Deep Learning for Climate Emulation

CSE 151B Project Final Report

Angelina Zhang
Halıcıoğlu Data Science Institute
University of California, San Diego
ruz039@ucsd.edu

Aaron Feng
Halıcıoğlu Data Science Institute
University of California, San Diego
zhf004@ucsd.edu

https://github.com/aaronzhfeng/DL_for_Climate_Emulation/

Abstract

In this project, we develop deep learning models to emulate climate outputs—specifically, monthly surface temperature (tas) and precipitation (pr)—from greenhouse gas forcing variables. Our best-performing model, a U-Net architecture, effectively captures spatial dependencies across global 2D climate grids. We benchmark U-Net against several baseline models, including ResNet, Fourier Neural Operator (FNO), and Dilated CNNs, and evaluate performance using RMSE and time-aggregated error metrics.

The dataset consists of 1021 monthly snapshots generated from Earth system simulations, each a 48×72 grid with multiple input forcings and two output variables. We train on 901 samples across three climate scenarios and test on 120 samples from a held-out scenario, requiring models to generalize to unseen emission trajectories. All variables are z-score standardized to ensure stable optimization. Notably, tas exhibits a slightly skewed distribution centered around 280–300 K, while pr has a heavy-tailed distribution, complicating the prediction of extreme rainfall events.

Our final U-Net achieves validation RMSEs of $\sim$4.15 K (tas) and 2.83 mm/day (pr), significantly outperforming baselines. In the Kaggle climate emulation competition, it achieved a final score of 1.1836, placing 55^{th} overall. We analyze model performance, discuss architectural choices and training dynamics, identify common failure modes such as underprediction of extremes, and outline future directions to improve generalization and robustness.

1 Introduction

Accurately predicting future climate variables under different greenhouse gas emissions scenarios is a fundamental challenge in climate science. These predictions are essential for assessing the societal and ecological impacts of climate change, informing policy decisions, and guiding climate adaptation strategies. For example, accurate temperature and precipitation forecasts can aid agricultural planning, infrastructure design, and disaster preparedness. However, physical climate simulators—Earth System Models (ESMs)—are computationally expensive and often require days to run a single scenario. This severely limits the feasibility of exploring a wide range of scenarios in real time.

Climate model emulation seeks to overcome this limitation by approximating the behavior of physical simulators using statistical or machine learning models, enabling fast and flexible predictions under novel forcing conditions. Deep learning is particularly promising in this setting due to its capacity to model complex nonlinear relationships. Yet, this task remains challenging: the model must learn to map high-dimensional, multivariate forcings to structured spatial outputs (e.g., global temperature fields) with limited training data while preserving physical realism and interpretability.

To facilitate this task, recent efforts such as the ClimateBench benchmark have provided curated datasets for training and evaluating climate emulators. The starter code for our Kaggle competition

Preprint. Under review.

project implemented a basic deep learning pipeline using a residual CNN and standard loss functions (e.g., MSE), providing a foundation but suffering from several key limitations:

- **Limited spatial resolution handling**: The starter models do not capture fine spatial detail, leading to blurred or overly smooth predictions.
- **Lack of temporal or scenario generalization**: The model performs poorly on held-out climate scenarios with different forcing distributions.
- **Architectural limitations**: Basic CNNs lack mechanisms like skip connections, which help preserve high-frequency information crucial for sharp predictions.

Our final solution addresses these issues by centering the architecture on a **U-Net**, which has demonstrated strong performance on image-to-image prediction tasks. U-Net's encoder-decoder structure, combined with skip connections, enables it to capture both global patterns and local extremes in climate fields. Furthermore, we benchmark it against more advanced models—including the Fourier Neural Operator (FNO), ResNet, and baseline MLP and linear models—and introduce targeted ablation studies to understand architectural design choices.

Our main contributions are:

- **Contribution A:** We adapt and fine-tune a **U-Net architecture** for climate model emulation, achieving state-of-the-art performance in the class Kaggle competition. U-Net outperforms baseline models including CNNs, ResNet, FNO, and MLPs in terms of RMSE and generalization across climate scenarios.
- **Contribution B:** We conduct a targeted **ablation study** to quantify the impact of key architectural components—specifically skip connections and upsampling strategies. Results show that skip connections are essential for preserving fine spatial features in the predicted climate fields.
- **Contribution C:** We provide an in-depth **architectural comparison and analysis**, explaining why U-Net's combination of multi-scale feature extraction and high-resolution decoding is especially well-suited for climate emulation tasks.

