GRU-ving to the Rain

By: BIG DATA

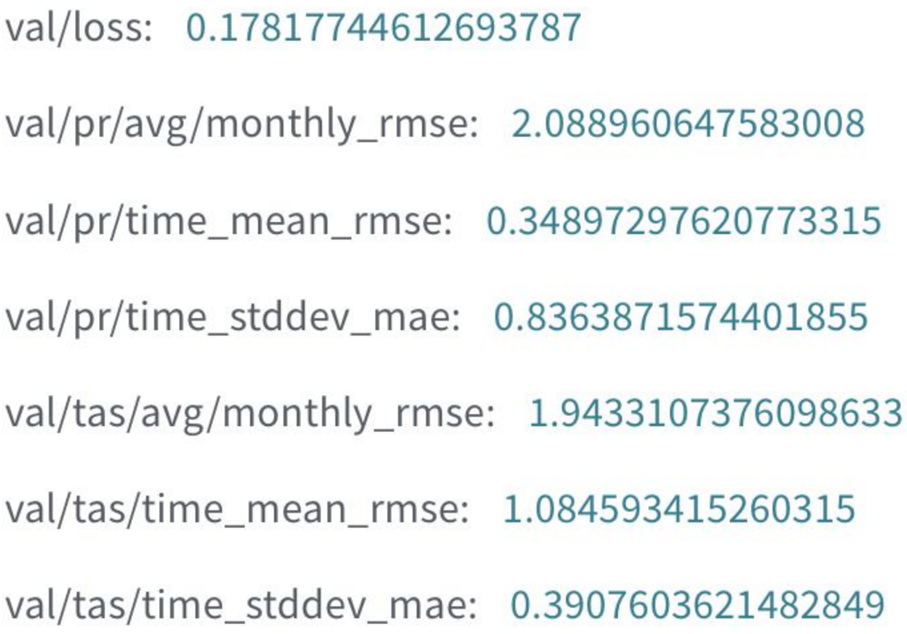
Kevin Jacob, Taanish Reja, Jose Barcelo, Josh Lee

https://github.com/Kevin12J/Climate-Emulation-Model

Summary

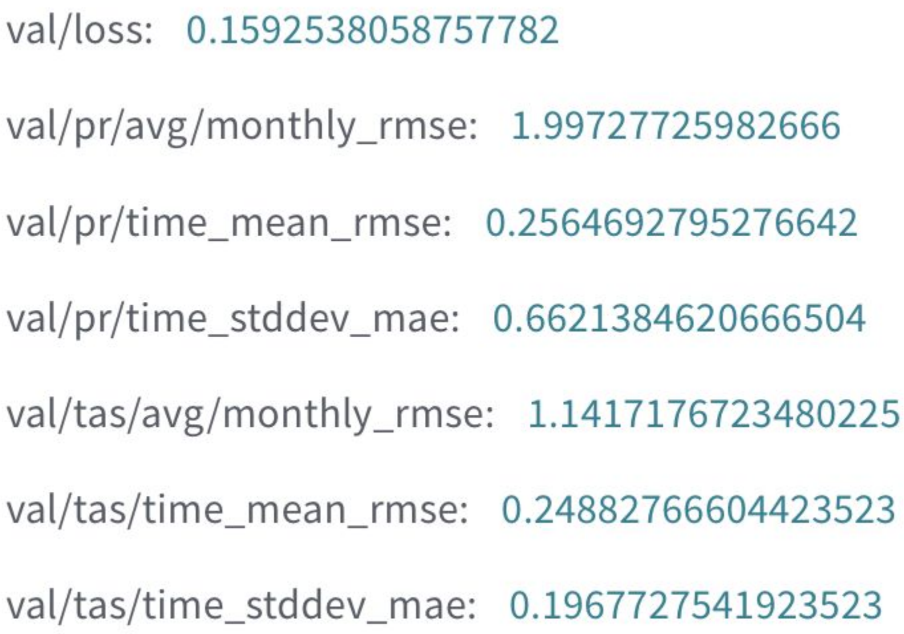
- **Hybrid CNN-GRU for spatiotemporal climate modeling**
- **Key innovations**
  - Combining spatial pattern recognition (CNN) with temporal sequence learning (GRU)
  - Dice loss (precipitation)
  - Edge-detecting loss
- **Performance improvements over baseline models**
  - The initial model we trained was based on the starter model, but we replaced the 2D convolutions with 3D convolutions to include the time dimension

