Machine Learning for Weather and Climate

Yifei Ding

April 22, 2025

Background and Preliminaries

Weather Forecasting

Given a dataset $\mathcal{D} = \{X_i\}_{i=1}^{N}$ of history weather data, forecast future weather conditions $X_T \in \mathbb{R}^{V \times H \times W}$ with initial conditions $X_0 \in \mathbb{R}^{V \times H \times W}$.

T: target lead time, V: number of input and output variables, $H \times W$: spatial resolution.

Three major approches:

- *direct forecasting*
- *continuous forecasting*
- *iterative forecasting*

ClimaX

Object: design and train a foundation and efficient model for weather and climate concerning atmosphere.

Model architecture:

- *Variable tokenization*: tokenize the input into $V \times (H/p) \times (w/p) = V \times h \times w$ patches, and embed each input patch os size p^2 to a vector of dimension D;
- *Variable aggregation*: perform a cross-attention operation for each spatial position in the $h \times w$ map;
- *Transformer*: extend a standard ViT to generate the output tokens.

ClimaX

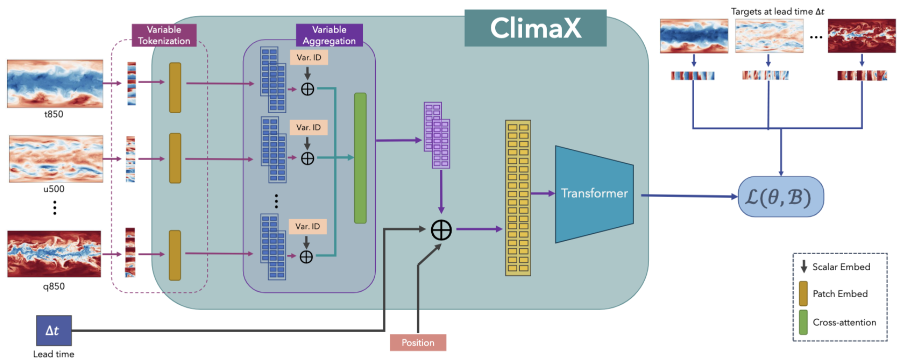

Figure 1: Pretraining phase of ClimaX

Stormer

A simple, skillful and competitive model for weather forecasting.

Training: *randomized dynamics forecasting objective*

$$\mathcal{L}(\theta) = \mathbb{E}_{\delta t \sim P(\delta t), (X_0, X_{\delta t}) \sim \mathcal{D}} \left[||f_\theta(X_0, \delta t) - \Delta_{\delta t}||_2^2 \right]$$

With pressure-weighted loss and multi-step finetuning, the final loss function is

$$\mathcal{L}(\theta) = \mathbb{E} \left[\frac{1}{KVHW} \sum_{k=1}^{K} \sum_{v=1}^{V} \sum_{i=1}^{H} \sum_{j=1}^{W} w(v) L(i) (\hat{\Delta}_{k\delta t}^{vij} - \Delta_{k\delta t}^{vij})^2 \right].$$

Stormer

Inference: use two inference strategies, *homogeneous* and *best m in n*, and obtain the final prediction by averaging the individual predictions.

Model architecture:

- apply *ViT* for this task.
- Weather-specific embedding: *variable tokenization* and *variable aggregation*.
- Stormer Transformer block: replace standard layer normalization with *adaLN* and regress parameters with an *one-layer MLP* from the embedding of δt.

ClimateLearn

Propose a PyTorch library implementing a range of functionalities for benchmarking of ML models for weather and climate.

- Tasks: weather forecasting, downscaling, climate projection.
- Datasets: ERA5, Extreme-ERA5, CMIP6, PRISM.
- Models: traditional baselines, deep learning models.
- Evaluations: forecasting metrics, downscaling metrics, climate projection metrics, visualization.

Acknowledgement

Thank you!