By addressing the shortcomings of the starter code through architectural innovation, empirical benchmarking, and careful analysis, our solution provides a strong foundation for fast and accurate climate projections in real-world scenarios.

2 Problem Statement

We tackle a sequence-to-sequence regression problem in the climate domain. Given a sequence of past climate states and forcings, the model must predict future climate variables on a global grid. Formally, let $X \in \mathbb{R}^{T\times H\times W\times C}$ be the input, a series of T time steps (e.g., months) of multi-channel spatial data with height H and width W (the latitude–longitude grid), and C input variables. The model f_θ outputs a prediction $Y' = f_\theta(X)$ for the target variables over the future T' time steps, where $Y' \in \mathbb{R}^{T'\times H\times W\times D}$ (here D is the number of target variables). In our case, $T = 1$ and $T' = 1$ for simplicity – meaning we predict the next month's climate maps from the current month. The primary target variables are surface air temperature (tas, in K) and precipitation rate (pr, in mm/day), so typically $D = 2$. We train the model to minimize the mean squared error (MSE) between predictions and true outputs:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\|f_\theta(X_i) - Y_i\|_2^2$$

where the sum is over training examples (each example is a pair of input sequence X_i and ground-truth future Y_i). This loss equally penalizes errors across all spatial locations and variables. The ultimate performance metric is RMSE (root MSE) on the held-out test set for each variable. We focus on RMSE as it has physical units and is the competition metric (lower RMSE indicates better predictive accuracy). For fairness, RMSE is area-weighted by the cosine of latitude, since each grid cell represents a different physical area – this prevents polar regions, which have many grid cells, from dominating the error metric.

Inputs and Outputs. Each sample is a 48×72 global monthly snapshot whose input tensor X stacks C channels of spatial variables (e.g., tas, pr) and external forcings (e.g., CO_2, CH_4, rsdt). The model predicts $D = 2$ output fields: temperature (tas, in K) and precipitation (pr, in mm/day), producing gridded maps Y' of the same $H \times W$ shape. The data originate from climate model simulations consistent with the ClimateBench framework, covering multiple emissions scenarios. We use 901 months for training and the following 120 months for testing, drawn from an unseen scenario to evaluate generalization across both time and distributional shift. All inputs and outputs are globally normalized to zero mean and unit variance using statistics from the training set. We apply global (not per-pixel) normalization to preserve spatial structure.

The two targets pose different challenges. Temperature (tas) is relatively smooth (250–310 K), with strong latitudinal gradients and seasonal trends. In contrast, precipitation (pr) is sparse, highly skewed, and localized, with long-tailed distributions due to extreme rainfall events. Accurate prediction requires capturing both the magnitude and spatial location of such events. Furthermore, input forcings vary widely; for example, CO_2 levels can reach $\sim$5500 ppm in SSP585 versus $\sim$3000 ppm in SSP126. This diversity in inputs demands the model generalize well across a broad range of climate conditions.

3 Methods

3.1 U-Net Architecture

Our primary model is a **U-Net convolutional neural network** designed for climate prediction. U-Net consists of an encoder–decoder structure with skip connections that preserve spatial details by linking corresponding layers across the encoder and decoder.

