremental. However, in 2018, DeepMind, a subsidiary of Google, made a significant leap with the introduction of AlphaFold, which won the 13th CASP competition. This success was dramatically surpassed in 2021 by its successor, AlphaFold 2, which demonstrated unprecedented accuracy at CASP14, solving the protein folding problem for a large number of proteins Jumper et al. (2021). This breakthrough was hailed as a landmark achievement in biology and resulted in Demis Hassabis, the CEO of Deepmind, John Jumper, the lead researcher on the AlphaFold project, receiving the 2024 Nobel Prize in Chemistry, an award they shared with David Baker, another pioneer in computational protein design. AlphaFold continues to see advancements to this day, with the third version demonstrating substantial accuracy improvements over the previous generation Abramson et al. (2024).

In this chapter, we demonstrate how the ClimSim 2023 NeurIPS Datasets and Benchmarks paper and 2024 LEAP Kaggle Competition facilitated a step change in entraining momentum in machine learning research towards designing better ML parameterizations. We detail the critical ingredients that made this a success, highlight follow-up and ongoing work that further this mission, and explain lessons learned in hindsight. Altogether, we hope

this yields a blueprint for similar endeavors in which well-defined machine learning problems in earth system modeling can be adequately exposed to the massive community of machine learning researchers eager to make an impact.

3.2 Methods

3.2.1 ClimSim NeurIPS Datasets and Benchmarks Paper

The NeurIPS Datasets and Benchmarks track is the premier venue for exposing machine learning researchers to new datasets worth using for algorithmic advances. Established in 2021 as an "incubator to bootstrap publication on data and benchmarks," it runs alongside the main NeurIPS conference—a top publication venue for machine learning research Neural Information Processing Systems Conference (2021). With a competitive acceptance rate (32.6% in 2023) and a strict annual submission deadline (typically in May), preparation and planning are critical. If ClimSim had missed the 2023 deadline or been rejected, that would have delayed or derailed follow-up developments for ClimSim by at least a year. However, maximizing ultimate downstream impact requires balancing immediate acceptance with long-term adoption. Below, we outline key factors for excelling on both fronts.

3.2.1.1 Relevance: Because NeurIPS reviewers are typically not domain scientists, it is helpful to ensure that claims of high scientific, societal, or practical relevance are supported by a history of existing scientific literature using similar datasets for a shared problem as well as collaborations with leading scientists who have already published on the topic or are adjacent to the field. ClimSim benefited from an existing legacy of papers on using neural networks to model subgrid processes in climate models as well as an extensive co-author list that included many of the original pioneers in the field Rasp et al. (2018); Han et al. (2020); Mooers et al. (2021); Han et al. (2023). Additionally, the ClimSim dataset made use of an operational climate model actively supported by the U.S. Department of Energy, the Energy Exascale Earth System Model - Multiscale Modeling Framework (E3SM-MMF).

This bolstered the credibility of claims that ML innovation using this dataset could have immediate downstream impacts on future long-term climate projections.

3.2.1.2 Scale: In this section, scale specifically refers to dataset volume. Deep learning methods thrive when trained on large amount of data, and the machine learning literature has well-established "neural scaling laws" that show how the relationship between model accuracy and training dataset size empirically follows predictable power law curves across multiple model architectures and multiple domains Hestness et al. (2017); Kaplan et al. (2020); Hoffmann et al. (2022); Baniodeh et al. (2025). The sheer size of ClimSim's dataset—40 TB with 10 years of simulation data across three different configurations (Low-Resolution Aquaplanet, Low-Resolution Real Geography, and High-Resolution Real Geography)—is rare for multivariate regression datasets and potentially makes this a valuable contribution to fundamental machine learning research even outside of the climate science context.

3.2.1.3 Benchmarks: A publication in this track should test multiple models (ideally at least 4-5) with fundamentally different architectures on multiple metrics. In the case of ClimSim, there were six different architectures with varying degrees of complexity and both deterministic and stochastic evaluation metrics. Although the reviewers generally agreed this was sufficient, at least one was interested in the addition of spatiotemporal or spectral metrics. This may be worth adding to similar work going forward. Additionally, since intra-model variance can potentially be nontrivial, it would have been useful to have reported the mean and variance in performance for each architecture when trained using different seeds Reuel et al. (2024). The test set should only be used once, at the very end, for the final evaluation of the fully developed models to ensure unbiased assessments of their performance. To prevent overfitting and data leakage during the prototyping phase, it is advisable to stick to the validation set for hyperparameter tuning, preliminary analyses, and generating intermediate figures and tables. In the case of ClimSim, a separate "scoring" dataset (distinct from the training, validation, and test split) was used to prevent data leakage for the competition.

However, this is nontraditional and not necessarily recommended as standard practice going forward. Outside of model evaluation, the number of trainable parameters and computational expense for inferencing a single data point should also be reported for each model. The computational expense can be measured in terms of Floating Point Operations (FLOPs) or Multiply-Accumulate Operations (MACs), and a useful tool for doing so with PyTorch models can be found at this URL: https://github.com/Lyken17/pytorch-OpCounter. Finally, it is helpful, but by no means required, to also report the training logs for each model. A convenient and common way to do this is via Weights and Biases, which is free for academics: https://wandb.ai/site.

3.2.1.4 GitHub and HuggingFace repositories: Software for preprocessing, model training, postprocessing, evaluations, and visualizations should be well-documented and maintained in the open on GitHub. Likewise, the datasets and model weights themselves should be uploaded (with dataset and model cards) to HuggingFace, a popular platform for open-sourcing datasets, models, and libraries that lower the barrier of entry for getting started building on the latest open-source tools in machine learning. If a competition is also planned, code for generating the test set should be omitted from GitHub and the test set itself should not be present on HuggingFace. If these are deleted after the fact via GitHub or HuggingFace commits, great care should be taken to prevent third parties from reverse engineering the test set by deleting all diffs and commits with this data on GitHub and HuggingFace using `git-filter-repo`. More information on removing sensitive data from a repository can be found at this URL: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/removing-sensitive-data-from-a-repository.

The layout of the GitHub repository is essential, as it serves as the primary interface for users to learn how to get started with training and evaluating their own models. Factors such as ease of navigation, comprehensive documentation, and clear presentation are all taken into consideration by reviewers during the evaluation process. Beyond peer review,

these elements can significantly influence adoption by third parties who may wish to build upon the work after it is published, ultimately impacting the publication's ability to stay relevant and influential in the long term. A poorly organized repository risks alienating users, obscuring the value of the research, and limiting its broader impact, while a well-structured one can enhance accessibility, foster collaboration, and encourage contributions. To achieve this, consider starting with a template that is already established and widely recognized within the research community. For instance, one such template, directly recommended by a NeurIPS reviewer for the ClimSim project, is a Lightning-Hydra template found at this URL: https://github.com/ashleve/lightning-hydra-template. When collaborating on the GitHub repository, outstanding tasks should be posted as GitHub issues, and contributors should develop on separate branches from main, only merging to main via a pull request reviewed by a senior contributor. Ideally, all contributors should be using the same conda environment or, better yet, a Docker-style container saved to the GitHub repository when performing model training and evaluation, as subtle yet important differences can crop up between versions of various packages that can change the results.

Oftentimes, as was the case with ClimSim, there exists considerable boilerplate domain-specific code for preprocessing, postprocessing, visualization, and evaluation that is universal across all models. To the extent that is possible, this code should be abstracted away using object-oriented programming in a `data_utils.py` script. This script should define a central class whose attributes store useful constants, lists, and dictionaries. Its methods should encapsulate modular functions for all aforementioned data tasks. Each method must include type hints and comprehensive docstrings that specify expected inputs and outputs, their types, and (where applicable) their intended dimensions.

Finally, to streamline the process of getting started, the GitHub repository should also include a "Quickstart" Jupyter Notebook or Google Colab notebook. Ideally, this notebook should be self-contained, make appropriate use of the aforementioned `data_utils.py` script, and contain the bare minimum code necessary to evaluate model predictions and create main

the visualizations of interest.

3.2.1.5 Presentation: Since the primary audience for the paper are machine learning researchers with presumably little to no exposure to the topic, the paper needs to be easy to read for non-domain-scientists. There should be clear explanations for domain-specific terminology, and the actual "machine-learning task" needs to be made explicit. In the context of ClimSim, the machine-learning task (when using the full variable list) was 1D $\rightarrow$ 1D regression mapping an input vector $x \in \mathbb{R}^{557}$ to an output vector $y \in \mathbb{R}^{368}$. Significant effort should be invested in creating a main figure that illustrates this task in a visually-appealing, yet accurate manner. Other ways to improve the presentation of the submission would be to accompany the paper with a website and YouTube video that provide a digestible summary of the paper's most important contents. A convenient website template increasingly being used for publicizing machine learning research can be found at this URL: `https://github.com/nerfies/nerfies.github.io` Park et al. (2021).

3.2.2 LEAP - Atmospheric Physics Using AI (ClimSim) Kaggle competition

Kaggle is the leading platform for data science and machine learning competitions. With a user base exceeding 24 million machine learning enthusiasts, data scientists, and researchers across the world, Kaggle serves as a dynamic hub for learning, collaboration, and innovation. Participants engage in real-world predictive modeling challenges, tackling problems sourced from both the private sector—including top tech companies and startups—as well as academic and research institutions, many of which can apply for sponsorship via Kaggle's Competitions Research Grants Program. Quantitative trading firms and hedge funds like Jane Street, Optiver, and Two Sigma have hosted competitions on Kaggle with $100,000 prize pools to learn from the broader Kaggle community Desai et al. (2024); Forbes et al. (2023); Hamner et al. (2016). Meanwhile, academic research institutions like Harvard Medical School, Stanford University, and the Max Planck Institute for Meteorology have hosted

competitions to classify seizures from EEGs, predict RNA folding, and classify clouds from satellite imagery, respectively Jing et al. (2024); He et al. (2025); Rasp et al. (2019).

To apply for a Competition Research Grant, the hosting institution must be an academic/research institution or non-profit. Grants are awarded on a quarterly basis with up to three awardees in each application round. If you plan to pair a Kaggle competition with a NeurIPS Datasets and Benchmarks paper, the paper must first be submitted and accepted. This ensures that the published work and its associated GitHub repository can serve as a reference and incentive for future Kaggle participants. A confirmed NeurIPS acceptance also strengthens the likelihood of securing a grant that covers service fees and fully funds the prize pool (which may exceed $50,000).

3.2.2.1 Leak Prevention: By far, the single biggest concern for Kaggle administrators and competition hosts are data leakages. A data leakage occurs when information that would not realistically be available in an operational setting is inadvertently included, giving models that use this leakage an artificial advantage. Because Kaggle is home to some of the world's best data scientists and machine learning engineers, competition hosts should not underestimate the ability of participants to identify ways to reverse engineer information from the test set or information generally not intended for use in the machine learning models themselves. During the application process, Kaggle staff were particularly concerned about the possibility of participants running the E3SM-MMF model on their own to generate the actual prediction targets for the provided inputs. We had to reassure them that this was not possible given how E3SM-MMF cannot be "inferenced" on individual CRM inputs the same way a machine learning emulator can. In addition, the CRM "state," which is required in actual MMF runs, is never saved. Finally, the GPU-enabled version of E3SM-MMF exhibits stochastic behavior, meaning that two runs using the same restart files (which were also not shared) would not be identical.

That being said, the 2024 LEAP - Atmospheric Physics Using AI (ClimSim) Kaggle com-

petition still experienced two unanticipated leakage issues that were preventable in hindsight. Both of these leakage issues concerned the ability to reverse engineer location and timestep information to go from treating the problem as a 1D $\rightarrow$ 1D regression problem to treating it as a 2D $\rightarrow$ 2D regression problem in which the 2nd dimension corresponds to geographic location. Allowing for this would be physically inappropriate because of the doubly periodic boundary conditions of the CRM and clean temporal and spatial scale separation between the subgrid physics of the CRM and the host climate model's dynamics. Additionally, a machine learning model that maps 2D states to 2D tendencies would require extensive, non-trivial engineering to implement inside a climate model.

The first leakage exploit resulted from the data being generated and saved in a consistent order. While the sample IDs for the data were scrambled, the actual ordering was not, resulting in the ability of participants to generate 2D maps for the inputs and targets. This resulted in the competition hosts having to upload a new test set, extend the competition deadline by two weeks, and invalidate all prior submissions, a necessary but unfortunate outcome. This first exploit could have been prevented if the data were shuffled prior to being uploaded to Kaggle. The second leakage exploit, unveiled five days before the extended deadline, was more pernicious. This involved using the ozone volume mixing ratio at the third highest level of the atmosphere to infer timestep and location information exactly. This second exploit was undoubtedly more difficult to defend against, but it could be mostly mitigated had the test set been shuffled and randomly subsampled (using different secret seeds) prior to uploading to Kaggle. This would have created random holes in the 2D maps, making a 2D $\rightarrow$ 2D regression approach impractical. In the future, competition hosts should keep multiple test sets in reserve in the event of extenuating circumstances (or to assist in verification after the conclusion of the competition) and ensure that data is randomly shuffled and subsampled with different secret seeds prior to upload. In addition, it would have also been prudent to spend time learning about previous Kaggle competitions with leakage issues to understand what could have been done differently. While it is impossible to anticipate

every possible exploit, such steps would have undoubtedly made for a smoother experience for Kaggle staff, competition hosts, and competition participants.

3.2.2.2 Scalar Metric and Benchmarks: Once potential leakage issues are sufficiently mitigated, the next priority is to establish a singular, scalar metric by which competition participants will be evaluated. Ideally, improved performance on this metric should portend improved performance in the operational setting of interest. The metric the ClimSim competition used was the average R^2 value across all 368 outputs (in which the top 12 atmospheric levels for moistening, cloud liquid mixing ratio tendency, cloud ice mixing ratio tendency, and wind tendencies are zeroed out). One challenge with this metric is that variables that were more difficult to predict (e.g. cloud liquid mixing ratio tendency values higher in the atmosphere) were given the same weighting as variables that were slightly easier to predict yet more important for sensible zonal mean biases. Because R^2 is unbounded from below, poor performance on a few of the more difficult variables could result in extremely negative values, severely dragging down the cumulative R^2 score across all targets. In hindsight, it would have made more sense to design a custom metric that bounded R^2 values below at negative one (or maybe even zero) to reduce the time spent by Kaggle participants improving predictions for inherently unpredictable values. Finally, a successful Kaggle competition needs a benchmark submitted by the host, so participants can better gauge when they have surpassed the competition hosts' best models. Since a publication showcasing the benchmark model was being prepared in parallel to the competition, we were unable to share details about this benchmark model during the competition beyond disclosing that it was a U-Net. If transparency is possible, however, that is generally preferred by both Kaggle participants and Kaggle staff.

3.3 Impact

3.3.1 Reception

The ClimSim NeurIPS Datasets and Benchmarks paper received an Outstanding Paper Award at the NeurIPS conference, which is the highest possible award for its track. All reviewers recommended acceptance, and at least one reviewer deemed it a seminal paper. Subsequently, the 2024 LEAP - Atmospheric Physics using AI (ClimSim) Kaggle Competition received a Kaggle Competition Research Grant that waived all service fees and funded $\$50,000$ in prize pool money. The competition, which ran from April 18, 2024 to July 15, 2024, had 877 participants hailing from every continent (except Antarctica) who submitted 11,059 submissions over the course of the competition. Despite leakage issues during the competition, we were able to verify that the solutions from the winning teams did not take advantage of any leak by using an additional secret held-out test set that was shuffled and subsampled prior to upload.

3.3.2 Follow-Up Work and The Road Ahead

During the competition and in the months that followed, two major ClimSim breakthroughs came to fruition. The first was work from Hu et al. (2024) demonstrating stable climatology across five year online runs, state-of-the-art zonal mean biases, and accurate precipitation distributions. The second was an update to ClimSim itself—ClimSim-Online—that democratized online hybrid model testing by containerizing the CPU version of E3SM-MMF for the first time Yu et al. (2023c). Finally, work that tests winning architectures from the Kaggle competition online is currently underway. Preliminary results for this work are described in the next chapter.

3.4 Conclusion

The original ClimSim paper, Kaggle competition, Hu et al. (2024), and ClimSim-Online all present a major advance in the development of hybrid physics-ML climate models. For the first time, there exists a standardized dataset and benchmark (for both offline and online performance) that outsiders from different institutions and different domains can use to experiment with new methods and build upon the state-of-the-art. At the time of this writing, researchers from multiple institutions are currently experimenting with new methods to build upon ClimSim, and there is high motivation to adopt many of the open-sourced lessons from the Kaggle competition to further hybrid physics-ML climate model development in contexts outside of ClimSim and E3SM-MMF. While getting hybrid physics-ML climate models ready for CMIP7 might be an extremely ambitious target, the developments from ClimSim and follow-up work have demonstrably gotten us significantly closer to this goal.

Chapter 4

4 Harvesting ideas for hybrid physics-ML climate simulation from a $50,000 Kaggle Competition

The following chapter details work that is still in progress. This work showcases preliminary results, and a more complete draft is being prepared for submission before the end of 2025. Expected co-authors at the time of writing include Zeyuan Hu (NVIDIA Corporation), Tom Beucler (University of Lausanne), Helge Heuer (German Aerospace Center, DLR), Laura Mansfield (University of Oxford), Hannah Christensen (University of Oxford), Walter Hannah (Lawrence Livermore National Laboratory), Katherine Frields (University of California, Irvine), Mike Pritchard (NVIDIA Corporation), and the winners of the Kaggle competition themselves. The specific ordering and complete list of co-authors is subject to change.

Keypoints

- Online stability in the low-resolution, real-geography limit is now the norm, not the exception, across architectures.
- All architectures exhibit similar offline and online bias patterns but some respond differently to architecture-agnostic design decisions.
- Some Kaggle-inspired architectures achieve state-of-the-art online zonal mean biases, but no single hybrid model yields a pareto-improvement over Hu et al. 2025.

Abstract

Subgrid machine-learning (ML) parameterizations that cheaply yet accurately emulate higher-resolution physics too expensive to resolve explicitly have the potential to usher in

a new generation of climate models that better calibrate leading sources of uncertainty in long-term climate projections. However, near-term operational consideration of hybrid physics-ML climate models has been precluded by foundational issues in getting ML emulators to work reliably when coupled to the dynamical core of a climate model. In order to more rapidly drive progress in solving these issues, domain scientists and machine learning researchers opened up this problem to the broader machine learning and data science community with the release of ClimSim, an award-winning NeurIPS Datasets and Benchmarks publication, and its associated 2024 Kaggle competition. This paper directly follows up on the 2024 Kaggle competition by systematically testing ideas and architectures pioneered by the winning teams online. Our results demonstrate that online stability in the low-resolution, real-geography limit is now the norm, not the exception, across architectures. Remarkably, all tested architectures exhibit strikingly similar patterns of offline and online bias, though their responses to architecture-agnostic design choices (e.g., expanding input variables) can differ significantly. Moreover, the vertical profile of the offline moistening bias changes in a systematic way with precipitation percentile, and this dependence appears robust across architectures. While some Kaggle-inspired architectures achieve state-of-the-art results on certain metrics like zonal mean bias patterns and global RMSE, no single architecture outperforms the U-Net from Hu et al. (2025) across all evaluated metrics.

Plain Language Summary

Future climate models may use machine learning (ML) to stand in for small-scale physical processes that are too costly to simulate directly over long timescales. These "hybrid" physics–ML models could improve predictions by reducing uncertainties from current approximations. But making them run reliably in full climate simulations has been a major challenge. To speed progress, scientists created an open dataset, benchmarking framework, and global competition to improve these ML components. This paper follows up on that

competition by testing ideas from the winning teams inside hybrid climate models. For the first time, we show that stable hybrid simulation is now routine across a range of ML architectures. We find that different architectures share similar patterns of errors both before and after coupling, although their responses to added training inputs can differ. One type of error—a moistening bias—varies in a consistent way with precipitation percentile across all tested models. Finally, some competition-inspired designs achieve state-of-the-art scores on individual performance measures, but no single approach beats the previous benchmark (Hu et al. 2025) on every metric.

4.1 Introduction

Climate models used for contemporary long-term projections operate at a coarse resolution ($>$ 50 km along the horizontal) that obscures the effects of subgrid processes like convection, radiation, and turbulence. Resolving these processes explicitly is prohibitively computationally expensive, especially when considering the fact that multiple realizations of future climates over long time horizons and different emissions scenarios are required for proper uncertainty quantification and actionable adaptation. Instead, hand-tuned parameterizations are used to approximate subgrid processes, introducing systematic biases and large error bars in the amount of expected warming O'Gorman (2012); Webb et al. (2013); Sherwood et al. (2014); Ceppi and Hartmann (2016); Schneider et al. (2017). One approach for more explicitly resolving these subgrid processes without relying on a natively high-resolution discretization (as seen in Global Cloud Resolving Models [GCRMs]) is Multiscale Modeling Framework (MMF), also known as superparameterization. In MMF, a Cloud Resolving Model (CRM) with doubly periodic boundary conditions and higher spatial and temporal resolution than the host climate model is embedded inside each grid cell. While MMF is more computationally efficient than GCRMs, it remains too expensive for operational long-term climate projections Gentine et al. (2018).

However, a convenient (albeit unintentional) perk of the MMF approach is that the clean

scale separation between the embedded CRMs and the host climate model make the CRM a natural emulation target for machine learning (ML) models. These ML models hold the promise of cheaply yet accurately emulating the coarsened effects of more explicitly resolved subgrid processes at a fraction of the cost Rasp et al. (2018); Gentine et al. (2018); Mooers et al. (2021). Unfortunately, this promise remains unrealized despite years of interest because of a lack of reliability, strong persistent biases, and emergent failure modes when deploying these ML models in an "online" setting (i.e. when dynamically coupled to the host climate model). To accelerate progress on fixing these issues, a group of climate scientists and machine learning researchers released ClimSim, a NeurIPS Datasets and Benchmarks publication that opened up this ML problem to the machine learning community at large. This was quickly followed up by Hu et al. (2024), which showcased stable five-year online simulations with explicit cloud condensate coupling for the first time, ClimSim-Online, which democratized online testing via a containerized version of E3SM-MMF, and an official ClimSim Kaggle competition with a $50,000 prize pool, which saw some of the world's best machine learning engineers and data scientists compete in a three month competition.

Our hypothesis is that by expanding both awareness and accessibility of this research problem to those in the machine learning community, we can facilitate advancements that would not have happened had this research problem stayed confined to domain scientists limited by time, personnel, and funding constraints. Indeed, the 2024 LEAP ClimSim Kaggle competition attracted thousands of entrants, hundreds of teams, and over ten thousand submissions from people across the globe. However, while hundreds of teams were able to surpass the offline R^2 baseline set using the U-Net architecture from Hu et al. 2025, the question remains—do offline competition-winning innovations also yield online benefits?

Whether or not the ideas pioneered by the competition translate online is an essential component to assessing the effect of democratizing access to this research problem on shortening the timeline to operational hybrid physics-ML climate models, and it is precisely the question this work aims to answer. In the following sections, we systematically evaluate

both architectures and architecture-agnostic design decisions online to achieve state-of-the-art (SOTA) values on several metrics, identify persistent biases whose structure is remarkably consistent across architectures, and demonstrate for the first time that online stability in the low-resolution, real-geography ClimSim limit is now regularly reproducible and should be the norm, not the exception, going forward.

4.2 Methods

In order to make this post-competition analysis tractable, we do not exhaustively investigate every architecture and architecture-agnostic design decision utilized in the winning solutions from the Kaggle competition. Winning solutions were often composed of a weighted average of vastly different architectures highly optimized for strong performance on a singular offline scalar metric (averaged R^2 across all target variables). Instead, we make use of six core architectures: the U-Net from Hu et al. (2024), a Squeezeformer from the first place team, a "Pure ResLSTM" from the second place team, a custom architecture called the "Pao Model" from the third place team, a "ConvNeXt" convolutional neural network from the fourth place team, and an "Encoder-Decoder LSTM" from the fifth place team. Feature selection, preprocessing, postprocessing, and training settings like batch size, number of epochs, loss function, learning rate, and learning rate scheduler are unified across all architectures to facilitate apples-to-apples inter-architecture comparisons. Each core architecture is evaluated across five configurations using "architecture-agnostic design decisions." For each architecture and architecture-agnostic decision, we train three models using different seeds, yielding a total of 90 models for the entire study. Input variables and input normalizations for the "standard," "confidence loss," and "difference loss" configurations are the same and are shown in Table 4.1. The "multirepresentation" configuration makes use of the same input variables but across three separate normalizations. The "expanded inputs" configuration makes use of an expanded variable list (shown in Table 4.3) that matches what was used for the U-Net-expanded model seen in Hu et al. (2024). Finally, all models across all

configurations share the same output variables and output normalizations, which are listed in Table 4.2.

Variable	Units	Description	Normalization
$T(z)$	K	Temperature	(x-mean)/(max-min)
$RH(z)$		Relative humidity	
$q_n(z)$	kg/kg	Liquid and ice cloud mixing ratio	1 - exp(-λx)
Liquid partition (z)		Diagnostic microphysics constraint	
$u(z)$	m/s	Zonal wind	(x-mean)/(max-min)
$v(z)$	m/s	Meridional wind	(x-mean)/(max-min)
O3(z)	mol/mol	Ozone volume mixing ratio	(x-mean)/(max-min)
CH4(z)	mol/mol	Methane volume mixing ratio	(x-mean)/(max-min)
N2O(z)	mol/mol	Nitrous volume mixing ratio	(x-mean)/(max-min)
PS	Pa	Surface pressure	(x-mean)/(max-min)
SOLIN	W/m^2	Solar insolation	x/(max-min)
LHFLX	W/m^2	Surface latent heat flux	x/(max-min)
SHFLX	W/m^2	Surface sensible heat flux	x/(max-min)
TAUX	W/m^2	Zonal surface stress	(x-mean)/(max-min)
TAUY	W/m^2	Meridional surface stress	(x-mean)/(max-min)
COSZRS		Cosine of solar zenith angle	(x-mean)/(max-min)
ALDIF		Albedo for diffuse longwave radiation	(x-mean)/(max-min)
ALDIR		Albedo for direct longwave radiation	(x-mean)/(max-min)
ASDIF		Albedo for diffuse shortwave radiation	(x-mean)/(max-min)
ASDIR		Albedo for direct shortwave radiation	(x-mean)/(max-min)
LWUP	W/m^2	Surface upward longwave flux	(x-mean)/(max-min)
ICEFRAC		Sea-ice area fraction	
LANDFRAC		Land area fraction	
OCNFRAC		Ocean area fraction	
SNOWHICE	m	Snow depth over ice	(x-mean)/(max-min)
SNOWHLAND	m	Snow depth over land	(x-mean)/(max-min)

Table 4.1: List of input features used in the standard, confidence loss, and difference loss configurations.

