

A Foundation Model for the Earth System

Corresponding Author: Dr Paris Perdikaris

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 1:

Reviewer comments:

Referee #1

(Remarks to the Author)

This paper presents Aurora, an AI foundation model for the atmosphere. In recent years, foundation models have emerged in the AI community and have been serving as an important building block for natural language processing, computer vision, and many other fields. A foundation model is a sufficiently large (or complex) neural network pre-trained on a large corpus of data, which can be used for various "downstream" purposes (probably after being fine-tuned) with superior performance compared to a model without any pre-training. This paper follows the above pipeline, e.g. Figure 1 shows a standard pre-training and fine-tuning procedure.

I will follow the guidelines for reviewers to evaluate this submission based on the key points below.

A. Summary of the key results

(a) Fundamental discovery or take-away. As said above, this paper shows that a foundation model can be trained for atmospheric purposes. All the proposed methods, including the pipeline (pre-training and fine-tuning), deep neural network (the Swin transformer with an encoder-decoder architecture), fine-tuning strategy (LoRA), etc., are existing and well-studied in the AI community. Specifically, from the prior work in the atmosphere, including ForeCastNet, Pangu-Weather, GraphCast, etc., it is widely known that this pipeline works well. So, this paper has delivered quite limited new knowledge to the community. The use of diverse training data is new, but I have some concerns regarding this point, which are described below.

(b) Model description. The pre-trained Aurora model has 1.3B parameters which is quite large compared to prior works. The training dataset is a mixture of multiple sources including the ERA5 that was widely used. The learning objective is still auto-regressive: the model is trained to predict the weather variables 6 hours in the future and the mean absolute error is the loss function. The training procedure takes 2.5 weeks in 32 NVIDIA-A100 GPUs.

1) The model shares part of architectural designs with Pangu-Weather, e.g. the use of 3D neural networks, Swin transformers, and the encoder-decoder architecture. However, each Pangu-Weather model has only 64M parameters, but the Aurora model has 1.3B parameters (20x). By the way, GraphCast, based on 2D neural networks, has about 37M parameters. I checked Appendix B and found that the main reason seems to lie in the number of embedding dimensions. Questions: Does the increase in dimension increase the overall training costs (if so, how heavy -- please report the training burden for each variant of the Aurora model)? Does the largest model fit the GPU memory of A100 (is it 80GB)? How does the smallest Aurora model compare against GraphCast? Does the increase of model parameters play a central role in learning, e.g. handling a mixture of multi-source datasets?

2) The model has used a mixture of training data, but the distribution of training data varies. In machine learning, it is common to mix a few data with different distributions (often called "domains" in machine learning), but it is based on an assumption that downstream tasks may span over multiple domains or be a combination of multiple domains. I am not sure if the atmospheric data falls in this case. For example, the Aurora model was pre-trained on both ERA and GFS data; when being tested on IFS data (close to ERA data), it is questionable why the addition of GFS data will benefit the accuracy. The authors are encouraged to discuss this topic because it is related to the fundamental logic (and one of the core contributions) of this work. There are sufficient experiments in the paper, but I am curious about the reason. As an AI researcher, I know something like "knowledge can be shared among different domains", but is there any evidence of shared knowledge or so? The authors are suggested to quantify "which knowledge has been shared" and "how much the knowledge has been

shared".

3) Although LoRA is a popular algorithm in AI, it was mostly used to tune large models (e.g. an LLM with tens of billions of parameters) to arrive at a tradeoff between accuracy and speed/costs. Although the Aurora model is large (1.3B) in the context of the atmosphere, it is quite small compared to LLMs. So, is there a need to use LoRA? Would full-parameter fine-tuning further improve the accuracy of downstream tasks?

(c) Results

The Aurora model was fine-tuned on a few downstream tasks, including the standard operational weather forecasting at 0.25 or 0.1 degrees and the prediction of other variables (e.g. chemistry and air pollution). However, the selection and validation of these tasks have raised my concerns below.

