Revisiting ClimaX: A Foundation Model for Climate and Weather

Kevin Shi[a], Enzo Uchiha[a], Jen Tang[a]
[a]Computer Science, New York University

Abstract

ClimaX (Nguyen et al. (2023)) from Microsoft is a flexible and generalized deep learning model based on transformer architecture that attempts to address broad spatio-temporal climate forecasting challenges. Through pretraining on diverse and heterogeneous climate datasets from CMIP6, ClimaX already demonstrates exceptional capabilities in capturing complex climate dynamics. In our study, we explore enhancing and modifying the influence of climaX. The model is initially adapted to predict extended lead times during its pretraining phase, further underscoring its ability to capture long-term climatic pattern and trends. The model is followed by a direct finetuning on era5 with specific emphasis on regional climate data. We also finetune the model based on the original pretrain scheme. Our results indicate that the modified ClimaX model has slight improvement on test accuracy with around 1% absolute increase.

Author roles: For determining author roles, please use following taxonomy: https://credit.niso.org/.

1. Kevin Shi: Investigation, Methodology/Design, Software, Review and Editing
2. Enzo Uchiha: Investigation, Original Draft
3. Jen Tang: Investigation, Review and Editing

1 Context and Motivation

Climate prediction is one of the most important task globally; accurate predictions are crucial for decision making, resource management and preparation for extreme weather conditions. Traditional weather prediction models rely on numerical simulations from physics equations or simulations of large systems using differential equations, which are related to the flow of energy and matter based on known physics of different Earth systems. These models are known to have weaknesses and limitations both at long-term and short-term horizons and take numerous compute time.

On the other hand, making predictions given certain features is a typical machine learning task which climate prediction could benefit from. Thus, the emerging models like ClimaX are designed to learn from diverse historical datasets and then fit spatial-temporal variables. Current sensors have plenty of weather data, but the above pure mathematical methods mentioned have a hard time fully processing this data. ClimaX combines two of these: it uses the incredible amount of available past data along with machine learning computations to model climate and weather. The use of deep learning models has quite a few benefits which listed below:

- **Computational Efficiency:** ClimaX aims to offer a more efficient alternative to performing resource-intensive computation.
- **Generalization:** Climax has the ability to deal with heterogeneous datasets which can learn from different sources; traditional methods require repetitively reformations on physics-based equations.
- **Self-Supervised Learning:** By employing self-supervised learning, ClimaX can be pretrained on large volumes of unlabeled climate data, capturing the complex relationships within the data.
- **Flexibility for Fine-tuning:** After pretraining, ClimaX can be fine-tuned for various downstream tasks, including those involving atmospheric variables and scales not seen during pretraining.
- **Performance Improvements:** ClimaX demonstrates superior performance on benchmarks for weather forecasting and climate projections compared to existing data-driven baselines.

In this project, we aim to improve the generalization capabilities for ClimaX by using different pretraining targets. Unlike the original ClimaX methodology, we , we ask the model to make predictions using the lead time predictions instead of relying on the specific timestamp. The model would adapt to both long-term and short-term climate predictions. Additionally, our changed pretraining offers the model an improved ability to easily transfer predictions on different tasks. The dual approach of extending lead times in pre-training coupled with regional fine-tuning in the subsequent phase allows ClimaX to not only maintain its broad predictive capabilities but also to refine its outputs based on localized data. This leads to improved forecast reliability and relevance,

particularly in regions with unique weather patterns not fully captured by global models. By leveraging localized datasets for fine-tuning, ClimaX can adapt to regional peculiarities, offering forecasts that are more actionable for local decision-makers and stakeholders. The benefits of these modifications would have significant real-world impacts. Enhanced lead time predictions allow for earlier warnings and extended preparation periods which are crucial for effective disaster management and mitigation strategies. Additionally, the high precision and localized focus of the fine-tuned forecasts enables better resource allocation, tailored policy-making, and improved climate adaptation strategies at a regional level.

