AURORA: A FOUNDATION MODEL OF THE ATMOSPHERE

Cristian Bodnar[*, 1], **Wessel P. Bruinsma**[*, 1], **Ana Lucic**[*, 1], **Megan Stanley**[*, 1], **Johannes Brandstetter**[3, †], **Patrick Garvan**[1], **Maik Riechert**[1], **Jonathan Weyn**[2], **Haiyu Dong**[2], **Anna Vaughan**[4], **Jayesh K. Gupta**[5, †], **Kit Tambiratnam**[2], **Alex Archibald**[4], **Elizabeth Heider**[1], **Max Welling**[6, †], **Richard E. Turner**[1, 4], **and Paris Perdikaris**[1]

[1]Microsoft Research AI for Science

[2]Microsoft Corporation [3]JKU Linz [4]University of Cambridge [5]Poly Corporation [6]University of Amsterdam

[*]Equal contribution [†]Work done while at Microsoft Research

ABSTRACT

Deep learning foundation models are revolutionizing many facets of science by leveraging vast amounts of data to learn general-purpose representations that can be adapted to tackle diverse downstream tasks. Foundation models hold the promise to also transform our ability to model our planet and its subsystems by exploiting the vast expanse of Earth system data. Here we introduce Aurora, a large-scale foundation model of the atmosphere trained on over a million hours of diverse weather and climate data. Aurora leverages the strengths of the foundation modelling approach to produce operational forecasts for a wide variety of atmospheric prediction problems, including those with limited training data, heterogeneous variables, and extreme events. In under a minute, Aurora produces 5-day global air pollution predictions and 10-day high-resolution weather forecasts that outperform state-of-the-art classical simulation tools and the best specialized deep learning models. Taken together, these results indicate that foundation models can transform environmental forecasting.

1 Introduction

Deep learning foundation models have revolutionised various scientific domains, such as protein structure prediction (Abramson et al., 2024), drug discovery (Chithrananda et al., 2020), computer vision (Betker et al., 2023), and natural language processing (OpenAI, 2024). The key tenets of foundation models include *pretraining*, where a single large-scale neural network learns to capture intricate patterns and structure from a large corpus of diverse data; and *fine-tuning*, which allows the model to leverage its learned representations to excel at new tasks with limited training data (et al., 2021; Brown et al., 2020).

The Earth system is a complex and interconnected network of subsystems, such as the atmosphere, oceans, land, and ice, which constantly interact in intricate ways. In a rapidly changing climate, accurate understanding of these subsystems becomes increasingly important. We envision that foundation models can revolutionise our ability to model subsystems of the Earth, and eventually the whole Earth.

Amongst the Earth's subsystems, the atmosphere stands out as particularly data-rich (Reichstein et al., 2019; Bauer et al., 2015) and therefore constitutes ripe ground for pretraining a foundation model. Classical atmospheric simulation approaches, such as numerical weather prediction (NWP), are costly and unable to exploit this wealth of data (Bauer et al., 2015). Recent deep learning approaches are cheaper, more flexible, and have shown great promise in specific prediction tasks with abundant training data (Lam et al., 2023a; Bi et al., 2023; Chen et al., 2023a,b; Han et al., 2024; Kochkov et al., 2024; Lessig et al., 2023; Pathak et al., 2022; Bonev et al., 2023; Andrychowicz et al., 2023; Ham et al., 2019; Nguyen et al., 2023a,b). However, these methods struggle when atmospheric training data are scarce (Chantry et al., 2021) or heterogeneous (Reichstein et al., 2019), and they lack robustness in predicting extremes (Charlton-Perez et al., 2024). By learning generalizable representations from vast amounts of diverse data, foundation models have been able to overcome analogous challenges in other domains (Zhai et al., 2022; Radford et al., 2021; et al., 2021; Nguyen et al., 2023a).

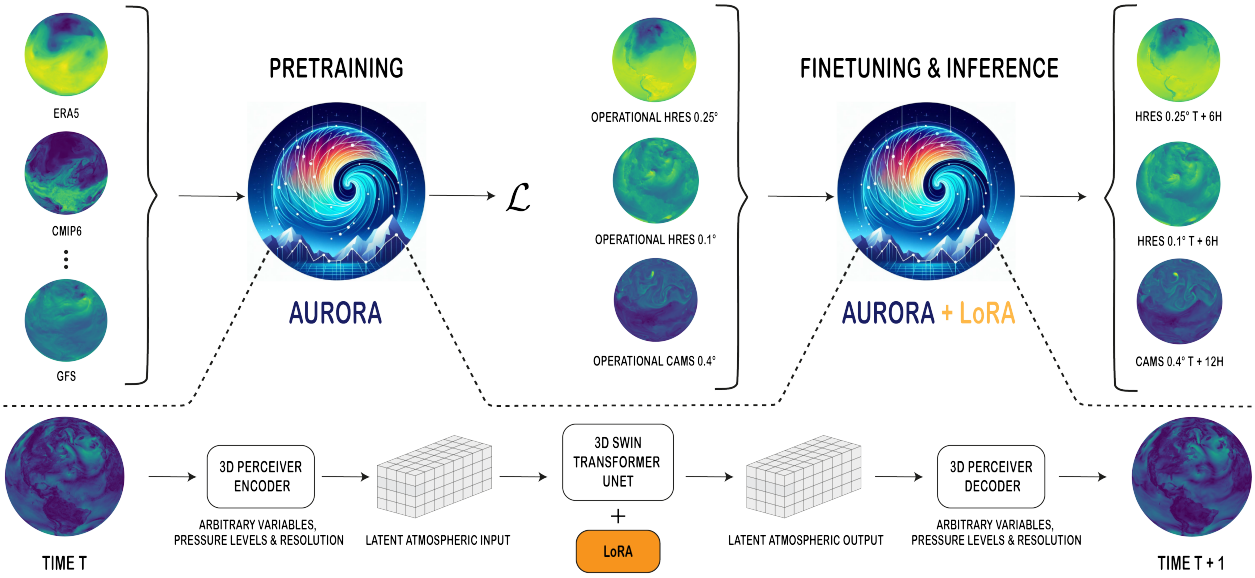

Figure 1: **Aurora is a 1.3 billion parameter foundation model for high-resolution forecasting of weather and atmospheric processes.** Aurora is a flexible 3D Swin Transformer with 3D Perceiver-based encoders and decoders. At pretraining time, Aurora is optimised to minimise a loss $\mathcal{L}$ on multiple heterogeneous datasets with different resolutions, variables, and pressure levels. The model is then fine-tuned in two stages: (1) short-lead time fine-tuning of the pretrained weights (2) long-lead time (rollout) fine-tuning using Low Rank Adaptation (LoRA). The fine-tuned models are then deployed to tackle a diverse collection of operational forecasting scenarios at different resolutions.

Here we introduce *Aurora*: a foundation model of the atmosphere. Aurora can produce forecasts for a wide variety of atmospheric forecasting problems, including those with limited training data, heterogeneous variables, and extreme events. We demonstrate this ability by producing operational forecasts for global air pollution and global high-resolution medium-term weather patterns that match or outperform state-of-the-art classical simulation tools, at orders of magnitude smaller computational cost. Specifically, in under a minute, Aurora generates 5-day air pollution forecasts at $0.4°$ spatial resolution surpassing state-of-the-art resource-intensive atmospheric chemistry simulations on 74% of all targets; 10-day global weather forecasts at $0.1°$ resolution surpassing the state-of-the-art NWP model on 92% of all targets, significantly improving performance in extreme events compared to other deep learning approaches; and 10-day global weather forecasts at $0.25°$ surpassing state-of-the-art NWP and GraphCast (Lam et al., 2023a) on over 91% of all targets.

Aurora is able to efficiently adapt to new atmospheric prediction tasks by learning a general-purpose representation of atmospheric dynamics. This representation is learned by scaling Aurora to $1.3\,\mathrm{B}$ parameters and pretraining on over a million hours of diverse atmospheric data, mostly consisting of simulations generated by weather and climate models (Reichstein et al., 2019; Bauer et al., 2015). For the first time, we demonstrate that training a large model on more diverse datasets improves atmospheric forecasting compared to training on single dataset. This major finding supports our foundation model approach and represents a departure from the single-dataset training followed by most of the aforementioned data-driven approaches.

Aurora serves as a blueprint for foundation models for other Earth subsystems, and is an advancement towards building a single comprehensive foundation model capable of tackling a wide range of environmental prediction tasks.

2 Aurora: A flexible 3D foundation model of the atmosphere

Aurora can ingest and make predictions for any collection of surface-level and meteorological variables, at any pressure levels, resolution, and level of fidelity. The model consists of three parts: (1) an encoder, whose role is to convert heterogeneous inputs into a standard 3D representation of the atmosphere across space and pressure-levels; (2) a processor which evolves the representation in time; and (3) a decoder which translates the standard 3D representation back into specific predictions. The processor is a Vision Transformer (Dosovitskiy et al., 2021; Liu et al., 2021) and the encoder and decoder are Perceiver-based modules (Jaegle et al., 2021, 2022) (Figure 1). Forecasts at different lead times can be generated by iteratively feeding predictions back into the model as inputs. We defer a more detailed discussion of the model to Supplementary B.

We pretrain Aurora on a vast body of weather and climate data. By including as much and as diverse data as possible, Aurora attempts to learn a general-purpose representation of atmospheric dynamics. The pretraining data is a mixture of six weather and climate datasets: ERA5, CMCC, IFS-HR, HRES Forecasts, GFS Analysis and GFS Forecasts. These datasets are a mixture of forecasts, analysis data, reanalysis data, and climate simulations. The pretraining datasets comprise a standard collection of meteorological variables (see Supplementary C for more details on the datasets). During pretraining, we minimise the next–time step mean absolute error (MAE) with a 6-hour lead time for $150\,\mathrm{k}$ steps on 32 A100 GPUs, which corresponds to approximately two and a half weeks of training. Once pretraining is completed, Aurora is ready to be fine-tuned for new atmospheric prediction tasks. Fine-tuning takes place in multiple stages, ending with a roll-out fine-tuning stage (Supplementary D).

For lead times up to 15 days, the Integrated Forecasting System (IFS; European Centre for Medium-Range Weather Forecasts, 2023a) is the gold-standard and state-of-the-art numerical forecasting system. This system, however, operates at considerable computational cost: producing a 10-day forecast takes approximately 65 minutes on 352 high-end CPU nodes with 36 cores each (see section 2.1.5 in Buizza et al., 2018), corresponding to approximately 5720 node-seconds per hour lead time. In contrast, Aurora can make predictions at approximately $1.1\,\mathrm{s}$ per hour lead time on a single A100 GPU, thus yielding roughly a $\times 5{,}000$ speedup over IFS.

3 Fast prediction of atmospheric chemistry and air pollution

We selected forecasting atmospheric chemistry and air pollution as our first application of Aurora due to its societal importance, the significant challenge it poses to data driven approaches, and the potential these methods hold for acceleration and improving accuracy.

Air quality is a key factor in non-communicable disease and therefore the health of humans, and is determined by concentrations of various gasses and aerosols in the atmosphere (World Health Organization, 2021). Accurately predicting global atmospheric composition (the distribution of trace gases and aerosols in the air) can aid mitigation of air pollution event.

Forecasting atmospheric composition is much more complex and costly than weather forecasting. Not only does atmospheric composition depend on transport via weather systems, atmospheric chemistry models also simulate nonlinear reactions between different chemical species through hundreds of stiff equations (Brasseur and Jacob, 2017). In addition, atmospheric composition strongly depends on anthropogenic emissions of trace gases and aerosols, which drive the heterogeneous levels of pollution seen across the globe. Predicting global atmospheric composition therefore requires a model to understand complex atmospheric chemistry and account for human behaviour.

The Copernicus Atmosphere Monitoring Service (CAMS) is an operational system that produces forecasts, analysis products, and reanalysis products for global atmospheric composition (European Centre for Medium-Range Weather Forecasts, 2023b) at 0.4° resolution. CAMS is an extension of IFS with additional modules for aerosols, reactive gases, and greenhouse gases. For the reasons mentioned above, these extension modules make the already computationally expensive IFS significantly more costly (by a factor of $10\times$, based on private communication with ECMWF). Although machine learning methods could amortise these computational costs and even improve accuracy, thus far no AI method has attempted to produce operational predictions for global atmospheric composition. In this experiment, we demonstrate that Aurora can be successfully fine-tuned to CAMS analysis data to produce operational forecasts that match or outperform CAMS forecasts in terms of RMSE on 74% of all targets, at orders of magnitude lower computational cost.

Fine-tuning Aurora (or training any AI model) on CAMS analysis data is extremely challenging for three reasons. First, unlike NWP systems (*e.g.*, IFS) which are relatively stable across update cycles, CAMS undergoes frequent updates that significantly affect the data distribution. In addition, CAMS analysis data only goes back to 2015, and quality of the data generally decreases the further you go back. This means that CAMS analysis data is very scarce and non-stationary. Second, unlike meteorological variables, air pollution variables are concentration values with a large dynamic range. These variables are highly heterogeneous (Figure 2a), and often extremely sparse and skewed. Finally, air pollution variables are strongly dependent on anthropogenic factors such as factory emissions. The global response to the COVID pandemic contributes to complex distribution shifts in pollutant data.

Experimental setup. Six air pollutants are the main drivers of poor air quality (World Health Organization, 2021): carbon monoxide (CO), nitrogen oxide (NO), nitrogen dioxide (NO_2), sulphur dioxide (SO_2), ozone (O_3), and particulate matter at $1\,\mu\mathrm{m}$ (PM_1), $2.5\,\mu\mathrm{m}$ ($PM_{2.5}$), and $10\,\mu\mathrm{m}$ (PM_{10}). Air quality warnings are usually based on threshold values for $PM_{2.5}$ and/or PM_{10}. Aurora models the five chemical species (CO, NO, NO_2, SO_2, and O_3) both as 3D atmospheric variables and as 2D surface-level variables as their total column values, and models the PMs as surface-level variables.

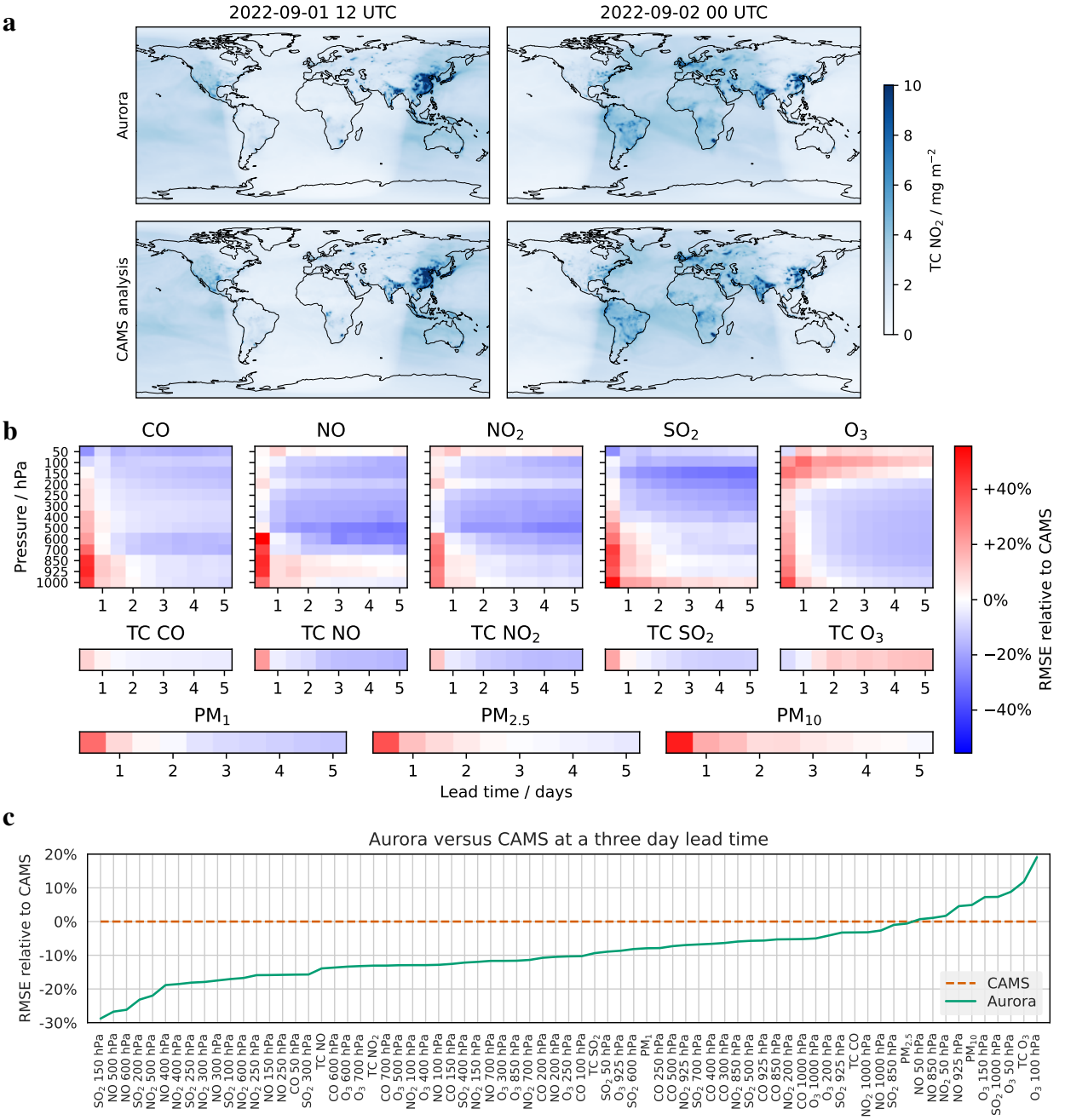

Figure 2: **Aurora outperforms operational CAMS across many targets. a**: Sample predictions for total column nitrogen dioxide (TC NO_2) by Aurora compared to CAMS analysis. Aurora was initialised with CAMS analysis at 1 Sep 2022 00 UTC. Predicting atmospheric gasses correctly is extremely challenging due to their spatially heterogeneous nature. In particular, NO_2, like most variables in CAMS, is skewed towards high values in areas with large anthropogenic emissions such as densely populated areas in East Asia. In addition, NO_2 exhibits a strong diurnal cycle; *e.g.*, sunlight reduces background levels of NO_2 through a process called photolysis. Aurora accurately captures both the extremes and background levels. **b**: Latitude-weighted root mean square error (RMSE) of Aurora relative to CAMS, where negative values (blue) mean that Aurora is better. The RMSEs are computed over the period Jun 2022 to Nov 2022 inclusive. Aurora matches or outperforms CAMS on 74% of the targets. See Supplementary G.7 for full results. **c**: Latitude-weighted RMSE of Aurora relative to CAMS at a three-day lead time, where negative values mean that Aurora is better. The RMSEs are computed over the period Jun 2022 to Nov 2022 inclusive. Aurora matches or outperforms CAMS on 86% of all variables. See Supplementary G.7 for full results. The full results show that, at a three day lead time, Aurora is also significantly better than the persistence prediction for all variables.

CAMS uses spatially and temporally disaggregated emissions data as inputs. These inputs allow natural factors (wildfires, vegetation, *etc.*) and anthropogenic factors (vehicle combustion, energy production, *etc.*) to influence the levels of the air pollutants simulated by CAMS. Aurora does not use any emissions data as inputs, except for some additional static variables which are fixed across all times and experiments (Supplementary B.7). Instead, by learning from historical data which is affected by natural and anthropogenic factors, Aurora implicitly learns to account for these effects.

We fine-tune Aurora on CAMS analysis data from Oct 2017 to May 2022 and test on CAMS analysis data from May 2022 to Nov 2022 (Supplementary C.3). Since CAMS analysis data is very limited in temporal extent, to better learn the dynamics of the pollutants, we also incorporate lower resolution and CAMS reanalysis data (EAC4; Inness et al., 2019) from Jan 2003 to Dec 2021 in the fine-tuning process. CAMS reanalysis data is considered to be lower quality because it uses a lower resolution and a significantly older version of IFS (Supplementary C).

Results. Aurora is competitive with CAMS (within 20% RMSE) on 95% of all targets, and matches or outperforms CAMS on 74% of all targets (Figure 2b). At the three day mark, Aurora is competitive with CAMS (within 20% RMSE) on all variables, and matches or outperforms CAMS on 86% of all variables (Figure 2c). Aurora is outperformed by CAMS on ozone in the very upper atmosphere and the twelve-hour prediction of all species in the lower part of the atmosphere. Note that variables near the surface and in the lower part of the atmosphere are primarily affected by anthropogenic factors, which Aurora does not explicitly account for. For these variables, assimilation by CAMS is also relatively weak. For example, the PMs are only weakly constrained via aerosol optic depth satellite measurements; CAMS does not assimilate any real-time emission observations. Weak assimilation means that CAMS analysis is more like CAMS forecasts than is the case for meteorological variables, especially at the 12 hour mark. Since we take CAMS analysis to be the ground truth, this may disadvantage Aurora in this benchmark.

In a case study, we evaluate Aurora's predictions for PM_{10} for 13 June 2023, when Iraq was hit by a particularly bad sandstorm (Figure 27 in Supplementary G.7). Aurora is able to predict the sandstorm a day in advance. This is one in a series of devastating sandstorm events that took place in the late Spring of 2022 leading to more than $5{,}000$ hospitalizations in the Middle East (Francis et al., 2023).

4 Skillful operational weather forecasting at 0.1° resolution

We selected weather forecasting at 0.1° resolution as our second application of Aurora because of its operational importance, and the significant demands it places on data driven approaches in terms of the criticality of performance on extreme events and data-efficiency.

To accurately resolve high-impact weather events such as severe storms, hurricanes, and heatwaves, it is essential that weather prediction systems operate at a high resolution to better resolve, among other things, convective effects at smaller scales of motion. The Integrated Forecasting System (IFS-HRES; Malardel et al., 2016), the gold-standard and state-of-the-art numerical medium-term weather forecasting system, operates at 0.1° (approximately $11\,\mathrm{km}$ at the equator), at considerable computational cost. Current state-of-the-art artificial intelligence weather prediction (AIWP) models (Lam et al., 2023a; Bi et al., 2023; Pathak et al., 2022; Chen et al., 2023a; Bonev et al., 2023) are designed to process and predict global weather states at 0.25° resolution, corresponding to approximately $28\,\mathrm{km}$ at the equator and a $2.5\times$ decrease in resolution compared to IFS-HRES. Adapting AIWP methods to 0.1° resolution faces a series of major hurdles (Chen et al., 2023b). The most salient issue is that high-resolution training data is scarce only going back to 2016, when IFS was upgraded to 0.1° (Malardel et al., 2016). Here we demonstrate that Aurora can efficiently adapt to this data-scarce setting and surpass the forecasting skill of IFS-HRES under operational evaluation protocols.

