CSE 151B Project Milestone Report

Zhencheng Lin*
Department of Computer Science
California University Of San Diego
La Jolla, CA 92092
zhl132@ucsd.edu

1 Project Milestone Task Description and Exploratory Analysis

https://github.com/ZhenchengLin/cse151b-spring2025-competition

1.1 Problem A[1 points]:

The goal of this project is to develop a deep learning model that emulates the output of complex, computationally expensive physics based climate models. These models project is future climate patterns under different greenhouse gas emission scenarios. Our task is to predict two key climate variables the, one the surface air temperature "tas" and second the precipitation "pr" across the globe, given a set of climate forcing variables like greenhouse gas concentrations and solar radiation as input in the model. After a little research, I can see why this is important, traditional climate models like CMIP6 is slow and computational intensive, especially when running for multiple scenarios or long time spans. A successful emulator can drastically reduce this computational cost, enabling faster climate research and real time decision making in the future using deep learning.

A deep understanding of the Coupled Model Interconversion Project Phase 6 data, including its inherent complexities, variable characteristics and the nuances of Shared Socioeconomic Pathways. forms the bedrock of this strategy. The task of emulating the intricate dynamics of the climate system necessitates a sophisticated approach that moves beyond genic deep learning techniques. The plan will detail how to establish a strong baseline model and iteratively enhance it by trying out different methods.

What we have as input is in size of $X_t \in \mathbb{R}^{\mathbf{C}\times\mathbf{H}\times\mathbf{W}}$, The C here means the number of input channels for example CO_2 etc. H, W means the spatial dimensions of latitude and longitude.

What we will produce here is $Y_t \in \mathbb{R}^{2\times\mathbf{W}\times\mathbf{H}}$ which is what we mention above 'tas' and 'pr'.

We are learning a function of $f_\theta : \mathbf{X}_t \rightarrow \mathbf{Y}_t$.

Building on top of this, we wanted to train a loss function. Some function like this:

$$\mathcal{L}_{MSE} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{H}\sum_{j=1}^{W} cos(\phi_i)\cdot||f_\theta(X_{t:i,j}) - Y_{t:i,j}||$$

Maybe above function also had to add two output as a viable.

The evaluation are:

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Preprint. Under review.

1. $\text{RMSE}_{\text{monthly}} = \sqrt{\mathbb{E}_{t,i,j}[w_i \cdot (f_\theta(\mathbf{x}_t) - \mathbf{y}_t)^2]}$

2. $\text{RMSE}_{\text{mean}} = \sqrt{\mathbb{E}_{i,j}[w_i \cdot (\bar{f}_\theta - \bar{\mathbf{y}})^2]}$

3. $\text{MAE}_{\text{stddev}} = \mathbb{E}_{i,j}[w_i \cdot |\sigma(f_\theta) - \sigma(\mathbf{y})|]$

1.2 Problem B [1 points]:

In this part I will using Jupyter Notebook to learn and explore the dataset, and report my findings with text and figures.

Here I found the answers for question one:

The Spatial Dimensions here is represented with latitude and longitude.
The module concatenates ssp126, ssp585, and ssp370, excluding valmonths for training.

SSP370:

Sptial dimensions Latitude (y) = 48, longitude (x) = 72
Approximate number of training samples (timesteps): 2703
Approximate number of validation samples (timesteps): 360
Approximate number of test samples (timesteps): 360
Each sample corresponds to a monthly time step.
Number of input variables: 5
Number of output variables: 2

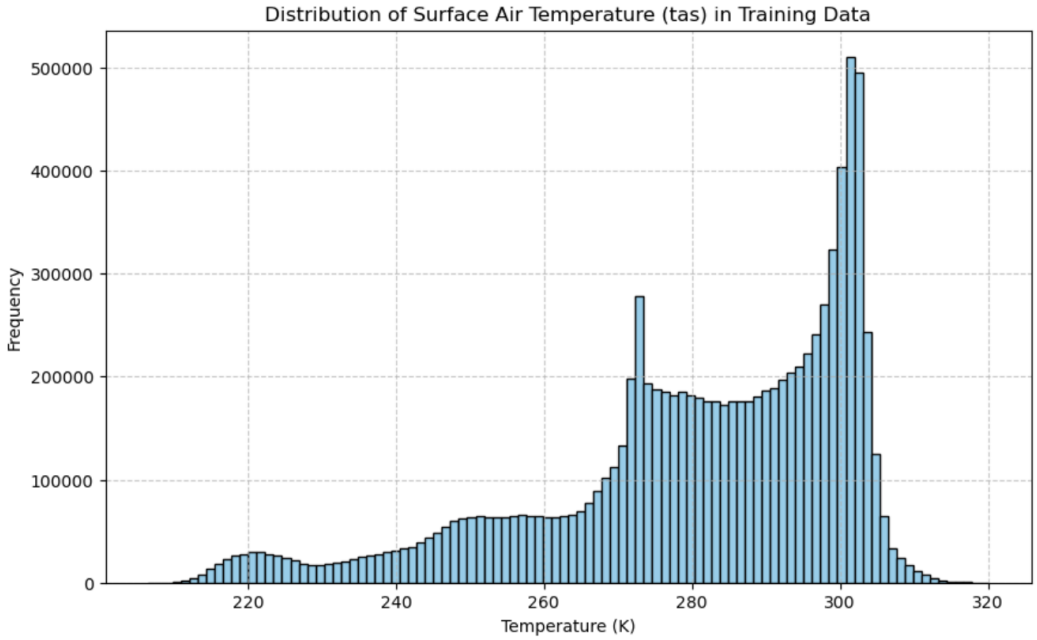
SSP126:

Approximate number of training samples (timesteps): 5406
Approximate number of validation samples (timesteps): 360
Approximate number of test samples (timesteps): 360
Each sample corresponds to a monthly time step.
Number of input variables: 5
Number of output variables: 2