4.2.1 Architectures

4.2.1.1 U-Net The U-Net is taken from Hu et al. (2024), which itself was adapted from a 2-D U-Net created for score-based diffusion modeling in Song et al. (2020). In this U-Net, an encoder progressively downsamples the vertical dimension while expanding the feature space, and the decoder reverses this operation. Skip connections directly link layers of the

Variable	Units	Description	Normalization
$dT(z, t_0)$	K/s	Temperature tendency	x/std
$dq_v(z, t_0)$	kg/kg/s	Water vapor tendency	x/std
$dq_n(z, t_0)$	kg/kg/s	Liquid and ice cloud tendency	x/std
$du(z, t_0)$	m/s^2	Zonal wind tendency	x/std
$dv(z, t_0)$	m/s^2	Meridional wind tendency	x/std
NETSW	W/m^2	Net shortwave flux at surface	x/std
FLWDS	W/m^2	Downward longwave flux at surface	x/std
PRECSC	m/s	Snow rate (liquid water equivalent)	x/std
PRECC	m/s	Rain rate	x/std
SOLS	W/m^2	Downward visible direct solar flux to surface	x/std
SOLL	W/m^2	Downward near-IR direct solar flux to surface	x/std
SOLSD	W/m^2	Downward visible diffuse solar flux to surface	x/std
SOLLD	W/m^2	Downward near-IR diffuse solar flux to surface	x/std

Table 4.2: List of output features used by all models.

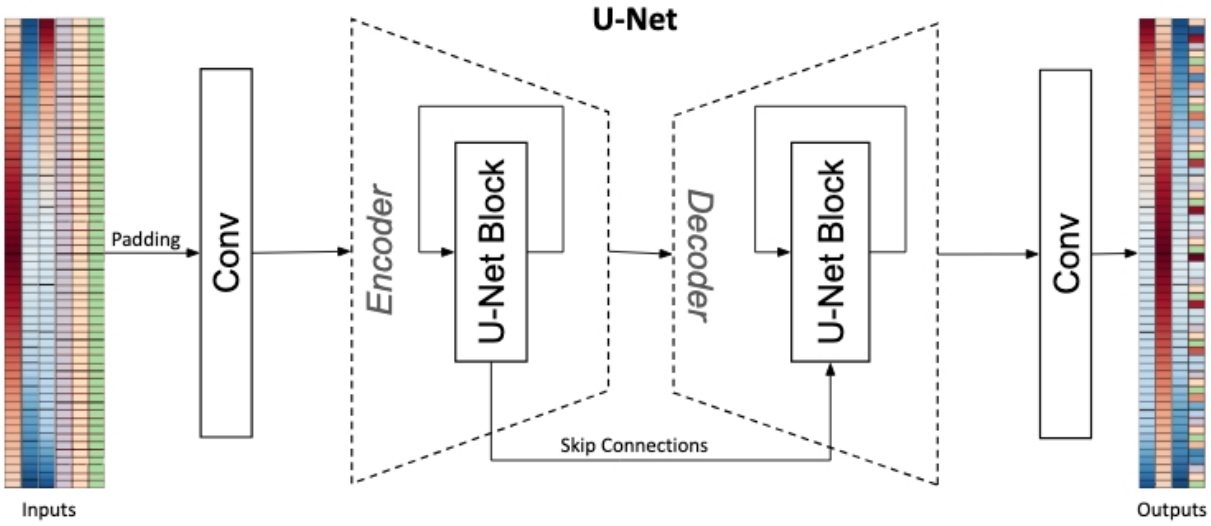

Fig. 4.1. Diagram for U-Net architecture.

encoder to corresponding layers of the decoder, preserving fine-grained details that might be lost during the downsampling process. This architecture serves as a useful baseline by which to compare winning architectures from the Kaggle competition. The diagram for this architecture is shown in Figure 4.1. The number of trainable parameters for the standard configuration of this architecture is 12,975,373.

4.2.1.2 Squeezeformer The Squeezeformer, shown in Figure 4.2, is an architecture used by the first place team in the Kaggle competition. Originally invented for automatic speech

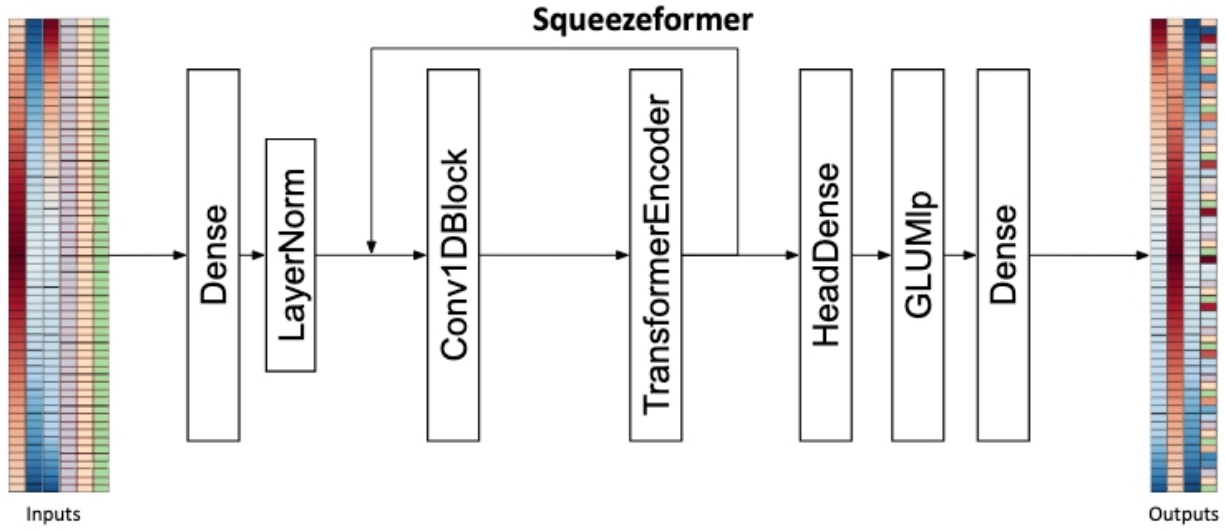

Fig. 4.2. Diagram for Squeezeformer architecture.

recognition by Kim et al. (2022), this architecture has shown incredible promise in other contexts, having been used for first and second place finishes in Kaggle competitions for fingerspelling recognition and RNA folding, respectively Das et al. (2023); Chow et al. (2023). This architecture integrates convolutional and transformer components to process sequential data. Initially, input features are projected into a high-dimensional embedding space by a linear layer, followed by layer normalization. The core of the network consists of a series of identical blocks stacked together. Each block first passes the data through a `Conv1DBlockSqueezeformer` to capture local contextual information, and then through a standard `TransformerEncoder` with multi-head self-attention to model global dependencies across the sequence. After processing through all the blocks, the final feature representation is passed to a projection head composed of a dense layer, a Gated Linear Unit (GLU) feed-forward network, and a final linear layer that maps the features to the desired output dimensions. The number of trainable parameters for the standard configuration of this architecture is 44,785,225.

4.2.1.3 Pure ResLSTM The "Pure ResLSTM", shown in Figure 4.3 is a multi-layer bidirectional LSTM (Long Short-Term Memory) network adapted from one of the architectures used by the second place team. This architecture consists of multiple LSTM layers,

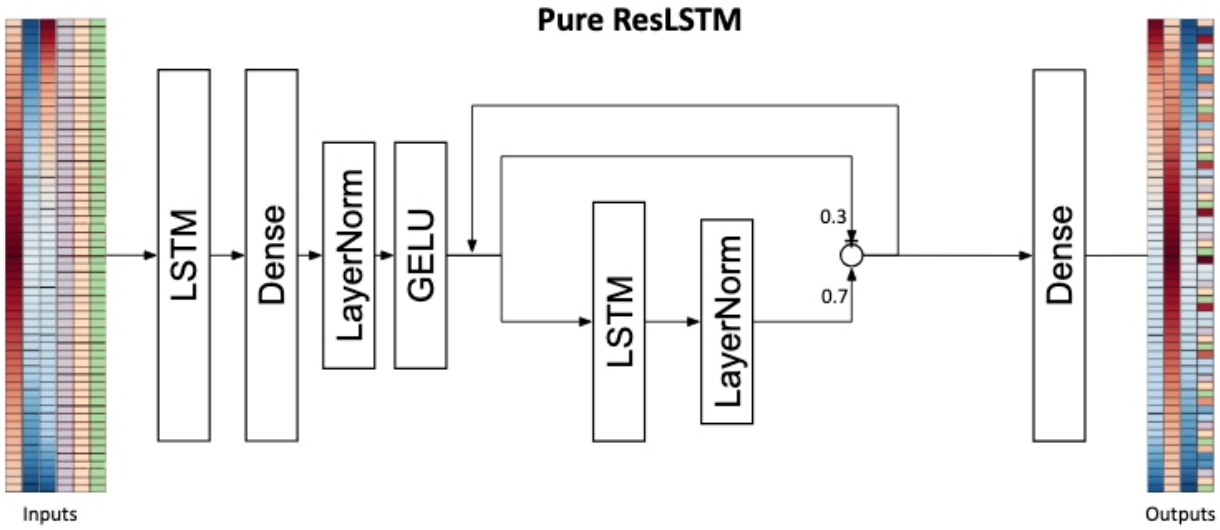

Fig. 4.3. Diagram for pure resLSTM architecture.

each followed by a layer normalization and GELU activation. The model takes input profiles and scalar features, processes them through 10 LSTM layers with 256 hidden units in each direction, and applies a unique residual connection strategy where outputs from later layers are combined with previous layer outputs using a weighted sum (0.7 * current layer + 0.3 * previous layer). After the LSTM stack, a fully connected layer transforms the features into target profiles and scalar predictions. The number of trainable parameters for the standard configuration of this architecture is 15,395,341.

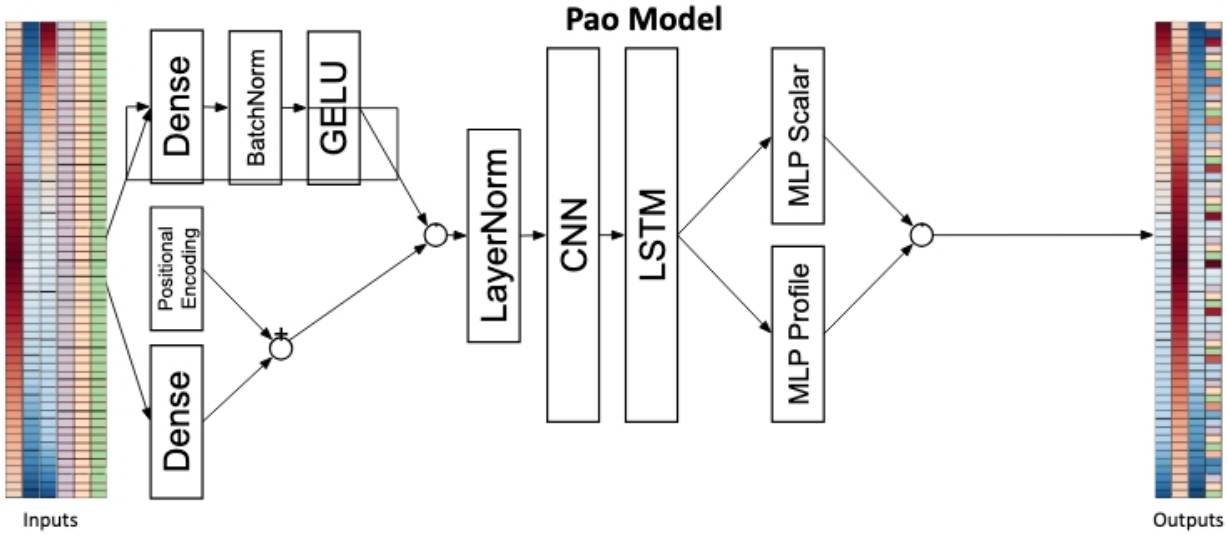

Fig. 4.4. Diagram for Pao Model architecture.

4.2.1.4 Pao Model The "Pao Model", shown in Figure 4.4, is adapted from an architecture created from scratch by the third place team in the Kaggle competition. This architecture explicitly treats vertically-resolved variables like temperature, relative humidity, and zonal wind in a different manner than scalar variables like surface pressure, latent heat flux, and sensible heat flux. Vertically-resolved variables are first linearly scaled, then all are concatenated and passed through a linear layer and positional encoding. Scalar variables are processed through a two-layer Multi-Layer Perceptron (MLP) with GELU activations and batch normalization, then projected to match the profile dimension at each vertical level. These two streams of features are then concatenated and fed into a core sequence of layers: a stack of four 1D convolutional residual blocks followed by a two-layer bidirectional LSTM. This hybrid CNN-LSTM backbone processes the features across the 60 vertical levels, learning both local patterns and long-range dependencies. The output is split into two branches: one MLP produces profile outputs for each vertical level, and another MLP produces scalar outputs from the flattened LSTM features. The number of trainable parameters for the standard configuration of this architecture is 18,876,133.

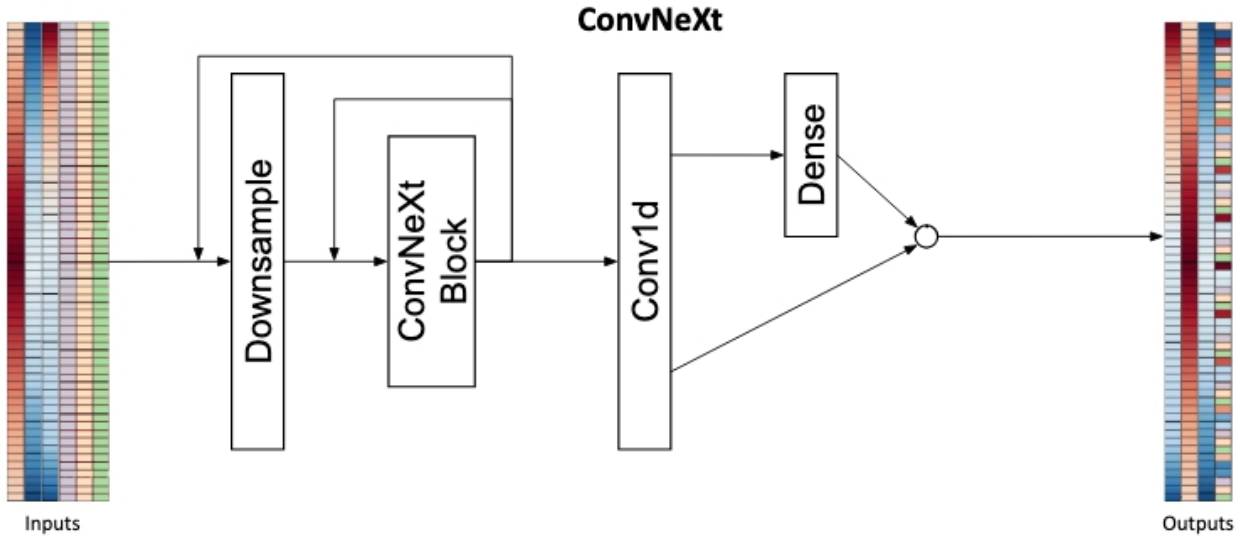

Fig. 4.5. Diagram for ConvNeXt architecture.

4.2.1.5 ConvNeXt The "ConvNeXt" model, shown in Figure 4.5, is an architecture used by the fourth place team in the Kaggle competition. It is adapted from work by Liu

et al. (2022), who invented this new generation of convolutional neural nets to be competitive with vision transformers, which have become increasingly dominant for vision tasks. The 1D ConvNeXt model here begins with a stem convolutional layer that processes concatenated input features (profile variables with vertical levels and scalar variables expanded across those levels) into an initial feature representation. This is followed by a series of downsampling stages, each consisting of a downsampling layer that reduces spatial dimensions and a sequence of residual blocks. The architecture employs four main stages with progressively increasing feature dimensions and decreasing spatial resolution through strided convolutions. Each stage contains multiple ConvNeXt blocks that use depthwise convolutions with large kernel sizes for efficient spatial mixing, combined with layer scaling and pointwise linear transformations. The final stage outputs are processed through a convolutional layer that produces combined profile and scalar predictions, with scalar variables further processed by a fully connected layer. The number of trainable parameters for the standard configuration of this architecture is 26,805,429.

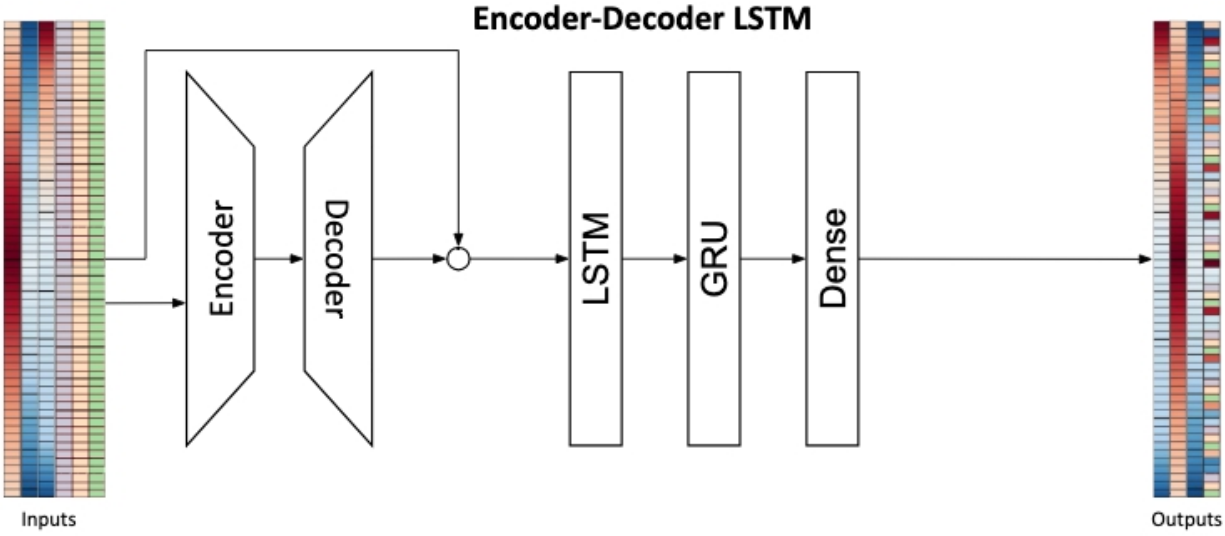

Fig. 4.6. Diagram for Encoder-Decoder LSTM architecture.

4.2.1.6 Encoder-Decoder LSTM The "Encoder-Decoder LSTM," shown in Figure 4.6, is an architecture used by the fifth place team in the Kaggle competition. This architecture first uses an encoder-decoder MLP structure to learn compressed representations of the

input data before recurrent processing. These processed features are concatenated and fed into a bidirectional LSTM layer, enabling the capture of vertical dependencies. The LSTM output is subsequently input into a bidirectional (Gated Recurrent Unit) GRU layer, which refines the sequence representations with a smaller hidden size. Finally, the GRU output is passed through another MLP to generate the model's output. The number of trainable parameters for the standard configuration of this architecture is 18,582,976.

4.2.2 Architecture-Agnostic Configurations (AAC)

Architecture-Agnostic Configurations refer to different modifications that are applied uniformly across all architectures. This facilitates a better understanding of intra-architecture variability that goes beyond choice of seed during training. Although hyperparameter tuning has been used to estimate intra-architecture variability in the past, as seen in Lin et al. (2025), smaller sample sizes and larger differences between core architectures make that a less practical strategy for this study. The modifications tested here are largely inspired by strategies that different Kaggle teams claimed were essential to improving their leaderboard scores. The only exception to this rule would be the "Expanded Inputs" configuration, which is informed by the input variable expansion used in Hu et al. (2024).

4.2.2.1 Standard The standard configuration presents no modifications and simply serves as a starting point from which other configurations make changes.

4.2.2.2 Confidence Loss The confidence loss configuration is inspired by the "confidence head" used by the first place team. This confidence head mirrors the final prediction layer of the architecture and is used to predict the loss, L_{pred}, for each output in the original prediction layer. The confidence head's prediction for this loss is denoted $\hat{L}_{pred}$. The loss function for this configuration applies the Huber loss to the sum of the original loss and the loss for the confidence head prediction, L_{loss}. The combined loss is denoted L_{conf}.

$$L_{pred} = huber(\hat{y}, y)$$

$$L_{loss} = huber(\hat{L}_{pred}, L_{pred})$$
$$L_{conf} = L_{pred} + L_{loss}$$

4.2.2.3 Difference Loss The difference loss configuration, inspired by the second place team, adds an additional term in the loss function that compares vertical differences for the vertically-resolved variables y_{diff}. The combined loss is denoted L_{total}.

$$L_{pred} = huber(\hat{y}, y)$$
$$L_{diff} = huber(\hat{y}_{diff}, y_{diff})$$
$$L_{total} = L_{pred} + L_{diff}$$

4.2.2.4 Multirepresentation The multirepresentation configuration, adapted from the first place team, concatenates three versions of the vertically-resolved inputs, each normalized in a different way. The first version, denoted as x_{lev}, is normalized using per-level mean and standard deviation. The second version is normalized using the mean and standard deviation for the entire vertical profile $x_{profile}$. The third version, denoted as x_{log}, uses a mirrored log normalization shown below. Finally, the scalar variables are normalized using their mean and standard deviation.

$$x_{log} = \begin{cases} \log\left(x_{lev} - \alpha + 1\right) & \text{where } x_{lev} \geq \alpha, \\ -[\log\left(\alpha - x_{lev} + 1\right)] & \text{where } x_{lev} < \alpha, \end{cases}$$

where α is set to the per-level minimum for x_{lev} found in the training data.

4.2.2.5 Expanded Inputs The expanded inputs configuration appends inputs listed in Table 4.3 to the baseline set of input variables listed in Table 4.1. This expanded variable list matches the variable list used to achieve state-of-the-art results shown in Hu et al. (2024). These expanded inputs consist of tendencies and large scale forcings at two timesteps as well as sin and cosine of latitude.

Variable	Units	Description	Normalization
$dT_{adv}(z, t_0)$	K/s	Large-scale forcing of temperature at (t)	x/(max-min)
$dq_{T,adv}(z, t_0)$	kg/kg/s	Large-scale forcing of total water at (t)	x/(max-min)
$du_{adv}(z, t_0)$	m/s^2	Large-scale forcing of zonal wind at (t)	x/(max-min)
$dT_{adv}(z, t_{-1})$	K/s	Large-scale forcing of temperature at (t-1)	x/(max-min)
$dq_{T,adv}(z, t_{-1})$	kg/kg/s	Large-scale forcing of total water at (t-1)	x/(max-min)
$du_{adv}(z, t_{-1})$	m/s^2	Large-scale forcing of zonal wind at (t-1)	x/(max-min)
$dT(z, t_{-1})$	K/s	Temperature tendency at (t-1)	x/std
$dq_v(z, t_{-1})$	kg/kg/s	Water vapor tendency at (t-1)	x/std
$dq_n(z, t_{-1})$	kg/kg/s	Total cloud tendency at (t-1)	x/std
$du(z, t_{-1})$	m/s^2	Zonal wind tendency at (t-1)	x/std
$dT(z, t_{-2})$	K/s	Temperature tendency at (t-2)	x/std
$dq_v(z, t_{-2})$	kg/kg/s	Water vapor tendency at (t-2)	x/std
$dq_n(z, t_{-2})$	kg/kg/s	Total cloud tendency at (t-2)	x/std
$du(z, t_{-2})$	m/s^2	Zonal wind tendency at (t-2)	x/std
cos(lat)		Cosine of latitude	
sin(lat)		Sine of latitude	

Table 4.3: List of additional input features used in the Expanded Inputs configuration.

4.3 Results

4.3.1 Offline R^2 Evaluation

After standardizing feature selection, preprocessing, postprocessing, and training settings across architectures, we see greater uniformity in R^2 scores than what may have been expected in the Kaggle competition, as seen in Figure 4.7. This is in spite of a considerable gap in leaderboard performance between the baseline U-Net and the winning solutions. Nevertheless, the Squeezeformer, which was used in the first place solution, shows clear separation from all architectures on all variables, and the ConvNeXt model, which was used in the fourth place solution, seems to slightly outperform the U-Net across all vertically-resolved variables. Meanwhile, the Encoder-Decoder LSTM (from the fifth place solution) appears to be neck-and-neck with the U-Net while the Pure ResLSTM and Pao Model (from the second and third place solutions, respectively) seem to perform worse.

