

Integrating Physical Laws in ClimSim

Clark Kaminsky
clarkkam@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Álvaro Vega-Hidalgo
alvarovh@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

1 Introduction

Modeling climate simulations over extended periods is a computationally intensive process, and reducing temporal or spatial resolution often compromises the expected accuracy. By utilizing standard machine learning or physics-informed ML methods, researchers are beginning to break this barrier of trading massive computational cost for accuracy. Especially with climate forecasting having high uncertainty over the coming decades, improving our estimations on the level of global climate change among temperature, extreme weather, and ocean patterns can have huge impacts on the real world, including land use, technological innovation, and governmental practices. To address the environmental impact, we must also focus on improving the runtime of climate simulations to make them less computationally costly and more energy-efficient. This not only reduces expenses but also minimizes the carbon footprint of AI-driven climate models, aligning with the goal of combating climate change rather than exacerbating it through resource-intensive processes[2].

In this project, we aim to work on improving climate simulation and forecasting by utilizing simulation data from ClimSim[13], a massive multivariate dataset built specifically for machine learning climate emulation, by incorporating physics informed domain knowledge with this useful simulation data on top of modern machine learning methods.

2 Problem Definition

Machine learning models are being used commonly for the task of climate prediction in the modern scientific community, and are becoming more and more effective and accurate by leveraging physics-informed domain knowledge. Our project will build upon this by adding physical constraints, focusing particularly on energy conservation, non-negativity, and water mass conservation principles. Instead of solving an ODE or PDE like physics-informed neural networks (PINNs), we aim to integrate these constraints directly within additional loss terms in a multi-layer perceptron (MLP). We hope that by enforcing energy and water mass conservation, we will guide the model to keep the incoming energy and water mass into the system equal to the resources leaving the system to improve the stability of the model, similarly to Hamiltonian Neural Networks [3]. This could also help in preventing drift of our variables like precipitation and temperature over long simulation periods, without which, the network might accumulate error over time. Our approach is designed to improve the physical consistency of temperature predictions, while also utilizing the foundations of a simple MLP model to retain simplicity. By building on the baseline models from the ClimSim [13] study, which include MLPs, convolutional neural networks, and encoder-decoder architectures, we will evaluate our knowledge-guided model's performance and stability in comparison to previous purely data driven works.

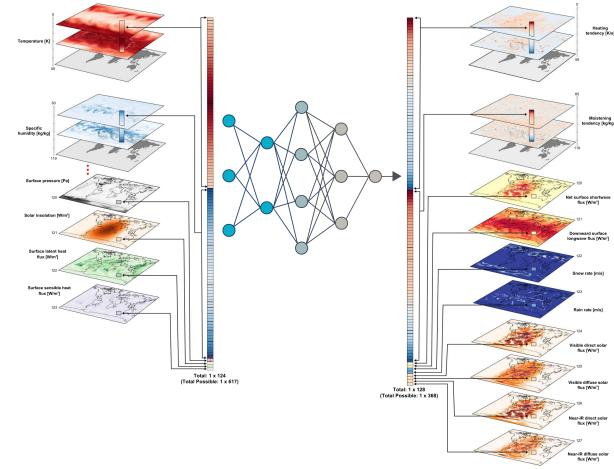


Figure 1: Input and output structure of ClimSim[13]. This figure is reproduced from its original publication.

3 Related Work

Simulating climate data is a challenging task due to the chaotic nature of weather systems, where complex interactions such as cloud and storm formation create significant difficulties for traditional numerical simulations. To address these issues, advanced climate models such as the Energy Exascale Earth System Model (E3SM) [4] have been developed to integrate high-performance computing with climate modeling, focusing on large-scale phenomena and their implications for climate change.

ClimSim [13] offers a dataset created from high-resolution E3SM climate simulations that can be used for machine learning (ML) applications. Figure 1 shows an example of how their input and output data is structured, and how it can be used in a neural network. Initial ML models applied to ClimSim data, such as convolutional neural networks and multi-layer perceptrons (MLPs), demonstrated strong predictive performance. However, these models lacked the incorporation of domain-specific physical constraints, which could improve their performance and robustness.

Purely data driven models have achieved recent success in generating global forecasts of key atmospheric variables like precipitation, wind speed, and temperature. For example, FourCastNet [8] uses a Fourier Neural Operator model with a vision transformer to allow for continuous global convolution in the Fourier domain, achieving short term predictions that compare to state of the art numerical weather prediction methods with orders of magnitude less computational expense. This system models current state with only a few variables, including temperature, zonal (east-west) and meridional (north-south) wind velocity, geopotential, relative humidity, and

total column water vapor; we hope to utilize the huge amount of data generated by ClimSim to improve upon these concepts, by integrating domain knowledge of energy conservation on top of a data driven model.