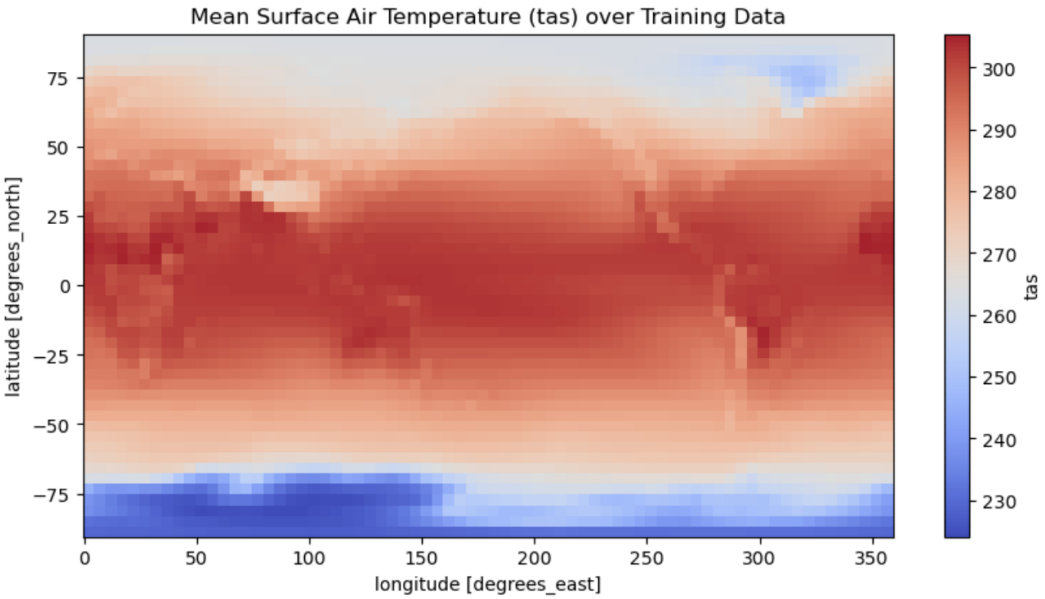
SSP585:

Approximate number of training samples (timesteps): 8109
Approximate number of validation samples (timesteps): 360
Approximate number of test samples (timesteps): 360
Each sample corresponds to a monthly time step.
Number of input variables: 5
Number of output variables: 2

Distribution of Target Variables (tas):

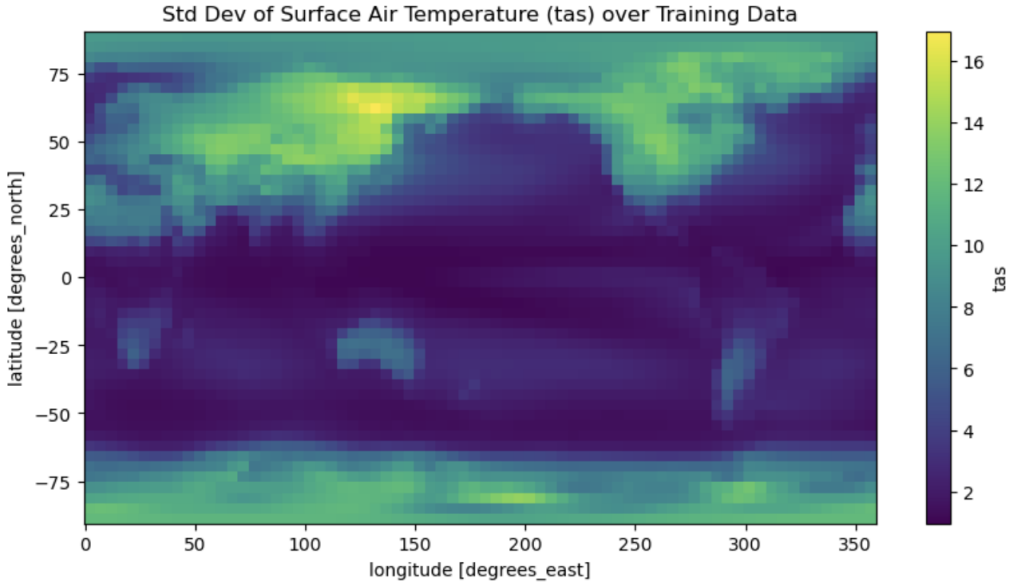


This histogram represents the distribution of surface air temperature (tas) from the training data. This graph clearly show the distribution is not normal. Instead it's skewed and there is multi modal. Multi modal pattern happened one cluster around 220-230 k meaning polar and high altitude regions, second cluster peak at 270-290 k is the normal temperate zones. Lastly we can see a strong concentration around 300 k maybe representing the desert regions.



This is the mean temperature over training data with latitude and longitude. Observing the graph, temperature near equator (0 latitude) clearly the warmest region, with mean temperatures exceeding about 300 k can also understand as this area is consistent with strong solar insolation year round in tropical zones. At mid latitudes ($\pm$ 30 to 60): temperatures there gradually decrease as you move away from the equator. At poles ($\pm$ 60+): here is a much colder regions, often below 250 k. Here we can also observe the difference between Continental and Ocean. Slightly warmer

areas over land (Sahara, India, etc), and cooler oceans in some regions due to ocean heat capacity.



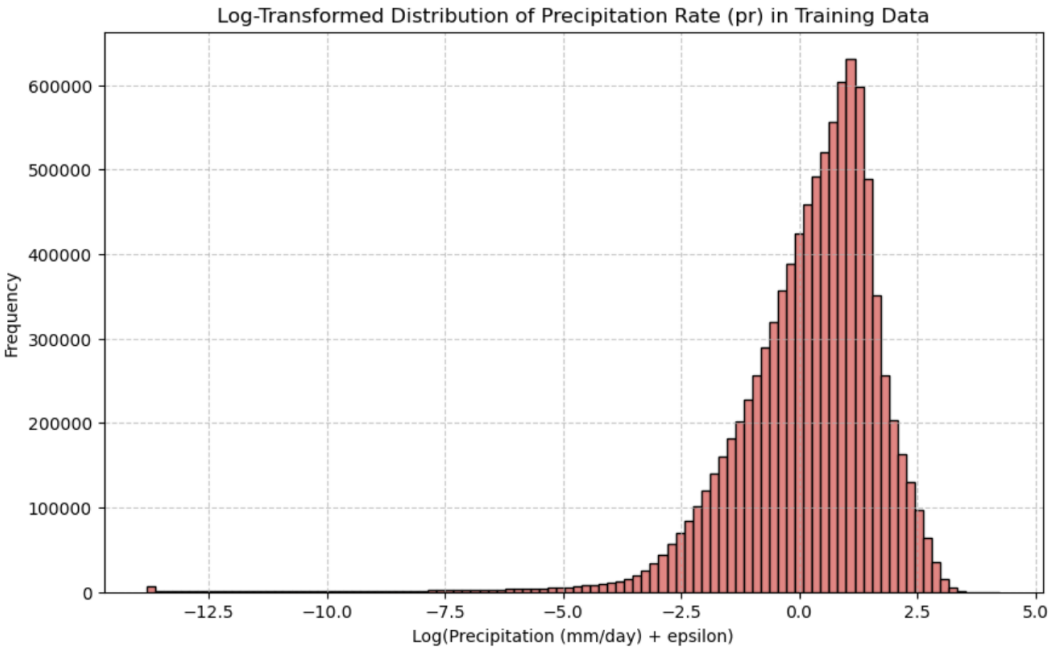
Temperature variability across the globe exhibits distinct regional patterns. In the mid-latitudes, particularly in the Northern Hemisphere, standard deviations range from 12 to 16 K, reflecting pronounced seasonal swings typical of continental climates with hot summers and cold winters. In contrast, tropical regions near the equator display very low variability, often just 1 to 3 K, consistent with their stable year-round temperatures. Polar regions, especially the Southern Ocean and Northern high latitudes, also exhibit elevated variability, largely driven by dramatic seasonal transitions between polar night and continuous summer daylight. Additionally, temperature variability is significantly higher over land—most notably in Asia and North America compared to oceans, due to land's more rapid response to temperature changes.