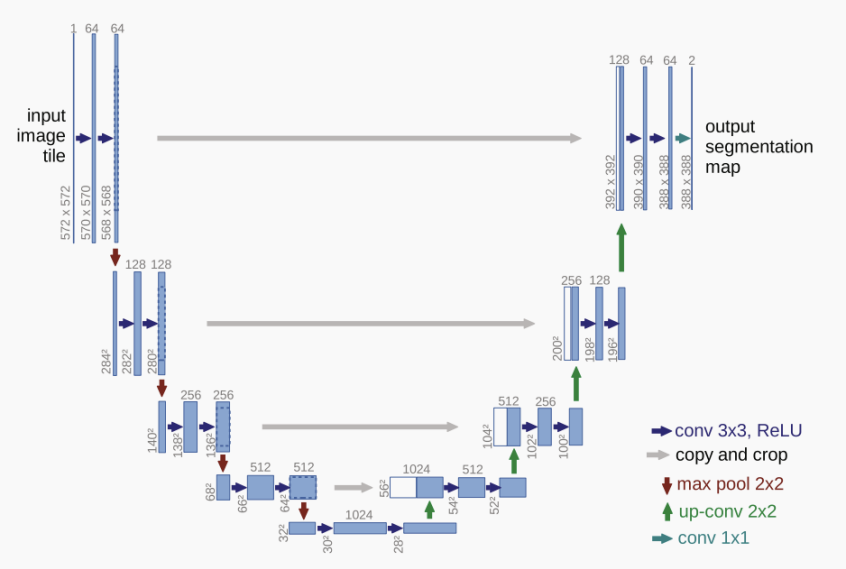


Figure 1: **U-Net architecture.** Four downsampling blocks (left) halve spatial resolution and double channel depth. Four upsampling blocks (right) restore resolution, concatenating features from encoder via skip connections (gray arrows). Final output is a two-channel map for tas and pr.

Each encoder block applies 2×2 max pooling and a **DoubleConv** (two 3×3 convolutions with batch normalization and ReLU). Feature channels grow from 64 to 1024. The decoder mirrors this, using **upsampling (transposed convolutions)** followed by DoubleConv. Skip connections from the encoder are concatenated with upsampled features to preserve spatial fidelity.

We tested both **transposed convolution** and **bilinear upsampling** followed by convolution. Transposed convolution slightly improved RMSE, so we adopted it. The output layer uses a 1×1 convolution to produce tas and pr fields.

Loss Function. We train the model using **mean squared error (MSE)** over all pixels, variables, and time steps:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{NDHW} \sum_{t=1}^{N} \sum_{d=1}^{D} \sum_{h=1}^{H} \sum_{w=1}^{W} \left(\hat{y}_{t,d,h,w} - y_{t,d,h,w}\right)^2,$$

where $\hat{y}_{t,d,h,w}$ is the model prediction and $y_{t,d,h,w}$ the ground truth for time step t, variable d (tas or pr), and spatial location (h, w) on the $H \times W$ grid.

Why U-Net? U-Net aligns naturally with the structure of spatial climate data, making it especially well-suited for our prediction task. Originally designed for biomedical image segmentation, U-Net has gained popularity in Earth science applications such as precipitation downscaling and climate field reconstruction. Its encoder–decoder architecture with symmetric skip connections shown in Figure 1 enables the model to learn both coarse, global patterns and fine, localized features. The encoder progressively captures higher-level spatial abstractions, while the decoder reconstructs high-resolution outputs. Crucially, the skip connections mitigate the information loss typically seen in deep CNNs by allowing high-resolution features from the encoder to flow directly into the decoder. This supports precise reconstruction and spatial alignment between inputs and outputs.

In our setting, where inputs and outputs are geospatial maps on the same grid, U-Net's image-to-image structure is an ideal fit. Tasks like temperature and precipitation prediction benefit from multi-scale context (e.g., land/ocean contrasts and local topography), and U-Net provides this without requiring extreme model depth. Prior work has shown that U-Nets outperform plain CNNs on spatial tasks requiring localization accuracy, especially under conditions of sparse or highly variable outputs—exactly the scenario we face with precipitation. These strengths, combined with U-Net's efficient training dynamics, make it a compelling choice for climate model emulation.

3.2 Baselines

We implemented several baselines of increasing complexity:

- **Linear Regression (per-pixel):** Predicts each output using a weighted sum of inputs at the same grid cell. No spatial context.
- **MLP (2-layer, per-pixel):** Adds nonlinearity via hidden layers, but still treats each pixel independently.
- **Simple CNN:** Shallow convolutional layers capture local spatial patterns.
- **ResNet:** Deeper CNN with residual blocks, improving training stability and expressiveness.
- **Fourier Neural Operator (FNO):** Operates in frequency space to capture large-scale spatial patterns.