1) As a foundation model trained in an auto-regressive manner, the standard weather forecasting shall be placed in an important position. However, currently, this part is organized in Appendix G. I think the main article shall mention these results. Moreover, it was shown that Aurora outperforms GraphCast in most reported variables; I hope to know the reason -- there is little analysis: is this caused by more data, stronger architectures, or other technical tricks? I would also suggest adding a comparison against other AI models such as Pangu-Weather and ForeCastNet.

2) Regarding the resolution of forecasting (0.25 degrees or 0.1 degrees), it seems that the gain of Aurora-0.1 over IFS-0.1 is smaller than that of Aurora-0.25 over IFS-0.25 (there is no direct comparison of Aurora-0.25 vs IFS-0.25, but Aurora-0.25 is better than GraphCast-0.25 which is much better than IFS-0.25). I hope to know the reason. I am also curious what the result will be if one simply applies an image super-resolution algorithm to rescale the Aurora-0.25 forecasts into Aurora-0.1, in particular, how much gain is obtained by fine-tuning the Aurora model on 0.1-degree data compared to direct rescaling?

3) It seems that very few downstream tasks have been investigated. As a foundation model, Aurora shall be tested on more such tasks. Maybe the authors could refer to prior work such as ForeCastNet and GraphCast, or the technical report "The rise of data-driven weather forecasting" by the ECMWF, for more available tasks. In addition, I am curious why tropical cyclone (hurricane) tracking was not investigated, especially considering the submitted code contains a script for this.

4) The paper also mentioned the climate simulation data (for training and evaluation), but details (including data details) were not explained. Does CMIP6 contain the weather variables "assuming" that climate changes in some ways? Are these data of the same scale (e.g. 0.25-degree resolution, multiple pressure levels, etc.) as the common weather data like ERA5? How were these data generated? Can these data be considered kind of climate-aware perturbations beyond common weather data? In the AI community, there is a widely-used technique named data augmentation, which adds (mostly random) perturbations beyond training data towards better performance. The authors may hope to connect the use of climate simulation data with data augmentation.

B. Originality and significance: if not novel, please include a reference

(a) Originality. As said, the paper follows a well-known pipeline in the AI community. Although it is one of the first trials of foundation models in the atmosphere, I will not evaluate it to be of high originality or technical novelty. By the way, existing AI models for the atmosphere (e.g. FourCastNet, Pangu-Weather, GraphCast, etc.) can also be considered foundation models; Aurora went one step further by fine-tuning (using LoRA) towards downstream tasks, but LoRA itself is a well-known and well-studied method in AI.

(b) Significance. As said, experimental results are good but not surprising (i.e. good enough to support publication by themselves). Further analysis is also required to show how and how much the downstream tasks benefit from the pre-training data.

C. Data & methodology: validity of approach, quality of data, quality of presentation

(a) Validity of approach: the method is reasonable (actually, well-studied in the AI community); some experimental settings are weird (see the above).

(b) Quality of data: satisfactory; the algorithm is based on a large corpus of atmospheric data.

(c) Quality of presentation: satisfactory; this paper is easy to read. Some experiments shall be reordered, as said above.

D. Appropriate use of statistics and treatment of uncertainties

Satisfactory (compared to the existing AI models for the atmosphere).

E. Conclusions: robustness, validity, reliability

I am generally satisfied with the robustness, validity, and reliability. The method follows a well-known pipeline in AI and shows promising experimental results. However, the limited technical novelty and some concerns in the experimental validation (see the above) make it difficult to suggest acceptance of this submission.

F. Suggested improvements: experiments, data for possible revision

See the above comments.

G. References: appropriate credit to previous work?

Yes. To the best of my knowledge, recent works about AI for the atmosphere have been cited.

H. Clarity and context: lucidity of abstract/summary, appropriateness of abstract, introduction and conclusions

I am generally satisfied with the writing of this submission. The abstract shows a clear picture of the research. The method (adding Section 6) is clearly presented with sufficient details. Sufficient experiments are demonstrated. The conclusion is properly delivered without overclaiming the effect of the algorithm.

However, I strongly suggest that more details can be added to the description of the approach (Section 1). At least, the architecture of the trained neural network shall be briefly described in the main article, e.g. telling the reader how the model consists of 1.3 billion parameters (much larger than the previous models for weather forecasting).

