ATML Report: Replicating ClimODE

Bhardwaj Lovnesh *bhardl@usi.ch*
Gobbetti Alessandro *gobbea@usi.ch*
Onoprishvili Tornike *onoprt@usi.ch*
Varese Lorenzo *varesl@usi.ch*

Abstract

High-confidence climate modeling requires complex numerical simulations of physical conservation equations. ClimODE, a neural ordinary differential equation (ODE) model, combines a convolutional local mechanism and a global attention mechanism to predict one time-step of weather evolution. Unlike traditional deep learning methods, ClimODE incorporates principles of advection from statistical mechanics, modeling weather changes as the spatial movement of quantities over time. This approach facilitates value-conserving dynamics and enables uncertainty quantification, outperforming existing methods in global, regional, and monthly forecasting with significantly fewer parameters. In this replicability challenge, we detail our efforts to independently replicate the claims and results of ClimODE. Specifically, we aimed to reproduce the outcomes in global, regional, and monthly weather predictions while gaining a critical and comprehensive understanding of the model's mechanisms.

1 Introduction

Climate and weather modeling rely on solving the *continuity equation* Broome & Ridenour (2014):

$$\frac{du}{dt} + \boldsymbol{v} \cdot \nabla u + u \nabla \cdot \boldsymbol{v} = s$$

The solution to this equation, $u(x,t)$, represents the quantity of interest at any given location x and time t. These quantities include ground temperature (t2m), atmospheric temperature (t), geopotential (z), and ground wind vectors (u10, v10). Traditionally, solving this differential equation required computationally intensive simulations performed on high-performance computing (HPC) clusters.

ClimODE Verma et al. (2024) offers an alternative approach by parameterizing $u(x,t)$ with a trainable model $f_\theta(x,t|\theta) = \dot{v}(x,t)$. Here, f_θ essentially models the rate of change of $v(x,t)$. To produce numerical estimates at a specific point in time t and space x, the outputs of f_θ must be integrated up to the desired point.

2 Scope of reproducibility

This section outlines the main claims presented in the original paper, along with their specific experimental settings. The claims serve as the foundation for our reproducibility study, as each of them can be supported or refuted based on the experimental evidence provided in Section 4.

- **Global Forecasting Claim:** *"ClimODE possesses superior performance across all metrics and variables over other neural baselines, while falling short against the gold-standard IFS, as expected."* The main objective for this claim is to replicate Table 6 (Appendix F) of the original paper, which compares latitude-weighted RMSE (↓) and ACC (↑) metrics on the global ERA5 dataset.
- **Regional Forecasting Claim:** *"ClimODE outperforms other competing methods in t2m, t, z and achieves competitive performance on u10, v10 across all regions."*

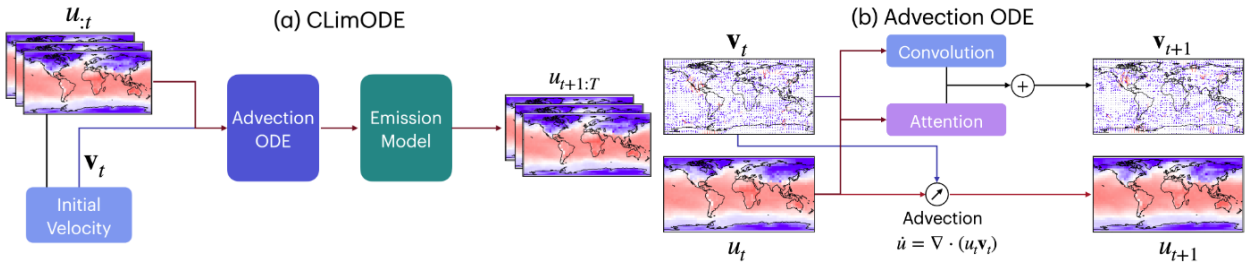

Figure 1: Model overview from the original paper.

The focus is on replicating Table 2 in the original paper to evaluate performance on regional forecasting across North America, South America, and Australia.

- **Monthly Forecasting Claim:** *"ClimODE demonstrates significantly improved monthly predictions as compared to FourCastNet, showing efficacy in climate forecasting."*
  The aim is to replicate Figure 5 from the original paper to validate that ClimODE exhibits a significant reduction in RMSE (↓) compared to FourCastNet (FCN) for monthly forecasting.

- **Longer Lead Time Prediction Claim:** *"ClimODE is remarkably stable over long predictions but has lower performance. Our method achieves better performance as compared to ClimaX for longer horizon predictions."*
  The goal is to replicate Table 7 in the original paper to ensure consistency in ClimODE's performance for longer horizon predictions and to validate a fair comparison with ClimaX.

