AtmoRep: Representation learning for atmospheric dynamics

Student: ***Pol Garcia Recasens***

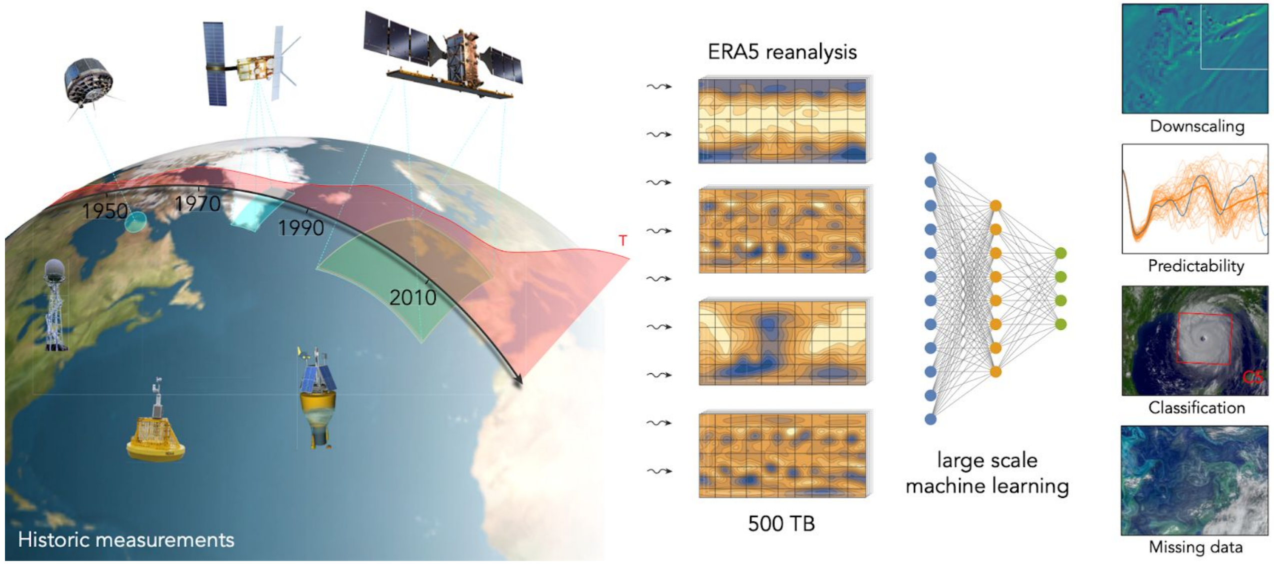
Supervisors: ***Sofia Vallecorsa, Ilaria Luise**, **Christian Lessig***

Openlab Summer Student Programme 2022

Overview

Introduction

- Large-scale ML-based 4D model (representation learning) on spatio-temporal atmospheric data
- Help further downstream applications like hurricane tracking or hyper-resolution simulations
- Flexible architecture that can be extended to more fields (e.g. ocean reanalysis)

