Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx)

Aaron S. Donahue[1], Elynn Wu[2], W. Andre Perkins[2], Peter M. Caldwell[1], Christopher S. Bretherton[2], Finn Rebassoo[1], and Jean-Christophe Golaz[1]

[1]Lawrence Livermore National Laboratory, Livermore, CA USA
[2]Allen Institute for Artificial Intelligence, Seattle, WA, USA

Correspondence: Aaron S. Donahue (donahue5@llnl.gov)

Abstract. The Simple Cloud-Resolving E3SM Atmosphere Model (SCREAM) is the newest addition to the family of earth system models capable of explicitly resolving convective systems. SCREAM is a kilometer–scale configuration of the advanced E3SM Atmosphere Model (EAMxx), designed for heterogeneous computing architectures. While the enhanced accuracy of kilometer–scale modeling offers significant benefits, it comes with a substantial computational cost, limiting feasible simulation durations to only a few years to a few decades, even on the fastest supercomputers. Machine learning presents an opportunity for scientists to achieve the high accuracy of storm-resolving models at a significantly reduced cost. Building on the previous success of applying corrective machine learning (ML) to the FV3GFS earth system model, this study explores the effects of implementing corrective–ML in EAMxx-SCREAM. We also address the computational challenges of integrating our implementation of corrective–ML, which is written in Python, with the C++/Kokkos EAMxx driver, as well as potential reasons why this approach has not proved as effective for EAMxx-SCREAM as for FV3GFS.

1 Introduction

Accurate future climate projections are crucial to a variety of sectors, including agricultural, energy, and public health. For instance, they can help predict shifts in growing seasons, optimize renewable energy deployment, and improve understanding of climate-sensitive diseases. Currently, physics-based earth system models, also referred to as general circulation models (GCMs), are responsible for generating these projections and require a massive amount of computing resources in order to produce one climate realization (Hohenegger et al., 2023). These models need to balance accuracy with feasibility, and most opt to use coarse spatial resolution, typically around 100 km, allowing them to produce ensembles of climate simulations for decades or centuries. However, the downside of using coarse–resolution GCMs is their inability to resolve storms, clouds, and complex topography (Satoh et al., 2019). The Simple Cloud-Resolving E3SM Atmosphere Model (SCREAM) addresses the resolution problem by using kilometer–scale resolution, but it is too computationally expensive to run more than a few years at a time (Caldwell et al., 2021). This project aims to apply a computationally efficient machine learning (ML) based emulator of high–resolution SCREAM nudging tendencies, thus replicating the high accuracy of the cloud–resolving GCM at the reduced computational cost of the coarse–resolution GCM.

ML has the potential of revolutionizing how weather forecasts and climate predictions are generated (Eyring et al., 2024). Within this active area of research, there are two primary approaches. The first approach relies solely on ML models trained on historical observations, reanalysis datasets, or outputs from existing GCMs (Keisler, 2022; Chen et al., 2023; Lam et al., 2023; Price et al., 2023). The second approach involves a hybrid framework that combines machine learning with traditional GCMs. In this method, machine learning methods are often used to replace a specific physical process of GCMs (Kochkov et al., 2023; Henn et al., 2024; Hu et al., 2025) or to apply column-wise correction to the coarse–grid GCM (Bretherton et al., 2022). This study builds on the success of a study using the second approach by the Allen Institute for Artificial Intelligence (Ai2) working with the Finite–Volume Cubed–Sphere Global Forecast System (FV3GFS) GCM (https://github.com/ai2cm/fv3gfs-fortran) (Zhou et al., 2019). Ai2 was able to improve the predictive accuracy of relatively coarse–resolution global simulations using a machine-learning based correction to the atmosphere state (Bretherton et al., 2022; Kwa et al., 2023; Sanford et al., 2023).

This study applies their corrective–ML approach to SCREAM. In addition, it developed software infrastructure that will be helpful for subsequent studies coupling ML to SCREAM. Section 2 provides the recipe for the development of the ML training data. Section 3 covers how the corrective–ML model was embedded into the SCREAM code base. Section 4 examines how well the ML–corrected coarse GCM performs with respect to the fine–resolution target, followed by discussion of the results and conclusions of the study in sections 5 and 6.

2 Methods

SCREAM is a configuration of the atmosphere component of the Energy Exascale Earth System Model (E3SM) targeting kilometer–scale global resolutions. In order to accomplish performant simulations at this scale, the E3SM Atmosphere Model (EAMxx) was rewritten from the original Fortran code to C++/Kokkos (Donahue et al., 2024). The adoption of Kokkos (Carter-Edwards et al., 2014) in EAMxx unlocks the computational power of mixed CPU/GPU machines and ensures that EAMxx maintains good performance across a number of high performance computing systems (Donahue et al., 2024). EAMxx employs a spectral-element, nonhydrostatic dynamical core for resolved-scale processes (Taylor et al., 2020). Major subgrid physics parameterizations include P3 microphysics (Morrison and Milbrandt, 2015), SHOC turbulence and boundary layer physics (Bogenschutz and Krueger, 2013), and RRTMG++ radiation (Pincus et al., 2019). Notably, the SCREAM configuration omits a deep convection parameterization, regardless of grid resolution (Caldwell et al., 2021). For all simulations in this work, the model configuration and initial start date are identical. Initial conditions for all simulations are generated from ERA5 (Hersbach et al., 2020).

While EAMxx achieves computational performance of more than one simulated year per wallclock day for a km-scale GCM (Taylor et al., 2023), its computational cost is still prohibitive for global km-scale simulations of a decade or more in length. This limitation makes it attractive to apply corrective–ML to EAMxx in order to represent effects of fine scale features at coarse resolution. We follow Ai2's approach with FV3GFS (Bretherton et al., 2022), using a km-scale EAMxx simulation to produce fine–scale training data for a corrective–ML model embedded in coarse–resolution EAMxx simulations. The approach

is modified to handle technical and architectural differences between FV3GFS and EAMxx, see sections 2.1 and 2.2 for more details.

The corrective–ML approach described in this study can be represented by four steps (Figure 1). The first step involves the generation of a fine–grid reference simulation. This simulation serves two purposes: to provide the nudging targets for the training data set and to provide a validation data set for the ML-corrected solution. The second step involves nudging an EAMxx simulation at coarse resolution to the reference state. 'Nudging tendencies' derived from this step are used for the third step, training of the corrective–ML models. Finally, a coarse–grid EAMxx simulation is run with inline corrections from these ML models; ideally the resulting ML-corrected coarse GCM will more closely approximate the time evolution and solution of the target fine–grid GCM. Each of these steps is described in more detail in the following subsections.

All tunable model parameters are held fixed between the coarse– and fine–resolution simulations, so in every case the parameter space is consistent with that of the target fine–resolution simulation. As a result, the climate states of the untuned coarse simulations would likely diverge from the fine resolution over long timescales. However, this study focuses on the shorter timescale of one year. We therefore expect the coarse simulations, when nudged toward the fine–resolution state, to agree more closely with the fine–resolution simulation.

