Lossy Compression of Climate Data using Convolutional Autoencoders

Silke Donayre Holtz, Uğur Çayoğlu, Pascal Friederich, Peter Sanders and Peter Braesicke

Institute of Theoretical Informatics (ITI), Steinbuch Centre for Computing (SCC), Institute of Meteorology and Climate Research (IMK)

Why do we want to compress climate data?

- **233TB** of data are generated per day [1].
- In research, data must be stored for at least **10 years**.
- Storage and bandwitdh must expand, **increasing costs**.

Solution: Store less data → **Compression**

[1] P. Dueben and P. Bauer. Geoscientific Model Development 11 (Oct 2018). pp. 3999-4009.

What is compression?

- Reduces a data size by removing redundant information leading to a compact representation.

  - **Lossless Compression**:
    No information is lost.

  - **Lossy Compression:**
    Higher compression factors depending on **maximal allowed error**.

$$\textbf{Compression factor} = \frac{Size\ Input\ Data}{Size\ compressed\ Data}$$

$$\textbf{Compressed data} = zfp.compress\_numpy(data, error)$$

- Comparison with **ZFP**[2] and **SZ**[3] (state-of-the-art).

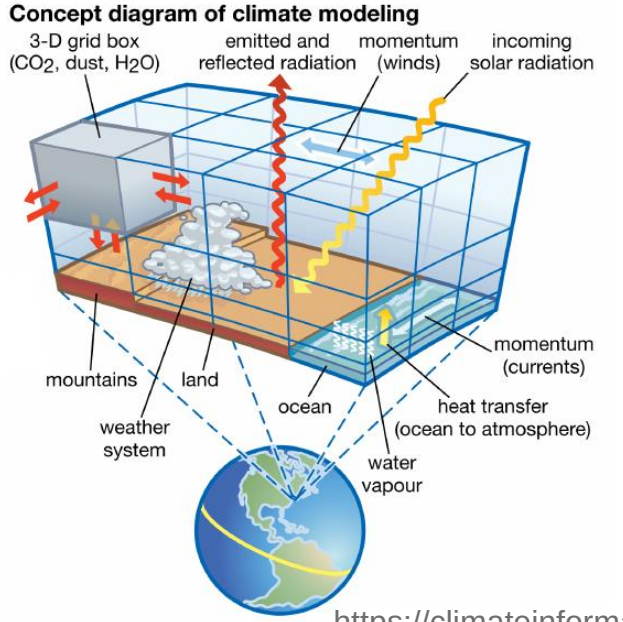
[2] S. Di and F. Capello. IEEE International Parallel and Distributed Processing Symposium (IPDPS) July 2016.
[3] Peter Lindstrom. IEEE Transactions on Visualization and Computer Graphics, 20(12):2674-2683, December 2014.

Climate Data

- Data from the European Centre for Medium-Range Weather Forecasts (ECMWF).
  - ERA5 Dataset: Hourly data on pressure levels from 1979 to present.

DATA DESCRIPTION	
Data type	Gridded
Horizontal coverage	Global
Horizontal resolution	Reanalysis: 0.25° x 0.25° Mean, spread and members: 0.5° x 0.5°
Vertical coverage	1000 hPa to 1 hPa
Vertical resolution	37 pressure levels
Temporal coverage	1979 to present
Temporal resolution	Hourly
File format	GRIB