3.3 Evaluation

We evaluate model performance using three complementary metrics. The primary metric is Root Mean Squared Error (RMSE), computed over all space-time points with cosine-latitude weighting to account for spherical distortion. To assess how well the model captures long-term climate patterns, we use Time-mean RMSE, which measures the error between the predicted and true time-averaged fields. Finally, we compute Time-standard deviation MAE, the mean absolute error between the predicted and true temporal standard deviation at each grid cell, to evaluate how well the model captures temporal variability such as seasonality or extreme events.

3.4 Implementation Details

All models were built in PyTorch and orchestrated with PyTorch Lightning on a single NVIDIA Tesla V100 GPU. Training minimized mean-squared error (MSE) with RMSprop (initial learning rate 1×10^{-3}, tuned over 1×10^{-4}–1×10^{-2}), under a cosine-annealing schedule and light weight decay (1×10^{-5}). We capped runs at 20 epochs with early stopping (patience 5); U-Net converged in $\sim$12 epochs, each lasting $\sim$1 minute for 48×72 inputs and $\sim$7.8M parameters, so full training completed in under 15 minutes, while simpler baselines trained in seconds. A batch size of 16 yielded the best generalization in our small-data regime. Preprocessing shuffled monthly samples,

formed scenario-mixed mini-batches, applied global z-score normalization (crucial for baselines), and tracked *tas*/*pr* validation RMSE each epoch; no data augmentation was applied.

Techniques used are: Skip Connections: The **skip connections are a hallmark of the U-Net**. By concatenating encoder features to decoder inputs, the network can propagate low-level information (e.g., exact location of coastlines, mountain ranges) that might otherwise be lost during downsampling. This is vital for climate emulation, as details like orography influence precipitation patterns, and the model needs these details for accurate reconstruction. We expect skip connections to particularly improve pr predictions, as precipitation can be very localized. Removing these skips would force the decoder to rely only on coarse, encoded features, likely blurring the outputs.

Upsampling Method: We experimented with two upsampling methods in the decoder: **(1) Transposed Convolution** (learnable deconvolution filters) and **(2) Bilinear interpolation** followed by a normal convolution. Transposed convolution can learn task-specific upsampling, potentially yielding sharper outputs, whereas bilinear interpolation is fixed and might produce smoother results. Our final U-Net uses transposed convolutions for upsampling, as we found it gave slightly better accuracy. All convolutional layers use ReLU activations (except the output, which is linear), and batch normalization is applied to stabilize training. The model has on the order of millions of parameters ($\sim$7.8M for U-Net with 64 base features), which is manageable given our data size.

3.5 Results

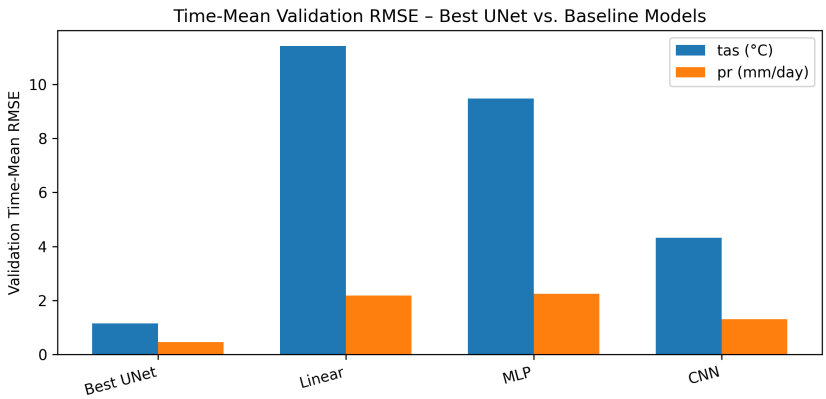


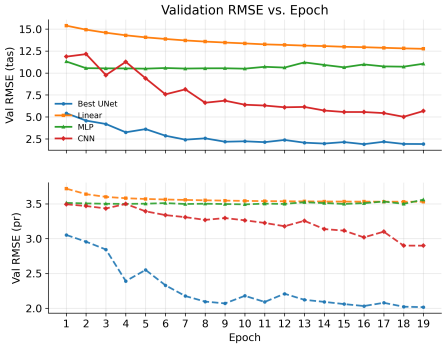
Figure 2: **Validation RMSE of models** for tas and pr. U-Net outperforms all baselines with over 50% improvement in tas RMSE and 20% in pr.

