Advancing Parsimonious Deep Learning Weather Prediction using the HEALPix Mesh

Matthias Karlbauer[1,2], Nathaniel Cresswell-Clay[2], Dale R Durran[2], Raul A Moreno[2], Thorsten Kurth[3], and Martin V Butz[1]

[1]Neuro-Cognitive Modeling Group, University of Tübingen
[2]Department of Atmospheric Sciences, University of Washington
[3]NVIDIA Switzerland AG

March 13, 2024

Posted on 30 Sep 2023 — The copyright holder is the author/funder. All rights reserved. No reuse without permission. — https://doi.org/10.22541/essoar.169603505.58030377/v1 — This is a preprint and has not been peer-reviewed. Data may be preliminary.

Advancing Parsimonious Deep Learning Weather Prediction using the HEALPix Mesh

Matthias Karlbauer[1], Nathaniel Cresswell-Clay[2], Dale R. Durran[2], Raul A. Moreno[2], Thorsten Kurth[3], Boris Bonev[3], Noah Brenowitz[4], and Martin V. Butz[1]

[1]Neuro-Cognitive Modeling Group, Department of Computer Science, University of Tübingen, Tübingen, Germany
[2]Department of Atmospheric Sciences, University of Washington, Seattle, WA, USA
[3]NVIDIA Switzerland AG, Zürich, Switzerland
[4]NVIDIA Corporation, Seattle, USA

Key Points:

- A U-Net is refined to forecast seven atmospheric variables on global scale, falling behind the state-of-the-art by only one day.
- Forecasts are generated on the HEALPix mesh, facilitating the development of location invariant convolution kernels.
- Without converging to climatology, the model produces stable and realistic states of the atmosphere in 365-days rollouts.

Corresponding author: Dale R. Durran, `drdee@uw.edu`

Abstract

We present a parsimonious deep learning weather prediction model on the Hierarchical Equal Area isoLatitude Pixelization (HEALPix) to forecast seven atmospheric variables for arbitrarily long lead times on a global approximately 110 km mesh at 3h time resolution. In comparison to state-of-the-art machine learning weather forecast models, such as Pangu-Weather and GraphCast, our DLWP-HPX model uses coarser resolution and far fewer prognostic variables. Yet, at one-week lead times its skill is only about one day behind the state-of-the-art numerical weather prediction model from the European Centre for Medium-Range Weather Forecasts. We report successive forecast improvements resulting from model design and data-related decisions, such as switching from the cubed sphere to the HEALPix mesh, inverting the channel depth of the U-Net, and introducing gated recurrent units (GRU) on each level of the U-Net hierarchy. The consistent east-west orientation of all cells on the HEALPix mesh facilitates the development of location-invariant convolution kernels that are successfully applied to propagate global weather patterns across our planet. Without any loss of spectral power after two days, the model can be unrolled autoregressively for hundreds of steps into the future to generate stable and realistic states of the atmosphere that respect seasonal trends, as showcased in one-year simulations. Our parsimonious DLWP-HPX model is research-friendly and potentially well-suited for sub-seasonal and seasonal forecasting.

Plain Language Summary

Weather forecasting is traditionally realized by numerical weather prediction models that solve physical equations to simulate the progression of the atmosphere. Numerical methods are compute intense and their performance is increasingly challenged by less compute demanding but still highly sophisticated machine learning approaches. Yet, a downside of these new models is their reliability: They are not guaranteed to generate physically plausible states, which often prevents them from generating stable and realistic forecasts beyond two weeks into the future. Here, a parsimonious machine learning model is developed to forecast just seven variables of the atmosphere (compared to more than 800 in numerical models and 67 or 218 in competitive machine learning models) over an entire year. Despite the small number of variables, our model generates forecasts that only fall behind expensive state-of-the-art predictions by a single day. That is, our error in a seven-days forecast matches that of a state-of-the-art forecast at day eight. Advancing weather forecasts with research friendly and parsimonious machine learning models beyond two weeks promises to extend horizons for planning in various fields that impact environment, economy, and society.

1 Introduction

Four years ago, Weyn et al. (2019) posed the question “Can machines learn to predict the weather?” and demonstrated that data driven convolutional neural networks can forecast the evolution of the 500 hPa surface much better than the alternative dynamical model, the barotropic vorticity equation, which was used in the first numerical weather prediction (NWP) model (Charney et al., 1950). An extremely rapid evolution of deep learning weather prediction (DLWP) models followed, culminating in the recent Pangu-Weather (Bi et al., 2023) and GraphCast models (Lam et al., 2022), which outperform the deterministic forecast from the state-of-the-art Integrated Forecast System (IFS) of the European Centre for Medium-Range Weather Forecasts (ECMWF).

NWP has continuously improved over the seven decades since the first barotropic model forecast (Benjamin et al., 2019). Current state-of-the-art models typically provide skillful predictions of global weather patterns at effective grid point spacings of roughly 0.1° of latitude (about 10 km) through at least seven days of forecast lead time (Bauer et al., 2015). The computational effort required to generate such global high-resolution forecasts is enormous and only available at a handful of advanced dedicated centers. Ensemble forecasts, which provide an important way to account for uncertainty by generating a set of equally plausible predictions and extend the limit of skillful forecasts beyond that of a single deterministic model run, are also limited by the computational burden of high-resolution NWP to about 50 members (Palmer, 2019).

Global NWP models represent 3D fields as sets of nested spherical shells in which the distance between each shell is the local vertical grid spacing. On every time step, the ECMWF Integrated Forecasting System (IFS), as configured for sub-seasonal forecasting, updates 10 prognostic 3D variables defined at 91 vertical levels. Along with surface pressure, this totals to over 900 spherical shells of data. Here, we use "spherical shell of data" to describe a single variable defined at a single vertical level on a spherical shell covering the globe. The large number of spherical shells of data (combined with the fine horizontal resolution) in NWP models is required to produce acceptably accurate numerical solutions to the equations governing atmospheric motions. The data at each individual point, however, cannot be independently perturbed while maintaining a meteorologically relevant atmospheric state. For example, on horizontal scales larger than about 10 km, the temperatures throughout a vertical column and the heights of constant pressure surfaces must satisfy hydrostatic balance.

The actual number of independent degrees of freedom required to represent the predictable components of the global atmosphere is unknown, but it clearly decreases with increasing forecast lead times (Lorenz, 1969). GraphCast (Lam et al., 2022), for example, has achieved success at lead times as short as 6 h with 227 spherical shells of data. It can produce forecasts using much less computation time than the ECMWF IFS, but it still requires large computing resources for training: 3 weeks using 32 TPU 4 processors. Pangu-Weather (Bi et al., 2023) cuts the number of spherical shells by almost 2/3 to 69. The spherical Fourier neural operator (SFNO) version of FourCastNet compared with the IFS in (Bonev et al., 2023) uses 73 spherical shells of data. Here, we take this reduction much farther, presenting a parsimonious DLWP model that uses just 7 spherical shells of data to efficiently provide forecasts approaching the skill of ECMWF. While not as accurate as GraphCast or Pangu-Weather for medium range forecasts with lead times less than two weeks, we demonstrate that our model generates far less bias in forecasts of 500 hPa height in one-year iterative forecasts. In addition, our model is potentially better suited for research applications such as computing the sensitivities of its compact state vector to custom diagnostic functions by backpropagation.

In contrast to many of the recent DLWP architectures, our approach relies on convolutional neural networks (CNN), building on early work by Scher and Messori (2018) and Weyn et al. (2019) and the U-Net configuration in Weyn et al. (2020) and Weyn et al. (2021). Here, we document substantial improvements over Weyn et al. (2021), obtained by replacing the cubed sphere data representation with the HEALPix mesh, which is widely employed in astronomy (Gorski et al., 2005). In addition, we improve the former model by implementing physically motivated modifications in form of residual connections, recurrent modules, and inverting the channel depth as compared with a standard U-Net.

2 Related Work

Pioneering efforts to create machine learning models to forecast the weather from reanalysis or general circulation model (GCM) output include the dense neural network of Dueben and Bauer (2018) and the CNN models of Scher and Messori (2019) and Weyn et al. (2019), all of which employed latitude longitude (lat-lon) meshes. Weyn et al. (2020) obtained significantly improved forecasts by switching to a cubed sphere mesh with a CNN in the standard U-Net architecture (Ronneberger et al., 2015). Their model was capable of generating realistic weather patterns when stepped forward for a full year (730 12 h steps). Retaining the cubed sphere, Weyn et al. (2021) produced forecasts out to sub-seasonal time scales using large multi-model ensembles, and Lopez-Gomez et al. (2022) migrated from the U-Net into a U-Net 3+ architecture (Huang et al., 2020)—which adds connections between multiple hierarchical levels in the U-Net—to generate forecasts of extreme surface temperatures.