Each of these claims will be tested through rigorous experimentation, and the results will be compared with the benchmarks established in the original paper.

3 Methodology

In this reproducibility study, we reused the original authors' code, which is publicly available on GitHub at Aalto-QuML/ClimODE. However, the repository presented several challenges due to its poor maintenance and lack of documentation. To overcome these limitations, we created a new repository at tornikeo/climode-reproducibility and began modifying and improving the code.

The paper reported using an NVIDIA V100 with 40 GB VRAM for their experiments. Since this hardware was not accessible to us, we rented an NVIDIA RTX 4090 with 24 GB VRAM from a GPU rental service.

To manage the division of tasks efficiently, we split the work based on claims or partial claims. For instance, tasks related to global forecasting were assigned to one member, while another focused on monthly forecasting. Regional forecasting was further divided among team members, allowing each person to adopt their preferred service for conducting experiments. Despite these individualized setups, we ensured consistent results by maintaining a unified experimental framework and systematically documenting all modifications in the forked repository.

3.1 Model descriptions

Model is a 2,75M-parameter convolutional neural network with a distinct, Y-shaped topology.

Specifically, the model takes as input u and v together, in a stacked tensor of shape $(N_{years}, 1, C_{channels}, H_{longitude}, W_{latitude})$. In the repository, these shapes are taken as $(10, 1, 5, 32, 64)$. This tensor is denoted as the red arrow in part b of Figure 1. As noted, this multidimensional tensor contains data from both v and u at this time step.

Notice that this tensor doesn't contain the common batch size dimension. Instead the batch size B_{batch} is contained in the *time* query. Time query is simply a list of size of B_{batch} of required prediction times in the future.

Next, the stacked tensors and times are input into the two convolutional networks, f_{conv} and f_{att}. Both networks use the same convolutional topology and input-output shapes as listed in Table 1. However, the attention network additionally uses a standard QKV self-attention layer at the very end of Conv2D layers. Specifically f_{att} uses three Conv2D sequential stacks to form query, key and value tensors later used to perform the QKV attention.

Table 1: Default hyperparameters for the convolution network f_{conv}

Hyperparameter	Meaning	Value
Padding size	Padding size of each convolution layer	1
Padding type	Padding mode of each convolution layer	X: Circular, Y: Reflection
Kernel size	Kernel size of each convolution layer	3
Stride	Stride of each convolution layer	1
Residual blocks	Number of residual blocks	[5,3,2]
Hidden dimension	Number of output channels of each residual block	$[128, 64, \text{out channels}]$
Dropout	Dropout rate	0.1

Finally, the outputs of the two networks f_{att} and f_{conv} are averaged with a learnable weighting parameter γ. This element-wise addition is possible as the two networks have same output shape.

The uncertainty of the predictions is predicted by a third network f_{uncert}, which outputs a value which is interpreted as standard deviation. The f_{uncert} is a plain Conv2D network, with same topology as f_{conv}.

Model outputs are shaped similar to the inputs. Concretely, the output is a tensor of shape $(6, 10, 5, 32, 64)$, where the leading 6 now denotes the batch size. The loss is calculated using `torch.distributions`, as negative log-likelihood of the true predictions. Model predictions are penalized for both incorrect and uncertain (high-std) estimates. The L_2 norm of all of the trainable network weights is also added to the loss with a weight of 0.001.

Authors use AdamW optimizer from Loshchilov & Hutter (2017) with learning rate of 0.0005 and a cosine annealing learning rate. While the authors use a default batch size of 8, we only use a batch size of 6, since this is the most a 24GB VRAM RTX4090 can hold.

3.2 Datasets

We used the ERA5 dataset with 5.625° resolution provided by WeatherBench Rasp et al. (2020), a standard benchmark dataset and evaluation framework for data-driven weather forecasting models. WeatherBench regrids the original ERA5 data, which is available at 0.25° resolution, into three coarser resolutions: 5.625°, 2.8125°, and 1.40625°. We utilized the 5.625° resolution dataset, as specified in the original paper.

The ERA5 data contains meteorological variables, including temperature, wind speed, and geopotential, over multiple spatial and temporal scales. To ensure reproducibility and ease of access, we hosted it publicly on the Hugging Face Hub. This dataset ensures consistency with the benchmark setup while facilitating reproducibility for future studies. For more details on the raw ERA5 data, refer to the official documentation.

3.3 Hyperparameters

Given that the primary objective of this report was to replicate the exact performance claims of ClimODE, our exploration of the hyperparameter space was intentionally conservative. We modified only the essential hyperparameters to validate the reported results. Hyperparameter tuning was conducted using the Ray Tune library, which integrates seamlessly with PyTorch for hyperparameter optimization.