First Model



```
val/loss:  0.17817744612693787

val/pr/avg/monthly_rmse:  2.088960647583008

val/pr/time_mean_rmse:  0.34897297620773315

val/pr/time_stddev_mae:  0.8363871574401855

val/tas/avg/monthly_rmse:  1.9433107376098633

val/tas/time_mean_rmse:  1.084593415260315

val/tas/time_stddev_mae:  0.3907603621482849
```

Final Model



```
val/loss:  0.1592538058757782

val/pr/avg/monthly_rmse:  1.99727725982666

val/pr/time_mean_rmse:  0.2564692795276642

val/pr/time_stddev_mae:  0.6621384620666504

val/tas/avg/monthly_rmse:  1.1417176723480225

val/tas/time_mean_rmse:  0.2488276660442352

val/tas/time_stddev_mae:  0.1967727541923523
```

Key Words

CNN-GRU

Spatiotemporal Modeling

Composite Loss

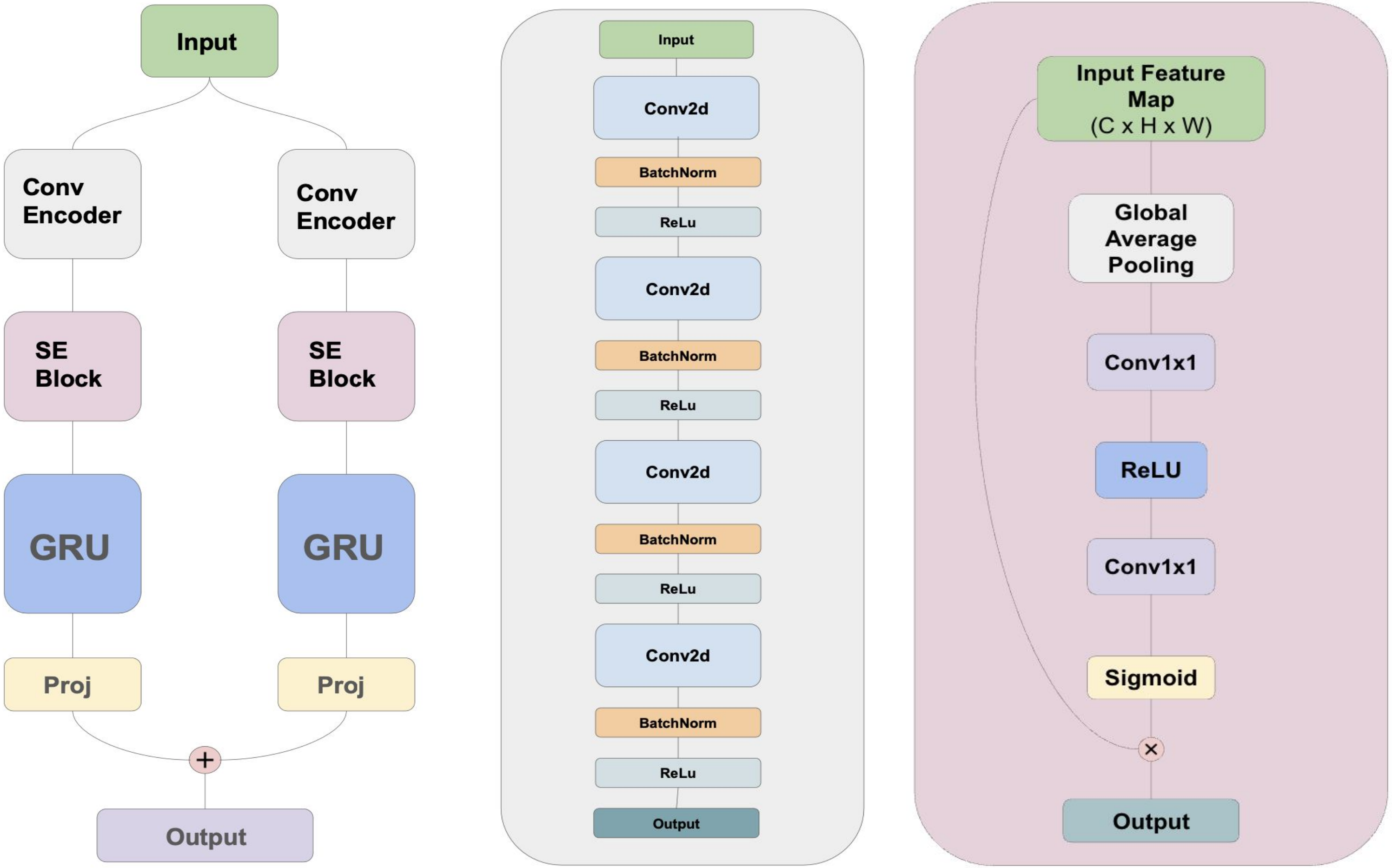
Two pathway

Introduction

Challenge

- **Climate emulation challenge**
  - Predicting temperature and precipitation from forcings **efficiently**
- **Why CNN-GRU?**
  - Spatial correlations
  - Temporal dependencies
  - Efficiency

Methodology


Input
Conv Encoder
Conv Encoder
SE Block
SE Block
GRU
GRU
Proj
Proj
+
Output
Input
Conv2d
BatchNorm
ReLu
Conv2d
BatchNorm
ReLu
Conv2d
BatchNorm
ReLu
Conv2d
BatchNorm
ReLu
Output
Input Feature Map
(C x H x W)
Global Average Pooling
Conv1x1
ReLU
Conv1x1
Sigmoid
×
Output

Training Overview

- **Model Parameters:** 352 M