(Remarks on code availability)

I scanned through the code. It was written in PyTorch. According to my expertise, the code offers sufficient details and instructions (in a README file) to run. I did not try to run the code myself.

Referee #2

(Remarks to the Author)

The authors present a new DL model that may serve as a foundation model in atmospheric sciences. Specifically, their contributions are:

1. A systematically motivated architecture based on a previous model ClimaX [1] that can process variable resolution as well as variable #inputs, carefully tuned training strategies and other optimizations
2. Pre-training on multiple datasets from weather/climate as well as a pollution dataset that helps the model learn from more data (and possibly from the additional data diversity)
3. Detailed validation of their model on different forecasting variables in weather, different pollutant concentrations
4. Demonstration of a promising scaling law w.r.t #parameters of the model, dataset size under a computational budget

Overall, the paper presents a detailed blueprint on building an atmospheric DL model (similar to the seminal GraphCast work [2]) that can be translated to some downstream tasks (a novel contribution). The paper is written well with lots of details in the supplementary and they also present interesting scientific results on the downstream tasks.

I have some major concerns and some minor concerns that I hope the authors will be able to clarify.

Major:

1. I think one of the main weaknesses of the paper is that it presents Aurora as a foundation model, but there seems to be only one non global weather downstream task. While I see a lot of value in Aurora as a systematically constructed DL weather/climate emulator, it is difficult to judge its value as a foundation model. My expectation is that the foundation model can be adapted to a wide range of downstream tasks that could be similar or very different to the pre-training. And especially in science, it may be possible to characterize this similarity (or out-of-distribution-ness/OOD-ness). Example, there could be so many different downstream climate model emulation tasks with different boundary conditions or under different warming scenarios or prescribed SSTs and Aurora could be used to adapt to each of these datasets and be quantified on its performance w.r.t the OODness. Similarly, the variable resolution/inputs could be stressed by focusing on harder (but critical) downstream tasks like regional forecasting (not global and hence stresses the geometry, more high resolution, more stochastic phenomena, more critical impact), extreme weather UQ with ensembles, etc. While the pollution example is very nice and appreciated, it does not seem sufficient to draw the conclusion of Aurora being a foundation model.

1.1 For OOD, it would be nice to show zero-shot performance of Aurora on pollution and other downstream datasets and motivate how much fine-tuning examples you need for each different task. A lot of building foundation model is to systematically quantify the data requirements for fine-tuning w.r.t how OOD the task is.

2. Another weakness that I see is the resolution of the weather forecasting. The authors say they do 0.1 deg, but in the Supp. it says the patch size is scaled up when the resolution goes up, effectively not changing the resolution of training. Why do the authors expect this to be useful? I also observe this in the spectral plots (say H.29) where there is not much additional power coming from the high resolution. I think the argument that the AI model does not fit on a GPU to run a true high resolution training is not strong since it is common for models to be trained with model parallelism today -- especially if the claim is that the model is a large foundation model.

3. Related to 2., do the authors feel the spatial resolution can be increased without similar increases in the temporal resolution? Here is where I think diversity in downstream tasks would really help (such as regional forecasting with HRRR, say) where process happen at a much faster time-scale leading to very high resolution features critical for events like storms. Then, we can really understand how a foundation model would handle different physics that is represented on different resolution.

4. I think there needs to be a short discussion on the computation costs or complexity w.r.t #parameters, #resolution, so that its clear how the model scales. Related to this, what is the cost of the variable level embeddings? It seems expensive to individually embed each level into a higher D and then back down to 1 (unless I misunderstood this). And also compare

hardware-aware training costs w.r.t GraphCast and other models since their performance is compared on accuracy. 5. I may have missed this but when the authors compare to GraphCast, do they compare to the 12 step (?) fine-tuned GraphCast? Or a 1-step GraphCast? If the former, I'm not sure the wind speed comparisons in (3d) are fair since rollout fine-tuning would smooth GraphCast(also seen in H.25 where LoRA Aurora performs poorer). Then (3d) should also show 1-step GraphCast. I also think in (3c), it would be nice to show Aurora not fine-tuned on 0.1deg data. In (3d) it would be good to also show the measured wind speed so all models can be compared to reality. Do the authors believe GraphCast fine-tuned on 0.1deg would match Aurora? To understand this, it would be good to show ERA5 wind speed on (3d) to see if a 25 km dataset could even capture the high speeds due to how big a grid cell would be there. The point here is to disentangle the architecture/model with information from the dataset.