Experimental setup. In order to unlock high-resolution prediction capabilities for AIWP, a series of substantial technical challenges must be overcome. First, the sheer size of 0.1° data ($1.78\,\mathrm{GB}$ per datapoint) presents a stress test for all the components of a training pipeline, from storage and network speeds, to data-loading performance, to model evaluation and memory management (see Supplementary D for details). And although the size of each individual datapoint at 0.1° resolution is much larger, the amount of available training datapoints is substantially smaller. While 0.25° data date back to 1950, including both analysis, reanalysis and forecast products, global analysis data at 0.1° resolution are only available from 2016 onwards and a reanalysis product does not yet exist. Aurora is well-equipped to operate in this setting, as it can be pretrained on 0.25° data and then finetuned on a smaller corpus of 0.1° data to unlock high-resolution forecasting capabilities. Here we demonstrate this by finetuning a pretrained 6-hour Aurora model on 2016-2022 IFS-HRES analysis data. To cope with the increase in resolution, we perform some minor modifications to the model after pretraining (see Supplementary B and Supplementary D for details). For evaluation, we follow the protocol in Ben-Bouallegue et al. (2024) initialising Aurora with IFS analysis and evaluating forecasts

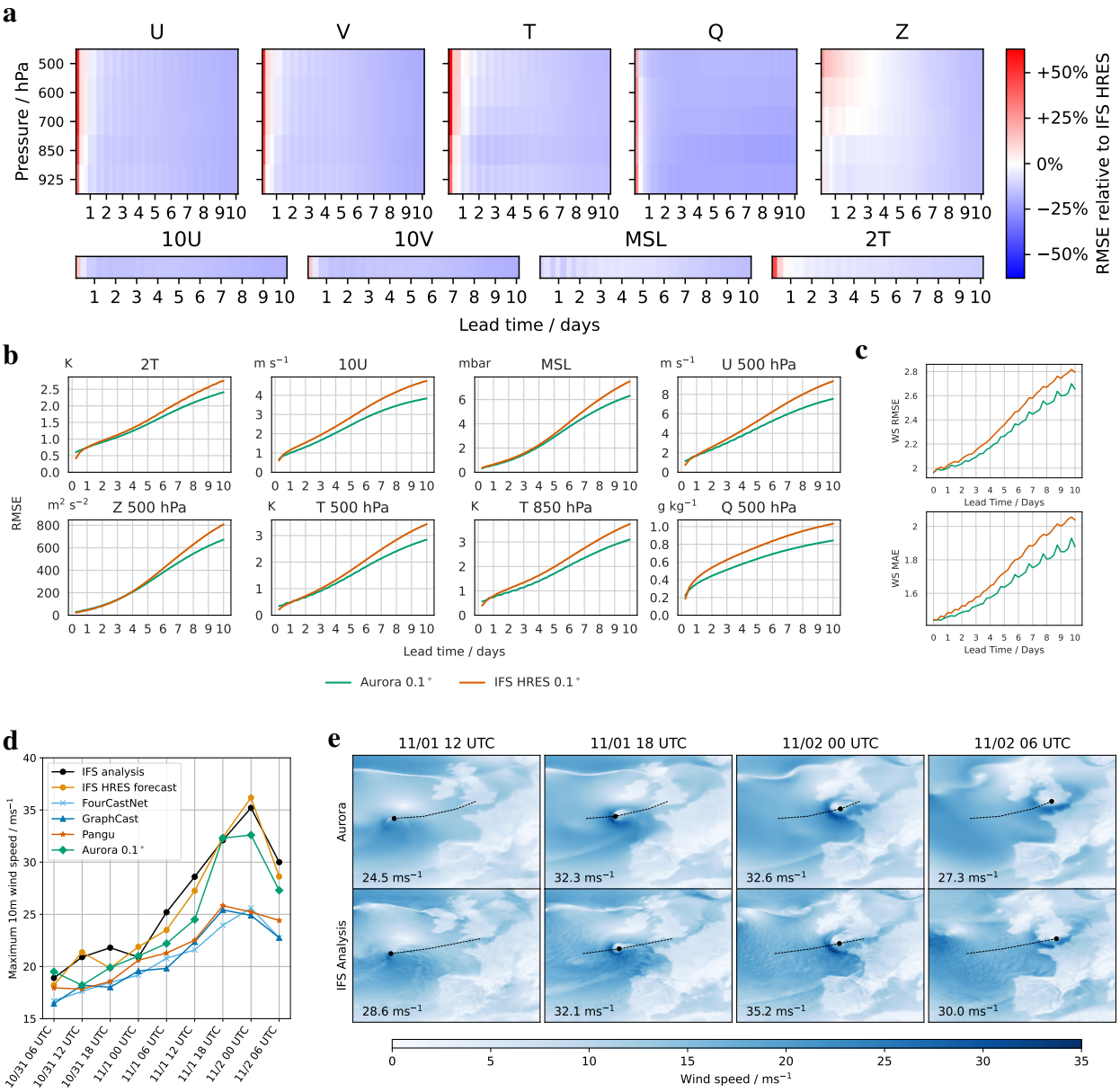

Figure 3: **Aurora outperforms operational HRES at 0.1° across the vast majority of targets and is the only AI model to accurately estimate the maximum $10\,\mathrm{m}$ wind speed in storm Ciarán.** **a**: Scorecard vs HRES at 0.1°. Aurora outperforms HRES on over 92% of targets. The scorecard is limited to pressure levels lower in the atmosphere due to restricted availability of the test year data. **b**: Results for headline metrics at 0.1°. Aurora outperforms NWP across all variables. **c**: Wind speed RMSE and MAE compared to ground truth at weather stations, for HRES at $0.1^\circ$and Aurora 0.1°. Aurora significantly outperforms HRES. **d**: Operational predictions for maximum $10\,\mathrm{m}$ wind speed during Storm Ciarán by Aurora, FourCastNet, GraphCast, and Pangu-Weather. The numbers for all AI models except Aurora have been extracted from Figure 3 by Charlton-Perez et al. (2024). Aurora is able to predict the sudden increase in $10\,\mathrm{m}$ wind speed, unlike the other AI models. **e**: Operational predictions for Storm Ciarán compared to IFS analysis at 0.1° resolution. Black dots show the location of minimum MSL and therefore trace the path of the storm. The maximum $10\,\mathrm{m}$ wind speed of the storm is shown in the bottom left corner of each prediction, and is computed without LoRA, according to a bounding box tracking Ciarán (Supplementary G.6).

against IFS analysis, which mimics an operational setting. To compensate for the fact that IFS uses a slightly different analysis product internally, called HRES-T0, we follow (Lam et al., 2023a) and evaluate IFS against HRES-T0.

Results. Aurora has lower RMSE than HRES across the vast majority of target variables, pressure levels and lead times (Figure 3b). The performance gains are most pronounced at lead times more than 12 hours into the future, where we observe a reduction in RMSE up to 60% at long lead times. The biggest gains are obtained for variables such as temperature and wind velocity components, whereas the smallest gains are observed for geopotential height, where HRES tends to yield more accurate predictions at the higher atmospheric levels. At long lead times Aurora's forecasts resemble the IFS ensemble mean (see Supplementary G.8), a common trait of AIWP. In contrast, HRES yields consistently better performance for all variables at very short lead times, up to 12 hours. Overall, this is the first time that an AIWP model is shown to consistently outperform HRES in terms of RMSE skill at 0.1° resolution when evaluated using operational initial conditions.

This use case illustrates the benefits of a foundation modelling approach which can operate across multiple resolutions at pretraining and fine-tuning stages, and handle downstream tasks with small amounts of data.

Verification against station observations. Most verification efforts for AIWP models have focused on gridded analysis data such as ERA5 (*c.f.* WeatherBench 2; Rasp et al., 2024). Not only are weather station observations a more accurate representation of weather experienced by people, but they can also reveal deficiencies in models relying on diagnostic parameters generated by the NWP models producing the analysis. Therefore, we also analyse the performance of Aurora at predicting raw observations. The observation dataset includes over 13 thousand weather observing stations globally from the Integrated Surface Database (ISD) (Smith et al., 2011). We apply some basic quality control checks including inconsistent variable checks and range checks. This evaluation includes forecasts issued at 00 UTC between 2 January 2023 and 19 December 2023.

As shown in Figure 3c, Aurora outperforms IFS-HRES for $10\,\mathrm{m}$ wind speed forecasts across all lead times up to 10 days. Further results for $2\,\mathrm{m}$ temperature are available in Supplementary F.2. This result, for the first time, shows how a global high-resolution AIWP model can set a new state-of-the-art as measured by real weather observations.

Case study at 0.1°: Storm Ciarán. Storm Ciarán was a high-impact windstorm which took place across North-West Europe in late 2023. It is considered to be a meteorological outlier event because it was the lowest pressure ever recorded in November in England (UKMO). The point of lowest pressure corresponds to the center of the storm.

Charlton-Perez et al. (2024) evaluate the performance of various AI models on predicting Storm Ciarán, in terms of (i) minimum mean sea level pressure (MSL) and (ii) maximum wind speed $10\,\mathrm{m}$ from the surface of the Earth. They find that, although the AI models are able to accurately forecast the minimum MSL, a common shortcoming is that none of them is able to capture the spike in maximum $10\,\mathrm{m}$ wind speed that occurs on 00 UTC 2 November 2023. In Figure 3d, we reproduce the findings by Charlton-Perez et al. (2024) and include a version of Aurora which is suitable for predicting wind speed at 0.1°resolution (see See Supplementary G.6). All models are initialized on 31 Oct 00 UTC using the latest cycle of IFS analysis, CY48R1. We observe that among the AI models tested, Aurora stands out as the only one capable of accurately predicting the abrupt rise in maximum $10\,\mathrm{m}$ wind speed. It forecasts approximately $7\ \mathrm{ms}^{-1}$ higher than the other AI techniques, closely matching IFS analysis, which we take to be the ground truth. Notably, FourCastNet (Pathak et al., 2022), GraphCast (Lam et al., 2023a), and Pangu-Weather (Bi et al., 2023) are constrained to a 0.25° resolution, potentially hindering their ability to capture rapid onset phenomena. However, even at 0.25° resolution, Aurora demonstrates some capability in capturing this sharp increase, albeit to a lesser degree than its 0.1° counterpart; see Figure 23 in Supplementary G.6.

Figure 3e shows Storm Ciarán's intensification over a $24\,\mathrm{hour}$ period, where we compare Aurora forecasts (top) to IFS analysis (bottom) visually in terms of $10\,\mathrm{m}$ wind speed. We also plot the minimum MSL, corresponding to the center of the storm, and find that the Aurora's forecasts are similar to IFS analysis. The primary difference lies in the level of detail: IFS analysis captures higher spatial frequency content, whereas Aurora's forecasts are smoother, although less so than those from the 0.25°model (see Supplementary G.6 for details and Supplementary G.8 for a power-spectrum analysis). Despite this disparity, the forecasts are visually similar.

5 Data diversity and model scaling improve atmospheric forecasting

A key tenet of the foundation model paradigm is that performance improves in a significant and predictable fashion as the data and the model are scaled (Hoffmann et al., 2024; Kaplan et al., 2020). For the first time, we provide evidence that these findings from the fields of natural language processing and computer vision also hold for weather forecasting.

Data scaling. Most AI-based medium range weather forecasting approaches have employed a single dataset in their training strategy, often consisting of ERA5 at 0.25°resolution. Training Aurora on multiple datasets allows us, for the first time, to examine the benefits of employing a selection of diverse datasets in pretraining. To that end, we pretrain the main Aurora model on four different dataset configurations (labeled C1-C4) and verify: (1) the RMSE of the models across different variables and levels on ERA5 2021 (Figure 4a), and (2) the RMSE of the models for extreme values at the surface on HRES 2022 (Figure 4b). Neither of these two years of data were included in pretraining and no further fine-tuning was applied to these models.

Adding climate simulation data consisting of CMIP6-CMCC and CMIP6-IFS-HR alongside high-quality ERA5 reanalysis data (C2) results in lower RMSEs than the ERA5-only model (C1) across all the groups of variables (except for 2T) (Figure 4a). Despite being lower fidelity than reanalysis datasets like ERA5, climate simulations increase the data diversity and, as our results suggest, generalization performance. By additionally using IFS ensemble data, IFS HRES forecast data, and the IFS ensemble mean (C3), the performance improves further at the surface and around the levels present in the IFS-ENS data, which are highlighted in the figure. Since the improvements of C3 diminish in other parts of the atmosphere, we also test an alternative configuration (C4), which provides a good coverage of the entire atmosphere. Additionally, it limits the samples used in training for the forecasts datasets to one-day lead-times to increase the data quality. This last configuration performs best overall and, for this reason, it is also the configuration used for the main results reported in this paper.

Including additional data in pretraining does not only improve the RMSEs in aggregate, but it also leads to significant improvements (up to 35%) in the tails of the distribution at the surface on HRES 2022 (Figure 4b). Climate simulations and diverse data sources expose the model to more rare events than historical data can provide, which ultimately improves the model robustness in these regimes. As before, C2 improves over C1, while C3 and C4 further improve upon C2. In all instances, the improvements compared to the ERA5-pretrained model (C1) are mostly in the extremes.

Overall, these results provide evidence that using a heterogeneous group of weather and climate datasets, of mixed fidelity, can improve forecasting performance. In Supplementary G.1 we verify that these findings also hold across the individual atmospheric variables and include additional results on HRES 2022. Finally, we also discuss one other dataset configuration that does not yield performance improvements.

Model scaling. Orthogonal to the data scaling dimension, we test the impact of model scaling on three pretrained Aurora configurations, each corresponding to a different model size: 113M, 660M and 1.3B parameters (see Supplementary B.8). All models are pretrained on the dataset mixture from C4 and validated throughout training on IFS analysis data. After the initial training phase, bigger models achieve lower validation loss for the same number of GPU hours (Figure 4c). The bigger models do so while also being trained on fewer data samples (measured in number of tokens in the figure). When fitting a power-law to the performance of the three models at the 5k GPU hours mark, we obtain a scaling law suggesting that validation performance improves by approximately 5% for every doubling of the parameters. Finally, it is important to note here that, in the particular case of weather modelling, training is typically bottlenecked by data loading (and not by the model) since a single datapoint is usually several hundreds of MB in size. Therefore, models that are $10\times$ smaller cannot necessarily process $10\times$ more tokens in the same amount of time. This creates an additional bias towards using larger models and indicates that training large models is not as costly as might first appear.

6 Comparison with AI models at 0.25° resolution

The vast majority of AI models for weather prediction have been (pre)trained at 0.25° resolution on a single dataset. As an additional validation of the benefits of fine-tuning a large model pretrained on many datasets, we compare Aurora against GraphCast (Lam et al., 2023a) – pretrained only on ERA5 and currently considered the most skillful AI model at 0.25°and lead times up to 5 days (Rasp et al., 2024). Additionally, we include IFS HRES in this comparison, the gold standard in numerical weather prediction. We show that Aurora outperforms both when measured against analysis, weather station observations, and extreme values.

Experimental setup. This comparison uses the HRES-T0 2022 dataset as a test year and is carried out at the standard 0.25° resolution used in AIWP. To this end, Aurora is finetuned on HRES-T0 "Analysis" data for years 2016-2021, the same years operational GraphCast was finetuned on. In order to ensure a consistent comparison against operational NWP systems, both GraphCast and Aurora are evaluated using a 00Z/12Z initialization from the HRES-T0 dataset provided by WeatherBench 2. We also treat HRES-T0 as the ground truth targets for this evaluation. To facilitate a complete comparison between models across different variables, pressure levels and lead times we compute and visualize RMSE skill scores which correspond to the relative RMSE difference between the models that are compared, normalized by the RMSE score of the model we compare against. We also include comparisons based on the ACC metric in Supplementary G.

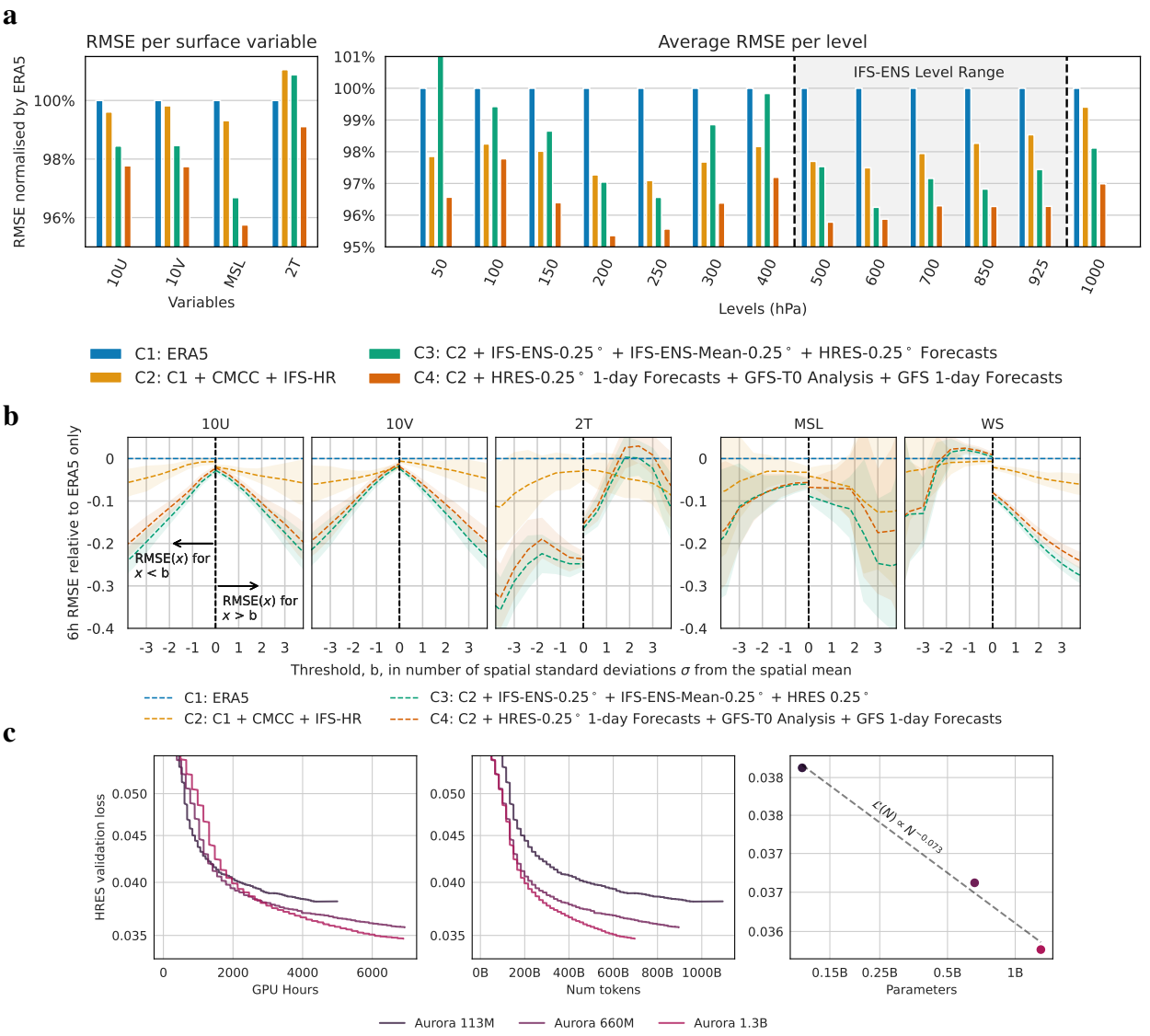

Figure 4: **Pretraining on diverse data and increasing model size improves performance. a**: Performance versus ERA5 2021 at 6 h lead time for models pretrained on different dataset configurations (*i.e.* no fine-tuning) labeled by C1-C4. The root mean square errors (RMSEs) are normalised by the performance of the ERA5-pretrained model (C1). Adding low-fidelity simulation data from CMIP6 (*i.e.* CMCC and IFS-HR) improves performance almost uniformly (C2). Adding even more simulation data improves performance further on most surface variables and for the atmospheric levels present in this newly added data (C3). Finally, configuration C4, which contains a good coverage of the entire atmosphere and also contains analysis data from GFS achieves the best overall performance with improvements across the board. **b**: Pretraining on many diverse data sources improves the forecasting of extreme values at 6h lead time across all surface variables of IFS-HRES 2022. This is consistent with the aggregate results in Figure 12. Additionally, the results also hold on wind-speed, which is a nonlinear function of 10U and 10V. **c**: Bigger models obtain lower validation loss for the same amount of GPU hours. We fit a power law $\mathcal{L}(N) \propto N^m$, where $\mathcal{L}$ is the training loss and N is the model size using the losses at 5 k GPU hours. We obtain a coefficient $m = -0.073$. This roughly translates into a 5% reduction in the training loss for every doubling of the model size.

Aurora vs GraphCast. Aurora consistently outperforms GraphCast (Lam et al., 2023a) across the vast majority of target variables, pressure levels and lead times (Figure 5a). The performance gains are most pronounced at lead times past 3 days, as well as for the upper atmospheric levels where we see a reduction in RMSE up to 40%. The biggest gains are observed for variables such as temperature, geopotential height and wind velocity components, whereas the smallest gains are observed for specific humidity where GraphCast tends to yield more accurate predictions at short lead times (1–3 days) for the surface and lower atmospheric levels. At long lead times Aurora's forecasts more closely resemble the ensemble mean than GraphCast (see Supplementary G.8), but Aurora also outperforms GraphCast at short lead times on U, V, T and Z. Taken together, these results are a collective consequence of Aurora's scale, both in terms of architecture design and training data corpus, as well as its pretraining and fine-tuning protocols. For additional visualisations of this comparison, see Supplementary G.2.

Verification against weather station measurements. As for the high-resolution model, we also evaluate Aurora against station observations for forecasts issued at 00 UTC between 2 January 2022 and 19 December, this time for 2022. Figure 5b shows the RMSE and MAE as a function of lead time for the 0.25°AIWP models versus IFS-HRES. The relative error in temperature between Aurora and GraphCast tracks very closely with the scorecard results in Figure 5a. Despite an advantage in resolution, IFS-HRES only leads within the first 12 hours. For wind speed, Aurora outperforms GraphCast by a larger margin over the first 6 days compared to evaluations on gridded data, but performs worse than GraphCast after day 7 on RMSE values, particularly at 18 UTC timestamps. These results show that AIWP models such as Aurora are still capable of achieving high forecast accuracy as measured by observations, even at the coarser 0.25°resolution.

Extreme event prediction. From a meteorological perspective, accurate forecasting of the extremes of surface weather, such as wind speed and temperature, is of critical importance in forward-planning to mitigate the impact on life. To verify the performance of Aurora in predicting the tails of the relevant distributions, we present a comparison with GraphCast and IFS-HRES using the 06Z/18Z initialization from the 2022 year of the IFS-HRES 0.25°dataset in Figure 5c. Here we show RMSEs computed in the upper and lower tails of the data distribution as described in Supplementary F.2, normalized against IFS-HRES. By sweeping a threshold we can analyse performance as we move further into the extremes. Values for thresholds above zero are shown on the right of the x-axis in each plot, and below zero on the left.

We find that Aurora outperforms GraphCast on wind speed prediction at the surface, and as lead times lengthen, the performance of AIWP models relative to IFS-HRES becomes markedly improved across the distribution. Note however that T2m exhibits a difference in behaviour between the warmer and colder sections of the distribution; it is reported that IFS-HRES performance for winter extremes is biased, resulting in less accurate forecasting during winter months (Ben-Bouallegue et al., 2024). Since AIWP models include such biased forecasting data rather than just forward simulation within physical constraints, there is a tendency to exhibit these biases when compared to NWP models such as IFS. A detailed exploration of the results including atmospheric variables is found in Supplementary G.5.

7 Discussion

The development of Aurora represents a significant step forward in environmental prediction, leveraging the scaling properties of AI foundation models to extract valuable insights from vast amounts of Earth system data. By improving predictive accuracy, resolution, and adaptability, Aurora demonstrates the potential for AI to advance operational weather forecasting and related fields. This progress highlights the importance of continued investment in AI research to address complex challenges in Earth system modeling. As we enter a new era of environmental prediction, Aurora provides a foundation for harnessing the power of AI to enhance our understanding of the planet's systems and thereby inform decisions that have far-reaching impacts on society and on the environment.

While Aurora introduces many new capabilities, there is still a lot of room for improvement along multiple axes. At the moment, the model can only generate deterministic forecasts. A probabilistic treatment is particularly important for variables such as precipitation whose behaviour is inherently stochastic (Price et al., 2024; Lessig et al., 2023; Li et al., 2024; Chen et al., 2023b; Zhong et al., 2024). In the future, this can be addressed by fine-tuning the model into a probabilistic version or by using ensembles of deterministic Aurora models, potentially trained on different data sources. In terms of datasets, while Aurora pushes the boundary of data diversity, it has only been trained on global datasets. With many local high-resolution datasets such as HRRR and CONUS404 being available (Dowell et al., 2022; Rasmussen et al., 2023), exploiting this new spatial scale remains a promising avenue for future work. In terms of computing, our infrastructure consisted of a small pool of preemptible A100 GPUs without InfiniBand, which greatly limited the extent to which we could run expensive data and model scaling ablations. Finding the compute-optimal model size and dataset configuration for pretraining such models can certainly benefit from additional experimentation.