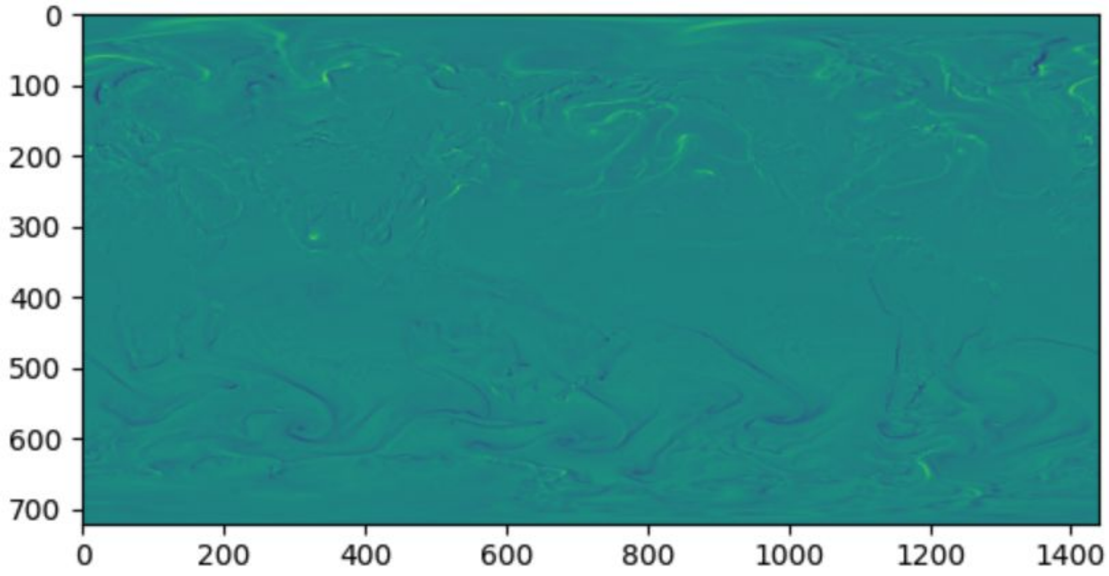


Overview

Data

- ERA5 reanalysis dataset (ECMWF)
- **Space**: 1440 x 720 x 137 vertical layers
- **Time**: 24 time steps per day for 365 days for 70 years (1970 onwards)
- **Physical fields**: vorticity, divergence, temperature, geopotential
- 500 TB -1000 TB of data

Atmospheric vorticity (ERA5)



Overview

Architecture

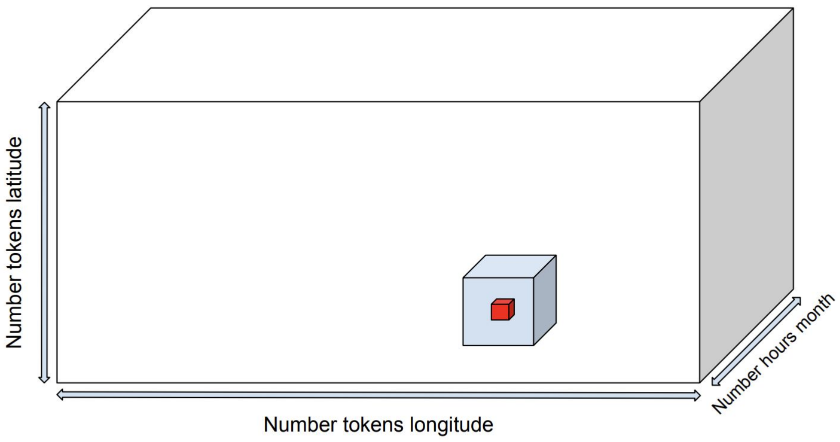
[1] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: (2019)

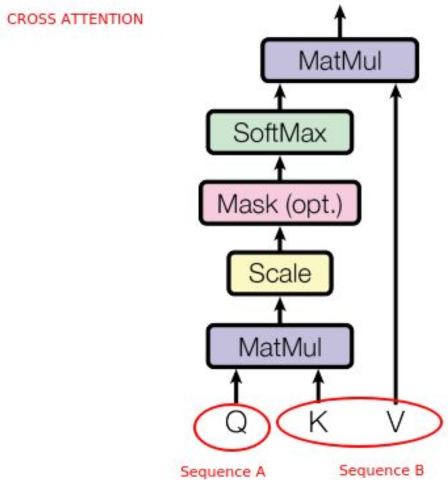
Extended transformer architecture

- Separate transformers for different fields coupled through attention
- Couple information between transformers via cross-attention