2.1 A km-scale reference solution

For the training and validation discussed in this paper, we used one year of customized outputs from a reference fine–grid simulation using the default SCREAM configuration of EAMxx (Donahue et al., 2024). This used a cubed–sphere spectral–element grid for the dynamics calculation which had a **n**umber of **e**lements per cube face of 1024×1024. Each element contains a **p**hysics **g**rid with a 2×2 arrangement of columns. Referred to in shorthand as the 'ne1024pg2' horizontal grid, this grid has approximately 3.25 km horizontal resolution, 128 vertical levels, and a physics timestep of 100 seconds.

The one–year simulations in this study follow the 'standard climate' configuration of a Cess-style SCREAM simulation pair further discussed by Terai et al. (2025). It started on August 1, 2019 and used a climatological seasonal cycle of sea-surface temperature (SST) and sea ice. The Terai et al. (2025) study provided the fine–resolution data for training and validation.

Temporally averaged SCREAM output was produced every 3 hours. Three-dimensional fields required for our ML study were vertically interpolated natively in SCREAM onto a set of fixed pressure levels (specified below), then horizontally averaged (area-weighted) along pressure surfaces to a target 'ne30pg2' (≈ 100 km) coarse grid, masking fine–grid points below the surface. Two dimensional ML-relevant fields such as surface precipitation were also horizontally averaged inline to the 'ne30pg2' grid. Only these horizontally coarsened 3D and 2D outputs were stored. Training data was produced from a 100km simulation that was nudged to the 3km state, see section 2.2 for details. Table 1 lists the variables used for nudging. Table 2 list all variables output for validation of the ML model.

The vertical grid used for pressure interpolation has 221 vertical levels. Starting with a lowest level of 1075 hPa, each subsequent level decreases by 5 hPa until 540 hPa. From 540 hPa to 270 hPa the spacing is 10 hPa, and from 270-hPa to 145 hPa the spacing is again 5 hPa. For the remainder of the upper atmosphere, the vertical grid uses the upper 60 typical isobaric vertical layers in an EAMxx grid, for a total of 221 layers. The interpolated data is masked for below-surface pressure

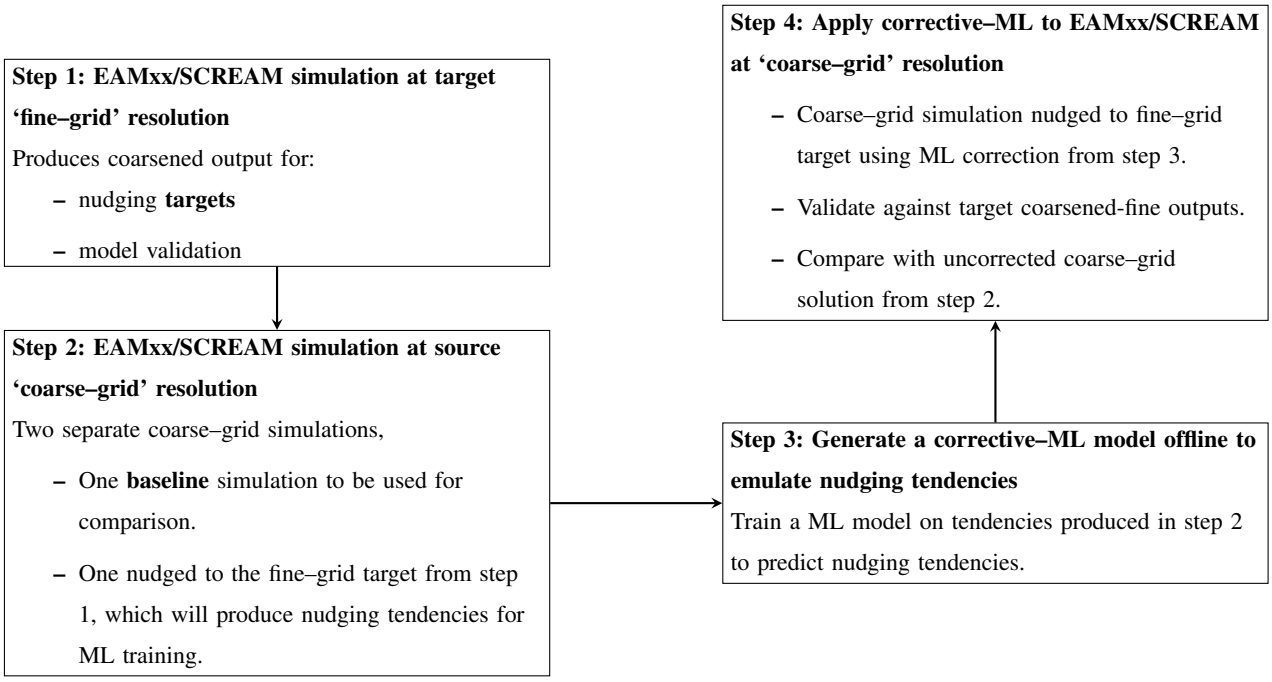

Figure 1. Flow diagram for training and applying the corrective machine learning (ML) model, which uses ML predicted nudging tendencies (step 3) to correct coarse–scale E3SM simulations (step 4). For this study 'coarse–grid' resolution refers to 100km global resolution and 'fine–grid' resolution refers to 3km global resolution. All simulations are conducted using the Simple Cloud–Resolving E3SM Atmosphere Model (SCREAM) configuration of the E3SM Atmosphere Model in C++/Kokkos (EAMxx).

levels, which is particularly important in mountainous terrain. The horizontal area-weighted averaging from the fine to the coarse grid is performed using the Tempest remap algorithm (Ullrich and Taylor, 2015; Ullrich et al., 2016), which efficiently redistributes data across different grid resolutions.

The approach to vertical remapping in this study differs from that used in the FV3GFS study. There, fine–grid data were vertically interpolated to the local pressure levels of the coarse–resolution data, masking out fine–grid data at any resulting grid points below the surface (since the FV3GFS model levels are terrain-following, these pressure levels differ between coarse grid cells). They then used the same method for horizontal coarsening as we do (Bretherton et al., 2022). Our coarse–grid ne30pg2 EAMxx configuration has only 72 terrain-following vertical levels, rather than the 128 levels of the fine–grid configuration. In the FV3GFS application, the coarse– and fine–grid model versions had the same set of vertical levels.

To maintain hydrostatic balance and conserve atmospheric mass during the vertical interpolation and horizontal coarsening process, the FV3GFS study also applied a small correction to the area-weighted surface elevation, which we neglect. See section 2.4 in Bretherton et al. (2022) for more details.