Specifically, the following adjustments were made:

- The batch size was reduced from 8 to 6 due to hardware constraints. While the original authors utilized a GPU with 40GB VRAM, our experiments were conducted on an RTX 4090, which has 24GB of VRAM.

- A learning rate search space was defined with uniformly sampled values ranging from 1×10^{-5} to 1×10^{-3}. A total of 10 trials were conducted, revealing that a learning rate of 2.0511×10^{-5}, combined with a Cosine Learning Rate Scheduler, provided the best model fit. Further details, including the corresponding dashboard, are provided in Appendix A.

3.4 Experimental setup and code

For our experiments, we rented an instance on the GPU rental provider Vast.ai. As mentioned in Section 3, the division of tasks across team members led to the use of different computational resources. Here, we provide a detailed description of one experimental setup to guide replication efforts.

After renting a GPU instance on Vast.ai, we accessed it via SSH, enabling both command-line interactions and a front-end experience using Jupyter Notebook or Visual Studio Code's remote SSH capabilities. The first step involved cloning our forked repository, `replication-climode`, recursively, which provided access to both our modified code and the original authors' codebase.

To prepare the dataset, we utilized the `download.py` script located in the `scripts` folder. This script downloads the preprocessed ERA5 dataset from Hugging Face and places it in the appropriate directory within the ClimODE repository. This ensures all files are ready for use in the experiments.

The software environment was set up using Conda to install all required dependencies. These dependencies are documented in a script named `install.sh`, located in the `scripts` folder.

We documented the entire process, including step-by-step instructions, to streamline the setup for other group members and to ensure future reproducibility. The documentation is publicly available in the repository notes under the name on_vastai.md. Our replication repository, which includes all scripts and documentation, can be accessed at GitHub repository.

3.5 Computational requirements

All experiments were conducted on hardware rented from Vast.ai. Each instance utilized a single NVIDIA RTX 4090 GPU with 24 GB of VRAM, supported by 32 AMD EPYC 7T83 CPU cores and 129 GB of RAM. The dataset and experimental setups required up to 50 GB of SSD disk space in total.

The original implementation of the code did not include any logging or tracking of computational requirements. To address this, we instrumented the code by integrating the Weight & Biases service. This allowed us to track not only hardware usage statistics but also the results of our experiments. Through this integration, it is possible to identify the parameters used in each run (available under the 'Overview' section) and monitor system metrics throughout training (on 'Workspace' section). Comprehensive links to the Weight & Biases dashboards for each experiment are provided in the Appendix A.

4 Results

Overall, we were able to replicate all the claims that we aimed to investigate from the original paper. The results are generally consistent with those reported, supporting the majority of the claims, including those on global forecasting, regional forecasting, and the conservation of mass.

An important observation arose during the replication of the monthly forecasting results. While our results demonstrated significant improvements compared to other methods, the exact data related on RMSE as reported in the original paper was not fully accessible for the comparison.

Regarding longer lead time predictions, we were only partially able to replicate the results due to incomplete instructions on how the model should be extended for longer lead times. The adjustments we implemented to enable longer horizon predictions did not confirm the claim of ClimODE's superiority over ClimaX. Instead, our results showed competitive performance but fell short of achieving a clear advantage.

4.1 Results reproducing original paper

In the following subsections, we provide a detailed discussion of the experimental results grouped by the claims outlined in Section 2. Each subsection explains whether the experiments successfully reproduced the associated results from the original paper, offering factual insights based on our findings. For further analysis and reflections, consult the discussion points in Section 5.

4.1.1 Result 1: Global Forecasting

We find that the global forecasting performance reported by ClimODE is replicable to a satisfactory extent. Among the variables, geopotential height (z) and temperature (t) show the highest consistency, while $t2m$, $u10$, and $v10$ exhibit relatively larger, albeit positive, deviations from the original authors' claims. Detailed performance comparisons can be found in Table 2, which is provided in the appendix.

4.1.2 Result 2: Regional Forecasting

This experiment supports the regional forecasting claim outlined in Section 2, which states that *"ClimODE outperforms other competing methods in t2m, t, z and achieves competitive performance on u10, v10 across all regions."* Using our replication experiments, we successfully reproduced the results of the original paper to a significant extent.

As shown in Table 4, the replicated results closely align with the original findings in terms of outperforming baselines across North America, South America, and Australia. ClimODE consistently achieved superior performance for z, t, and $t2m$ metrics, demonstrating robustness across all regions. Additionally, for $u10$ and $v10$, our replication confirmed that ClimODE maintains competitive performance. Minor discrepancies in RMSE values, particularly for certain metrics and regions (e.g., $t2m$ in South America), suggest that slight differences in the experimental setup might account for the variation. Overall, the replication validates the claim that ClimODE is highly effective for regional forecasting tasks.