Self-supervised training

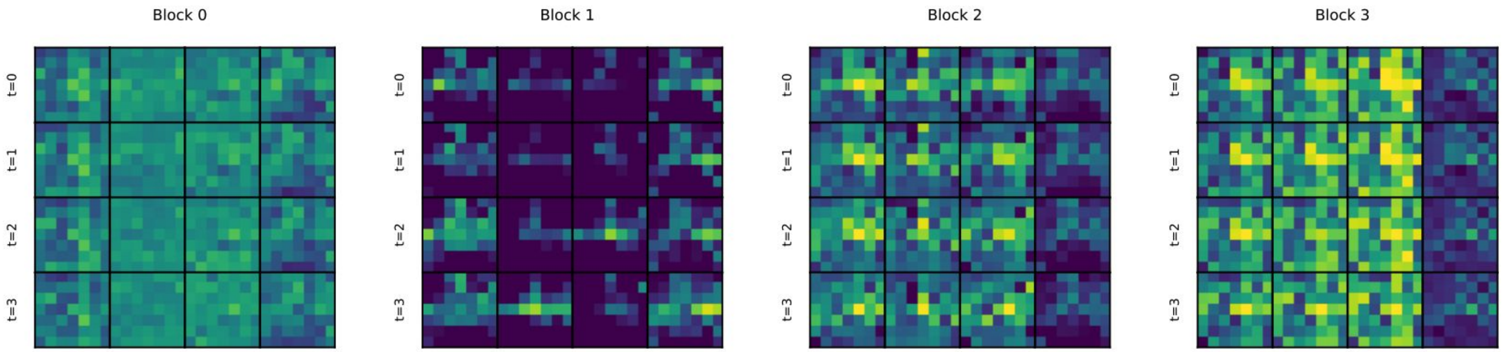
- Predict atmospheric state at time t from a small local neighborhood
- Forecasting, BERT [1]



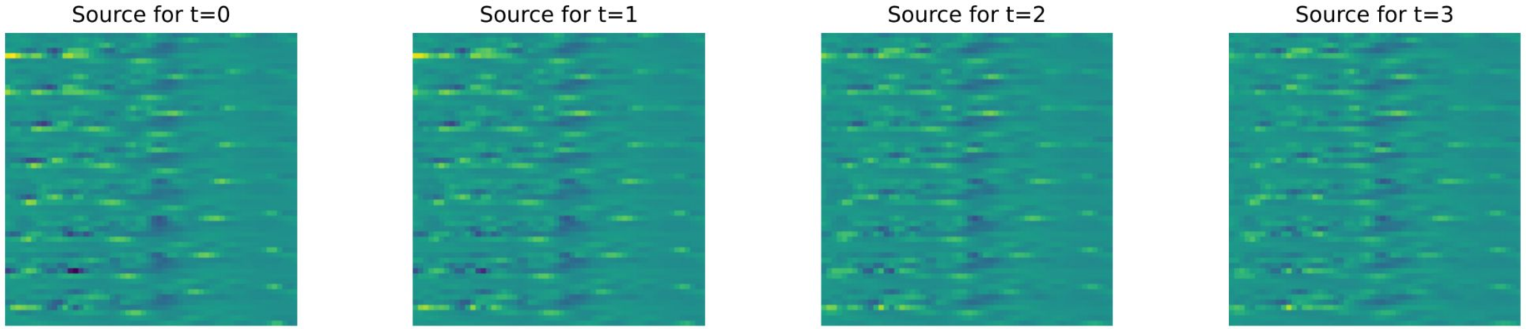


Contribution

- Implemented a data loader class that abstracts the file representation
- Benchmarked the performance of $CSCRATCH (Jülich Supercomputing Center)
- Implemented a plotting pipeline to assess the training of the model