Returning to the lat-lon mesh, Rasp and Thuerey (2021) demonstrated that a deep Resnet could be pre-trained on GCM data and then fine-tuned by transfer learning on ERA5 data to produce up to 5-day forecasts at coarse 5.65 ° grid spacing. Building on transformer models from computer vision (Dosovitskiy et al., 2020; Guibas et al., 2021), Pathak et al. (2022) and Kurth et al. (2022) used Fourier neural operators (Li et al., 2020) to develop FourCastNet on a 0.25 ° lat-lon mesh to generate forecasts approaching the accuracy of ECMWF's IFS. FourCastNet was not, however, capable of stable long-lead-time autoregressive rollouts. This difficulty was overcome by switching from 2D Fourier modes on a lat-lon mesh to spherical harmonic functions Bonev et al. (2023). The resulting SFNO model eliminated much of the vision transformer architecture while improving accuracy and remaining stable for one-year forecasts.

Again on a 5.65° lat-lon mesh, Hu et al. (2022) used a shifted window (Swin) transformer (Liu et al., 2021) to produce single forecasts as well as ensembles generated by perturbing the latent state using samples from their learned distribution. Bi et al. (2023) also applied Swin transformers on a lat-lon mesh, but used a fine 0.25 ° lat-lon grid spacing, 3D transformers, and included latitude and longitude fields as input to train a "3D Earth-specific transformer" at four different forecast lead times of 1, 3, 6, and 24 h, which are used in combination to span an arbitrary hourly forecast period with minimal model steps. If the ECMWF IFS NWP forecasts are averaged to the coarser 0.25 ° lat-lon mesh, Pangu-Weather outperforms NWP on several metrics.

In contrast to the preceding approaches, graph neural networks (Gori et al., 2005; Scarselli et al., 2008; Kipf & Welling, 2016; Battaglia et al., 2018; Pfaff et al., 2020) where applied on icosahedral meshes at course resolution by Keisler (2022) and at fine resolution in the Graphcast model (Lam et al., 2022). Similarly to Pangu-Weather, GraphCast appears to outperform the coarsened ECMWF IFS forecast on several metrics.

3 Methods

3.1 Data

3.1.1 Choice of Variables

Beginning with the same six prognostic variables used in Weyn et al. (2021)—geopotential height at 1000 hPa and 500 hPa (Z_{1000}, Z_{500}),[1] 700 hPa to 300 hPa thickness ($\tau_{700-300}$) defined as $Z_{300} - Z_{700}$, temperature at 2 m height above ground (T_{2m}), temperature at 850 hPa (T_{850}), and total column water vapor ($TCWV$)—we add Z_{250} based on its importance in the model of Rasp and Thuerey (2021) and to provide an upper tropospheric variable. As in Weyn et al. (2021), three prescribed fields are also provided: topographic height, land-sea mask, and top-of-atmosphere (TOA) incident solar radiation. We do not include prescribed or predicted sea-surface temperature or surface fluxes above the land or ocean. *No* specific information about position on the globe, such as latitude and longitude, is provided. Three-hourly data from the ERA5 reanalysis (Hersbach et al., 2020) provide training data from 1979-2012, a validation set from 2013-2016, and a test set from 2017-2018.

[1] The related variable in the ERA5 dataset is geopotential and named z, whereas the geopotential height, typically referred to as Z, represents the actual height above sea level of the respective pressure surface and is obtained by dividing geopotential by the gravitational constant.

3.1.2 HEALPix Mesh

We discretize all fields using the Hierarchical Equal Area isoLatitude Pixelization (HEALPix) (Gorski et al., 2005). As depicted in Figure 1, a HEALPix mesh is formed by dividing the sphere into twelve equal-area diamond-shaped faces, with four faces lying in the northern and southern hemispheres, and four in the tropics. According to Gorski et al. (2005), the HEALPix mesh has three important properties. *(1) Hierarchical structure of the database:* Each of the twelve base faces can be progressively subdivided into smaller patches. *(2) Equal areas for the discrete elements of the partition:* All patches are the same size. *(3) Isolatitude distribution for the discrete area elements on the sphere:* The patches line up with lines of latitudes, facilitating the computation of zonal averages and one-dimensional zonal spectra. Importantly, this last property makes the HEALPix mesh an "east is to the right" grid, which facilitates the training of CNN kernels to capture the motion of typical weather disturbances, as discussed in subsection 4.1.

The HEALPix can be considered a graph and does not allow a seamless application of convolution operations. Thus, Perraudin et al. (2019) explicitly define a graph from the HEALPix—by connecting adjacent neighbors with weighted edges—and perform a graph convolution to classify weak lensing maps from cosmology. In a different approach, Krachmalnicoff and Tomasi (2019) classify digits and determine cosmic parameters from simulated cosmic microwave background maps. They apply 1D convolutions to the flattened HEALPix data with a kernel size k and stride s both equal to 9, appending a zero to those cases where only seven instead of eight neighbors are defined (top corner of the tropical faces). In contrast, we treat the twelve faces as distinct images and pad their boundaries using data from neighboring faces to allow the computation of 2D convolutions and averaging operators directly, as detailed in section Appendix A. To accelerate the padding operation, we have implemented a custom CUDA kernel, which is available in our repository.[2]

The grid spacing, or shortest inter-node spacing, on the HEALPix mesh is the diagonal distance between a pair of nodes on adjacent latitude lines. Denoting a HEALPix mesh with n divisions along one side of the original 12 faces as HPXn. The grid spacing is approximately 220 km ($\approx 2°$) for HPX32 and 110 km ($\approx 1°$) for HPX64.[3]

3.2 Machine Learning Architecture

Relating to Tobler's first law of geography: "All things are related, but nearby things are more related than distant things." (Tobler, 1970), we mostly retain the comparably simple U-Net structure from Weyn et al. (2020). U-Nets (Ronneberger et al., 2015) are hierarchically structured feed-forward convolutional neural networks that were originally proposed for segmenting biomedical images. The U-Net structure proposed here introduces several physically motivated advancements to the vanilla U-Net used by Weyn et al. (2021) for time-series forecasting. The advancements and model configurations are visualized in Figure 2, detailed in Table B1, and described in the following.

3.2.1 Residual Prediction

We switch to a residual prediction approach both for the full predictive step and within each ConvNeXt block.[4] Predicting changes over a time step, instead of the full fields, is similar to the discretization of time derivatives when solving partial or ordinary

[2] `https://github.com/CognitiveModeling/dlwp-hpx`

[3] We provide download explanations and projection scripts in our repository. The 3D HEALPix figures are drawn in Blender 3.4.1; respective Blender files are provided in the repository too.

[4] As detailed in Figure 2, we modify the original ConvNeXt block from Liu et al. (2022) by removing the bottleneck and employing a two-stage convolution as done in Weyn et al. (2021).

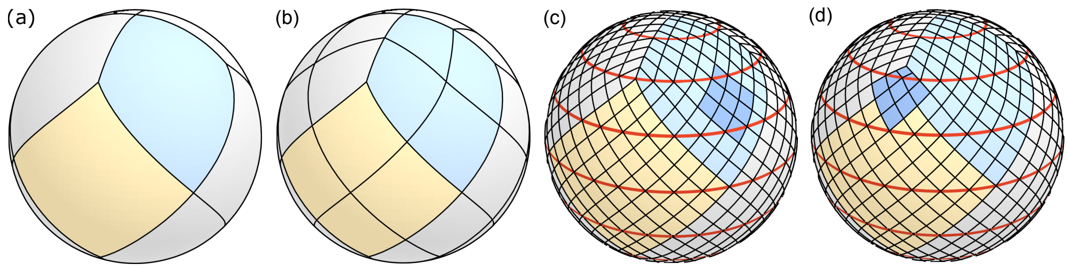

Figure 1: Division of the sphere into twelve faces according to the HEALPix. Four faces to represent either the northern (blue) and southern extratropics, while four more faces arrange around the equator to represent the tropics (yellow). Each face can be subdivided into patches with divisions along the side of each face given by powers of two. The sphere in (a) has a pixel-count of one per face side; we call it `hpx1`. The sphere in (b) counts two pixels per side (`hpx2`), whereas the two spheres in (c) and (d) have eight pixels per side, i.e., `hpx8`. Several latitude lines in red emphasize the iso-latitudinal arrangement of the patches. The saturated blue area depicts a 3×3 stencil, as applied by a standard convolution. To apply the 3×3 stencil at the top corner of the equatorial faces, i.e., stencil position in (d), we simulate a hypothetical patch by computing the average from the according extratropical face patches.

differential equations, and has been used successfully in previous deep-learning weather prediction models (Pathak et al., 2022; Keisler, 2022; Hu et al., 2022; Lam et al., 2022).

3.2.2 Inverting the Ordering of Channel Depth

The standard U-Net for semantic segmentation (Ronneberger et al., 2015) and its successors (Zhou et al., 2018; Huang et al., 2020) employ relatively few channels on the highest level and successively double the channel depth, while halving the spatial resolution in each deeper layer. This ordering is useful in image segmentation tasks, where deeper channels are required to create increasingly abstract filters to identify semantic features and express complex objects. In weather prediction, however, we find it is better to devote more capacity to the layers in the first level, where a wide variety of fine grained weather phenomena must be captured. Deeper layers at coarser resolution, on the other hand, need only encode larger scale atmospheric motions, which can be adequately represented with comparably fewer channels.