Precipitation (pr):

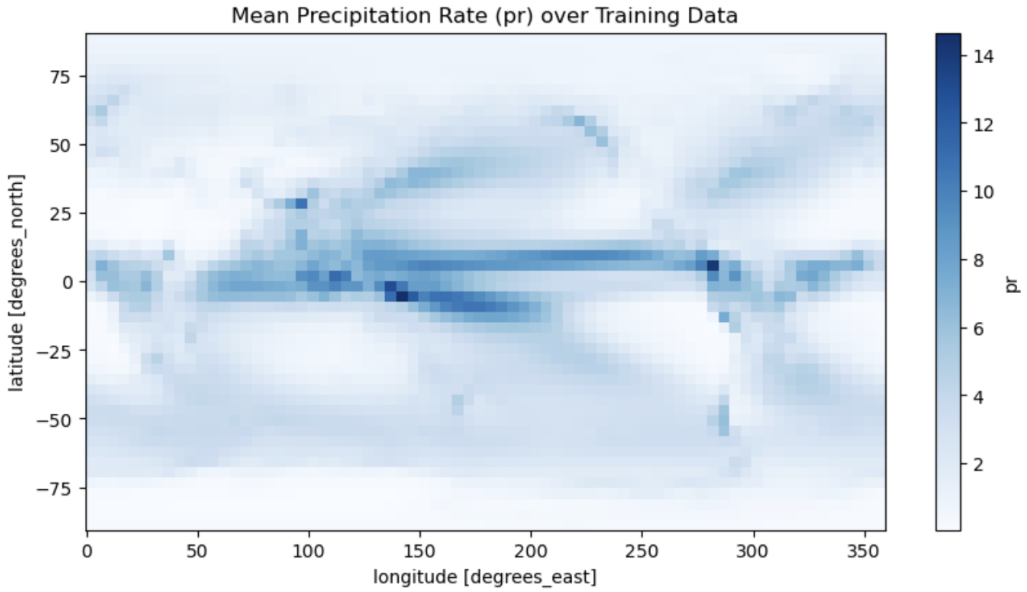
```
[TEST] tas: RMSE=290.1043, Time-Mean RMSE=290.0723, Time-Stddev MAE=3.0856, ACC=nan
[TEST] pr: RMSE=3.9675, Time-Mean RMSE=3.6310, Time-Stddev MAE=1.1323, ACC=nan
✅ Submission saved to: submissions/enhanced_kaggle_submission_20250523_211651.csv
```

Test metric	DataLoader 0
test/pr/acc	nan
test/pr/rmse	3.967543601989746
test/pr/time_mean_rmse	3.630955457687378
test/pr/time_std_mae	1.1322628259658813
test/tas/acc	nan
test/tas/rmse	290.1043395996094
test/tas/time_mean_rmse	290.072265625
test/tas/time_std_mae	3.08559513092041

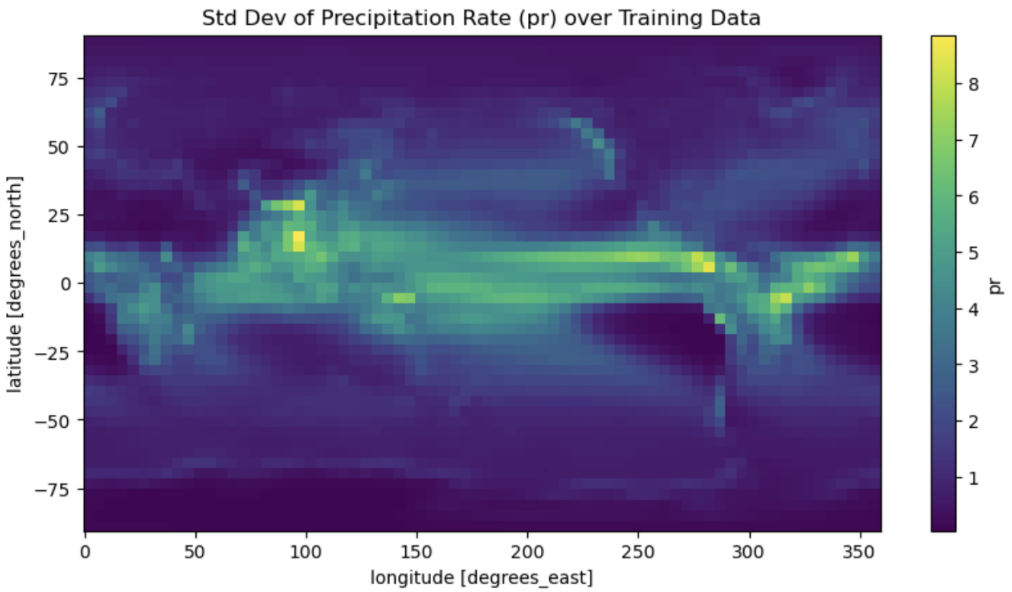
The distribution of precipitation rate (pr) in the training data is highly skewed, as evident in the histogram of raw values. The majority of the data points are concentrated near zero precipitation, with a rapid decline in frequency as precipitation increases. This results in a long right tail, indicating that while high-precipitation events do occur, they are relatively rare compared to the many instances of little or no rainfall. This kind of skewness is typical in meteorological datasets, where dry conditions are much more frequent than heavy rain.