For the FV3GFS study, instantaneous 3-hourly snapshots of the prognostic fields were stored, rather than the 3-hourly average fields used here. Our approach produces smoother nudging tendencies, whereas the FV3GFS approach has the interpretational advantage that 3-hourly average tendencies are simply differences between successive 3-hourly snapshots.

Table 1. List of variables used for the 100km to 3km nudged simulations. Column 1 is the typical variable description, column 2 is the variable name as provided in EAMxx output files, column 3 is the units of the variable, column 4 describes the dimensionality of the variable. 2D has no vertical extent, 3D has vertical layers. Column 5 states if the variable was nudged or prescribed in the nudged simulation. Surface variables were prescribed.

Variable Common Name	**EAMxx Variable Name**	**Units**	**Dimension**	
Temperature	T_mid	K	3D	Nudged
Specific Humidity	qv	kg/kg	3D	Nudged
Eastward Wind	U	m/s	3D	Nudged
Northward Wind	V	m/s	3D	Nudged
Surface near-IR direct flux	sfc_flux_dir_nir	W/m2	2D	Prescribed
Surface UV/visible direct flux	sfc_flux_dir_vis	W/m2	2D	Prescribed
Surface UV/visible diffuse flux	sfc_flux_dif_vis	W/m2	2D	Prescribed
Net (down - up) SW flux at surface	sfc_flux_sw_net	W/m2	2D	Prescribed
Downwelling LW flux at surface	sfc_flux_lw_dn	W/m2	2D	Prescribed
Liquid precipitation flux	precip_liq_surf_mass_flux	kg/m2	2D	Prescribed

2.2 Machine Learning Training Data - Nudged Simulation

Similar to the work with FV3GFS (Bretherton et al., 2022), our training dataset for corrective–ML was constructed from a "low-resolution" 100km global simulation that was nudged to track the vertically interpolated and horizontally coarsened reference solution evolution described in section 2.1. At the end of each atmospheric timestep, the atmosphere state was adjusted to match the fine–resolution solution state using data from the reference simulations, with a nudging timescale of 3 hours. At each timestep, the reference data was interpolated to the current time before nudging.

EAMxx employs a hybrid pressure coordinate system vertically (Dennis et al., 2005), thus the reference data also required vertical interpolation from its fixed vertical pressure grid onto the EAMxx pressure coordinates at each timestep. For the training dataset simulation, only the three-dimensional variables—temperature, specific humidity, and horizontal velocities—were nudged. The other surface variables, as detailed in Table 1, were directly prescribed at the end of the atmospheric timestep. The nudging tendencies from this simulation were averaged every 3 hours and written to output. In addition, the training data requires a record of the atmosphere state that led to each nudging tendency, recorded at the same 3-hourly times as the tendencies.

Table 2. List of variables from the nudged simulation used for validation of the corrective–ML model. Column 1 is the typical variable description, column 2 is the variable name as provided in EAMxx output files, column 3 is the units of the variable and column 4 describes the dimensionality of the variable. 2D has no vertical extent, 3D has vertical layers.

Variable Common Name	**EAMxx Variable Name**	**Units**	**Dimension**
Water vapor path	VapWaterPath	mm	2D
Total precipitation to the surface	precip_liq_surf_mass_flux + precip_ice_surf_mass_flux	mm/day	2D
Surface temperature	surf_radiative_T	K	2D
Top of atmosphere upwelling LW	LW_flux_up_at_model_top	W/m2	2D
Top of atmosphere upwelling SW	SW_flux_up_at_model_top	W/m2	2D
Sensible heat flux	surf_sens_flux	W/m2	2D
Latent heat flux	surf_evap	W/m2	2D
Surface SW net flux	sfc_flux_sw_net	W/m2	2D
Surface downwelling LW flux	sfc_flux_lw_dn	W/m2	2D
Eastward wind	U	m/s	3D
Northward wind	V	m/s	3D

2.3 Training the Model

Following Ai2's prior work with FV3GFS, the nudging tendencies recorded during the nudged simulation discussed in section 2.2 are used to train three separate corrective–ML models for different state variables in EAMxx. All models utilize the Ai2 Climate Modeling Group's open-source `fv3net` GitHub repository (Donahue and Wu, 2025a). The training process is column–independent, meaning that each training data point does not consider neighboring spatial columns in the grid. However, the training dataset does include some grid location information through the latitude and cosine zenith angle associated with each point. Below, we outline the parameters for each model.

1. **Thermodynamic Model:** This model focuses on the thermodynamic state of the atmosphere. The input variables for training include temperature, specific humidity, latitude, surface geopotential, and cosine zenith angle. The training

outputs are the prescribed nudging tendencies for temperature and specific humidity. We follow Kwa et al. (2023) where a fully-connected dense network with 3 hidden layers with 419 neurons per layer is used.

2. **Momentum Model:** This model targets the atmospheric momentum state. The outputs are the eastward and northward velocity nudging tendencies. The input variables are the same as those used in the thermodynamic model, with the addition of eastward and northward velocity. This model uses the same dense network as the thermodynamic model.

3. **Surface Forcing Model:** This model addresses the surface forcing passed to other components of the earth system. Unlike the other models, it does not use nudging tendencies as input. Instead, it relies on the prescribed surface fluxes calculated in the reference solution. The inputs for this model match those of the thermodynamic model, while the outputs are the surface downwelling longwave flux and surface net shortwave flux broken down into near-infrared and visible direct and diffuse flux fraction. Like the thermodynamic model, this model also uses the same dense network. We apply output limiters to ensure physical consistency (e.g., downward near-infrared diffuse fraction is between 0-1).

All models are trained for 300 epochs, with an early stopping condition if convergence is detected. Random subsets of dates from the EAMxx dataset are selected for training and validation.

2.4 Coarse–grid GCM

The three machine learning models developed in section 2.3 are integrated into EAMxx to create a coarse–grid GCM that aspirationally produces solutions closer to fine–grid accuracy. For more details on how the ML models were embedded in the EAMxx code base please refer to the section on Python/C++ coupling, section 3. The corrective–ML is applied at the end of each atmospheric timestep, with the thermodynamic and momentum models operating independently but using the same model input state. The surface forcing model updates the surface fluxes based on the ML-corrected state. The set of variables saved for validation of the ML-corrected solution is given in Table 2.

Special care is needed when applying corrections from the thermodynamic model. Nudging specific humidity effectively adds or removes water vapor mass from the system, which typically reflects precipitation differences in the fine–resolution solution compared with the coarse–resolution solution. To account for this, the total change to the column water mass is calculated after adjusting specific humidity, and this result is then added to or subtracted from the total liquid precipitation. A mass clipper, in which negative values are set to zero, is employed to ensure that precipitation values remain non-negative. This correction method does not account for ice precipitation. Future studies could explore separate ice and liquid precipitation adjustments, depending e.g. on near-surface air temperature. For the short simulations considered in this study, we do not expect the change in mass due to clipping to have a significant impact, but for simulations that would be expected to run for hundreds of years a detailed analysis of the mass clipping frequency would be beneficial.