Thus, we invert the channel order, employing 136, 68, and 34 channels in each convolution on the first, second, and third layer, respectively (cf. Figure 2). While this modification improves the model performance significantly, it also increases the computational burden, since more computations and data processing are required to evaluate the additional convolutions at fine spatial resolution. Tests which preserved the total number of trainable parameters, but completely eliminated the deeper layers in the U-Net gave worse results, demonstrating that the longer-range connections and richer latent space structures enabled by the full U-Net architecture remain important.

3.2.3 Recurrent Modules

The vanilla U-Net is a feed-forward network, which treats successive inputs independently even if the data represents a continuous sequence over time. Feed-forward networks do not have any memory capacity. They do not maintain an internal state between time steps. To enable the exploitation of information from previous latent states, we include a gated recurrent unit (GRU) (Cho et al., 2014) at the end of each decoder block

Figure 2: Schematic representation of the DLWP-HPX architecture for our best performing model. There is one ConvNeXt block at each level in both the encoder and the decoder. In contrast to the configuration in typical image processing applications, the channel, or latent-layer, depth decreases from 136 to 68 to 34 at deeper layers in the U-Net.

with kernel size $k = 1$. We chose GRUs over LSTMs (Hochreiter & Schmidhuber, 1997) since we re-initialize the recurrent data over each 24 h-cycle, and therefore do not require forget-gates (as confirmed experimentally, not shown).

3.2.4 Miscellaneous Modifications

Several other components of the original Weyn et al. (2021) model were modified based on recent results from deep learning research: the capped leaky ReLU was replaced by capped GELU activations (Hendrycks & Gimpel, 2016); upsampling was changed from nearest-neighbor sampling (knn-sampling with $k = 1$) to a transposed convolution; finally, the pairs of two successive convolutions were replaced at each encoder and decoder level in the U-Net by a modified ConvNeXt block (Liu et al., 2022), as visualized in Figure 2.

3.2.5 Time Stepping Scheme

Similarly to Weyn et al. (2021), we apply a two-in-two-out mapping with a temporal resolution twice as fine as the actual time step. For example, two atmospheric states 3 h apart (each consisting of seven prognostic, along with three prescribed fields) are concatenated and input to the model, which generates a new pair of states, each characterising the atmosphere 6 h later in time. This strategy is observed to stabilize and accelerate the training, since the model receives additional information about the atmosphere's rate of change and only has to be called half as often.

Table 2: Root mean squared errors (RMSE) and anomaly correlation coefficient (ACC) scores for Weyn et al. (2021) (W21), our HPX64, and ECMWF's IFS models, evaluated on geopotential at 500 hPa (Z_{500}), temperature 2 m above ground (T_{2m}), and temperature at 850 hPa (T_{850}) on lead times of 3 and 5 days.

		Z_{500}			T_{2m}			T_{850}		
	Lead time	W21	HPX64	IFS	W21	HPX64	IFS	W21	HPX64	IFS
RMSE	3 days	36.26	21.88	14.91	1.17	0.82	1.02	1.95	1.49	1.35
	5 days	59.01	41.91	31.30	1.67	1.27	1.27	2.83	2.28	1.96
ACC	3 days	0.90	0.96	0.98	0.84	0.92	0.91	0.84	0.91	0.94
	5 days	0.70	0.84	0.92	0.66	0.78	0.83	0.64	0.76	0.84

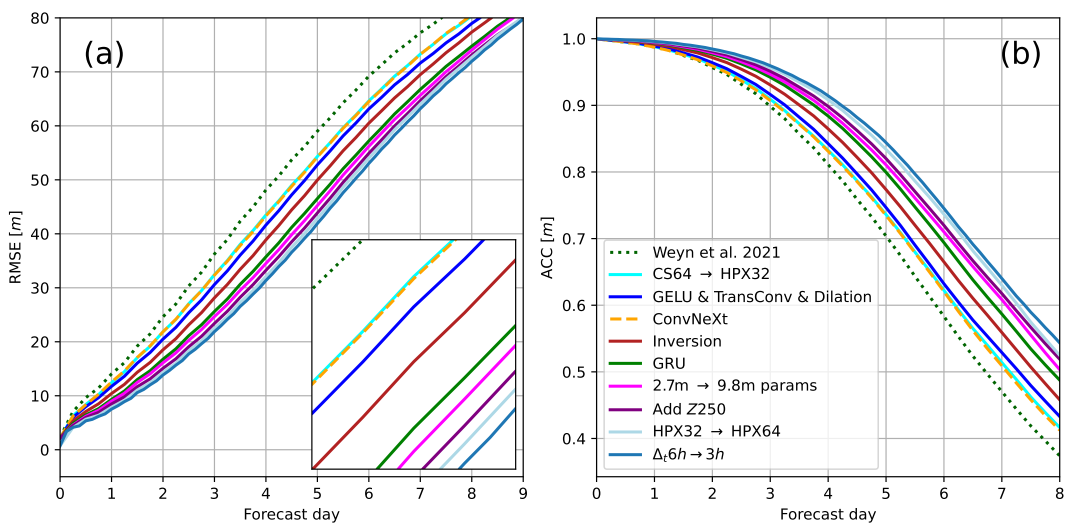

Figure 5: Impact of successive model improvements on the accuracy of Z_{500} building from WDCC to our HPX64 model with $\Delta_t = 3$ h. Each successive change builds on top of the previous architecture, adding the modification indicated in the legend: (a) RMSE, (b) ACC. Inset in (a) provides a magnified view of the error growth between 5 and 6 forecast days.

field, increasing the horizontal resolution to HPX64 (which is more important for ACC than RMSE particularly on T_{2m}), and decreasing the time resolution to 3 h. Benefits from the use of 3 h time resolution were only obtained if the model was configured with the GRUs.

The single most effective modification in the preceding set of successive improvements is the migration from the cubed sphere to the HEALPix mesh, even though the 64×64 cubed sphere has twice the total number of grid-points as the HPX32 mesh. The most likely explanation for the superiority of the HEALPix mesh is not simply that it is a more uniform covering of the globe than that provided by the cube-sphere, but that east and west have the same orientation in every HEALPix cell; we refer to this property as "east to the right." In particular, the center and the east and west corners of each HEALPix cell are all at the same latitude. (A similar relationship holds in the north-south direction for meridians passing through those cells lying equatorward of the maximum north-south extent of the four equatorial faces in Figure 1 (a).) Thus, on the HEALPix mesh, eastward motion at all points and at all latitudes would be in the same direction

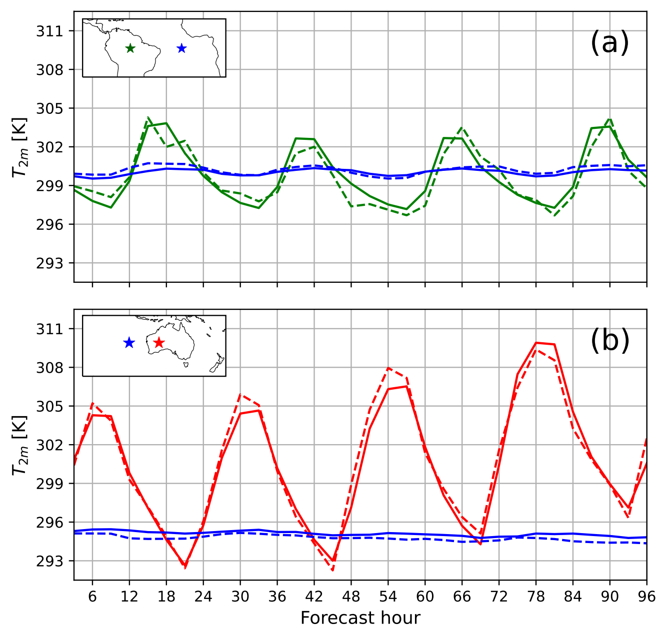

Figure 6: HPX64 simulation of the diurnal cycle of T_{2m} (solid curves) at the four locations shown in the insets starting from 00 UTC on 12 March 2018. ERA5 values for the same $1^\circ \times 1^\circ$ lat-lon cell are shown as dashed lines. Values are plotted every 3 h.

across the diamond-shaped 3×3 stencil in Figure 1 (c). In contrast, at any point on either of the polar faces on the cubed sphere, east could map to any of four directions along the axes of the 3×3 convolutional stencil, depending on its longitude, as visualized in section Appendix A.

In mid- and high-latitudes, most large-scale weather systems move in a generally eastward direction. We believe this allows a fixed number of kernel elements to more efficiently and effectively produce the required set of flow evolutions in the latent layers. To a lesser extent, this same consideration also applies to the four equatorial faces of the cubed sphere, where, for example, eastward flow near the northeastern corner of a face would need to move at an angle relative to the northern side of the stencil that is opposite in sign to that required in the northwestern corner.