Minor:

1. In (4b), what does the shading represent?
2. It might be interesting to also show a heat wave extreme as 2T seems to be differently behaved than the rest.
3. L137, it says the model is a ViT and not a Swin-Unet?
4. In parameter scaling law, it might be nice to also see how the scaling is w.r.t each variable rather than a averaged loss value. The latter may be fine for LLMs but in science there is so many more variables that are forecasted it could be valuable to see how each RMSEs scale.
5. Training is done for 150K steps - it would be good to add how many epochs that is, so it is clear how many times the dataset is passed over during training.
6. I'm not able to judge what a 40% error decrease compared to CAMS means in a realistic setting. It could be useful to show the CAMS prediction and Aurora prediction for this case so at least it's visually clear what the gains from Aurora are (are the losses in the cases where CAMS is better).
7. In H18, what is the red highlighted line exactly (not the box). Is it the sandstorm extent from observations? Why is it constant w.r.t time? And how are the authors concluding that the sandstorm was predicted a day in advance? I see high concentrations in other places as well. What is the quantity on that plot that tells you that it is a sandstorm?
8. Maybe this was just for me, but references to figures in the Supp. weren't working for me and was taking me somewhere else.

Once again, I really appreciate the engineering efforts and detailed validation errors in this work (reminiscent of the GraphCast [2] paper with large Supp.) and I think focusing on the science-aspect w.r.t downstream datasets would greatly strengthen this work. With regards to data, methodology, the paper seems sound to me, building systematically from previous efforts of ClimaX, GraphCast etc. with new datasets (novel). For treatment of uncertainties, the contribution is less because Aurora is a deterministic model and hence does not consider initial condition uncertainties, model uncertainties. General writing is clear, appropriate, and well-motivated.

References:

1. ClimaX: <https://arxiv.org/abs/2301.10343>
2. GraphCast: <https://arxiv.org/abs/2212.12794> or <https://www.science.org/doi/10.1126/science.adi2336>

(Remarks on code availability)

I have only partially reviewed the code by going through the files and READMEs and it seems to be very detailed with information on how to run the model. The repo has all the necessary PyTorch code for the different model components. I ran one of the quick tests on using the pretrained ckpt on a single GPU and it works (inference). The attached repo has a lot more scripts (for training, analysis, testing, and results generation) than the github link and it looks complete.

My assessment is that the code is a usable resource for the community, easy to install, up to the standards of open-sourced PyTorch code, and has detailed instructions on how to train and use Aurora.

Referee #3

(Remarks to the Author)

The authors develop Aurora, a foundation model for the atmosphere that is capable of generating forecasts for various atmospheric forecasting problems, including those with limited training data, heterogeneous variables, and extreme events. The model efficiently adapts to new atmospheric forecasting tasks by learning a general-purpose representation of atmospheric dynamics.

The paper is written and makes a significant contribution in atmospheric modeling. The results are quite impressive. The authors show that the Aurora foundation model generates i) 5-day air pollution forecasts at 0.4° spatial resolution, surpassing state-of-the-art atmospheric chemistry simulations on 74% of all targets at a significantly lower computational cost; ii) 10-day global weather forecasts at 0.1° resolution, surpassing state-of-the-art NWP models on 92% of all targets, with improved performance in extreme events; and iii) 10-day global weather forecasts at 0.25° resolution, surpassing state-of-the-art NWP models and GraphCast on over 91% of all targets. Aurora is scaled to 1.3 billion parameters and pretrained on over a million hours of diverse atmospheric data, which includes simulations from weather and climate models. The large-scale pretraining on diverse datasets improves forecasting accuracy compared to single-dataset training.