2 Dataset description

Repository location

- CMIP6 DOI:10.5281/zenodo.3888445
- ERA5 DOI:10.24381/c8jw8xh8

Repository name

- ESGF Data Node: https://esgf-data.dkrz.de/search/cmip6-dkrz/
- British Atmospheric Data Centre (BADC):https://help.ceda.ac.uk/article/4801-cmip6-data#Data%20Access:%20CEDA
- ECMWF data Centre: https://confluence.ecmwf.int/display/CKB/ERA5%3A+data+documentation
- Copernicus Climate Data Store(C3S) https://cds.climate.copernicus.eu/cdsapp#!/home

Object name filename:

- *year*.nc follow the above name format

Format names and versions

- netCDF
- npz

Creation dates

- Start: 1850-01-01
- End: 2015-01-01

Dataset creators

- **Veronika Eyring**-She has been one of the leading figures in CMIP6, serving as a co-chair for the project. Her role has been pivotal in guiding the project's direction and coordinating the efforts of participating teams across the globe.
- **Gerald Meehl**-Although not directly a project leader for CMIP6, Gerald Meehl is a experienced (senior) scientist known for his substantial contributions to climate modeling and the CMIP project series. His work has had a profound impact on the understanding of climate change and the development of climate models.
- **Paul Berrisford**-He has been involved in various capacities, including his work on the development and application of global weather models and reanalysis projects such as ERA5. His expertise spans areas such as data assimilation, which is a crucial component in the production of reanalysis datasets that combine observational data with model simulations to create a comprehensive picture of the atmosphere's past states.
- **Peter Bauer**-A senior scientist at ECMWF, Peter Bauer has played a key role in the ERA5 project and related weather forecasting initiatives. His expertise has been invaluable in advancing the field of numerical weather prediction and climate modeling.

Language

- English

License

- CC by 4.0 https://creativecommons.org/licenses/by/4.0/
- ECMWF https://www.ecmwf.int/en/copyright-attribution

Publication date

- 2025-xx-xx

3 Method

3.1 Backbone Model

Vision Transformer(Dosovitskiy et al. (2020)) is used as the main backbone in the entire climax module. It is a transformer based model that absorbs the sequential data from image and trains on a large amount dataset to overcome the inductive bias that typically talked about in Convolutional Neural Networks. The core idea of ViT is to input images as sequences of fixed patches like how we treat word tokens in NLP. Each image is divided into numerous patches and each patch is encoded to a fix length embedding that then enters a standard transformer. The self-attention part is actually similar to what happens in sentence sequences; it captures the relations between patches within an image and summarizes the information in a cls token to visual token in this sense.

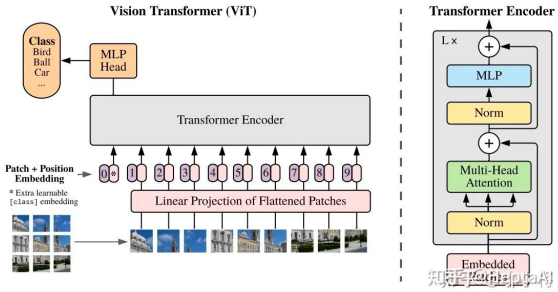


Figure 1: vit structure

The use of ViT in Climax particularly for handling weather and climate datasets from CMIP6 offers several benefits given the characteristics of such datasets. These dataset consist of spatial and temporal structured data, which makes them somewhat analogous to images. Therefore, using a ViT is particularly suitable for climate tasks. Self-attention also plays an important role in capturing relations and information across different features with long-range dependencies. Recent success on generative models shows that transformer based models are very scalable; with increasing amount of data and increasing number of parameters(model capacity), we could expect unlimited performance improvement instructed by scaling law. We have everyday global climate data generated over the years which the ViT model takes advantage of. Last but not least, ViT is successful in transfer learning so that finetuning the model could be done on other datasets for region-based forecasting.

3.2 Aggregate Variables

Suppose that we have in total number of V variables per sample. And each image is decomposed to $h * w$ patches. Each patch will be linearly transformed into a dimension D embedding. Therefore the input fed into the model has the shape $V * h * w * D$. In order to reduce the computation cost, we cloud eliminate the first dimension V by introducing a learning query vector in each patch position. This query vector would attend all the embeddings across all the features in the same patch position. And in essence, each patch position will have a fused embedding that absorbs the information from all the different features. The resultant features have the shape $h * w * D$.