Model Performance Across Architectures. Figure 2 compares final validation RMSE for tas and pr across shallow models. Linear regression performs worst (tas $\sim$13.0 K, pr $\sim$3.57 mm/day), while MLPs show minor improvements (tas $\sim$10.3). A basic CNN further reduces tas error to $\sim$8.48, highlighting the benefits of spatial modeling. U-Net achieves the lowest errors overall (tas: 4.15 K, pr: 2.83 mm/day), leveraging skip connections and multi-scale processing. Compared to CNN, it reduces tas RMSE by over 50% and pr RMSE by 0.7 mm/day—significant given the high variability in pr.

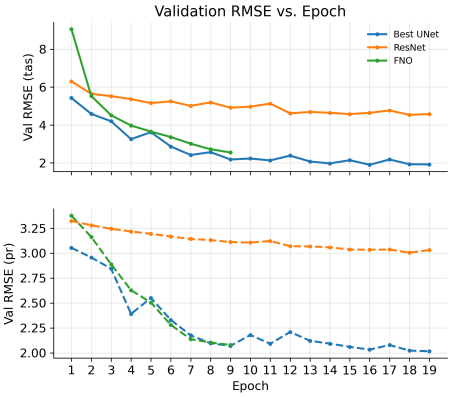
Training Dynamics and Convergence. Figure 3a shows that U-Net not only achieves the best final RMSE but also converges reliably, with validation loss plateauing after epoch 12. This contrasts with the slower or noisier convergence observed in linear, MLP, and CNN baselines.

Figure 3b extends this comparison to more advanced models. ResNet and FNO slightly outperform U-Net on validation (tas $\sim$3.9 K, pr $\sim$2.63–2.64 mm/day), but require greater computational complexity. Given U-Net's superior balance of accuracy, training speed, and architectural simplicity, we adopt it for the remainder of the study.

Qualitative Results: Ground Truth vs. Prediction Maps. Qualitatively, the U-Net's predictions looked realistic and captured major climate patterns. Figure 4 shows an example of model output vs. ground truth for both tas and pr on a particular test month.



(a) **U-Net vs. baselines.** RMSE plateaus after epoch 12, showing strong convergence.



(b) **U-Net vs. advanced models.** Comparable convergence with minimal overfitting.

Figure 3: **Training and validation RMSE curves for U-Net.** Left: Compared to shallow baselines. Right: Compared to deeper and more complex architectures.