To better understand and model this skewed data, a logarithmic transformation is applied, as shown in the second histogram. By taking the logarithm of the precipitation values (after adding a small epsilon to avoid taking the log of zero), the data distribution becomes more symmetric and bell-shaped, resembling a normal distribution. This transformation is a common preprocessing step that helps stabilize variance, reduce skewness, and improve the performance of statistical and machine learning models. Despite the transformation, a slight asymmetry remains due to the nature of the original data, but the overall shape is far more regular and manageable.



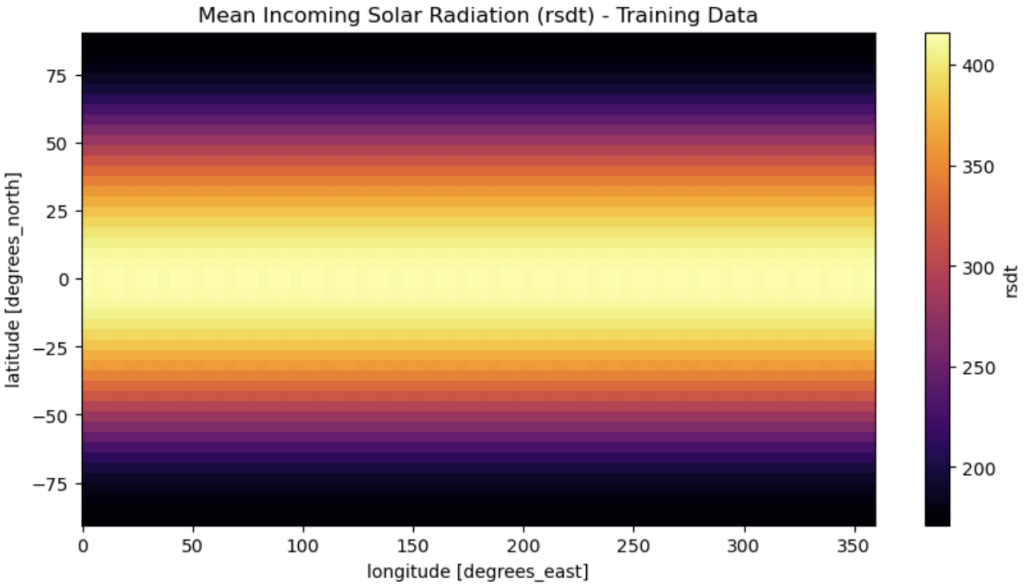
This map displays the mean precipitation rate across the globe. A strong horizontal band of high precipitation is visible near the equator, especially from about 100°E to 300°E longitude. This is consistent with the Intertropical Convergence Zone (ITCZ), a region where trade winds converge and drive intense convective rainfall. Other zones of elevated mean precipitation align with tropical monsoon regions and maritime continents such as the western Pacific, Indonesia, and parts of Central Africa and South America. In contrast, large subtropical regions such as the Sahara Desert, central Australia, and parts of the Middle East show very low mean precipitation, indicative of arid climates.



This map shows the standard deviation of precipitation, indicating where precipitation is most variable. High standard deviations are also concentrated along the ITCZ but extend further into the storm track regions of the mid-latitudes, particularly in the North Pacific and North Atlantic, as well as regions like Southeast Asia and the western coasts of Central America. These areas experience frequent or intense rainfall variability, often driven by tropical cyclones, monsoons, or frontal systems.

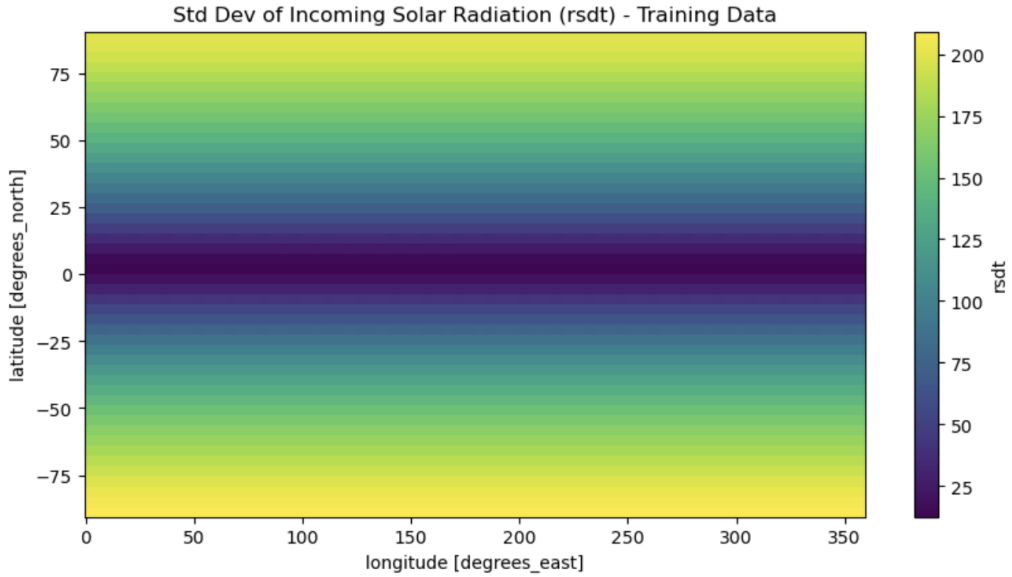
This spatial pattern underscores the non-uniformity and skewness of precipitation data—certain regions exhibit high mean and high variability, while vast areas show consistently low precipitation. This uneven distribution necessitates special normalization or transformation strategies, such as log-scaling, to stabilize variance and make the data more suitable for learning algorithms. Failing to account for these differences can lead to biased model performance, especially if the model overfits frequent dry conditions and underrepresents critical but rare heavy rainfall events.