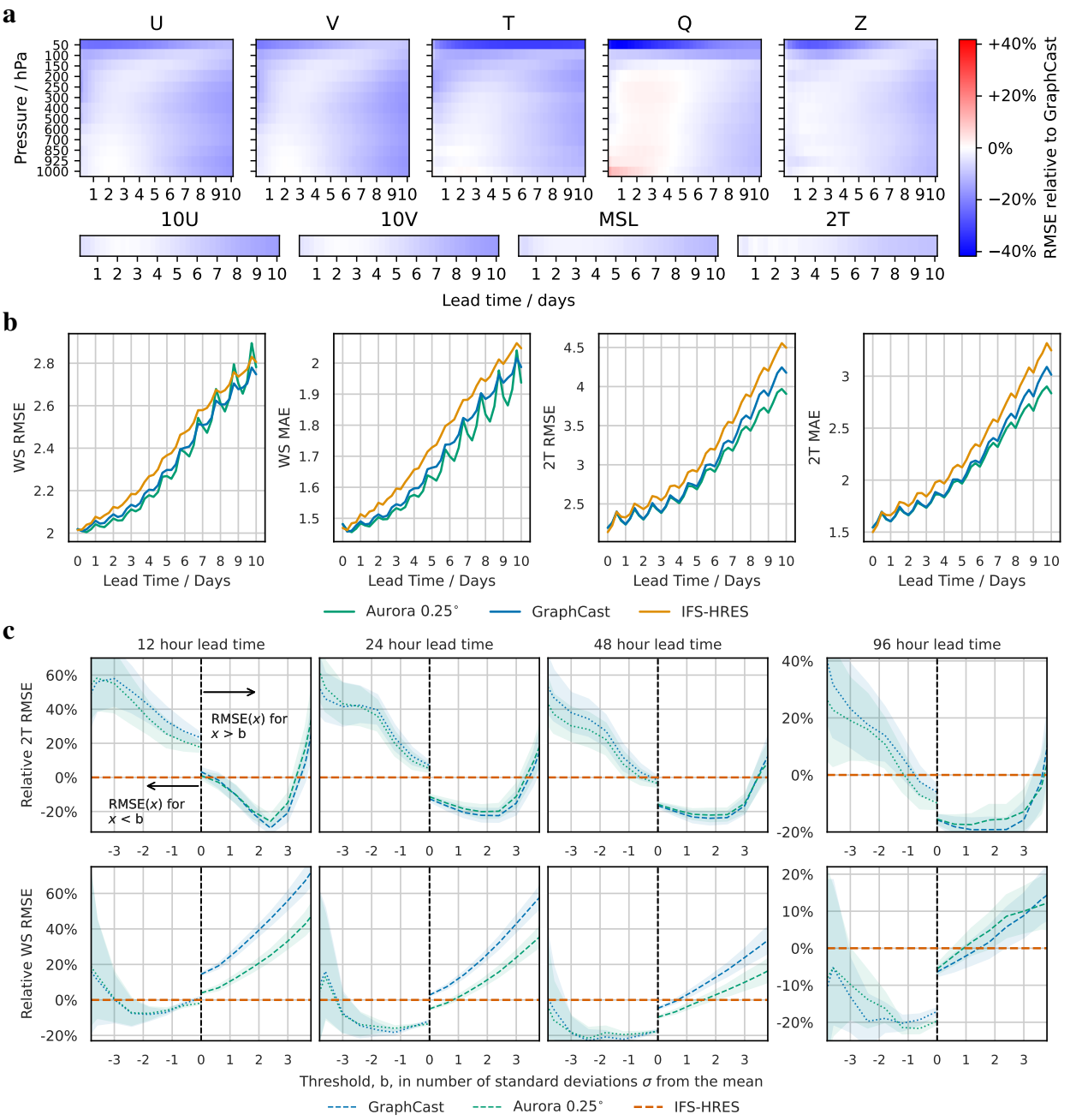

Figure 5: **Aurora outperforms operational GraphCast across the vast majority of targets. a**: Scorecard versus GraphCast at 0.25°. Aurora matches or outperforms GraphCast on 94% of targets. Aurora obtains the biggest gains (40%) over GraphCast in the upper atmosphere, where GraphCast performance is known to be poor. Large improvements up to 10-15% are observed at short and long lead times. The two models are closest to each other in the lower atmosphere at the 2–3 day lead time, which corresponds to the lead time GraphCast was rollout-finetuned on. At the same time, GraphCast shows slightly better performance up to five days and at most levels on specific humidity (Q). **b**: Root mean square error (RMSE) and mean absolute error (MAE) for Aurora, GraphCast, and IFS-HRES as measured by global weather stations during 2022 for wind speed (left two panels) and surface temperature (right two panels). **c**: Thresholded RMSE for Aurora, GraphCast and IFS-HRES normalized by IFS-HRES performance. Aurora demonstrates improved prediction for the extreme values, or tails, of the surface variable distributions. In each plot values to the right of the centre line are cumulative RMSEs for targets found to sit above the threshold, and those to the left represent target values sitting below the threshold. The full procedure is outlined in Supplementary F.2.

Finally, while Aurora outperforms NWP at multiple scales and resolutions, we believe that a lot more work has to be done in terms of model robustness and verification before AI models can truly replace NWP.

The implications of Aurora extend far beyond the realm of atmospheric forecasting. With its extensive potential applications in weather and climate sciences, Aurora represents a significant first step towards the development of a comprehensive foundation model encompassing the entire Earth system. Moreover, the ability of foundation models to excel at downstream tasks with scarce data could have profound implications for environmental forecasting in data-sparse regions, such as the developing world and the polar regions. By leveraging the knowledge learned from data-rich regions, foundation models could enable accurate forecasts even in areas where observational data is limited, thereby democratizing access to life-saving weather and climate information. This could have far-reaching impacts on sectors such as agriculture, transportation, energy harvesting, and disaster preparedness, enabling communities to better adapt to the challenges posed by climate change.

8 Acknowledgements

We thank ECMWF and NOAA for their commitment to open science and their major efforts to generate, curate and openly disseminate all the datasets that enabled our work, and we thank Matthew Chantry for helpful advice on ECMWF's data sources. We thank the CAMS team at ECMWF for insightful discussions. We thank Wenlei Shi, Yue Wang, Pipi Hu, Qi Meng from MSR AI for Science, and Remi Tachet des Combes and Shuhang Chen from MSR for helpful inputs in the early stages of this work. We thank Divya Kumar, Weixin Jin, Sylwester Klocek, Siqi Xiang from MS Start Weather for their technical feedback throughout this project. We also thank Dieter Schwarenthorer for his help with Azure computing and licensing. Finally, we thank Andrew Foong and Frank Noe for constructive feedback during the writing of this manuscript.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*, 2(3):8, 2023. URL `https://cdn.openai.com/papers/dall-e-3.pdf`.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

Rishi Bommasani et al. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL `https://crfm.stanford.edu/assets/report.pdf`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525 (7567):47–55, 2015.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023a. doi: 10.1126/science.adi2336. URL `https://www.science.org/doi/abs/10.1126/science.adi2336`.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, Jul 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06185-3. URL `https://doi.org/10.1038/s41586-023-06185-3`.

Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. FuXi: a cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6(1):190, 2023a.

Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, Yuanzheng Ci, Bin Li, Xiaokang Yang, and Wanli Ouyang. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023b.

Tao Han, Song Guo, Fenghua Ling, Kang Chen, Junchao Gong, Jingjia Luo, Junxia Gu, Kan Dai, Wanli Ouyang, and Lei Bai. FengWu-GHR: Learning the kilometer-scale medium-range global weather forecasting. *arXiv preprint arXiv:2402.00059*, 2024.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. Neural general circulation models for weather and climate. *arXiv preprint arXiv:2311.07222*, 2024.

Christian Lessig, Ilaria Luise, Bing Gong, Michael Langguth, Scarlet Stadler, and Martin Schultz. Atmorep: A stochastic model of atmosphere dynamics using large scale representation learning. *arXiv preprint arXiv:2308.13280*, 2023.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical Fourier neural operators: Learning stable dynamics on the sphere. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2806–2823. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/bonev23a.html`.

Marcin Andrychowicz, Lasse Espeholt, Di Li, Samier Merchant, Alex Merose, Fred Zyda, Shreya Agrawal, and Nal Kalchbrenner. Deep learning for day forecasts from sparse observations. *arXiv preprint arXiv:2306.06079*, 2023.

Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year enso forecasts. *Nature*, 573(7775): 568–572, 2019.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25904–25938. PMLR, 23–29 Jul 2023a. URL `https://proceedings.mlr.press/v202/nguyen23a.html`.

Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Sandeep Madireddy, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. *arXiv preprint arXiv:2312.03876*, 2023b.

Matthew Chantry, Hannah Christensen, Peter Dueben, and Tim Palmer. Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft ai. *Philosophical Transactions of the Royal Society A*, 379(2194):20200083, 2021.

Andrew J. Charlton-Perez, Helen F. Dacre, Simon Driscoll, Suzanne L. Gray, Ben Harvey, Natalie J. Harvey, Kieran M. R. Hunt, Robert W. Lee, Ranjini Swaminathan, Remy Vandaele, and Ambrogio Volonté. Do AI models produce better weather forecasts than physics-based models? a quantitative evaluation case study of storm Ciarán. *npj Climate and Atmospheric Science*, 7(1), 2024.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

European Centre for Medium-Range Weather Forecasts. *IFS Documentation CY48R1*. Number 8. ECMWF, 06/2023 2023a. doi: 10.21957/0f360ba4ca.

R. Buizza, Magdalena Alonso-Balmaseda, Andrew Brown, S.J. English, Richard Forbes, Alan Geer, T. Haiden, Martin Leutbecher, L. Magnusson, Mark Rodwell, M. Sleigh, Tim Stockdale, Frédéric Vitart, and N. Wedi. The development and evaluation process followed at ECMWF to upgrade the integrated forecasting system (IFS). Technical Report 829, 10/2018 2018. URL https://www.ecmwf.int/node/18658.

World Health Organization. *WHO global air quality guidelines: particulate matter ($PM_{2.5}$ and PM_{10}), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. World Health Organization, 2021.

Guy P Brasseur and Daniel J Jacob. *Modeling of atmospheric chemistry*. Cambridge University Press, 2017.

European Centre for Medium-Range Weather Forecasts. *IFS Documentation CY48R1 - Part VIII: Atmospheric Composition*. Number 8. ECMWF, 06/2023 2023b. doi: 10.21957/749dc09059.

A. Inness, M. Ades, A. Agustí-Panareda, J. Barré, A. Benedictow, A.-M. Blechschmidt, J. J. Dominguez, R. Engelen, H. Eskes, J. Flemming, V. Huijnen, L. Jones, Z. Kipling, S. Massart, M. Parrington, V.-H. Peuch, M. Razinger, S. Remy, M. Schulz, and M. Suttie. The CAMS reanalysis of atmospheric composition. *Atmospheric Chemistry and Physics*, 19(6):3515–3556, 2019. doi: 10.5194/acp-19-3515-2019. URL https://acp.copernicus.org/articles/19/3515/2019/.

Diana Francis, Ricardo Fonseca, Narendra Nelli, Deniz Bozkurt, Juan Cuesta, and Emmanuel Bosc. On the middle east's severe dust storms in spring 2022: Triggers and impacts. *Atmospheric Environment*, 296:119539, 2023.

Sylvie Malardel, Nils Wedi, Willem Deconinck, Michail Diamantakis, Christian Kühnlein, George Mozdzynski, Mats Hamrud, and Piotr Smolarkiewicz. A new grid for the ifs. *ECMWF newsletter*, 146(23-28):321, 2016.

Zied Ben-Bouallegue, Mariana C A Clare, Linus Magnusson, Estibaliz Gascon, Michael Maier-Gerber, Martin Janousek, Mark Rodwell, Florian Pinault, Jesper S Dramsch, Simon T K Lang, Baudouin Raoult, Florence Rabier, Matthieu Chevallier, Irina Sandu, Peter Dueben, Matthew Chantry, and Florian Pappenberger. The rise of data-driven weather forecasting: A first statistical assessment of machine learning-based weather forecasts in an operational-like context. *Bulletin of the American Meterological Society*, July 2024.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *arXiv preprint arXiv:2308.15560*, 2024.

Adam Smith, Neal Lott, and Russ Vose. The integrated surface database: Recent developments and partnerships. *Bulletin of the American Meteorological Society*, 92(6):704–708, June 2011. doi: 10.1175/2011BAMS3015.1. URL https://journals.ametsoc.org/view/journals/bams/92/6/2011bams3015_1.xml. Publisher: American Meteorological Society Section: Bulletin of the American Meteorological Society.

UKMO. Storm ciaran, 1 to 2 november 2023.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. ISBN 9781713871088.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2024.

Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13):eadk4489, 2024. doi: 10.1126/sciadv.adk4489. URL https://www.science.org/doi/abs/10.1126/sciadv.adk4489.

Xiaohui Zhong, Lei Chen, Hao Li, Jie Feng, and Bo Lu. Fuxi-ens: A machine learning model for medium-range ensemble weather forecasting. *arXiv preprint arXiv:2405.05925*, 2024.

David C Dowell, Curtis R Alexander, Eric P James, Stephen S Weygandt, Stanley G Benjamin, Geoffrey S Manikin, Benjamin T Blake, John M Brown, Joseph B Olson, Ming Hu, et al. The High-Resolution Rapid Refresh (HRRR): An hourly updating convection-allowing forecast model. part i: Motivation and system description. *Weather and Forecasting*, 37(8):1371–1395, 2022.

RM Rasmussen, Fei Chen, CH Liu, Kyoko Ikeda, A Prein, J Kim, T Schneider, A Dai, D Gochis, A Dugger, et al. CONUS404: The NCAR–USGS 4-km long-term regional hydroclimate reanalysis over the CONUS. *Bulletin of the American Meteorological Society*, 104(8):E1382–E1408, 2023.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12009–12019, June 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. FlexiViT: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14496–14506, 2023.

H. Hersbach, B. Bell, P. Berrisford, G. Biavati, A. Horanyi, J. Munoz Sabater, J. Nicolas, C. Peubey, R. Radu, I. Rozum, D. Schepers, A. Simmons, C. Soci, D. Dee, and J-N. Thepaut. Era5 hourly data on single levels from 1940 to present, 2018a. URL https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview.

H. Hersbach, B. Bell, P. Berrisford, G. Biavati, A. Horanyi, J. Munoz Sabater, J. Nicolas, C. Peubey, R. Radu, I. Rozum, D. Schepers, A. Simmons, C. Soci, D. Dee, and J-N. Thepaut. Era5 hourly data on pressure levels from 1940 to present, 2018b. URL https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview.

ECMWF. Section 2.1.2.4 hres - high resolution forecasts, 2024a. URL https://confluence.ecmwf.int/display/FUG/Section+2.1.2.4+HRES+-+High+Resolution+Forecasts.

ECMWF. Section 5 forecast ensemble (ens) - rationale and construction, 2024b. URL https://confluence.ecmwf.int/display/FUG/Section+5+Forecast+Ensemble+%28ENS%29+-+Rationale+and+Construction.

NOAA. Noaa global forecast system (gfs), 2024a. URL https://registry.opendata.aws/noaa-gfs-bdp-pds.

NOAA. Noaa global ensemble forecast system (gefs), 2024b. URL https://registry.opendata.aws/noaa-gefs.

Enrico Scoccimarro, Alessio Bellucci, and Daniele Peano. Cmcc cmcc-cm2-vhr4 model output prepared for cmip6 highresmip hist-1950, 2018. URL https://doi.org/10.22033/ESGF/CMIP6.3818.

ECMWF. Ecmwf-ifs-hr model output for the "hist-1950" experiment, 2022. URL https://catalogue.ceda.ac.uk/uuid/470e43e166c44e5990f4f74bc90562d6.

GMAO. Merra-2: 2d,1-hourly,time-averaged,single-level,assimilation,single-level diagnostics v5.12.4, 2022. URL https://disc.gsfc.nasa.gov/information/mission-project?title=MERRA-2.

ECMWF. Cams: Global atmospheric composition forecast data documentation, 2024c. URL https://confluence.ecmwf.int/display/CKB/CAMS%3A+Global+atmospheric+composition+forecast+data+documentation.

A. Inness, M Ades, A. Agusti-Panareda, J. Barre, A. Benedictow, A. Blechschmidt, J. Dominguez, R. Engelen, H. Eskes, J. Flemming, V. Huijnen, L. Jones, Z. Kipling, S. Massart, M. Parrington, V-H. Peuch, Razinger M., S. Remy,

M. Schulz, and M. Suttie. Cams global reanalysis (eac4), 2024. URL https://ads.atmosphere.copernicus.eu/cdsapp#!/dataset/cams-global-reanalysis-eac4?tab=overview.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1VaB4cex.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009.

Long-Ji Lin. *Reinforcement learning for robots using neural networks*. PhD thesis, USA, 1992. UMI Order No. GAX93-22750.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey Gritsenko, Mario Lucic, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 2252–2274, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/06ea400b9b7cfce6428ec27a371632eb-Paper-Conference.pdf.

Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020. doi: https://doi.org/10.1029/2020MS002203. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002203. e2020MS002203 10.1029/2020MS002203.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023b.

Supplementary Materials for
Aurora: A Foundation Model of the Atmosphere

Contents

A Notation and problem statement

We denote the observed state of the atmosphere (including the surface) at a certain time t via a $V \times H \times W$ tensor X^t, where V denotes the total number of variables, H specifies the number of latitude coordinates (i.e., the height) and W denotes the number of longitude coordinates (i.e., the width). Thus, we use the $X^t_{v,i,j}$ indexing scheme to refer to the state of variable v at time t and latitude-longitude coordinates given by (i, j). It is sometimes convenient to split this observed state X^t into its surface (S^t) and atmospheric components (A^t). The surface state S^t is a $V_S \times H \times W$ tensor, where V_S denotes the number of surface variables. The atmospheric state is a $V_A \times C \times H \times W$ tensor where V_A denotes the number of atmospheric variables and C represents the number of pressure levels throughout the atmosphere. Thus, the total number of variables is given by $V = V_S + V_A \times C$.

Given a system state at time t, our goal is to learn to predict the next state at a time $t' > t$. For simplicity, we operate with discrete time units of Δt (which is sufficient for any practical purposes), and consider $t' = t + k\Delta t$ for some positive integer k. To predict the state at the next time step, we learn a simulator $\Phi\colon (X^{t-\Delta t}, X^t) \mapsto \hat{X}^{t+\Delta t}$, which maps the observed state of the world at the previous time and the current time to a predicted state $\hat{X}^{t+\Delta t}$ at future time $t + \Delta t$. Since Δt might vary from one task to another, we will often simplify this notation to $\Phi\colon (X^{t-1}, X^t) \mapsto \hat{X}^{t+1}$.

To generate predictions for later time increments, the predictions of the learned simulator can be stacked together in an autoregressive manner. Assuming $\hat{X}^t = X^t, \hat{X}^{t-1} = X^{t-1}$, this can be described recursively as

$$\hat{X}^{t+k} = \Phi(\hat{X}^{t+k-2}, \hat{X}^{t+k-1}). \tag{1}$$

We call this an autoregressive roll-out. It is obtained by applying the function Φ iteratively, a total of k times. Finally, for a certain prediction $\hat{X}^{t'}$, we denote the surface and atmospheric components of that prediction by $\hat{S}^{t'}$ and $\hat{A}^{t'}$, respectively.

B The Aurora model

In this section, we provide a more detailed description of each component of the Aurora model architecture.

B.1 3D Perceiver encoder

Compared to the textual data, training large models on a diverse collection of weather datasets is a substantially more challenging due to the heterogeneous nature of the data. While language is generally homogeneous, weather datasets come equipped with different variables, pressure levels and resolutions. Moreover, due to the costs of storing the complete outputs of a simulation, many sources only make subsets of the total data available by reducing the dataset size along one or more of the axes mentioned above. To accommodate such heterogeneous datasets, we design a flexible encoder that maps different datasets into a standardised 3D tensor that enters the model backbone.

Inputs. The encoder treats all the variables as $H \times W$ images on a regular latitude-longitude grid. For each variable, we include the state at the current time t and the state at time $t - 1$. This results in a $T \times H \times W$ tensor, where $T = 2$ is the time dimension. For a dataset with C pressure levels and V_A atmospheric variables, the state of the atmosphere is represented as a $V_A \times C \times T \times H \times W$. Similarly, for a dataset with V_S surface variables, the state of the surface level is represented as a $V_S \times T \times H \times W$ tensor. In practice, all computations are batched, which adds an additional batch dimension in front of these tensors, which we omit for simplicity.

Static variables. Aurora includes three so-called static variables: (1) information about local orography in the form of geopotential at the surface (Z), (2) a land–sea mask (LST), (3) a soil-type mask (LSM). Internally in the model, the static variables are incorporated by appending them to the collection of surface-level variables. The static variables are part of ERA5's invariant surface-level variables.

Level embeddings. As in standard ViTs, we split each of the $H \times W$ images in $P \times P$ patches. The patches at each level are then mapped into a vector in $\mathbb{R}^D$ by a linear layer, i.e. $C \times V_A \times T \times P \times P \mapsto C \times D$ and $V_S \times T \times P \times P \mapsto 1 \times D$. In order to accommodate datasets with different variables, this linear transformation is constructed dynamically for each variable v using a set of weights W_v specific to that variable. Each of the level embeddings is then tagged with an additive level encoding. For the atmospheric levels, the level encoding is computed via a sine/cosine encoding of the atmospheric pressure associated with each level (e.g., 150 hPa). For the surface level, we use a fully-learned vector of dimension D.

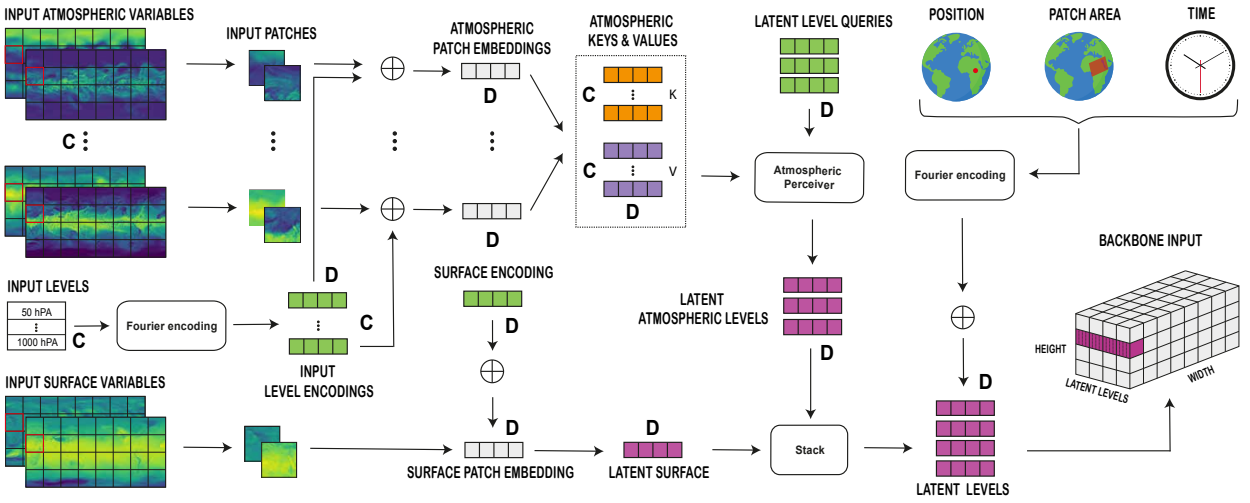

Figure 6: An illustration of Aurora's encoder module. Input weather states are tokenized and compressed into a 3D latent representation using Perceiver-style cross-attention blocks. The resulting latent tokens are augmented with appropriate encodings that provide spatial, temporal, and scale information.

Level aggregation. The next step is to reduce the physical pressure levels in the atmosphere, which can vary in number across datasets, to a fixed set of latent pressure levels. (Nguyen et al., 2023a) introduced a simple aggregation mechanism using attention pooling. Here, we adopt a more expressive scheme that leverages Perceiver modules (Jaegle et al., 2021) consisting of one cross-attention layer followed by a residual MLP. The inputs to the Perceiver are $C_L = 3$ latent query vectors, along with C keys and values vectors computed from the level embeddings via a linear transform. The output of the Perceiver is a $C_L \times D$ tensor, encoding the latent state of the atmosphere. Concurrently, the surface level embedding is simply passed through a residual MLP. This latent state of the surface is then concatenated with the latent state of the atmosphere across the vertical dimension, which yields a $(C_L + 1) \times D$ latent representation of the weather state at the location of each patch.

Scale, position and absolute time embeddings. By gathering the latent representations at each patch, one obtains a 3D tensor $C_L \times H/P \times W/P$, which can be seen as a latent grid on which the backbone will perform the simulation. Therefore, we equip these tokens with information about their latitude and longitude coordinates as well as their physical size (ie., km^2 per patch – which differs depending on the latitude). Finally, we include absolute time information for each of the patches. For more information about these encodings are computed, see Supplementary B.4.

B.2 Multi-scale 3D Swin Transformer U-Net backbone

If the 3D tensor that the backbone receives as input can be viewed as a latent 3D mesh on which the simulation is performed, then the backbone can be viewed as a (neural) simulator. Transformers, due to their proven scaling properties (Kaplan et al., 2020) and connections with numerical integration (Brandstetter et al., 2022), are a natural architectural choice for this simulation engine. To this end, we opt for a 3D Swin Transformer U-Net (Liu et al., 2021, 2022) architecture (Figure 7), which has also been successfully employed by (Bi et al., 2023).