A growing body of research emphasizes the integration of physical principles within ML models to enhance their predictions. The concept of physics-informed neural networks (PINNs), introduced by Raissi et al. [9], incorporates differential equations into the loss function, aligning the learning process with physical laws. This approach has proven effective for tasks requiring explicit ODE/PDE solutions but may not be ideal when the physical relationships are more complex or involve indirect constraints.

Recent work, such as ClimODE [11], has demonstrated the potential of combining deep learning with physical modeling. ClimODE embeds physical dynamics into a neural ODE framework, achieving state-of-the-art performance by ensuring that outputs remain consistent with known physical behavior. The WeatherGFT model [12] extends this approach by integrating PDE kernels and neural networks to achieve fine-grained temporal forecasting, thus addressing the limitations of black-box ML models that struggle with finer time resolutions.

Transformers have also been explored for weather forecasting, showing promise in medium-range predictions. The Stormer model [7] employs a transformer backbone to outperform complex architectures like Pangu-Weather and GraphCast. Its innovations include randomized dynamics forecasting and pressure-weighted loss, enabling robust predictions with fewer data and computational resources.

In summary, while previous approaches have either focused on direct data-driven forecasting or employed complex physics-informed methods, our proposed framework seeks to bridge these by incorporating energy conservation principles into an MLP. This approach builds on the baseline models of ClimSim [13] while leveraging recent advancements in physics-informed ML [7, 11, 12].

Our objective is to create a new MLP-based model that prioritizes physical realism by adding constraints according to conservation of mass and energy to the training process, ensuring that predicted temperature changes are consistent with fundamental energy balance principles.

4 Proposed Approach

We will be using a multi-layer perceptron model with multiple new loss terms corresponding to several physical constraints to predict the following variables that are key to weather forecasting: specific humidity (kg/kg), air temperature at Earth's surface (K), as well as variables related to energy flux through the climactic system: net shortwave flux at surface (W/m^2), downward longwave flux at surface (W/m^2), downward visible direct solar flux to surface (W/m^2), downward near-infrared direct solar flux to surface (W/m^2), downward visible diffuse solar flux to surface (W/m^2), and downward near-infrared diffuse solar flux to surface (W/m^2). Specifically, our task is to predict these output variables at each of 384 global locations, 20 minutes in the future from when the input measurements were taken.

We will investigate which of energy conservation, water mass conservation, and non-negativity constraints can improve our predictions over a baseline MLP that utilizes only data-driven loss. Each of these methods will be further explained in the following sections.

Our data is taken from the ClimSim lower resolution dataset, a set of 10 years of time series data with features corresponding to an $11.5^\circ \times 11.5^\circ$ horizontal resolution over the planet. In summary, we incorporate and test multiple loss terms based on predicted radiation, precipitation, specific humidity, and heat fluxes at each sampling location: an energy-conserving loss term to guide the model to keep the net incoming solar radiation equal to the sum of net outgoing radiation variables, a water-mass conserving loss to preserve atmospheric water, and a non-negativity loss term to ensure realistic predictions, like the fact that temperature predictions in Kelvin will always be positive. Each term is normalized in the loss function and scaled by a weighting factor.

4.1 Conservation of Energy

To integrate energy conservation, we implement a loss function that ensures the total change in energy is balanced by the incoming and outgoing energy fluxes. The main idea is to maintain the net energy balance at the Earth's surface using a few key energy fluxes from the ClimSim data:

- Shortwave radiation, which includes incoming solar insolation (SOLIN) and net shortwave flux at the surface (SW).
- Longwave radiation, which is composed of downward longwave flux at the surface (LWDS) and upward longwave flux (LWUP).
- Surface fluxes, including surface latent heat flux (LH) and surface sensible heat flux (SH).

The latter three components represent the different ways energy is transferred from the Earth's surface to the atmosphere. The upward longwave radiation is the thermal radiation emitted by the Earth's surface due to its temperature, where a warmer surface emits more longwave radiation. The surface latent heat flux represents the energy required for the phase change of water during evaporation and transpiration, absorbed by water molecules in the atmosphere. The surface sensible heat flux is the energy transferred due to the temperature difference between the surface and the air above it through conduction and convection.