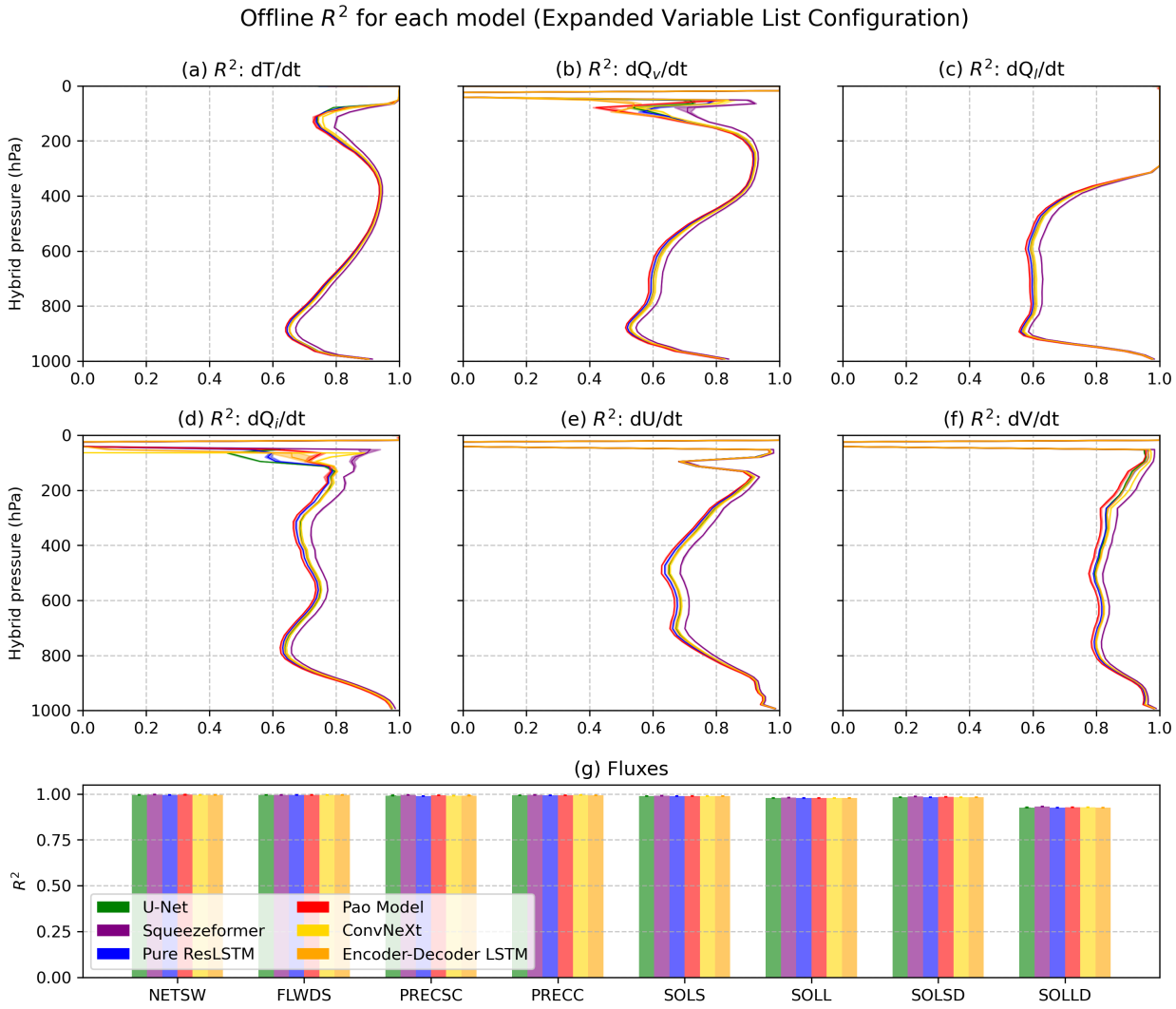

Fig. 4.7. This figure shows offline R^2 values for each variable across architectures, using the expanded variable list configuration—the setup that achieved the highest offline R^2 among all architectures.

4.3.2 A new state-of-the-art in online zonal mean biases

Across all ninety models evaluated online, a few achieve state-of-the-art zonal mean biases when compared to Hu et al. (2024). In Figure 4.8, we show five year zonal mean biases across all vertically-resolved variables for the Encoder-Decoder LSTM with confidence loss. There is a marked reduction for both liquid and ice cloud bias when compared to the previous state-of-the-art, and the biases for temperature and specific humidity are also less severe in magnitude. However, online results for this model do not constitute a pareto-improvement over Hu et al. (2024) as the cold tropical temperature bias extends lower in the troposphere while the dry moisture bias extends across more latitudes. Additionally, precipitation extremes are underestimated with this particular model, as seen in Figure 4.9.

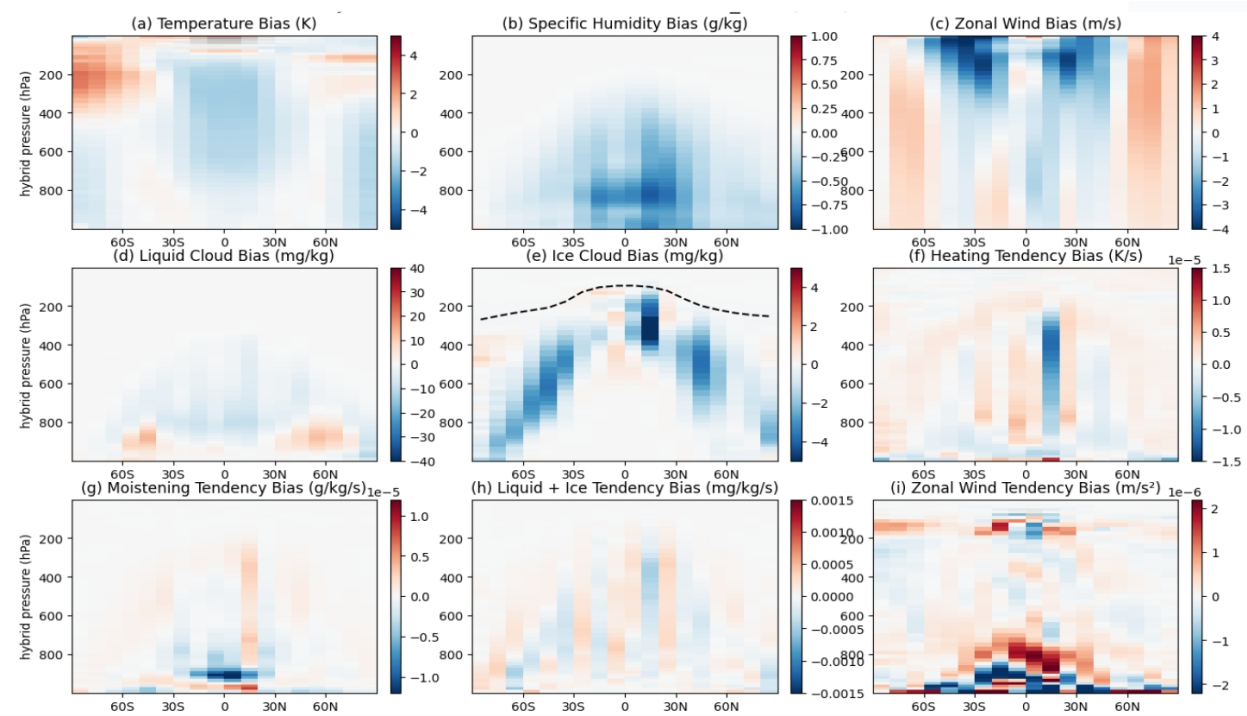

Fig. 4.8. SOTA online five year zonal mean biases for Encoder-Decoder LSTM trained with confidence loss (seed 7).

4.3.3 Common online pathologies

The remaining biases shown in Figure 4.8 are similar in spatial distribution and structure to those not only in Hu et al. (2024) but also other architectures evaluated as part of this study.

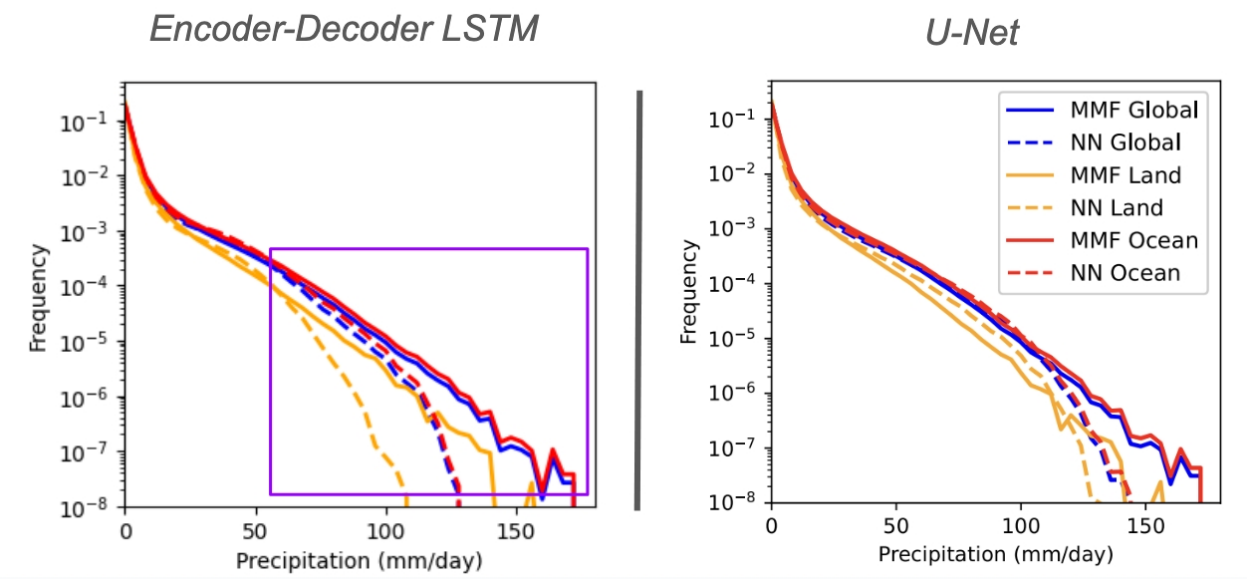

Fig. 4.9. Precipitation Histograms for Encoder-Decoder LSTM trained with confidence loss (seed 7). The purple box highlights areas where precipitation is underestimated. The U-Net refers to the results shown in Hu et al. 2025.

For example, we see similar online temperature and moisture zonal mean biases, shown in Figures 4.10 and 4.11, respectively. All architectures also consistently underestimate total precipitable water online, as seen in Figure 4.12.

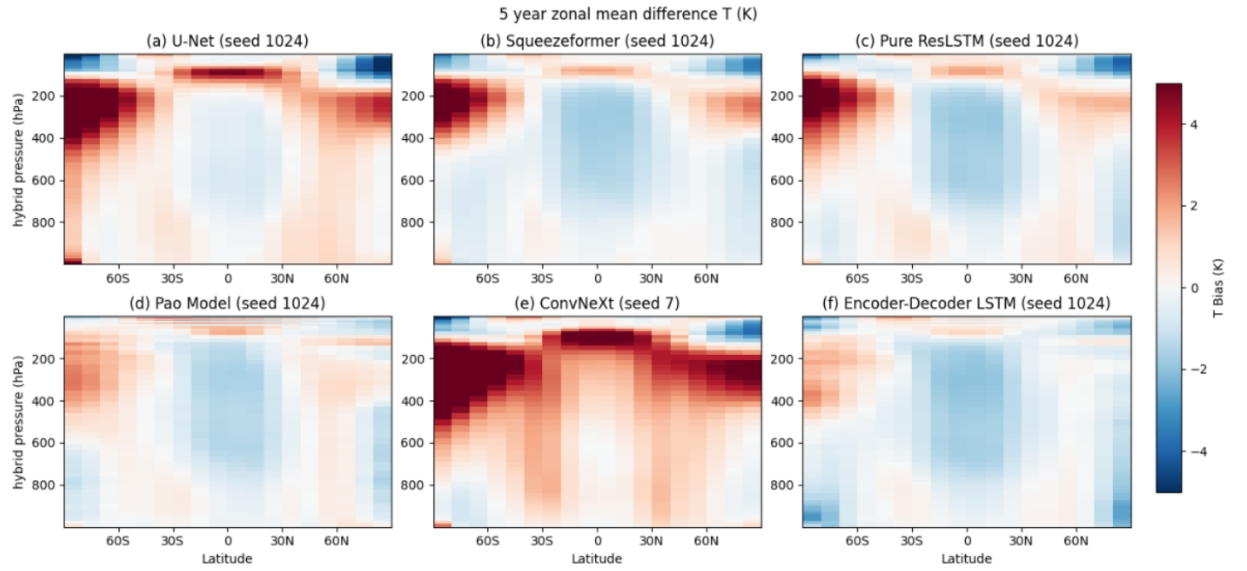

Fig. 4.10. This figure shows online zonal mean temperature biases across architectures in the standard configuration.

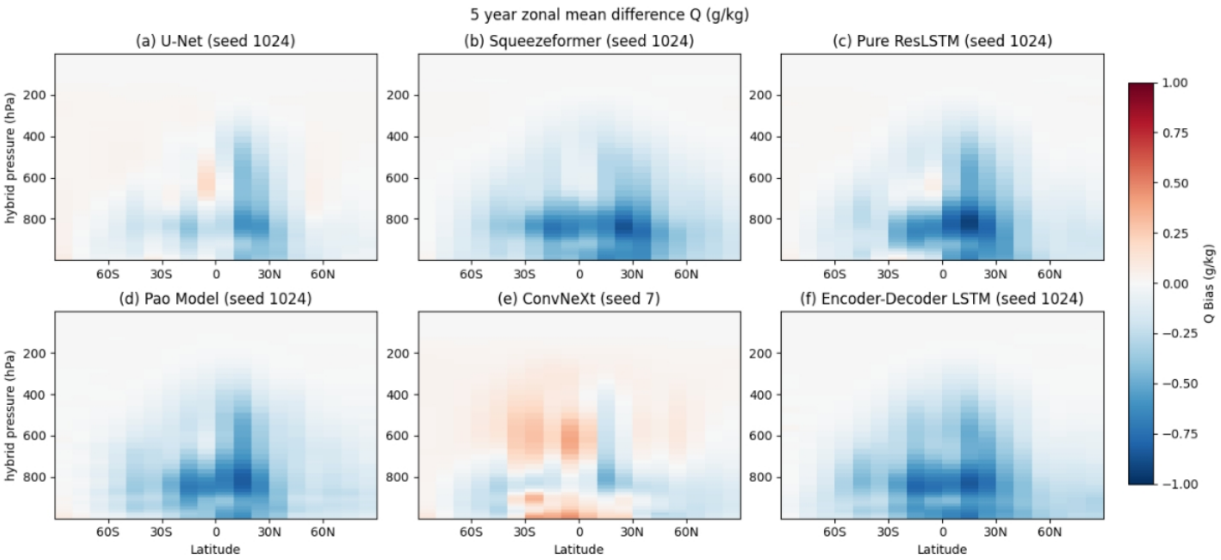

Fig. 4.11. This figure shows online zonal mean moisture biases across architectures in the standard configuration.

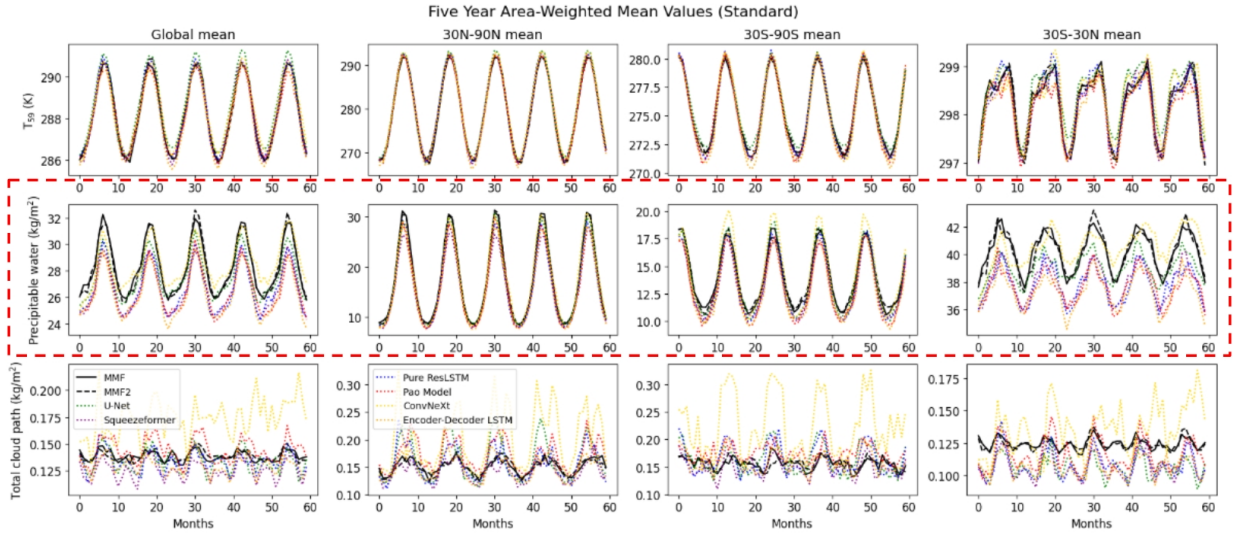

Fig. 4.12. This figure shows online five year global means for models in the standard configuration for surface temperature (T_{59}), total precipitable water, and total cloud path. All models consistently underestimate total precipitable water, particularly in the tropics.

4.3.4 Universal offline biases

All architectures tested in this study exhibit structurally similar offline moistening, liquid cloud tendency, and ice cloud tendency biases, as seen in Figure 4.13, Figure 4.14, and Figure 4.15. However, the pattern of these latter two biases are not present in predictions directly produced by the top two Kaggle teams, shown in Figure 4.16. This implies that they are almost certainly unintended side effects of using the microphysics constraint pioneered by Hu et al. (2024), which was applied to all models in this study. This constraint diagnoses liquid and ice cloud from the total as a function of temperature and holds exactly on the CRM grid for any instantaneous CRM timestep. However, because the neural network is learning the spatially and temporally *coarsened* effect of the CRM, the coarsened temperature may be structurally inconsistent with the heterogeneous CRM grid temperature ordinarily used for the one-moment microphysics scheme.

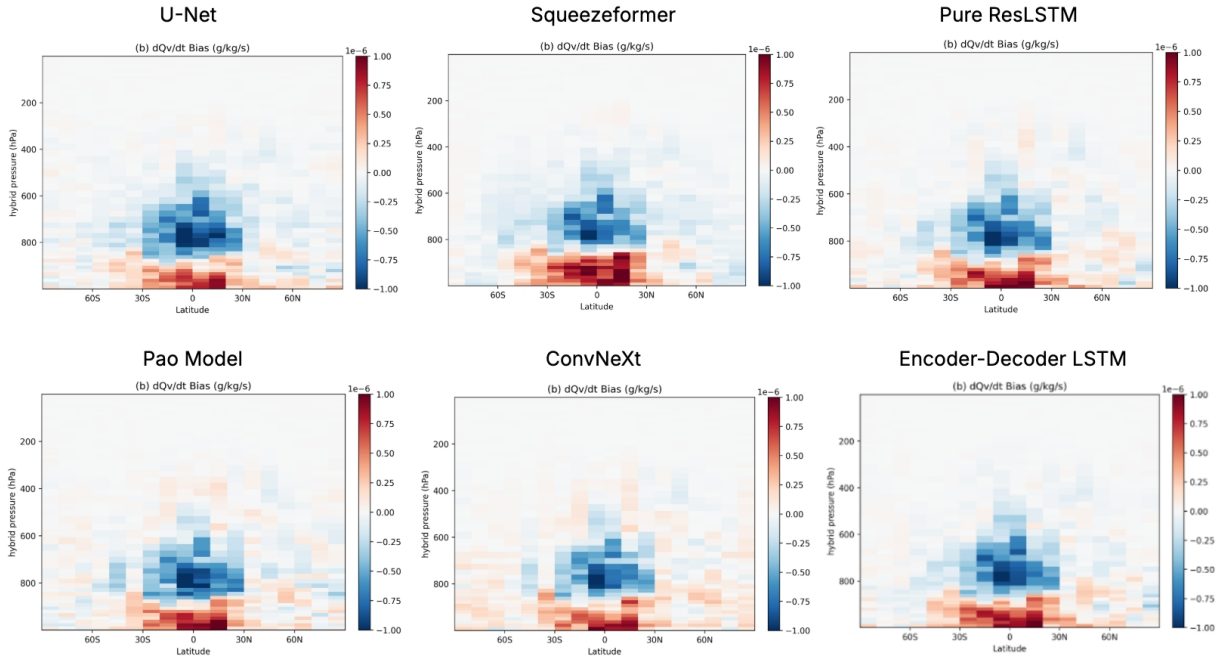

Fig. 4.13. This figure shows offline zonal mean moistening biases for models in the standard configuration.

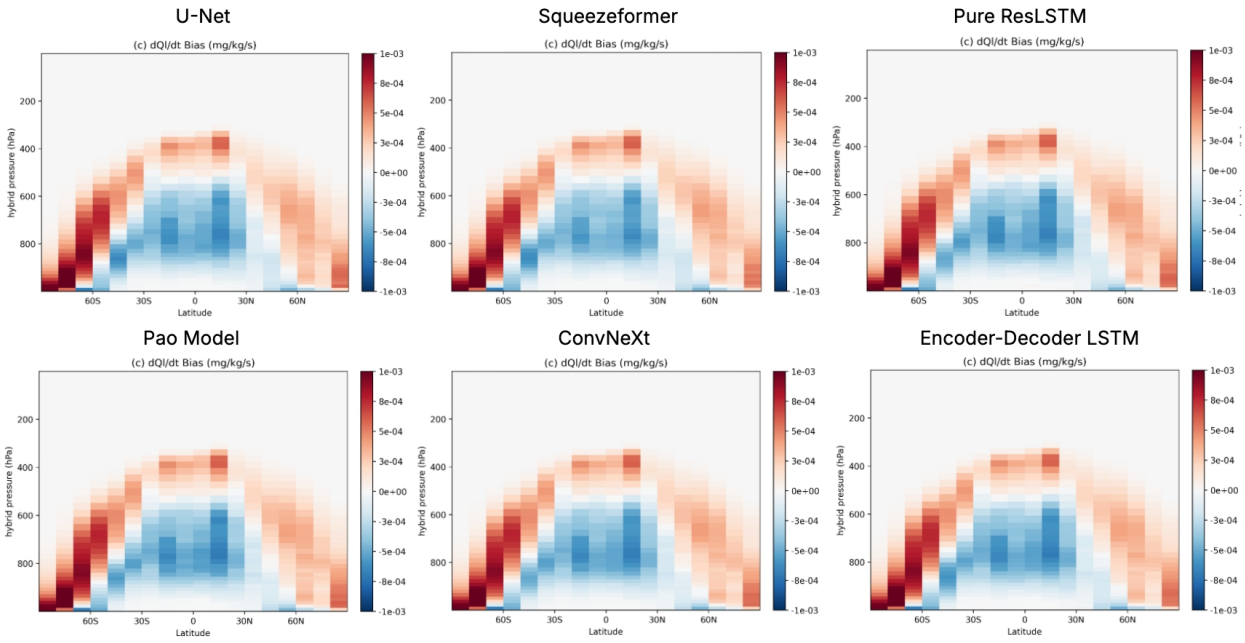

Fig. 4.14. This figure shows offline zonal mean liquid cloud tendency biases for models in the standard configuration.

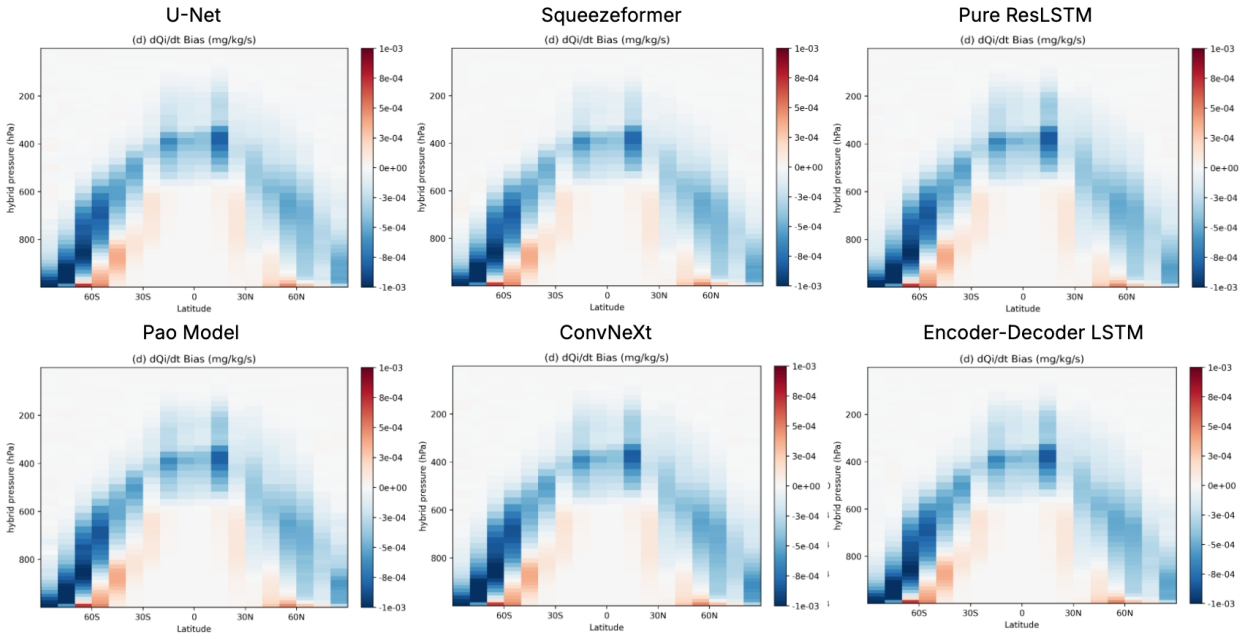

Fig. 4.15. This figure shows offline zonal mean ice cloud tendency biases for models in the standard configuration.

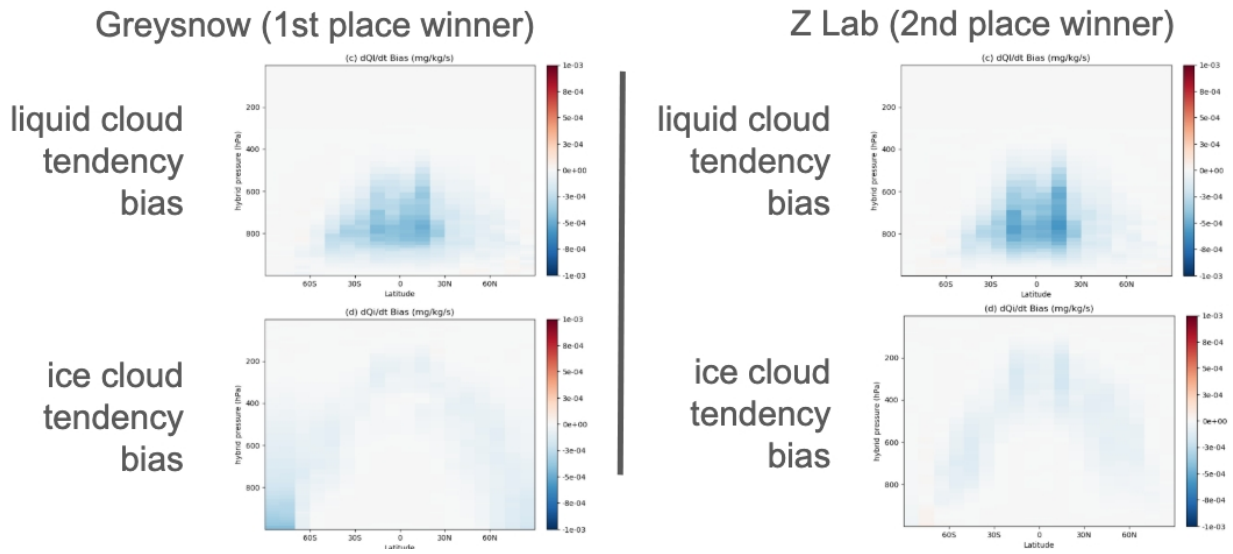

Fig. 4.16. This figure shows offline zonal mean liquid and ice cloud tendency biases for the top two Kaggle teams.