Continuing with viewing the Distribution of Input Variables:

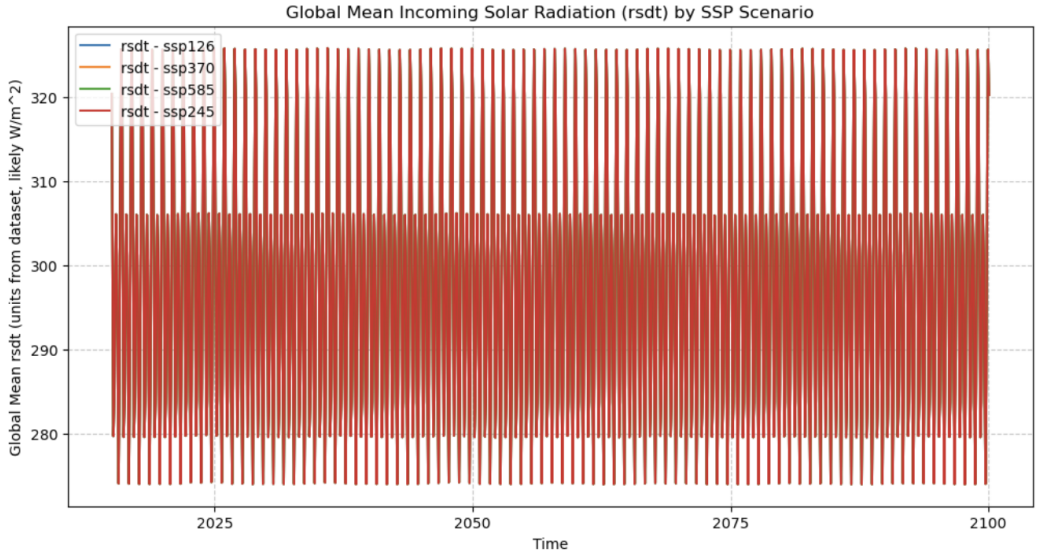


The first map shows the mean rsdt over the training period. The distribution is strongly latitude-dependent, with the highest values at the equator, tapering off symmetrically toward the poles. This is due to the angle of solar incidence: equatorial regions receive more direct sunlight year-round, while polar regions receive less, especially during their respective winters. The pattern is zonally uniform,

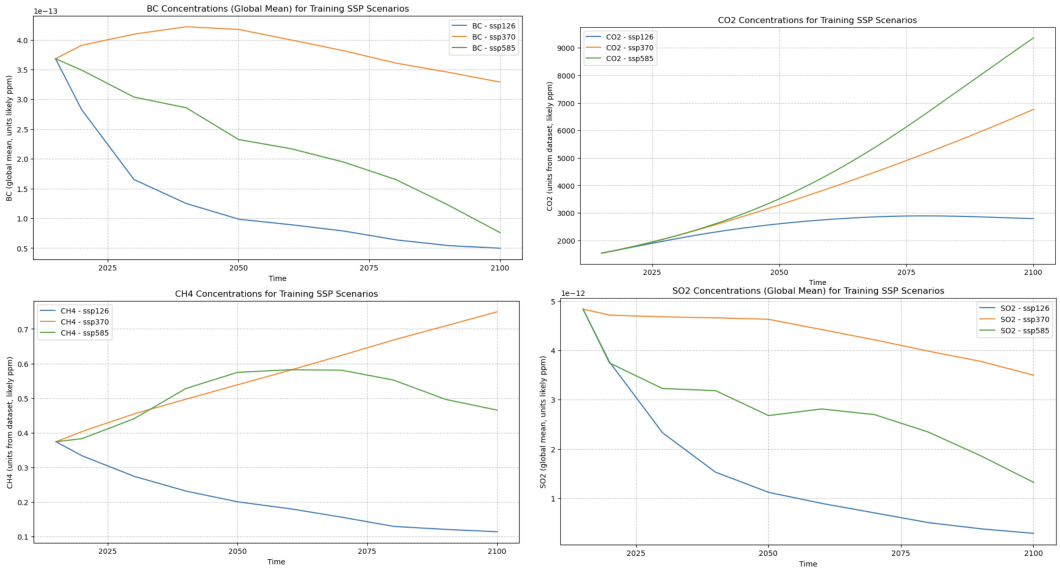
reflecting the spatial averaging over time and the dominance of latitude over longitude in determining mean solar input.



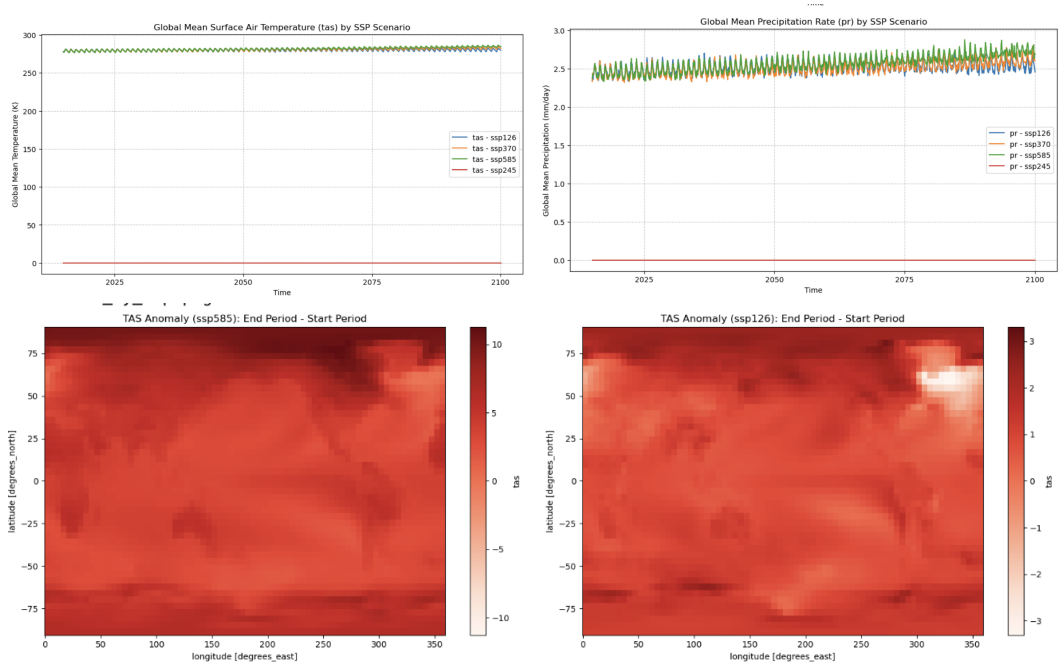
The second map presents the standard deviation of rsdt, which highlights regions with the most seasonal variability. Interestingly, the lowest variability occurs near the equator, where day length and solar angle remain relatively constant throughout the year. In contrast, standard deviation increases sharply with latitude, peaking toward the poles. This indicates a strong seasonal signal in solar radiation, driven by Earth's axial tilt, which causes large differences in incoming radiation between summer and winter at higher latitudes.