4.2 Eliminating the Need for Boundary-Layer Parameterizations

Accurate forecasts of surface temperatures in NWP models rely on the empirical parameterization of multi-scale processes near the Earth's surface in the atmospheric boundary layer (ABL). The bottom of the ABL includes the roughness layer (2–5 times the height of roughness elements such as vegetation), and the surface layer (often 10–100 m deep), where shear-driven turbulence dominates generation by convection. The depth of the full ABL, where larger-scale eddies and circulations communicate the processes in the surface layer to the free atmosphere, can vary from $O(100)$ m in calm stable nighttime conditions to several kilometers during the day over deserts.

No effort is made to explicitly account for ABL processes in our model; the T_{2m} field is treated the same as the other six prognostic fields. The same CNN kernels are employed everywhere over the globe on the HEALPix mesh; the only data that might distinguish one location from another are the land-sea mask, the terrain elevation, and the TOA solar forcing; neither longitude nor latitude are provided. Yet our model does a good job of capturing the diurnal cycle in multi-day forecasts over very different surfaces. Figure 6 shows the diurnal cycle in T_{2m} at locations over the Amazon forest, the Australian desert, and two adjacent oceans over a 4-day simulation starting at 00 UTC on 12 March 2018.

Compared to over land, the diurnal T_{2m} variations are modest over the oceans, and they are well captured by our model. The land-sea mask is undoubtedly important in distinguishing the ocean locations from those over land. More interestingly, the model does an excellent job of capturing the large diurnal temperature range over the Australian desert, while correctly generating a much lower amplitude signal over the Amazon. The prognostic field that has most likely facilitated this distinction is $TCWV$, which is significantly higher over the Amazon than over the Australian desert. The model also captures the 4-day trend for increasing temperatures over Australia, which is linked to the evolution of larger-scale weather systems. Overall, the ability of the model to capture the diurnal T_{2m} cycle with just seven prognostic fields, without any special treatment of the ABL, and without geo-specific inputs such as latitude and longitude is suggestive of the power and potential of DLWP-HPX.

4.3 Iterative Rollouts Over Subseasonal to Annual Time Scales

There are three time scales of primary interest for global atmospheric simulations: medium-range weather forecasting for lead times of up to two weeks, sub-seasonal and seasonal forecasts for lead times up to 6–9 months, and climate simulations over periods of tens to hundreds of years. Our focus is on the sub-seasonal to seasonal time scale; therefore, in this section we examine the model's performance in iterative rollouts over periods up to one year.

To investigate the stability and drift in model simulations over a full annual cycle, we initialize it using ERA5 data for 00 UTC on 1 June 2017 (together with the 21 UTC fields on 31 May). Using 6 h time steps (with 3 h time resolution), we perform 1460 iterations to generate a 365-day simulation. The three-day running mean of Z_{500}, averaged around each latitude, is plotted as a function of latitude and time in Figure 7, along with the corresponding averages from the ERA5 data. Despite being trained to minimize RMSE over a single day and not enforcing any physical constraints, the DLWP-HPX simulation responds to the TOA solar forcing to generate the annual cycle reasonably well.

One region where the errors are significant is the arctic. About 5 months into the simulation, the simulated heights in the arctic region drop as much as 60 m below those in the reanalysis during the boreal winter. In contrast, at 5–8-month lead times, the heights in the antarctic region increase to approximately correct values in the austral summer. The asymmetry between the response in arctic and antarctic flips if the one-year rollout begins six months later. When the simulation is initialized on January 2, 2018, the heights in the arctic during boreal winter are approximately correct, while those in the antarctic are too cold (Figure 8d).

There is also a long-term drift toward lower heights in the subtropics and mid-latitudes, creating a roughly 30 m loss in Z_{500} by the end of the 1-year forecast.[7] Climate models are tuned to avoid long-term drift in the predicted fields, but operational NWP models

[7] 30 m deviation amounts to 0.5% of the full Z_{500} value and to 8.7% of the Z_{500} standard deviation (computed from the reanalysis data of the forecasted period).

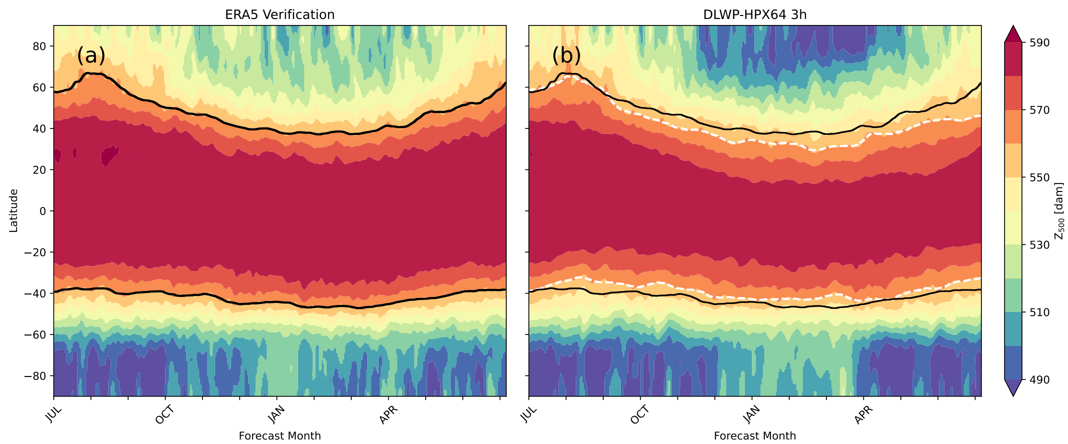

Figure 7: Zonally averaged three-day mean of Z_{500} plotted as a function of time and latitude for one year beginning on July 1 2017 for: (a) the ERA5 reanalysis, and (b) a recursive one-year rollout of the DLWP-HPX model. Also shown are 15-day averaged values of the 5600 m contour of Z_{500} for the ERA5 data (black lines) the DLWP-HPX simulation (white dashed lines).

are not so tuned. For example, significant model biases that grow over a time scale of several weeks are removed to create sub-seasonal ECMWF IFS S2S forecasts (Vitart, 2004; Weigel et al., 2008). To facilitate comparison of model drift with the ERA5 reanalysis, the pair of black lines in both panels show the 15-day mean of the zonally averaged 560-dam Z_{500} contours in the northern and southern hemisphere. The white lines in Figure 7b show the corresponding 560-dam Z_{500} contours for the DLWP-HPX simulation. The drift toward lower heights starts to become evident after two months in the northern hemisphere and continues to grow slowly for the remainder of the year. Differences show up earlier in the southern hemisphere, but the average drift is smaller and even disappears at a few times later in the year. As will be discussed in a forthcoming paper, both the errors near the poles and the drift in the tropics in Z_{500} can be corrected by incorporating SST forecasts from a coupled atmosphere-ocean model.

The performance of three additional state-of-the-art DLWP models is compared with our model using this same metric in Figure 8, which shows the evolution of zonally averaged Z_{500} heights over a one-year rollout beginning January 2, 2018. This year is part of the test set for all of the models: our DLWP-HPX, Pangu-Weather, GraphCast, and FourCastNetv2 based on spherical Fourier neural operators (SFNO) (Bonev et al., 2023). Details about the code used to generate these rollouts can be found in section Appendix B.

The Pangu-Weather model does not include solar forcing, and therefore, it does not follow the annual cycle. When stepped forward with a 24-h time step (Figure 8b), significant drift is apparent after about 1.5 months, which grows through the year without pushing the simulation into grossly unrealistic states. Based on the discussion of Extended Data, Fig. 7a in (Bi et al., 2023), one would not expect good performance from Pangu-Weather if rolled out with a 3-h time step, and indeed the 3-h rollout starts to produce significant errors after 1.5 months and generates completely unrealistic results after about 5 months (Figure 8f). We nevertheless, show its performance to contrast it with our 3-h-time-resolution rollout (Figure 8e).