- **Batch Size:** 32
- **Target Member IDs:** 0,1, and 2
- **Learning Rate:** 0.0001
- **Optimizer:** AdamW with weight decay of 0.01
- **Learning Rate Scheduler:** Cosine Annealing LR

Data Processing

- **Input Variable Engineering**
  - Seasonal encoding
    - Accounts for hemisphere
    - Normalizes months to 0 to 2pi range
    - Cyclical (uses cosine)
  - Improves models ability to capture seasonal changes (i.e. in precipitation)

- **Window based training**
  - 12 months windows
  - Capture medium term dependencies

Deep Learning Model

CNN Component

- Spatial feature extraction layers

- Kernel sizes

- Global vs local pattern recognition

GRU Component

- Temporal sequence modeling

- Hidden state dimensions

- Bidirectional vs unidirectional

- Integration with CNN features

Engineering Tricks (loss function)

- Average of Dice Loss (precipitation) and Gabor Loss (temperature)
    - Gabor Loss - Capture difference in fine edges between predictions and targets
    - Dice loss[1] - Handle imbalance[2] in precipitation

[1] Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations (Sudre et al)

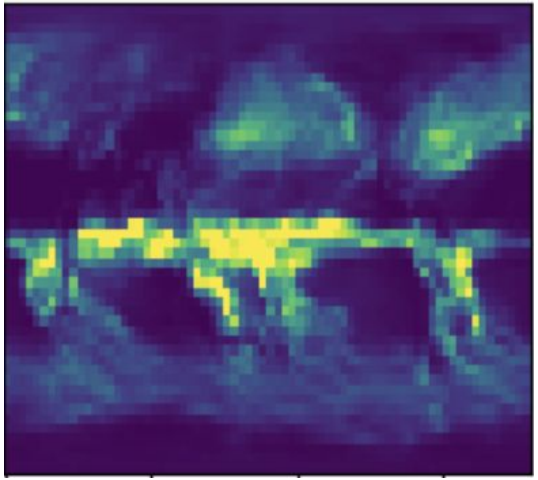
[2] A study on loss function against data imbalance in deep learning correction of precipitation forecasts (You et al)

Engineering Tricks (loss function)