The third plot shows the global mean rsdt time series across various Shared Socioeconomic Pathway (SSP) scenarios. Despite the different future emissions trajectories, the incoming solar radiation remains nearly constant across scenarios, with minimal long-term trend and a strong seasonal cycle. This aligns with expectations, as rsdt is primarily governed by celestial mechanics rather than human-induced climate change.



While rsdt remains stable over time and across scenarios, global scalar variables show marked SSP-dependent trends. CO_2 concentrations rise steadily under SSP370 and SSP585, but level off under SSP126. CH_4 increases in high-emission scenarios and declines under mitigation. SO_2 and black carbon (BC), which contribute to aerosol cooling, generally decrease over time, especially in cleaner pathways like SSP126. These spatial gradients and scenario-based divergences highlight the importance of tailored preprocessing and scenario-aware modeling for robust climate prediction.



The provided visualizations illustrate how surface air temperature (tas) and precipitation rate (pr) evolve globally over time across different Shared Socioeconomic Pathways (SSPs), highlighting clear contrasts in projected climate outcomes. The global mean surface temperature (first plot) shows a consistent warming trend across all SSPs, but the magnitude varies dramatically. SSP585, a high-emissions scenario, results in the most significant increase, with temperatures rising steadily through 2100. In contrast, SSP126, representing aggressive mitigation, exhibits only modest warming. The annual cycle is visible in all scenarios due to seasonal variability, but the long-term upward trend in SSP370 and SSP585 reflects strong greenhouse forcing. Meanwhile, global mean precipitation

(pr) (second plot) also increases slightly over time, especially under SSP585, but the signal is more complex and less pronounced than temperature. Variability dominates, and the trends differ subtly among SSPs, indicating a more nonlinear response of global precipitation to warming.These trends are driven by changes in input forcings, particularly CO concentrations (discussed in earlier plots). CO increases most rapidly in SSP585, with values exceeding 9000 ppm by 2100, compared to a plateau under SSP126. This divergence in radiative forcing helps explain the temperature trajectory differences. Other gases like CH, SO, and BC show scenario-dependent behavior and contribute to short-term climate dynamics and regional effects.

2 Deep Learning Model and Experiment Design

2.1 Problem A [1 points]:

Our training and validation pipeline is orchestrated using PyTorch Lightning, centered around our custom EnhancedClimateDataModule and EnhancedClimateEmulationModule to implement advanced strategies from our project plan. The EnhancedClimateDataModule ingests CMIP6 climate data from the Zarr store, selecting input variables (CO2, SO2, CH4, BC, rsdt) to predict tas (temperature) and pr (precipitation) for specified SSP scenarios (ssp126, ssp370, ssp585 for training). A key preprocessing step involves climatology removal for output variables, where we subtract a 30-year (e.g., 2015-2045 from config["data"]["reference-period"]) monthly mean climatology calculated from the training data, allowing the model to focus on predicting anomalies; this climatology is added back during evaluation. For precipitation, which exhibits a skewed distribution, we apply a Yeo-Johnson transformation before standard Z-score normalization to stabilize learning. To aid temporal understanding, month and normalized year are encoded as cyclical sine/cosine features and included as scalar inputs, which are processed separately from spatial inputs like rsdt. The data is split into training, a 360-month (30-year) validation set from ssp370, and a 360-month internal test set from ssp245. Our EnhancedClimateEmulationModule employs a multi-component loss function designed to capture diverse aspects of climate fidelity: it includes a latitude-weighted Root Mean Squared Error (RMSE) as the primary accuracy term (weight 1.0), supplemented by terms penalizing errors in 10-year decadal means (weight 0.1) and decadal standard deviations (weight 0.1), and a small penalty (weight 0.01) to discourage physically implausible negative precipitation predictions. For training, we utilize an AdamW optimizer with an initial learning rate of 1e-3 and weight decay of 1e-5, paired with a CosineAnnealingLR scheduler that reduces the learning rate to 1e-6 over the max-epochs (currently 50). We train with a batch-size of 32, using 16-bit mixed-precision (precision: 16) on an automatically selected accelerator (GPU if available, e.g., GTX 2080ti). Early stopping (patience of 10 epochs on val/loss) is used to prevent overfitting. A single training epoch on this setup takes approximately 6 hours. These design choices—from sophisticated data preprocessing like climatology removal and Yeo-Johnson transformation, to a tailored multi-component loss and robust training configuration—were made to directly address the complexities of climate data, align with our strategic plan for building an advanced emulator, and optimize for the competition's multifaceted evaluation criteria.

2.2 Problem B [1 points]:

While our primary focus has been on developing an advanced architecture, our model exploration conceptually began from simpler baselines, such as the SimpleCNN outlined in the competition's starter materials, which serves as a reference for basic spatial feature extraction. Our main predictive model is the EnhancedUNetWithFiLM, a sophisticated U-Net based architecture specifically designed to learn complex spatio-temporal climate patterns and effectively integrate diverse input types. This model features a U-Net encoder-decoder structure with channel configurations [64, 128, 256, 512] for the encoder and [256, 128, 64] for the decoder, facilitating multi-scale feature learning and detailed spatial reconstruction via skip connections. A key innovation is the integration of FiLM (Feature-wise Linear Modulation) layers at each stage of the encoder and decoder; these layers allow global scalar inputs (like CO2 concentrations and engineered cyclical time features), after being processed by a scalar embedding MLP, to dynamically modulate the learned spatial feature maps by generating per-feature scaling $((\gamma))and shifting((\beta))parameters.This enables the model to condition its local spatial processing on the broader global clim$
$attention is enabled in the configuration, an Attention Block(self - attention mechanism) is incorporated into the U - Net's bottleneck to capture long - range spatial dependencies across the feature maps. The model outputs separate channels for tas and pr, with a ReLU activ$
$negativity, further supported by a dropout layer before the final 1x1 convolution for regularization. This Enhanced UNe$
$Net'sproven strength in spatial image-to-image tasks with advanced techniques like FiLM for robust scalar-spatial fusion and attention for capturing global interactions, aiming for a highly accurate and context-aware climate emulator.$