4.1.3 Result 3: Monthly Forecasting

The results for monthly forecasting, as replicated using ClimODE, are presented in Table 5 and Figure 4. This experiment supports the claim in Section 2 that ClimODE can forecast key observables with varying lead times. Our replication aimed to reproduce the original paper's results, focusing on the root mean square deviation (RMSD) metric.

Unlike the original paper, which presented results solely through plots, our tabular format provides a clearer numerical comparison. Notably, the original paper's plots lack a defined y-axis scale, complicating direct comparisons.

Our findings show that RMSD increases with lead time across all observables, consistent with expected performance degradation over longer horizons. For ACC, we observed moderate performance for shorter lead times (e.g., z, t, $t2m$), but accuracy decreased significantly for longer lead times and certain variables (e.g., $v10$ at 3 months).

4.1.4 Result 4: Longer Lead Time Prediction

This experiment supports the **Longer Lead Time Prediction Claim** in Section 2, which states that *"ClimODE is remarkably stable over long predictions but has lower performance. Our method achieves better performance as compared to ClimaX for longer horizon predictions."* The goal was to replicate Table 7 from the original paper, ensuring consistency in ClimODE's performance for long lead time predictions and validating the comparison with ClimaX.

The results, as shown in Table 3, indicate partial reproduction of ClimODE's reported performance. Our replicate demonstrates similar trends but generally underperforms compared to the original results. For instance, RMSD values for variables such as z and $t2m$ are notably higher in the replicate, suggesting potential discrepancies in the experimental setup or preprocessing. Similarly, ACC values exhibit a significant drop for variables $u10$ and $v10$ in the replicate compared to the original ClimODE results.

We observed a critical issue with lead time handling, where batch size is used as a proxy for lead time. For example, a batch size of 6 corresponds to a 36-hour forward prediction using a 6-hour resampling scheme. Due to memory constraints, we modified the sampling strategy to daily, which allowed us to complete the experiment. This adjustment likely influenced the results, and further clarification on the intended methodology for handling longer lead times is needed to ensure an accurate comparison.

4.2 Results beyond original paper

4.2.1 Addressing Batch Size and Memory Constraints in Lead Time Predictions

One challenge we encountered during the replication of the experiments was related to the relationship between batch size and lead time prediction, particularly for longer lead time studies. In the original setup, batch size was effectively used as a proxy for lead time; for example, a batch size of 6 corresponds to a forward prediction of 36 hours with a 6-hour resampling scheme.

To overcome memory limitations associated with longer lead times, we explored adjustments to the sampling strategy. Specifically, we modified the sampling interval from a 6-hour resampling scheme to a daily scheme. This adjustment reduced memory requirements significantly and allowed us to successfully run the experiments for longer lead time predictions.

While this modification enabled us to obtain results, it introduces a potential trade-off in temporal resolution. Further insights into the feasibility and implications of this adjustment, including whether it aligns with the original intent of the experiments, would be valuable for refining this approach.

4.2.2 Additional Logging Capabilities Aid in Tracking Model Training

The original implementation provided by the authors lacked comprehensive logging capabilities, as mentioned in 3.5. To address this, we integrated a Weights & Biases (wandb) logger into the training phase for all models. This enhancement allowed us to monitor key metrics and detect issues such as gradient explosion, which we discuss further in Section 5.2.

During the training of the regional models, we observed an interesting behavior in the loss progression. Although the authors specified 300 epochs for training, our logs revealed a local minimum in the loss at around epoch 170. Beyond this point, the loss began to worsen, reaching a new global minimum at epoch 290. Despite the eventual improvement, the performance gains did not justify doubling the training duration in many cases. This behavior was consistent across all regions, suggesting an important consideration for experiments with constrained budgets. Early stopping based on loss tracking could be an efficient strategy to optimize resource usage without significant loss in performance.

5 Discussion

Our experimental results indicate that the authors' claims about the performance of ClimODE are mostly valid and, with sufficient effort, replicable. While we were able to reproduce the majority of the claims, some discrepancies and challenges emerged, particularly in replicating monthly forecasting and longer lead-time predictions. This section discusses the strengths and weaknesses of our replication effort and evaluates the accessibility and reproducibility of the original work.