3.3 Model Specifications

Parameter	Value
Embedding Dimension	512
Number of Heads	16
MLP Ratio	2.0
Drop Path Rate	0.1
Dropout Rate	0.1
Number of Layers	4
Normalization Type	norm_layer
Decoder FC Depth	2

Table 1: Vision Transformer Model Parameters

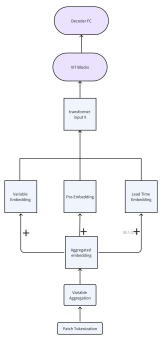


Figure 2: Graph For Entire Module

3.4 Training Specifications

In the original paper, they use a self-supervised pretrain task where the model would predict future climate statuses or states. The states could be any point from 6 hours and 18 hours in terms of the lead time which means any point within a week. The loss function is a geometrically modified MSE loss-latitude-weighted mean squared error.

$$L = L(i)\left(\tilde{X}^{v,i,j}_{t+\Delta t} - X^{v,i,j}_{t+\Delta t}\right)^2, \tag{1}$$

$$L(i) = \frac{\cos(\text{lat}(i))}{\frac{1}{H}\sum_{i'=1}^{H}\cos(\text{lat}(i'))}, \tag{2}$$

In our implementation, we change the predict targets to lead time. The overall model is the same as the original one. We make following modifications:

- Omit the lead time embedding sum pooling step with input features since we are predicting it.
- Fully connected decoder layer is changed from feature-like shape output to one single scalar output in order to adapt the leading time predictions. Specifically we have code differences like this.

```
self.head = nn.ModuleList()
for _ in range(decoder_depth):
    self.head.append(nn.Linear(embed_dim, embed_dim))
    self.head.append(nn.GELU())
self.head.append(nn.Linear(embed_dim, len(self.default_vars) * patch_size**2))
# our impelementations
self.pred_head = nn.ModuleList()
self.pred_head.append(nn.Linear(embed_dim * embed_dim, 1))
self.pred_head.append(nn.GELU())
self.pred_head = nn.Sequential(*self.pred_head)
```

- Direct return on vanilla MSE loss in our module.
- We use a smaller portion of the CMIP6 dataset and omit a few features like "geopotential, specific_humidity" .etc to accelerate the training.

During the finetune phase, we finetune two downstream tasks

- Finetune the entire model.
- Replace the embedding layers and predictions head; And then we conduct both a finetune on other components and parameters frozen on them (only train on lead time embedding layers and image head in code).

Table 2: Parameters Set for Pretrain

Parameter	Value
Learning Rate (lr)	5×10^{-4}
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.95
Weight Decay	1×10^{-5}
Warmup Steps	10,000
Max Steps	200,000
Warmup Start Learning Rate	1×10^{-8}
Minimum Eta (η_{min})	1×10^{-8}
Batch Size	8
Number of Workers	1

Table 3: Parameters Set for Finetune

Parameter	Value
Learning Rate (lr)	5×10^{-4}
Beta 1 (β_1)	0.9
Beta 2 (β_2)	0.99
Weight Decay	1×10^{-5}
Warmup Epochs	10,000
Max Steps	100,000
Warmup Start Learning Rate	1×10^{-8}
Minimum Eta (η_{min})	1×10^{-8}
Batch Size	128
Number of Workers	1

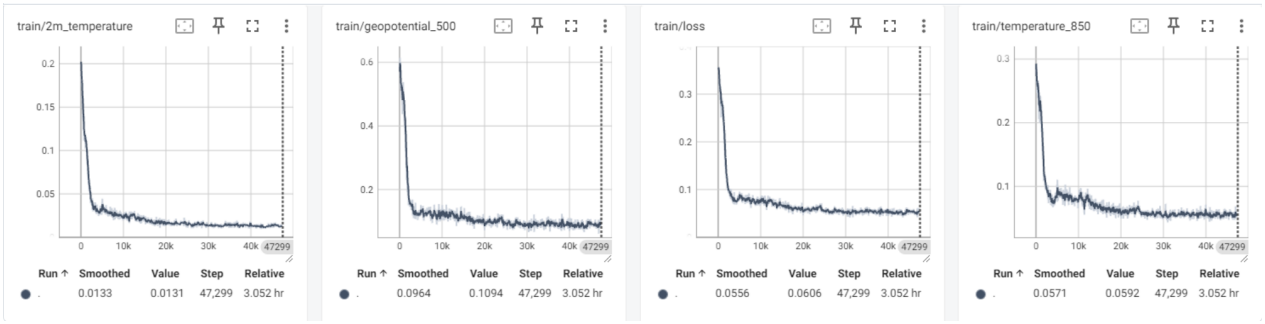
3.5 Pytorch Lightning Trainer

The entire training including pretrain and finetune is realized under the pytorch lightning module. It is a comprehensive package that wraps up the pytorch basic functionality but takes over most of the training details. Users only need to accomplish the model architecture and loss functions. Once the model is defined and the only thing we need to do is forward the model and calculate the loss in the training step. Everything else is taken care of by the lightning module including optimizer step and loss backwards. It has more advanced functions like callback tools, early stopping and logging. The training loss and step accuracy can be automatically logged by the Lightning Trainer and plotted out in the tensorboard. In climaX, we have model completed in general torch.nn.Module and we configure all the checkpoint setting, weights initialization and evaluation steps in lightning module.