4.4 Discussion and Conclusion

The 2024 ClimSim Kaggle competition introduced the ML parameterization community to a diverse array of architectures that would have otherwise been impractical to explore in the same timeframe. Given the labor-intensive nature of online testing and the necessity of testing ensembles for generalizable insights, exploration of novel architectures that have not been previously tried for ML parameterization is generally a slow, multi-year process. This chapter only evaluates offline and online performance of a subset of the architectures used by the winning teams, and several teams who did not win a cash prize were still generous enough to share their findings in Kaggle discussion posts.

Nevertheless, uniformity in offline and online zonal mean biases suggests that testing increasingly sophisticated architectures is unlikely to be the most important factor for achieving a new state-of-the-art in online results. That being said, the choice of architecture is only one component of ML parameterization design. The design of the machine learning pipeline, curation of training and validation data, and choice of loss function can all have major impacts on the performance of ML parameterizations, and all of these factors were fixed for this study to enable easier comparisons of competing architectures.

We suspect that further progress in reducing persistent online biases will require investigating the state-dependent nature of structural offline biases (e.g. convectively active vs quiescent cases). Skewed distributions of subgrid tendencies may cause issues during training and make the offline fit across different variables and vertical levels particularly sensitive to the choice of loss function. Results from Lin et al. (2025) indicate that the choice of loss function can have "emergent" effects online, but these effects are perhaps only emergent when offline performance is dictated by a scalar loss metric. With a suite of more comprehensive offline diagnostics, moderate sampling across both seeds and architectures, and close inspection of high frequency online output in short simulations, causal patterns for error growth and instabilities may emerge.

Finally, areas for future work highlighted in Hu et al. (2024) like high-resolution online simulation and initial condition variation remain unexplored. The former is particularly essential for operational relevance, as the effective horizontal resolution in this and prior work is $11.5^\circ \times 11.5^\circ$ while the corresponding resolution for the high-resolution version of ClimSim is closer to $1.5^\circ \times 1.5^\circ$. That being said, the persistent biases that exist in the lower-resolution limit are unlikely to simply disappear in the high-resolution limit, and prototyping possible solutions is significantly easier with the former because of the associated reduced training data volume and faster online simulations. We hope that our findings help elucidate the path towards eliminating online biases in the low-resolution limit, thereby opening the frontier for high-resolution experiments and potential operational impact.

Chapter 5

5 Conclusion

5.1 Summary of the work

We are closer than ever to hybrid physics-ML climate simulations that make use of the latest advances in AI to reliably emulate subgrid processes too computationally expensive to resolve explicitly. The work detailed in this dissertation contains major contributions to realizing this eventual vision. In Chapter 2, we sampled hybrid simulation at never-before-seen scale to uncover insights that would have been impossible to confirm with statistical confidence otherwise. We found that using convective memory, turning on batch normalization, and training on multiple climates were unambiguously beneficial for both online accuracy and stability and can be recommended for full-physics subgrid ML parameterization going forward. We also found that, counterintuitively, improved online error can trade off against improved online stability and vice versa, contradicting prior work and intuition that claimed that offline error, online error, and online stability generally improve together. In Chapter 3, we opened up the ML convective parameterization problem to the machine learning research community through the release of an award-winning NeurIPS Datasets and Benchmarks paper and an official follow-up Kaggle competition. The NeurIPS Datasets and Benchmarks paper directly facilitated the start of work that ultimately resulted in a breakthrough in hybrid physics-ML climate simulation Hu et al. (2024). In Chapter 4, we directly build off of Hu et al. (2024) and the results of the Kaggle competition to achieve yet another state-of-the-art result for online zonal mean biases. We also uncover similarities and differences in the offline and online pathologies of different architectures that directly inform the next steps for tackling the online biases that have persisted across multiple independent publications in hybrid physics-ML climate modeling.

5.2 Discussion and Outlook for the Future

These results were made possible by the confluence of researchers and contributors in different fields, sectors, institutions, and countries. They exist in the context of years of effort towards ML parameterizations that might one day work reliably online and generalize to unseen climates. And there is still much more work to be done. Even if the issue of the remaining online biases are solved and the best architectures for this problem are identified, there is still substantial work left before a climate model with ML convective parameterizations can be used for operational long-term climate projections. High-resolution hybrid simulations for ClimSim remain unexplored in the literature. Furthermore, a future version of ClimSim-Online needs to be released for continued progress once that challenge is also solved. This new version should contain multiple climates, scaling laws, aerosol-cloud interaction, stochastic online benchmarks, and a comprehensive variable list that allows for mass and energy conservation as well as (if possible) the separation of different coarse-grained physical processes. Finally, because of a lack of long-term support for MMF, whatever lessons learned in this work need to eventually be translated to ML parameterizations that learn from GCRMs without clean-scale separation.

In theory, realizing this vision is technically possible, but the prospect of doing so is under existential threat. Without basic science funding, researchers will be unable to continue this work and progress will grind to a halt. It is tempting to think that there is a clean distinction between science and politics. On paper, science is an objective pursuit of truth whereas politics is corrupted by subjective beliefs, desires, and competing, mutually exclusive interests. In reality, science exists as it does today because democratically elected politicians made it a priority and were convinced of its benefits to society. Because this was generally the case for nearly a century, it has been easy to take this situation for granted and come to the conclusion that science and politics could reside in hermetically sealed spheres of influence. As much as this dissertation is intended to be a culmination of previous work completed over the course of a 5-year PhD, it is also a clarion call to you, the reader, to

recognize that the future of scientific innovation cannot be divorced from the political will to sustain it. Just as democratic decisions historically catalyzed scientific progress, so too must they now confront the escalating threats to its survival. This dissertation, while rooted in rigorous academic inquiry, serves as both a product of that system and a witness to its fragility—a dual identity reflecting the reality that truth, in the end, is not immune to the forces that shape the world beyond the lab.

Supplementary Information

The following text is Supplementary Information for Chapter 2 reproduced with permission from:

Lin, Jerry, Sungduk Yu, Liran Peng, Tom Beucler, Eliot Wong-Toi, Zeyuan Hu, Pierre Gentine, Margarita Geleta, and Mike Pritchard. 2025. "Navigating the Noise: Bringing Clarity to ML Parameterization Design with $\mathcal{O}(100)$ Ensembles." Journal of Advances in Modeling Earth Systems 17 (4): e2024MS004551.

Contents of this file

1. Text S1 to S3
2. Tables S1 to S2
3. Figures S1 to S35

Introduction

Text S1 provides additional detail regarding preprocessing and imperfections in SP-CAM3 simulation data, and Text S2 explains how offline and online error were calculated. Text S3 provides more details on sample size estimation. Table S1 is analogous to Table 3 in the main manuscript, except we report statistics for offline moistening and online moisture instead of offline heating and online temperature. Table S2 shows offline heating and moistening error using a multiclimate, out-of-distribution test set along with the corresponding Spearman correlations to "in-distribution" online error. In this context, in-distribution refers to the fact that the prescribed SSTs are the same in offline training and online testing. Figure S1 shows the Benjamini-Hochberg correction being applied to all hypothesis tests conducted in this study. Figure S2 shows the percent change in in-distribution offline test error with a doubling and quadrupling in size for the offline test set. Figure S3 is analogous to Figure 2 in the main text, except the offline error curves are for the multiclimate, out-of-distribution test set. Figures S4 to S12 show online error growth across configurations,

colorized by offline error. Figure S13 (S14) shows median (mean) online error growth for each configuration. Figure S15 shows histograms of number of months completed without crashing for hybrid simulations across each configuration. Figures S16 to S24 show offline error vs. online error for all configurations. Figure S25 shows offline vs. online error boxplots for all configurations. Figures S26, S27, S28, S29, S30, and S31 show the effect of dropout, leak (in Leaky ReLu), learning rate, parameter count, choice of optimizer, and batch normalization on offline and online error across all NNs that did not crash after marginalizing out choice of configuration. Figure S32 shows the effects of the aforementioned model properties on stability after marginalizing out choice of configuration. Figure S33 shows mean online temperature and moisture error growth for hybrid simulations with and without batch normalization. Figures S34 and S35 show Laplacian distributions being fit to histograms of the heating and moistening tendencies in the training data, respectively.

Text S1.

Without further preprocessing, the formula for number of samples for the training data after subsampling longitudinally would be $N_{samples} = N_{days} \times 48 \times 64 \times 128 \times \frac{1}{8}$ where 48 corresponds to timesteps per day, 64 corresponds to number of latitudes, 128 corresponds to the number of longitudes, and 1/8 corresponds to longitudinal subsampling. The formula for validation data is similar; however 19 longitudes are subsampled as 128 is not divisible by 7. Additionally, the final timesteps of every month and the simulation itself are removed because of strange behavior at the final timestep of SP-CAM3 simulations and the modular structure of the preprocessing code.

This yields $365 \times 48 \times 64 \times 128 \times (1/8) - 13 \times 64 \times 128 \times (1/8) = 17,927,168$ samples for the training data and $61 \times 48 \times 64 \times 19 - 3 * 64 * 19 = 3,556,800$ samples for the validation data. Since the offline test data is not subsampled, the number of samples becomes $7 \times 48 \times 64 \times 128 = 2,752,512$ samples for offline test data. For the multiclimate configuration, months 2 through 5 were similarly subsampled across -4K, +0K, and +4K climates. To control for the data volume, only the first 5,975,723 samples were kept for each climate. Although this

yields one additional sample in the training data compared to the other configurations, we do not believe the addition of a single example significantly impacts our results.

Text S2.

Both offline and online error are mass-weighted using area and pressure (from the SPCAM3 reference simulation). However, offline RMSE is calculated across all samples while online RMSE error is first calculated on a simulation-month basis and, for the main manuscript, averaged across all months for runs that do not crash.

Offline heating and moistening RMSE are both defined as:

$$\sqrt{\sum_i W_i \circ (\hat{Y}_i - Y_i)^2}$$

where:

- $\circ$ denotes element-wise multiplication.
- $(\cdot)^2$ indicates squaring each element individually.
- $\hat{Y} \in \mathbb{R}^{336\times30\times64\times128}$: the NN prediction tensor.
- $Y \in \mathbb{R}^{336\times30\times64\times128}$: the ground truth tensor.
- $W \in \mathbb{R}^{336\times30\times64\times128}$: the weight tensor, whose values sum to one.
- i: the flattened index over all dimensions of the tensors: time (336 timesteps), vertical levels (30), latitude (64), and longitude (128).

Online temperature and moisture RMSE are both defined as:

$$\sqrt{\sum_i W_i \circ (Error_i^*)^2}$$

where:

$$\text{Error}[m] = \frac{\sum_{\text{lon}=1}^{8} \sum_{t=1}^{T_m} \hat{y}_{m,t,\text{lon}}}{8 \cdot T_m} - \frac{\sum_{\text{lon}=1}^{8} \sum_{t=1}^{T_m} y_{m,t,\text{lon}}}{8 \cdot T_m}, \quad \text{for } m = 1, \ldots, 12$$

and:

- $\hat{y}_{m,t,lon}$ refers to coupled NN simulation output for a given month, day of month, and longitude
- $y_{m,t,lon}$ refers to reference SPCAM 3 simulation output for a given month, day of month, and longitude
- T_m denotes number of days in month m
- $\circ$ denotes element-wise multiplication.
- $(\cdot)^2$ indicates squaring each element individually.
- Error[m] refers to the unweighted online error for a given month m, averaged over eight longitudes and T_m days.
- $Error^* \in \mathbb{R}^{N_M \times 30 \times 64}$: the error tensor with N_M denoting number of months fully integrated without crashing. It is formed by concatenating monthly errors (i.e. Error[m] for a given month) along the time dimension.
- $W \in \mathbb{R}^{N_M \times 30 \times 64}$: the mass-weight tensor that, when summed along the vertical level and latitude dimensions, results in a vector of ones $\in \mathbb{R}^{N_M}$. Again, N_m denotes the number of months fully integrated without crashing.
- i: the flattened index over the vertical level and latitude dimensions of the tensors
- the final result is a vector $\in \mathbb{R}^{12}$

Only eight longitudes are used for online error because every 16th longitude is automatically subsampled in the semi-automated pipeline to mitigate storage consumption. For online simulations that crash, the formula is the same except the time dimension for both the weight and error tensors is limited to the number of months completed without crashing.

Text S3. In line with work from Noether1987-po and Hamilton1991-xa for power calculations for Mann Whitney U-Tests, we use the following formula to estimate necessary sample sizes:

$$n^* = \left\lceil \left\lceil \frac{\left(\Phi^{-1}(\kappa_0) + \Phi^{-1}(1-\alpha)\right)^2}{6(P-.5)^2} \right\rceil * \frac{1}{\gamma} \right\rceil$$

where n^* is our sample size estimate, $\kappa_0 = .8$ is our desired power, $\alpha = .05$ is our significance level, $\gamma = 141/330$ is our assumed survival rate (i.e., the surival rate of the standard configuration), and P is $P(Y > X)$, where X and Y are realizations from the populations that our configurations are each sampling from. Φ^{-1} is the inverse of the cumulative distribution function of a normal distribution, and the ceiling notation indicates rounding up to the next integer. In the context of the sample size estimate that uses the difference in online error when omitting ozone, we get a U-statistic of 8,363. This means $P \approx 42.1\%$ and $n^* \approx 384$.

Configuration	**Survival**	**Online Error (g/kg)**	**Offline Error (g/kg/day)**	**Spearman's ρ**
standard	42.7%	.423 g/kg	1.76 g/kg/day	**.811**
specific humidity	**-26.1%**	+.032 g/kg	**+.00789 g/kg/day**	**.806**
no memory	**-32.7%**	**+.246 g/kg**	**+.256 g/kg/day**	.306
no wind	+1.82%	-.004 g/kg	+.00412 g/kg/day	**.716**
no ozone	**+13.3%**	+.0354 g/kg	+.00252 g/kg/day	**.847**
no zenith angle	**+11.2%**	-.0357 g/kg	+.00134 g/kg/day	**.798**
MAE	**+32.4%**	**+.335 g/kg**	**-.00952 g/kg/day**	**.790**
no dropout	**-14.2%**	**-.0850 g/kg**	**-.0545 g/kg/day**	.194
multiclimate	**+27.3%**	**-.0535 g/kg**	-.00360 g/kg/day	**.773**

Table S1: This table is analogous to Table 3 in the main manuscript except we report statistics for offline moistening error and online moisture error instead of offline heating error and online temperature error.

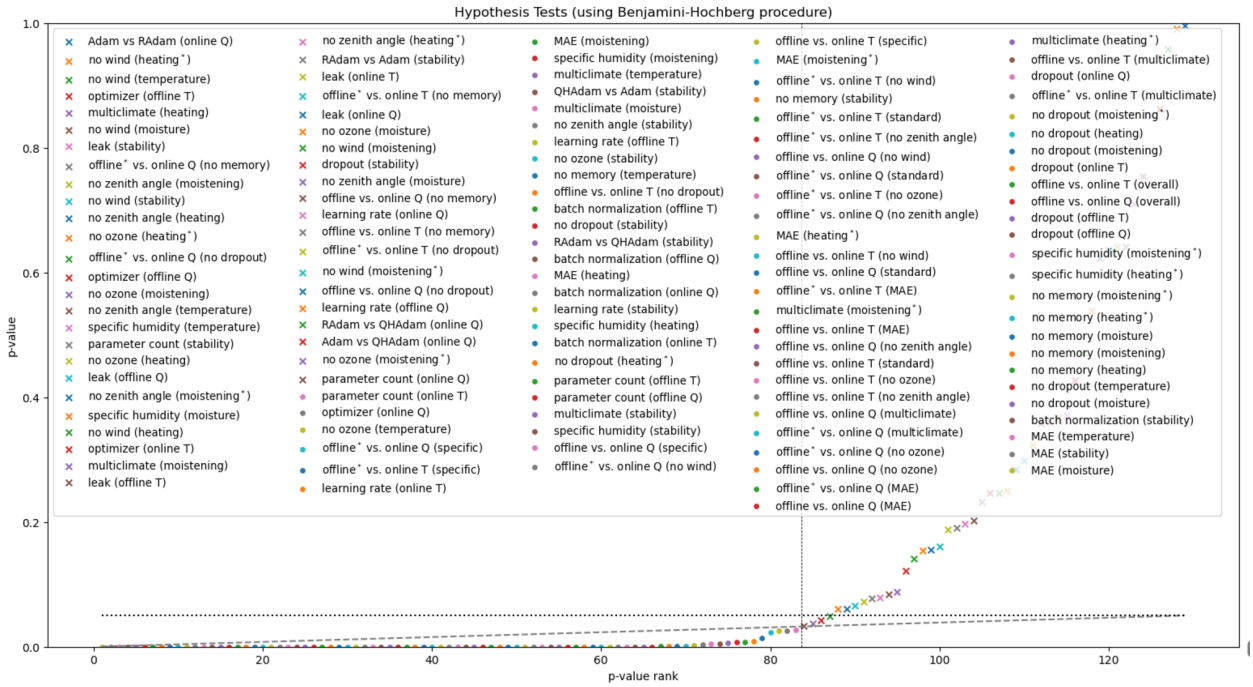

Fig. S1. This figure illustrates the Benjamini-Hochberg correction being applied to ranked unadjusted p-values. Dots to the left of the dashed line correspond to statistically significant p-values. The p-value associated with the permutation test comparing online temperature error from the standard and no ozone configurations is the largest statistically significant p-value. Heating and moistening correspond to offline error while temperature and moisture correspond to online error. Labels with an asterisk indicate inference on a multiclimate offline test set.

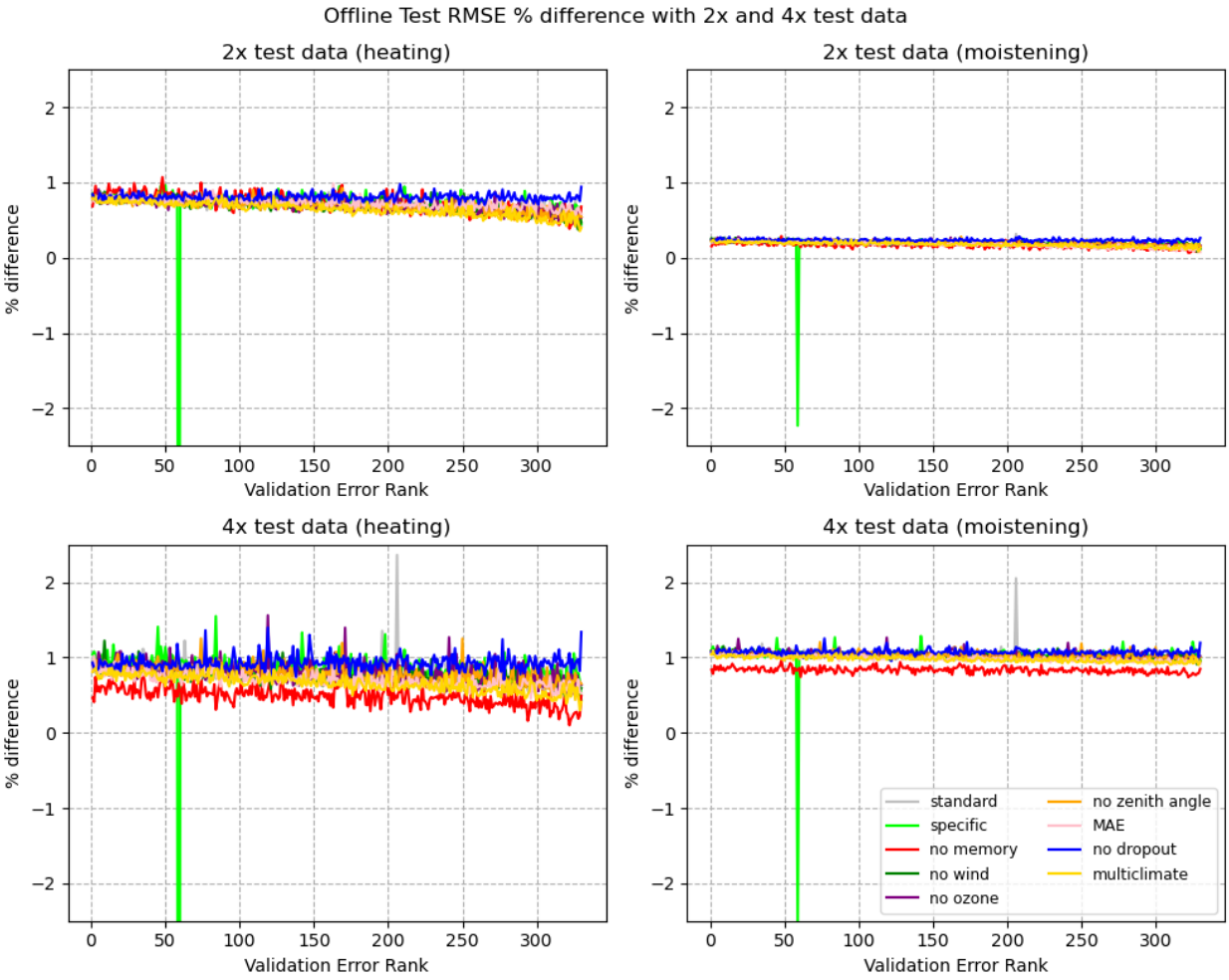

Fig. S2. This shows the percent change in offline test heating and moistening RMSE when doubling and quadrupling the test data.

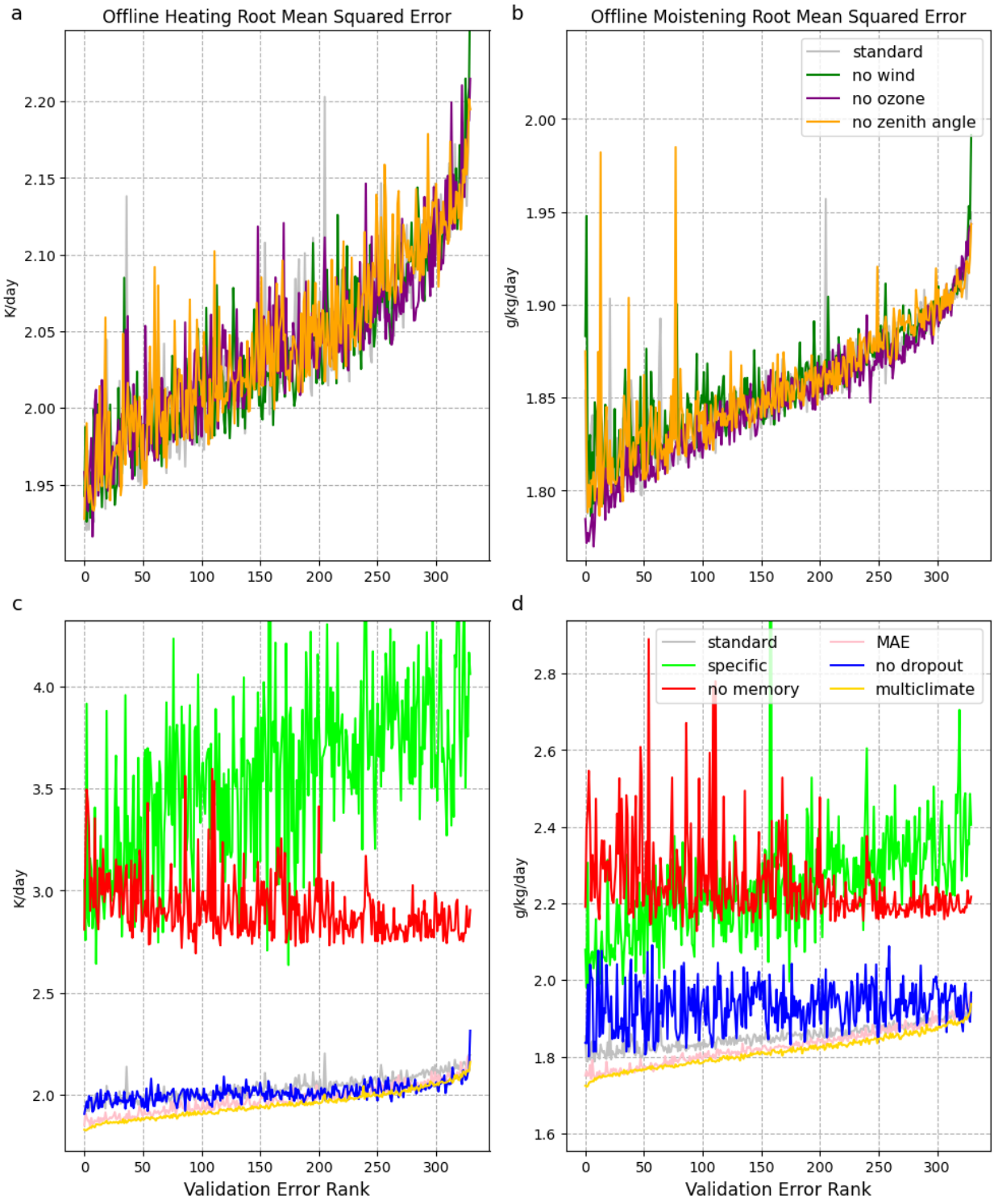

Fig. S3. This figure is analogous to Figure 2 in the main text, showing the offline RMSE for all 2,970 models from nine configurations on a multiclimate test set. S2c and S2d show configurations with statistically distinct average RMSE from the standard configuration.