Dice Loss

- Similarity metric used to calculate overlap

$$L_{\text{dice}} = 1.0 - 2.0 \cdot \frac{\mathbf{x}_f \cdot \mathbf{y}_f}{\mathbf{x}_f^T \mathbf{x}_f + \mathbf{y}_f^T \mathbf{y}_f + \epsilon}$$

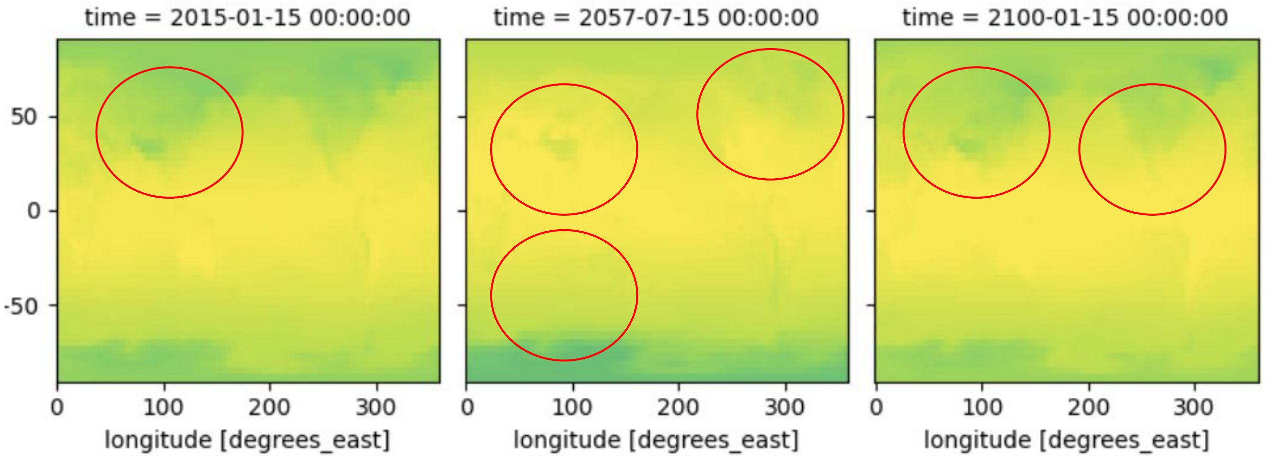


(Prediction and target vectors are flattened**)**

1. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations (Sudre et al)

Gabor Loss

- Goal: Change training objective to capture fine edges/patterns in data targets
- Convolve gabor filters[3] (edge-detection filters) over predictions and targets
- Return difference in filters



[3] 3D Segmentation with Fully Trainable Gabor Kernels and Pearson's Correlation Coefficient (Wong et al.)