The backbone itself is also composed of an encoder and a decoder, each made out of three stages. After each stage of the encoder, the spatial resolution of the 3D tensor is halved. After each decoder stage, the resolution is doubled and the output is combined with the corresponding outputs of the encoder. This structure enables the backbone to simulate the underlying physics at multiple scales.

Each layer of the backbone is a 3D Swin Transformer layer performing local self-attention operations between tokens in the same regions (called windows), which can be seen as a form of message passing (Gilmer et al., 2017). One such window (of size (2, 4, 2)) is highlighted in Figure 7. Every other layer, all the windows shift across all dimensions by half of the window size along that dimension. In this case, the window would shift by (1, 2, 1) along the three axes, which allows the model to propagate information between neighbouring regions. Due to the spherical topology of the Earth, when the window is shifted, we allow the left and right side of the images to communicate directly as in Bi et al. (2023). Overall, this procedure emulates the local computations performed by numerical integration methods, while

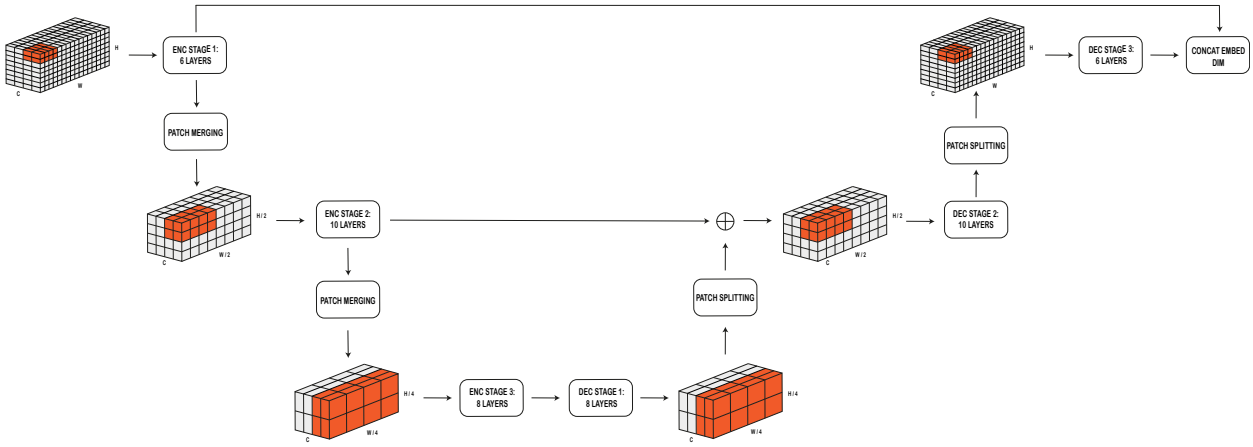

Figure 7: Overview of the 3D Swin Transformer U-Net which is the backbone of Aurora. The U-Net is formed by an encoder (*left*) and a decoder (*right*). Each of these is composed of three stages with (6, 10, 8) and (8, 10, 6) Swin 3D Transformer layers, respectively. In total, the backbone has 48 layers. The highlighted tokens show a single 3D self-attention window of size (2, 4, 2) along the depth, width and height dimensions, respectively. The patch merging layers halve the spatial resolution of the 3D tensor, while the patch splitting layers double it. This allows the backbone to simulate the underlying physics at multiple scales. Although not shown in the figure, the embedding dimension of each token is doubled after each patch merging layer and halved after each patch splitting layer.

avoiding the quadratic complexity of vanilla Transformers, which is impractical for the resolutions Aurora operates at. In practice, we use a window size of (2, 12, 6) for all of our experiments, similarly to Bi et al. (2023).

Notably, to increase stability throughout training, we use the res-post-norm layer normalisation introduced in Liu et al. (2022), but opt for a standard dot-product attention operation instead of the cosine attention procedure from Swin v2 (Liu et al., 2022). Furthermore, to obtain a simple and flexible backbone that can operate at multiple resolutions, we do not use any form of positional bias in the attention operations as previously done by Bi et al. (2023); Liu et al. (2021, 2022), which requires inputs with a fixed spatial resolution. Instead, we opt for positional and scale encodigs in the encoder, as described in the previous section, which does not impose any constraints on the inputs.

Finally, we note that the encoding procedure we employ allows us to flexibly use a small number of latent levels, which in turn frees up memory for massively scaling the number of parameters in the backbone. Compared to the backbone in Bi et al. (2023), which has 16 layers and two stages, our backbone has an additional stage and contains 48 layers.

B.3 3D Perceiver decoder

The decoder maps the standardised outputs of the latent simulation back into images on the regular lat-lon grid. Its structure mirrors that of the encoder. The three latent amtopsheric pressure levels are de-aggregated into C, D-dimensional embeddings (i.e., one per output atmospheric level). As in the encoder, this is done via a Perceiver layer (Jaegle et al., 2021), which uses the sine/cosine embeddings of the output levels' pressure as queries. These vectors are then decoded into $P \times P$ patches via a linear layer. The latent surface level is decoded directly. Analogously to the patch embedding layer in the encoder, the linear layer constructing the output patches is constructed dynamically by selecting the weights associated with each variable. This overall architecture allows the decoder to output predictions at arbitrary pressure levels, for an arbitrary set of variables.

B.4 Position, scale, level, and time encodings

Given the minimum and maximum wavelengths λ_{min} and λ_{max} we want to capture, we use a Fourier encoding of the following form to encode a value x into a D-dimensional vector,

$$\text{Emb}(x) = \left[\cos \frac{2\pi x}{\lambda_i}, \sin \frac{2\pi x}{\lambda_i}\right] \text{ for } 0 \leq i < D/2, \tag{2}$$

where the f_i are log-spaced values between the minimum and maximum wavelength, i.e.,

$$\lambda_i = \exp\left(\log \lambda_{\text{min}} + i \cdot \frac{\log \lambda_{\text{max}} - \log \lambda_{\text{min}}}{D/2 - 1}\right). \tag{3}$$

In the following, we describe how the value x is computed for all the encodings and how λ_{min} and λ_{max} are chosen.

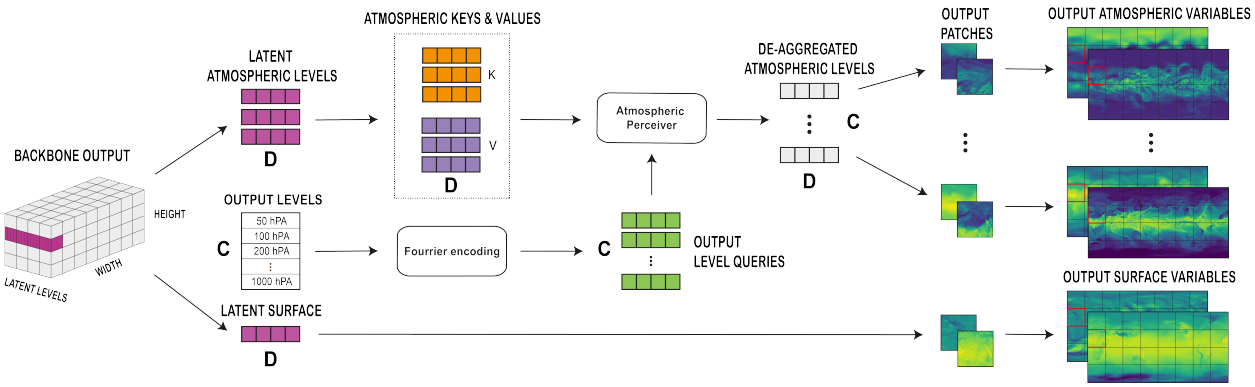

Figure 8: An illustration of Aurora's decoder module. The target output variables are reconstructed in spatial patches by decoding Aurora's 3D latent state using Perceiver-style cross-attention blocks.

Positional encoding. A two-dimensional positional encoding is used to account for the spatial position of each token that enters the Transformer backbone. For an input token of dimension D, the first $D/2$ dimensions are used for latitude information while the next $D/2$ dimensions for longitude information. In contrast to standard positional encodings used in ViT (Dosovitskiy et al., 2021; Nguyen et al., 2023a), where the coordinate IDs of the patch are used, we use a geometry-aware positional encoding given by the average latitude and longitude of each patch. We set $\lambda_{\text{min}} = 0.01$ and $\lambda_{\text{max}} = 720$.

Scale encoding. A separate, non-trainable, two-dimensional positional encoding is added to account for the physical scale of each patch token. Specifically, give a patch with minimum and maximum latitudes ϕ_1, ϕ_2, and minimum and maximum longitudes θ_1, θ_2 (measured in radians), we compute the area of the patch using the formula

$$A = R^2(\sin\phi_2 - \sin\phi_1)(\theta_2 - \theta_1), \tag{4}$$

where $R = 6371$ km is the radius of the Earth. Therefore, patches that originate from a high-resolution input data-set (e.g., IFS HRES data) will yield small areas, while patches that originate from coarser data-sets (e.g., CAMS) will be assigned encodings with larger magnitude. We set λ_{min} to a sufficiently small value and λ_{max} to the surface area of the entire Earth.

Pressure level encoding. Here, we simply encode the value of the atmospheric pressure associated with each level (*e.g.*, 500 hPa). We set $\lambda_{\text{min}} = 0.01$ and $\lambda_{\text{max}} = 10,000$.

Absolute time encoding. We encode the absolute time associated with a certain input as the number of hours since January 1, 1970. We set λ_{min} to one hour and λ_{max} to the number of hours in a year. Therefore, this encoding is able to capture important information such as time of day, time of the week, month of the year, *etc.*.

B.5 Data normalisation

Aurora normalises all variables before processing them in the encoder and unnormalises the output of the decoder to produce the final predictions. Every surface-level variable and every pressure level of every atmospheric variable is normalised separately by a spatially constant scale and centre:

$$X^t_{v,i,j,\text{normalised}} = \frac{X^t_{v,i,j} - \text{centre}_v}{\text{scale}_v}. \tag{5}$$

We collectively call the scales and centres the normalisation statistics. The centres are estimated by the empirical means computed over the whole ERA5 training data, and the scales are estimated by the empirical standard deviations computed over the whole ERA5 training data. These normalisation statistics are then used for all datasets. Final predictions are produced by unnormalising the output of the decoder:

$$\hat{X}^t_{v,i,j} = \text{scale}_v \cdot \hat{X}^t_{v,i,j,\text{normalised}} + \text{centre}_v, \tag{6}$$

where $\hat{X}^t_{v,i,j,\text{normalised}} = \Phi(X^{t-1}_{v,i,j,\text{normalised}}, X^{t-2}_{v,i,j,\text{normalised}})$ represents the raw, unnormalised output of the decoder.

B.6 Extensions for 0.1° weather forecasting

Due to increase in resolution, it becomes extremely challenging to make a large vision model like Aurora fit in the available GPU memory. To that, end we perform a few minor modifications to the architecture at finetuning time.

Patch size increase. We increase patch size $2.5\times$, from 4 (used in pretraining) to 10. Since this corresponds to the increase in resolution from $0.25°$ to $0.1°$, the spatial resolution of the backbone input remains the same as in pretraining. Therefore, there is no memory increase in the backbone computations, where most of the parameters reside. To ensure maximum transfer of skills from pretraining to finetuning, we use a bilinear interpolation to interpolate from the smaller patch size to the larger patch size and scale the magnitude of the interpolated weights by the ratio of the two patch sizes in order to preserve the magnitude of the inputs (Beyer et al., 2023). This ensures that the pretrained model can make reasonable predictions at $0.1°$ even before it is fine-tuned. Ultimately, this also makes fine-tunig significantly faster.

Remove backbone layers. We remove two layers from the two middle stages of the backbone encoder and decoder, respectively. Thus, the number of layers for all stages becomes (6, 8, 8) for the backbone encoder and (8, 8, 6) for the backbone deocoder. Since the model is pretrained using stochastic depth, the model is robust to this change and we do not notice any significant drop in the initial loss before fine-tunning.

B.7 Extensions for air pollution forecasting

In Section 3, we fine-tune Aurora to predict concentrations of air pollutants in addition to a standard collection of meteorological variables. For the reasons described in Section 3, predicting concentrations of air pollutants is considerably challenging. We therefore slightly adapt Aurora to better predict these variables. All adaptations are carefully described in this section.

12-hour model. Some pollutants, such as NO, show clear diurnal cycles (Figure 2a). We therefore fine-tune a version of Aurora that was pretrained with time step equal to 12 hours instead of 6 hours. A 12-hour model which takes in two previous time steps can use the state at time $t - 24\,\mathrm{h}$ to make predictions for time t. The 12-hour model was pretrained exactly like the 6-hour model, but only for $80.5\,\mathrm{k}$ steps, corresponding to roughly a week and a half of training.

Differencing. Since some pollutants show clear diurnal cycles, we build the ability to predict the difference with respect to a previous state explicitly into the model. Only for the new pollution variables, Aurora explicitly predicts the difference with respect to either time $t - 12\,\mathrm{h}$ or time $t - 24\,\mathrm{h}$. Variables NO, TC NO, NO_2, TC NO_2, SO_2, TC SO_2, PM_1, $PM_{2.5}$, and PM_{10} show clear diurnal cycles, so for these variables we predict the difference with respect to time $t - 24\,\mathrm{h}$. For variables CO, TC CO, O_3, and TC O_3 the behaviour is less clear, so for these variables we predict the difference with respect to time $t - 12\,\mathrm{h}$.

We do not restrict Aurora to necessarily predict the difference. In the decoder, we introduce a modulation factor, initialised to one, produced by an additional head:

$$\hat{X}^t = \underbrace{\left[\Phi_{\text{mod}}(X^{t-1}, X^{t-2}) + 1\right]}_{\text{modulation factor}} X^a + \underbrace{\Phi_{\text{pred}}(X^{t-1}, X^{t-2})}_{\text{delta or direct prediction}} \tag{7}$$

where a is either $t - 1$ or $t - 2$, depending on the variable. Φ_{mod} and Φ_{pred} are two different heads of the decoder, both initialised to zero. Therefore, at initialisation, Aurora predicts the appropriate persistence prediction only for the new pollution variables.

Normalisation for concentrations of air pollutants. Normalisation statistics for the new variables are computed over the whole CAMS reanalysis fine-tuning data. Due to the heterogeneous and skewed nature of air pollution variables, the empirical standard deviation is not a good estimate of the scale. Instead, for all air pollution variables v, we set $\text{centre}_v = 0$, and estimate scale_v by half the spatial maximum averaged over time,

$$\text{scale}_v = \frac{1}{2} \cdot \frac{1}{T} \sum_{t=1}^{T} \max \{X^t_{v,i,j} : i = 1, \ldots, H \; j = 1, \ldots, W\}. \tag{8}$$

By construction of this normalisation, the normalised air pollution variables will typically be in the range $[0, 2]$.

The normalisation statistics for the air pollution variables are computed over the whole CAMS reanalysis fine-tuning data (see Table 4). However, these statistics might not optimally match the CAMS analysis fine-tuning data. Based on some heuristics, we increase scale_v for specific v by a specific factor (Table 1) when running Aurora on CAMS.

Table 1: Heuristic factors to increase scale_v by when running Aurora on CAMS analysis data. The normalisation statistics were computed on CAMS reanalysis data. These factors intend to very roughly account for the difference in distribution between CAMS analysis and CAMS reanalysis.

Variable	Pressure level	Factor
PM_1		2
NO	$50\,\text{hPa}$	3
NO	$500\,\text{hPa}$	4
NO	$600\,\text{hPa}$	10
NO	$700\,\text{hPa}$	4
NO_2	$50\,\text{hPa}$	3
SO_2	$200\,\text{hPa}$	2
SO_2	$250\,\text{hPa}$	3
SO_2	$300\,\text{hPa}$	5
SO_2	$400\,\text{hPa}$	10
SO_2	$500\,\text{hPa}$	14
SO_2	$600\,\text{hPa}$	9

Transformation of concentration variables. Variables which are concentration values have a large dynamic range by exhibiting important structure on varying orders of magnitude. Including these variables without any modification makes the network sensitive to high-magnitude values (spikes) and insensitive to low-magnitude values. The usual solution is to use the logarithm of these variables. However, including the logarithm of positive variables makes the network overly sensitive to extremely-low-magnitude values (*e.g.*, zero becomes negative infinity). We therefore include a linear combination of the original variable x and $\log(x)$ which clips away the weaknesses of x and $\log(x)$:

$$x_{\text{transformed}} = c_1 \min(x, 2.5) + c_2 \frac{\log(\max(x, 10^{-4})) - \log(10^{-4})}{\log(10^{-4})}, \tag{9}$$

where c_1 and c_2 are initialised to $\frac{1}{2}$. The particular constants in the above transformation assume that V is normalised to usually be in the range $[0, 2]$. For $10^{-4} \leq x \leq 2.5$, which is the most interesting range, $x_{\text{transformed}}$ depends both on x, which is sensitive to normal-magnitude values, and on $\log(x)$, which is sensitive to low-magnitude values. For $x \geq 2.5$, $\min(x, 2.5)$ clips x to 2.5, so $x_{\text{transformed}}$ then only depends on x through $\log(x)$. For $x \leq 10^{-4}$, the c_2-term is zero exactly, so $x_{\text{transformed}}$ then only depends on x through x.

Additional static variables. To help Aurora better understand diurnal cycles and human behaviour, we extend the static variables to include the time of day (normalised to $[0, 1)$), the day of week (normalised to $[0, 1)$), and the day of the year (divided by 365.25). These normalised values x are included as two constant masks $\cos(2\pi x)$ and $\sin(2\pi x)$. We also include more pollution-related static variables: the monthly average anthropogenic emissions in the year 2019 summed across all industries for (1) ammonia, (2) CO, (3) the sum of NO and NO_2, and (4) SO_2. These monthly averages x are included in two forms: as

$$\frac{x}{\max(x)} \qquad \text{and} \qquad \left[\log\left(\max\left(\frac{x}{\max(x)}, 10^{-10}\right)\right) - \log(10^{-10})\right]/10. \tag{10}$$

The monthly averages can be downloaded from CAMS global emission inventories at `https://ads.atmosphere.copernicus.eu/cdsapp#!/dataset/cams-global-emission-inventories` and are regridded to the resolution of CAMS analysis data. In total, there are $3 \cdot 2 + 4 \cdot 2 = 14$ additional static variables.

Finally, whereas Aurora originally only includes the static variables as additional surface-level variables, we now also include them as additional atmospheric variables by replicating them for every pressure level.

Variable-specific clipping. SO_2 at 850 hPa and above tends to be particularly spikey. To help stabilise roll-outs, during and after LoRA roll-out fine-tuning, we clip the predictions only for SO_2 at $850\,\text{hPa}$ and above at 1. This clipping happens before unnormalisation.

Patch size decrease. The lower resolution of CAMS analysis data ($0.4°$ instead of $0.25°$) allows us to use a smaller patch size of 3 instead of patch size 4. To ensure maximum transfer, we initialise the size 3 patches following the procedure described in Supplementary B.6.

Table 2: The three Aurora model configurations used for the model scaling experiments in section Section 5.

Model size	Backbone enc layers	Backbone dec layers	Embed dim	Attention head dim	Perceiver heads
117M	(2, 6, 2)	(2, 6, 2)	256	64	8
660M	(6, 8, 8)	(6, 8, 8)	384	64	12
1.3B	(6, 10, 8)	(8, 10, 6)	512	64	16

Pressure-level-specific patch embeddings. Near the surface, air pollution variables are strongly affected by human behaviour. However, near the top of the atmosphere, air pollution variables evolve according to complicated atmospheric chemistry dynamics. To deal with the increased variety of behaviours across pressure levels for the atmospheric variables, we make the patch embeddings in the 3D Perceiver encoder and the patch reconstructions in the 3D Perceiver decoder dependent on the pressure level. The pressure-level-dependent patches are initialised with existing pressure-level-independent patches wherever possible and otherwise with uniform random values on the encoder side and with zeros on the decoder side.

Additional 3D Perceiver decoder. To help Aurora with the difference in behaviour between meteorological and air pollution variables, we use a separate 3D Perceiver decoder for the air pollution variables. This separate 3D Perceiver decoder is initialised to the one learned during pretraining.

32-bit floating point computation. To better handle low-magnitude values, all computations before and after the backbone are performed in `float32`. The backbone operates in `bf16` mixed precision mode.

B.8 Model hyperparameters and configurations

The 1.3B parameter Aurora model which we use in the main experiments instantiates the architecture as follows. The embedding dimension in the encoder and first stage of the U-Net backbone is 512. This dimension doubles at every subsequent stage of the backbone. The number of attention heads in the backbone is selected such that the embedding dimension per head is 64 throughout the backbone. Due to the concatenation at the end of the backbone, the embedding dimension in the decoder is 1024. In the Perceiver layers of the encoder and decoder, we use an increased number of cross-attention heads (16), in order to give the model fine-grained control over how the latent state of the atmosphere is constructed.

For the model scaling experiment in section Section 5, we instantiate smaller versions of this model by reducing the number of backbone layers, the embedding dimension and the number of (cross) attention heads, while always preserving the attention head dimension of 64 (Table 2).

C Datasets

We use several different datasets throughout our experiments with Aurora. This section provides a detailed inventory of all these datasets and the various details behind them.

C.1 Types of datasets

The datasets used in the paper can be classified in five categories: analysis, reanalysis, forecast, reforecast and climate simulation datasets. Reanalysis is generated by using a fixed numerical model and data assimilation scheme to recreate a best-guess approximation of observed weather on a 3D grid. The numerical model is needed to estimate weather parameters that are not directly observed. Such *reanalyses* include an assimilation of real observations from time-points forward in time from the frame of interest. However, to initialize operational NWP forecasts, a best-guess state must be provided with incorporation of observations at run time – these are *operational analyses*, resulting in *operational forecasts*.

In addition, one can make after-the-fact *reforecasts* using reanalysis as an initial condition, but since these initialisation values would not be available when making a true forecast, and since they, in effect, 'cheat' by assimilating observational data from the future, these are not representative of realistic forecast settings, but do serve to indicate how NWP models would have predicted the weather if they had access to the best possible global state at initialisation time. Finally, *climate simulations* model the physics, chemistry and biology of the atmosphere to generate potential climate scenarios under different forcing factors (e.g. various emission levels).

C.2 Dataset Inventory

We enumerate all the datasets used in the paper below. Unless otherwise specified, we use the 13 pressure levels from WeatherBench2 (Rasp et al., 2024): $50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000$.

- **ERA5** (Hersbach et al., 2018a,b) is a global reanalysis dataset from ECMWF for weather and climate. Since reanalysis data combines observational data with simulation data, it provides the best estimate we have of the total state of the weather at $0.25°$.
- **HRES Forecasts** (ECMWF, 2024a) refers to the high-resolution version of the operational NWP forecasting model run by ECMWF. It is considered to be the most accurate NWP forecasting model and it runs at $0.1°$resolution. We use it both at its native resolution and regrided to $0.25°$.
- **HRES-T0 "Analysis"** provides the initial conditions used to initialise the HRES forecasts and it is often considered as the ground truth against which to evaluate the quality of the forecasts (Rasp et al., 2024).
- **HRES Analysis** represents the official analysis product of ECWMF and it contains an additional assimilation step on top of HRES-T0. In this sense, it provides a slightly more accurate source of truth.
- **IFS ENS** (ECMWF, 2024b) is the ensemble version of IFS, with 50 members run at a slightly coarser resolution of 18 km (prior to June 27 2023, after which the resolution increased to 9 km with model cycle 48R1). The 50 ensemble members are generated with perturbed initial conditions and stochastic model physics within the IFS model. We use the dataset from the WeatherBench2 repository (Rasp et al., 2024), where there are only 3 pressure levels: $500, 700, 850$.
- **IFS ENS mean** contains the mean predictions for each variable based on IFS ENS. It is provided by WeatherBench2 (Rasp et al., 2024) and contains the same 3 pressure levels.
- **GFS Forecast** (NOAA, 2024a) provides operational forecasts with a base resolution of 18 km. Here we use the data re-gridded to $0.25°$. The zero time (initialisation) of these forecasts is the real-time operational analysis, derived analogously to IFS-HRES zero time.
- **GFS-T0 Analysis** (NOAA, 2024a) refers to the real-time operational analysis, obtained from the zero time (initialisation) of the GFS forecasts.
- **GEFS Reforecast** (NOAA, 2024b) is based on 21 ensemble members to address underlying uncertainties in the input data such limited coverage, instruments or observing systems biases, as well as the limitations of the model itself. In practice, such large quantities of data are only archived for a few preceding months. Therefore, we use the *reforecast* data, spanning 2000-2019, initialised with reanalysis initial conditions at 00 UTC each day. In this setting, there are only five ensemble members, all of which are included. GEFS has 6 pressure levels, we use 3 that line up with those from WeatherBench2: $850, 925, 1000$.
- **CMIP6** is a climate model inter-comparison project, combining various climate modeling experiments, which include land, sea, atmosphere, and aerosol variables. We have two datasets from CMIP6: CMCC-CM2-VHR4 (Scoccimarro et al., 2018) and ECMWF-IFS-HR (ECMWF, 2022), which each have 7 pressure levels: $50, 250, 500, 600, 700, 850, 925$.
- **MERRA-2** (GMAO, 2022) is an atmospheric reanalysis dataset from NASA'S Global Modeling and Assimilation Office, incorporating space-based observations of aerosols.
- **CAMS** (ECMWF, 2024c) refers to analysis and forecast data from the Copernicus Atmospheric Monitoring Service. The data has $0.4°$ resolution and includes meteorological variables as well as variables describing the composition of the atmosphere, such as concentrations of air pollutants. CAMS undergoes frequent model updates.
- **CAMSRA** (Inness et al., 2019) refers to the fourth generation ECMWF global reanalysis of atmospheric composition (EAC4) from the Copernicus Atmospheric Monitoring Service (Inness et al., 2024). The data has $0.75°$ resolution and, like CAMS, includes meteorological and variables as well as variables describing the composition of the atmosphere. However, unlike CAMS, CAMSRA is produced with a single, albeit considerably older, IFS model cycle: CY42R1. Because the resolution is coarser the model cycle is so much older, CAMSRA is considered to be less accurate than recent CAMS data; a quantitive comparison of CAMS and CAMRA against station observations can be found at `https://aeroval.met.no/`.