5.1 What was easy

The easiest aspect of this replication was reading and understanding the paper. The authors excelled in scientific writing, providing clear and concise descriptions of the methodology and presenting the fundamental

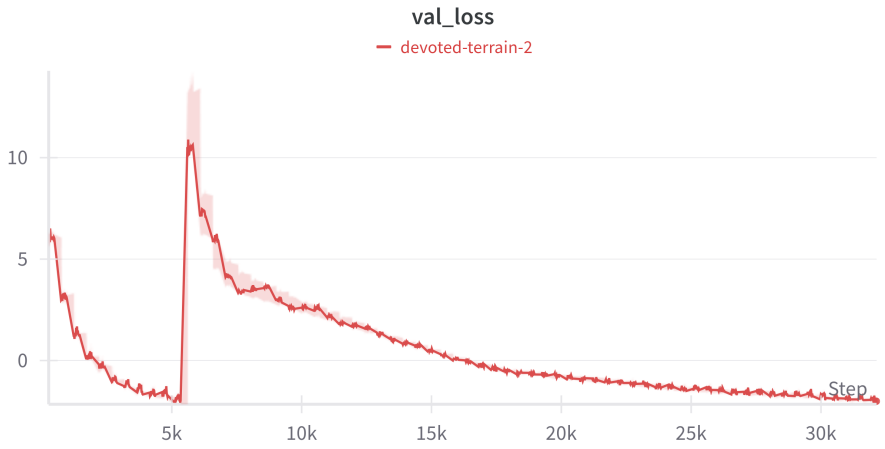

Figure 2: ClimODE exhibits a near-total gradient explosion during training at approximately the 6,000th update step.

equations in an accessible manner. The logical flow of the paper made it straightforward to understand the problem being addressed and the proposed solution. For an applied machine learning paper, this level of clarity is a significant strength.

Regarding the code, while certain aspects of the repository lacked comprehensive documentation, the provided instructions were sufficient for key initial steps. Specifically, the guidance on sourcing and organizing the climate dataset was clear, and the installation of dependencies was mostly painless.

5.2 What was difficult

One of the most challenging aspects of this project was understanding and debugging the existing codebase. The primary reason for these difficulties was the poor design and organization of the repository, which significantly hindered its usability and comprehensibility. Specific examples of programming practices that contributed to these challenges include:

- The use of wildcard imports (e.g., `from file import *`), which obfuscate dependencies and make the code difficult to follow.
- Functions with excessively large numbers of arguments (over 10) and outputs, complicating their readability and maintainability.
- The complete absence of docstrings or inline comments to explain the functionality of the code.
- Confusing and inconsistent use of standard Python constructs, leading to further ambiguity in implementation.

In addition, we encountered an undocumented instability in the training process. Specifically, by the 100th epoch, the model experienced a near-gradient explosion, as depicted in Figure 2. This issue caused the model to regress to a near-random state before restarting the training process, significantly impacting reproducibility and stability.

Furthermore, the overall rigidity and lack of testing in the codebase posed significant obstacles. Modifying key training and validation parameters, such as batch size, floating-point precision, or training device, frequently resulted in a cascade of errors, making it extremely difficult to adapt the model for various configurations.

These issues highlight the urgent need for more robust testing, documentation, and validation practices in machine learning research. Improving the quality and reliability of such implementations is essential to ensuring their usability and reproducibility for future studies.

5.3 Communication with original authors

During our reproducibility study of ClimODE, we encountered specific challenges related to the fair comparison of monthly predictions and the correct approach for longer lead time experiments. To address these issues, we contacted the authors via email, seeking clarification on the following two aspects. (1) The missing y-axis scale in the original paper's Figure 5, which complicates direct comparisons for monthly predictions. We requested the expected values for all variables to ensure consistency. (2) The use of batch size as a proxy for lead time in longer lead time experiments. We asked for confirmation of this methodology and whether adjustments, such as daily sampling, are valid to reduce memory requirements.

We have relied extensively on the resources provided in the `Aalto-QuML/ClimODE` GitHub repository and the supplementary appendix of the original paper to guide our replication process. Despite this, certain ambiguities remain unresolved, and we are awaiting a response from the authors to proceed confidently with the aforementioned parts of our analysis.

6 Conclusion

In this report, we detailed our efforts to replicate the results of Verma et al. (2024). We introduced the settings and the mathematical climate model on which ClimODE is based, and defined the precise scope of our reproducibility study according to the claims in the original paper. A detailed mechanistic description of ClimODE, including exact tensor shapes and specifics of the training dataset, was provided. To facilitate easier replication of our work, a snapshot of the climate dataset was made freely available on the Huggingface Hub. Additionally, we documented the necessary hardware and software dependencies and shared a simple guide to our open-source repository.