The comparison to state-of-the-art methods in the paper is not entirely satisfactory. Notably, the omission of ClimaX, a

relevant foundation model in this context, raises concerns. Although the input datasets for pretraining differ, it is feasible to evaluate Aurora on the same ERA5 dataset used by Climax and present the results. This would provide a fair comparison, given that Climax is a foundation model, whereas GraphCast, though high-performing, is trained and fine-tuned specifically on ERA5 data. The authors should consider comparing the accuracy metrics obtained on the ERA5 dataset with those reported by various methods in the WeatherBench benchmark. This will provide a more comprehensive assessment of Aurora's performance and help the community better understand how Aurora compares across multiple methodologies. Such a comparison will enhance the paper's contribution by placing Aurora's results within the broader landscape of existing AI models.

The generalization regarding Aurora's performance in predicting Storm Ciarán requires further experimentation. To ensure that Aurora's performance on extreme events and storms is generalizable, it is essential to conduct more extensive evaluations on a broader range of such events. Specifically, the authors should adopt a more rigorous approach by providing statistical significance through the evaluation of the model on multiple extreme events and storms. This would involve presenting a statement or conclusion that includes the sample size and confidence level, thereby offering a more robust and reliable assessment of Aurora's generalizability across different scenarios. Such analysis can allow the authors to strengthen their claims and provide the community with statistical confidence in the model's capabilities.

The paper presents empirical evidence demonstrating Aurora's superior performance over GraphCast. However, it is equally important to compare and contrast other critical aspects such as the volume of training data, training time, associated costs, resource requirements, and inference costs. While accuracy is crucial, these underlying factors play a pivotal role in evaluating the overall feasibility and practicality of the models. Understanding these dimensions will provide the community with a more comprehensive perspective on what is required to train similar models for other tasks. Although the authors can certainly justify the emphasis on accuracy and its potential impact, addressing these additional factors will enable readers to better assess the resources necessary for replicating or adapting these models to different applications.

It is unclear how the variability in the results is being assessed in the experiments. Even after fine-tuning when the model weights are fixed, variability can still occur during inference due to the inherent parallelism in GPU operations. This variability can stem from factors such as non-deterministic operations in floating-point calculations, race conditions in parallel threads, and other hardware-specific behaviors. These factors can lead to slight differences in the outputs, even when the model is provided with the same input multiple times. Therefore, a thorough analysis of this variability is required to ensure that the model's accuracy is consistently reliable and meets the reported accuracy across different runs.

In many settings, fine-tuning can significantly impact accuracy, sometimes even more so than pre-training. The paper, however, does not clearly establish that pre-training is the primary factor driving the model's performance, as opposed to fine-tuning. While the model benefits from pre-training on a large dataset, it still requires substantial fine-tuning to achieve optimal results. This raises questions about the relative contributions of pre-training and fine-tuning to the model's overall accuracy. Given the extensive pre-training involved, it would be valuable to assess the model's performance in zero-shot and few-shot fine-tuning scenarios. This would provide insights into how well the model leverages its pre-trained knowledge when faced with minimal additional training data, and whether fine-tuning plays a more crucial role than pre-training in achieving high accuracy. Such an analysis will offer a clear understanding of the interplay between pre-training and fine-tuning, helping the community to better gauge the model's strengths and limitations.

Robustness testing is essential for evaluating the Aurora model, particularly given that the input data originates from sensors, in particular for the extreme events, which are inherently prone to noise and variability. It is crucial to assess how the model's predictions and accuracy might be affected by the presence of noise in the data. Without such testing, it is difficult to assess the model's reliability and robustness in real-world scenarios where sensor data can be imperfect. Through systematic introduction of noise into the input data and analyzing the model's responses, the authors can provide a clearer understanding of how resilient Aurora is to variations and imperfections in the data. This will allow for validation of the model's applicability in diverse and potentially noisy environments but also offer insights into its limitations. Ensuring that Aurora maintains high accuracy despite noisy inputs is critical for its deployment in practical applications where data quality cannot always be guaranteed.