With these movements of energy in the system, the conservation of energy, balancing surface fluxes with incoming radiation, can be expressed as:

$$\text{NETFLX} = \text{NETSW} + \text{NETLW}, \quad (1)$$

where:

$$\text{NETSW} = \text{SOLIN} - \text{SW}, \quad (\text{net shortwave radiation})$$

$$\text{NETLW} = \text{FLWDS} - \text{LWUP}, \quad (\text{net longwave radiation})$$

$$\text{NETFLX} = \text{SH} + \text{LH}, \quad (\text{surface fluxes})$$

Loss Function:

$$L_{\text{energy}} = \frac{1}{N} \sum_{i=1}^N \left| (\text{NETSW}_i + \text{NETLW}_i) - (\text{SH}_i + \text{LH}_i) \right| \quad (2)$$

This loss term should guide our model towards energy conservation at the Earth's surface by balancing net radiation (the sum of shortwave and longwave radiation) with sensible and latent heat fluxes.

4.2 Mass Conservation of Water Vapor and Precipitation

Similarly to energy conservation, we want to maintain the net water mass balance between Earth's surface and the atmosphere, as water is simply transported from one place to another across the globe, not being created or destroyed. To achieve a balance in incoming and outgoing water mass, we want the change in the amount of water vapor in the air to be equal to the difference between the amount of water that evaporates from surfaces (E) and the amount of water that falls as precipitation (P). Intuitively, if precipitation is higher than evaporation, specific humidity should decrease equivalently. Using rain rate (PRECC), snow rate (PRECSC), latent heat flux (LH), and the change of specific humidity over time (q_{tend}) from the ClimSim dataset, and the constant latent heat of vaporization (L_v), we can use the following equations:

$$\Delta q_{\text{water}} = E - P \quad (3)$$

where:

$$\Delta q_{\text{water}} = q_{tend} \times \Delta t, \quad (\text{change in specific humidity over time})$$

$$E = \frac{LH}{L_v}, \quad (\text{evaporation rate})$$

$$P = \text{PRECC} + \text{PRECSC}, \quad (\text{total precipitation rate}).$$

Loss Function:

$$L_{\text{mass}} = \frac{1}{N} \sum_{i=1}^N |\Delta q_{\text{water}_i} - (E_i - P_i)| \quad (4)$$

This loss should ensure the conservation of water mass in the atmosphere, often required in numerical simulation climate models, being fundamental to weather formation, hydrological processes, and energy transport. We hope that this simple method of introducing water mass conservation will be able to guide our model's predictions.

4.3 Non-Negativity Constraints

The third loss term we introduce is simple, with the goal of constraining variables to the positive domain when we know they can never be negative from domain knowledge. For example, snow rate (in m/s), rain rate (m/s), specific humidity (kg/kg), or temperature (°K) are all guaranteed to be greater than 0, so we want our model's loss to be higher if it predicts these terms to be negative. We can formulate this loss simply, given a list of variables that are defined as non-negative:

$$v \geq 0 \quad \forall v \in \{\text{TEMP}, \text{PRECC}, \text{PRECSC}, \dots\} \quad (5)$$

Loss Function:

$$L_{\text{nonneg}} = \sum_v \frac{1}{N} \sum_{i=1}^N \max(0, -v_i) \quad (6)$$

This loss penalizes predictions that violate non-negativity constraints in physical systems, guiding the model to more easily predict physically plausible outputs for some variables.

4.4 Implementation

To combine these physics-informed loss functions in one model, we created a framework where weights can be tuned to weigh each term's relative contributions more or less:

$$L_{\text{total}} = L_{\text{data}} + \lambda_1 L_{\text{energy}} + \lambda_2 L_{\text{mass}} + \lambda_3 L_{\text{nonneg}} \quad (7)$$

Additionally, before we scale the losses by their weights (λ), we balance each of the losses following a simple function in equation 8, with the goal of having each loss contribute equally. Here, we estimate the initial loss with the first few batches and scale each loss value with the reciprocal of its initial value. Intuitively, by dividing each loss by its corresponding initial value, larger losses are effectively reduced, while smaller losses are made larger, helping to approximately balance the contributions of each loss. This is not a perfect solution, but is a simple implementation that can bring the losses closer in scale to one another; the task of adaptive loss balancing is difficult, with possible solutions described in [1, 5, 10].

$$L = \sum_{i=1}^n \frac{L_i}{L_i^{(\text{initial})}} \quad (8)$$

5 Evaluation

We use similar metrics as what was used for the ClimSim[13] baseline models. This evaluation will include the following methods:

Mean Squared Error:

$$MSE = \frac{1}{n} \sum_{i=1}^n (X_i - y)^2$$

Coefficient of Determination R^2

$$R^2 = 1 - \frac{\sum_{i=1}^n |X_i - y|^2}{\sum_{i=1}^n |X_i - \bar{X}|^2}$$

where X_i represents the true values, y represents the predicted values, and \bar{X} represents the mean of the true values of the dependent variable.