3 Experiment Results and Future Work

3.1 Problem A [1 points]:

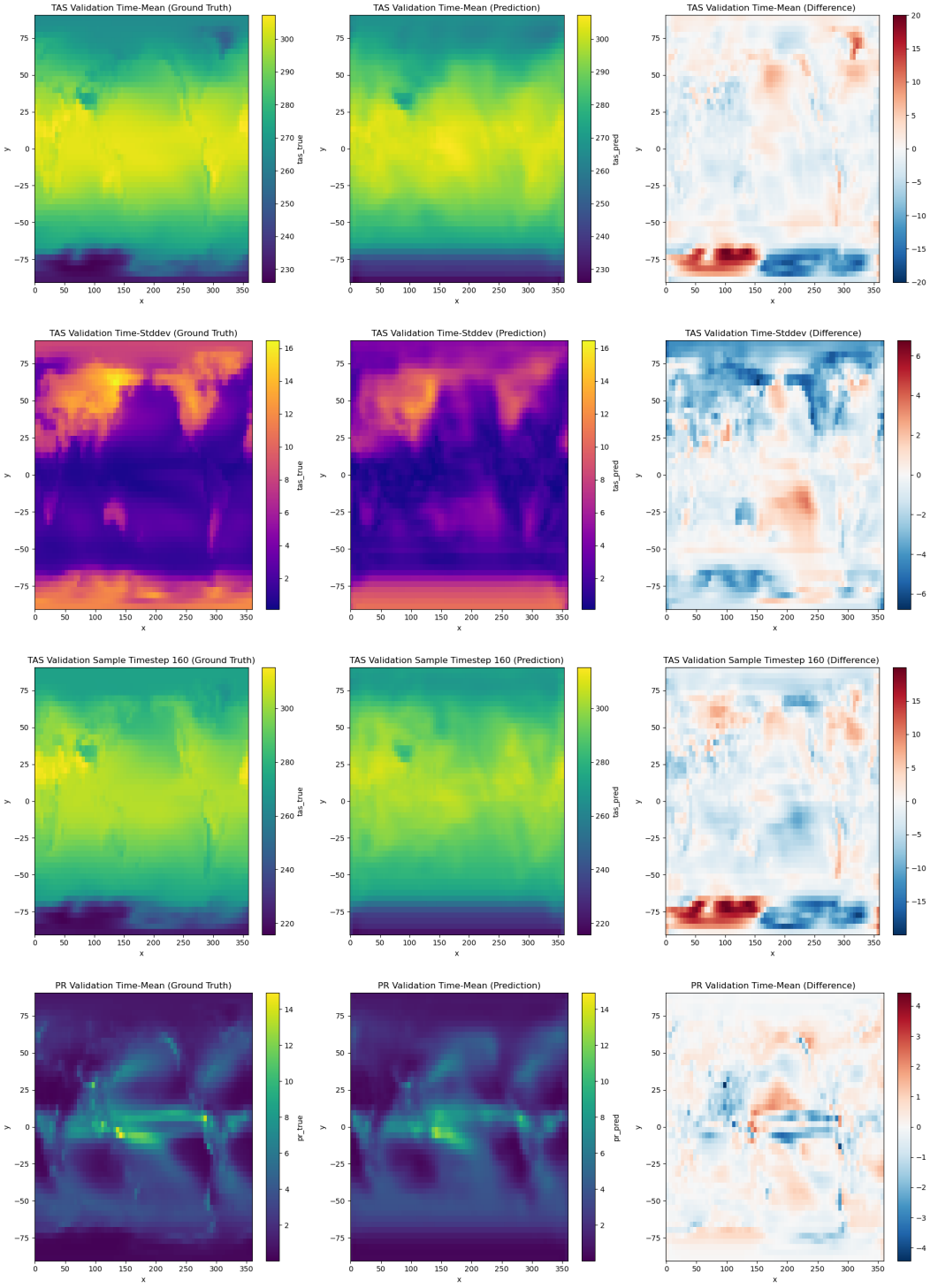
Our current best-performing model is the EnhancedUNetWithFiLM, and its evaluation involves monitoring learning dynamics, analyzing challenging samples, and assessing quantitative metrics. Training and validation loss curves (Figure X, [User: to be generated from logs, e.g., TensorBoard]) will show the learning progression; based on our console output, the validation loss (val/loss) reached a minimum of 0.9824 at epoch 1, after which early stopping was triggered due to no improvement for 10 subsequent records, indicating effective initial learning before performance on the validation set plateaued. To understand specific failure modes, we will visually inspect 2-3 validation samples with the highest prediction error using our plot-comparison function (Figures Y and Z, [User: to be generated]); this analysis for tas and pr will reveal if errors are concentrated in particular regions (e.g., coasts, high latitudes), during specific phenomena (e.g., extreme events, seasonal transitions), or if systematic biases exist. Quantitatively, for the best model checkpoint (epoch 1), the validation scores are: for tas, RMSE=1.9829, Time-Mean RMSE=0.5948, Time-Stddev MAE=0.4815, and ACC=0.9261; for pr, RMSE=2.2807, Time-Mean RMSE=0.3678, Time-Stddev MAE=1.0145, and ACC=0.5772. Our submission to the Kaggle public leaderboard yielded a score of 1.4825. The calculated test metrics from our script (e.g., test/tas/rmse: 290.1043, test/pr/rmse: 3.9675, and nan for ACCs) reflect the competition's note about potentially corrupted public test targets for ssp245, making our validation scores a more reliable current indicator of generalization performance.