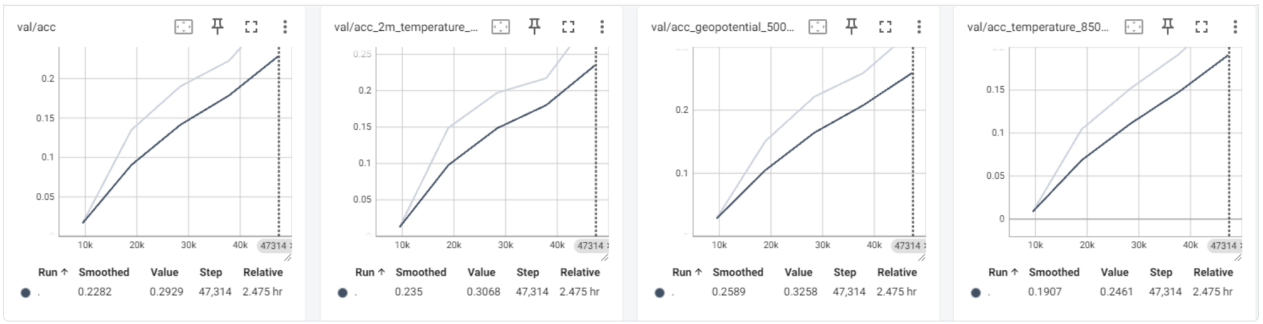
4 Results

We have logged the training loss, validation loss, validation accuracy and test accuracy in the tensorboard. Firstly we have the forecasting results on the dataset(ERA5) under the official ClimaX Pretrain. Then we have the forecasting results on the dataset(ERA5) under our pretrain.

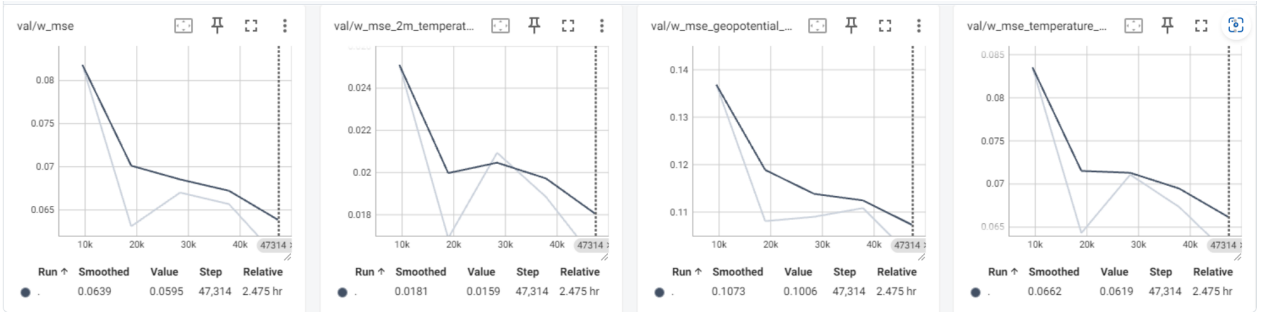
- Finetune training loss under Original Pretrain



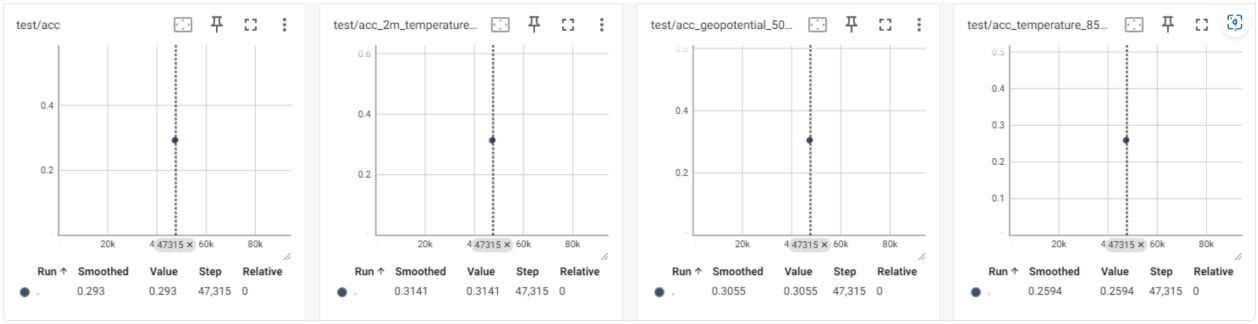
- Finetune validation accuracy under Original Pretrain