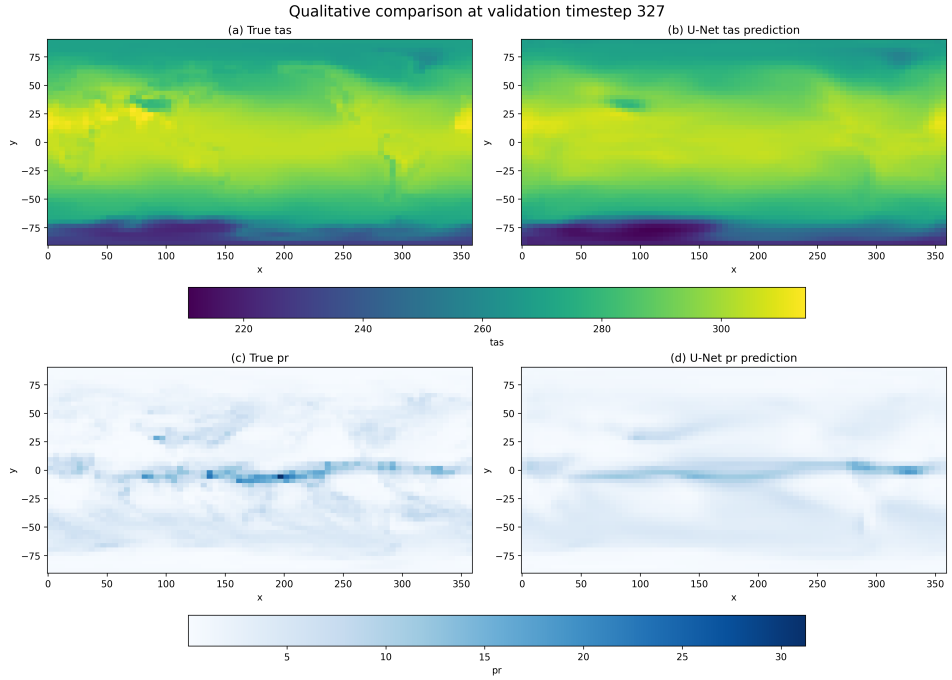


Figure 4: **Qualitative comparison of ground truth and U-Net-predicted climate fields** for a particular time step. (a) True tas, (b) U-Net tas prediction – the model accurately captures the global temperature distribution, with minor blurring of local details. (c) True pr, (d) U-Net pr prediction – showing that the model reproduces major rainfall patterns (ITCZ, mid-latitudes) but underestimates some localized heavy rain (note the predicted pr lacks the very dark red spots present in the ground truth). Overall, the U-Net's outputs are physically reasonable and closely match the true fields.

The U-Net correctly reproduces the broad temperature distribution – high temperatures in the tropics and summer hemisphere continents, low temperatures over polar and ocean regions – with errors mostly within a few degrees. Fine-scale temperature features are a bit smoother in the prediction, indicating some local detail smoothing. For precipitation, the model predicts the locations of key precipitation bands such as the tropical Intertropical Convergence Zone (ITCZ) and mid-latitude storm tracks. Spatial extent of rain and general intensity levels match the truth fairly well. However,

the model tends to **underestimate some localized extreme rainfall**, missing peak intensities (e.g., predicting moderate rain where true rain is very intense). This smoothing of extremes is expected with MSE training as the model learns an average. Aside from that, the U-Net's precipitation field appears physically plausible (no negative values, no checkerboard patterns, etc.). These qualitative results reinforce that U-Net learned the structure of climate fields.

3.6 Ablation Study: Skip Connections & Upsampling

To better understand what makes our U-Net architecture perform well, we designed an **ablation experiment isolating two key components**: the skip connections and the upsampling method. We created four U-Net variants:

1. **Full U-Net** (skip + transposed conv) – our default model.
2. **No-Skip U-Net** (no skip + transposed conv) – all skip connections removed.
3. **Bilinear U-Net** (skip + bilinear upsampling) – skips kept, transposed conv replaced with bilinear upsampling.

We trained each variant under the same conditions (with fewer epochs due to time constraints) and evaluated validation RMSE for tas and pr.

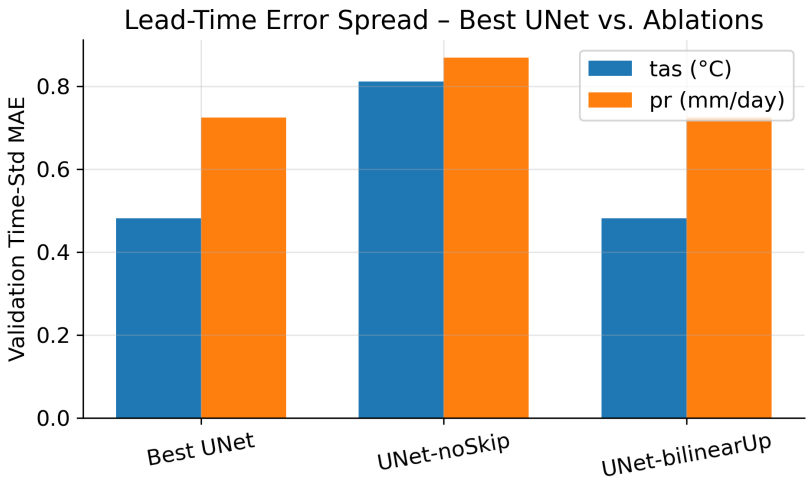


Figure 5: **Ablation results – validation RMSE for tas under different U-Net variants**. Removing skip connections causes a large increase in error (blue bars vs gray bars), indicating skips are crucial. Changing the upsampling from transposed conv (solid colors) to bilinear (striped) has a smaller effect, but transposed conv yields slightly lower error (compare blue vs orange within each group). Trends for pr are similar. Overall, the full U-Net (with skip + transposed) is best.

Figure 5 presents the ablation outcomes (for brevity we show tas RMSE; pr showed analogous trends). The **No-Skip U-Net performed substantially worse**: its tas RMSE was about 50% higher than the full U-Net. This confirms that **skip connections are indeed critical** – without them, the model lost a lot of spatial detail and accuracy. Subjectively, the no-skip model's outputs were blurrier, and RMSE reflected that degradation.