Due to time constraints, we were unable to fully document and organize the code according to best coding practices. Therefore, future efforts to replicate and build upon ClimODE's climate modeling should focus on developing a well-documented, test-driven Python repository from scratch, rather than attempting to adapt the existing codebase.

Member contributions

Our group collaborated effectively to ensure a balanced distribution of tasks and responsibilities, allowing us to maximize each member's strengths. Each of us contributed approximately $20 on Vast.ai credits, totaling $80 for renting the necessary instances to run the experiments. Below is a detailed account of individual contributions:

- Tornike Onoprishvili: Led the initial setup of the repository, developed a comprehensive guide for replicating the experiments on Vast.ai, conducted experiments on global forecasting, and ensured the repository was clean and well-organized prior to submission.

- Alessandro Gobbetti: Focused on experimentation for monthly and regional forecasting, with particular attention to experiments for North and South America. He resolved critical shape mismatch issues in the regional model and contributed to improving the overall performance.

- Lorenzo Varese: Conducted experimentation for regional forecasting on Australia and collaborated with Alessandro Gobbetti on debugging the regional model. He also replicated the monthly forecasting results to ensure fairness in comparison, given missing details in the original paper. Lorenzo made a significant contribution to writing the report.

- Lovnesh Bhardwaj: Provided theoretical insights into the model, leveraging prior experience in the field. He tackled the challenging aspect of longer-time predictions, requiring substantial code adjustments to run the simulation effectively. Lovnesh also introduced hyperparameter tuning to extend the search space explored by the original authors.

References

S. P. Broome and Jonathan Ridenour. A pde perspective on climate modeling. 2014. URL https://api.semanticscholar.org/CorpusID:127183516.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL https://api.semanticscholar.org/CorpusID:53592270.

Stephan Rasp, Peter Düben, Sebastian Scher, Jonathan 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 2020. doi: 10.1029/2020MS002203.

Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xuY33XhEGR.

A Weight & Biases Dashboards

These Weight & Biases dashboards provide detailed visualizations of training metrics, system resource utilization, and parameter configurations, facilitating the understanding of the experimental setups.

- **Monthly Forecasting:** Weight & Biases Dashboard - Monthly Forecasting
- **Global Forecasting:** Weight & Biases Dashboard - Global Forecasting
- **Regional Forecasting (North America):** Weight & Biases Dashboard - North America
- **Regional Forecasting (South America):** Weight & Biases Dashboard - South America
- **Regional Forecasting (Australia):** Weight & Biases Dashboard - Australia
- **Hyperparameter Search:** Weight & Biases Dashboard - Hyperparameter Search
- **Longer Lead Time Forecasting:** Weight & Biases Dashboard - Longer Lead Time Forecasting

B Figures

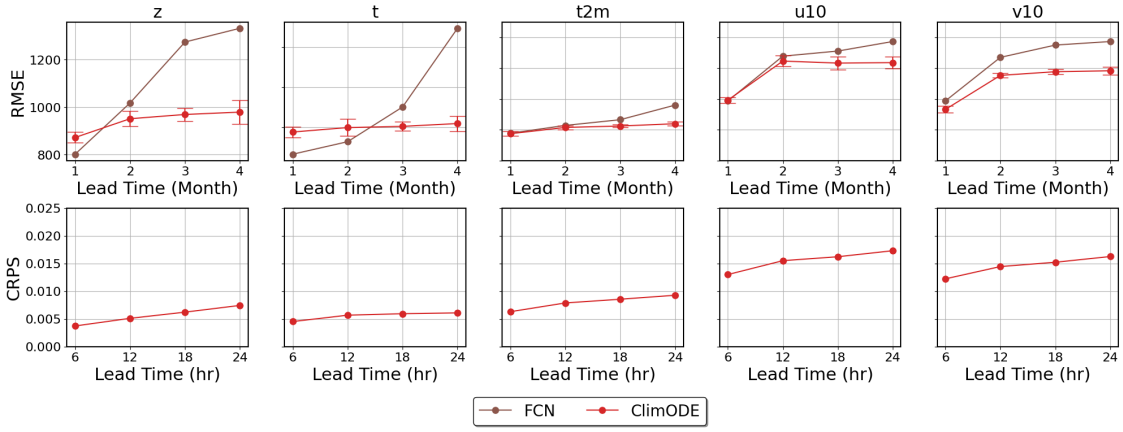

Figure 3: CRPS and Monthly Forecasting present in the original paper: RMSE(↓) comparison with FourCastNet (FCN) for monthly forecasting and CRPS scores for ClimODE.