Among the ClimSim baselines, the MLP achieved the highest results, with the lowest MSE and highest R^2 in almost every target variable prediction. Therefore, this MLP model's performance will be our target to evaluate against – our goal is to reach a lower MSE and higher R^2 than the plain MLP by incorporating physical domain knowledge.

Additionally, we include geospatial plots of these metrics across latitude and longitude to give supplementary information on where the predictions succeed and where they fail, or to visually evaluate whether there is a common pattern across better or worse results.

6 Data

The ClimSim[13] dataset was created by the Earth with Artificial Intelligence and Physics (LEAP) institution, which is an NSF Science Technology Center (STC) launched in 2021. It was created through running the E3SM-MMF multi-scale simulator for 9800 GPU-hours,

Input	Size	Target	Size
Temperature [K]	60	Heating tendency, dT/dt [K/s]	60
Specific humidity [kg/kg]	60	Moistening tendency, dg/dt [kg/kg/s]	60
Surface pressure [Pa]	1	Net surface shortwave flux, NETSW [W/m ²]	1
Insolation [W/m ²]	1	Downward surface longwave flux, FLWDS [W/m ²]	1
Surface latent heat flux [W/m ²]	1	Snow rate, PRECSC [m/s]	1
Surface sensible heat flux [W/m ²]	1	Rain rate, PRECC [m/s]	1
		Visible direct solar flux, SOLS [W/m ²]	1
		Near-IR direct solar flux, SOLL [W/m ²]	1
		Visible diffused solar flux, SOLSD [W/m ²]	1
		Near-IR diffused solar flux, SOLLD [W/m ²]	1

Figure 2: Input and target variables used in most of the Clim-Sim original paper experiments. Table taken from its original publication [13].

with data collected at 20-minute intervals over a span of 10 simulated years. It totals 526,600 files per configuration. The dataset is available in three configurations: high-resolution real geography with a $1.5^\circ \times 1.5^\circ$ horizontal grid (21,600 grid columns) producing 5.7 billion samples (41.2 TB), low-resolution real geography with an $11.5^\circ \times 11.5^\circ$ grid (384 grid columns) resulting in 100 million samples (744 GB), and low-resolution aquaplanet (an idealized Earth climate model with only ocean, no land), which has the same resolution and size as the low-resolution real geography configuration.

ClimSim includes 24 2D variables such as Surface Pressure (Pa) and Rain rate (m/s) and 10 3D variables (discretized to 60 vertical atmospheric levels) such as Temperature (K) and Specific humidity (kg/kg). The available data was down-sampled to make sure that was usable and adequately scoped for the resources available in our project. We followed a similar approach to what the ClimSim authors described in their paper, using a subset of crucial variables (Table 1; as presented in the original paper), and down-sampling spatially and temporally. We explored different down-sampling factors to find what was most usable with the hardware available for our experiments, starting with the provided downsampled low resolution dataset provided in the ClimSim repository (26.13GB), and moving on to 1% and 10% of the full dataset. Additionally, in our preprocessing pipeline, we normalized the variables for training stability, and subsampled multilevel variables to 1 altitude level (surface) instead of 60 to avoid further hardware limitations.

7 Experiments and Results

7.1 Research Questions

In our experiments, our goal was to find answers to the following questions:

- Can incorporating simple conservation laws improve weather forecasting predictions in a multi-layer perceptron?
- How are knowledge-guided prediction errors distributed geospatially? Are they uniform across the globe, or are there locations that are more prone to higher errors?
- Which physical constraint (energy conservation, water mass conservation, or non-negativity) has the most significant impact on improving the predictive performance of the MLP model for weather forecasting variables?
- What are the optimal weights to balance the influence of each loss constraint?

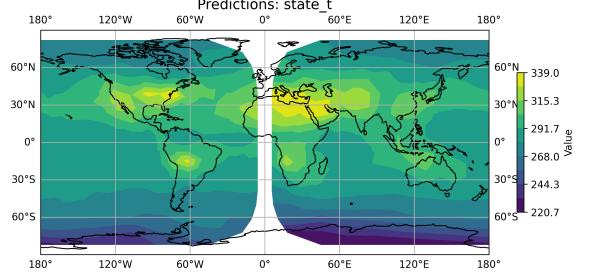


Figure 3: Predicted temperature map (°K).