https://cds.climate.copernicus.eu/

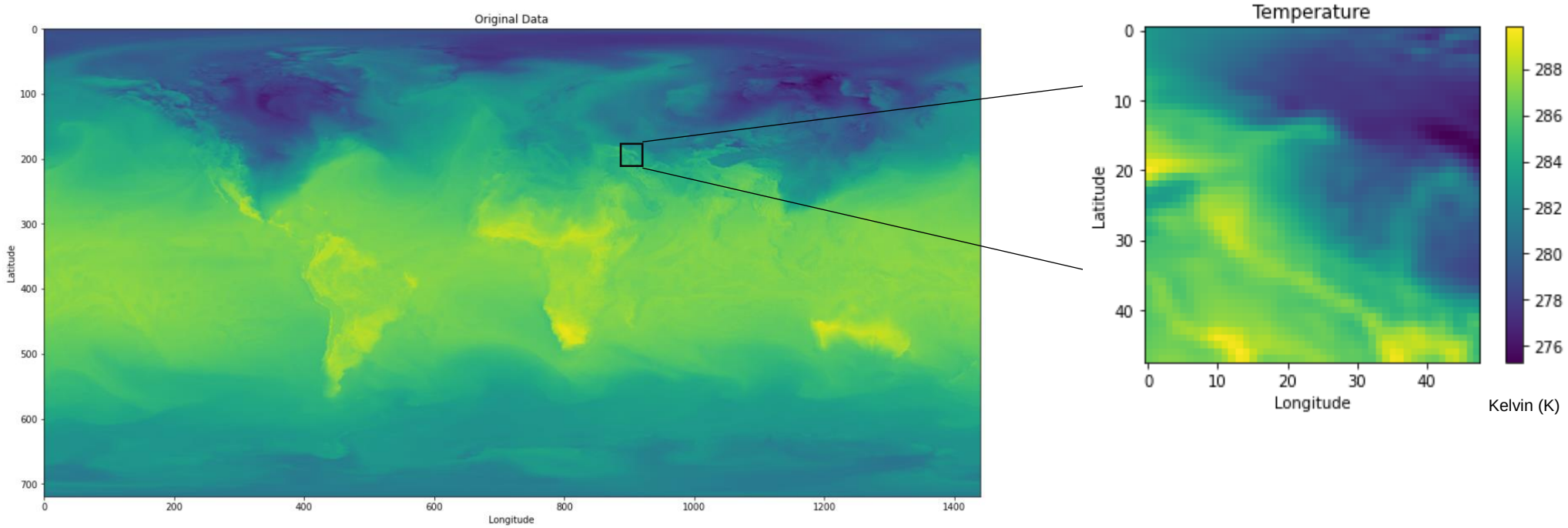


https://climateinformation.org/

Data Preprocessing

- Dimensions used: Time, latitude, longitude set pressure level to 1000hPa.
- Split the data into [16, 48, 48, 1] chunks.

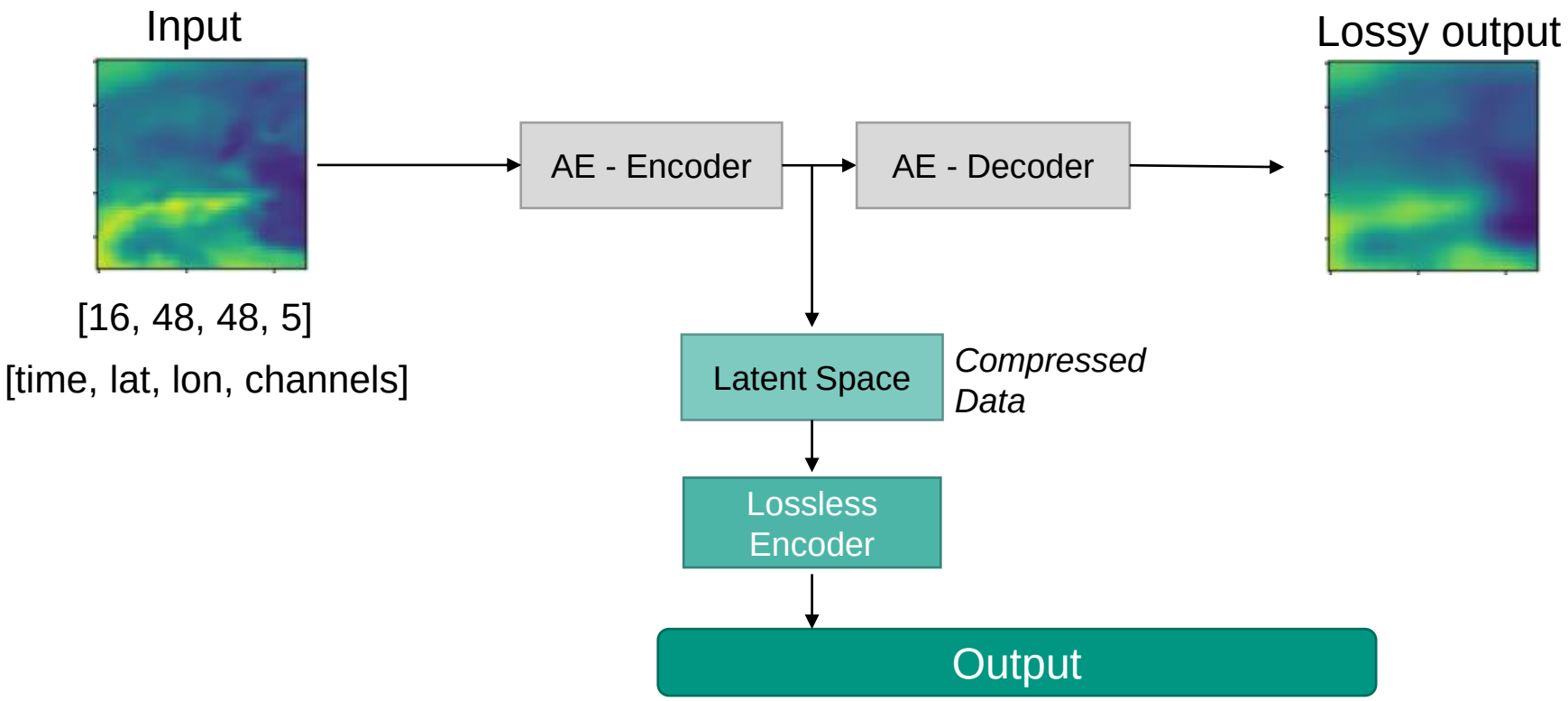
[Time, latitude, longitude, temperature]