Likewise, the adjustment to specific humidity does not correct the cloud state toward the reference solution. Coarse GCM cloud biases affect the accuracy of the radiative forcing to the surface components in the GCM, which could feed back on the atmospheric evolution. Since our corrective–ML model directly learns the surface fluxes from the fine–grid reference data

rather than predicting corrective fluxes, we expect it to have surface fluxes consistent with the fine–grid reference simulations given the input profiles.

3 Implementation of ML workflow in EAMxx (Python/C++ coupling)

We opted to utilize an existing framework, `fv3net` (Donahue and Wu, 2025a), built by Ai2 and used in Bretherton et al. (2022); Clark et al. (2022); Kwa et al. (2023). `fv3net` includes all aspects of the ML workflow, including pre-processing native GCM output, ML training, offline and online testing, and reporting. `fv3net` was originally developed for Geophysical Fluid Dynamics Laboratory's (GFDL) FV3GFS. We made necessary modifications to work with EAMxx's data, and all updates are available in the open-source `fv3net` repository.

One of the main challenges is that `fv3net` is written in Python whereas EAMxx is in C++. We use `pybind11` to bridge between C++ and Python (W. Jakob, J. Rhinelander, D. Moldovan and others, 2017). The strategies for running on CPU and GPU are slightly different. On CPU, we utilize `pybind11`'s numpy bindings to transfer EAMxx's Kokkos view data to Python and declare them as numpy arrays. We do this as a non-copy operation, making the operations relatively cheap. On GPU, the data passing is different since numpy does not support GPU. In this case we pass EAMxx's Kokkos view data as pointers to Python and reconstruct numpy-like arrays from the Cupy package through unmanaged memory access (Okuta et al., 2017). As in the CPU case, we do not do any memory copies. Once Cupy arrays are constructed on the Python side, we are able to reuse the same workflow as in the CPU case. For both the CPU and GPU implementation the state variables are directly overwritten during the Python calls. The ML workflow adds 10% overhead for the CPU runs and 6% for GPU. There is also a one-time initialization cost for loading the necessary libraries in Python, which becomes negligible once the simulation is sufficiently long (e.g., 1 simulation month for the 100km resolution case).

While it is possible to avoid Python and instead directly interface with a trained ML model in C++ (e.g. via Tensorflow's C++ interface), our approach allowed us to take advantage of the existing `fv3net` package's robustness and stability. For example, we are able to take advantage of safeguards built into `fv3net` to ensure ML has physically-consistent output. Since `fv3net` is an end-to-end ML workflow, we also utilize its reporting for data analysis and visualization. Overall, using `fv3net` drastically reduced development time in every step of the ML workflow.

4 Results from km-scale experiment

We present results from a one-year free-running coarse–grid simulation that includes 'online' ML correction toward the fine–grid reference (Donahue and Wu, 2025b). We use an uncorrected 100km run as a baseline comparison. While the ML model captures the overall vertical structure of the target nudging tendencies for both temperature and specific humidity, it consistently underestimates the strength. This is illustrated in Figure 2 for the temperature and specific humidity fields. Figures 3 and 4 show the annual-mean bias and root mean square error (RMSE) of baseline and ML-corrected runs from the reference data (3.25 km coarsened to 100 km, also used as ML training data). For this experiment, we found the largest improvement in surface

downwelling longwave flux, with 16% improvement in global RMSE when compared to the baseline. Additionally, we found marginal improvement (5-10%) for net shortwave flux at the surface, precipitation, and surface pressure. However, RMSE was worse for surface temperature (24%) and total water vapor path (43%) in the ML-corrected run. Not shown are the eastward and northward wind comparisons, which showed little to no change between the nudged and ML–corrected solutions.

As shown in Figure 2, the ML model exhibits its largest bias in the tropics. This underestimation of the target nudging tendencies likely contributes to the observed weakening of the ITCZ, because the model provides insufficient local heating and moistening to sustain the expected convergence and convective intensity. Recall that EAMxx currently does not include a deep convection scheme. This omission may help explain the greater weakening observed in the tropics, where stronger ML corrections would be required to reduce the bias.

The pattern of variables that improve or degrade is broadly consistent with the underestimation of nudging tendencies shown in Figure 3. Surface radiative fluxes show the greatest improvement, which is plausible because the surface forcing model learns these fluxes directly from the fine–grid reference rather than inferring them from predicted tendencies. Precipitation and surface pressure improve more modestly. By contrast, water vapor path and surface temperature are closely linked to the temperature and specific humidity profiles that the model is designed to correct. When these corrections are systematically too weak, particularly in the tropics, as evident in Figure 3, the ML-corrected simulation may become less self-consistent than the uncorrected baseline, leading to larger annual-mean errors in these fields.

To evaluate the stability and robustness of the corrective–ML approach, we conducted a series of training experiments using three different random seeds in the neural networks. During the online prognostic runs, models trained with seeds 1 and 3 crashed due to the inputs being out-of-sample. To address this issue, we adopted the method outlined in Sanford et al. (2023), training a separate out-of-sample novelty detector to complement the existing neural networks for temperature and specific humidity modeling. This approach proved reliable, allowing all random seeds to successfully complete a full-year simulation. We found the ML–predicted tendencies to be similar across random seeds, although these results are not shown.

5 Discussion

This study investigated the transferability of an ML parameterization from one GCM to another, a subject of growing importance as modeling groups plan to integrate ML machinery from other groups. Although we adopted the same strategy as Ai2, it quickly became clear that applying the workflow to a different atmospheric model would be challenging because of differences in GCM design and implementation. Each step in the corrective–ML workflow— nudging from coarse to fine–resolution simulation, prescribing surface fluxes for the coarse simulation, integrating ML, and diagnostics— is tightly coupled to the specifics of the underlying GCM, making generalization difficult. Even though this project was a collaboration between the team that developed EAMxx and the team that successfully applied this corrective–ML approach to FV3GFS, the end result fell short of the desired improvement from corrective–ML.