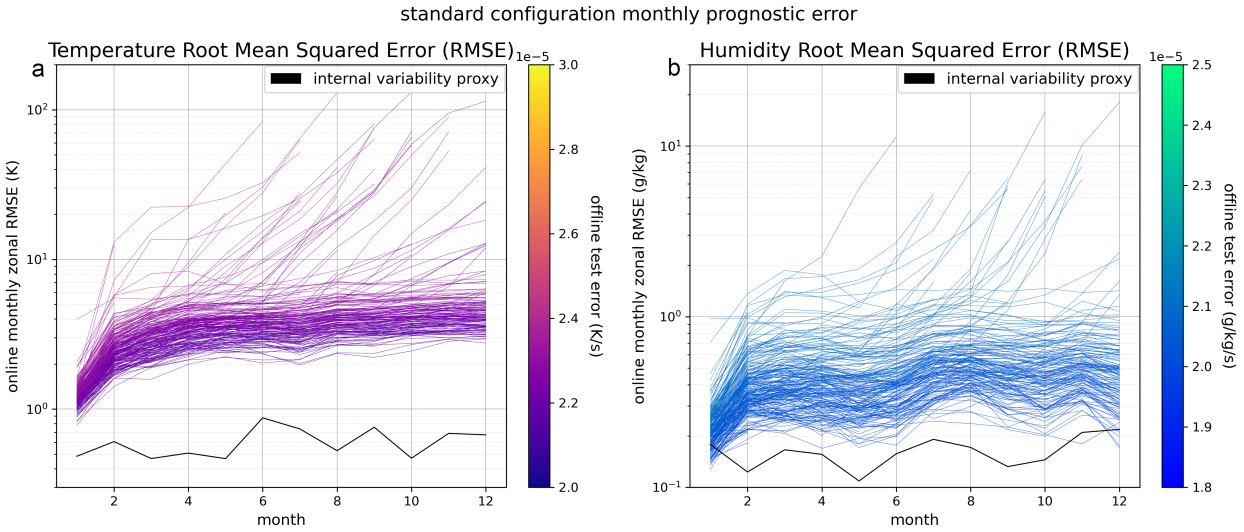

Fig. S4. This figure shows online error over time across all 330s NNs trained in the standard configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

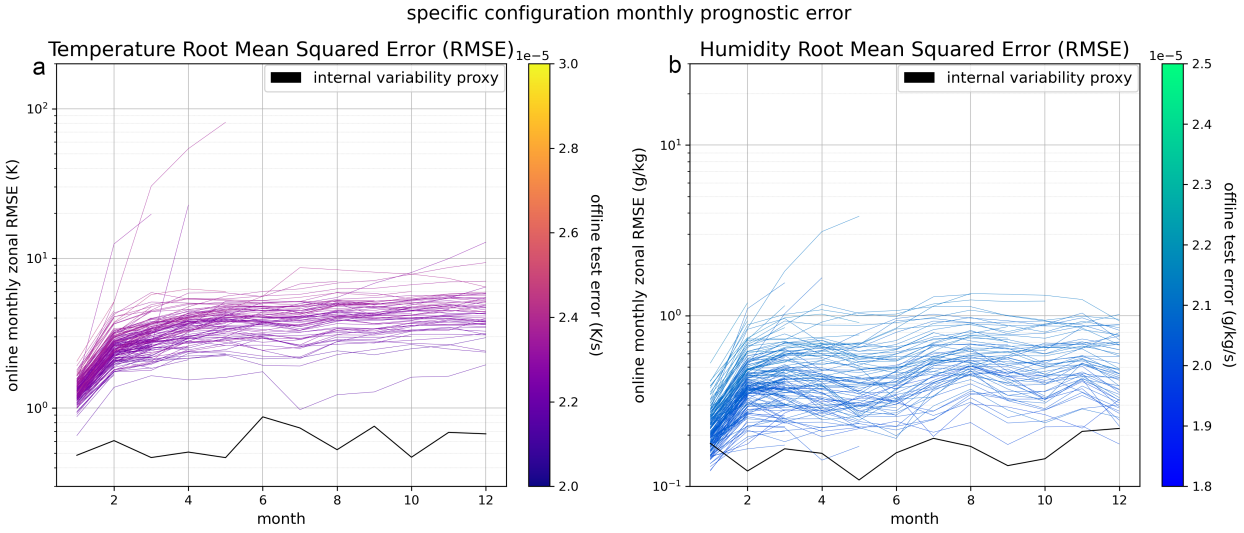

Fig. S5. This figure shows online error over time across all 330s NNs trained in the specific configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

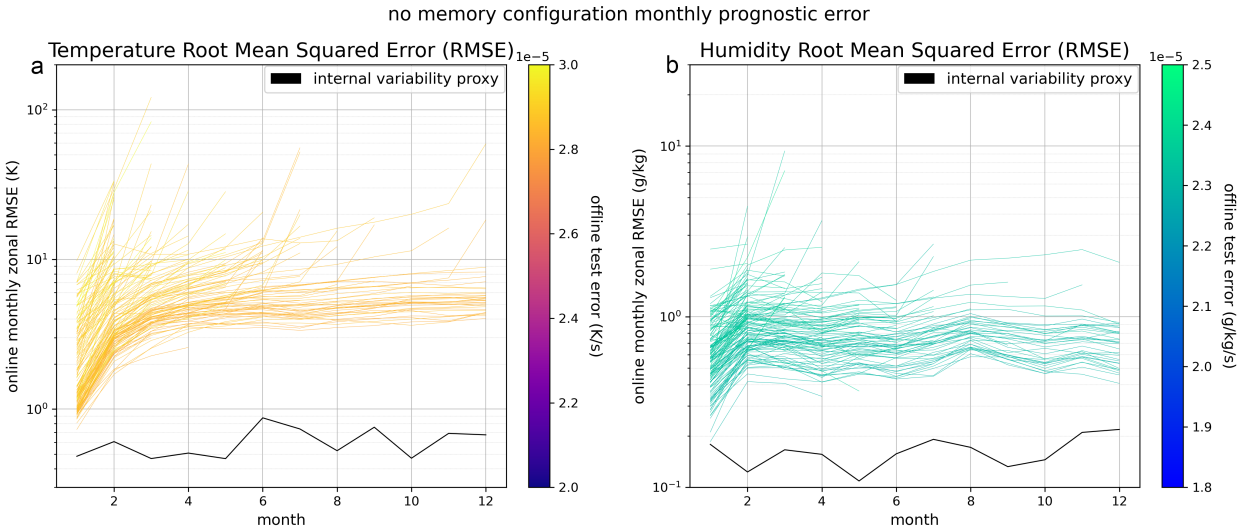

Fig. S6. This figure shows online error over time across all 330s NNs trained in the no memory configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

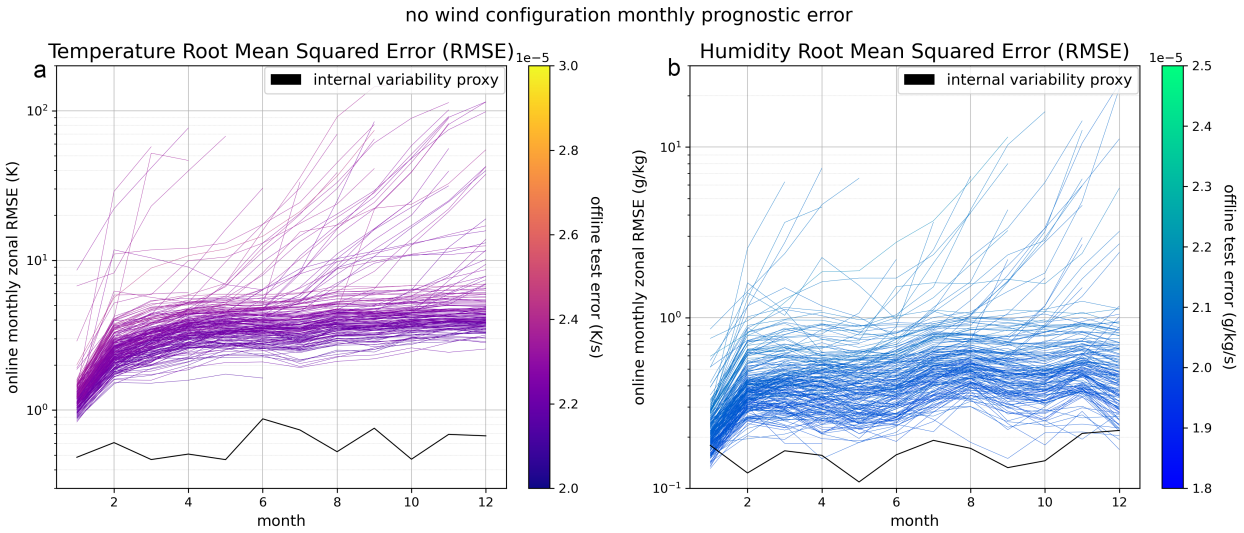

Fig. S7. This figure shows online error over time across all 330s NNs trained in the no wind configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

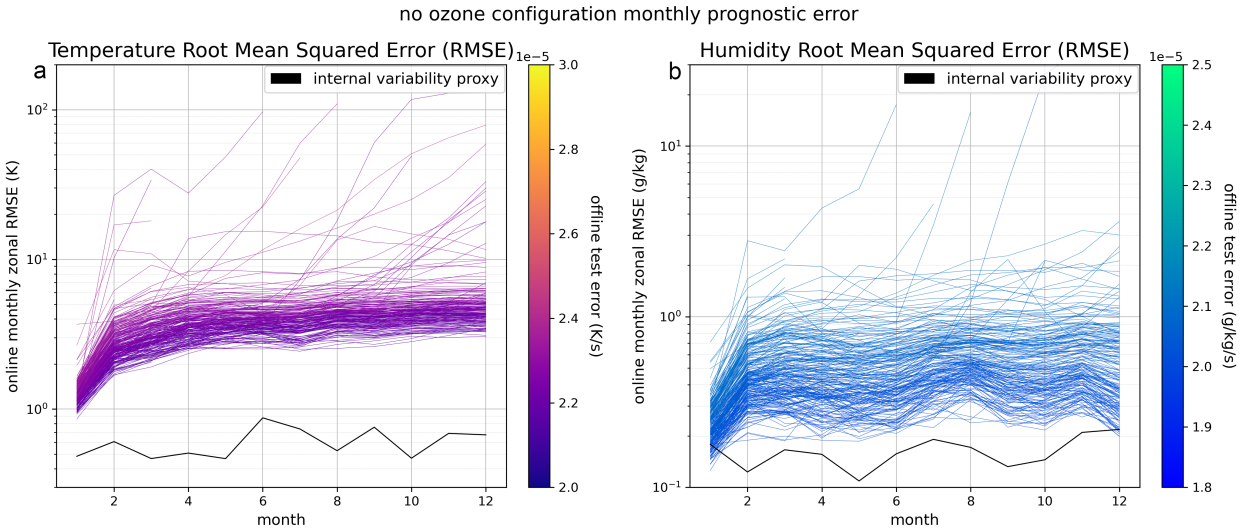

Fig. S8. This figure shows online error over time across all 330s NNs trained in the no ozone configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

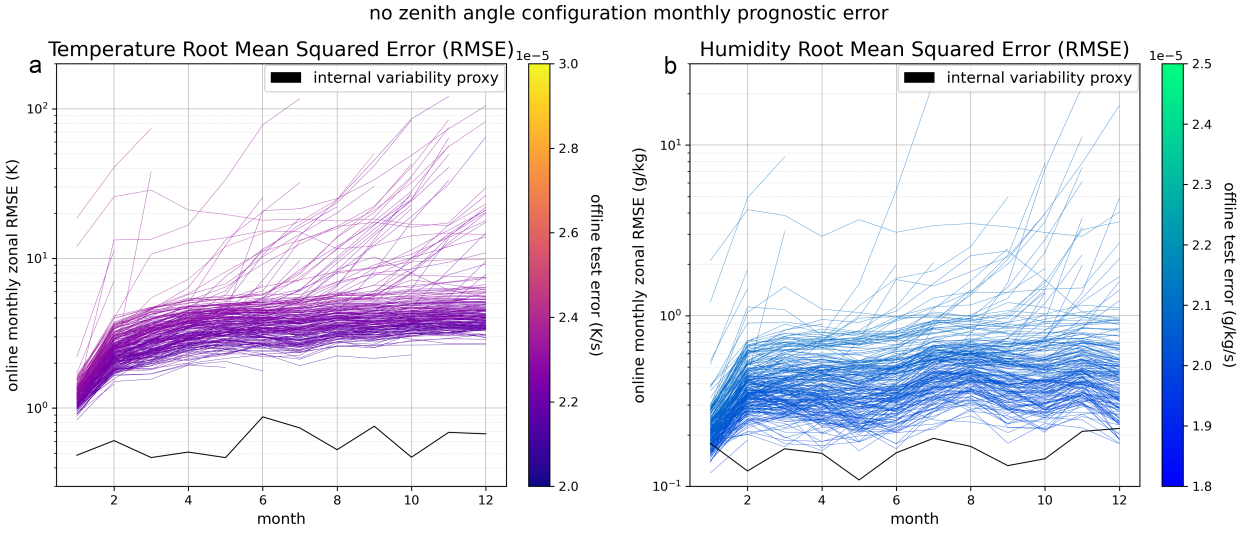

Fig. S9. This figure shows online error over time across all 330s NNs trained in the no zenith angle configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

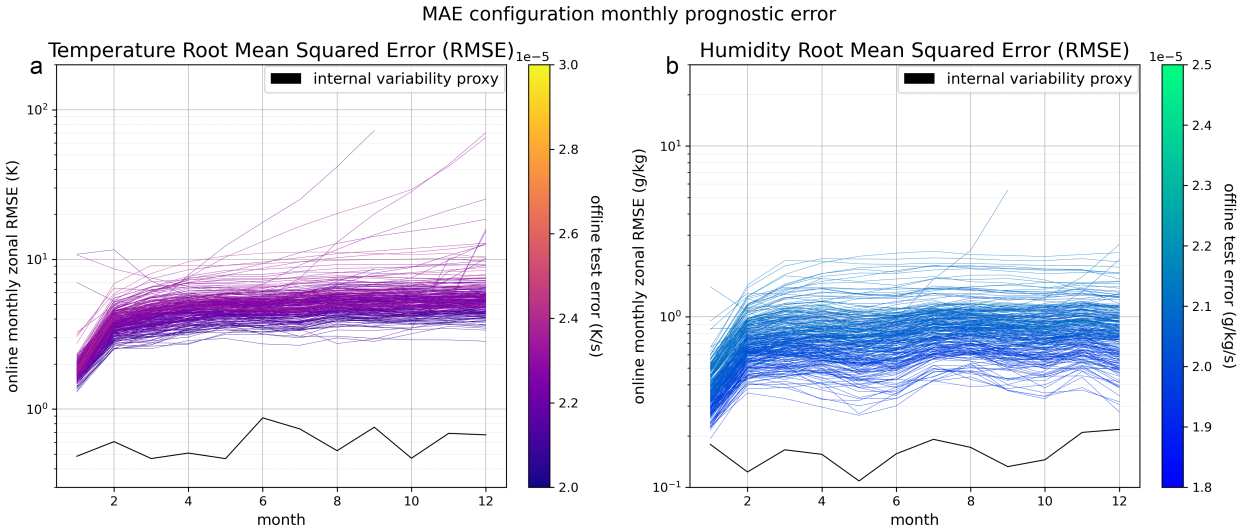

Fig. S10. This figure shows online error over time across all 330s NNs trained in the MAE configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

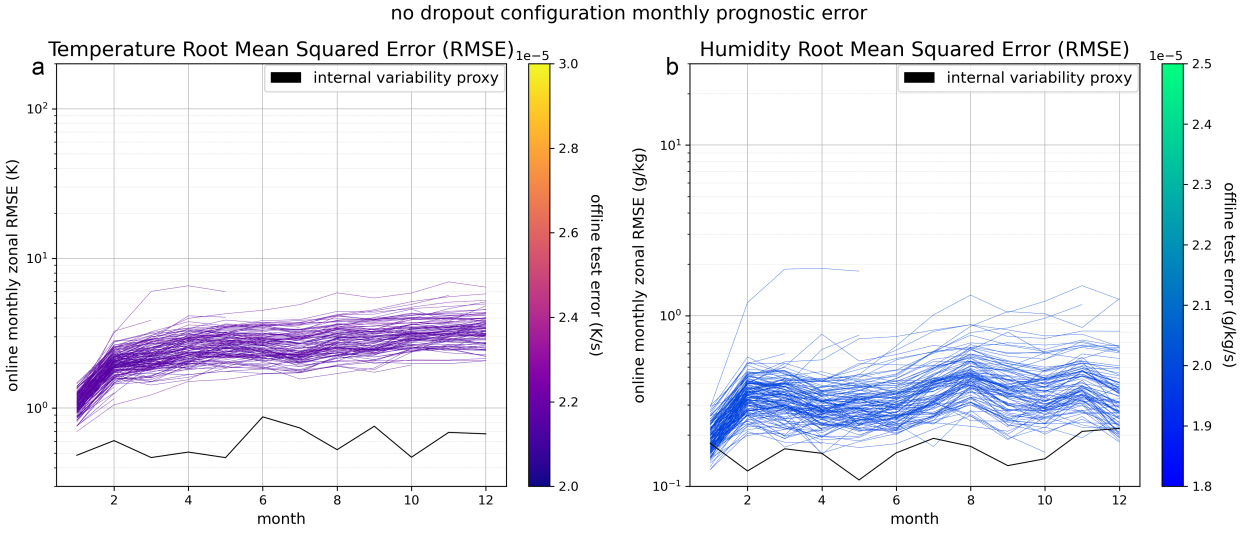

Fig. S11. This figure shows online error over time across all 330s NNs trained in the no dropout configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

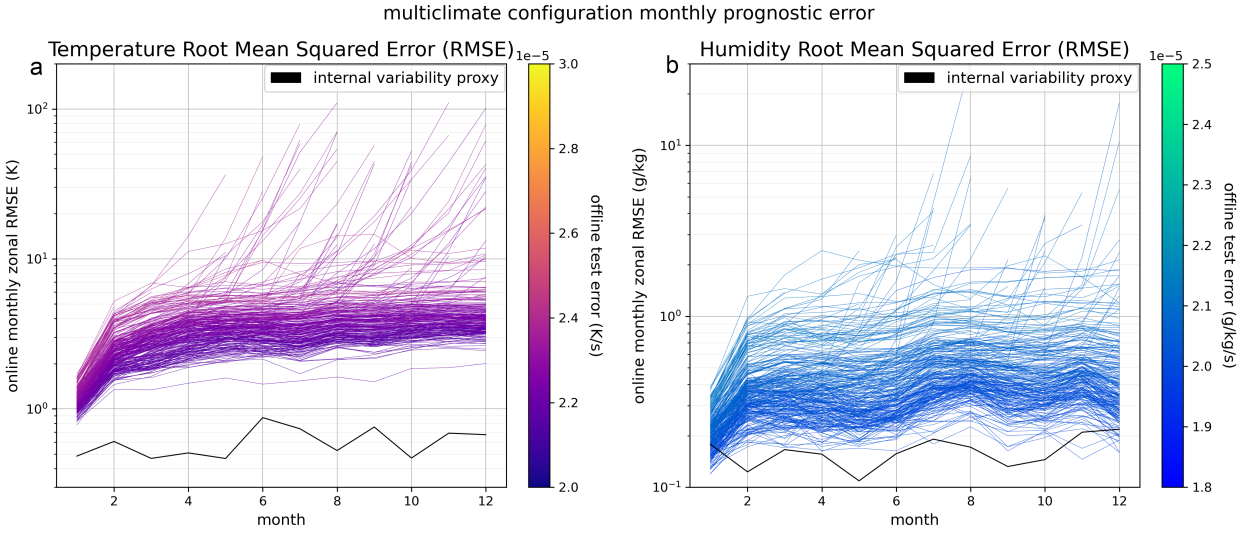

Fig. S12. This figure shows online error over time across all 330s NNs trained in the multiclimate configuration. An internal variability proxy that is calculated via the error of a 31-day lagged version of the reference simulation is included for comparison.

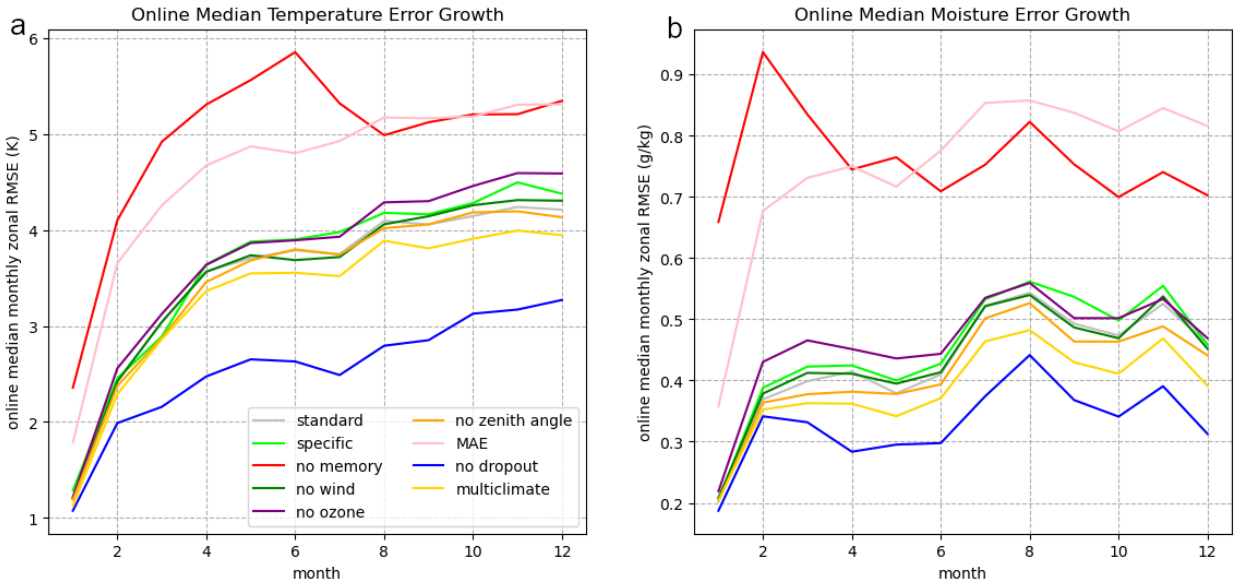

Fig. S13. This figure shows median online error for both temperature and moisture over time for each configuration. Ensemble-median online error is only recorded for simulations that do not crash, and they are calculated across the entire duration of the simulations.

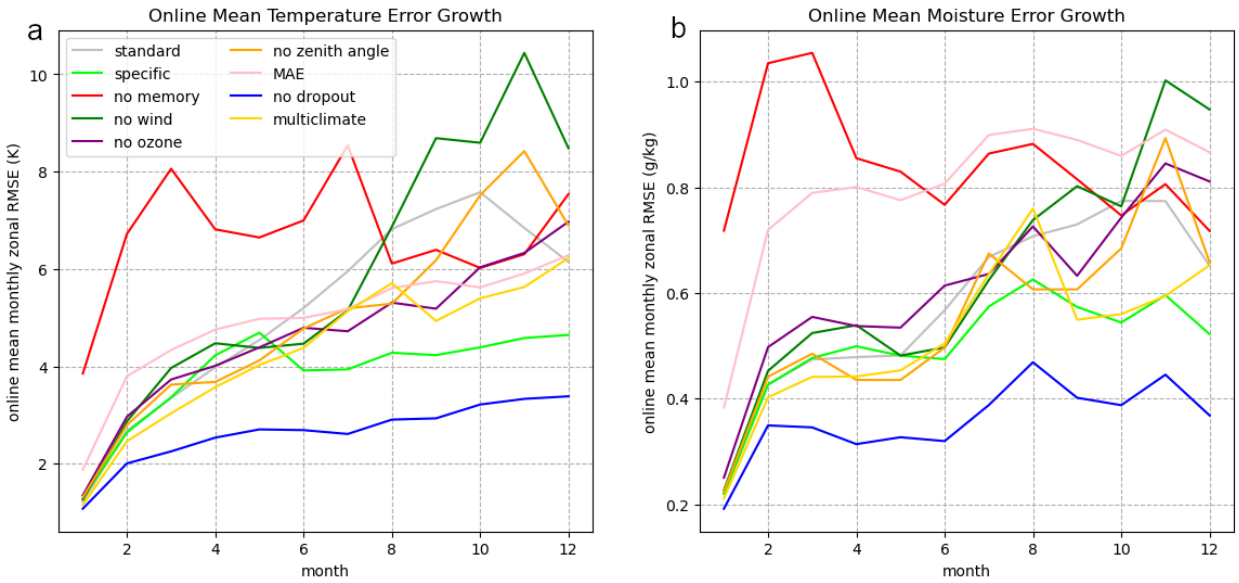

Fig. S14. This figure shows mean online error for both temperature and moisture over time for each configuration. Ensemble-mean online error is only recorded for simulations that do not crash, and they are calculated across the entire duration of the simulations.

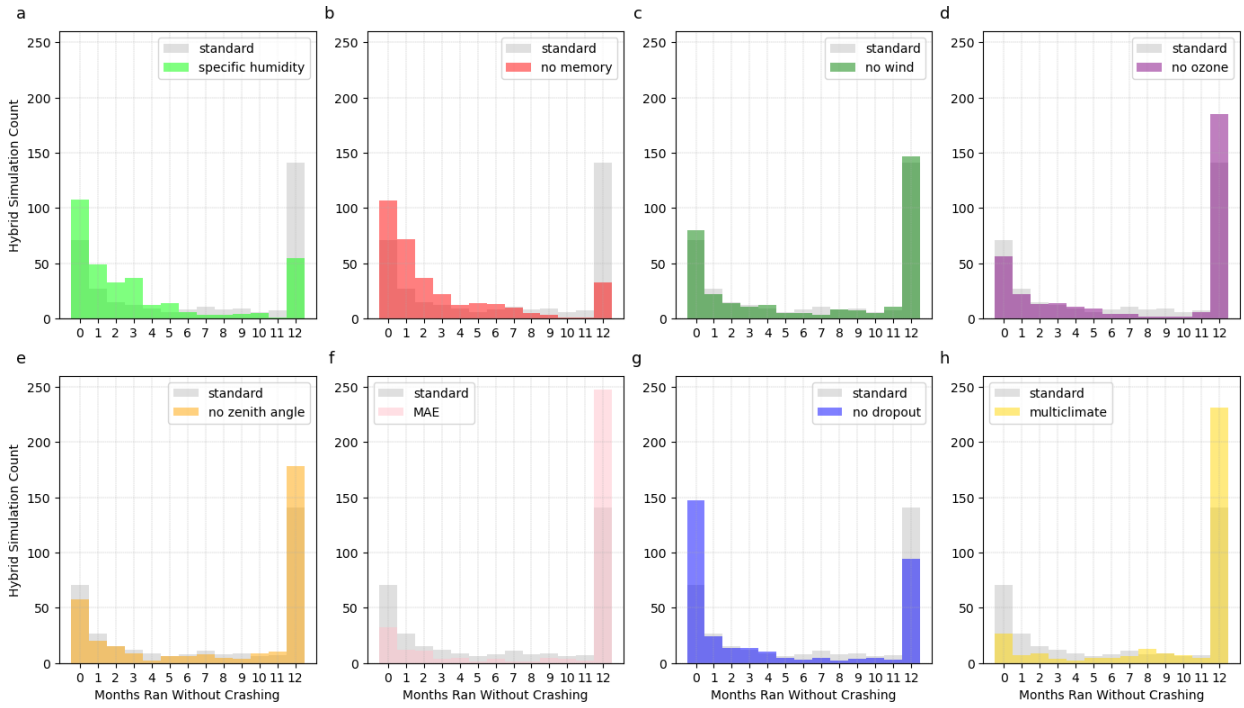

Fig. S15. This figure shows the number of simulation months integrated for each configuration compared to the standard configuration.