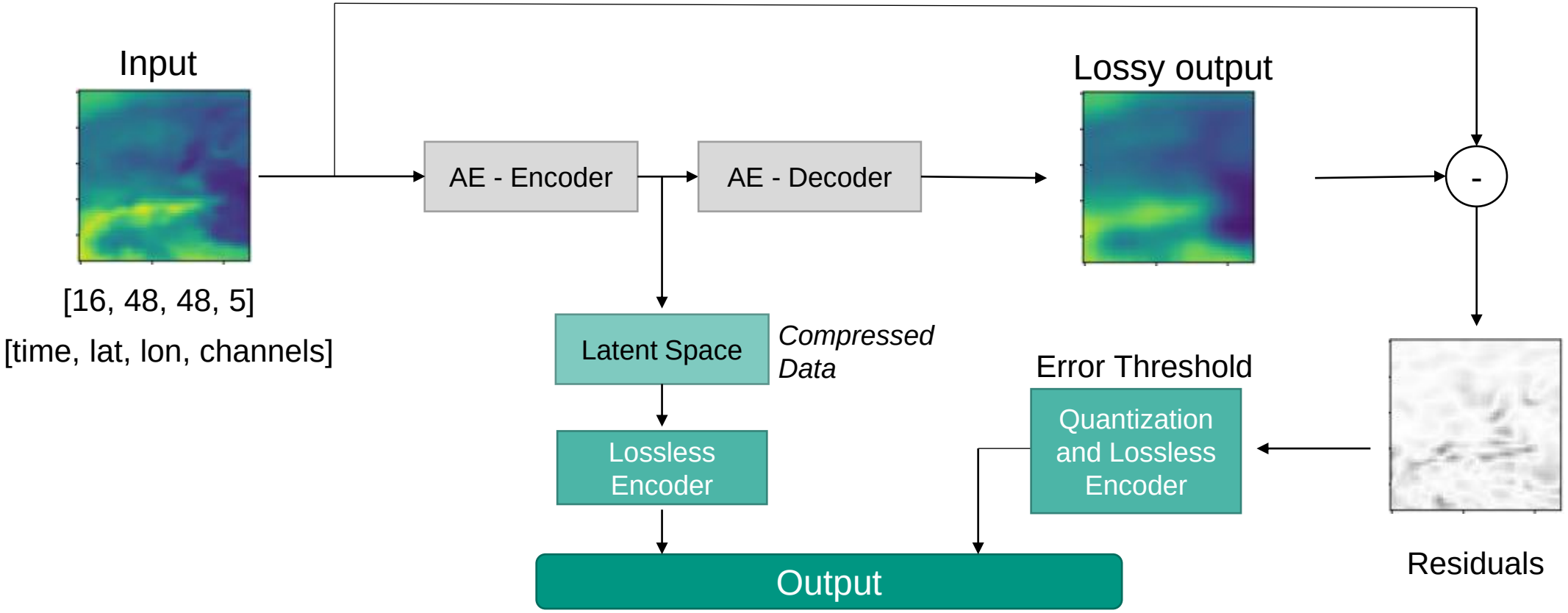
Data Preprocessing

- Added encoded longitude and latitude as extra information to the model.
- Input of size [16, 48, 48, **5**]
  - Where 5 channels = temperature + encoded lat1 + encoded lat2 + encoded lon1 + encoded lon2
- Standardized the data
- Used **temperature** data from 1979 and 1980
- Randomly sampled chunks