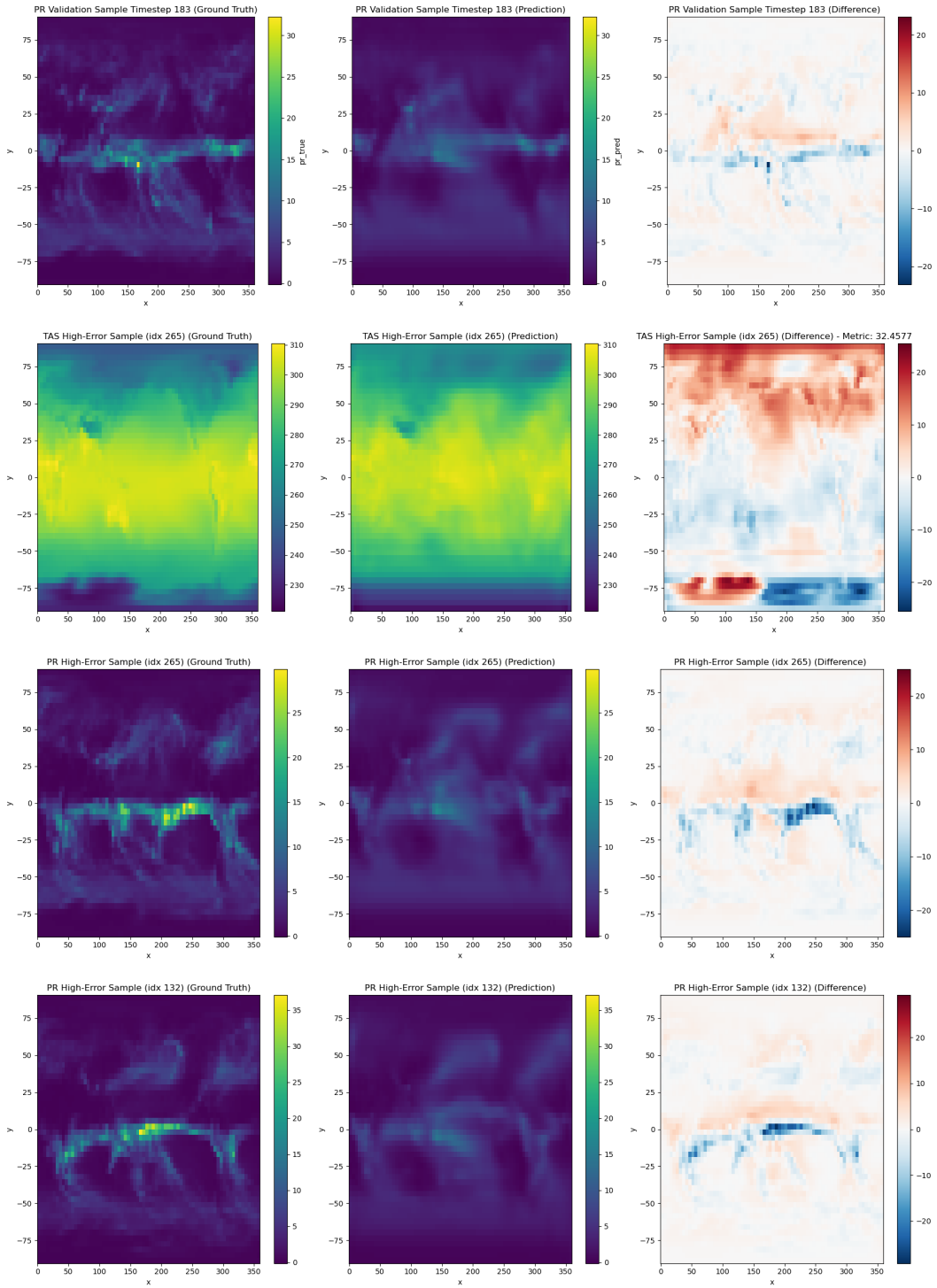
3.2 Problem B [1 points]:

We will compile a summary table comparing design iterations from baseline models to the current EnhancedUNetWithFiLM, tracking architectural changes, key hyperparameters, and their impact on validation metrics to identify effective strategies. Reflecting on the EnhancedUNetWithFiLM, its strengths appear to stem from the U-Net's multi-scale spatial processing, the effective integration of global forcings via FiLM layers, the sophisticated multi-component loss guiding towards climatologically relevant and physically plausible predictions, and the robust data preprocessing including climatology removal and Yeo-Johnson transformation for precipitation. Potential challenges include the model's computational expense, its sensitivity to hyperparameter configurations (especially loss weights), and ensuring genuine long-term statistical fidelity. Our future work will focus on several key areas to further improve performance: systematic hyperparameter optimization using tools like Optuna for architectural parameters and loss weights; exploring advanced architectures such as Transformer-based models (ViTs, hybrid CNN-Transformers) or further U-Net refinements; enhancing data utilization by incorporating all CMIP6 ensemble members; implementing true autoregressive multi-step training for improved long-term stability; continued loss function refinement, potentially with adaptive weighting or auxiliary heads for decadal statistics; and finally, developing model ensembling techniques by combining predictions from several strong, diverse models. This iterative approach, focusing on robust methodologies and systematic experimentation, will guide our efforts to build a leading climate emulator.

```
[TEST] tas: RMSE=290.1043, Time-Mean RMSE=290.0723, Time-Stddev MAE=3.0856, ACC=nan
[TEST] pr: RMSE=3.9675, Time-Mean RMSE=3.6310, Time-Stddev MAE=1.1323, ACC=nan
✅ Submission saved to: submissions/enhanced_kaggle_submission_20250523_211651.csv
```

Test metric	DataLoader 0
test/pr/acc	nan
test/pr/rmse	3.967543601989746
test/pr/time_mean_rmse	3.630955457687378
test/pr/time_std_mae	1.1322628259658813
test/tas/acc	nan
test/tas/rmse	290.1043395996094
test/tas/time_mean_rmse	290.072265625
test/tas/time_std_mae	3.08559513092041


TAS Validation Time-Mean (Ground Truth)
TAS Validation Time-Mean (Prediction)
TAS Validation Time-Mean (Difference)
TAS Validation Time-Stddev (Ground Truth)
TAS Validation Time-Stddev (Prediction)
TAS Validation Time-Stddev (Difference)
TAS Validation Sample Timestep 160 (Ground Truth)
TAS Validation Sample Timestep 160 (Prediction)
TAS Validation Sample Timestep 160 (Difference)
PR Validation Time-Mean (Ground Truth)
PR Validation Time-Mean (Prediction)
PR Validation Time-Mean (Difference)
x
y
tas_true
tas_pred
pr_true
pr_pred



Figures presents a comprehensive evaluation of the model's performance on the validation set by comparing predicted and ground truth spatial fields for surface air temperature (tas) and precipitation rate (pr). For each variable, the first column displays the ground truth statistics, the second shows the model predictions, and the third visualizes the difference (prediction minus ground truth). The time-mean tas plots indicate that the model successfully captures the global pattern of surface temperature, with warmer values near the equator and colder values at the poles, though some regional biases persist as seen in the difference map. The standard deviation maps demonstrate that the model broadly reproduces year-to-year temperature variability but tends to underestimate or overestimate variability

in certain areas, as highlighted by blue and red regions in the difference plot. A sample timestep comparison confirms that the model can capture realistic spatial patterns at individual moments, though some systematic regional errors remain. For precipitation, the model accurately predicts the overall structure of mean rainfall, particularly the high precipitation band along the equator, but difference maps reveal that errors are largest in regions with complex or highly variable rainfall such as the tropics. Overall, these plots illustrate that while the emulator performs well in replicating the large-scale climate features, there remain localized areas—especially in precipitation—where model predictions could be further improved.

Over all I am trying everything that says is good online