The version of GraphCast from NVIDIA's Earth2MIP gives reasonable results for just the first 1.5 months (Figure 8c), while that from DeepMind goes bad after a cou-

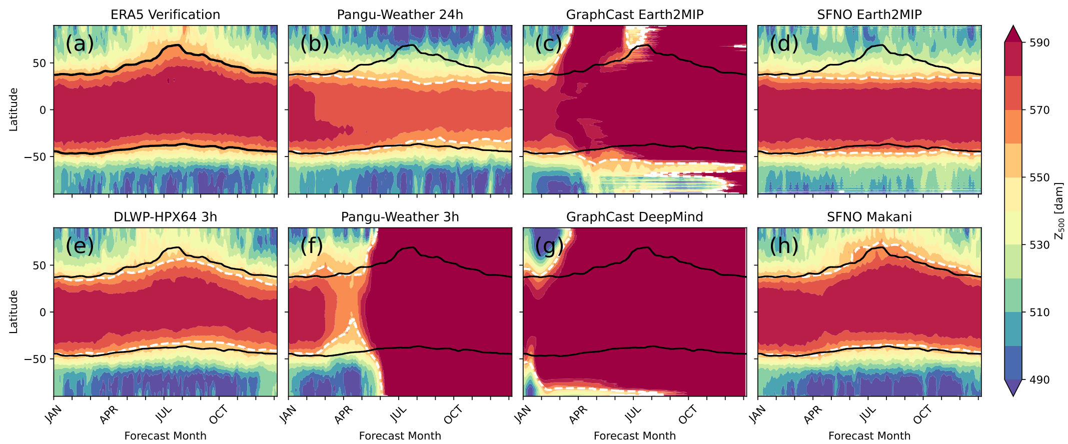

Figure 8: Zonally averaged three-day mean of Z_{500} plotted as a function of time and latitude: (a) for ERA5 reanalysis, (b)-(h) for recursive one-year simulations for each model as identified in the titles, initialized on January 2, 2018. Also shown are 15-day averaged values of the 5600 m contour of Z_{500} for the ERA5 data (black lines) each model simulation (white dashed lines).

ple weeks (Figure 8g). The SFNO Earth2MIP model (FourCastNetv2-small) shows essentially no drift over a full year (Figure 8d), although surprisingly, it does not follow the annual cycle despite including solar zenith angle as an input field. Some artifacts (horizontal stripes) are visible near the south pole within a month and at the north pole much later in the simulation. In contrast, the SFNO Makani model (Figure 8h), also includes solar zenith angle as an input field, and it does follow the annual cycle reasonably well. On balance, the performance of the SFNO Makani model is roughly similar to our DLWP-HPX model; it has larger errors near the poles, but less drift in the tropics.

In an ablation study (not shown), we investigated the effect of the top-of atmosphere solar forcing input on the 365-day DLWP-HPX rollout by training a model that did not receive solar forcing input. In that case, the model still generated a stable forecast over the entire rollout period, but did not produce the full annual cycle. Interestingly, that simulation did roughly approximate the transition from summer into a perpetual autumn.

One qualitative way to appreciate the stable behavior of our one-year simulations is illustrated by comparing a 360.5 day simulation initialized on 1 April 2017 (with 6 h time steps and 3 h resolution) and the corresponding 27 March 2018 reanalysis in Figure 9. The roughly one-year lead time is well beyond the limits of atmospheric predictability, so there is no reason to expect a close match between simulation and reanalysis. The 360.5-day simulation time was chosen to display the simulated strong low-pressure center in the northeastern Pacific. The intensity of the system is typical for strong systems in our simulation, but about 40 m higher than the strongest systems periodically appearing in the ERA5 reanalysis. Lower-amplitude signals also appear in the Z_{1000} field, which is somewhat less than 50 m too low in the tropics. On balance, the overall character of this late-March weather pattern is quite plausible.

A more quantitative assessment of any tendency of our model to distort the atmospheric state by damping or amplifying mid-latitude perturbations at different wavelengths is provided by the plots of the Z_{500} power spectral density around 45 °N in Figure 10. These spectra are averaged over 208 biweekly forecasts from the 2017-2018 test set; as such, the initial spectrum in black represents the average state of the atmosphere in the ERA5 reanalysis.

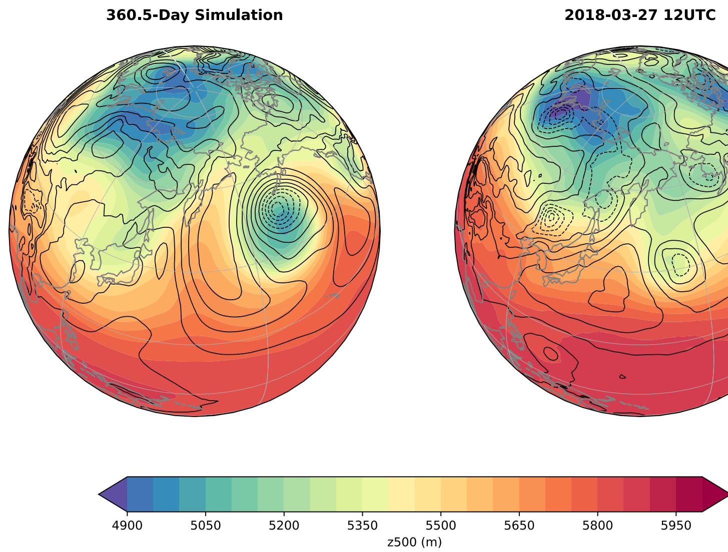

Figure 9: Z_{500} (color fill: 50 dam contour interval) and Z_{1000} (black contours: 40 m interval) for a free-running 360.5-day simulation and the corresponding ERA5 reanalysis for 00 UTC on 27 March 2018. Dashed black lines indicate values of $Z_{1000} \leq 40$ m (corresponding to sea-level pressures less than roughly 1008 hPa).

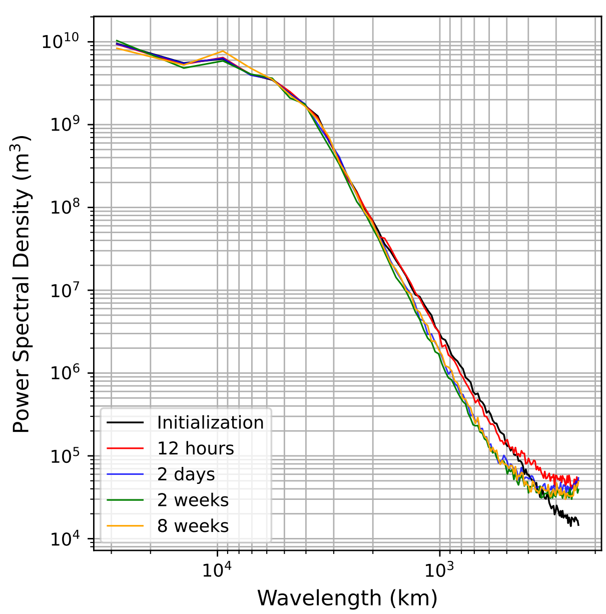

Figure 10: One dimensional power spectral density of the Z_{500} field around the 45° N latitude, averaged over 208 bi-weekly forecasts from 2017-2018 at: initialization (black), and at forecast lead times of 12 h, 2 d, 2, and 8 weeks.

Twelve hours (2 recursive steps) after initialization there is very little change in the spectra for wavelengths λ longer than 500 km (roughly 5 grid intervals), but the power in the shorter waves is amplified. Over the next 36 h, there is a gradual reduction in the amplitude at wavelengths $\lambda < 1800$ km to yield a spectrum that is modestly damped over the interval $380 < \lambda < 1800$ km and amplified at the shortest wavelengths. Surprisingly, the spectral distribution at two days remains essentially unchanged through at least sub-seasonal forecast lead times of eight weeks, which is consistent with the impression obtained examining images such as those in Figure 9. There is no gradual amplification or loss of amplitude in the simulated atmospheric systems after the first two days.

5 Conclusion

We have presented an improved CNN-based DLWP-HPX model that stably forecasts atmospheric evolution over a full one-year cycle using a very limited set of prognostic variables. The number of actual degrees of freedom characterising predictable atmospheric states at forecast lead times beyond 3–5 days is not known, but is far less than the total number of prognostic variables carried at every grid cell in state-of-the-art NWP models. Here, we have demonstrated that realistic atmospheric simulations can be performed using just seven prognostic variables above each node on a HEALPix mesh with 110 km between the nodes.

The HEALPix mesh (Gorski et al., 2005) has been used in astronomy for almost two decades, but has previously seen very little use in atmospheric science. The mesh covers the sphere with a hierarchical grid of equal-area cells uniformly spaced along circles at constant latitudes. A particularly important advantage of the HEALPix mesh for weather forecasting with CNNs is that it is an "east to the right" mesh, i.e., east has the same orientation in every HEALPix cell. Weather systems tend to travel west-to-east in mid- and high-latitudes and both east-to-west (tropical cyclones) or west-to-east (Madden-Julian Oscillation, convectively coupled Kelvin waves) in the tropics. The kernel weights in our convolutional stencils can more economically learn this behavior than on our previous cubed sphere mesh in which the eastward orientation across the stencil varies with longitude, particularly on the polar faces. Although switching from a cube-sphere mesh with 64×64 cells on each of the six faces to a HEALPix mesh with 32×32 cells on each of the 12 faces reduces the total number of grid points covering the sphere by half, it improves the Z_{500} RMSE error by almost one day at a 4-day forecast lead time (Figure 5).

Two other significant improvements to our model architecture were obtained by adding recursion via GRUs and by inverting the standard way channel depth is refined at deeper layers in the U-Net. In contrast to the original U-Net architecture Ronneberger et al. (2015), our channel depth halves instead of doubles as the spatial resolution is also halved in each successively deeper U-Net layer. This allows the model to devote more trainable parameters to describing the wide variety of fine-scale weather patterns while using comparatively fewer parameters to describe the simpler set of global weather patterns. Although this modification pushes the U-Net toward the basic ResNet architecture (He et al., 2016), we find the deeper U-Net layers continue to provide significant skill to the forecasts.