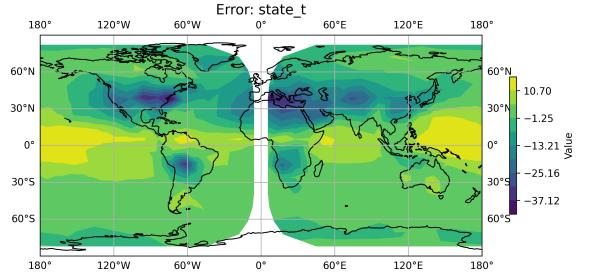


Figure 4: Map of error between predicted temperature and ground truth (°K).

7.2 Results

For each snapshot of input data we are given, we predict the output variables corresponding to each location 20 minutes in the future. For example, in Figure 3, we can see an example of one of our models’ predictions of temperature (°K) at one time step. We can also see the corresponding error compared to ground truth from this example in Figure 4, where we generally see strong predictions over oceans, and pockets of high errors around continental borders and the equator.

In Figure 5, we compare the results of testing each loss function in terms of mean squared error of the predicted temperatures. The baseline MLP that uses only data loss to guide its optimization performs relatively well, being denoted by ‘Pure MSE’. Each boxplot to the right of ‘Pure MSE’ in the figure achieves equal or better results than the baseline, and each to its left performed equally or worse. When testing each loss function alone, all other losses except data loss were set to 0, and we performed five experiments for each method. First, the loss’s weight (λ) was varied over the values [0.05, 0.25, 0.5, 1], and then in a following experiment, set as a trainable parameter. Additionally, we tested a combined model that used all of the losses, allowing each of $\lambda_{1:3}$ to be trainable.

In these results, we can see that some of our added losses outperformed this baseline, while others made the model notably worse. Most importantly, we found the best model to be where we included all of our additional loss terms together, with each scaling factor ($\lambda_{1:3}$) being trainable, showing promise that our physics-based losses do offer some improvement for the model to capture physical laws. This model was able to achieve a mean squared

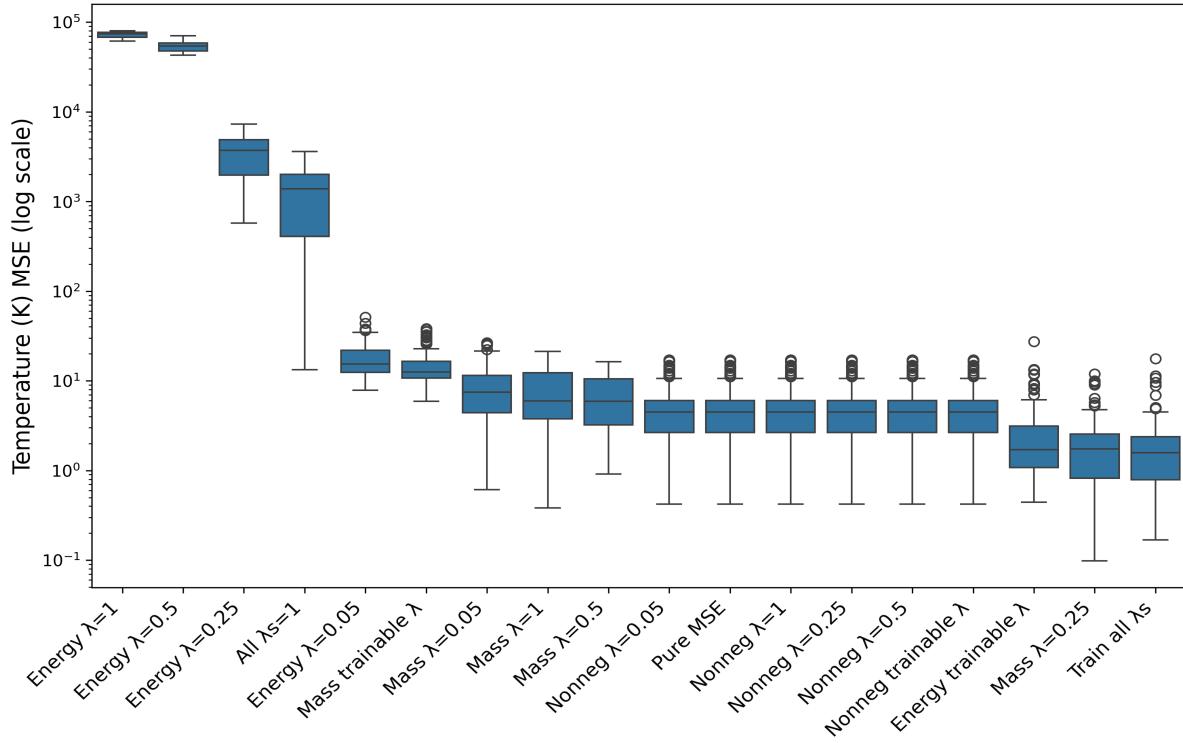


Figure 5: MSE of each loss function experiment on the validation set.