The authors mentioned that pre-training Aurora on multiple datasets enhances its prediction performance. However, the justification for selecting specific CMIP6 datasets for configurations C2 to C4 remains unclear. The authors should provide more detailed information about the datasets chosen for these carefully designed experiments, which demonstrate improved model performance through the inclusion of additional datasets.

In addition, in configuration C5, which includes a superset of all other configurations, the model performed worse than with ERA5. The authors hypothesize that the newly introduced datasets may not align well with the evaluation dataset. Is there any domain-specific analysis to support this hypothesis? Furthermore, from a general perspective, a large foundational model trained on abundant and diverse data is expected to enhance not only prediction accuracy but also generalization, leading to better performance on out-of-distribution (OOD) samples. However, the C5 configuration appears to show the opposite results. What could be the underlying reason for this discrepancy?

The Aurora model architecture appears to be largely inspired by the Pangu-Weather model, which also employs an encoder-decoder structure with downsampling-upsampling 3D Swin-Transformer blocks. The encoder-decoder layers used in Aurora are larger than those in Pangu-Weather, with notable differences including data variability, the addition of perceiver layers before and after the backbone, and the removal of positional bias. Given the similarities in the backbone, a broader comparison of results with the Pangu-Weather model would provide a better understanding of the impact of the

innovations introduced in this paper. Additionally, while the computational benefits of transitioning from plain ViT with dense attention to Swin-Transformer are evident, the overall impact on the solution remains unclear.

Additionally, the flexibility of the Aurora model in handling multiple resolutions is mentioned as a benefit. It would be helpful to explain this in terms of sequence length. Can the Aurora model handle different sequence lengths in the ViT backbone? For instance, when transitioning from a 0.25-degree resolution in pre-training to a 0.1-degree resolution in fine-tuning, the patch size increases to match the spatial resolution used during pre-training. Did this result in equal sequence lengths for both pre-training and fine-tuning? While this is described as a memory constraint issue, can the Aurora backbone manage different sequence length inputs, or did you find it more effective to fix the spatial resolution of the ViT backbone regardless of the actual data resolution? Additionally, when pre-training on the ERA5 and CMIP6 datasets, which have different resolutions, did you use the same patch size for both, or did you adjust it accordingly?

In the scaling studies, do the GPU hours on the x-axis correspond to pre-training? For the y-axis, which dataset is this validation loss derived from? It would be helpful to explore the effect of model size beyond just validation loss by showing the RMSE of some key atmospheric variables. Additionally, the statement regarding data loading being a bottleneck is not quite convincing. The file system of the cluster will have an impact, and this could vary depending on the model architecture and how much preprocessing occurs on the CPU versus the GPU.

(Remarks on code availability)

It was found that the code cannot be run locally because i) the necessary data is not available, and ii) it heavily relies on Microsoft Azure infrastructure. Although the README file mentions the use of nine heterogeneous pre-training datasets, no data links are provided for these datasets. Consequently, the pre-training or fine-tuning code cannot be executed.

Version 2:

Reviewer comments:

Referee #1

(Remarks to the Author)

I read the authors' rebuttal as well as other referees' reviews. Many of my concerns have been addressed. My opinion about this submission has not changed significantly: it is a technically solid paper that integrates state-of-the-art AI techniques for the atmosphere and shows that large-scale pre-training indeed brings benefits to various downstream tasks (e.g. TC tracking), but I am not 100% sure if it is a breakthrough that deserves being published at Nature. On the upside, the authors did a lot of hard work to organize training data from different sources, constructed a deep neural network with a much larger size (1.3B parameters), and trained the model carefully (Appendix D.1 shows full details), and will release the code and models to the community. This will advance the research on the atmosphere. On the downside, however, the technical contribution is limited -- Yes, I read the rebuttal about the contribution, which mostly emphasized the use of diverse data and state-of-the-art performance in 3-4 scenarios, but these facts do not dispel my concerns about *technical* contributions.

Below are a few minor questions that I hope/believe the authors can answer. However, addressing these concerns is not likely to overturn the overall justification for this submission.