Additional modest improvements were implemented by switching to the GELU activation function and to 2×2 transposed strided convolutions when up-sampling; by increasing the total number of trainable parameters from 2.7 M to 9.8 M, adding the Z_{250} field, increasing the resolution to HPX64, and increasing the time resolution to 3 h (which gives us a 6 h time step). The benefits of 3-h time resolution were only realized when the model included the GRUs. The 3-h time resolution gives a good forecast of the daily cycle of surface temperature, and the model also learns the difference in the range of that cycle between regions of tropical forest and desert without geo-specific input data.

Finally, we replaced the pairs of successive convolutions in Weyn et al. (2020) with modified ConvNeXt blocks. The switch to the ConvNeXt blocks was only advantageous at higher resolutions, where in addition to improving accuracy, it reduced the memory footprint.

At one-week forecast lead time, the resulting model is roughly 1 day behind the ECMWF IFS S2S forecast error in Z_{500} RMSE and 1.5 days behind in ACC. These statistics are worse than those for Pangu-Weather (Bi et al., 2023) and GraphCast (Lam et al., 2022), both of which provide Z_{500} RMSE and ACC forecasts at $0.25^\circ \times 0.25^\circ$ resolution that are superior to the deterministic ECMWF IFS high-resolution model averaged to the same $0.25^\circ \times 0.25^\circ$ grid. Despite having less accuracy in medium range forecasts, our model can be recursively stepped forward to generate better 500 hPa forecasts over seasonal and one-year rollouts than GraphCast and Pangu-Weather. It is also superior to the SFNO version of FourCastNetv2 currently on NVIDIA Earth2MIP, though it behaves similarly to the recently checkpointed version of SFNO Makani. Realistic low pressure systems and upper-level trough and ridge patterns continue to be generated by our model at the end of the one-year rollout.

Deep learning models for weather forecasting are evolving rapidly, with important advancements using a wide variety of architectures. Our DLWP-HPX model provides an example of what can be achieved using a relatively parsimonius approach. As such, it may be particularly useful for scientific investigations where it is advantageous to work with a minimal set of unknown variables to more concisely characterize sensitivities that might be revealed by techniques such as backpropagation with respect loss functions customized for analysis (as opposed to model training).

There are many avenues along which our DLPW-HPX model might be improved. One would be to adding additional prognostic fields while carefully examining the resulting performance. Another one would lie in refining the CNN architecture, where the choice of particular inductive biases may be crucial (Thuemmel et al., 2023). A related important aspect of improving the modelled processes might be to incorporate explicit physical constraints, yielding physics-informed differentiable artificial neural networks (Beucler et al., 2021; Shen et al., 2023). Other natural extensions of this work lie in examining the performance of the DLPW-HPX model in ensemble forecasts, which are crucial to sub-seasonal and seasonal prediction and to couple the atmospheric model with the ocean, thus moving toward a deep learning earth system model (Bauer et al., 2023). Preliminary results suggest that coupling our model with a deep learning ocean model that predicts sea surface temperatures (which are not incorporated in the current model) stabilizes the simulations and removes model drift in multi-decadal rollouts.

Appendix A Deep Learning on the HEALPix

A1 Seamless Evolution of Location Invariant Kernels

The Hierarchical Equal Area isoLatitude Pixelization (HEALPix) is a partitioning of the sphere that has found wide application in astronomy since it was introduced by Gorski et al. (2005). It divides the sphere into 12 base faces that can be hierarchically subdivided into patches of equal size. A key property for training CNNs on this mesh is the isolatitudinal alignment, that is, patches are aligned along lines of latitude and each patch has the same orientation, which we describe as "east to the right" in subsection 4.1.

To contrast and emphasize the difficulty that CNN kernels are facing on the cubed sphere mesh, we plot the lines of constant latitude on the six faces of the cubed sphere and on the twelve faces of the HEALPix in Figure A1. Except for the equator, all lines of constant latitude are bent on the cubed sphere, imposing challenges for a limited set of convolution kernels that must evolve location invariant pattern detectors and functions. For example, on the cubed sphere, kernels need to learn a wider range of behaviors to

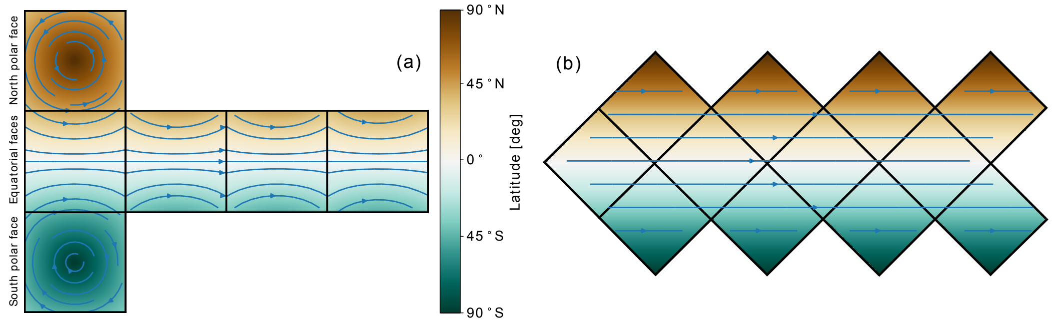

Figure A1: Lines of latitudes depicted as blue streamline arrows on the cubed sphere (a) and on the HEALPix (b). While the lines corresponding to constant eastward motion describe arcs of different radii on the cubed sphere mesh, the same motion translates to straight lines on the HEALPix mesh.

propagate eastward motions at the top-left versus the top right corners of the cubed sphere faces.

On the other hand, lines of constant latitude map to straight lines on the HEALPix mesh. This facilitates the formulation of location-invariant convolutional kernels for the propagation of weather systems, which tend to migrate eastward outside the tropics.

A2 Technical Implementation Details

Since deep learning libraries are optimized for image processing tasks, we consider each of the HEALPix's 12 base faces as a regular two-dimensional tensor, i.e., we interpret the sphere as a composition of twelve images (cf. Figure 1 and Figure A2).

To simulate the spatial propagation of dynamics beyond individual faces, such that weather patterns can evolve globally on the sphere, we implement custom padding operations to concatenate the relevant information of all neighboring faces to each respective face of interest.

Figure A2 showcases our planet's coastlines projected on the HEALPix faces in (a) and outlines the spatial organization of the twelve faces in (b). The arrangement of neighboring faces is exemplarily detailed for the northern (N) and southern (S) hemisphere, as well as for the equatorial faces (E). To simulate the neighborhood of, say, face `E3`, the face `N2` must be concatenated to the left of `E3`, while face `S3` is concatenated to the right. On the northern and southern hemispheres, neighboring faces are partially required to be rotated, as indicated in Figure A2 (c), (d), and (e).

A particular case occurs in the north and south corners of the tropical faces, where no natural neighbor exists—cf. Figure 1 and Figure A2 (f) for an illustration. To simulate the ninth neighbor of the respective corner, we interpolate the values from the according faces on the northern/southern hemisphere, by simply averaging the two corresponding values and writing the result in the simulated neighboring face. For example, to simulate the top left neighboring face of `E3`, we average the respective values from `N2` and `N3`, as detailed by the straight red arrows in Figure A2 (g). Values that do not lie on the main diagonal of the simulated face are not required to be interpolated, but are copied from the adjacent faces instead, denoted by the curved red arrows in Figure A2 (g). The exemplary corner padding shows the case for the application of a 3×3 kernel with dilation of 1 or 2. Note that a 5×5 kernel could be applied in the same way. Importantly, the padding should not extend one neighboring face, which depends on the resolution of the HEALPix mesh and the configuration of the applied convolution (kernel size and dilation). Otherwise, a hierarchy of padding operations would be required to be implemented and considered.

Appendix B DLWP Model Details

Configuration and parameter counts of all layers in our best performing model are detailed in Table B1, where c_{in} denotes the number of input channels, k is the kernel size, s the stride, and d the dilation. The receptive field of each layer with respect to the network input is reported under RF and the output shape takes (F, H, W, C) with F the number of faces, H and W height and width, and C the number of output channels. The dashed line separates the model's encoder (above) and decoder (below) components. All `ConvNeXt`- and `GRU`-blocks are additionally broken down into their operations, visualized by the indented layer names. Numbers in brackets following individual layer names correspond to outputs, which are concatenated to the respective `Concat` layers in the decoder. All convolution layers with $k = 3$ are followed by GELU activation functions. Residual connections are not reported as they neither change the spatial resolution nor the number of channels, and they do not contribute to the parameter count. Color codes approximate those used in the model schematic in Figure 2.