Collectively, these data sources open different windows onto the underlying atmospheric dynamics, and expose Aurora to different factors that reflect variability with respect to initial conditions, model parametrizations, and chaotic dynamics.

In Table 3, Table 4, and Table 5, we detail the dataset details used for pretraining, fine-tuning, and evaluating Aurora, respectively.

Table 3: Summary of the 10 datasets used to pretrain the different Aurora configurations presented in this work.

Pretraining Datasets							
Name	Resolution	Timeframe	Surface Variables	Atmospheric Variables	Num levels	Size (TB)	Num frames
ERA5	$0.25° \times 0.25°$	1979-2020	2T, U10, V10, MSL	U, V, T, Q, Z	13	105.43	367,920
HRES-0.25	$0.25° \times 0.25°$	2016-2020	2T, U10, V10, MSL	U, V, T, Q, Z	13	42.88	149,650
IFS-ENS-0.25	$0.25° \times 0.25°$	2018-2020	2T, U10, V10, MSL	U, V, T, Q, Z	3	518.41	6,570,000
GFS Forecast	$0.25° \times 0.25°$	2015-2020	2T, U10, V10, MSL	U, V, T, Q, Z	13	130.39	560,640
GFS Analysis	$0.25° \times 0.25°$	2015-2020	2T, U10, V10, MSL	U, V, T, Q, Z	13	2.04	8,760
GEFS Reforecast	$0.25° \times 0.25°$	2000-2019	2T, MSL	U, V, T, Q, Z	3	194.02	2,920,000
CMCC-CM2-VHR4	$0.25° \times 0.25°$	1950-2014	2T, U10, V10, MSL	U, V, T, Q	7	12.6	94,900
ECMWF-IFS-HR	$0.45° \times 0.45°$	1950-2014	2T, U10, V10, MSL	U, V, T, Q	7	3.89	94,900
MERRA-2	$0.625° \times 0.5°$	1980-2022	2T, U10, V10, MSL	U, V, T, Q	13	5.85	125,560
IFS-ENS-Mean	$0.25° \times 0.25°$	2018-2022	2T, U10, V10, MSL	U, V, T, Q, Z	3	10.37	131,400
					Total	1,219.91	11,023,730

Table 4: Summary of the datasets used to fine-tune the different Aurora experiments presented in this work.

Fine-tuning Datasets							
Name	Resolution	Timeframe	Surface Variables	Atmospheric Variables	Num levels	Size (TB)	Num frames
HRES-0.25	$0.25° \times 0.25°$	2016-2021	2T, U10, V10, MSL	U, V, T, Q, Z	13	51.46	179,580
HRES-0.1	$0.10° \times 0.10°$	2016-2022	2T, U10, V10, MSL	U, V, T, Q, Z	13	18.29	10,220
CAMSRA	$0.75° \times 0.75°$	2003-2021	2T, U10, V10, MSL, TC CO, TC NO, TC NO_2, TC SO_2, TC O_3, PM_1, $PM_{2.5}$, PM_10	U, V, T, Q, Z, CO, NO, NO_2, SO_2, O_3	13	3.64	55,480
CAMS Analysis	$0.40° \times 0.40°$	Oct 2017-May 2022	2T, U10, V10, MSL, TC CO, TC NO, TC NO_2, TC SO_2, TC O_3, PM_1, $PM_{2.5}$, PM_{10}	U, V, T, Q, Z, CO, NO, NO_2, SO_2, O_3	13	0.79	3,408

C.3 Training, validation, and test splits

For validation while pretraining, we use one year of IFS HRES at 0.25°resolution: 2020. Our test years are 2022 and 2023, depending on the dataset. Details of dataset splits can be found in Table 3, Table 4, and Table 5.

Motivation for CAMS analysis train–test split. CAMS undergoes frequent updates that dramatically affect the data distribution. Particularly notable are the updates in Sep 2017, when an issue was fixed that caused $PM_{2.5}$ and PM_{10} to be zero; July 2019, when the vertical resolution was increased; May 2021, which significantly alters the behaviour of the PMs; Dec 2022, which alters the background error covariances and significantly alters the distribution of the data; and June 2023, when CAMS is updated to the latest cycle C48R1 and is officially included as a chapter in the IFS documentation (European Centre for Medium-Range Weather Forecasts, 2023b). Note that there were many more updates, but the mentioned ones were observed to be the most significant. Between these updates, the period May 2021 to Nov 2022 inclusive appears to be the most stable period. From this period, we use the last five months, Jun 2022 to Nov 2022 inclusive, for testing; and we include the first twelve months, Jun 2021 to May 2022 inclusive, in the fine-tuning data. We additionally include in the fine-tuning data all data going back to Oct 2017, when the issue where $PM_{2.5}$ and PM_{10} were zero was fixed.

D Training methods

The overall training procedure is composed of three stages: (1) pretraining, (2) short-lead-time fine-tuning, (3) rollout fine-tuning. We describe each of these stages in detail in the following subsections.

Table 5: Summary of the datasets used to evaluate the different Aurora experiments presented in this work.

Evaluation Datasets							
Name	Resolution	Timeframe	Surface Variables	Atmospheric Variables	Num levels	Size (TB)	Num frames
HRES-0.25	$0.25° \times 0.25°$	2022	2T, U10, V10, MSL	U, V, T, Q, Z	13	8.58	29,930
HRES-0.1	$0.10° \times 0.10°$	2023	2T, U10, V10, MSL	U, V, T, Q, Z	13	2.61	1460
CAMS Analysis	$0.40° \times 0.40°$	June 2022-Nov 2022	2T, U10, V10, MSL, TC CO, TC NO, TC NO_2, TC SO_2, TC O_3, PM_1, $PM_{2.5}$, PM_{10}	U, V, T, Q, Z, CO, NO, NO_2, SO_2, O_3	13	0.08	366

D.1 Training objective

Throughout pretraining and fine-tuning, we use the mean absolute error (MAE) as our training objective $\mathcal{L}(\hat{X}^t, X^t)$. Decomposing the predicted state $\hat{X}^t$ and ground-truth state X^t into surface-level variables and atmospheric variables, $\hat{X}^t = (\hat{S}^t, \hat{A}^t)$ and $X^t = (S^t, A^t)$ (Supplementary A), the loss can be written as

$$\mathcal{L}(\hat{X}^t, X^t) = \frac{\gamma}{V_S + V_A}\left[\alpha\left(\sum_{k=1}^{V_S} \frac{w_k^S}{H \times W} \sum_{i=1}^{H}\sum_{j=1}^{W} |\hat{S}_{k,i,j}^t - S_{k,i,j}^t|\right) + \beta\left(\sum_{k=1}^{V_A} \frac{1}{C \times H \times W} \sum_{c=1}^{C} w_{k,c}^A \sum_{i=1}^{H}\sum_{j=1}^{W} |\hat{A}_{k,c,i,j}^t - A_{k,c,i,j}^t|\right)\right], \tag{11}$$

where w_k^S denotes the weight associated with surface-level variable k and $w_{k,c}^A$ denotes the weight associated with atmospheric variable k at pressure level c. The overall surface loss is weighted by $\alpha = \frac{1}{4}$, while the overall atmospheric loss is weighted by $\beta = 1$. Finally, the entire loss for a particular example is weighted by a dataset weight γ, which allows us to upweight the datasets with higher fidelity like ERA5 and GFS-T0. Specifically, we use $\gamma_{\text{ERA5}} = 2.0$, $\gamma_{\text{GFS-T0}} = 1.5$, and set the rest of the dataset weights to 1. When training, we minimise the expected value of this loss computed over a mini-batch of samples.

Variable weighting for pretraining. During pretraining, we set $w_{\text{MSL}}^S = 1.5$, $w_{\text{10U}}^S = 0.77$, $w_{\text{10V}}^S = 0.66$, and $w_{\text{2T}}^S = 3.0$. For the atmospheric variables, for all pressure levels c, we set $w_{\text{Z},c}^A = 2.8$, $w_{\text{Q},c}^A = 0.78$, $w_{\text{T},c}^A = 1.7$, $w_{\text{U},c}^A = 0.87$, and $w_{\text{V},c}^A = 0.6$. The weights are chosen to balance the losses of the individual variables and have been inspired by the weights used by Bi et al. (2023).

Variable weighting for concentrations of air pollutants. Air pollutants are extremely sparse. To balance these variables with the meteorological variables, we require a radically different approach. For the air pollution variables, we will engineer the weights in a way that the per-variable normalised MAE is roughly one.

To begin with, for all air pollution variables, we compute the MAE for the persistence prediction of $12\,\text{h}$ or $24\,\text{h}$, depending on whether the model predicts the difference with respect to the state $12\,\text{h}$ ago or $24\,\text{h}$ ago (Supplementary B.7). These persistence errors are computed on CAMS reanalysis data. We then set

$$w_v^S = \frac{\text{scale}_v}{\text{persistence MAE}_v}, \qquad w_{v,c}^A = \frac{\text{scale}_{v,c}}{\text{persistence MAE}_{v,c}}. \tag{12}$$

Intuitively, multiplication by the scale first undoes the data normalisation and then dividing by the persistence MAE brings the normalised MAE to roughly one.

Variable weighting for IFS analysis fine-tuning. During fine-tuning on IFS analysis $0.25°$ and $0.1°$-resolution data, we slightly adjust the pretraining weights. For the surface-level variables, we set $w_{\text{MSL}}^S = 1.6$, $w_{\text{10U}}^S = 0.77$, $w_{\text{10V}}^S = 0.66$, and $w_{\text{2T}}^S = 3.5$. For the atmospheric variables, for all pressure levels c, we set $w_{\text{Z},c}^A = 3.5$, $w_{\text{Q},c}^A = 0.8$, $w_{\text{T},c}^A = 1.7$, $w_{\text{U},c}^A = 0.87$,and $w_{\text{V},c}^A = 0.6$.

D.2 Pretraining methods

All models are pretrained for $150\,\mathrm{k}$ steps on 32 GPUs, with a batch size of one per GPU. We use a (half) cosine decay with a linear warm-up from zero for $1\,\mathrm{k}$ steps. The base learning rate is $5\mathrm{e}{-4}$, which the schedule reduces by a factor $10\times$ at the end of training. The optimizer we use is AdamW (Loshchilov and Hutter, 2019). We set the weight decay of AdamW to $5\mathrm{e}{-6}$. The only other form of regularisation we use is drop path (*i.e.*, stochastic depth) (Larsson et al., 2017), with the drop probability set to 0.2. To make the model fit in memory, we use activation checkpointing for the backbone layers and we shard all the model gradients across the GPUs. The model is trained using `bf16` mixed precision.

12-hour model. The 12-hour model, used in the air pollution experiments, was trained in exactly the same setting, but only for $80.5\,\mathrm{k}$ steps.

D.3 Short lead-time finetuning

For each task we wish to adapt the pretrained Aurora model to, we start by fine-tuning the entire architecture through one or two roll-out steps (depending on the task and its memory constraints). In all cases, we use a task-dependent hyperparameter selection, which we describe below.

HRES 0.25° Analysis. We fine-tune the weights of the entire model for $8\,\mathrm{k}$ training steps across 8 GPUs, with a batch size of 1 per GPU. At each iteration, we perform two roll-out steps and backpropagate through both of these steps. The model is optimised to minimise the MAE loss averaged across both rollout steps. For this regime, we use a 1k step learning rate warm-up, followed by a constant learning rate of $5\mathrm{e}{-5}$. We use the same weight decay as in pretraining and disable drop path. To ensure the model fits in memory for two rollout steps, we also use activation checkpointing for the encoder and the decoder, along with gradient sharding as in pretraining.

HRES 0.1° Analysis. We fine-tune the weights of the entire model for 12.5k steps across 8 GPUs, with a batch size of 1 per GPU. Due to the increased memory constraints at this higher resolution, we train the model only through a single step prediction. We use a 1k step learning rate warm-up, followed by a constant learning rate of $2\mathrm{e}{-4}$. We set the weight decay to zero and disable drop path. To accommodate the higher memory requirements, we use activation checkpointing for all the layers of the model and use sharding for the weights and gradients.

CAMS 0.4° Analysis. By the high learning rate, we mean $1\mathrm{e}{-3}$ for the encoder patch embeddings of *only* the new pollution variables and $1\mathrm{e}{-4}$ for the rest of the the network. By the low learning rate, we mean $1\mathrm{e}{-4}$ for the encoder patch embeddings of only the new pollution variables and $1\mathrm{e}{-4}$ for the rest of the network. We train with single-step prediction, $12\,\mathrm{h}$ in this case, and the batch size is fixed to 1 per GPU. We use a linear warmup of 100 steps from zero, but we do not use a learning rate schedule after that. We also use no weight decay, disable drop path, use activation checkpointing for all the layers of the model, and use sharding for the weights and gradients.

Fine-tuning on CAMS analysis data proceeds in two steps. In the first step, we fine-tune on CAMS reanalysis data using 16 GPUs for $22\,\mathrm{k}$ steps at the high learning rate and then for $14.5\,\mathrm{k}$ steps at the low learning rate. To ensure maximum transfer from the CAMS reanalysis data to the CAMS analysis data, the CAMS reanalysis data is regridded to the resolution of CAMS analysis data, $0.4°$. In the second step, we fine-tune on CAMS analysis data using 8 GPUs for $7.5\,\mathrm{k}$ steps at the high learning rate and finally for $5.5\,\mathrm{k}$ steps at the low learning rate. The final model is fine-tuned for $49.5\,\mathrm{k}$ steps in total.

D.4 Roll-out fine-tuning

To ensure long-term multi-step dynamics, AI models typically fine-tune the model specifically for rollouts. Backpropagating through the auto-regressive rollouts for a large number of steps is unfeasible for a 1.3B parameter model such as Aurora. This is particularly true at $0.1°$ resolution, where even a single step rollout is close to the memory limit of an A100 GPU with 80GB of memory.

To take advantage of the large size of the model and the fact that it can be easily adapted, for rollout finetuning we use Low Rank Adaption (LoRA) (Hu et al., 2021) layers for all the linear layers involved in the self-attention operations of the backbone. That is, for each linear transformation W involved in the Swin self-attention layers, we learn low-rank matrices A, B to modulate the outputs of W for an input x via $Wx + BAx$. For more details, see Hu et al. (2021).

Furthermore, to avoid any memory increases compared to single-step fine-tuning, we use the "pushforward trick" introduced in Brandstetter et al. (2022), where gradients are propagated only through the last roll-out step. However, to

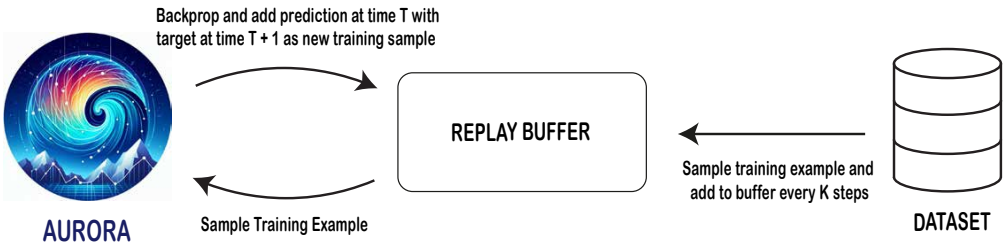

Figure 9: Diagram of the Rollout finetuning procedure. The replay buffer is initially populated with samples from the dataset. At each fine-tuning step, the model fetches a training sample from the replay buffer, performs a training step, and then it adds this new prediction (together with its next step target from the dataset) to the replay buffer. Every K steps, the replay buffer is refreshed with a new training sample from the dataset. Since the replay buffer is generally much smaller than the dataset size, this ensures that enough samples from the dataset are seen. Overall, this procedure allows the model to train fast on an evolving distribution of auto-regressive rollout steps coming from a mixture of model versions. This avoids the need to run expensive rollout procedures at each step.

avoid delays with generating long roll-outs on each training step, we run this at scale by using an in-memory replay buffer, similarly to how it is used in deep reinforcement learning (Lin, 1992; Mnih et al., 2015) (Figure 9). At each training step, the model samples an initial condition from the replay buffer, computes a prediction at the next time step, and then adds this prediction back to the replay buffer. Periodically, fresh initial conditions are fetched from dataset and added to the replay buffer (the dataset sampling period). This procedure allows the model to train at all roll-out steps without extra memory or speed penalties.

HRES 0.25° Analysis. We use 20 GPUs with a buffer size of 200 on each GPU to fine-tune the LoRA layers for $13\,\mathrm{k}$ steps. This results in a total replay buffer size of 4000 samples. We use a dataset sampling period of 10. To ensure the model learns to predict the early steps well before predicting at later steps, we use a schedule, where for the first $5\,\mathrm{k}$ steps, we only keep in the buffer predictions up to 4 days ahead. The 4-10 days lead times are allowed in the buffer only after $5\,\mathrm{k}$. We use a constant learning rate of $5\mathrm{e}{-5}$.

HRES 0.1° Analysis. Since the $0.1°$ data is $6.25\times$ larger than $0.25°$ data, we use 32 GPUs with a buffer size of 20 on each GPU, the maximum we can fit in the CPU memory of each node (*i.e.* $880\,\mathrm{GB}$). We use a dataset sampling period of 10. We train the LoRA weights of the model for $6.25\,\mathrm{k}$ steps. We use a constant learning rate of $5\mathrm{e}{-5}$.

CAMS 0.4° Analysis. We use 16 GPUs with a buffer size of 200 on each GPU and a dataset sampling period of 10. We train the LoRA weights of the model for $6.5\,\mathrm{k}$ steps, and use a constant learning rate of $5\mathrm{e}{-5}$.

E Data infrastructure

Dataloading represented one of the major technical challenges when training Aurora. First, due to the sheer size of one datapoint (close to 2GB for $0.1°$ data), loading the data efficiently on the GPUs becomes extremely challenging. Second, training Aurora on many heterogeneous datasets with different resolutions, variables, and pressure levels, makes the task even more difficult as batching together samples from different datasets becomes nontrivial and the workload of the GPUs must be balanced. In what follows, we describe the data storage and dataloading infrastruture we built to handle these two problems.

Data storage and preprocessing. Because of the large size of the datasets, the data cannot be stored locally, so must be stored with a cloud solution. We use Azure's blob storage. To ensure that data can be efficiently downloaded from blob storage, we perform several optimizations in how the data is stored:

1. Collocate data and compute to minimise latency and costs.
2. To make sure that a worker does not have to download a lot more data than necessary to access a given sample from the dataset, we store each dataset into small chunks/files, containing one or just a few samples. This is particularly necessary as many datasets store in their raw form a month of data (or more) into a single file.
3. For large datasets (like ERA5), we compress the files in order to minimise network bandwidth when downloading samples.
4. For large datasets, we also make sure that all the variables at a given timestep are stored in the same file. This minimises the number of concurrent file downloads that have to be performed to fetch one sample.

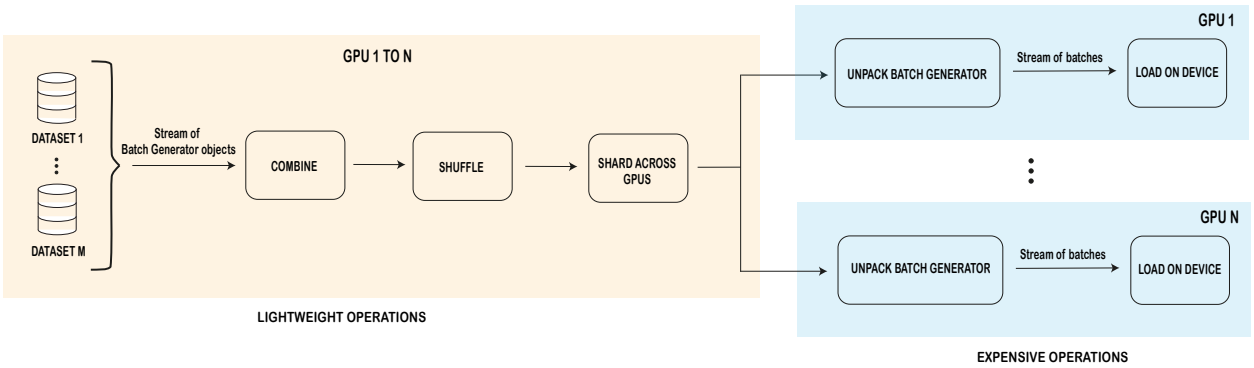

Figure 10: Overview of the multi-source dataloader. Each dataset produces a stream of lightweight `BatchGenerator` objects. The streams are combined, shuffled, and sharded across GPUs. After sharding, each GPU receives a fraction of the `BatchGenerator` objects, which are then unpacked. This step performs the expensive procedures like downloading the data associated with the object from Azure blob storage. This procedure ensures that only samples from the same together are batched together and provides the flexibility to perform custom per-dataset operations.

Data loading. To load heterogoenous data efficiently, we have developed an advanced multi-source data loading pipeline that can satisfy these requirements (Figure 10). Each dataset is instantiated by a `yaml` config specifying how the data from that dataset should be loaded (e.g., dataset location, what lead time should be used, the number of past steps to be supplied as input, etc.). Then, each dataset generates a stream of `BatchGenerator` objects. These are lightweight objects containing the necessary metadata (e.g., file paths) to download and generate at least one batch from the associated dataset. The streams of `BatchGenerator` objects from all the datasets are combined into one stream via sampling, which allows us to adjust the relative proportions of the datasets within each epoch. This is followed by shuffling all the `BatchGenerator` objects and sharding them across GPUs. Up to sharding, all these lightweight operations are run identically on all the GPUs using the same random seeds.

The expensive operations of the dataloading pipeline follow the sharding. The `BatchGenerator` objects sent to each GPU are unpacked into actual batches of data that can be used for training and inference. This unpacking process, consists of a series of sequential transformations that are specific to each dataset and the particular way it instantiates its `BatchGenerator` objects. In general, the sequence of transformation includes: downloading some (compressed) dataset files, decompressing them, reading the data from the files, generating the samples, mapping the variable names to a canonical set of names, performing data checks (e.g., handling NaNs), batching, and, finally, loading batches to device. Additionally, all these operations are performed across multiple workers.

This dataloading pipeline is able to enable efficient training on multiple heterogeneous datasets. Because only samples from the same dataset are batched together, performing batched computations becomes easy and does not impose any additional constraints on the model. Furthermore, since samples from different datasets have different sizes, one can use bigger batch sizes for the smaller datasets and smaller batch sizes for the bigger datasets to balance the workloads across GPUs/workers.