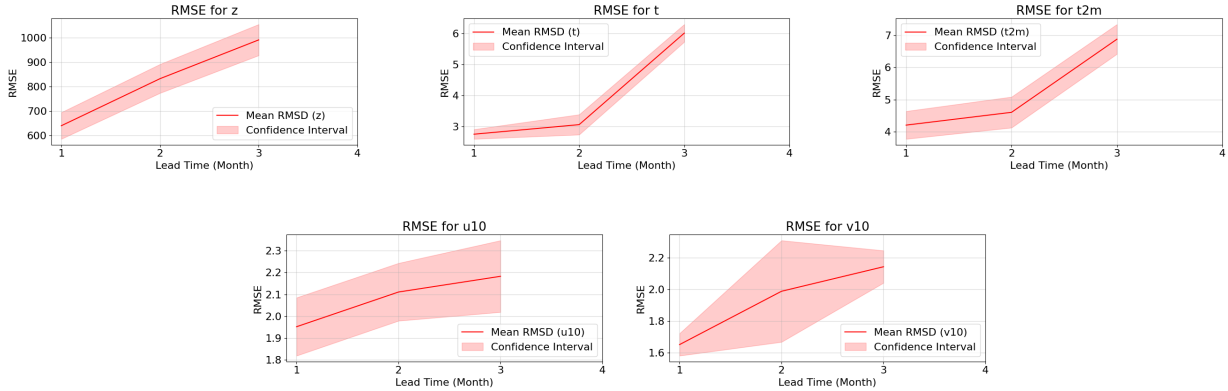

Figure 4: RMSE(↓) for monthly forecasting: Replicated results for variables z, t, $t2m$, $u10$, and $v10$. Each subplot shows the mean and confidence intervals for RMSD.

C Tables

Table 2: Performance of ClimODE model with global forecasting.

Variable	Lead-Time (hours)	RMSE (↓)		ACC (↑)	
		ClimODE	Replicate	ClimODE	Replicate
z	6	102.9 ± 9.3	102.89 ± 9.25	1.00	0.995
	12	134.8 ± 12.3	134.40 ± 12.18	0.99	0.991
	18	162.7 ± 14.4	162.19 ± 13.92	0.98	0.987
	24	193.4 ± 16.3	193.98 ± 16.23	0.96	0.981
	36	259.6 ± 22.3	259.31 ± 22.12	0.96	0.966
t	6	1.21 ± 0.09	1.11 ± 0.05	0.99	0.975
	12	1.47 ± 0.13	1.32 ± 0.06	0.99	0.964
	24	1.55 ± 0.18	1.50 ± 0.07	0.99	0.954
	36	1.83 ± 0.20	1.70 ± 0.08	0.98	0.940
$t2m$	6	1.21 ± 0.09	1.23 ± 0.07	0.97	0.970
	12	1.48 ± 0.15	1.45 ± 0.11	0.98	0.957
	24	2.22 ± 0.18	1.42 ± 0.10	0.92	0.964
	36	2.55 ± 0.18	1.71 ± 0.16	0.91	0.940
$u10$	6	1.41 ± 0.08	1.64 ± 0.08	0.91	0.918
	12	1.89 ± 0.13	1.85 ± 0.09	0.89	0.892
	24	2.48 ± 0.15	2.02 ± 0.09	0.89	0.872
	36	3.18 ± 0.18	2.26 ± 0.10	0.85	0.836
$v10$	6	1.51 ± 0.07	1.66 ± 0.09	0.92	0.918
	12	1.83 ± 0.10	1.88 ± 0.09	0.91	0.890
	24	2.29 ± 0.24	2.05 ± 0.08	0.89	0.869
	36	2.29 ± 0.24	2.30 ± 0.10	0.89	0.831

Table 3: Performance of ClimODE and our replicate for various variables and long lead times.

Variable	Lead-Time (hours)	RMSE (↓)		ACC (↑)	
		ClimODE	Replicate	ClimODE	Replicate
z		478.7 ± 48.3	736.2 ± 147.9	0.88 ± 0.04	0.72 ± 0.12
t		2.58 ± 0.16	2.96 ± 0.24	0.85 ± 0.06	0.80 ± 0.10
$t2m$	72	2.75 ± 0.49	3.52 ± 1.15	0.85 ± 0.14	0.77 ± 0.16
$u10$		3.19 ± 0.18	3.60 ± 0.17	0.66 ± 0.04	0.49 ± 0.06
$v10$		3.30 ± 0.22	3.73 ± 0.21	0.63 ± 0.05	0.43 ± 0.06

Table 4: Performance comparison of ClimODE on regional forecasting.