To generate 1-year rollouts for Pangu-Weather, GraphCast, and FourCastNet2 (SFNO), as plotted in Figure 8, we considered the respective public repositories with the pretrained model weights. More concretely, we generated the SFNO Earth2MIP (`fcnv2_sm`) and GraphCast Earth2MIP (`graphcast`) forecasts with NVIDIA's earth2mip package,[8] specifically developing a custom script for long rollouts.[9] The `SFNO Makani` forecast, which responds reasonably to the solar forcing by receiving an additional $\cos\phi$ input (where ϕ is the solar zenith angle), was generated with NVIDIA's Makani package.[10] Interestingly, the original `GraphCast DeepMind` code base[11] produced slightly different results and saturated even faster than the Earth2MIP version, which might result from different random seeds. For the DeepMind version of GraphCast, we downloaded the model weights[12] provided through their repository. Pangu-Weather forecasts in 24 h and 3 h resolution (with respective checkpoint files for the 24 h[13] and 3 h[14] models) were generated by using the original repository.[15]

Open Research Section

Instructions for training, and a trained model for inference, are available at `https://github.com/CognitiveModeling/dlwp-hpx/`. In addition, PyTorch code for training the DLWP-HPX models along with checkpoints of trained models will be provided via NVIDIA's Modulus framework. Accompanying scripts for data preprocessing, including the projection to and from the HEALPix mesh, as well as postprocessing utilities, including evaluation routines, will be made available in the repository at `https://github.com/NVIDIA/modulus/tree/main/examples/weather`. All spherical shells of data from ERA5 (Hersbach et al., 2020) were downloaded from Copernicus, where variables on various con-

[8] `https://github.com/NVIDIA/earth2mip`

[9] `https://github.com/NVIDIA/earth2mip/blob/main/examples/utils/workflows/1_year_run.py`

[10] `https://github.com/NVIDIA/makani`

[11] `https://github.com/google-deepmind/graphcast`

[12] `https://storage.googleapis.com/dm_graphcast/params/GraphCast%20-%20ERA5%201979-2017%20-%20resolution%200.25%20-%20pressure%20levels%2037%20-%20mesh%202to6%20-%20precipitation%20input%20and%20output.npz`

[13] `https://drive.google.com/file/d/1lweQlxcn9fG0zKNW8ne1Khr9ehRTI6HP/view`

[14] `https://drive.google.com/file/d/1EdoLlAXqE9iZLt9Ej9i-JW9LTJ9Jtewt/view`

[15] `https://github.com/198808xc/Pangu-Weather`

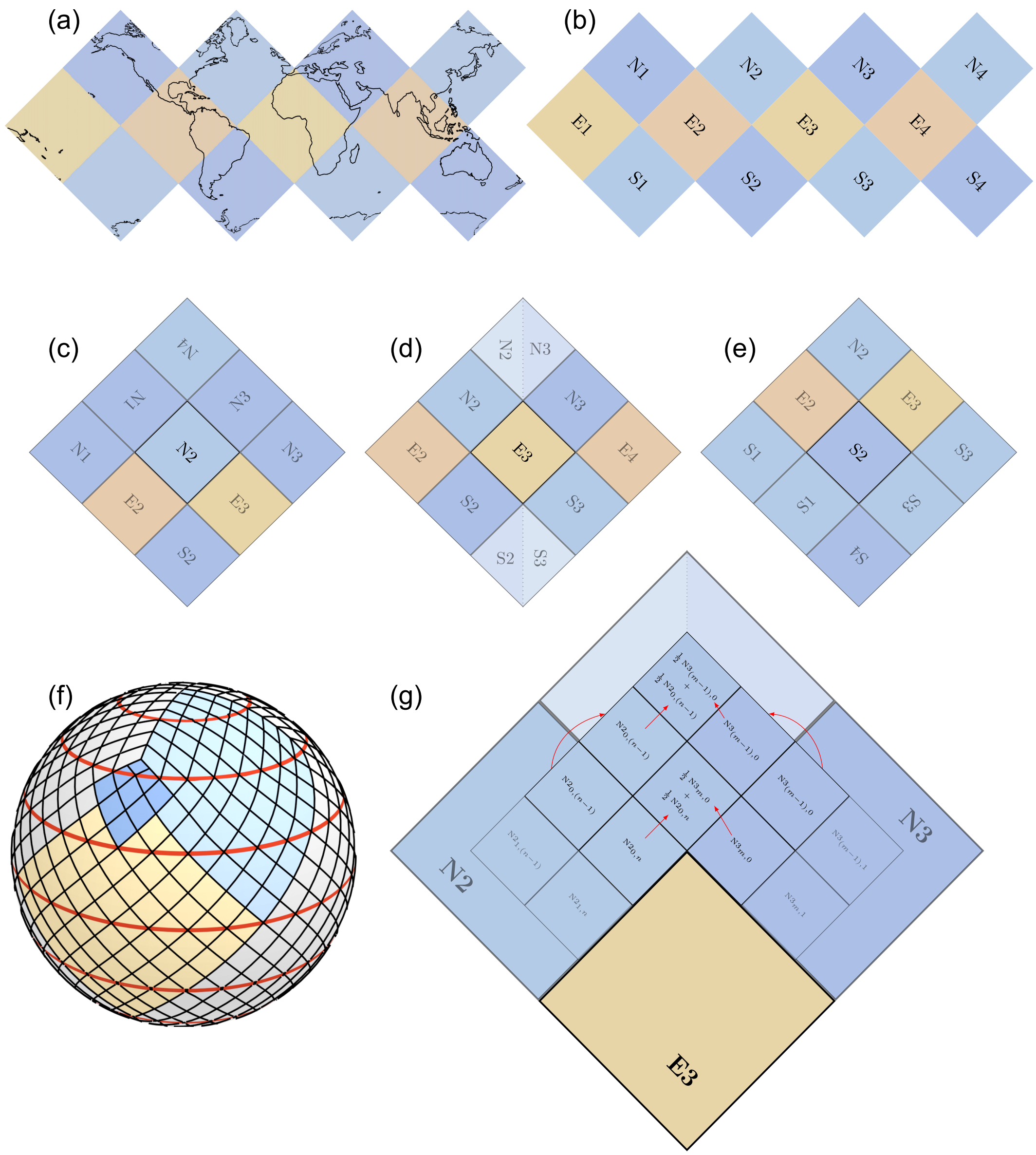

Figure A2: 2D HEALPix face arrangement and padding. (a) depicts the distribution of coastlines over the twelve HEALPix faces. (b) enumerates the twelve faces of the HEALPix with each four faces on the northern and southern hemisphere and around the equator. (c), (d), and (e): Exemplary alignment and rotations of neighboring faces before applying the padding operation on northern (c), equatorial (d), and southern faces (e). (f) emphasizes the special corner case, which is detailed in (g) to visualize the padding, where a ninth pixel is simulated by averaging the two respective values from the adjacent faces.

Table B1: Details of the best performing model. Description of color codes and abbreviations are reported in section Appendix B

							Parameter count		
Layer	c_{in}	k	s	d	RF	Output shape	Weights	Biases	Σ
`ConvNeXt`									
`Conv2d`	18	1	1	1	1×1	(12, 64, 64, 136)	2 448	136	2 584
`Conv2d`	18	3	1	1	3×3	(12, 64, 64, 544)	88 128	544	88 672
`Conv2d`	544	3	1	1	5×5	(12, 64, 64, 544)	2 663 424	544	2 663 968
`Conv2d` (1)	544	1	1	1	5×5	(12, 64, 64, 136)	73 984	136	74 120
`AvgPool2d`	136	2	2	—	6×6	(12, 32, 32, 136)	0	0	0
`ConvNeXt`									
`Conv2d`	136	1	1	1	6×6	(12, 32, 32, 68)	9 248	68	9 316
`Conv2d`	136	3	1	2	14×14	(12, 32, 32, 272)	332 928	272	333 200
`Conv2d`	272	3	1	2	22×22	(12, 32, 32, 272)	665 856	272	666 128
`Conv2d` (2)	272	1	1	1	22×22	(12, 32, 32, 68)	18 496	68	18 564
`AvgPool2d`	68	2	2	—	24×24	(12, 16, 16, 68)	0	0	0
`ConvNeXt`									
`Conv2d`	68	1	1	1	24×24	(12, 16, 16, 34)	2 312	34	2 346
`Conv2d`	68	3	1	4	56×56	(12, 16, 16, 136)	83 232	136	83 368
`Conv2d`	136	3	1	4	88×88	(12, 16, 16, 136)	166 464	136	166 600
`Conv2d`	136	1	1	1	88×88	(12, 16, 16, 34)	4 624	34	4 658
`ConvNeXt`									
`Conv2d`	34	1	1	1	88×88	(12, 16, 16, 68)	2 312	68	2 380
`Conv2d`	34	3	1	4	120×120	(12, 16, 16, 136)	41 616	136	41 752
`Conv2d`	136	3	1	4	152×152	(12, 16, 16, 136)	166 464	136	166 600
`Conv2d`	136	1	1	1	152×152	(12, 16, 16, 68)	9 248	68	9 316
`GRU`									
`Conv2d`	136	1	1	1	152×152	(12, 16, 16, 136)	18 496	136	18 632
`Conv2d`	136	1	1	1	152×152	(12, 16, 16, 68)	9 248	68	9 316
`ConvTrans2d`	68	2	2	1	154×154	(12, 32, 32, 68)	18 496	68	18 476
`Concat` (2)	—	—	—	—	—	(12, 32, 32, 136)	0	0	0
`ConvNeXt`									
`Conv2d`	136	3	1	2	154×154	(12, 32, 32, 272)	332 928	272	333 200
`Conv2d`	272	3	1	2	162×162	(12, 32, 32, 272)	665 856	272	666 128
`Conv2d`	272	1	1	1	170×170	(12, 32, 32, 136)	36 992	136	37 128
`GRU`									
`Conv2d`	272	1	1	1	170×170	(12, 32, 32, 272)	73 984	272	74 256
`Conv2d`	136	1	1	1	170×170	(12, 32, 32, 136)	36 992	136	37 128
`ConvTrans2d`	136	2	2	1	171×171	(12, 64, 64, 136)	73 984	136	74 120
`Concat` (1)	—	—	—	—	—	(12, 64, 64, 272)	0	0	0
`ConvNeXt`									
`Conv2d`	272	1	1	1	171×171	(12, 64, 64, 136)	36 992	136	37 128
`Conv2d`	272	3	1	1	173×173	(12, 64, 64, 544)	1 331 712	544	1 332 256
`Conv2d`	544	3	1	1	175×175	(12, 64, 64, 544)	2 663 424	544	2 663 968
`Conv2d`	544	1	1	1	175×175	(12, 64, 64, 136)	73 984	136	74 120
`GRU`									
`Conv2d`	272	1	1	1	175×175	(12, 64, 64, 272)	73 984	272	74 256
`Conv2d`	272	1	1	1	175×175	(12, 64, 64, 136)	36 992	136	37 128
`Conv2d`	136	1	1	1	175×175	(12, 64, 64, 14)	1 904	14	1 918
							9 816 752	6 066	9 822 818