1. The justification for the diverse training data.

I agree with most of the analyses except one. The authors said "Different datasets have varying spatial resolutions, numbers of pressure levels, and quality of physics models. By training on this heterogeneous mix, Aurora learns to extract the best features from each source, e.g., high-resolution details from some datasets, better vertical structure from others, and improved physical consistency from the most sophisticated models." I am rather confused about the statement. Why should we use atmospheric data in different resolutions rather than pre-processing them into the same resolution (if possible)? This is not like in computer vision where objects can appear in different resolutions and the training data shall be diverse to cover them all; the atmospheric system can have a fixed resolution (e.g. 0.1-deg) and every other resolution can align to it with rescaling. For a model with a fixed number of parameters (even 1.3B), it would be better to learn from single-resolution inputs and predict single-resolution output rather than learning from a mixture of multi-resolution data. I wonder if the authors can provide further insights about this point.

2. Model scaling law vs. data scaling law.

Aurora has 4 models of different sizes, but only the largest one has been thoroughly tested. According to Fig G4c, the smaller models are inferior to the largest one (1.3B). I wonder if the authors can test the smaller models on standard benchmarks and compare them to GraphCast or Pangu (preferably of similar parameter count) to validate (1) the Aurora model is stronger even in the same size, and more importantly, (2) the Aurora model benefits from the diverse data.

3. The accuracy of TC tracking.

The paper claims that Aurora reports state-of-the-art TC tracking accuracy, but I am not sure about this because the model was not compared against GraphCast or Pangu. The authors explained "The analysis of other large-scale global machine learning models [Lam et al., 2023, Pathak et al., 2022, Bi et al., 2023] has been limited to comparisons of tracks produced by

global models...", but at least the authors can make a comparison under the same test condition.

4. Other downstream tasks.

I have very limited expertise to judge these parts, especially on how the performance compares against the best methods in these individual fields. This concern is not for the authors to address, but for the editor's information.

(Remarks on code availability)

Referee #2

(Remarks to the Author)

Thank you very much to the authors for a detailed point-by-point response to my review, I really appreciate it. My assessment is that the revised manuscript is now stronger. Below are some comments regarding specific questions asked in the previous review:

1. Focus of the paper towards new downstream tasks: The authors have added an ocean wave dynamics task and demonstrated that Aurora can be fine-tuned on this new dataset and achieve good prediction accuracy compared to the numerical model. I believe this strengthens the contribution regarding the flexibility of the Aurora model on new tasks. The authors also introduced a tropical cyclone tracking analysis as a second downstream task. I believe other DL models (GraphCast etc) can also handle this task, given its zero-shot nature with inputs from weather variables. There could be a discussion on why Aurora is uniquely positioned to handle this task or achieve high performance. If not, then this should be clearly discussed. Further, why is Aurora 0.25deg used for this task and not 0.1deg (which one might hope is better)?
2. Few-shot performance of Aurora: The authors have shown that fine-tuned Aurora enjoys accuracy gains over training from scratch for the weather/climate tasks - I think this is a good, positive result. The authors should also add this for the ocean wave dynamics task and show that training from scratch is less advantageous than fine-tuning Aurora. What is the performance gain here from pre-trained Aurora? It is more important to show it for this task because the ocean wave dynamics task represents a downstream task different from global weather forecasting.
3. Effective resolution of Aurora: The authors added some new experiments that show the Aurora 0.1deg is able to capture more physics that 0.25deg cannot. While I am okay with the experiments, it still seems from the power spectra plot that Aurora 0.1deg doesn't seem to have a lot of power in the high wavenumbers compared to 0.25deg. My concern was this is because of the re-scaling of patch sizes in the higher resolution data to keep the sequence length the same. I am not entirely convinced by the answer, but am sympathetic to the fact that computational constraints (necessary model parallelism) prevent further analysis and it is easier to not have model parallelism. As long as this is discussed in the paper (which I believe it is), the response seems satisfactory for this paper.

All other answers were satisfactory and the revised manuscript is stronger as a result of the authors' revisions. My only new request is point (2) in showing the pre-trained vs training from scratch performance of Aurora on ocean wave dynamics task - I believe this will make the analysis more complete.