Experiments

Experiment 1: Comparing performance with and without seasonal encoding

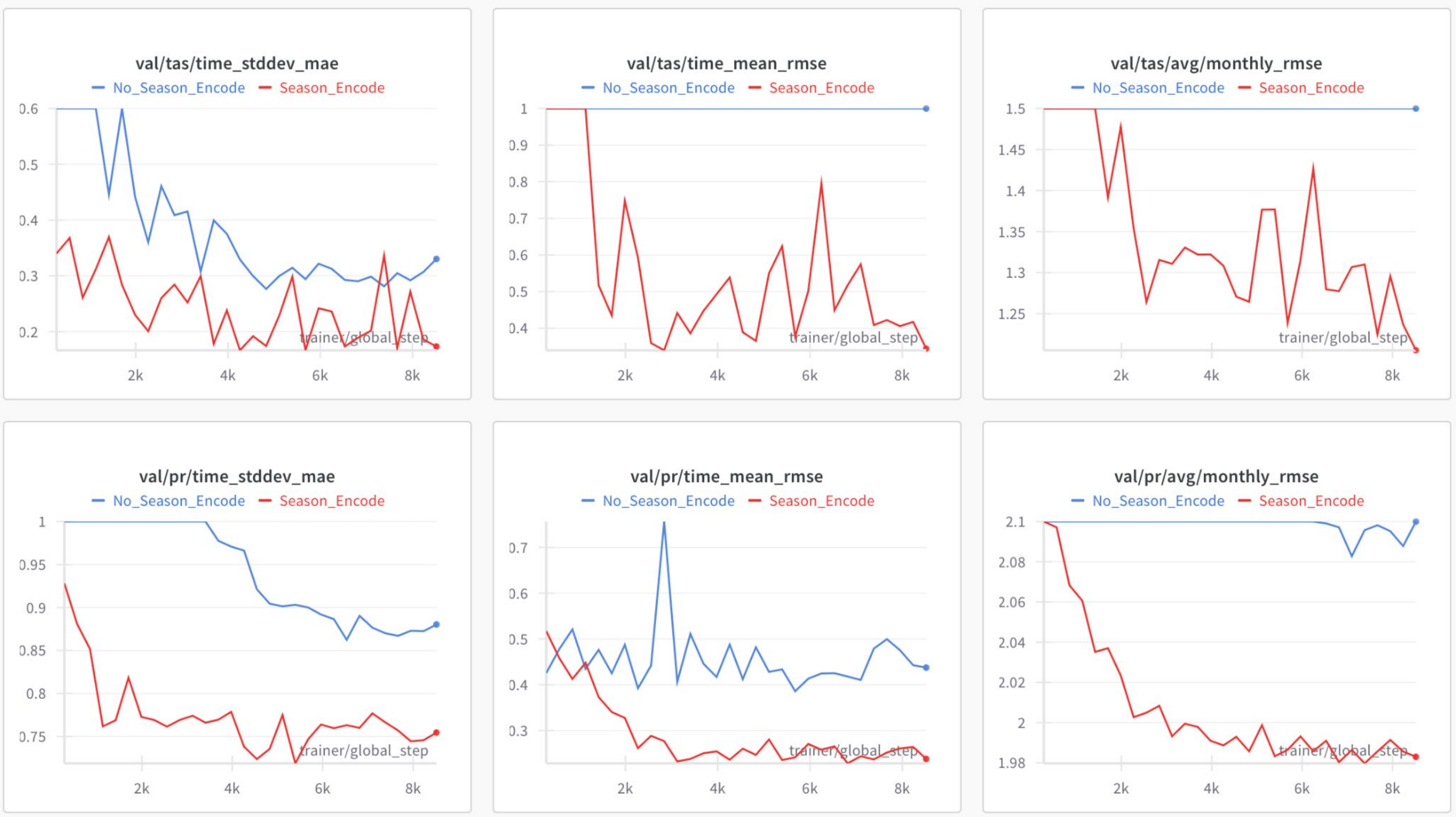
This experiment was performed using an earlier mode from our project

🔴 With Seasonal Encoding
🔵 Without Seasonal Encoding

Results

- The experiment shows that the seasonal encoding significantly improved the model performance
- The improvement was greater in tas and a potential reason for this is that surface temperature often follows a cyclical pattern with seasons

Experiment 1: Comparing performance with and without seasonal encoding



Experiment 2: Modifying weights for dice loss and MSE in PR loss calculation

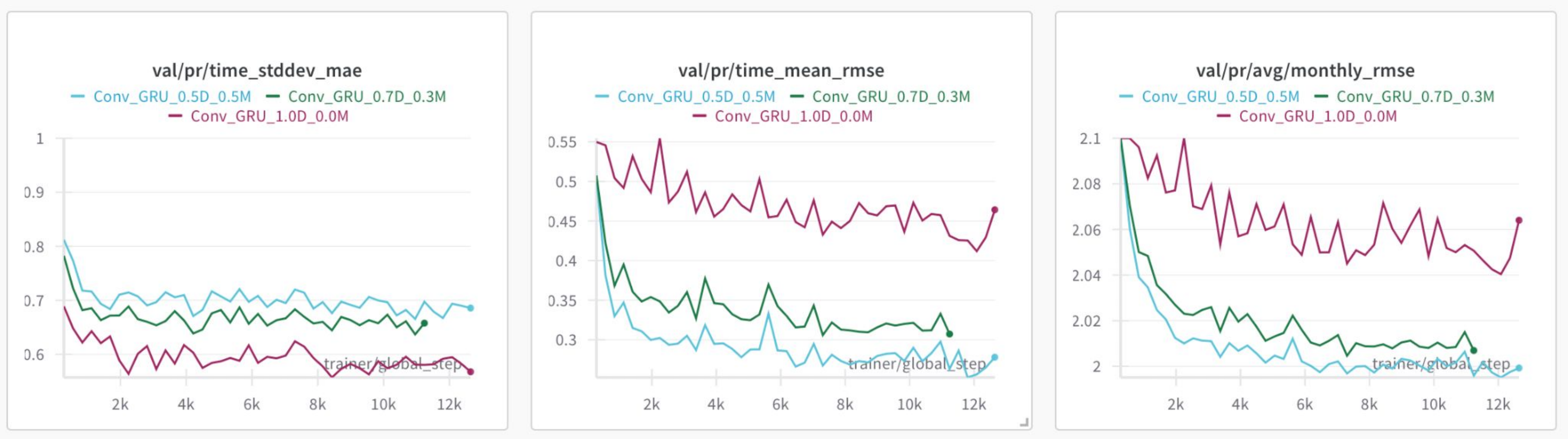
In our training process, the loss for pr was calculated using a weighted sum