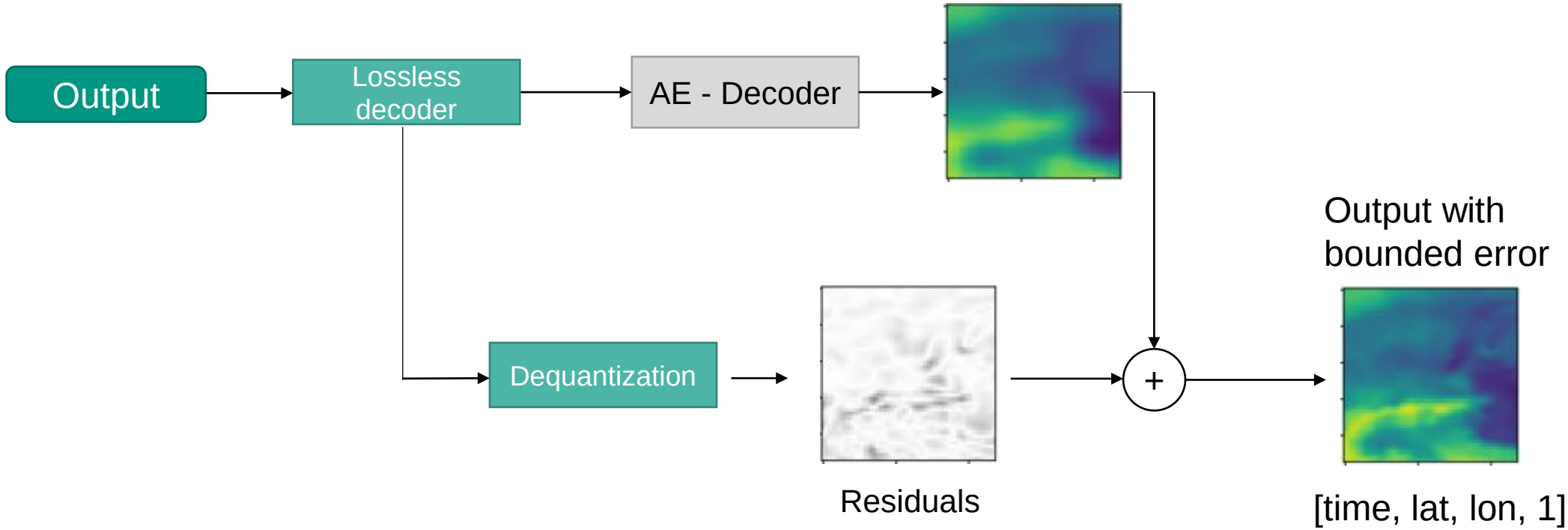
Encoder Architecture



Encoder Architecture with bounded error

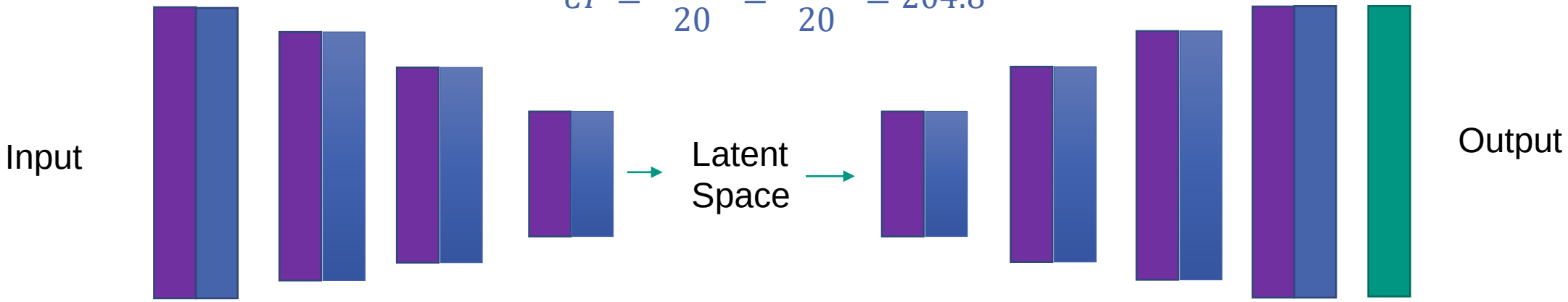


Decoder Architecture



Convolutional Autoencoder

$$CF = \frac{(2^3)^4}{20} = \frac{4096}{20} = 204.8$$