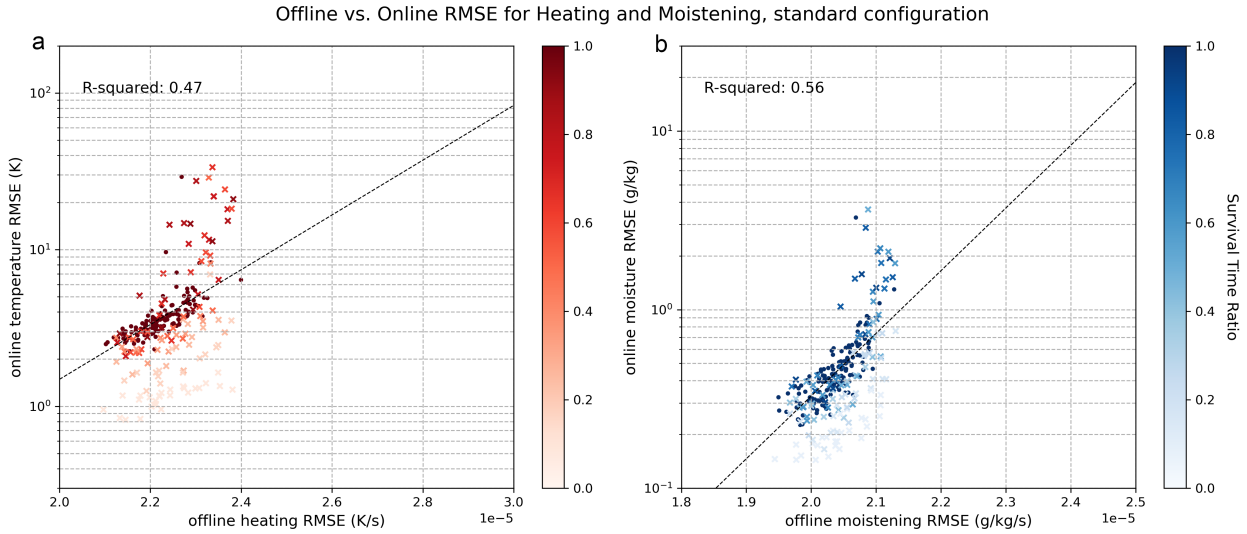

Fig. S16. This figure shows offline error vs. online error across all 330 models in the standard configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

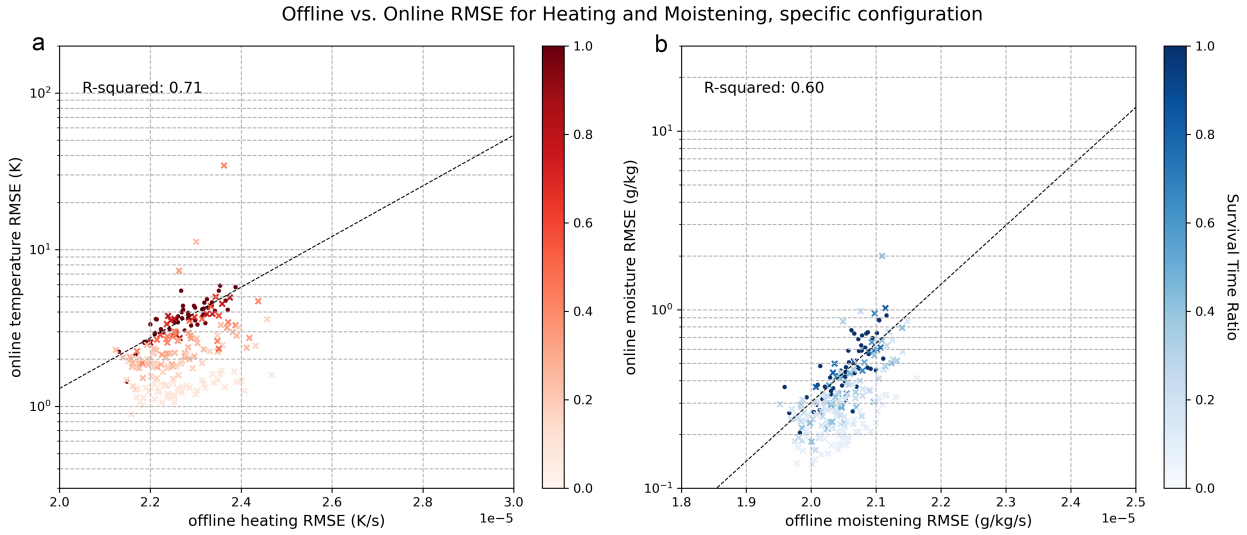

Fig. S17. This figure shows offline error vs. online error across all 330 models in the specific configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

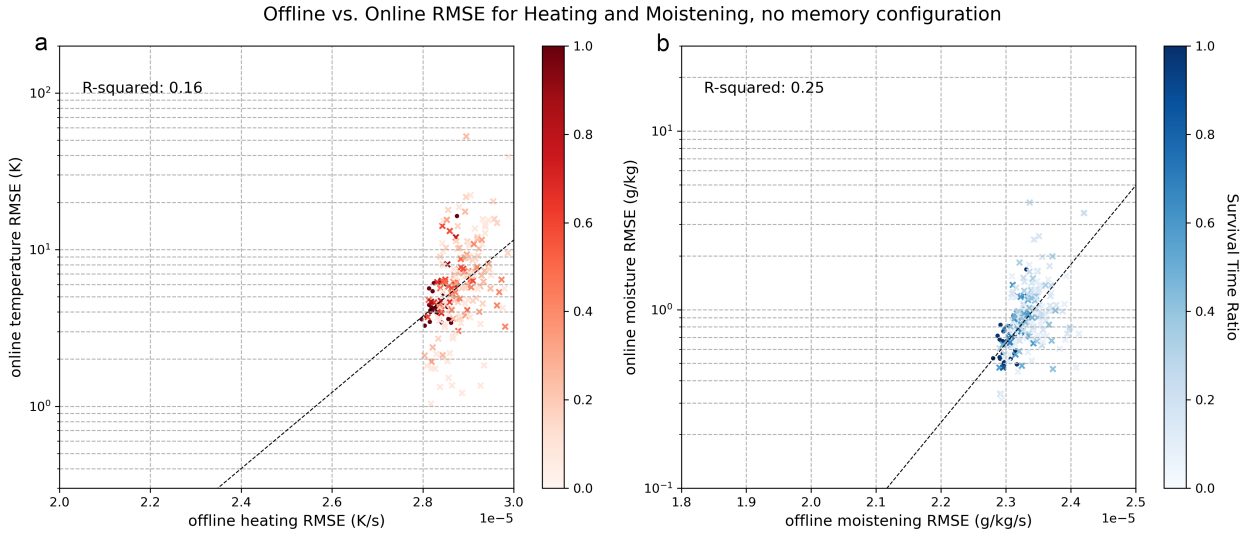

Fig. S18. This figure shows offline error vs. online error across all 330 models in the no memory configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

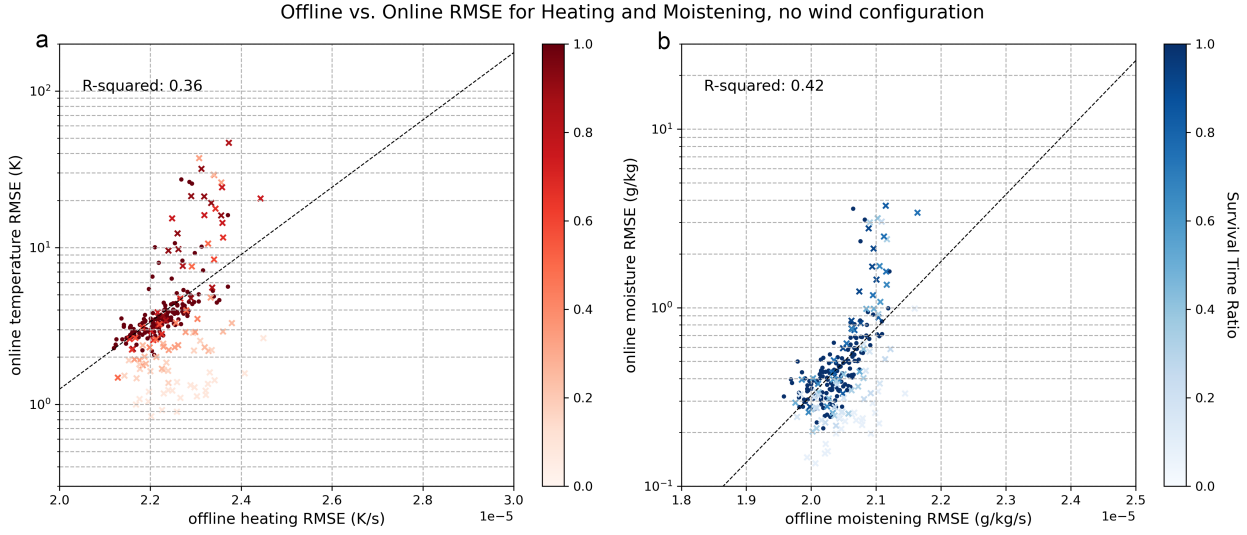

Fig. S19. This figure shows offline error vs. online error across all 330 models in the no wind configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

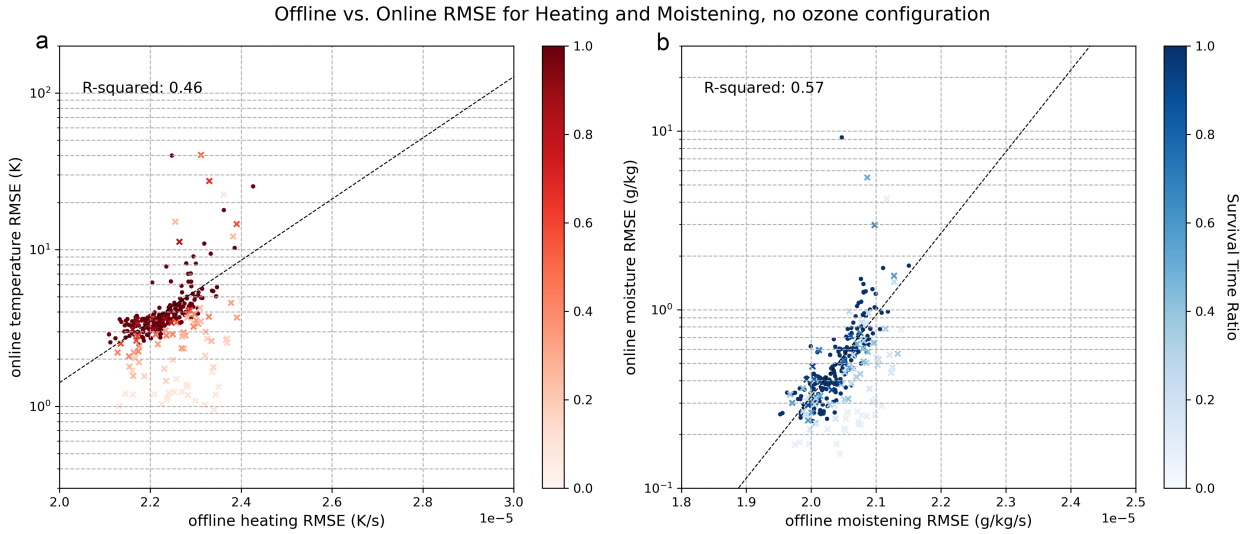

Fig. S20. This figure shows offline error vs. online error across all 330 models in the no ozone configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

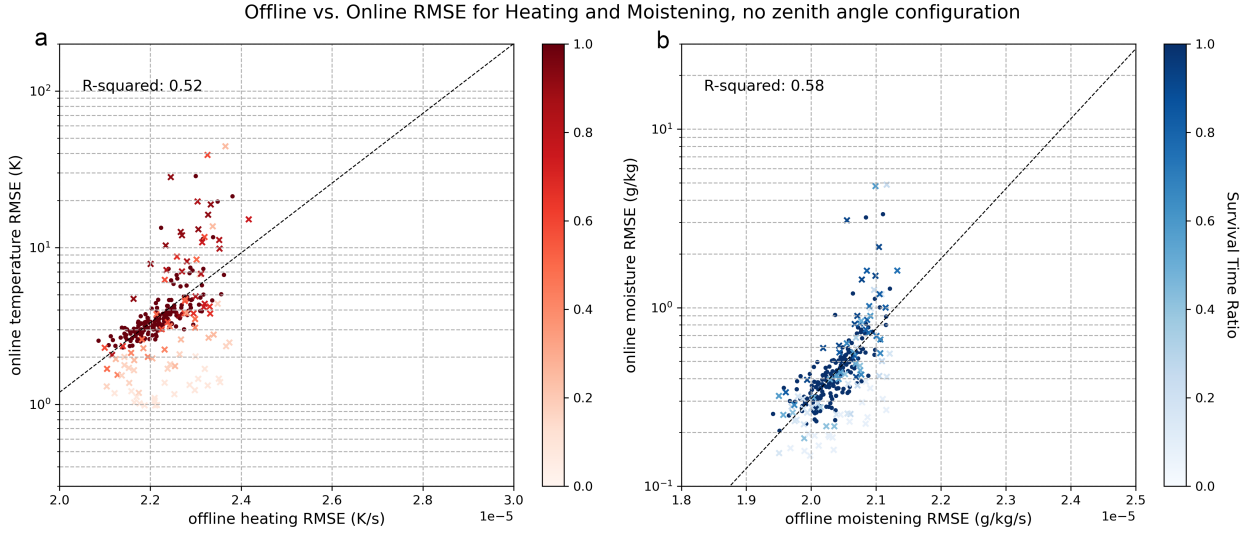

Fig. S21. This figure shows offline error vs. online error across all 330 models in the no zenith angle configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

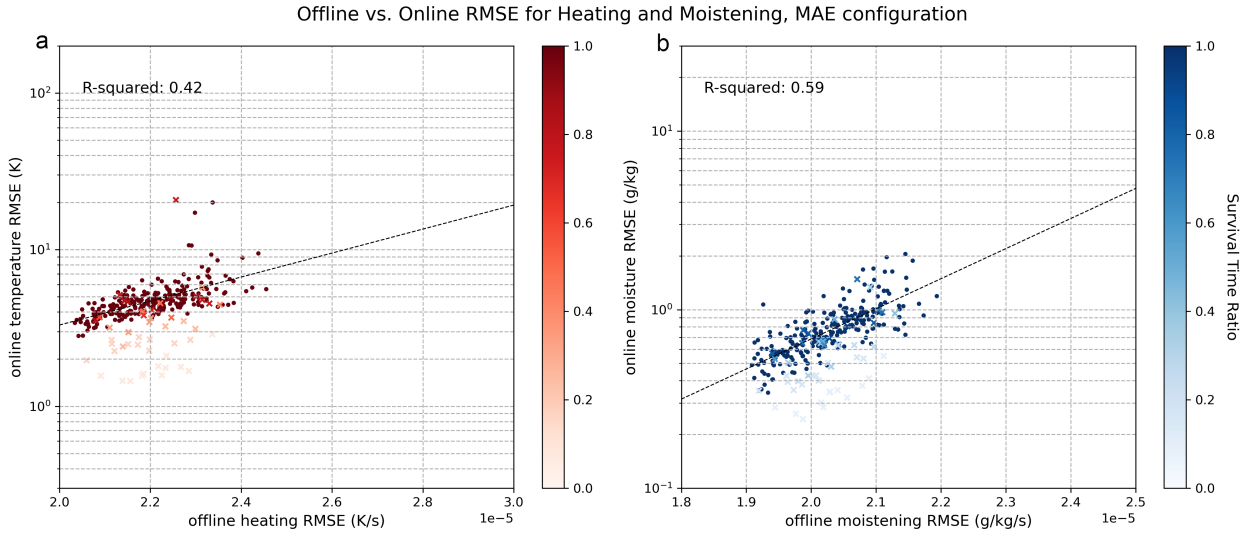

Fig. S22. This figure shows offline error vs. online error across all 330 models in the MAE configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

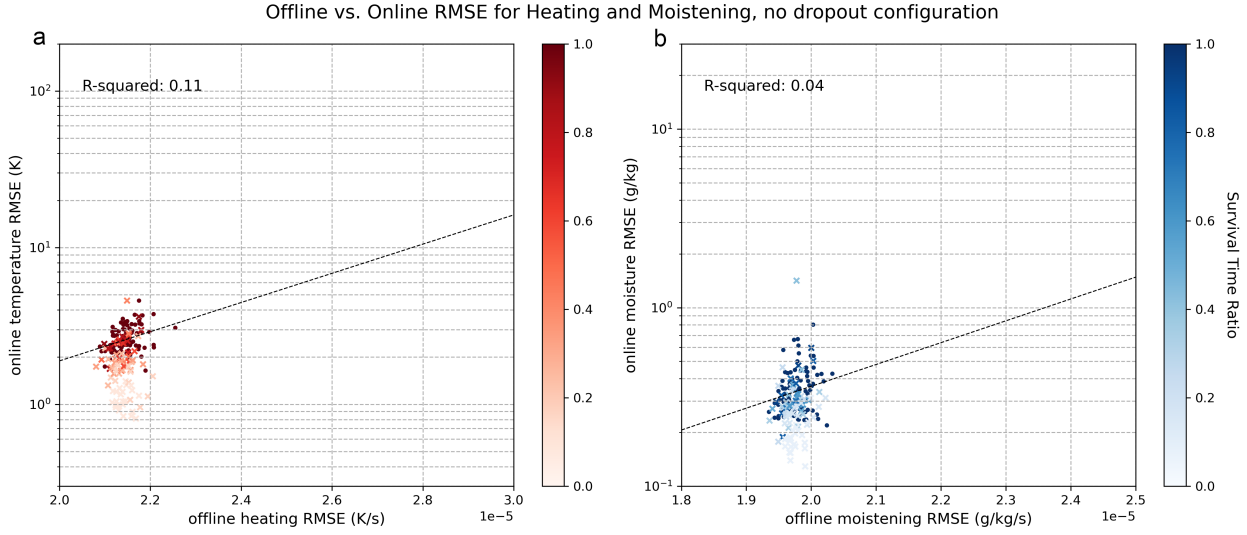

Fig. S23. This figure shows offline error vs. online error across all 330 models in the no dropout configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

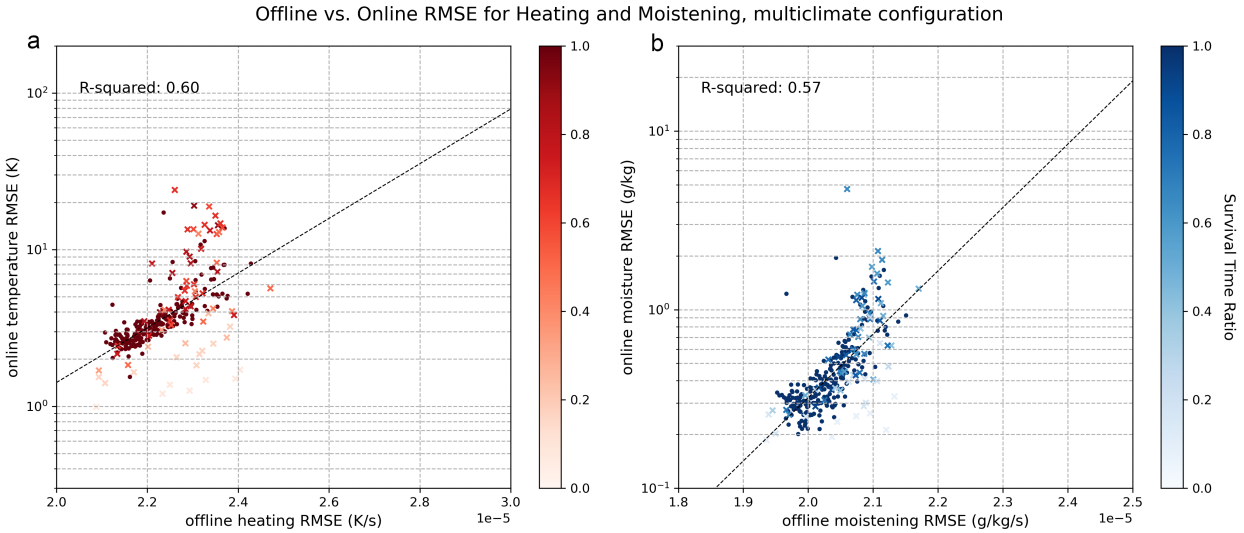

Fig. S24. This figure shows offline error vs. online error across all 330 models in the multiclimate configuration. Models that crash mid-simulation are denoted with an 'x'. A linear regression is fit using offline error to predict log-transformed online error for NNs that do not crash, and the corresponding Pearson's R-squared is shown in the top left-hand corner.

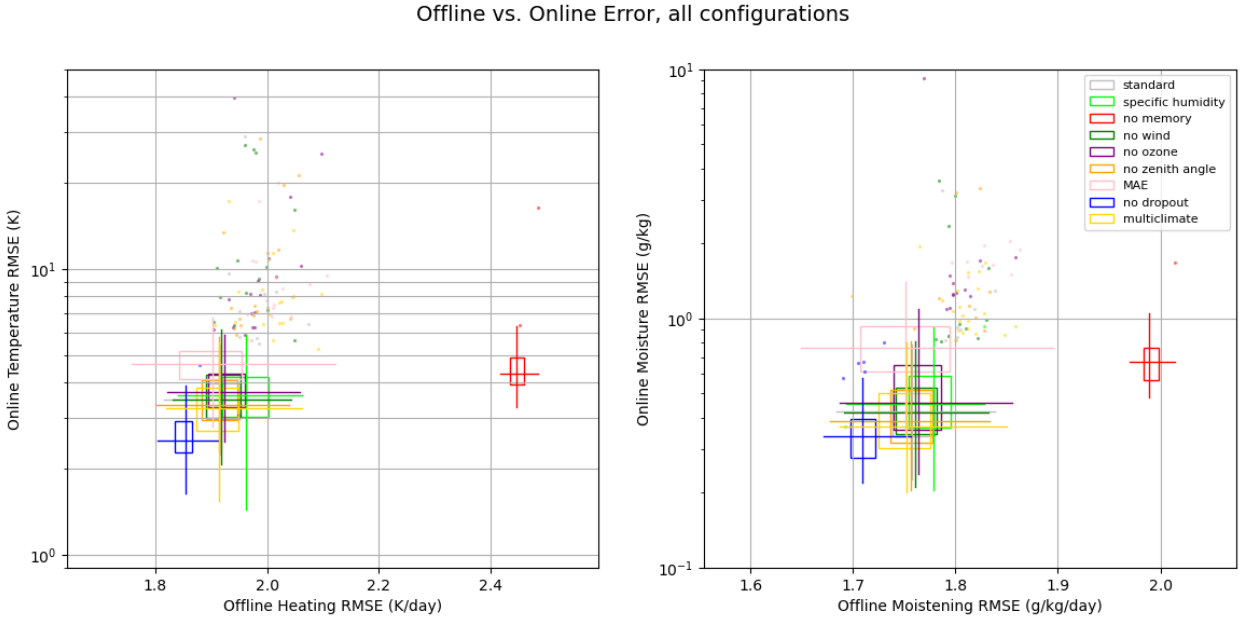

Fig. S25. This figure shows offline error vs. online error boxplots for all configurations. The Spearman correlation between offline heating error and online temperature error across all configurations is .549, and the corresponding spearman correlation for offline moistening error and online temperature is .531. These are both statistically significant.

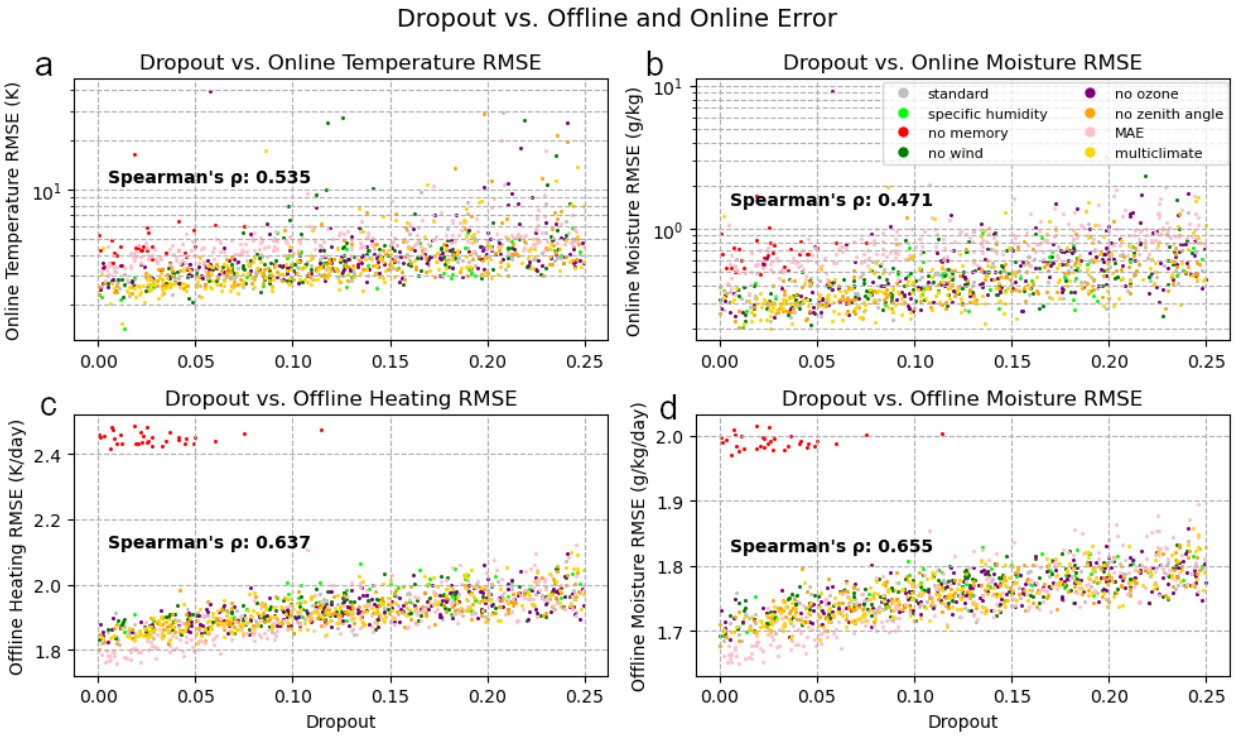

Fig. S26. This shows the effect of dropout on both offline heating (moistening) and online temperature (moisture) error after excluding the "no dropout" configuration and marginalizing out choice of configuration. Bolded values are statistically significant. Offline and online error values for hybrid runs that crashed are excluded.