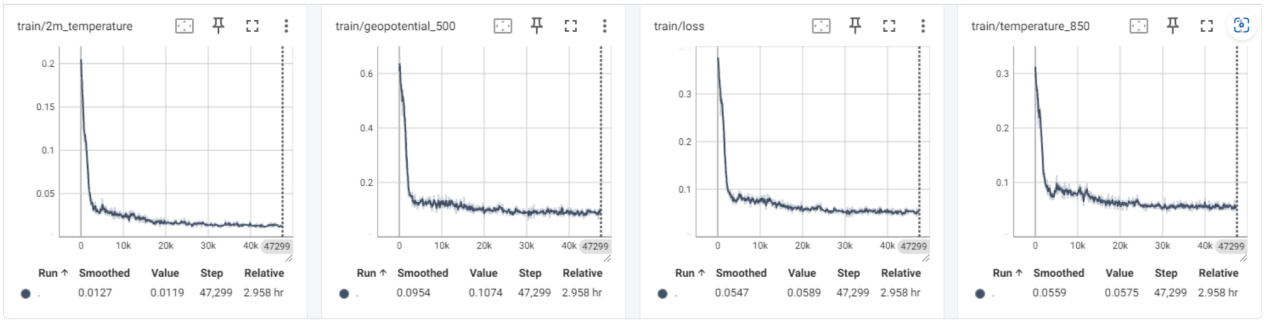
- Finetune validation loss under Original Pretrain



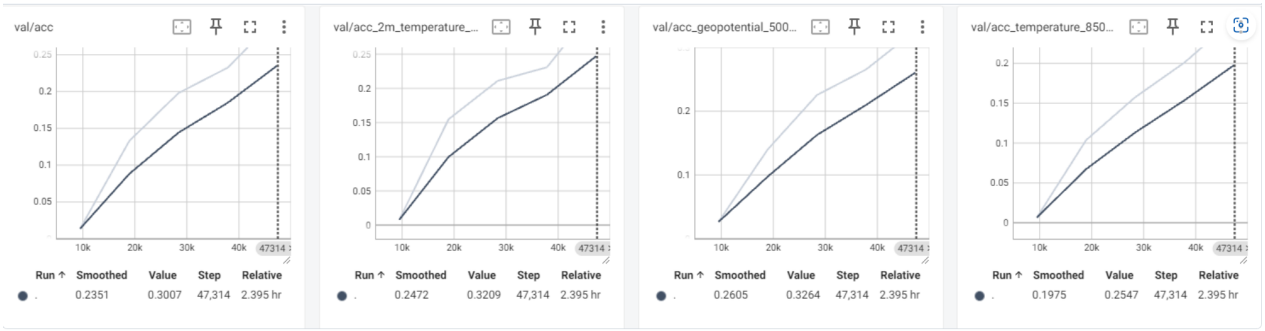
- Finetune Test accuracy under Original Pretrain



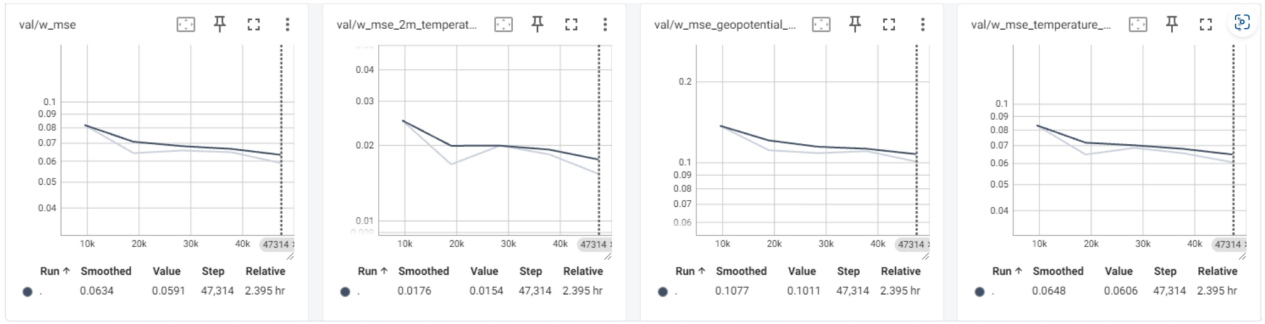
- Finetune training loss under Our Pretrain