There are several possibilities for why corrective–ML was less successful for EAMxx than it was for FV3GFS. The most obvious difference between the two modeling frameworks is that the ML-corrected version of EAMxx does not include a

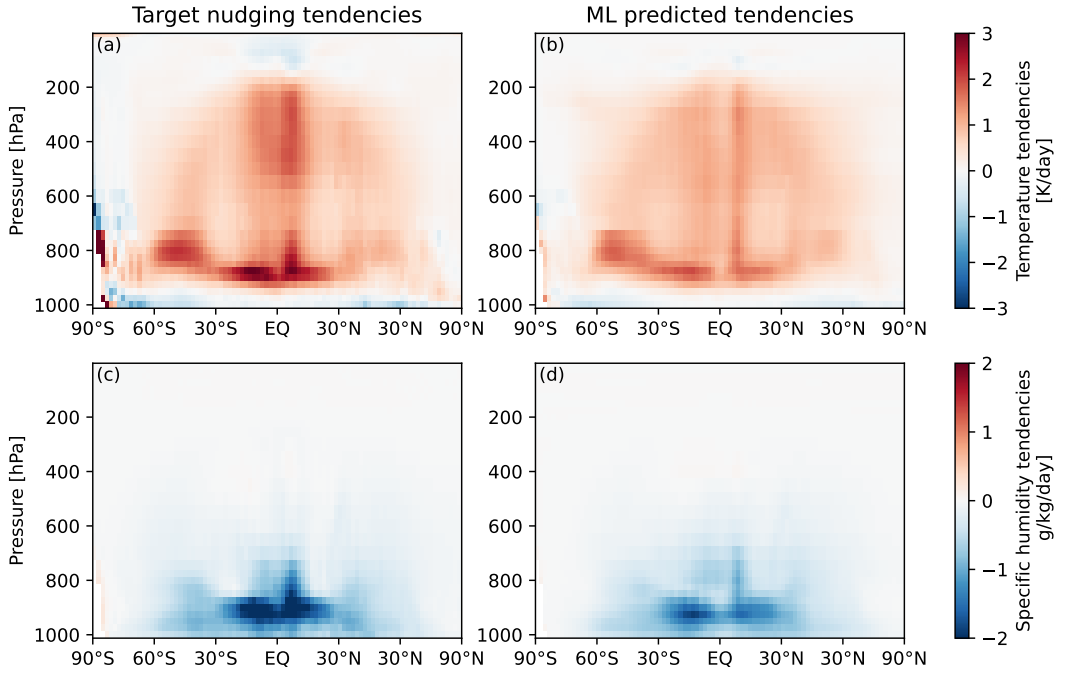

Figure 2. Zonal mean of target nudging tendencies (a and c) and ML–predicted tendencies (b and d) of air temperature and specific humidity over the one-year free–running simulation.

deep convective parameterization and therefore produces a less realistic large–scale state. As a result, more is being asked of corrective–ML in EAMxx than in FV3GFS. Our initial expectation was that larger differences between the nudged and free-running low-resolution simulations would provide a stronger signal for training the ML. But in such an environment, imperfections in the corrective–ML will also stand out more. Potential improvements in the ML model from the stronger signal may be too small to offset the larger biases in EAMxx associated with the absence of a subgrid convection scheme. Additionally, this study was limited to 3-hourly averaged fields, rather than the instantaneous fields used in the FV3GFS study. Training on these smoother averaged fields may have reduced the ML model's skill in predicting the appropriate nudging tendencies for instantaneous states. Future work will address these hypotheses when EAMxx is enhanced to support a low-resolution configuration with parameterized deep convection.

As discussed in the Methods section, there are three independent ML models that make up the full corrective–ML implementation, each with its own set of parameters to train on. In this study, we focused on training the models following the same approach as with the FV3GFS test case (Bretherton et al., 2022). Given the differences between EAMxx and FV3GFS, it is possible that a successfully trained model may require an adjustment to the training hyperparameters. An example is the use of latitude as an input into the ML training data set. We found that the inclusion of latitude in the training input improved model accuracy, presumably as a proxy for physical variables that correlate strongly with latitude but are not well represented in the

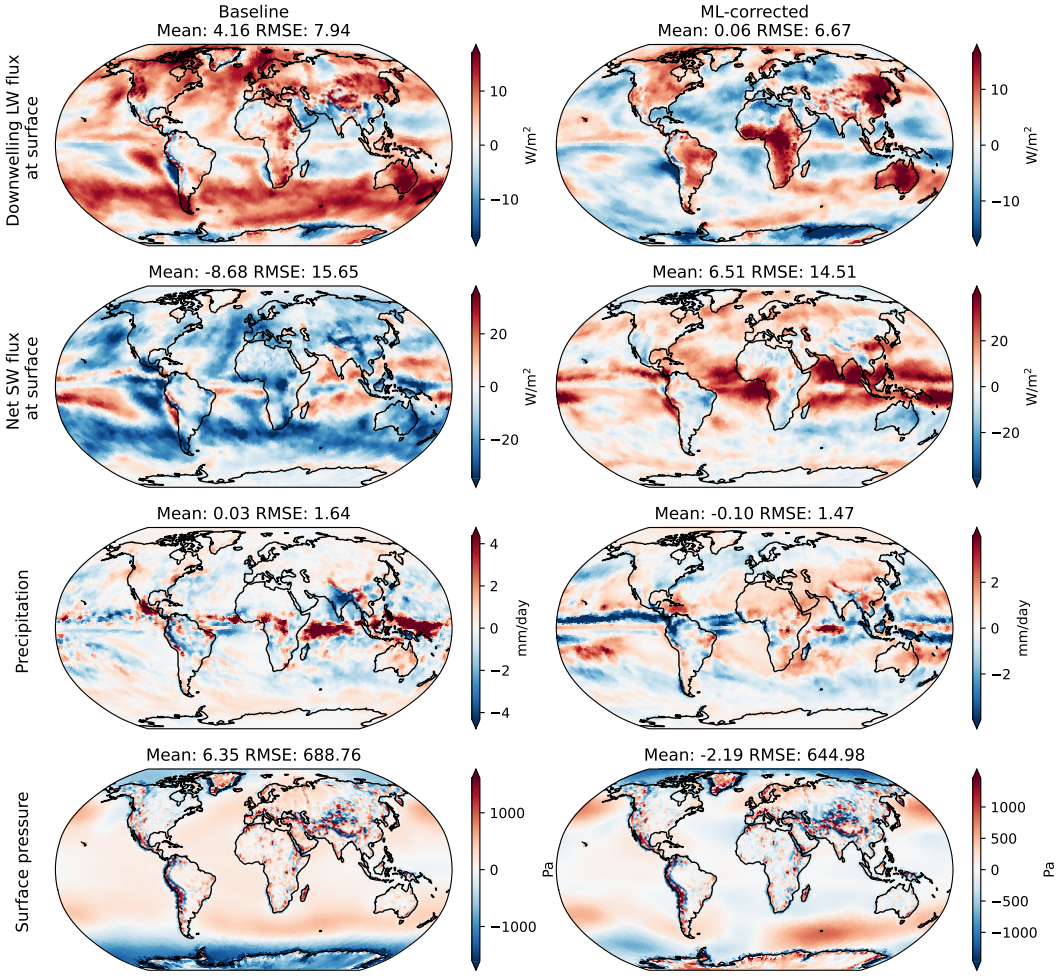

Figure 3. Spatial pattern of time mean simulation error with respect to the coarsened fine–grid reference simulation. Variables shown here, from top to bottom are the downwelling longwave flux and the net shortwave flux at the surface, the precipitation and the surface pressure. Shown here are the validation variables where the uncorrected coarse–resolution baseline (left column) has a higher RMSE than the coarse simulation including corrective–ML (right column). The RMSE is calculated as global area-weighted mean.

coarse GCM. A subsequent study could investigate what other parameters and input variables improve the performance of the ML models.