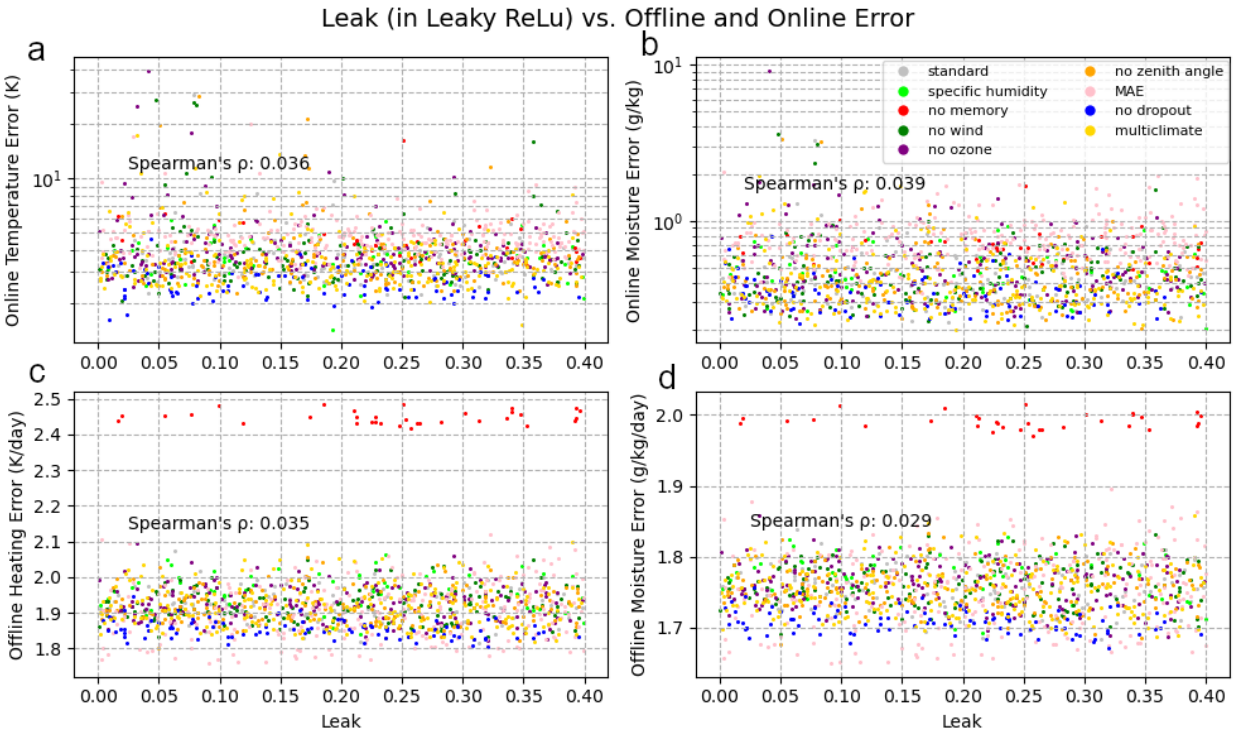

Fig. S27. This shows the effect of leak in Leaky ReLu on both offline heating (moistening) and online temperature (moisture) error after marginalizing out choice of configuration. Bolded values are statistically significant. Offline and online error values for hybrid runs that crashed are excluded.

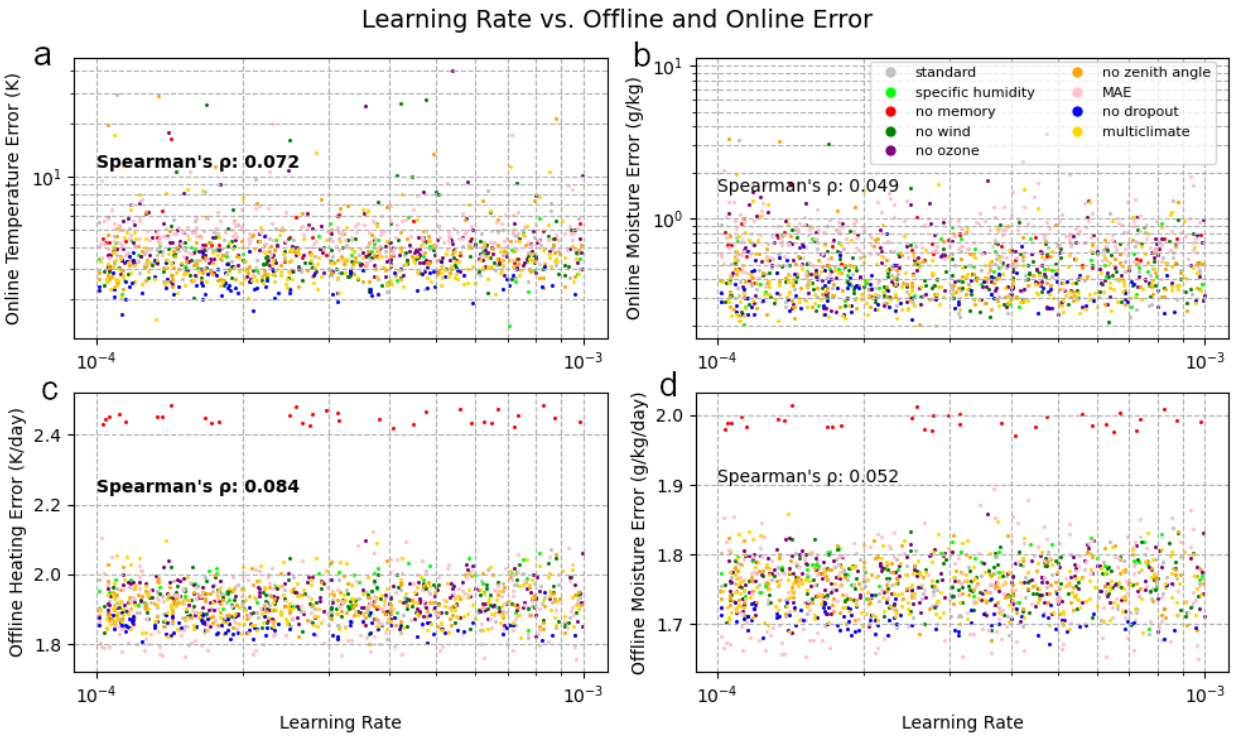

Fig. S28. This shows the effect of learning rate on both offline heating (moistening) and online temperature (moisture) error after marginalizing out choice of configuration. Bolded values are statistically significant. Offline and online error values for hybrid runs that crashed are excluded.

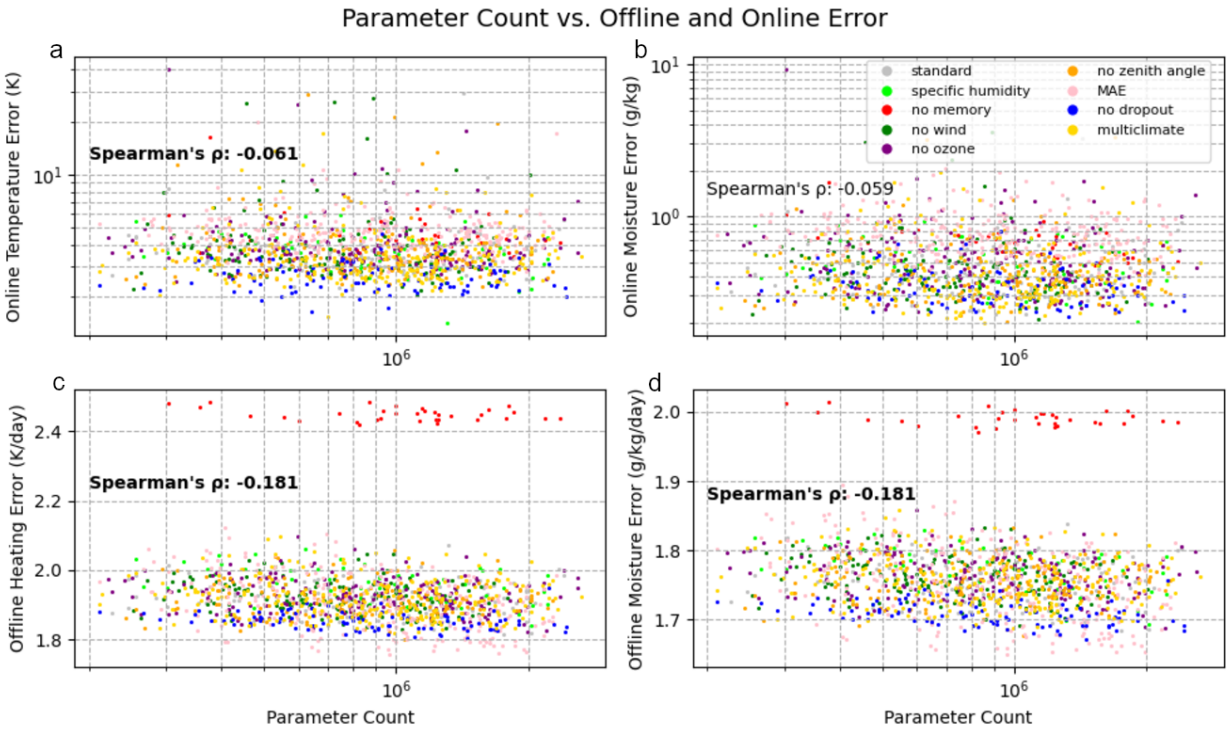

Fig. S29. This shows the effect of parameter count on both offline heating (moistening) and online temperature (moisture) error after marginalizing out choice of configuration. Bolded values are statistically significant. Offline and online error values for hybrid runs that crashed are excluded.

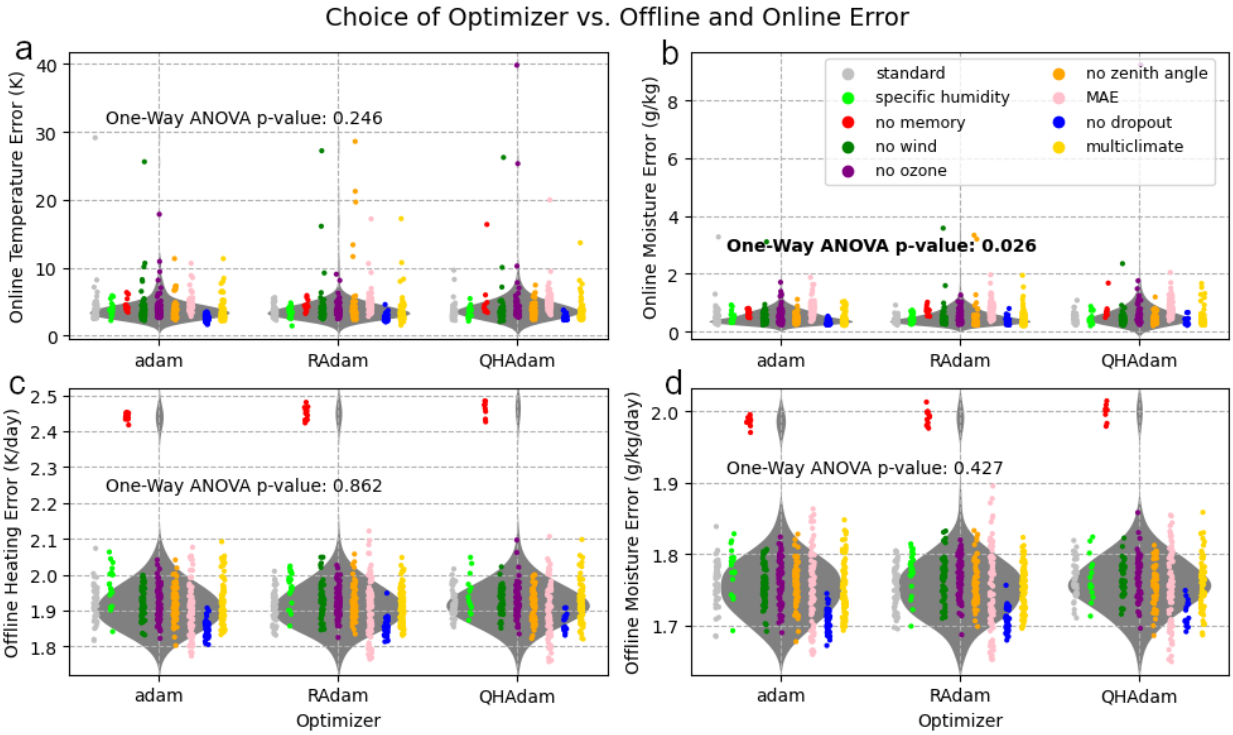

Fig. S30. This shows offline heating (moistening) and online temperature (moisture) error after conditioning on choice of optimizer and marginalizing out choice of configuration. We test for differences in offline and online error using one-way ANOVA tests. Although the one-way ANOVA test for online moisture error is significant, the subsequent Tukey's Honestly Significant Difference post hoc test did not uncover significant differences after correcting for the false discovery rate. Offline and online error values for hybrid runs that crashed are excluded.

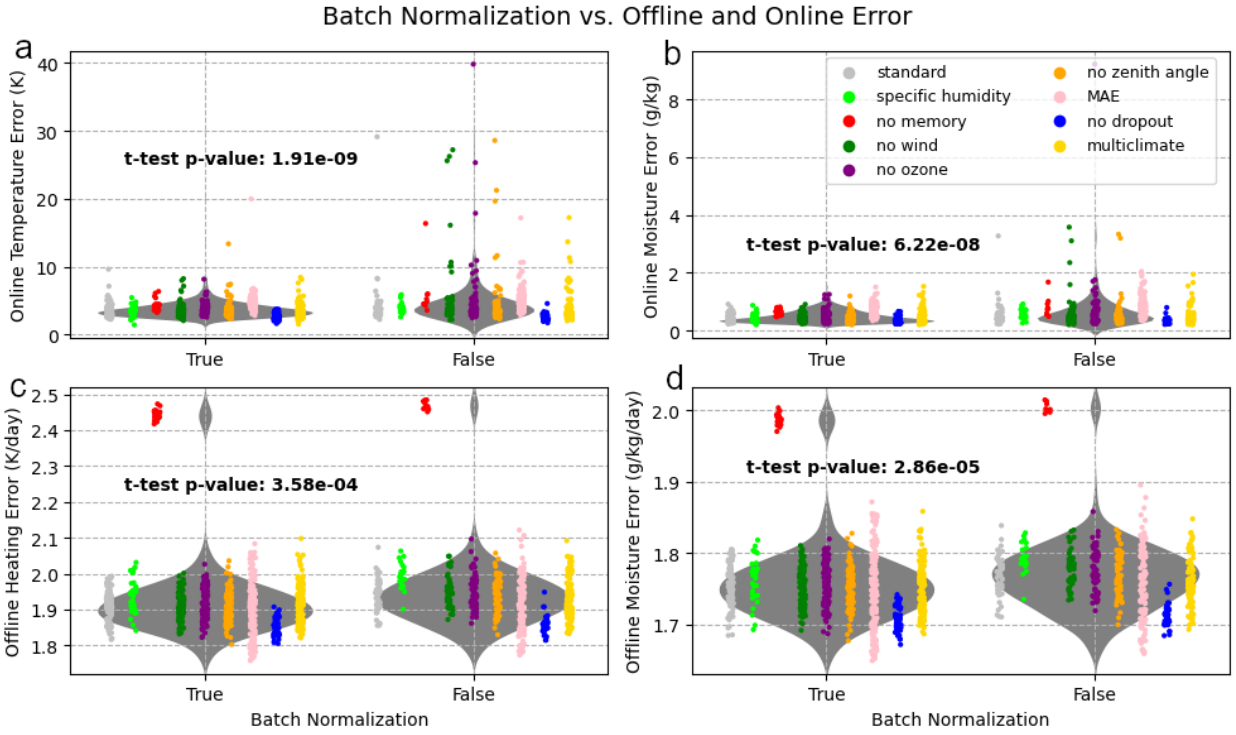

Fig. S31. This shows offline heating (moistening) and online temperature (moisture) error after conditioning on batch normalization and marginalizing out choice of configuration. We check for differences in mean offline and online error with and without batch normalization using two-tailed t-tests. Bolded values are statistically significant. Offline and online error values for hybrid runs that crashed are excluded.

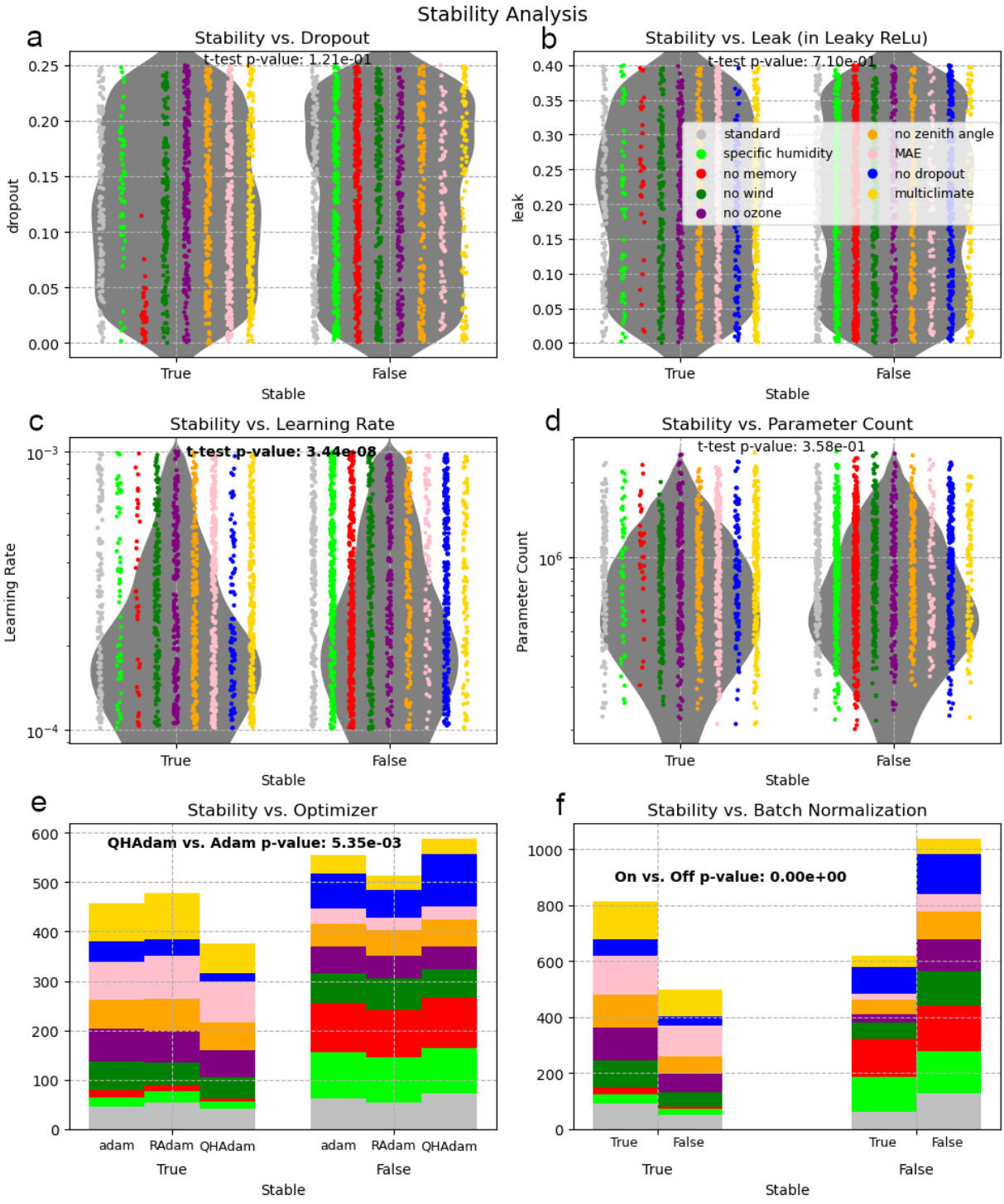

Fig. S32. This figure shows the effect of dropout, leak (in Leaky ReLu), learning rate, parameter count, and choice of optimizer on stability. We check for differences in mean dropout, leak, learning rate, and parameter count in stable and unstable NNs using two-tailed t-tests. For choice of optimizer and batch normalization, we make use of pairwise two-sample two-tailed proportion tests. RAdam is statistically equivalent to Adam, but QHAdam results in more unstable NNs than either. Bolded values are statistically significant.

Configuration	Offline Error* (T)	ρ^* (T)	Offline Error* (Q)	ρ^* (Q)
standard	2.04 K/day	**.698**	1.85 g/kg/day	**.734**
specific humidity	**+1.54 K/day**	**.331**	**+.393 g/kg/day**	**.304**
no memory	**+.872 K/day**	.250	**+.395 g/kg/day**	.0842
no wind	$+5.43 \times 10^{-5}$ K/day	**.624**	+.00445 g/kg/day	**.585**
no ozone	+.00259 K/day	**.698**	-.00534 g/kg/day	**.846**
no zenith angle	+.00561 K/day	**.659**	+.00267 g/kg/day	**.708**
MAE	**-0.0575 K/day**	**.688**	**-.0258 g/kg/day**	**.783**
no dropout	**-0.0259 K/day**	.186	**+.0777 g/kg/day**	.0752
multiclimate	**-0.0842 K/day**	**.860**	**-.0405 g/kg/day**	**.774**

Table S2: This table shows offline error for heating in K/day, Spearman correlation between offline heating and online temperature RMSE, offline error for moistening in g/kg/day, and Spearman correlation between offline moistening and online moisture RMSE using a multiclimate offline test set. As expected, the multiclimate configuration performs the best on this test set. The significantly worse performance for the specific humidity configuration here also confirms the "climate-invariant" benefits of a relative humidity transformation for moisture input first shown in Beucler et al. (2024).

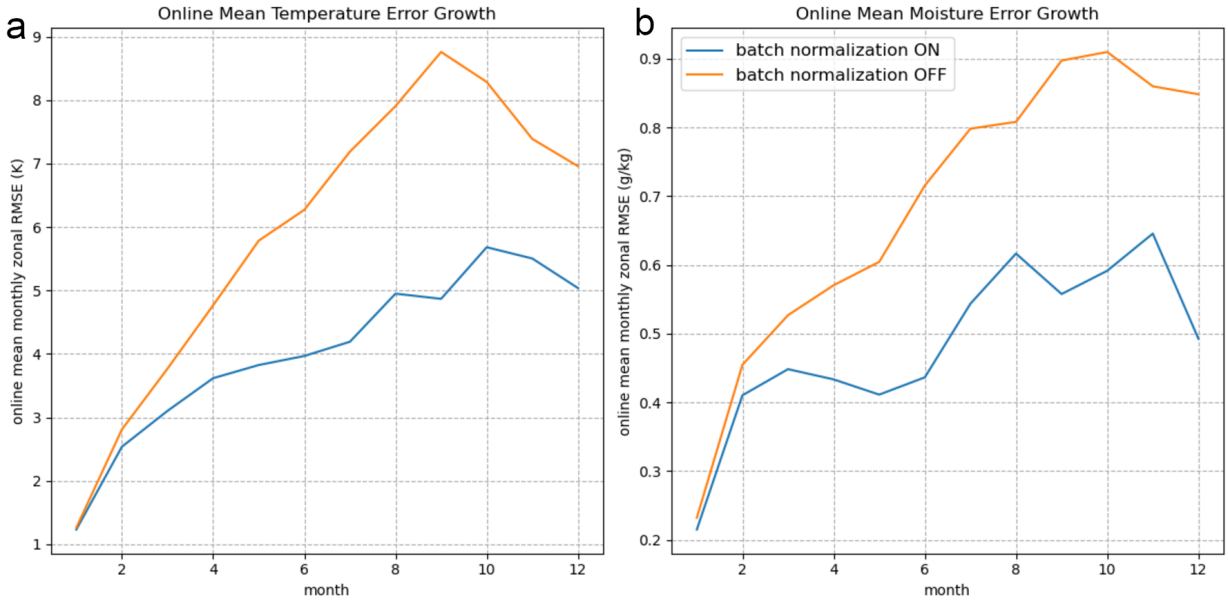

Fig. S33. This figure shows mean online error for both temperature and moisture over time for each hybrid simulations with and without batch normalization. Ensemble-mean online error is only recorded for simulations that do not crash, and they are calculated across the entire duration of the simulations.

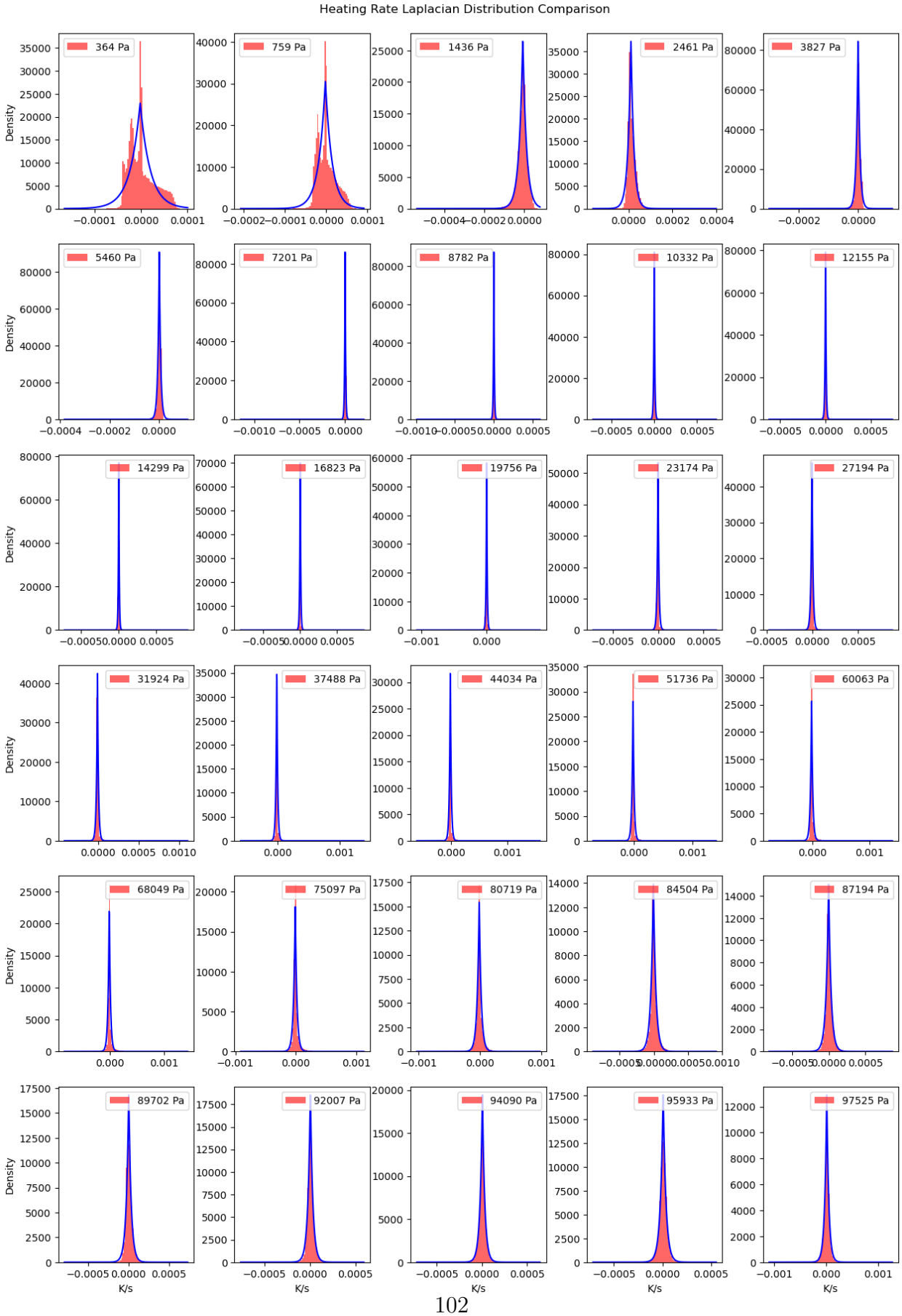

Fig. S34. This figure shows Laplacian distributions being fit to histograms of the heating tendencies at each vertical level.