Finally, we note that the simpler problem of training ViTs on multi-resolution images has also been tackled by Dehghani et al. (2023). Their proposed solution is to solve the issue at the model layer by concatenating tokens from different images and using masking to prevent tokens from different images to communicate during self-attention operations. However, their approach imposes a constraint on the model architecture and it is more difficult to implement in the context of more sophisticated ViT architectures like the Swin Transformer. Furthermore, masked self-attention, even if implemented with sparse matrix multiplications, is not able to achieve the expected theoretical performance due to the lack of semi-structured sparsity support on current hardware. Therefore, the advantage of our dataloading pipeline is that it is simple and allows the freedom to experiment on the model architecture side.

F Verification methods

F.1 Main evaluation metrics

The main metrics we use to measure the performance of Aurora against other methods are the root mean square error (RMSE) and the anomaly correlation coefficient (ACC). Because the variables and targets live on a non-uniform grid,

these metrics use a latitude weighting,

$$w(i) = \frac{\cos(\text{lat}(i))}{\frac{1}{H}\sum_{i'=1}^{H}\cos(\text{lat}(i'))}, \tag{13}$$

which downweights smaller cells (closer to the poles) and upweights bigger cells (closer to the equator). The weights are also normalised to unit mean. While various versions of this weighting are used in practice, the form used here is the one from Rasp et al. (2020).

Root mean square error (RMSE). The latitude-weighted RMSE measures the magnitude of the errors between the predictions and the ground truth. It is given by

$$\text{RMSE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} w(i)(\hat{X}_{i,j}^{t} - X_{i,j}^{t})^2}, \tag{14}$$

where t indexes over the sample datasets, i, j index over the latitude and longitude of each image, and $w(i)$ is the latitude weighting factor.

Anomaly correlation coefficient (ACC). The ACC measures the correlation between the deviation of the prediction and the ground truth from the daily climatology (*i.e.*, the daily mean of a variable for that day of the year). It takes the form

$$\text{ACC} = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{i=1}^{H}\sum_{j=1}^{W} w(i)(\hat{X}_{i,j}^{t} - C_{i,j}^{t})(X_{i,j}^{k} - C_{i,j}^{t})}{\sqrt{\left[\sum_{i=1}^{H}\sum_{j=1}^{W} w(i)(\hat{X}_{i,j}^{t} - C_{i,j}^{t})^2\right]\left[\sum_{i=1}^{H}\sum_{j=1}^{W} w(i)(X_{i,j}^{t} - C_{i,j}^{t})^2\right]}}, \tag{15}$$

where $C_{i,j}^{t}$ is the daily climatology for location i, j for the day of the year corresponding to time t. Our daily climatology is computed based on the climatology provided by Rasp et al. (2024).

F.2 Extreme weather metrics

Thresholded RMSE. The thresholded RMSE measures the magnitude of the errors between the predictions and the ground truth. However, here an additional threshold is applied to indicate which latitude-longitude gridpoints should be included in the sum. This can we written as:

$$\text{RMSE}_g = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W}\tilde{w}_{i,j}^{g,t}(\hat{X}_{i,j}^{t} - X_{i,j}^{t})^2}, \tag{16}$$

where, in the case of computing RMSEs above a threshold, the weighting factor is now defined by:

$$\tilde{w}_{i,j}^{g,t} = \frac{\mathbb{1}(X_{i,j}^{t} > b_{i,j}^{g})w(i)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}(X_{i,j}^{g} > b_{i,j}^{g})w(i)}. \tag{17}$$

For the right-sided thresholds (plotted on the right side of the relevant plots). For the left-sided thresholds this is instead defined by:

$$\tilde{w}_{i,j}^{g,t} = \frac{\mathbb{1}(X_{i,j}^{t} < b_{i,j}^{g})w(i)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbb{1}(X_{i,j}^{g} < b_{i,j}^{g})w(i)}. \tag{18}$$

This incorporates the threshold b defined as:

$$b_{i,j} = \mu_{i,j} + g\cdot\sigma_{i,j}, \tag{19}$$

using the mean and standard deviation of the ERA5 reanalysis data over all training years, computed for each latitude-longitude point i, j. Here, g is a factor that is varied linearly for both positive and negative values and is used to obtain the thresholded RMSEs at a number of points, and in the figures we plot the resulting curves. The latitude weighting factor $w(i)$ is as defined above. The value is computed for every example frame, and it is this set of values sampled

with replacement to bootstrap 95% confidence intervals by recomputing Equation (16). For $g > 0$, we plot RMSE_g computed using Equation (17), and for $g < 0$ we plot RMSE_g computed using Equation (18).

The plots show a discontinuity at zero; this is merely due to the lower or upper half of the right-hand or left-hand side curve, respectively, not being shown to prevent over-crowding. However, it is important to point out that RMSE_g is directly comparable to RMSE, and that these checks were performed. Specifically, RMSE_g computed using Equation (18) follows $\text{RMSE}_g \to \text{RMSE}$ as $g \to \infty$, while RMSE_g computed using Equation (17) follows $\text{RMSE}_g \to \text{RMSE}$ as $g \to -\infty$.

G Additional results

In this section, we present numerous supplementary results.

G.1 Effects of data diversity

In addition to the data scaling analysis in Section 5, here we examine with higher granularity the effects of increasing data diversity across all variables and levels. Besides the four dataset configurations C1-C4 discussed in the main body, we also consider here an even bigger dataset configuration based on all the datasets described in Table 3, which we label as C5. This latter configuration contains all the datasets included in C3 and additionally Merra-2, GFS-T0 Analysis, GFS Forecasts and GEFS Reforecasts. We evaluated all these pretrained jobs both on ERA5 2021 (Figure 11) and HRES-T0 Analysis 2022 (Figure 12). No further fine-tuning was performed after the pretraining.

On ERA5, we observe that the trend described in Section 5 holds across all the individual atmospheric variables. C2 improves over C1 almost universally, C3 improves further over C2 on most atmospheric variables for those levels that are covered by the IFS-ENS and IFS-ENS-Mean datasets (which are the biggest datasets in C3). Similarly, C4 improves over C2, as expected due to the inclusion relationship between them. Furthermore, C4 also generally performs better than C3.

Perhaps surprisingly, we notice that C5, which is a superset of all the other configurations, does not perform as well as other configurations and often does worse than the ERA5 job. There are couple of potential hypothesis behind this. A first possibility is that some of the newly introduced datasets in C5 do not align well with the evaluation dataset (ERA5). For instance, Merra-2 is the lowest resolution dataset in the entire pretraining mix and its addition might harm performance on a high resolution dataset like ERA5. At the same time, the GEFS Reforecast dataset only includes three levels in the lower part of the atmosphere (850, 925, 1000 hPa), so it does not provide a good coverage of the atmosphere. A second possibility is that adding such a large amount of data likely requires significantly more training than 150k steps to yield good performance.

On HRES-T0 2022 Analysis, the trend on the surface variabes looks looks similar to ERA5, with the exception of 2T. Since IFS-T0 Analysis is missing an extra assimilation step for the surface, 2T exhibits different biases compared to 2T on ERA5. Therefore, adding any kind of IFS forecast data in pretraining helps alleviate this distribution mismatch and leads to massive improvements. In terms of the atmosphere, performance also generally improves across most variables with the addition of more data, but the order of the configurations fluctuates more. One interesting case is specific humidity (Q), where for unknown reasons, C3 and C4 perform significantly worse than the ERA5-pretrained model on some levels. This could be a potential explanation for why Aurora performs slightly worse than GraphCast at certain lead times and levels of Q in the scorecard from Figure 5a.

Finally, we note that on both test datasets, the biggest improvements from additional data on the atmospheric variables are obtained on the geopotential (Z). This is likely explained by the fact that the physical equations describing the evolution of the geopotential are well-understood and, therefore, the NWP-simulations we added in pretraining provide very high-quality data for this variable. Interestingly, the geopotential also improves universally on C2, despite being absent from the new datasets in this configuration. This demonstrates the ability of the model to learn useful patterns even from incomplete variable sets.

Figure 11: Effects of data scaling measured vs ERA5 2021 across five dataset configurations. The top part of the figure shows aggregate RMSE per level normalised by the performance of the ERA5 pretrained model. The bottom part shows performance across all the individual atmospheric variables and levels.

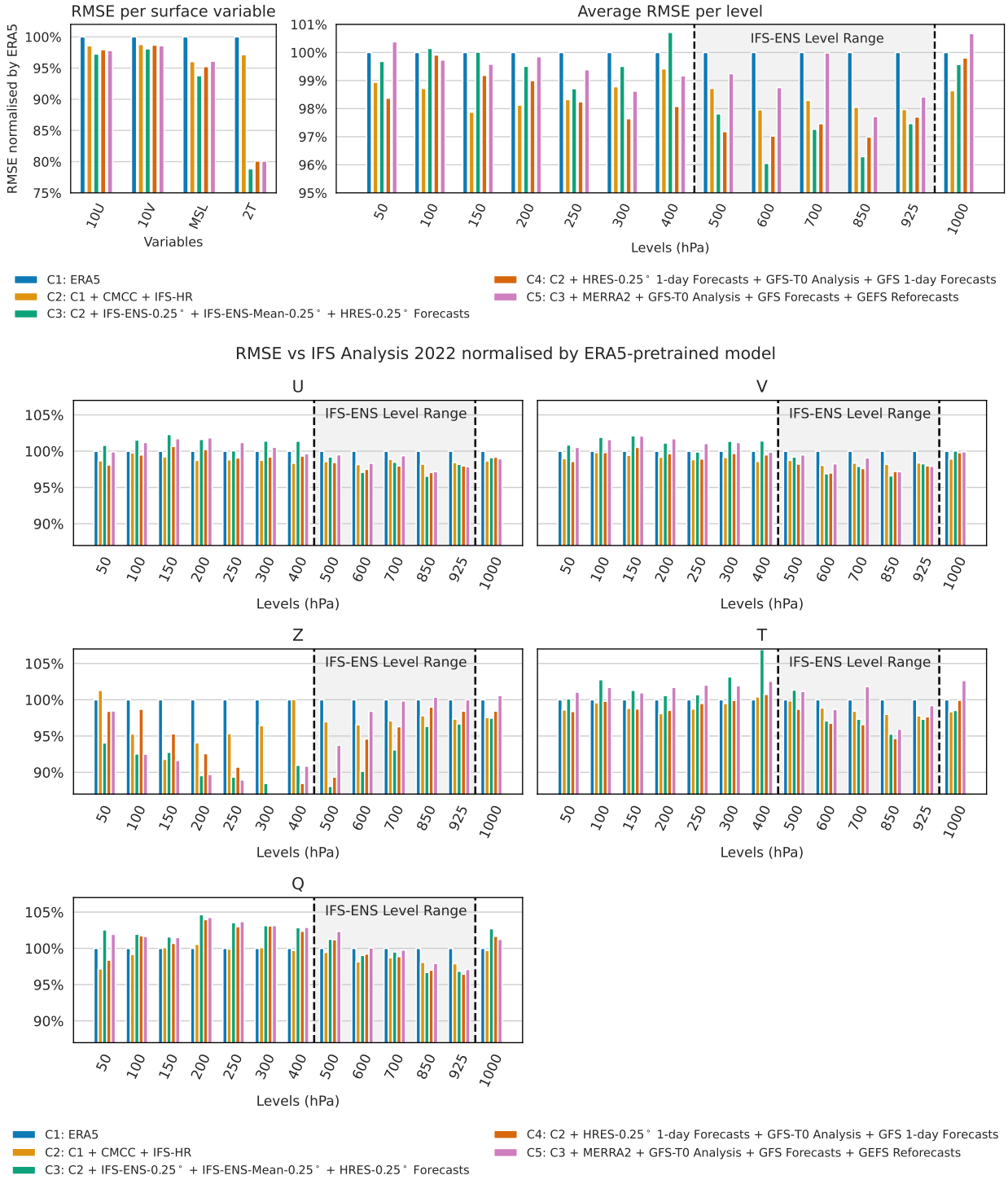

Figure 12: Effects of data scaling measured vs HRES-T0 Analysis 2022 across five dataset configurations. The top part of the figure shows aggregate RMSE per level normalised by the performance of the ERA5 pretrained model. The bottom part shows performance across all the individual atmospheric variables and levels.

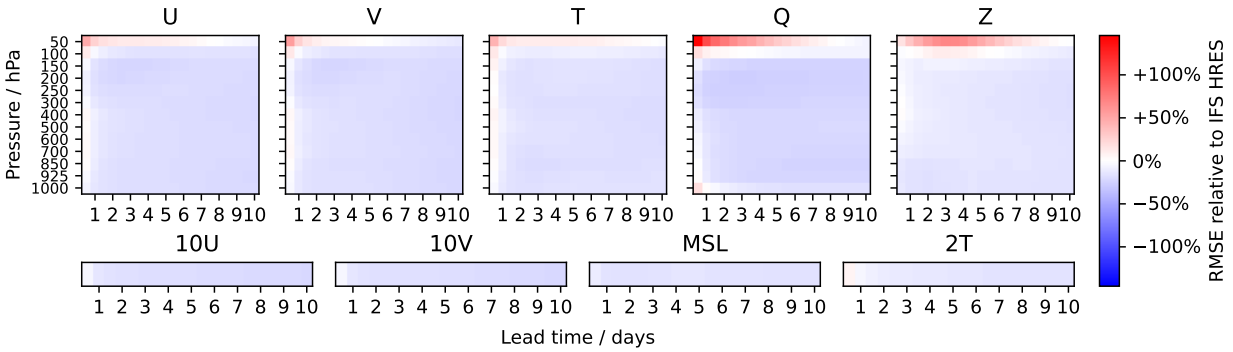

Figure 13: Scorecard of Aurora 0.25° versus HRES 0.25°. Despite significantly improving versus GraphCast in the higher atmosphere, Aurora is also worse than HRES at the top of the atmosphere. However, Aurora generally outperforms HRES on other levels.

G.2 Comparison against operational IFS-HRES and GraphCast at 0.25° resolution

To expand upon the evaluation of Aurora at 0.25°, we include a scorecard comparison with HRES 0.25° in Figure 13. Aurora largely outperforms HRES, but despite its improvements in the upper atmosphere compared to GraphCast, it still performs worse than HRES at the top of the atmosphere.

For a more granular comparison with HRES, we include additional rollout plots comparing Aurora with GraphCast along the following axis: RMSEs in the lower atmosphere (Figure 14), ACCs in the lower atmosphere (Figure 15), RMSEs in the upper atmosphere (Figure 16), ACCs in the upper atmosphere (Figure 17), and RMSEs for the headline metrics (Figure 18).

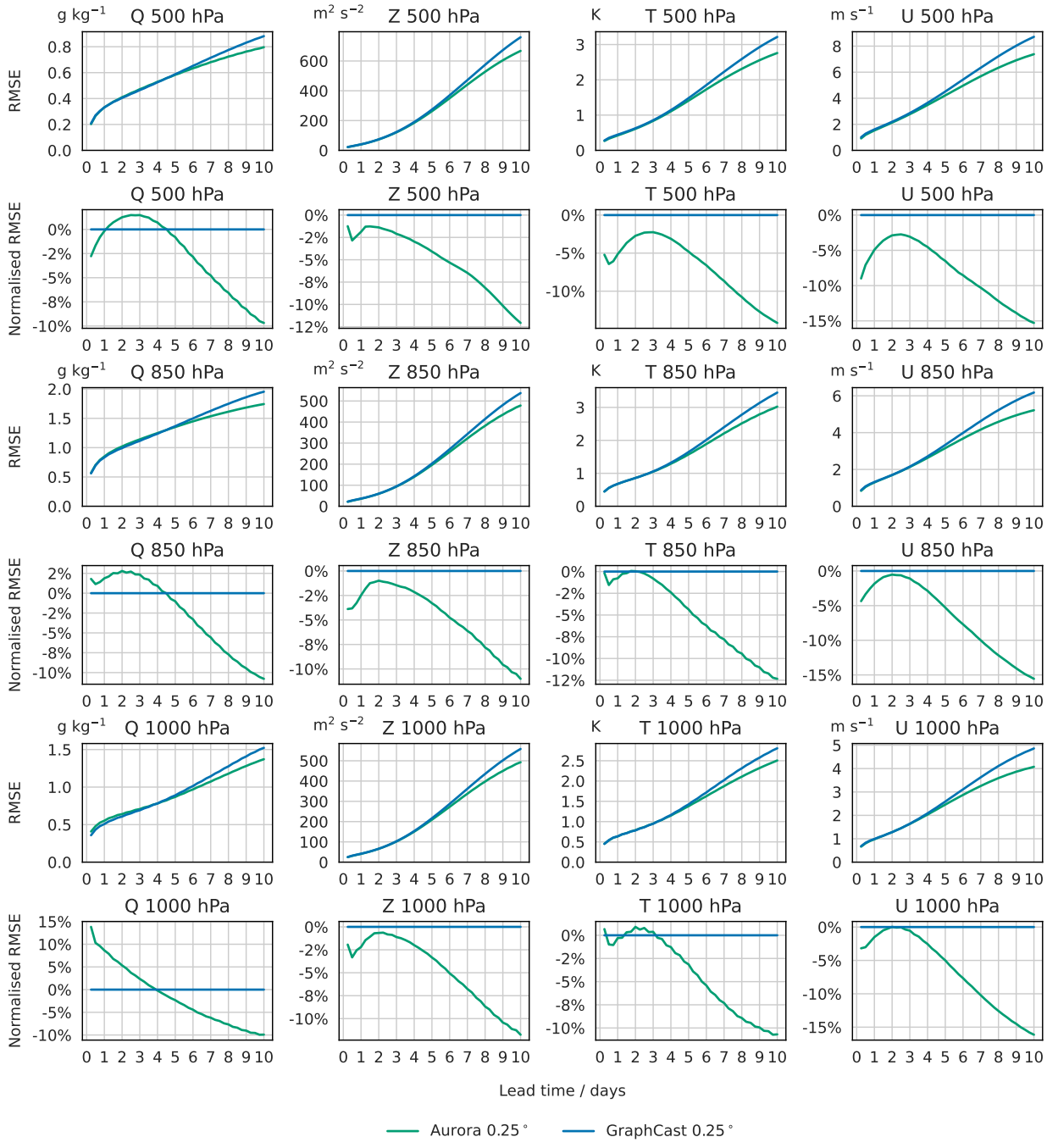

Figure 14: Root mean square error (RMSE) of Aurora 0.25° compared to GraphCast in the lower atmosphere. Shows both unnormalised and normalised RMSEs.

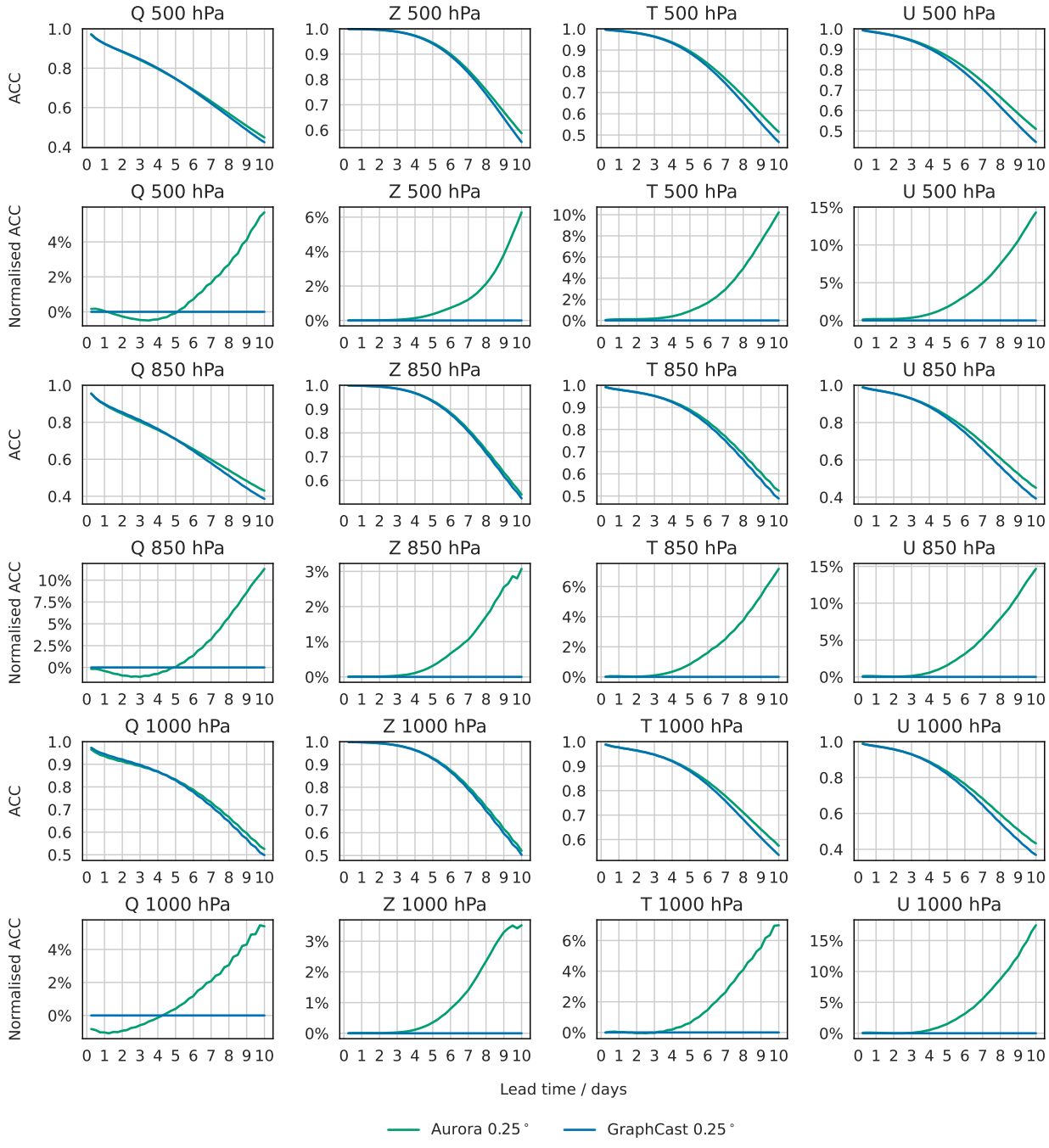

Figure 15: Anomaly correlation coefficient (ACC) of Aurora $0.25°$ compared to GraphCast in the lower atmosphere. Show both unnormalised and normalised ACCs.

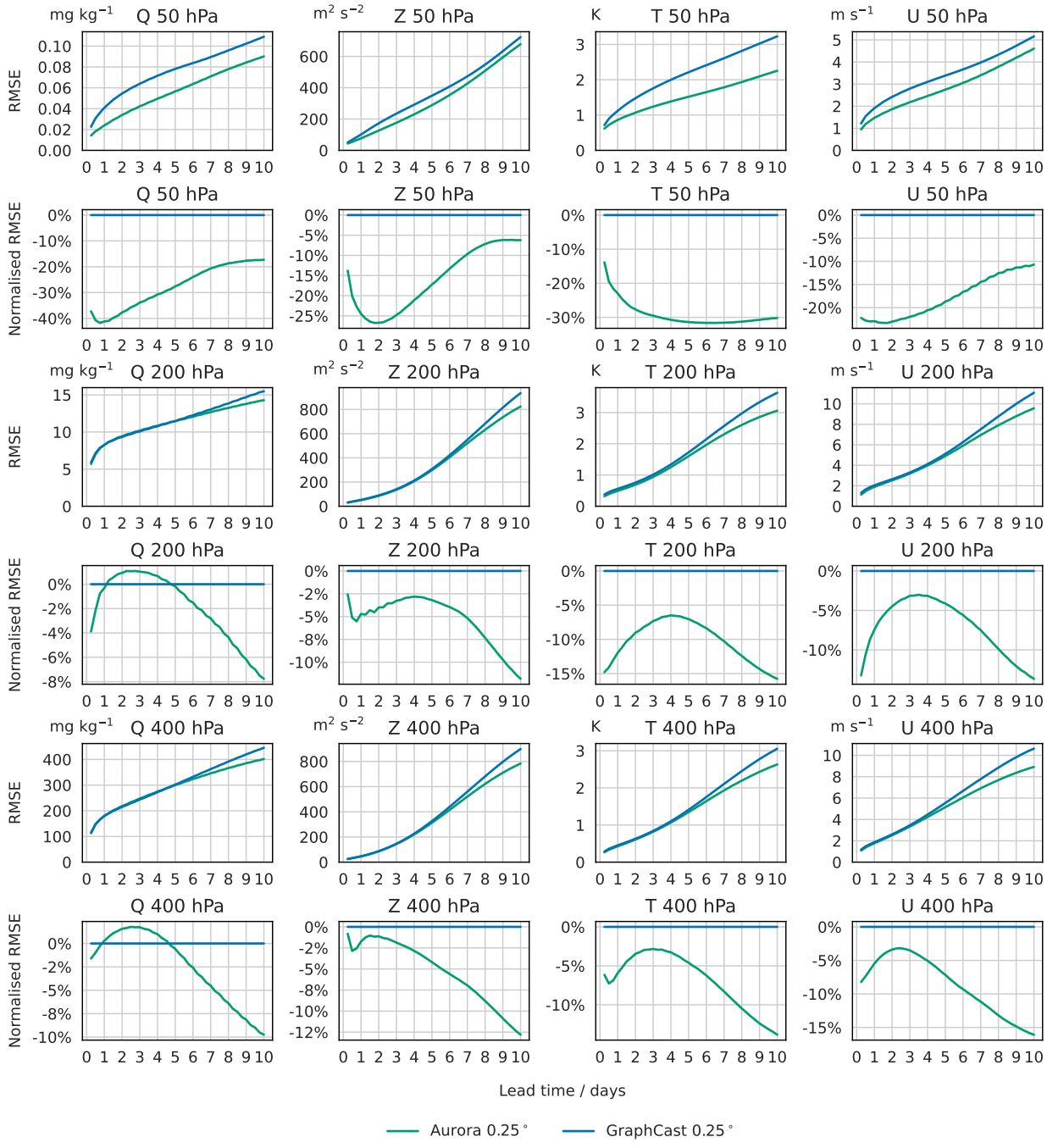

Figure 16: Root mean square error (RMSE) of Aurora $0.25°$ compared to GraphCast in the upper atmosphere. Shows both unnormalised and normalised RMSEs.