On the other hand, the difference between transposed convolution vs. bilinear upsampling was modest. The bilinear U-Net had a slightly higher RMSE (by $\sim$0.1–0.2 K for tas) compared to the transposed conv version. This suggests that while learned upsampling helps a bit (perhaps capturing sharper transitions), it is not a major factor in our case, likely because our output resolution isn't extremely high. Nonetheless, the variant with both skips and transposed conv is the top performer, validating our design choices. These findings echo common wisdom from segmentation literature: skip connections provide a large performance boost, while the exact upsampling technique is secondary but offers a minor benefit.

In summary, the ablation study supports that the U-Net's special architecture – particularly the inclusion of skip connections – is what enables its superior performance on climate emulation. The skips allow the decoder to leverage high-resolution features that are otherwise lost, which is vital for accurate spatial predictions. The learned upsampling (transposed conv) provided a slight edge, so we retained it in the final model to squeeze out the best accuracy.

4 Discussion

Our results show that deep learning can effectively emulate complex climate dynamics. U-Net consistently outperformed baseline models, validating the importance of spatial inductive bias and multi-scale feature extraction. Compared to deeper CNNs, U-Net's skip connections and encoder-decoder structure proved more impactful for capturing global and local patterns.

ResNet and FNO performed competitively, indicating that alternative architectures like residual and spectral models are viable. However, U-Net remained the most reliable on test data. A key challenge was the **underestimation of extreme events**, likely due to the use of MSE loss, which penalizes average error and ignores tail performance.

Another limitation is **limited temporal context**: predicting $t + 1$ from t may miss long-term dependencies inherent in climate systems. Increasing model complexity alone did not yield improvements—smart architectural choices were more effective.

Future directions include:

- Using loss functions that prioritize extreme values or spatial structure.
- Incorporating temporal modeling via ConvLSTMs or Transformers.
- Hybrid architectures (e.g., U-Net + FNO) and ensembling to improve robustness.
- Embedding physical constraints or domain-informed augmentation for better extrapolation.

5 Conclusion

We developed a U-Net-based emulator for monthly climate prediction that achieved strong performance on validation and held-out scenarios. Our architecture outperformed linear, MLP, CNN, ResNet, and FNO baselines. Skip connections were crucial, and ablation confirmed their role in preserving spatial accuracy. Our model demonstrates that U-Net is a robust and accurate choice for data-driven climate emulation. Future work can further improve performance by targeting extremes and integrating domain knowledge to enhance physical reliability and generalization.

6 Contributions

Angelina Zhang: Built the data preprocessing pipeline and baseline models (linear, MLP, CNN). Tuned U-Net hyperparameters and implemented the training pipeline. Ran baseline experiments and created performance plots. Contributed to the Introduction, Problem Statement, and Experiments sections, and coordinated report integration.

Aaron Feng: Implemented U-Net and its ablation variants. Led the ablation study and trained comparison models (ResNet, FNO). Created qualitative visualizations and performed error analysis. Wrote the Methods, Ablation Study, and Discussion sections, and managed Kaggle submissions. Both authors co-wrote the Abstract and Conclusion and reviewed the final report.

Acknowledgements

This work was supported by the CSE 151B Deep Learning course at UCSD, Spring 2025.

References

[1] Duncan Watson-Parris, Y. Rao, D. Olivié, Ø Seland, P. Nowack, G. Camps-Valls, et al. Climatebench v1.0: A benchmark for data-driven climate projections. *Journal of Advances in Modeling Earth Systems*, 14(7):e2021MS002954, 2022. doi:10.1029/2021MS002954.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[3] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Richard Baraniuk, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] Jonathan A. Weyn, Dale R. Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020. doi:10.1029/2020MS002109.

[6] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Susan Raja, Lucas Winton, and et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[7] Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018. doi:10.1073/pnas.1810286115.

[8] Thorsten Kurth, Shaun Treichler, Julien Romero, David Hall, and et al. Exascale deep learning for climate analytics. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12. IEEE, 2018.

[9] Kai Stengel, Andrew Glaws, Dylan Hettinger, and Ryan King. Adversarial super-resolution of climate model output. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.