$$\alpha \times \mathrm{DiceLoss} + \beta \times \mathrm{MSELoss}$$

We performed 3 different training runs with the following weights

- 🔴 $\alpha = 1.0 \quad \beta = 0.0$
- 🟢 $\alpha = 0.7 \quad \beta = 0.3$
- 🔵 $\alpha = 0.5 \quad \beta = 0.5$

Experiment 2: Modifying weights for dice loss and MSE in PR loss calculation



Results

- The experiment shows that placing a higher weight on the dice loss lowered the pr time standard deviation mse at the expense of the time mean rmse and monthly mse
- A good balance of weights was to use 0.5 for both dice and mse loss

Discussion

Strength/Weakness/Improvement

- **3 Strengths**
  - **Hybrid Architecture:** The CNN-GRU combination effectively captures both spatial climate patterns (through CNN's convolutional layers) and temporal dependencies (through GRU)
  - **Custom Loss Functions:** Dice loss for precipitation and Gabor (edge-detecting) loss for temperature addresses the unique characteristics of each climate variable
  - **Seasonal Encodings:** Engineered features helps the model capture periodic climate patterns, particularly important for precipitation strong seasonal variability
- **3 Weaknesses**
  - **Hyperparameter Tuning:** We did not explore making certain parameters learnable and had to manually test a lot of different hyperparameters manually per run
  - **Model Size:** Trained extremely large models on A6000’s, A5000’s, RTX 4090’s. Model may not perform well on constrained hardware
  - **Limited Val Dataset:** Val dataset consisted of exclusively high emission simulations. Model may not generalize well on outlier scenarios

What have you learned

- **Architecture vs. Data Quality:**
  - While sophisticated architectures can help, at some point, the architecture itself does not play a key role, instead the quality of input features and loss function design has a larger impact.
    - While the Conv-GRU won, Conv-LSTM, U-Net and Transformers were all viable models **(all would’ve placed third)**
- **Domain Knowledge Integration**
  - How to effectively search for and adapt ideas from climate science literature to improve our model. We integrated the ideas from papers from ML climate researchers to solve problems as they came up.
- **Trial and Error & Learning your Data**
  - Never assume a strategy is good or bad
  - The distribution and pictures of your data are invaluable for finding the margins where models can be improved

Future Work

Below are some future adjustments we believe can further improve our model

- We can explore our loss function weights more
- Experiment with alternative spatial encoders and output heads
  - Replacing our CNN encoder with Transformers or U-Net encoders could better capture both local weather patterns and global climate connections.
- Experiment with a greater variety of data sequences
  - Testing different temporal window sizes (6, 18, 24 months) could help the model capture both short/long-term climate trends more effectively.

References & Credit

- Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations (Sudre et al.)
- A study on loss function against data imbalance in deep learning correction of precipitation forecasts (You et al.)
- Squeeze-and-Excitation Networks (Hu et al.)
- 3D Segmentation with Fully Trainable Gabor Kernels and Pearson's Correlation Coefficient (Wong et al.)
- LLMs