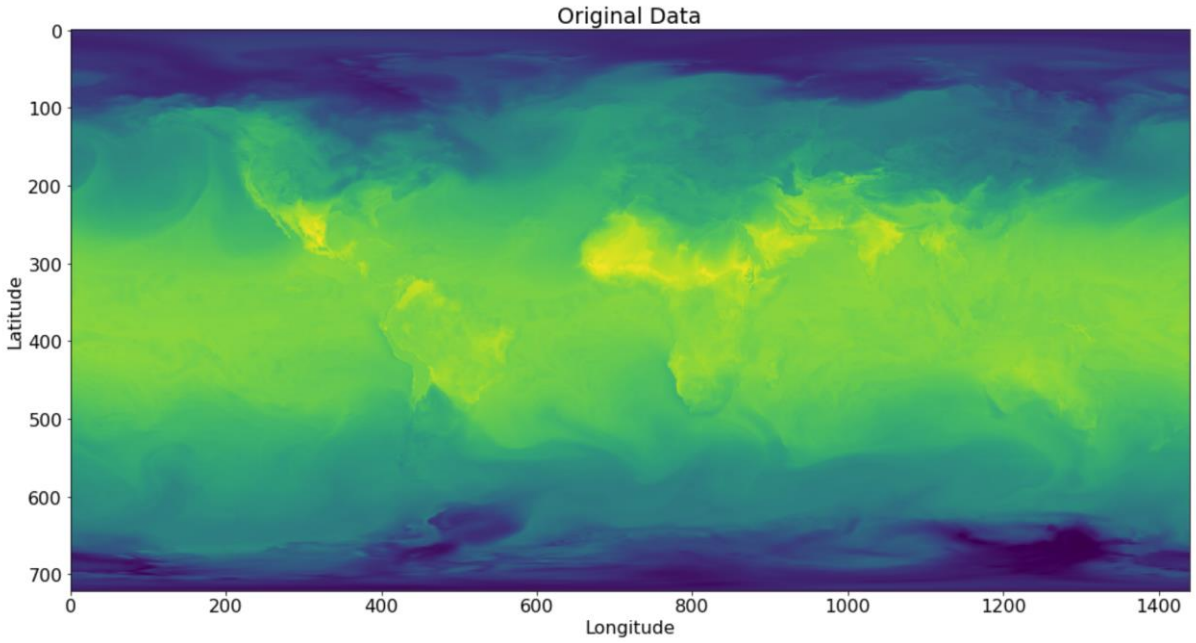
Conv3D
ReLU
De/Conv3D – Stride 2
Filters [10,20,20,20]

Loss function:
Mean Squared Error

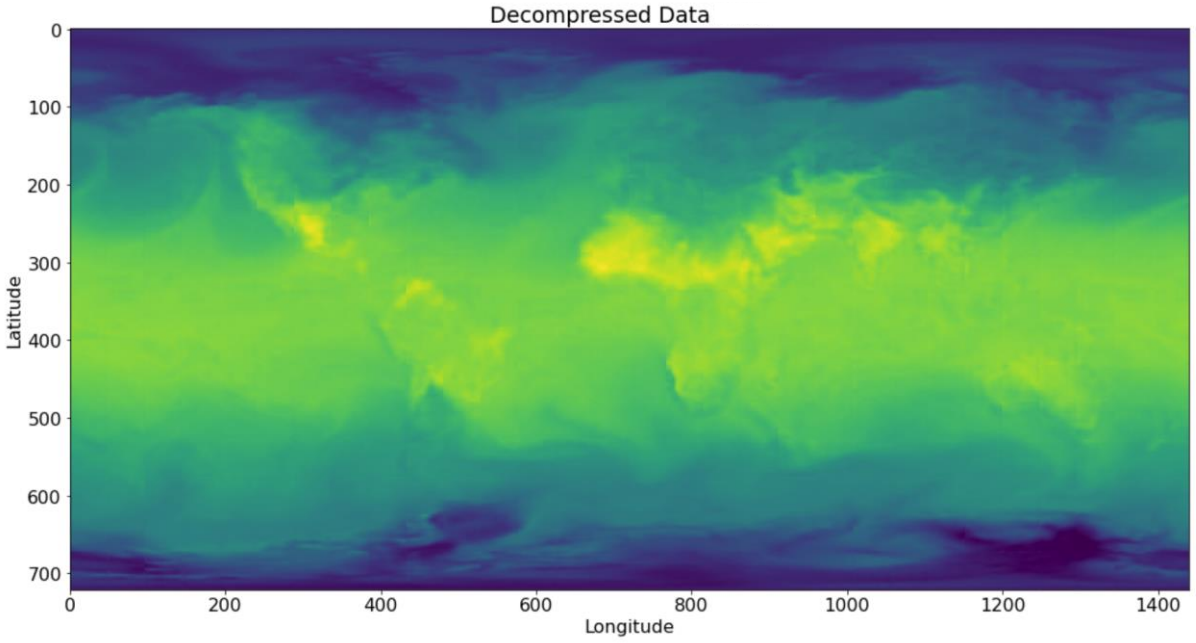
Training Data: 400k chunks (1979)
Validation Data: 40k chunks (1979)
Testing Data: 100k chunks (1980)