error on temperature predictions of only 2.05, which is a 60% improvement over the baseline’s MSE of 5.10. After the best model using all losses, we see one of our water-mass conservation losses (with $\lambda = 0.25$) and one of our energy conservation losses (with a trainable λ) performing highly, with losses in a similar range as our top-performing model. Interestingly, all of the non-negativity models on their own performed identically to the purely data driven model. Additionally, the majority of the energy loss experiments performed worse than simply using data loss on its own, many of which resulting in orders of magnitude higher MSE, but the third highest performing model was using energy loss with a trainable scaling factor.

Looking at the water mass conservation loss in particular, we can see that all of its experiments performed relatively well overall, with its best ($\lambda = 0.25$) being the second lowest MSE of any experiment, and its worst falling relatively close to the baseline data-driven loss. We believe that the water mass conservation loss performed very well because its effects are quite localized; with our predictions not being long-term forecasting, but instead just 20 minutes in the future, it makes sense for the model to be able capture and maintain the water mass balance effectively – a short forecasting horizon allows the water mass conservation constraint to have a more direct impact on the accuracy of predictions, as there is less time for water cycle processes to proliferate further from each location. Additionally, given that specific humidity and other water-related variables are highly forcing variables for temperature and

precipitation, preserving the water mass balance seems to directly help the prediction of the output variables we chose to include.

While the previous addition found mostly strong success, the performance of the energy conservation loss was more inconsistent, with the majority of its experiments showing significant decreases in performance under the baseline, but one experiment (trainable λ) being the third highest performing overall. While the interpretability of these results is not perfectly clear, the generally poorer performance of the energy conservation loss could be attributed to several potential issues. We believe that the relationships between net incoming solar radiation and other energy transfer processes might be too complex for our simple constraint without incorporating convolution or attention mechanisms. In looking at the ground truth data, the actual difference between net radiation and surface fluxes was quite variable, which could make it difficult for our model to learn well. Additionally, this loss term might have failed due to part of our subsampling strategy, where we removed the higher levels of atmospheric variables to reduce our data size and focus on surface-level interactions. Since energy flux often involves complex interactions between different layers of the atmosphere, like direct radiation being scattered and reflected off of atmospheric particles with various feedback mechanisms and atmospheric circulation patterns, we might have lost the ability to capture some information. However, this energy conserving loss does definitely have the potential to improve predictions, as we can see by the impact of the experiment with a trainable λ performing significantly better than the purely data-driven loss.

Looking at the non-negativity constraint, we find it performs equally to the baseline data loss across the board. Generally, we would expect that a non-negativity constraint would generally guide the optimization process to avoid regions of the parameter space that would result in impossible predictions – however, since we see no change or improvements over the baseline, perhaps data loss is sufficient in capturing this constraint already. If the model is never predicting negative values of our strictly non-negative variables, we can imagine that this added loss term would always be 0, which would be the same as simply the data loss on its own in these isolated experiments.

Experiment	Temperature MSE ($^{\circ}\text{K}^2$)
Energy $\lambda=1$	72712.51
Energy $\lambda=0.5$	54177.82
Energy $\lambda=0.25$	3514.68
All λ s=1	1321.56
Energy $\lambda=0.05$	17.73
Mass trainable λ	14.40
Mass $\lambda=0.05$	8.48
Mass $\lambda=1$	7.87
Mass $\lambda=0.5$	6.84
Nonneg $\lambda=0.25$	5.10
Pure MSE	5.10
Nonneg $\lambda=0.05$	5.10
Nonneg $\lambda=1$	5.10
Nonneg $\lambda=0.5$	5.10
Nonneg trainable λ	5.10
Energy trainable λ	2.70
Mass $\lambda=0.25$	2.14
Train all λs	2.05

In Figure 6, we can see what the training process looked like for the best resulting model, including the loss and trainable scaling factors for water mass, energy, and non-negativity across each batch in our training set. The loss decreases over time and looks to be converging, but there is still some clear variation between each batch. This might be due to the fact that we are treating each global location the same, but physical properties interact differently based on geographic relationships.