- Finetune validation accuracy under Our Pretrain



- Finetune validation loss under Our Pretrain



- Finetune Test accuracy under Our Pretrain

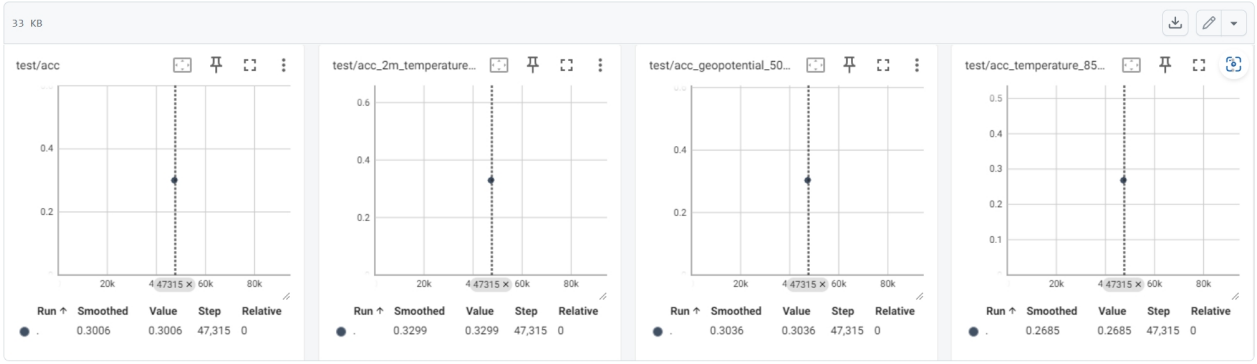


Table 4:Accuracy Comparison

Model	Test Accuracy	Validation Accuracy
Our Model	0.3006	0.2351
Original Model	0.293	0.2282

From the table above, we could see that our pretrain model offers **slightly higher** accuracy in both validation and test set.

5 Implications/Applications

As a research question about alternative pretrain methods, we had some inquiries about multi-task training. We decided to choose completely different pretraining targets with the exception of the self-training. Due to compute limitations, the dataset we use is smaller than than the original studies; however we still see an improvement in model performance. We hypothesize that if we combine these two trainings together in the first place and fully utilize the entire dataset, we could achieve better results.

The second takeaway from this project is that pretraining has once again proven its effectiveness for downstream tasks across various scientific computing disciplines. Although its efficacy has been demonstrated countless times in the fields of Computer Vision (CV) and Natural Language Processing (NLP), the Climax project has provided yet another testament to the value of pretraining.

We acknowledge the power of the Transformer architecture, which excels under various types of heterogeneous data and diverse features. Transformers can quickly adapt and learn different features, providing promising predictive outcomes. From the perspective of the ClimaX project, if the features are diverse, rich, and not sparse, we should consider the Transformer architecture as a priority in more fields, such as finance and healthcare. For data similar to images, the ViT (Vision Transformer) can be explored, while for purely discrete data, the BERT approach may be worth trying.

ClimaX significantly improves weather and climate forecasting, offering profound practical applications and broader implications. It improves the accuracy and lead times of weather forecasts, crucial for disaster preparedness, policy-making and sustainability planning. ClimaX also plays a critical role in public health by enabling proactive responses to weather-related health risks, assists in designing resilient urban infrastructure, and supports environmental conservation efforts by predicting ecological changes. Overall, ClimaX is a transformative tool across various sectors, improving decision-making and operational efficiency in response to climate dynamics.

6 Conclusion

In our research we have already see the importance and strong transfer ability about ClimaX; we propose another pretraining scheme to test its effectiveness through transfer learning. We have not modified much in the overall model backbone and preprocessed data. The results are optimistic because we improved the downstream performance despite only using a portion of dataset as the original paper. In the future, we could try out more pretrain tasks such as masking part of the features to improve the generalization for models. With more incoming data, we could also plot the scaling law for ClimaX similar to what researchers do in large language models. Once we know the marginal benefit for Climate predictions, we would get the optimal amount of training resources and be able to obtain the confidence of better results with an even larger dataset.

Acknowledgements

Saining Xie - Helped develop original research idea
Tung Nguyen - Assisted with environment setup and helped troubleshoot data processing issues.

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... others (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., & Grover, A. (2023). Climax: A foundation model for weather and climate.

Code base

The GitHub Link to the code base is through this link: ML-ClimateResearch
Pretrained model checkpoints are hosted on Google Drive here: Model Checkpoints