Adoption of the corrective–ML framework involved the development of several new features in EAMxx (https://github.com/E3SM-Project/E3SM). EAMxx is heavily unit tested, but there is always a chance that a difficult bug or untested edge case is able to make it past the testing. There are three major developments that were added to EAMxx as part of this effort; Python to C++ bridging for ML, ability to nudge the EAMxx model state, and prescribed atmospheric surface fluxes in EAMxx. While

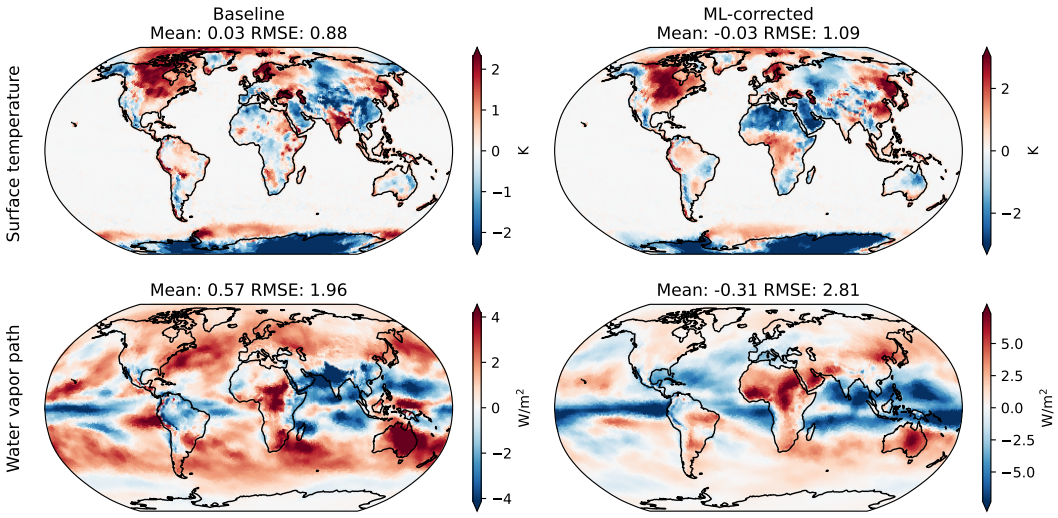

Figure 4. Same as figure 3, showing variables where the RMSE is worse for the ML–corrected simulation. These variables are the surface temperature (top) and the total integrated water vapor path (bottom).

significant effort went into developing and testing these new features, future work may want to revisit their implementation as a potential source of error in the ML implementation.

Finally, as outlined in the Methods section, the fine–resolution solution was coarsened to the target resolution both horizontally and vertically. The decision to map onto fixed vertical pressure coordinates resulted in situations where grid points over topography could lead to many or all data points being masked. This could create either masked data in the target state or grid points with disproportionate weighting based on only a few source columns. To investigate this hypothesis, a separate lightweight study was conducted using a 25 km fine–resolution simulation, comparing two corrective–ML models. One model utilized the approach described in this study, while the other did not apply vertical interpolation for the target state. Preliminary analysis showed that the latter model had slightly better performance. Unfortunately, due to the high cost of 3.25 km simulations, it was not feasible within the scope of this project to repeat this study using horizontal-only interpolation. A more detailed investigation of the impact from vertical interpolation should be included in any subsequent studies.

6 Conclusions

Although this work did not achieve the anticipated accuracy improvements from corrective–ML in EAMxx (Donahue et al., 2024), it did have several beneficial effects. In particular, this study marked the first time Python packages were integrated into EAMxx, providing a blueprint for future hybrid ML without degrading EAMxx performance on either CPU or GPU architectures. This work also led to the development of new infrastructure within EAMxx to support prescribed surface fluxes and atmospheric state nudging. While the latter was already a planned milestone for the regionally-refined configuration of

EAMxx, this effort expanded that capability. The introduction of prescribed surface fluxes is important for some doubly-periodic use cases and useful for debugging and hypothesis testing (Bogenschutz et al., 2025). Finally, the fact that EAMxx benefited less from corrective–ML than FV3GFS (Bretherton et al., 2022) is a useful data point in the quest to understand generalizability of ML approaches in GCMs.

There is increasing interest in the earth system modeling community in utilizing differentiable algorithms and software. This strategy has already been successfully applied to NeuralGCM, which marries a spectral dycore coded in the Jax machine learning language with a learned ML representation of the combined physical parameterizations of a global model, trained to evolve following a gridded reanalysis. In this setting, the ML model can be trained end-to-end including feedbacks with other earth system components within this differentiable framework, dramatically improving the prognostic performance of this hybrid model such that it has highly accurate weather forecasts and strong climate skill (Kochkov et al., 2024). In principle, an analogous approach could be implemented to dramatically improve the prognostic skill of corrective–ML within EAMxx. However, refactoring EAMxx to enable such a capability while retaining leadership-class performance would be a substantial software engineering challenge.

Code and data availability. We provide a Zenodo archival repository containing (1) the modified SCREAMv1 Code with the corrective-ML improvements and (2) the fv3net model used for ML training. The archival repository is available at https://zenodo.org/records/17469329 (Donahue and Wu, 2025a). The SCREAM code is publicly available at https://github.com/E3SM-Project/scream (specifically commit 3a0cee9be388187492bd9447d35f34d17bd44b94 used for simulations in this paper). The fv3net model used to train the corrective–ML model is open–source and available at https://github.com/ai2cm/fv3net (specifically commit 79edf0dc6f7a53fba3e2d42835def4975b33662b used for training in this work). Instructions for accessing the high resolution SCREAM simulation data using HPSS can be found at https://zenodo.org/records/14579433 (Terai, 2024b). The data is also publicly available for download via Zenodo. Given the size of the dataset, it has been divided by month and stored in 12 separate Zenodo repositories which can be found at (Donahue, 2026a), (Donahue, 2026b), (Donahue, 2026c), (Donahue, 2026d), (Donahue, 2026e), (Donahue, 2026f), (Donahue, 2026g), (Donahue, 2026h), (Donahue, 2026i), (Donahue, 2026j), (Donahue, 2026k), and (Donahue, 2026l). The version of the code that was used to perform the simulations is archived at https://zenodo.org/records/14578966 (Terai, 2024a). All data used for analysis and figure generation is publicly available at (Donahue and Wu, 2025b), doi: https://doi.org/10.5281/zenodo.17469234

Author contributions. AD, EW and WP wrote and tested the code. AD and EW prepared the training and validation data sets. CB, CG, PC and FR supervised experimental design and contributed to the interpretation of the results. AD and EW wrote the paper with feedback from WP, CB, CG, PC and FR.