The plots below show us how the model optimized the scaling factors ($\lambda_{1:3}$) over the course of training, where we can see different trends for each of the physical losses. The water-mass conserving loss's λ dropped from an initially very high level, but still mostly flattened out at a high scaling factor, indicating that including this loss term was useful for guiding the optimization of the system. Surprisingly, after about 300 batches, the energy conservation's scaling factor was brought close to 0. This trend could show that our energy conserving term was not consistently useful for predictions, perhaps due to the issues discussed above. However, this is surprising after finding that the experiment with a trainable energy loss λ performed significantly above the baseline. This could be due to the difficulty of optimizing multiple objectives: the model has to both minimize its loss and also explore the optimal weights for each individual loss term concurrently, which could have led to the model finding a stable λ around 0 for the energy loss term and not

exploring much further. Lastly, the non-negativity term remained constant at an extremely large value, strongly disincentivizing negative values over the entire training duration.

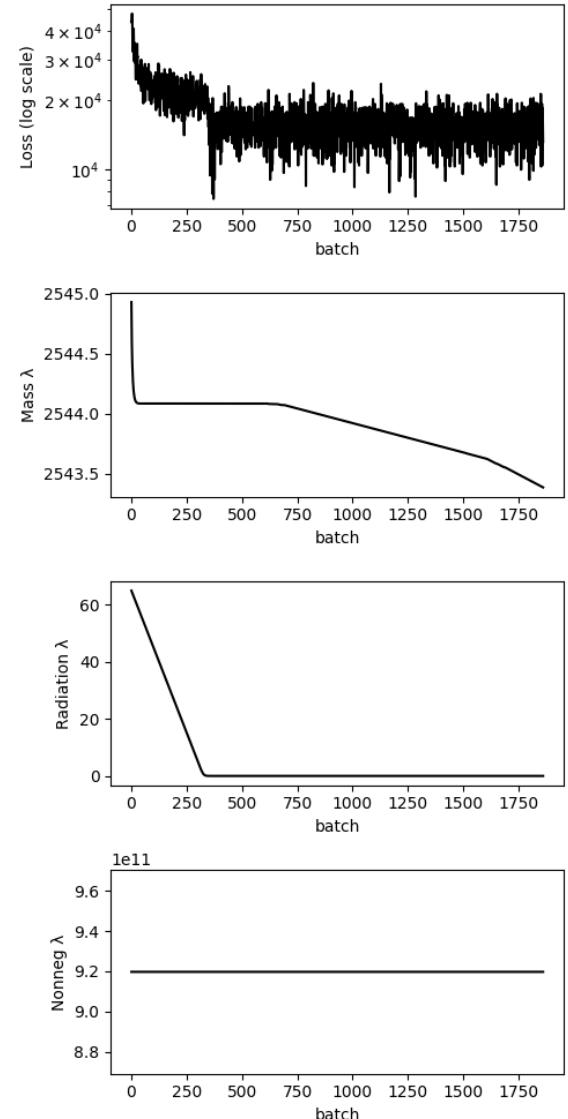


Figure 6: Highest performing experiment, with $\lambda_{1:3}$ being learnable: a) Loss during training. b) Water mass loss scaling factor. c) Energy loss scaling factor. d) Non-negativity loss scaling factor.

8 Challenges and Limitations

One of the most time-consuming challenges we faced in this project was working with and analyzing the ClimSim dataset. The documentation is limited, and there are multiple inconsistencies in the version control system, making it difficult to progress with the suggested implementation. Different versions of the datasets are

preprocessed in different ways, and the baseline models' code is dependent on specific versions of the datasets, but none are explicitly described.

9 Conclusion

If there was more time to work on this project, exploring incorporating temporal dependencies through techniques like sequence-to-sequence learning [6] would be an interesting direction to explore. This could enhance the model's ability to leverage convection memory and capture the correlations between successive time intervals, a common feature in atmospheric data.

Additionally, we only ended up being able to use 10% of our entire dataset while training due to time and computational resource constraints. We believe that increasing the amount of data we were using could further improve the models, and might even allow some of our weaker experiments to capture relationships that were unseen before.

We would also like to improve the interpretability of our experiments in the future. Some of the trends we studied here were not fully explainable due to the black-box nature of neural networks; we are especially interested about the energy conserving loss term's relationship with our dataset. It seems like it has the potential to guide the model in some situations, but otherwise we saw the model scaling it to nearly 0, like in Figure 6.