stant pressure levels, such as Z_{500} or T_{850}, and variables on single levels, such as T_{2m} or $TCWV$, are hosted open to the public, available at `https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=form` and `https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview`.

Acknowledgments

We would like to thank Mauro Bisson from NVIDIA Corp. for providing optimized CUDA kernels for the HEALPix padding implementation, and Jonathan Weyn who previously implemented a code base on which this work was built. We thank Peter Düben and a second anonymous reviewer for encouraging us to generate and compare the 1-year roll-outs for other state-of-the-art DLWP methods and for other valuable suggestions. This work received funding from Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2064 – 390727645 and from the Office of Naval Research under grants N0014-21-1-2827 and N00014-22-1-2807. We thank the Deutscher Akademischer Austauschdienst (DAAD, German Academic Exchange Service) as well as the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Matthias Karlbauer. Nathaniel was supported by a National Defense Science and Engineering Graduate Fellowship. We are grateful to NVIDIA and Stan Posey for the donation of A100 GPU cards. This research was additionally supported by a grant from the NVIDIA Applied Research Accelerator Program and utilized an NVIDIA DGX-100 Workstation. Moreover, this work benefited substantially from the barrier-free high quality ERA5 dataset provided by the ECMWF.

Author Roles

Matthias implemented model, training and evaluation routines in `PyTorch`, as well as the HEALPix-related projection scripts under consideration of the `healpy` package, and drafted the manuscript together with Dale who supervised this project closely and who also made the model schematic in Figure 2. Nathaniel was involved in discussions about model evolution and code structures and generated Figure 6, Figure 7, and Figure 10. Raul was involved in model discussions and generated Figure 9. Thorsten helped with implementing the distributed PyTorch pipeline for multi-GPU training and with accelerating the process pipeline. Noah Brenowitz and Boris Bonev generated the 365-days rollouts with the Earth2MIP and Makani packages for SFNO and GraphCast. Martin co-supervised this project and helped with proofreading and writing.

References

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... others (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Bauer, P., Dueben, P., Chantry, M., Doblas-Reyes, F., Hoefler, T., McGovern, A., & Stevens, B. (2023). Deep learning and a changing economy in weather and climate prediction. *Nature Reviews Earth & Environment*, *4*(8), 507–509. Retrieved from `https://doi.org/10.1038/s43017-023-00468-z` doi: 10.1038/s43017-023-00468-z

Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*(7567), 47–55.

Benjamin, S. G., Brown, J. M., Brunet, G., Lynch, P., Saito, K., & Schlatter, T. W. (2019). 100 years of progress in forecasting and nwp applications. *Meteorological Monographs*, *59*, 13–1.

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Physical*

Review Letters, *126*(9), 098302.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2023). Accurate medium-range global weather forecasting with 3d neural networks. *Nature*. doi: doi.org/10.1038/s41586-023-06185-3

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., & Anandkumar, A. (2023). Spherical fourier neural operators: Learning stable dynamics on the sphere. *arXiv preprint arXiv:2306.03838*.

Charney, J. G., Fjörtoft, R., & Neumann, J. V. (1950). Numerical Integration of the Barotropic Vorticity Equation. *Tellus A*, *2*(4).

Chen, K., Han, T., Gong, J., Bai, L., Ling, F., Luo, J.-J., ... Ouyang, W. (2023). Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on empirical methods in natural language processing (emnlp 2014).*

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... others (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, *11*(10), 3999–4009.

Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 ieee international joint conference on neural networks, 2005.* (Vol. 2, pp. 729–734).

Gorski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. (2005). Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, *622*(2), 759.

Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A., & Catanzaro, B. (2021). Efficient token mixing for transformers via adaptive fourier neural operators. In *International conference on learning representations.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... others (2020). The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hu, Y., Chen, L., Wang, Z., & Li, H. (2022). Swinvrnn: A data-driven ensemble forecasting model via learned distribution perturbation. *arXiv preprint arXiv:2205.13158*.

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., ... Wu, J. (2020). Unet 3+: A full-scale connected unet for medical image segmentation. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 1055–1059).

Keisler, R. (2022). Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krachmalnicoff, N., & Tomasi, M. (2019). Convolutional neural networks on the healpix sphere: a pixel-based algorithm and its application to cmb data analysis. *Astronomy & Astrophysics*, *628*, A129.
Kurth, T., Subramanian, S., Harrington, P., Pathak, J., Mardani, M., Hall, D., ... Anandkumar, A. (2022). Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. *arXiv preprint arXiv:2208.05419*.
Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Pritzel, A., ... others (2022). Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*.
Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.
Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 10012–10022).
Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 11976–11986).
Lopez-Gomez, I., McGovern, A., Agrawal, S., & Hickey, J. (2022). Global extreme heat forecasting using neural weather models. *arXiv preprint arXiv:2205.10972*.
Lorenz, E. N. (1969). The predictability of a flow which possesses many scales of motion. *Tellus*, *21*(3), 289–307.
Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. In *International conference on learning representations.*
Palmer, T. (2019). The ecmwf ensemble prediction system: Looking back (more than) 25 years and projecting forward 25 years. *Quarterly Journal of the Royal Meteorological Society*, *145*, 12–24.
Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., ... others (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*.
Perraudin, N., Defferrard, M., Kacprzak, T., & Sgier, R. (2019). Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. *Astronomy and Computing*, *27*, 130–146.
Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., & Battaglia, P. W. (2020). Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*.
Rasp, S., & Thuerey, N. (2021). Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for weatherbench. *Journal of Advances in Modeling Earth Systems*, *13*(2), e2020MS002405.
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241).
Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, *20*(1), 61–80.
Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, *144*(717), 2830–2841.
Scher, S., & Messori, G. (2019). Weather and climate forecasting with neural networks: using GCMs with different complexity as study-ground. *Geoscientific Model Development*, *12*, 2797–2809.
Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., ... Lawson, K. (2023). Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, *4*(8),

552–567. Retrieved from https://doi.org/10.1038/s43017-023-00450-9 doi: 10.1038/s43017-023-00450-9

Thuemmel, J., Karlbauer, M., Otte, S., Zarfl, C., Martius, G., Ludwig, N., … others (2023). Inductive biases in deep learning models for weather prediction. *arXiv preprint arXiv:2304.04664*.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, *46*(sup1), 234–240.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vitart, F. (2004). Monthly forecasting at ECMWF. *Monthly Weather Review*, *132*, 2761–2779. doi: 10.1175/MWR2826.1

Weigel, A. P., Baggenstos, D., Liniger, M. A., Vitart, F., & Appenzeller, C. (2008). Probabilistic Verification of Monthly Temperature Forecasts. *Monthly Weather Review*, *136*, 5162–5182. doi: 10.1175/2008MWR2551.1

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

Weyn, J. A., Durran, D. R., & Caruana, R. (2020). Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, *12*(9), e2020MS002109.

Weyn, J. A., Durran, D. R., Caruana, R., & Cresswell-Clay, N. (2021). Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002502.

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 3–11). Springer.