Competing interests. Some authors are members of the editorial board of journal GMD.

Acknowledgements. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. IM Release number LLNPUB-2006061. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award BER-ERCAPm4492. The authors would especially like to thank the two anonymous reviewers whose careful review and invaluable suggestions greatly improved the manuscript. The authors acknowledge the use of an AI language model to assist with grammar correction and sentence structure improvements during manuscript preparation.

References

Bogenschutz, P. and Krueger, S. K.: A simplified PDF parameterization of subgrid-scale clouds and turbulence for cloud-resolving models, Journal of Advances in Modeling Earth Systems, 5, 195–211, https://doi.org/https://doi.org/10.1002/jame.20018, 2013.

Bogenschutz, P. A., Clevenger, T. C., Bradley, A. M., Caldwell, P. M., Beydoun, H., Mahfouz, N., Keen, N. D., Guba, O., Bertagna, L., Foucar, J., Zhang, J., and Donahue, A. S.: High Performance, High Fidelity: A GPU-Accelerated Doubly-Periodic Configuration of the Simple Cloud-Resolving E3SM Atmosphere Model Version 1 (DP-SCREAMv1), Journal of Advances in Modeling Earth Systems, 17, e2025MS005 127, https://doi.org/https://doi.org/10.1029/2025MS005127, e2025MS005127 2025MS005127, 2025.

Bretherton, C. S., Henn, B., Kwa, A., Brenowitz, N. D., Watt-Meyer, O., McGibbon, J., Perkins, W. A., Clark, S. K., and Harris, L.: Correcting Coarse-Grid Weather and Climate Models by Machine Learning From Global Storm-Resolving Simulations, Journal of Advances in Modeling Earth Systems, 14, e2021MS002 794, https://doi.org/https://doi.org/10.1029/2021MS002794, e2021MS002794 2021MS002794, 2022.

Caldwell, P. M., Terai, C. R., Hillman, B., Keen, N. D., Bogenschutz, P., Lin, W., Beydoun, H., Taylor, M., Bertagna, L., Bradley, A. M., Clevenger, T. C., Donahue, A. S., Eldred, C., Foucar, J., Golaz, J.-C., Guba, O., Jacob, R., Johnson, J., Krishna, J., Liu, W., Pressel, K., Salinger, A. G., Singh, B., Steyer, A., Ullrich, P., Wu, D., Yuan, X., Shpund, J., Ma, H.-Y., and Zender, C. S.: Convection-Permitting Simulations With the E3SM Global Atmosphere Model, Journal of Advances in Modeling Earth Systems, 13, e2021MS002 544, https://doi.org/https://doi.org/10.1029/2021MS002544, e2021MS002544 2021MS002544, 2021.

Carter-Edwards, H., Trott, C. R., and Sunderland, D.: Kokkos: Enabling manycore performance portability through polymorphic memory access patterns, J. Parallel and Dist. Comp., 74, 3202–3216, 2014.

Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.-J., Chen, X., Ma, L., Zhang, T., Su, R., Ci, Y., Li, B., Yang, X., and Ouyang, W.: FengWu: Pushing the Skillful Global Medium-range Weather Forecast beyond 10 Days Lead, http://arxiv.org/abs/2304.02948, arXiv:2304.02948 [physics], 2023.

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.: Correcting a 200 km Resolution Climate Model in Multiple Climates by Machine Learning From 25 km Resolution Simulations, Journal of Advances in Modeling Earth Systems, 14, e2022MS003 219, https://doi.org/https://doi.org/10.1029/2022MS003219, e2022MS003219 2022MS003219, 2022.

Dennis, J., Fournier, A., Spotz, W. F., St-Cyr, A., Taylor, M. A., Thomas, S. J., and Tufo, H.: High-resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric dynamical core, Int. J. High Perform. Comput. Appl., 19, 225–235, 2005.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 1 of 12, https://doi.org/10.5281/zenodo.18191405, 2026a.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 2 of 12, https://doi.org/10.5281/zenodo.18202660, 2026b.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 3 of 12, https://doi.org/10.5281/zenodo.18202837, 2026c.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 4 of 12, https://doi.org/10.5281/zenodo.18226680, 2026d.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 5 of 12, https://doi.org/10.5281/zenodo.18225295, 2026e.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 6 of 12, https://doi.org/10.5281/zenodo.18225633, 2026f.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 7 of 12, https://doi.org/10.5281/zenodo.18225841, 2026g.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 8 of 12, https://doi.org/10.5281/zenodo.18225985, 2026h.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 9 of 12, https://doi.org/10.5281/zenodo.18226245, 2026i.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 10 of 12, https://doi.org/10.5281/zenodo.18226448, 2026j.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 11 of 12, https://doi.org/10.5281/zenodo.18226541, 2026k.

Donahue, A.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset 12 of 12, https://doi.org/10.5281/zenodo.18226665, 2026l.

Donahue, A. and Wu, E.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) - Software, https://doi.org/10.5281/zenodo.17469329, 2025a.

Donahue, A. and Wu, E.: Applying Corrective Machine Learning in the E3SM Atmosphere Model in C++ (EAMxx) Dataset, https://doi.org/10.5281/zenodo.17469234, 2025b.

Donahue, A. S., Caldwell, P. M., Bertagna, L., Beydoun, H., Bogenschutz, P. A., Bradley, A. M., Clevenger, T. C., Foucar, J., Golaz, C., Guba, O., Hannah, W., Hillman, B. R., Johnson, J. N., Keen, N., Lin, W., Singh, B., Sreepathi, S., Taylor, M. A., Tian, J., Terai, C. R., Ullrich, P. A., Yuan, X., and Zhang, Y.: To Exascale and Beyond—The Simple Cloud-Resolving E3SM Atmosphere Model (SCREAM), a Performance Portable Global Atmosphere Model for Cloud-Resolving Scales, Journal of Advances in Modeling Earth Systems, 16, e2024MS004 314, https://doi.org/https://doi.org/10.1029/2024MS004314, e2024MS004314 2024MS004314, 2024.

Eyring, V., Collins, W. D., Gentine, P., Barnes, E. A., Barreiro, M., Beucler, T., Bocquet, M., Bretherton, C. S., Christensen, H. M., Dagon, K., et al.: Pushing the frontiers in climate modelling and analysis with machine learning, Nature Climate Change, 14, 916–928, https://doi.org/10.1038/s41558-024-02095-y, 2024.

Henn, B., Jauregui, Y. R., Clark, S. K., Brenowitz, N. D., McGibbon, J., Watt-Meyer, O., Pauling, A. G., and Bretherton, C. S.: A Machine Learning Parameterization of Clouds in a Coarse-Resolution Climate Model for Unbiased Radiation, Journal of Advances in Modeling Earth Systems, 16, e2023MS003 949, https://doi.org/10.1029/2023MS003949, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2023MS003949, 2024.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N.: The ERA5 global reanalysis, Quarterly Journal of the Royal Meteorological Society, 146, 1999–2049, https://doi.org/https://doi.org/10.1002/qj.3803, 2020.