Another area to explore in the future would be to try to improve the energy conservation loss term by setting boundary conditions at the top and bottom of the model to constrain energy dynamics. The authors of ClimSim suggested that preventing the surface radiative flux from exceeding the downward shortwave flux at the model's top, or similarly, bounding the net shortwave flux at the surface level between zero (full reflection) and the downward shortwave flux (full absorption) could improve energy predictions.

Overall, we found some significant results by adding our physics-based loss terms to an MLP with the ClimSim dataset: while our energy conservation loss was very inconsistent, we found our water conservation loss and non-negativity constraints to strongly guide our model's predictions, improving results over the baseline data-driven model.

References

- [1] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multi-task Networks. arXiv:1711.02257 [cs.CV] <https://arxiv.org/abs/1711.02257>
- [2] Josh Cowls, Andreas Tsamados, Mariarosaria Taddeo, and Luciano Floridi. 2023. The AI gambit: leveraging artificial intelligence to combat climate change—opportunities, challenges, and recommendations. *Ai & Society* (2023), 1–25.
- [3] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. 2019. Hamiltonian neural networks. *Advances in neural information processing systems* 32 (2019).
- [4] Walter M Hannah, Christopher R Jones, Benjamin R Hillman, Matthew R Norman, David C Bader, Mark A Taylor, LR Leung, Michael S Pritchard, Mark D Branson, Guangxing Lin, et al. 2020. Initial results from the super-parameterized E3SM. *Journal of Advances in Modeling Earth Systems* 12, 1 (2020), e2019MS001863.
- [5] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. arXiv:1705.07115 [cs.CV] <https://arxiv.org/abs/1705.07115>
- [6] Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. 2021. Characterizing possible failure modes in physics-informed neural networks. arXiv:2109.01050 [cs.LG] <https://arxiv.org/abs/2109.01050>
- [7] Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. 2023. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. *arXiv preprint arXiv:2312.03876* (2023).
- [8] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamvar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. 2022. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. arXiv:2202.11214 [physics.ao-ph] <https://arxiv.org/abs/2202.11214>
- [9] M. Raissi, P. Perdikaris, and G.E. Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [10] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-Task Learning for Dense Prediction Tasks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. <https://doi.org/10.1109/tipami.2021.3054719>
- [11] Yogesh Verma, Markus Heinonen, and Vikas Garg. 2024. Climode: Climate and weather forecasting with physics-informed neural odes. *arXiv preprint arXiv:2404.10024* (2024).
- [12] Wanghan Xu, Fenghua Ling, Wenlong Zhang, Tao Han, Hao Chen, Wanli Ouyang, and Lei Bai. 2024. Generalizing Weather Forecast to Fine-grained Temporal Scales via Physics-AI Hybrid Modeling. *arXiv preprint arXiv:2405.13796* (2024).
- [13] Sungduk Yu, Walter Hannah, Liran Peng, Jerry Lin, Mohamed Aziz Bhouria, Ritwik Gupta, Björn Lütjens, Justus C. Will, Gunnar Behrens, Julius Busecke, Nora Loose, Charles Stern, Tom Beucler, Bryce Harrop, Benjamin Hillman, Andrea Jenney, Savannah L. Ferretti, Nana Liu, Animashree Anandkumar, Noah Brenowitz, Veronika Eyring, Nicholas Geneva, Pierre Gentine, Stephan Mandt, Jaideep Pathak, Akshay Subramanian, Carl Vondrick, Rose Yu, Laure Zanna, Tian Zheng, Ryan Abernathy, Fiaz Ahmed, David Bader, Pierre Baldi, Elizabeth Barnes, Christopher Bretherton, Peter Caldwell, Wayne Chuang, Yilun Han, YU HUANG, Fernando Iglesias-Suarez, Sanket Jantra, Karthik Kashinath, Marat Kharoutdinov, Thorsten Kurth, Nicholas Lutsko, Po-Lun Ma, Griffin Mooers, J. David Neelin, David Randall, Sara Shamekh, Mark Taylor, Nathan Urban, Janni Yuval, Guang Zhang, and Mike Pritchard. 2023. ClimSim: A large multi-scale dataset for hybrid physics-ML climate emulation. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 22070–22084. https://proceedings.neurips.cc/paper_files/paper/2023/file/45fbcc01349292f5e059a0b8b02c8c3f-Paper-Datasets_and_Benchmarks.pdf