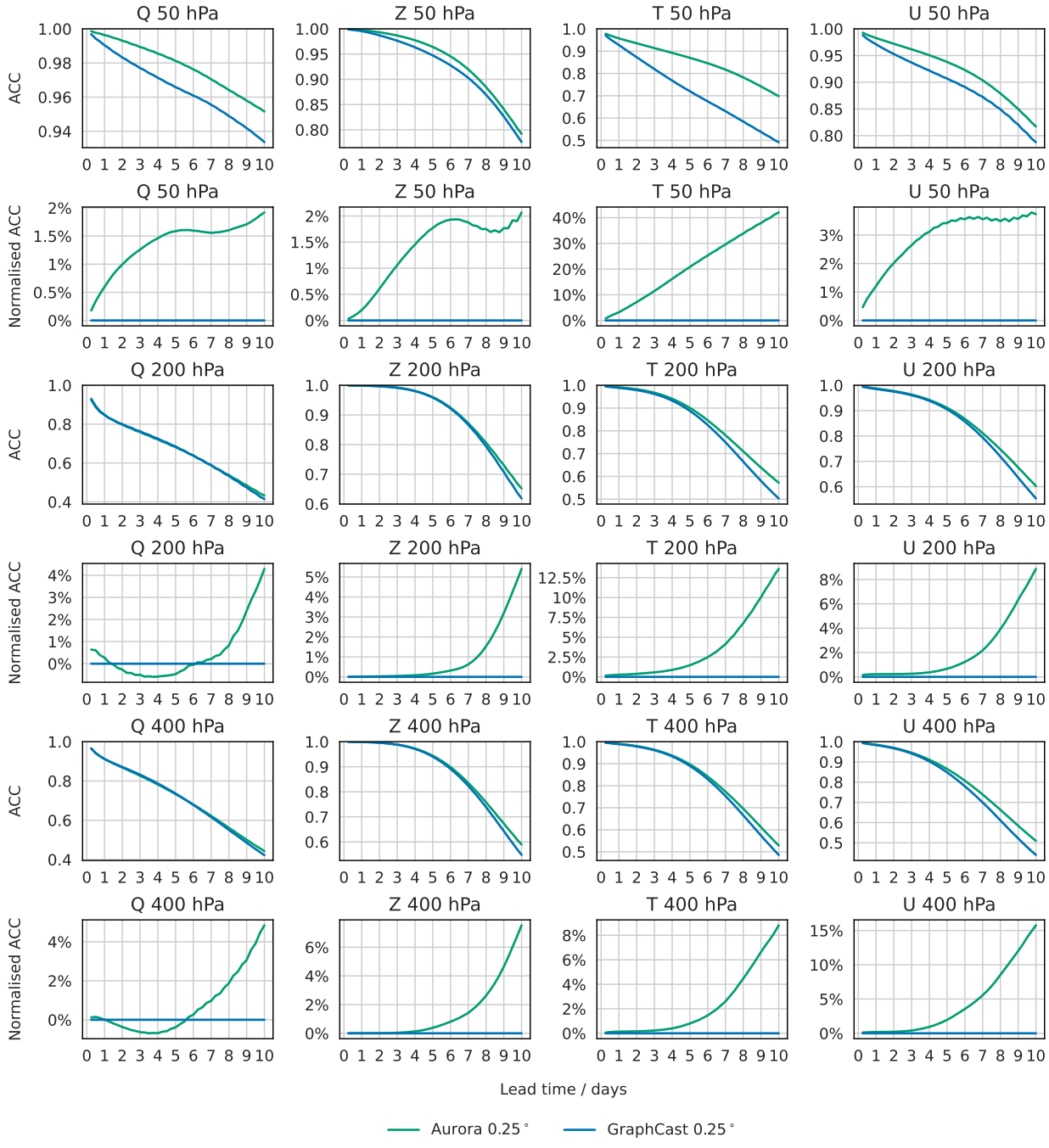

Figure 17: Anomaly correlation coefficient (ACC) of Aurora 0.25° compared to GraphCast in the upper atmosphere. Shows both unnormalised and normalised ACCs.

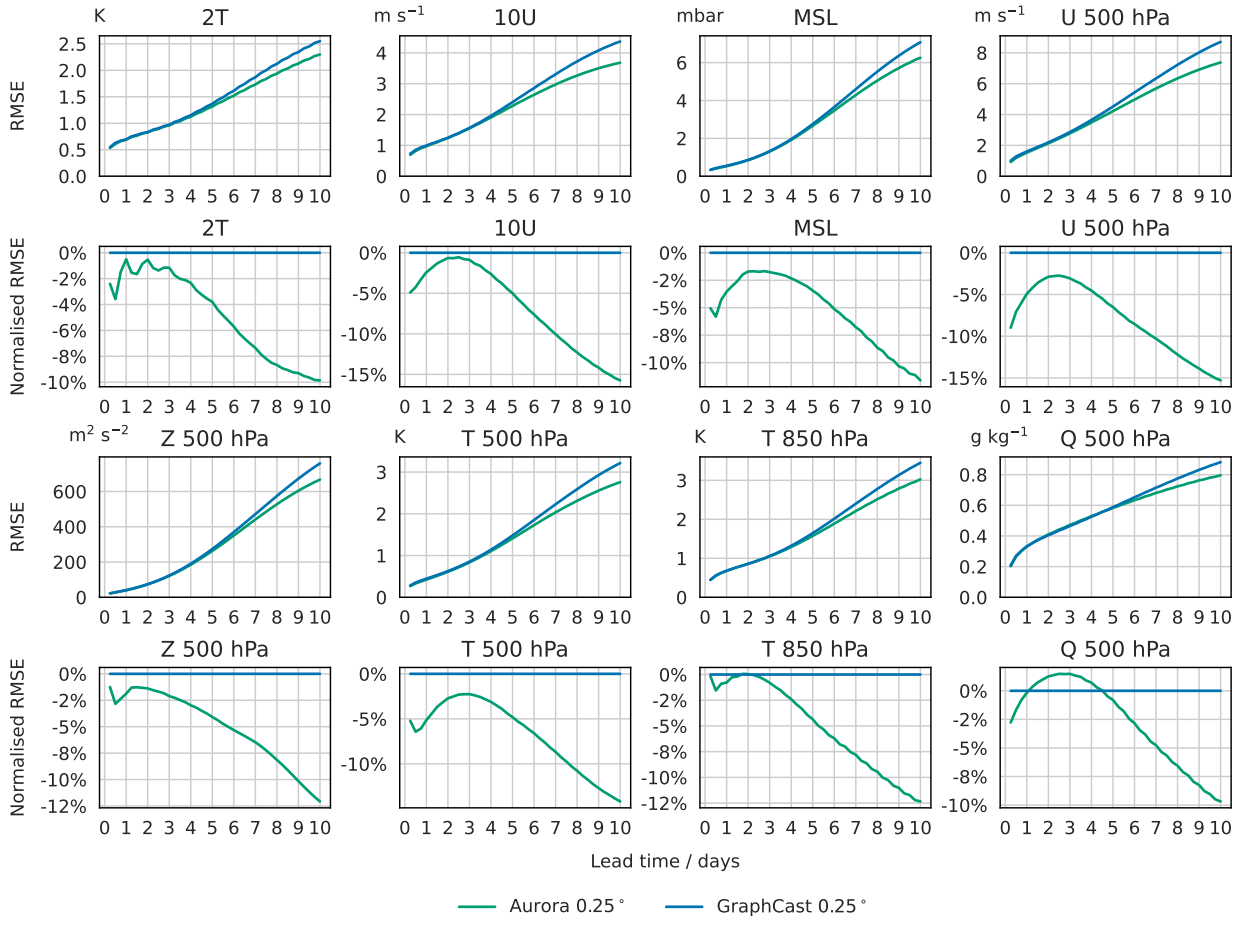

Figure 18: Root mean square error (RMSE) of Aurora $0.25°$ compared to GraphCast for the headline variables. Shows both unnormalised and normalised RMSEs.

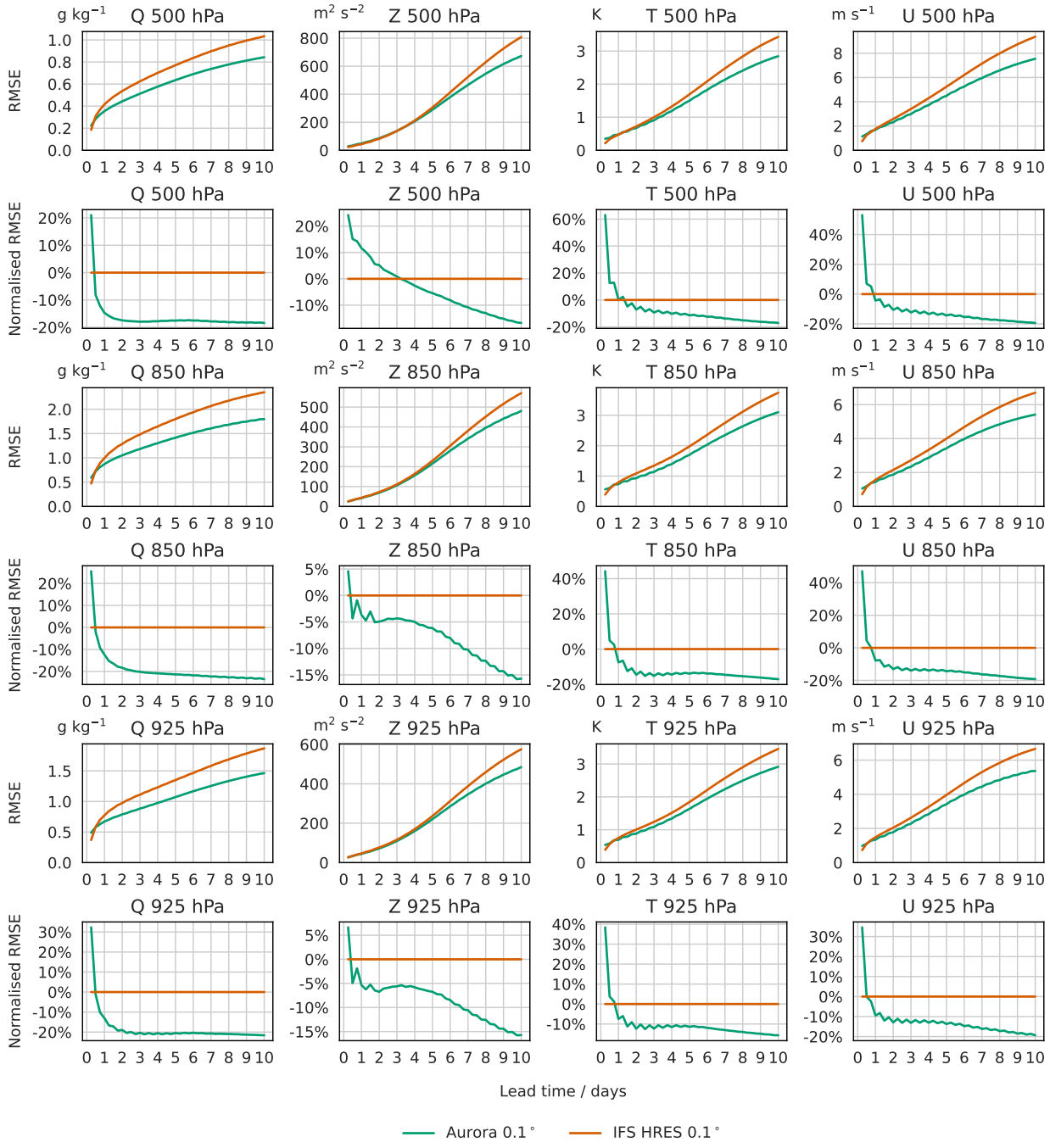

Figure 19: Root mean square error (RMSE) of Aurora 0.25° compared to HRES 0.1° in the lower atmosphere. Shows both unnormalised and normalised RMSEs.

G.3 Comparison against operational IFS-HRES at 0.1 degrees resolution

We include rollout plots vs HRES 0.1°for the lower atmosphere RMSE (Figure 19). Due to the lack of 0.1°forecasts in the upper atmosphere, we include as a preliminary result, a scorecard vs HRES 0.25°in Figure 20. Although this is not an apples to apples comparison, HRES 0.25°RMSEs tend to be lower than HRES 0.1°RMSEs since higher-resolution models have more fine-grained details that they need to get right. Therefore, the scorecard provides a good approximation to the performance of Aurora 0.1°in the upper atmosphere.

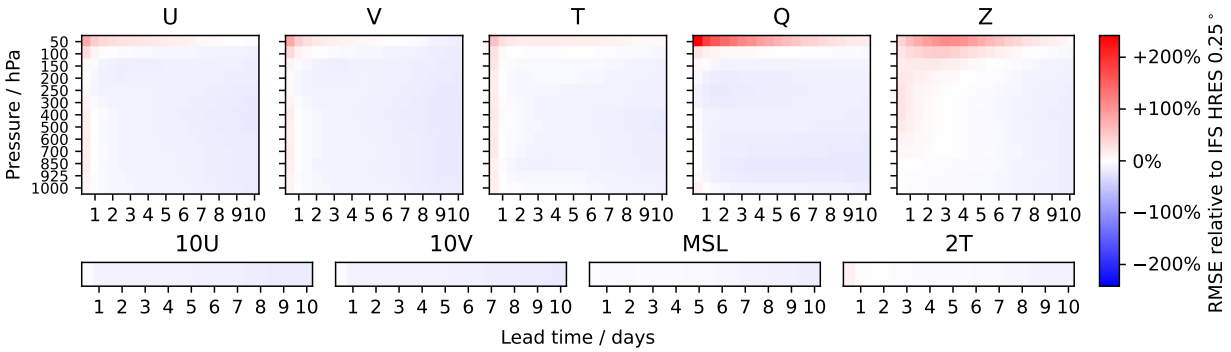

Figure 20: Scorecard comparing Aurora 0.1°vs HRES 0.25°across all levels. We note that the RMSEs of 0.25°HRES are generally lower than those of 0.1°HRES. Nonetheless, Aurora shows good performance and generally outperforms HRES also in the higher atmosphere.

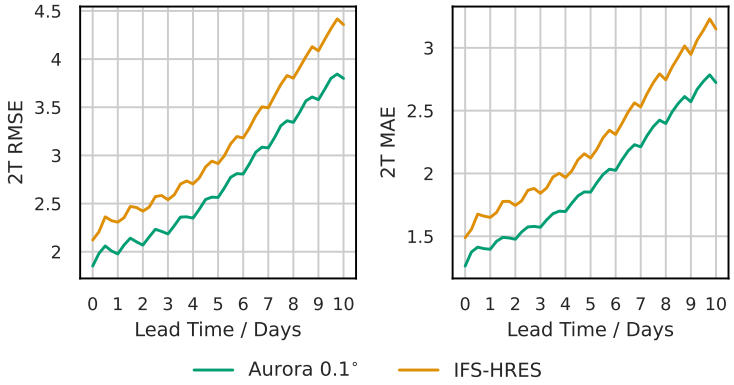

Figure 21: 2 m temperature performance at weather station observation points for Aurora 0.1°versus IFS-HRES 0.1. RMSE and MAE. Due to the different initialisations, Aurora has an advantage. Nonetheless, Aurora's forecast performance degrades at roughly the same rate as HRES as the lead time increases.

G.4 Comparison against weather station measurements

As discussed in Section 4 most verification efforts for AIWP models have focused on gridded analysis data such as ERA5 (*c.f.* WeatherBench 2). Here we make a comparison of Aurora and IFS-HRES at 0.1°resolution. This evaluation includes forecasts issued at 00 UTC between 2 January 2023 and 19 December 2023. As we showed in Figure 3c, Aurora outperforms IFS-HRES for 10 m wind speed forecasts across all lead times up to 10 days. This section includes further results on further results for 2 m temperature (Figure 21).

As explained in Section 4, Aurora is initialised from the official analysis product, while forecasts are initialised from T0 "Analysis". The former includes an additional assimilation step for the surface, which improves the 2T estimate. Therefore, Aurora has an unfair advantage in terms of initial conditions when compared on 2T (Figure 21). Nonetheless, the errors of HRES and Aurora remain parallel as the lead time increases which shows that the forecast quality of Aurora does not degrade over time compared to HRES.

G.5 Extreme events forecasting

In addition to an evaluation of model performance on extremes of surface variables, here we extend the analysis to include the atmospheric variables. In Figure 22 we show the thresholded RMSES for atmospheric variables distributed over four pressure levels throughout the atmosphere. At the higher pressure levels (near surface) results are consistent with those shown for the surface variables in Section 6. However, the temperature in the atmosphere is more typically distributed as it is less susceptible to bias through near-surface effects. In the higher atmosphere, overall performance is more challenging, although Aurora remains able to out-perform GraphCast, consistent with total RMSE results shown in Figure 5a.

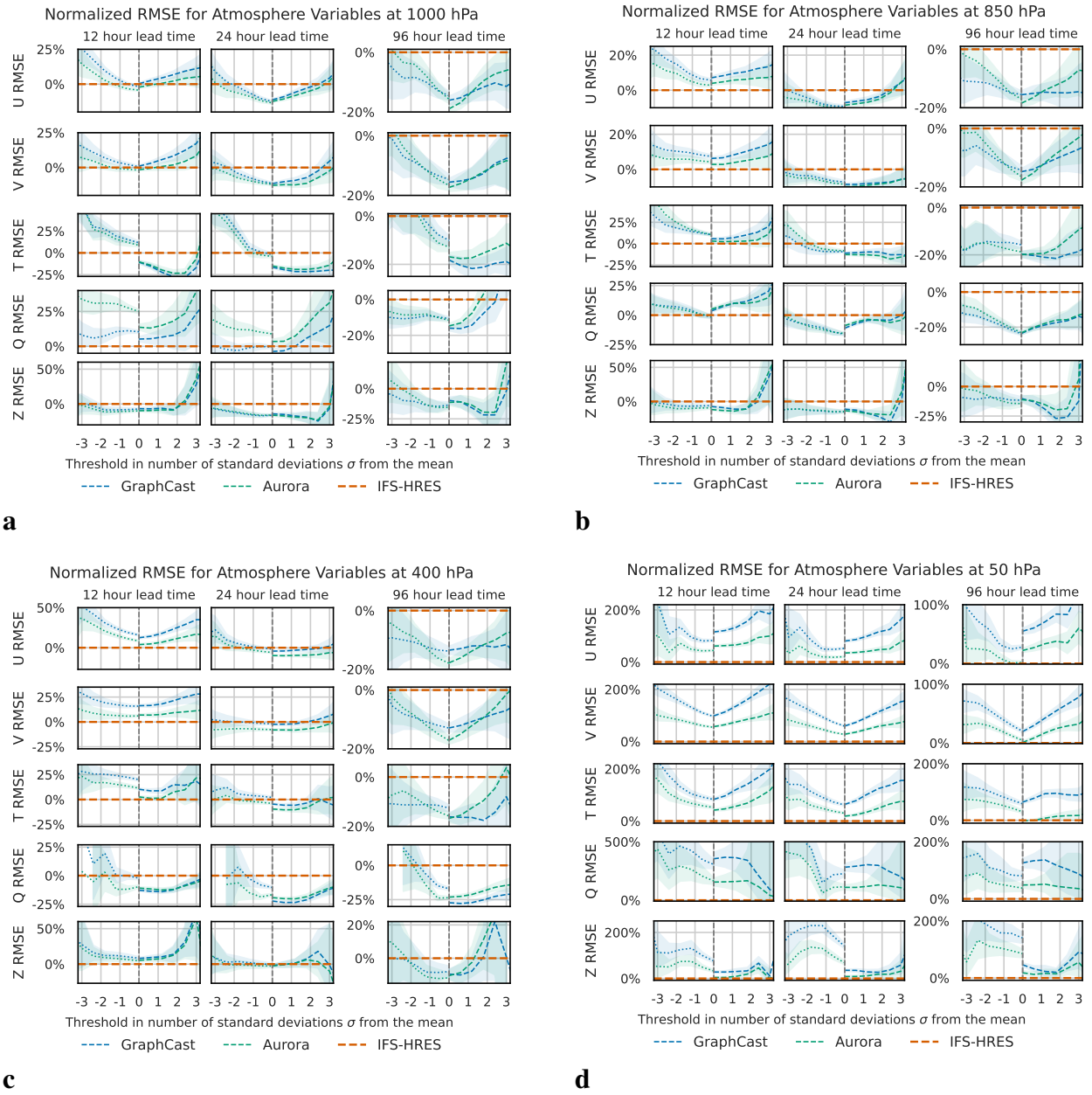

Figure 22: Thresholded RMSE results for atmospheric variables at four pressure levels distributed throughout the atmosphere, presented as percentages relative to the performance of IFS-HRES at 0.25°resolution. The origin of these curves is explained in Supplementary F.2. The discontinuity at zero reflects that only one side of the curve is shown; on the right hand side of each figure we show RMSE_g where target values are above the threshold determined by g, and on the LHS of the figure, RMSE_g is the cumulative RMSE found against target values below the threshold.

G.6 Storm Ciarán

Figure 23 shows minimum mean sea level pressure (left) and maximum $10\,\text{m}$ wind speed (right), including additional variants of Aurora, namely, with and without LoRA fine-tuning, at both 0.25°and 0.1°resolution. We evaluate all methods based on how closely they resemble IFS analysis at the corresponding resolution, and find that (as expected) the Aurora variants at 0.1°resolution clearly outperform those at 0.25°resolution. For minimum mean sea level pressure, we see that Aurora at 0.1°(with LoRA fine-tuning) aligns almost perfectly with IFS analysis at 0.1°. In Section 4, we showed that Aurora is the only AIWP model capable of capturing the peak in maximum $10\,\text{m}$ wind speed on 2 November 2023 at 00 UTC, where the best version of Aurora is at 0.1°, without LoRA fine-tuning.

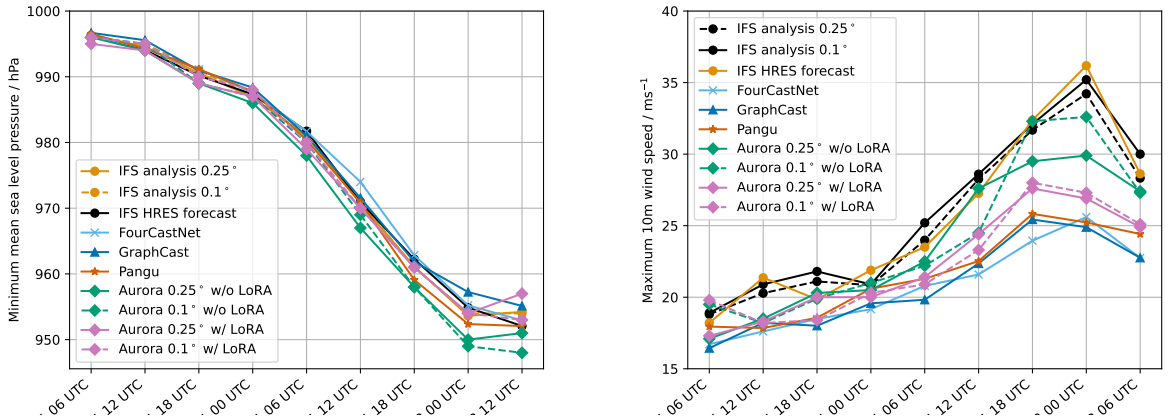

Figure 23: Minimum mean sea level pressure (left) and maximum $10\,\mathrm{m}$ wind speed (right) for Storm Ciarán, at $0.25°$ and $0.1°$ resolution, both with and without LoRA fine-tuning.