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.: ICON-Sapphire: simulating the components of the Earth system and their interactions at kilometer and subkilometer scales, Geoscientific Model Development, 16, 779–811, https://doi.org/10.5194/gmd-16-779-2023, 2023.

Hu, Z., Subramaniam, A., Kuang, Z., Lin, J., Yu, S., Hannah, W. M., Brenowitz, N. D., Romero, J., and Pritchard, M. S.: Stable Machine-Learning Parameterization of Subgrid Processes in a Comprehensive Atmospheric Model Learned From Embedded Convection-Permitting Simulations, Journal of Advances in Modeling Earth Systems, 17, e2024MS004 618, https://doi.org/https://doi.org/10.1029/2024MS004618, e2024MS004618 2024MS004618, 2025.

Keisler, R.: Forecasting Global Weather with Graph Neural Networks, https://doi.org/10.48550/arXiv.2202.07575, arXiv:2202.07575 [physics], 2022.

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., Lottes, J., Rasp, S., Düben, P., Klöwer, M., Hatfield, S., Battaglia, P., Sanchez-Gonzalez, A., Willson, M., Brenner, M. P., and Hoyer, S.: Neural General Circulation Models, http://arxiv.org/abs/2311.07222, arXiv:2311.07222 [physics], 2023.

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.: Neural general circulation models for weather and climate, Nature, 632, 1060–1066, https://doi.org/10.1038/s41586-024-07744-y, 2024.

Kwa, A., Clark, S. K., Henn, B., Brenowitz, N. D., McGibbon, J., Watt-Meyer, O., Perkins, W. A., Harris, L., and Bretherton, C. S.: Machine-Learned Climate Model Corrections From a Global Storm-Resolving Model: Performance Across the Annual Cycle, Journal of Advances in Modeling Earth Systems, 15, e2022MS003 400, https://doi.org/https://doi.org/10.1029/2022MS003400, e2022MS003400 2022MS003400, 2023.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., and Battaglia, P.: GraphCast: Learning skillful medium-range global weather forecasting, http://arxiv.org/abs/2212.12794, arXiv:2212.12794 [physics], 2023.

Morrison, H. and Milbrandt, J. A.: Parameterization of cloud microphysics based on the prediction of the bulk ice particle properties. Part I: Scheme description and idealized tests, J. Atmos. Sci., 72, 287–311, 2015.

Okuta, R., Unno, Y., Nishino, D., Hido, S., and Loomis, C.: CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations, in: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS), http://learningsys.org/nips17/assets/papers/paper_16.pdf, 2017.

Pincus, R., Mlawer, E. J., and Delamere, J. S.: Balancing Accuracy, Efficiency, and Flexibility in Radiation Calculations for Dynamical Models, Journal of Advances in Modeling Earth Systems, 11, 3074–3089, https://doi.org/https://doi.org/10.1029/2019MS001621, 2019.

Price, I., Sanchez-Gonzalez, A., Alet, F., Ewalds, T., El-Kadi, A., Stott, J., Mohamed, S., Battaglia, P. W., Lam, R., and Willson, M.: GenCast: Diffusion-based ensemble forecasting for medium-range weather, https://www.semanticscholar.org/paper/f6e7e8b96b8ce1f32973318b3b88bc54f6eb4ab7, 2023.

Sanford, C., Kwa, A., Watt-Meyer, O., Clark, S. K., Brenowitz, N., McGibbon, J., and Bretherton, C.: Improving the Reliability of ML-Corrected Climate Models With Novelty Detection, Journal of Advances in Modeling Earth Systems, 15, e2023MS003 809, https://doi.org/https://doi.org/10.1029/2023MS003809, e2023MS003809 2023MS003809, 2023.

Satoh, M., Stevens, B., Judt, F., Khairoutdinov, M., Lin, S.-J., Putman, W. M., and Düben, P.: Global cloud-resolving models, Current Climate Change Reports, 5, 172–184, https://doi.org/10.1007/s40641-019-00131-0, 2019.

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.: 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, SC '23, Association for Computing Machinery, New York, NY, USA, ISBN 9798400701092, https://doi.org/10.1145/3581784.3627044, 2023.

Taylor, M. A., Guba, O., Steyer, A., Ullrich, P. A., Hall, D. M., and Eldrid, C.: An Energy Consistent Discretization of the Nonhydrostatic Equations in Primitive Variables, Journal of Advances in Modeling Earth Systems, 12, https://doi.org/10.1029/2019MS001783, 2020.

Terai, C.: Model code used for Cess-Potter experiments with SCREAMv1, https://doi.org/10.5281/zenodo.14578966, 2024a.

Terai, C.: Simulation output and input data from Cess-Potter experiments with SCREAMv1, https://doi.org/10.5281/zenodo.14579433, 2024b.

Terai, C. R., Keen, N. D., Caldwell, P. M., Beydoun, H., Bogenschutz, P. A., Chao, L.-W., Hillman, B. R., Ma, H.-Y., Zelinka, M. D., Bertagna, L., Bradley, A., Clevenger, T. C., Donahue, A. S., Foucar, J. G., Golaz, J.-C., Guba, O., Hannah, W. M., Lee, J., Lin, W., Mahfouz, N. G. A., Mülmenstädt, J., Salinger, A. G., Singh, B., Sreepathi, S., Qin, Y., Taylor, M. A., Ullrich, P. A., Wu, W.-Y., Yuan, X., Zender, C. S., and Zhang, Y.: Climate response to warming in Cess-Potter simulations using the global 3-km SCREAM, ESS Open Archive, https://doi.org/10.22541/essoar.173655643.33295443/v1, 2025.

Ullrich, P. A. and Taylor, M. A.: Arbitrary-Order Conservative and Consistent Remapping and a Theory of Linear Maps: Part I, Monthly Weather Review, 143, 2419 – 2440, https://doi.org/10.1175/MWR-D-14-00343.1, 2015.

Ullrich, P. A., Devendran, D., and Johansen, H.: Arbitrary-Order Conservative and Consistent Remapping and a Theory of Linear Maps: Part II, Monthly Weather Review, 144, 1529 – 1549, https://doi.org/10.1175/MWR-D-15-0301.1, 2016.

W. Jakob, J. Rhinelander, D. Moldovan and others: pybind11 – Seamless operability between C++11 and Python, 2017.

Zhou, L., Lin, S.-J., Chen, J.-H., Harris, L. M., Chen, X., and Rees, S. L.: Toward Convective-Scale Prediction within the Next Generation Global Prediction System, Bulletin of the American Meteorological Society, 100, 1225 – 1243, https://doi.org/10.1175/BAMS-D-17-0246.1, 2019.