(Remarks on code availability)

No change in my review of the code from my original review. Many thanks to the authors for open-sourcing the code.

Referee #3

(Remarks to the Author)

The revised manuscript presents Aurora, a foundation model for the Earth system trained on diverse atmospheric and climate datasets. Aurora achieves state-of-the-art performance in several key areas of atmospheric forecasting, including air pollution prediction, tropical cyclone track forecasting, ocean wave prediction, and high-resolution weather forecasting. The model demonstrates significant computational efficiency compared to traditional numerical methods and other AI-based approaches. Notably, it introduces several innovations in model architecture, including a flexible 3D Swin Transformer and Perceiver-based encoders and decoders, enabling it to handle heterogeneous inputs across multiple domains. The revision is well-written, with clear motivation, comprehensive experimental evaluation, and detailed technical descriptions. It represents a substantial contribution to Earth system forecasting, especially in demonstrating how foundation models can surpass existing operational systems in multiple forecasting domains. The authors have also effectively addressed the concerns raised during the review process, which significantly strengthens the manuscript. Below, I provide an evaluation of the current manuscript based on the authors' responses and the updated paper. Aurora addresses key limitations of traditional models, such as high computational costs and inflexibility, while outperforming state-of-the-art NWP models and AI approaches across multiple domains. The inclusion of operational forecasting tasks, such as tropical cyclones and ocean waves, shows the practical applicability of Aurora.

The authors have provided extensive comparisons against GraphCast, Pangu-Weather, and operational IFS-HRES, demonstrating Aurora's accuracy in both accuracy and computational efficiency. The addition of new experiments on tropical cyclones and the evaluation of extreme events significantly improve the robustness and generalizability of the model.

Detailed discussions of pretraining and fine-tuning protocols, data selection rationale, and evaluation metrics are provided in the main text and appendices.

Clear explanations of architectural innovations (e.g., Perceiver layers, resolution-agnostic design) highlight the technical novelty of the work.

Responses to reviewer concerns:

The authors have addressed all reviewer comments comprehensively. The authors clarify that ClimaX, while relevant, operates at significantly lower resolutions and performs poorly compared to Aurora. They also justify their focus on IFS-HRES as a benchmark due to its operational relevance. A comparison with WeatherBench-2 is acknowledged but deemed less relevant for the high-resolution operational tasks Aurora is designed to address. The inclusion of tropical cyclone experiments and statistical analyses of extreme events provides convincing evidence of Aurora's robustness in predicting rare phenomena. New experiments quantifying the benefits of pretraining demonstrate significant performance improvements over training from scratch, particularly for data-scarce downstream tasks. The authors provide a detailed comparison of training and inference costs, demonstrating Aurora's efficiency relative to other models (e.g., GraphCast, Pangu-Weather). While Aurora primarily operates on clean numerical datasets, the authors justify the absence of robustness testing with noisy inputs, as such scenarios are outside the model's intended use case.

Minor suggestions for further improvement:

While the manuscript provides strong empirical results, further discussion on the interpretability of Aurora's predictions could enhance its impact. For example, how do Perceiver layers contribute to handling heterogeneous data, and can specific patterns learned by the model be linked to physical processes?

The manuscript mentions that Aurora's performance scales with data and model size. Including more details on potential computational trade-offs (e.g., diminishing returns with additional parameters or data) would be valuable for practitioners. An analysis of cases where Aurora underperforms (e.g., ozone prediction in the upper atmosphere) could provide insights into areas for further improvement.

The discussion section could expand on opportunities for extending Aurora's capabilities, such as incorporating ensemble forecasts or directly handling observational data for end-to-end weather prediction.

Recommendation: Based on the manuscript's strengths, the comprehensive responses to reviewer feedback, and the additional experiments presented, I recommend acceptance after minor revisions. Addressing the suggestions above will further solidify the manuscript's contribution and broaden its impact on the Earth system modeling community.

(Remarks on code availability)

Thanks for making the code available in github. It is easier to review and run.

credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>