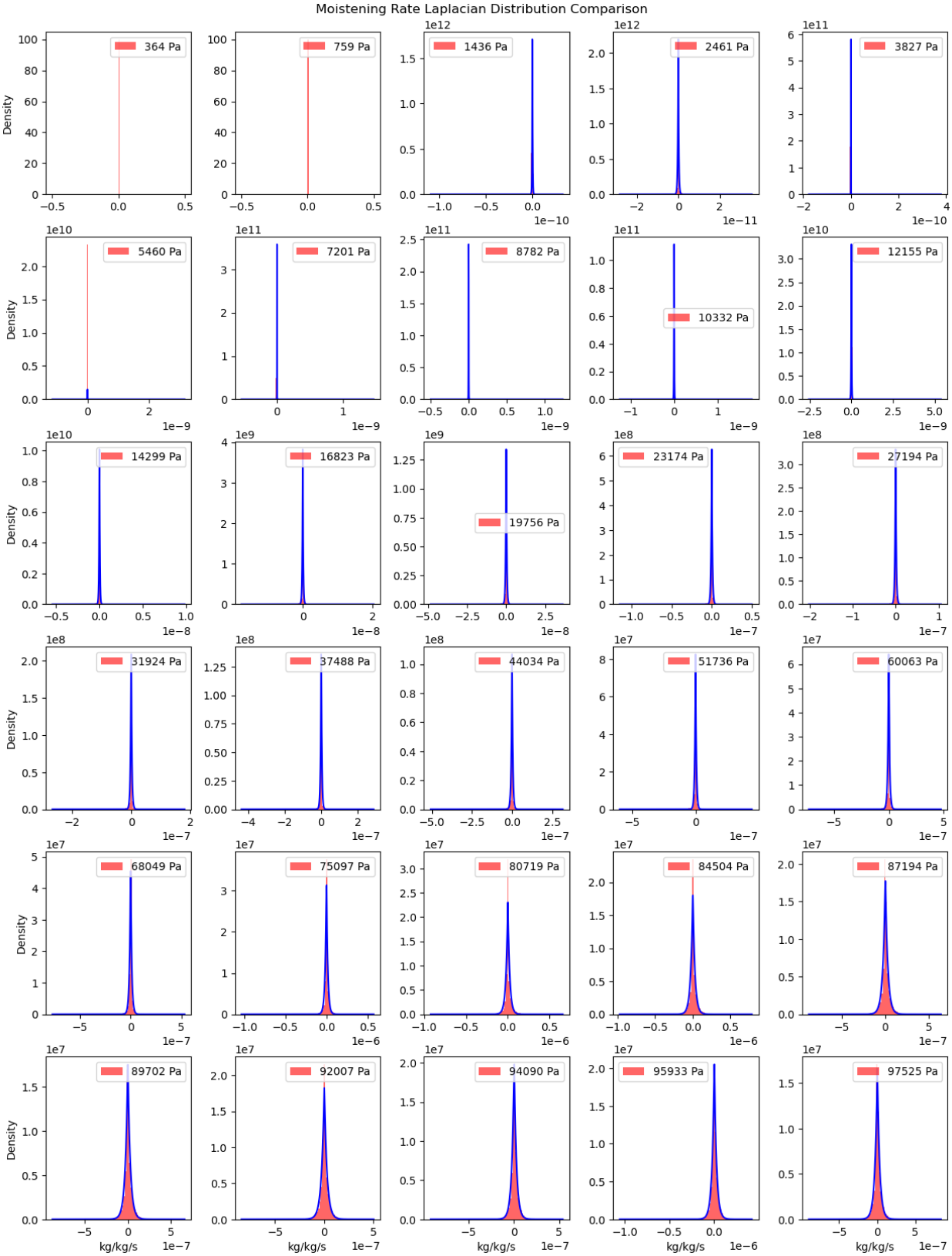

Fig. S35. This figure shows Laplacian distributions being fit to histograms of the moistening tendencies at each vertical level. The top five levels are included in this figure for completeness but are not used for the NN emulators.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500.

Balogh, B., Saint-Martin, D., and Ribes, A. (2021). A toy model to investigate stability of AI-based dynamical systems. *Geophys. Res. Lett.*, 48(8):e2020GL092133.

Baniodeh, M., Goel, K., Ettinger, S., Fuertes, C., Seff, A., Shen, T., Gulino, C., Yang, C., Jerfel, G., Choe, D., Wang, R., Kallem, V., Casas, S., Al-Rfou, R., Sapp, B., and Anguelov, D. (2025). Scaling laws of motion forecasting and planning – a technical report. *arXiv [cs.LG]*.

Behrens, G., Beucler, T., Gentine, P., Iglesias-Suarez, F., Pritchard, M., and Eyring, V. (2022). Non-Linear dimensionality reduction with a variational encoder decoder to understand convective processes in climate models. *J Adv Model Earth Syst*, 14(8):e2022MS003130.

Behrens, G., Beucler, T., Iglesias-Suarez, F., Yu, S., Gentine, P., Pritchard, M., Schwabe, M., and Eyring, V. (2024). Improving atmospheric processes in earth system models with deep learning ensembles and stochastic parameterizations. *arXiv [physics.ao-ph]*.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical

and powerful approach to multiple testing. *J. R. Stat. Soc. Series B Stat. Methodol.*, 57(1):289–300.

Beucler, T., Gentine, P., Yuval, J., Gupta, A., Peng, L., Lin, J., Yu, S., Rasp, S., Ahmed, F., O'Gorman, P. A., Neelin, J. D., Lutsko, N. J., and Pritchard, M. (2024). Climate-invariant machine learning. *Sci Adv*, 10(6):eadj7250.

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., and Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.*, 126(9):098302.

Bhouri, M. A., Peng, L., Pritchard, M. S., and Gentine, P. (2023). Multi-fidelity climate model parameterization for better generalization and extrapolation. *arXiv.org*.

Brenowitz, N. D., Beucler, T., Pritchard, M., and Bretherton, C. S. (2020). Interpreting and stabilizing Machine-Learning parametrizations of convection. *J. Atmos. Sci.*, 77(12):4357–4375.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cachay, S. R., Ramesh, V., Cole, J. N. S., Barker, H., and Rolnick, D. (2021). ClimART: A benchmark dataset for emulating atmospheric radiative transfer in weather and climate models.

Ceppi, P. and Hartmann, D. L. (2016). Clouds and the atmospheric circulation response to warming. *J. Clim.*, 29(2):783–799.

Chen, J., Zhang, M., Zhang, T., Lin, W., and Xue, W. (2025). Stable simulation of the community atmosphere model using machine-learning physical parameterization trained with experience replay. *J. Adv. Model. Earth Syst.*, 17(6):e2024MS004722.

Chevallier, F., Chéruy, F., Scott, N. A., and Chédin, A. (1998). A neural network approach for a fast and accurate computation of a longwave radiative budget. *J. Appl. Meteorol. Climatol.*, 37(11):1385–1397.

Chow, A., Cameron, G., Georg, M., Sherwood, M., Culliton, P., Sepah, S., Dane, S., and Starner, T. (2023). Google - american sign language fingerspelling recognition. https://kaggle.com/competitions/asl-fingerspelling. Kaggle.

Christensen, H. M., Kouhen, S., Miller, G., and Parthipan, R. (2024). Machine learning for stochastic parametrization. *Environ. Data Science*, 3(e38):e38.

Clark, S. K., Brenowitz, N. D., Henn, B., Kwa, A., McGibbon, J., Perkins, W. A., Watt-Meyer, O., Bretherton, C. S., and Harris, L. M. (2022). Correcting a 200 km resolution climate model in multiple climates by machine learning from 25 km resolution simulations. *J. Adv. Model. Earth Syst.*, 14(9).

Colin, M., Sherwood, S., Geoffroy, O., Bony, S., and Fuchs, D. (2019). Identifying the sources of convective memory in Cloud-Resolving simulations. *J. Atmos. Sci.*, 76(3):947–962.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):273–297.

Das, R., He, S., Huang, R., Townley, J., Kretsch, R., Karagianes, T., Nicol, J., Nye, G., Choe, C., Romano, J., Demkin, M., Reade, W., and players, E. (2023). Stanford ribonanza rna folding. https://kaggle.com/competitions/stanford-ribonanza-rna-folding. Kaggle.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Desai, M., Zhang, Y., Holbrook, R., O'Neil, K., and Demkin, M. (2024). Jane street real-time market data forecasting. https://kaggle.com/competitions/jane-street-real-time-market-data-forecasting. Kaggle.

Forbes, T., Macgillivray, J., Pietrobon, M., Dane, S., and Demkin, M. (2023). Optiver - trading at the close. https://kaggle.com/competitions/optiver-trading-at-the-close. Kaggle.

Gagne, II, D. J., Christensen, H. M., Subramanian, A. C., and Monahan, A. H. (2020). Machine learning for stochastic parameterization: Generative adversarial networks in the lorenz '96 model. *J. Adv. Model. Earth Syst.*, 12(3):e2019MS001896.

Gal, Y. and Ghahramani, Z. (2015). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.

Geleta, M., Montserrat, D. M., Giro-i Nieto, X., and Ioannidis, A. G. (2023). Deep variational autoencoders for population genetics.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.*, 45(11):5742–5751.

Guillaumin, A. P. and Zanna, L. (2021). Stochastic-deep learning parameterization of ocean momentum forcing. *J. Adv. Model. Earth Syst.*, 13(9).

Hamilton, M. A. and Collings, B. J. (1991). Determining the appropriate sample size for nonparametric tests for location shift. *Technometrics*.

Hamner, B., databits, fjanoos, JasonBeverage, tes1985, Valsalam, V., and Cukierski, W. (2016). Two sigma financial modeling challenge. https://kaggle.com/competitions/two-sigma-financial-modeling. Kaggle.

Han, Y., Zhang, G. J., Huang, X., and Wang, Y. (2020). A moist physics parameterization based on deep learning. *J. Adv. Model. Earth Syst.*, 12(9).

Han, Y., Zhang, G. J., and Wang, Y. (2023). An ensemble of neural networks for moist physics processes, its generalizability and stable integration. *J. Adv. Model. Earth Syst.*, 15(10).

He, S., organizers, C., experimentalists, C. R., consortium, R.-P., team, V., Kretsch, R., Hummer, A., Favor, A., Reade, W., Demkin, M., Das, R., et al. (2025). Stanford rna 3d folding. https://kaggle.com/competitions/stanford-rna-3d-folding. Kaggle.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. (2017). Deep learning scaling is predictable, empirically. *arXiv [cs.LG]*.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.

Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geosci. Model Dev.*, 15(14):5481–5487.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. (2022). Training compute-optimal large language models. *arXiv [cs.CL]*.

Hohenegger, C., Korn, P., Linardakis, L., Redler, R., Schnur, R., Adamidis, P., Bao, J., Bastin, S., Behravesh, M., Bergemann, M., Biercamp, J., Bockelmann, H., Brokopf, R., Brüggemann, N., Casaroli, L., Chegini, F., Datseris, G., Esch, M., George, G., Giorgetta, M., Gutjahr, O., Haak, H., Hanke, M., Ilyina, T., Jahns, T., Jungclaus, J., Kern, M., Klocke, D., Kluft, L., Kölling, T., Kornblueh, L., Kosukhin, S., Kroll, C., Lee, J., Mauritsen, T., Mehlmann, C., Mieslinger, T., Naumann, A. K., Paccini, L., Peinado, A., Praturi, D. S., Putrasahan, D., Rast, S., Riddick, T., Roeber, N., Schmidt, H., Schulzweida, U., Schütte, F., Segura, H., Shevchenko, R., Singh, V., Specht, M., Stephan, C. C., von Storch, J.-S., Vogel, R., Wengel, C., Winkler, M., Ziemen, F., Marotzke, J., and Stevens, B. (2023). ICON-sapphire: simulating the components of the earth system and their interactions at kilometer and subkilometer scales. *Geosci. Model Dev.*, 16(2):779–811.

Hu, Z., Subramaniam, A., Kuang, Z., Lin, J., Yu, S., Hannah, W. M., Brenowitz, N. D., Romero, J., and Pritchard, M. S. (2024). Stable machine-learning parameterization of subgrid processes with real geography and full-physics emulation. *arXiv [physics.ao-ph]*.

Iglesias-Suarez, F., Gentine, P., Solino-Fernandez, B., Beucler, T., Pritchard, M., Runge, J., and Eyring, V. (2024). Causally-informed deep learning to improve climate models and projections. *Journal of Geophysical Research: Atmospheres*, 129(4):e2023JD039202.

IPCC (2021). *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*.

Jing, J., Lin, Z., Yang, C., Chow, A., Dane, S., Sun, J., and Westover, M. B. (2024). Hms - harmful brain activity classification. https://kaggle.com/competitions/hms-harmful-brain-activity-classification. Kaggle.

Jones, T. R., Randall, D. A., and Branson, M. D. (2019). Multiple-instance superparam-

eterization: 1. concept, and predictability of precipitation. *J. Adv. Model. Earth Syst.*, 11(11):3497–3520.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv [cs.LG]*.

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

Kim, S., Gholami, A., Shaw, A., Lee, N., Mangalam, K., Malik, J., Mahoney, M. W., and Keutzer, K. (2022). Squeezeformer: An efficient transformer for automatic speech recognition. *arxiv:2206.00888*.

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., Klöwer, M., Lottes, J., Rasp, S., Düben, P., Hatfield, S., Battaglia, P., Sanchez-Gonzalez, A., Willson, M., Brenner, M. P., and Hoyer, S. (2024). Neural general circulation models for weather and climate. *Nature*, pages 1–7.

Kohl, G., Chen, L.-W., and Thuerey, N. (2023). Turbulent flow simulation using autoregressive conditional diffusion models. *arXiv [cs.LG]*.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

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

Li, F., Rosa, D., Collins, W. D., and Wehner, M. F. (2012). "super-parameterization": A better way to simulate regional extreme precipitation? *J. Adv. Model. Earth Syst.*, 4(2).

Lin, J., Bhouri, M. A., Beucler, T., Yu, S., and Pritchard, M. (2024a). Stress-testing the coupled behavior of hybrid physics-machine learning climate simulations on an unseen, warmer climate. *arXiv [physics.ao-ph]*.

Lin, J., Hu, Z., Yu, S., Pritchard, M., Gupta, R., Zheng, T., Hannah, W., Mansfield, L., Qu, Y., Geleta, M., Lopez, M., Rudolph, M., Chow, A., and Reade, W. (2024b). Leap - atmospheric physics using ai (climsim).

Lin, J. and Yu, S. (2024). Climscale.

Lin, J., Yu, S., Peng, L., Beucler, T., Wong-Toi, E., Hu, Z., Gentine, P., Geleta, M., and Pritchard, M. (2025). Navigating the noise: Bringing clarity to ML parameterization design with O ϱ(100) ensembles. *J. Adv. Model. Earth Syst.*, 17(4):e2024MS004551.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2019). On the variance of the adaptive learning rate and beyond.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lopez-Gomez, I., Christopoulos, C., Langeland Ervik, H. L., Dunbar, O. R. A., Cohen, Y., and Schneider, T. (2022). Training physics-based machine-learning parameterizations with gradient-free ensemble kalman methods. *J. Adv. Model. Earth Syst.*, 14(8):e2022MS003105.

Ma, J. and Yarats, D. (2018). Quasi-hyperbolic momentum and adam for deep learning.

Mansfield, L. A. and Sheshadri, A. (2024). Uncertainty quantification of a machine learning subgrid-scale parameterization for atmospheric gravity waves. *J. Adv. Model. Earth Syst.*, 16(7):e2024MS004292.

Meehl, G. A., Senior, C. A., Eyring, V., Flato, G., Lamarque, J.-F., Stouffer, R. J., Taylor, K. E., and Schlund, M. (2020). Context for interpreting equilibrium climate sensitivity and transient climate response from the CMIP6 earth system models. *Science Advances*, 6(26):eaba1981.

Molina, M. J., Gagne, D. J., and Prein, A. F. (2021). A benchmark to test generalization capabilities of deep learning methods to classify severe convective storms in a changing climate. *Earth Space Sci.*, 8(9).

Mooers, G., Pritchard, M., Beucler, T., Ott, J., Yacalis, G., Baldi, P., and Gentine, P. (2021). Assessing the potential of deep learning for emulating cloud superparameterization in climate models with real-geography boundary conditions. *J. Adv. Model. Earth Syst.*, 13(5).

Moult, J., Fidelis, K., Zemla, A., and Hubbard, T. (1995). A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Genetics*, 23(3):ii–iv.

Neural Information Processing Systems Conference (2021). Announcing the NeurIPS 2021 datasets and benchmarks track. https://neuripsconf.medium.com/announcing-the-neurips-2021-datasets-and-benchmarks-track-644e27c1e66c. Accessed: 2025-6-14.

Noether, G. E. (1987). Sample size determination for some common nonparametric tests. *J. Am. Stat. Assoc.*, 82(398):645–647.

O'Gorman, P. A. (2012). Sensitivity of tropical precipitation extremes to climate change. *Nat. Geosci.*, 5(10):697–700.

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). Kerastuner. https://github.com/keras-team/keras-tuner.

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., Baldi, P., and Acacio Sanchez, M. E. (2020). A Fortran-Keras deep learning bridge for scientific computing. *Sci. Program.*, 2020.

Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R. (2021). Nerfies: Deformable neural radiance fields. *ICCV*.

Pritchard, M., Yu, S., and Lin, J. (2024). spcam3.0-neural-net-spreadtesting.

Pritchard, M. S. and Bretherton, C. S. (2014). Causal evidence that rotational moisture advection is critical to the superparameterized Madden–Julian oscillation. *J. Atmos. Sci.*, 71(2):800–815.

Randall, D. (2013). Beyond deadlock. *Geophys. Res. Lett.*, 40(22):5970–5976.

Randall, D., Khairoutdinov, M., Arakawa, A., and Grabowski, W. (2003). Breaking the cloud parameterization deadlock. *Bull. Am. Meteorol. Soc.*, 84(11):1547–1564.

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.

Rasp, S., Pritchard, M., and Gentine, P. (2018). Deep learning to represent subgrid processes in climate models.

Rasp, S., Schulz, H., Reade, W., and Demkin, M. (2019). Understanding clouds from satellite images. https://kaggle.com/competitions/understanding_cloud_organization. Kaggle.

Reuel, A., Hardy, A. F., Smith, C., Lamparth, M., Hardy, M., and Kochenderfer, M. J. (2024). BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. *Neural Inf Process Syst*, abs/2411.12990:21763–21813.

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

Shamekh, S., Lamb, K. D., Huang, Y., and Gentine, P. (2023). Implicit learning of convective organization explains precipitation stochasticity. *Proceedings of the National Academy of Sciences*, 120(20):e2216158120.

Shepherd, T. G. (2014). Atmospheric circulation as a source of uncertainty in climate change projections. *Nat. Geosci.*, 7(10):703–708.

Sherwood, S. C., Bony, S., and Dufresne, J.-L. (2014). Spread in model climate sensitivity traced to atmospheric convective mixing. *Nature*, 505(7481):37–42.

Sherwood, S. C., Webb, M. J., Annan, J. D., Armour, K. C., Forster, P. M., Hargreaves, J. C., Hegerl, G., Klein, S. A., Marvel, K. D., Rohling, E. J., Watanabe, M., Andrews, T., Braconnot, P., Bretherton, C. S., Foster, G. L., Hausfather, Z., von der Heydt, A. S., Knutti, R., Mauritsen, T., Norris, J. R., Proistosescu, C., Rugenstein, M., Schmidt, G. A., Tokarska, K. B., and Zelinka, M. D. (2020). An assessment of earth's climate sensitivity using multiple lines of evidence. *Rev. Geophys.*, 58(4):e2019RG000678.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(56):1929–1958.

Stan, C., Khairoutdinov, M., DeMott, C. A., Krishnamurthy, V., Straus, D. M., Randall, D. A., Kinter, III, J. L., and Shukla, J. (2010). An ocean-atmosphere climate simulation with an embedded cloud resolving model: A GLOBAL CLIMATE SIMULATION. *Geophys. Res. Lett.*, 37(1).

Tange, O. (2018). *GNU Parallel 2018*. Ole Tange.

Taylor, M., Caldwell, P. M., Bertagna, L., Clevenger, C., Donahue, A., Foucar, J., Guba, O., Hillman, B., Keen, N., Krishna, J., Norman, M., Sreepathi, S., Terai, C., White, J. B., Salinger, A. G., McCoy, R. B., Leung, L.-Y. R., Bader, D. C., and Wu, D. (2023). The simple cloud-resolving E3SM atmosphere model running on the frontier exascale system. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, number Article 7 in SC '23, pages 1–11, New York, NY, USA. Association for Computing Machinery.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wang, X., Han, Y., Xue, W., Yang, G., and Zhang, G. J. (2022). Stable climate simulations using a realistic general circulation model with neural network parameterizations for atmospheric moist physics and radiation processes. *Geosci. Model Dev.*, 15(9):3923–3940.

Webb, M. J., Lambert, F. H., and Gregory, J. M. (2013). Origins of differences in climate sensitivity, forcing and feedback in climate models. *Clim. Dyn.*, 40(3-4):677–707.

Woelfle, M. D., Yu, S., Bretherton, C. S., and Pritchard, M. S. (2018). Sensitivity of coupled tropical pacific model biases to convective parameterization in CESM1. *Journal of Advances in Modeling Earth Systems*, 10(1):126–144.

Yu, S., Hannah, W., Peng, L., Lin, J., Bhouri, M. A., Gupta, R., Lütjens, B., Will, J. C., Behrens, G., Busecke, J., Loose, N., Stern, C., Beucler, T., Harrop, B., Hillman, B., Jenney, A., Ferretti, S. L., Liu, N., Anandkumar, A., Brenowitz, N., Eyring, V., Geneva, N., Gentine, P., Mandt, S., Pathak, J., Subramaniam, A., Vondrick, C., Yu, R., Zanna, L., Zheng, T., Abernathey, R., Ahmed, F., Bader, D., Baldi, P., Barnes, E., Bretherton, C., Caldwell, P., Chuang, W., Han, Y., HUANG, Y., Iglesias-Suarez, F., Jantre, S., Kashinath, K., Khairoutdinov, M., Kurth, T., Lutsko, N., Ma, P.-L., Mooers, G., Neelin, J. D., Randall, D., Shamekh, S., Taylor, M., Urban, N., Yuval, J., Zhang, G., and Pritchard, M. (2023a). Climsim: A large multi-scale dataset for hybrid physics-ml climate emulation. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Ad-*

vances in Neural Information Processing Systems, volume 36, pages 22070–22084. Curran Associates, Inc.

Yu, S., Hannah, W., Peng, L., Lin, J., Bhouri, M. A., Gupta, R., Lütjens, B., Will, J. C., Behrens, G., Busecke, J., Loose, N., Stern, C. I., Beucler, T., Harrop, B., Hillman, B. R., Jenney, A., Ferretti, S., Liu, N., Anandkumar, A., Brenowitz, N. D., Eyring, V., Geneva, N., Gentine, P., Mandt, S., Pathak, J., Subramaniam, A., Vondrick, C., Yu, R., Zanna, L., Zheng, T., Abernathey, R., Ahmed, F., Bader, D. C., Baldi, P., Barnes, E., Bretherton, C., Caldwell, P., Chuang, W., Han, Y., Huang, Y., Iglesias-Suarez, F., Jantre, S., Kashinath, K., Khairoutdinov, M., Kurth, T., Lutsko, N., Ma, P.-L., Mooers, G., Neelin, J. D., Randall, D., Shamekh, S., Taylor, M. A., Urban, N., Yuval, J., Zhang, G., and Pritchard, M. (2023b). ClimSim: A large multi-scale dataset for hybrid physics-ML climate emulation.

Yu, S., Hu, Z., Subramaniam, A., Hannah, W., Peng, L., Lin, J., Bhouri, M. A., Gupta, R., Lütjens, B., Will, J. C., Behrens, G., Busecke, J. J. M., Loose, N., Stern, C. I., Beucler, T., Harrop, B., Heuer, H., Hillman, B. R., Jenney, A., Liu, N., White, A., Zheng, T., Kuang, Z., Ahmed, F., Barnes, E., Brenowitz, N. D., Bretherton, C., Eyring, V., Ferretti, S., Lutsko, N., Gentine, P., Mandt, S., Neelin, J. D., Yu, R., Zanna, L., Urban, N., Yuval, J., Abernathey, R., Baldi, P., Chuang, W., Huang, Y., Iglesias-Suarez, F., Jantre, S., Ma, P.-L., Shamekh, S., Zhang, G., and Pritchard, M. (2023c). ClimSim-online: A large multi-scale dataset and framework for hybrid ML-physics climate emulation. *arXiv [cs.LG]*.

Yuval, J., O'Gorman, P. A., and Hill, C. N. (2021). Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophys. Res. Lett.*, 48(6).

Zender, C. S. (2008). Analysis of self-describing gridded geoscience data with netCDF operators (NCO). *Environmental Modelling & Software*, 23(10-11):1338–1342.