		North-America		South-America		Australia	
Value	Hours	ClimODE	Replicate	ClimODE	Replicate	ClimODE	Replicate
z	6	134.5 ± 10.6	161.2 ± 44.5	107.7 ± 20.2	122.8 ± 28.8	103.8 ± 14.6	81.1 ± 19.9
	12	225.0 ± 17.3	254.1 ± 65.1	169.4 ± 29.6	160.7 ± 35.6	170.7 ± 21.0	127.0 ± 33.7
	18	307.7 ± 25.4	334.9 ± 84.2	237.8 ± 32.2	208.6 ± 48.5	211.1 ± 31.6	169.4 ± 46.6
	24	390.1 ± 32.3	421.4 ± 110.9	292.0 ± 38.9	268.6 ± 60.0	308.2 ± 30.6	217.4 ± 59.6
t	6	1.28 ± 0.06	1.60 ± 0.25	0.97 ± 0.13	1.29 ± 0.27	1.05 ± 0.12	0.86 ± 1.14
	12	1.81 ± 0.13	2.38 ± 0.40	1.25 ± 0.18	1.38 ± 0.21	1.20 ± 0.16	1.07 ± 0.17
	18	2.03 ± 0.16	2.39 ± 0.42	1.43 ± 0.20	1.46 ± 0.22	1.33 ± 0.21	1.17 ± 0.19
	24	2.23 ± 0.18	2.72 ± 0.50	1.65 ± 0.26	1.65 ± 0.25	1.63 ± 0.24	1.30 ± 0.21
$t2m$	6	1.61 ± 0.2	4.53 ± 1.82	1.33 ± 0.26	3.29 ± 1.34	0.80 ± 0.13	0.87 ± 0.16
	12	2.13 ± 0.37	5.08 ± 1.52	1.04 ± 0.17	1.83 ± 1.52	1.10 ± 0.22	0.98 ± 0.14
	18	1.96 ± 0.33	4.10 ± 1.25	0.98 ± 0.17	2.64 ± 1.84	1.23 ± 0.24	1.14 ± 0.21
	24	2.15 ± 0.20	4.88 ± 2.29	1.17 ± 0.26	2.68 ± 1.11	1.25 ± 0.25	1.03 ± 0.18
$u10$	6	1.54 ± 0.19	1.72 ± 0.30	1.25 ± 0.18	1.50 ± 0.25	1.35 ± 0.17	1.37 ± 0.17
	12	2.01 ± 0.20	2.20 ± 0.37	1.49 ± 0.23	1.68 ± 0.27	1.78 ± 0.21	1.73 ± 0.21
	18	2.17 ± 0.34	2.19 ± 0.43	1.81 ± 0.29	1.80 ± 0.29	1.96 ± 0.25	1.92 ± 0.26
	24	2.34 ± 0.32	2.36 ± 0.44	2.08 ± 0.35	2.00 ± 0.31	2.33 ± 0.33	2.10 ± 0.32
$v10$	6	1.67 ± 0.23	1.86 ± 0.34	1.30 ± 0.21	1.64 ± 0.31	1.44 ± 0.20	1.50 ± 0.18
	12	2.03 ± 0.31	2.22 ± 0.35	1.71 ± 0.28	1.84 ± 0.29	1.87 ± 0.26	1.88 ± 0.23
	18	2.31 ± 0.37	2.30 ± 0.45	2.07 ± 0.31	2.04 ± 0.35	2.23 ± 0.23	2.07 ± 0.28
	24	2.50 ± 0.41	2.48 ± 0.44	2.43 ± 0.34	2.29 ± 0.38	2.53 ± 0.32	2.28 ± 0.33

Table 5: Performance of ClimODE replication on monthly forecasting.

Observable	Lead-Time (Months)	RMSD (↓)	ACC (↑)
z	1	639.95 ± 54.27	0.53 ± 0.19
	2	832.32 ± 58.96	0.31 ± 0.32
	3	990.86 ± 63.56	0.35 ± 0.11
t	1	2.75 ± 0.16	0.59 ± 0.25
	2	3.06 ± 0.33	0.64 ± 0.18
	3	6.01 ± 0.28	0.38 ± 0.20
$t2m$	1	4.21 ± 0.43	0.49 ± 0.26
	2	4.61 ± 0.48	0.57 ± 0.17
	3	6.88 ± 0.46	0.39 ± 0.22
$u10$	1	1.95 ± 0.13	0.30 ± 0.17
	2	2.11 ± 0.13	0.26 ± 0.20
	3	2.18 ± 0.16	0.35 ± 0.07
$v10$	1	1.65 ± 0.07	0.34 ± 0.17
	2	1.99 ± 0.32	0.10 ± 0.28
	3	2.14 ± 0.10	0.10 ± 0.02