We now compare Aurora with and without LoRA fine-tuning. For minimum mean sea level pressure, we find that the Aurora variants *with LoRA fine-tuning* perform better (*i.e.*, align more closely with IFS analysis) than those without LoRA fine-tuning. For maximum $10\,\mathrm{m}$ wind speed, we find that the opposite is true: Aurora variants *without LoRA fine-tuning* perform better. We postulate that LoRA fine-tuning is more useful when forecasting quantities such as mean sea level pressure because this is a smoother quantity. This is in contrast to forecasting maximum wind speed, which relies on generating realistic forecasts as opposed to smoothed-out ones.

We view that the difference between Aurora forecasts with and without LoRA fine-tuning is analogous to the difference between the forecasts by the IFS ensemble mean and IFS HRES. Forecasts based on the ensemble mean give us the "best prediction" according to minimizing the RMSE, which typically results in less realistic, more smoothed-out predictions. In contrast, forecasts based on IFS HRES give more realistic-looking samples. Choosing whether or not to employ LoRA fine-tuning therefore allows us to offer two versions of Aurora: one for when we want to minimize the forecast RMSE, and another for when we want to generate realistic predictions.

G.7 Fast prediction of atmospheric chemistry and air pollution

Figure 24 shows a scorecard for the full collection of variables. For the full collection of variables, Aurora is competitive to CAMS (within 20% RMSE) on 96% of all targets, and Aurora matches or outperforms CAMS on 76% of all targets. For just the surface-level variables, Figure 25 compares RMSEs to CAMS. Figure 26 compares the RMSE of Aurora, CAMS, and the persistence prediction at a three day lead time. Finally, Figure 27 visualises predictions for PM_{10} on 13 June 2022, when Iraq was hit by a particularly bad sandbox. Aurora predicts the sandstorm a day in advance.

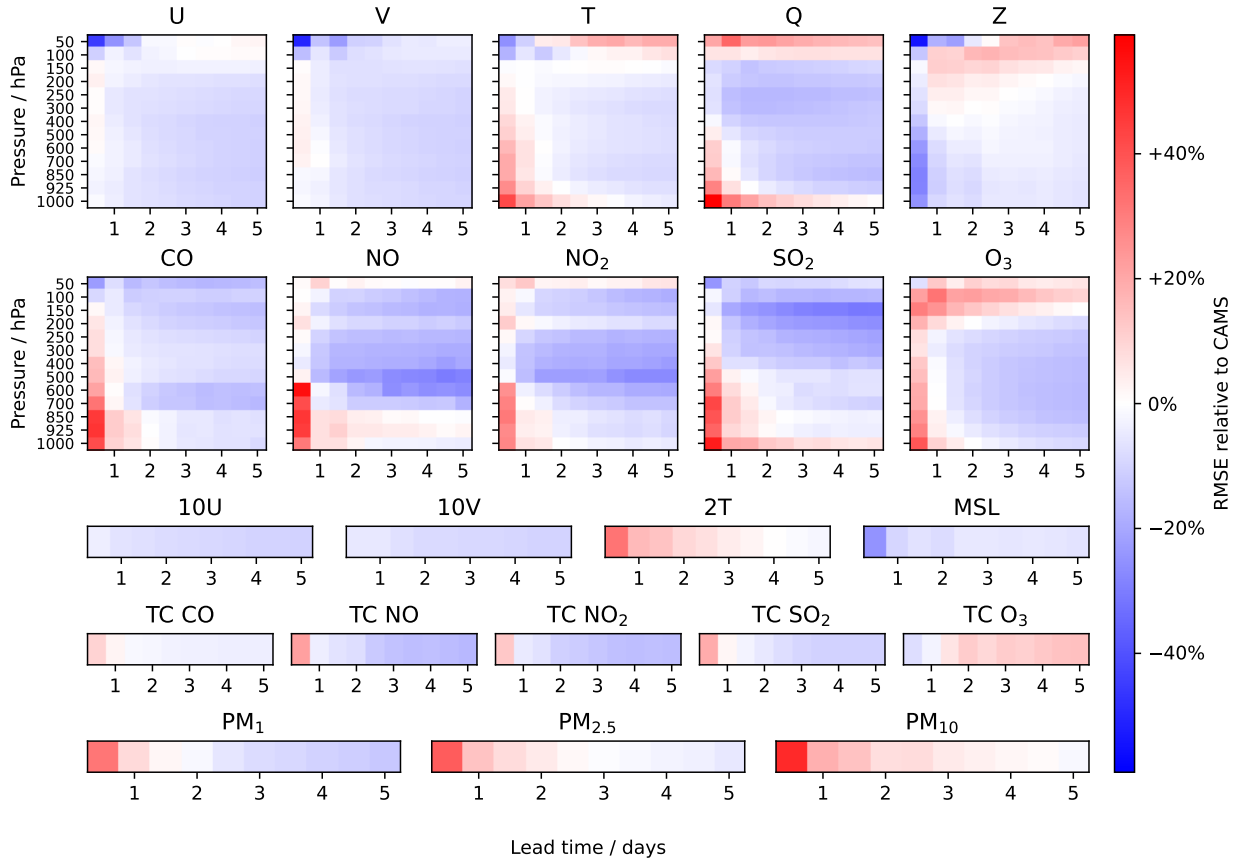

Figure 24: Latitude-weighted root mean square error (RMSE) of Aurora relative to CAMS, where negative values (blue) mean that Aurora is better. The RMSEs are computed over the period Jun 2022 to Nov 2022 inclusive. Aurora is competitive to CAMS (within 20% RMSE) on 96% of all targets, and Aurora matches or outperforms CAMS on 76% of all targets.

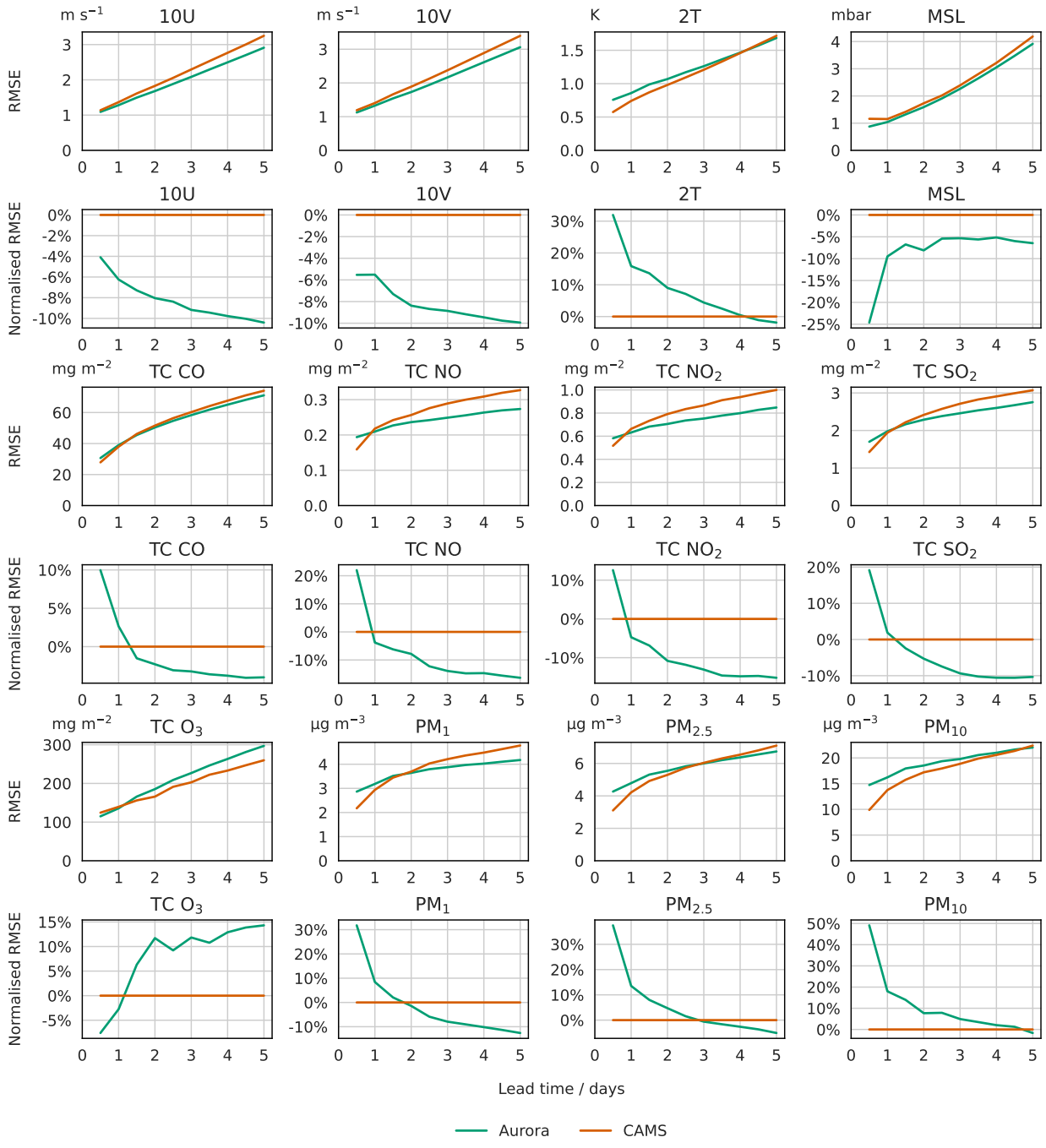

Figure 25: Latitude-weighted root mean square error (RMSE) of the surface-level variables compared to CAMS. The RMSEs are computed over the period Jun 2022 to Nov 2022 inclusive.

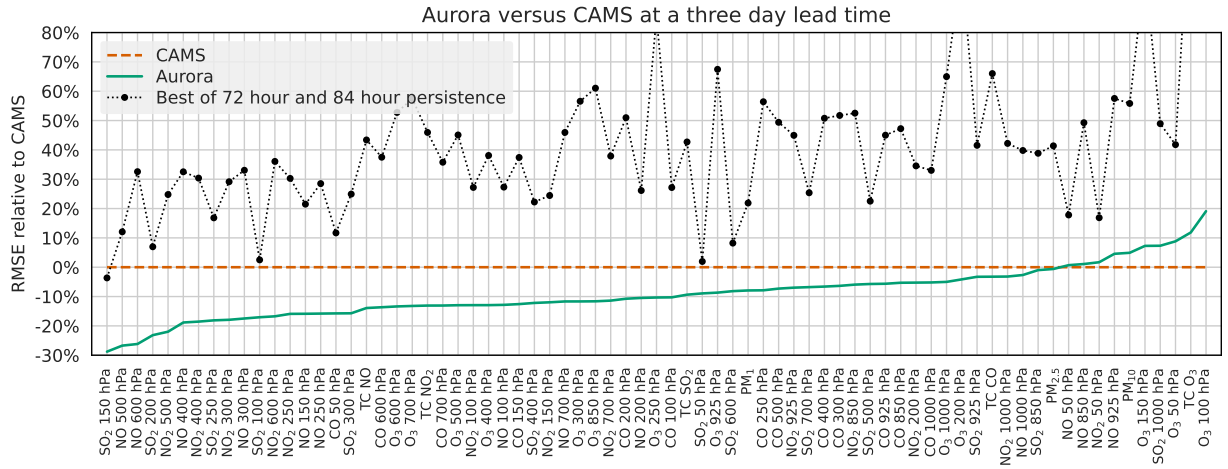

Figure 26: Latitude-weighted root mean square error (RMSE) of Aurora to CAMS at a three day lead time, where negative values mean that Aurora is better. Also shows the RMSE of the persistence prediction relative to CAMS. The RMSEs are computed over the period Jun 2022 to Nov 2022 inclusive. Aurora matches or outperforms CAMS on 86% of all variables and is strictly significantly better than the persistence prediction.

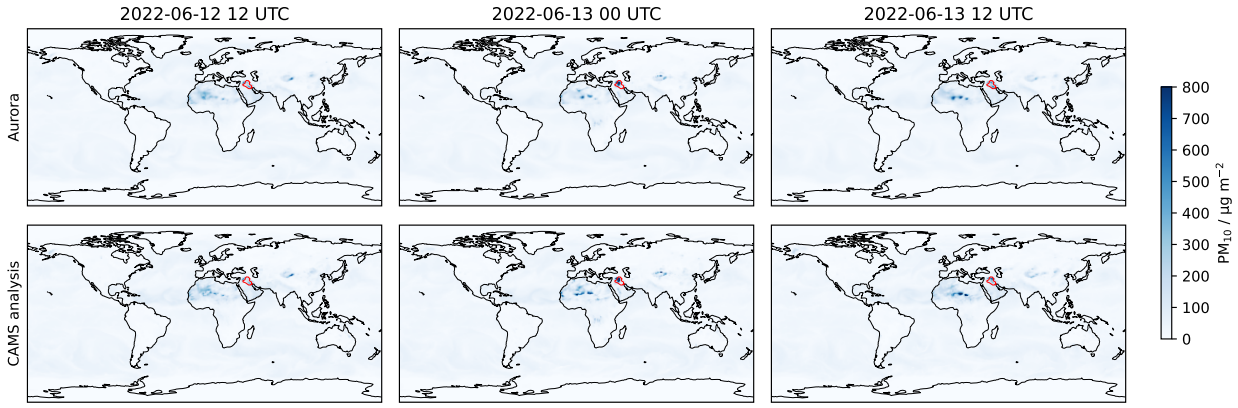

Figure 27: Predictions for particulate matter at $10\,\mu\text{m}$ (PM_{10}) by Aurora compared to CAMS analysis. Aurora was initialised with CAMS analysis at 12 Jun 2022 00 UTC. On 13 Jun 2023, Iraq (highlighted in red) was hit by a particularly bad sandstorm. Aurora predicts the sandbox a day in advance.

G.8 Power spectra

To assess the quality of predictions, a useful characteristic is the power spectrum. This section contains plots of power spectra for predictions and targets calculated using the method described in (Lam et al., 2023b).

At short wavelengths (*i.e.* high frequencies) Aurora's forecasts are uniformly higher in power than those of Graphcast, which means less blurring (Figure 28). In the mid-range of the spectrum, Aurora's forecasts have higher power than those of Graphcast for lead times of 1 and 2 days, but then drop in power below Graphcast's forecasts with 5 and 10 days lead times. In general, as more LoRA steps are added, the power in the forecast is reduced, especially in the mid-range of the spectrum (Figure 29). For the longest lead time of 10 days and with 40 LoRA steps, there is a clear reduction in power at the short wavelengths (*i.e.* high frequencies) which results in blurring.

When comparing the spectrum of Aurora between 0.25°and 0.1°, the power of the 0.1°forecasts is generally lower than the power of the 0.25°forecasts. However, the power of the 0.1°forecasts extends to shorter wavelengths (*i.e.* higher frequencies) than 0.25°forecasts, reflecting the higher resolution of the latter. These observations are in agreement with the discussion provided in Supplementary G.6, where we pointed out that the difference between Aurora forecasts with and without LoRA fine-tuning can be viewed analogously to the difference between the forecasts produced by the IFS ensemble mean and IFS HRES.

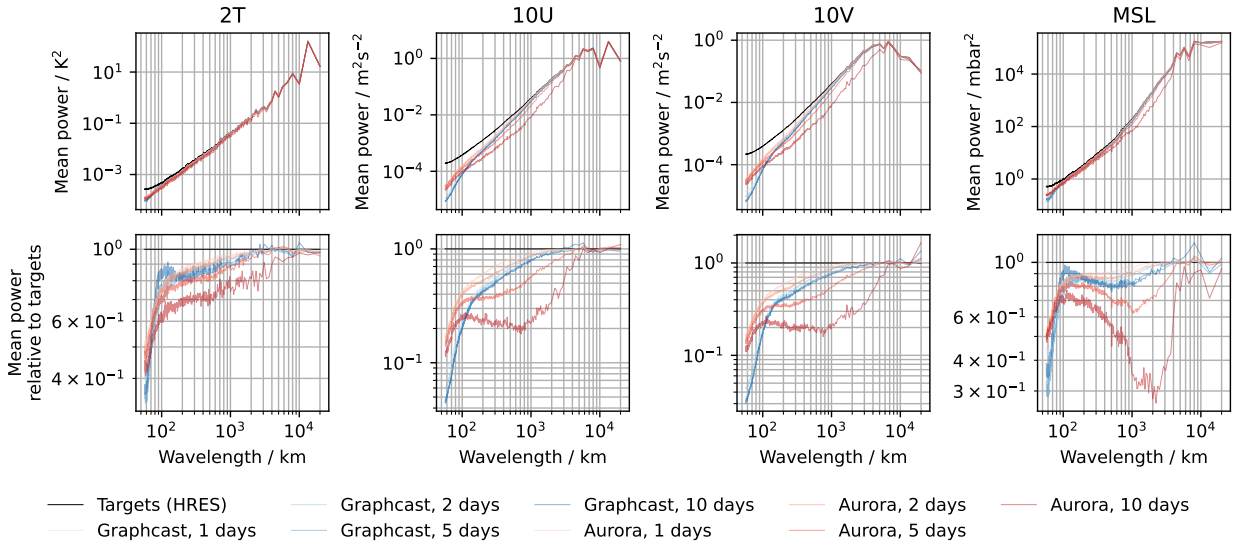

Figure 28: Power spectra of Aurora's predictions and Graphcast's predictions for 2022. The first row plots the mean power for predictions. The columns represent 2T, 10U, 10V, and MSL, respectively. The second row plots the mean power relative to the HRES targets. Each individual chart illustrates the change in the power spectrum for lead times of 1, 2, 5, and 10 days.

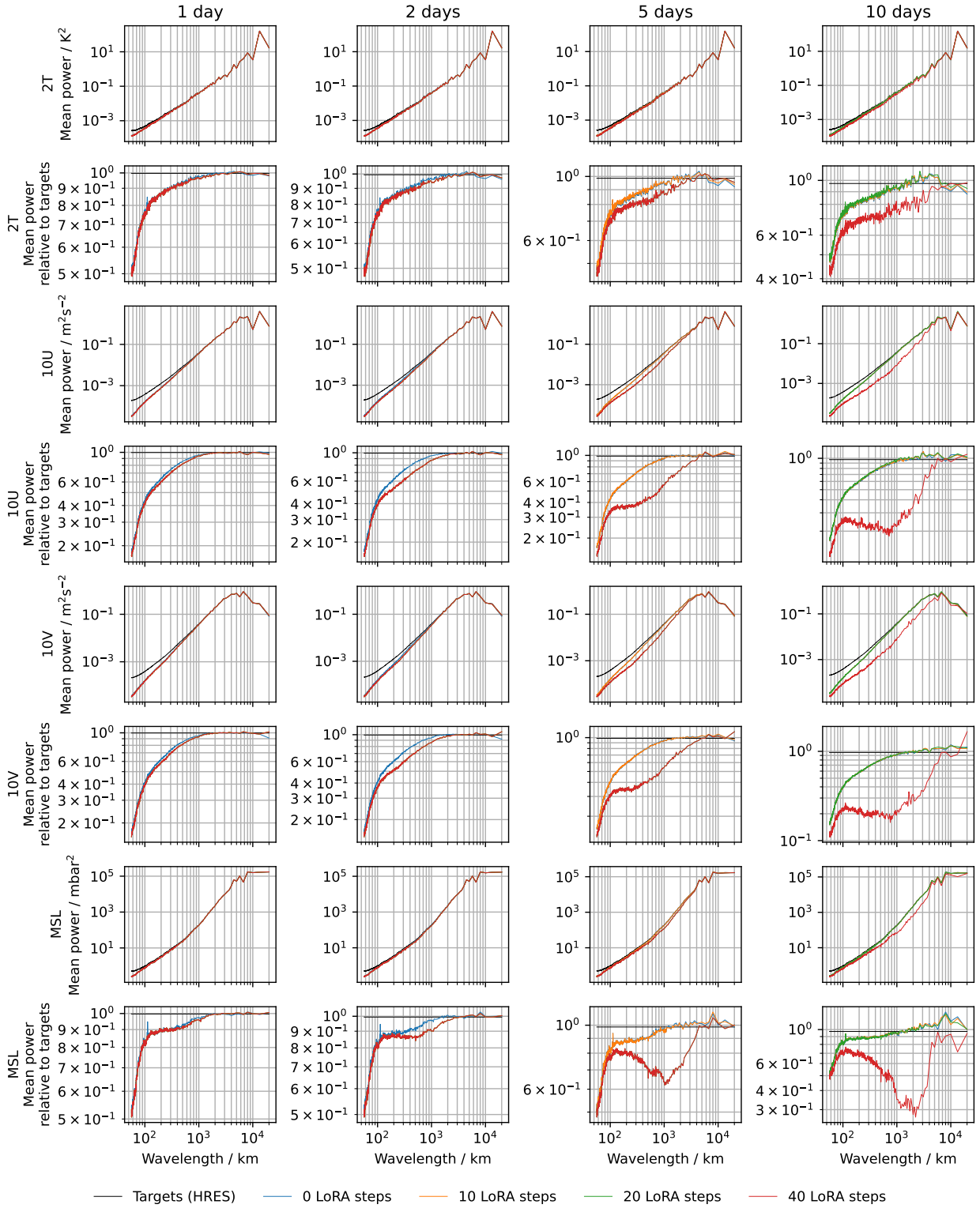

Figure 29: Power spectra of Aurora's predictions with different numbers of LoRA steps for 2022. The first row plots the mean power for 2T predictions. Each column represents a different lead time of 1, 2, 5, or 10 days, respectively. The second row plots the mean power for 2T relative to the HRES targets. Similarly, subsequent pairs of plots represent, respectively, the power and relative power of predictions of 10U, 10V, and MSL.

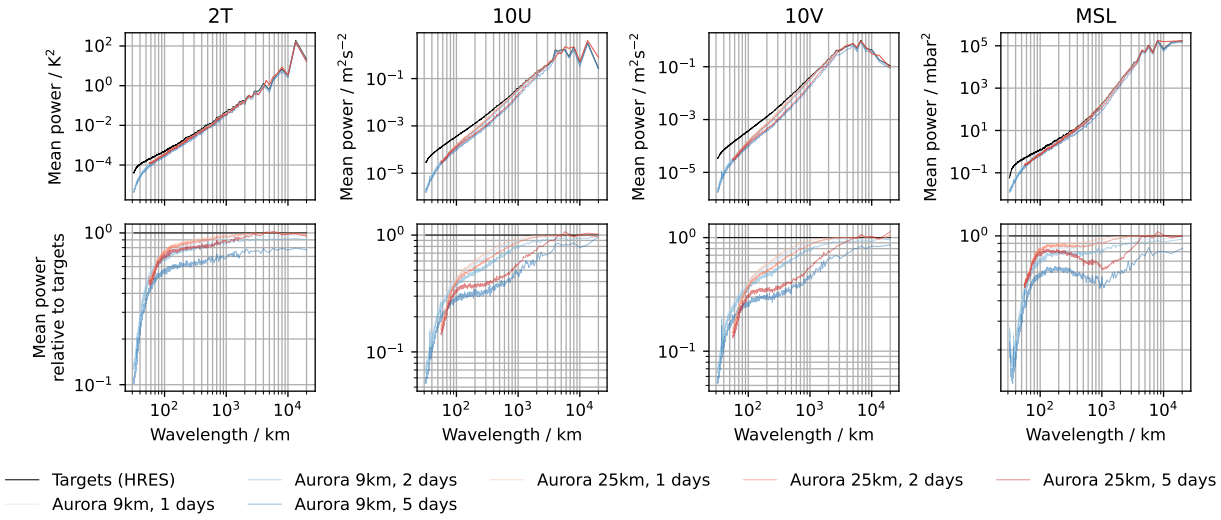

Figure 30: Power spectra of Aurora's predictions at both 9km and 25km resolutions for January 2022. The first row plots the mean power for predictions. The columns represent 2T, 10U, 10V, and MSL, respectively. The second row plots the mean power relative to the HRES targets.