Attention maps for each layer and head



Time evolution of the source tokens

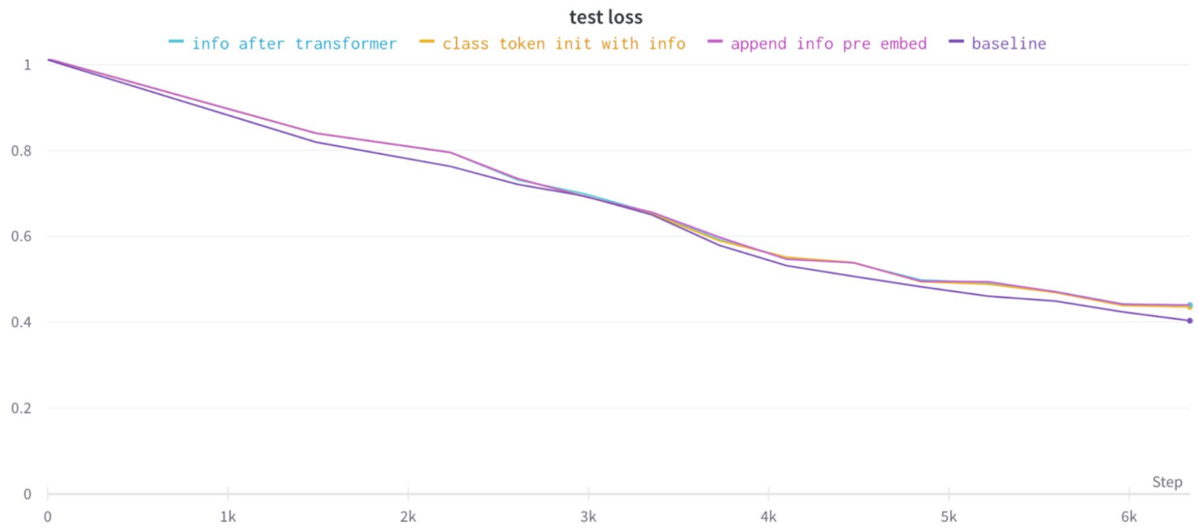
Contribution

Ablation studies to understand how the network makes use of the external information:

- Year, month, hour, latitude, longitude of the token

<u>ALTERNATIVES</u>

- Project the information tensor before appending
- Learnable embedding of the class token
- Re-append the information after the attention layer



l2 loss

Contribution

Ablation studies to understand how the network makes use of the external information:

- Year, month, hour, latitude, longitude of the token

ALTERNATIVES

Blurred data to simplify the problem

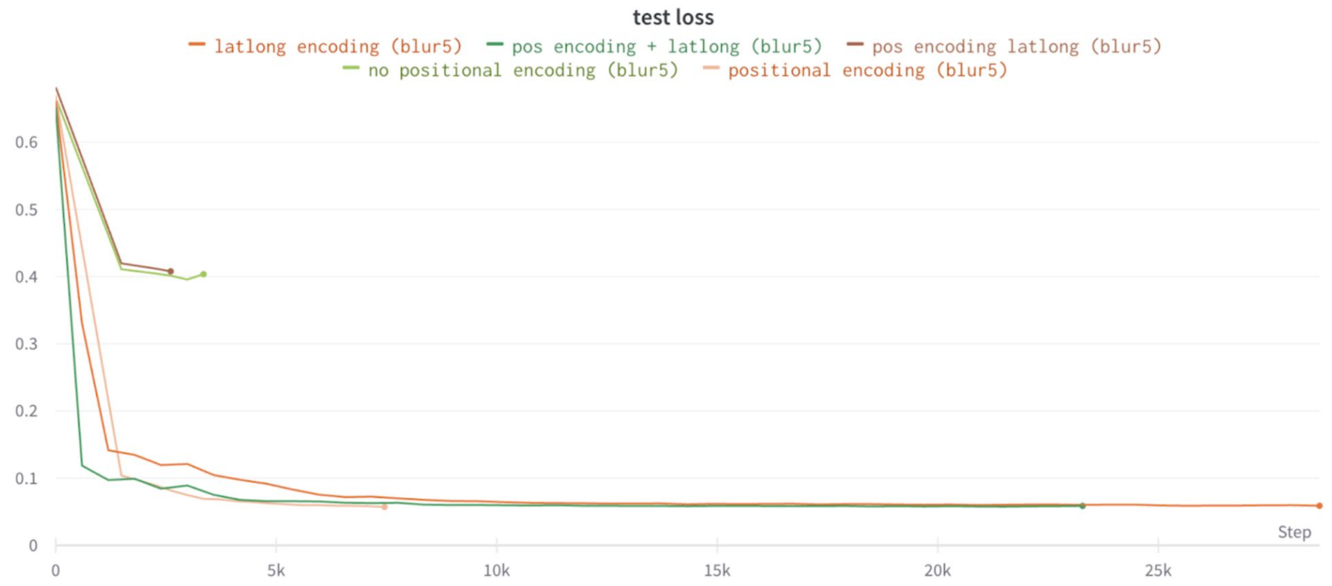
- New **positional encoding** (introducing global latitude and longitude)

```
pe[:,:,0::2] = torch.sin((target_t_idx * div)) + 0.5 * torch.sin((target_lat_idx * div))
pe[:,:,1::2] = torch.cos((target_t_idx * div)) + 0.5 * torch.sin((target_lon_idx * div))

x = x + pe[:, :x.size(1)]

return x
```

- Add the information to the standard positional encoding
- Standard positional encoding



CERN openlab

QUESTIONS?

polgarciarecasens@gmail.com