Autoencoder Output

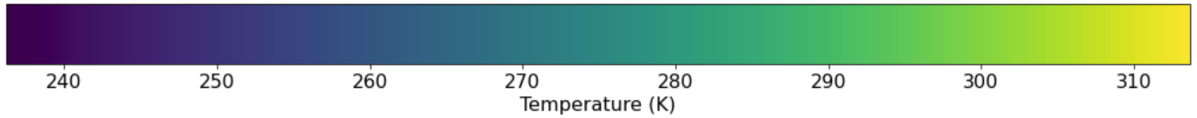
Input Data



Autoencoder Output



1980-04-12

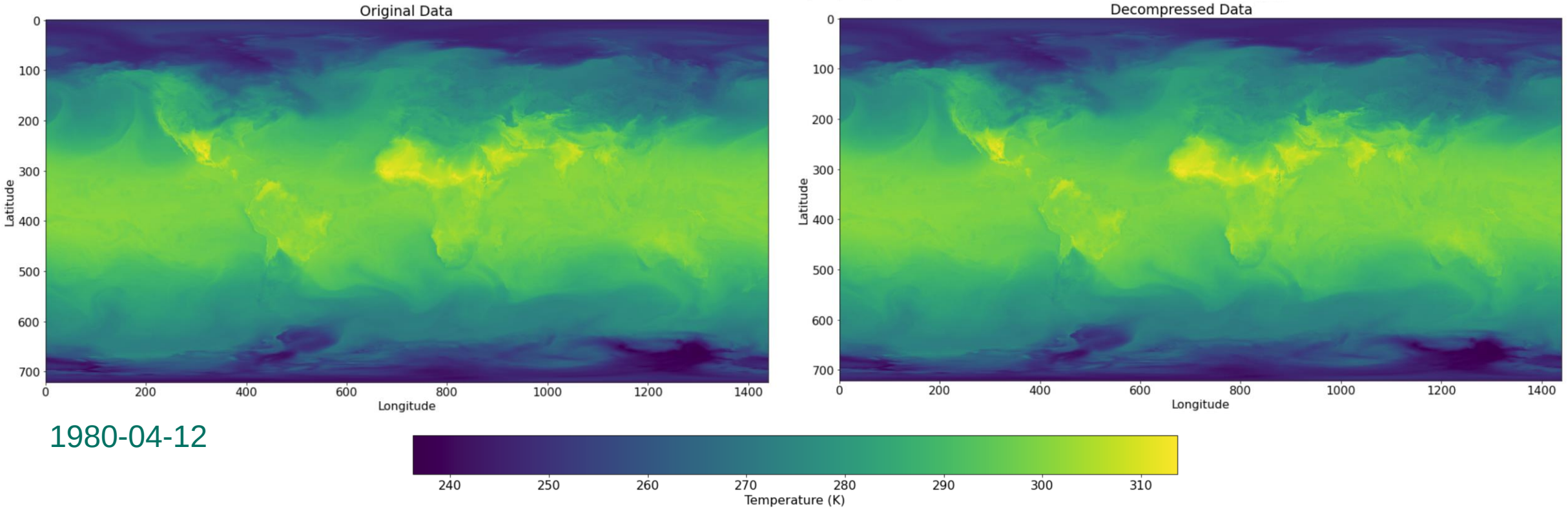


Model Output with bounded error

$Threshold = 0.5\ K$ $Compression\ Factor = 29$

Input Data

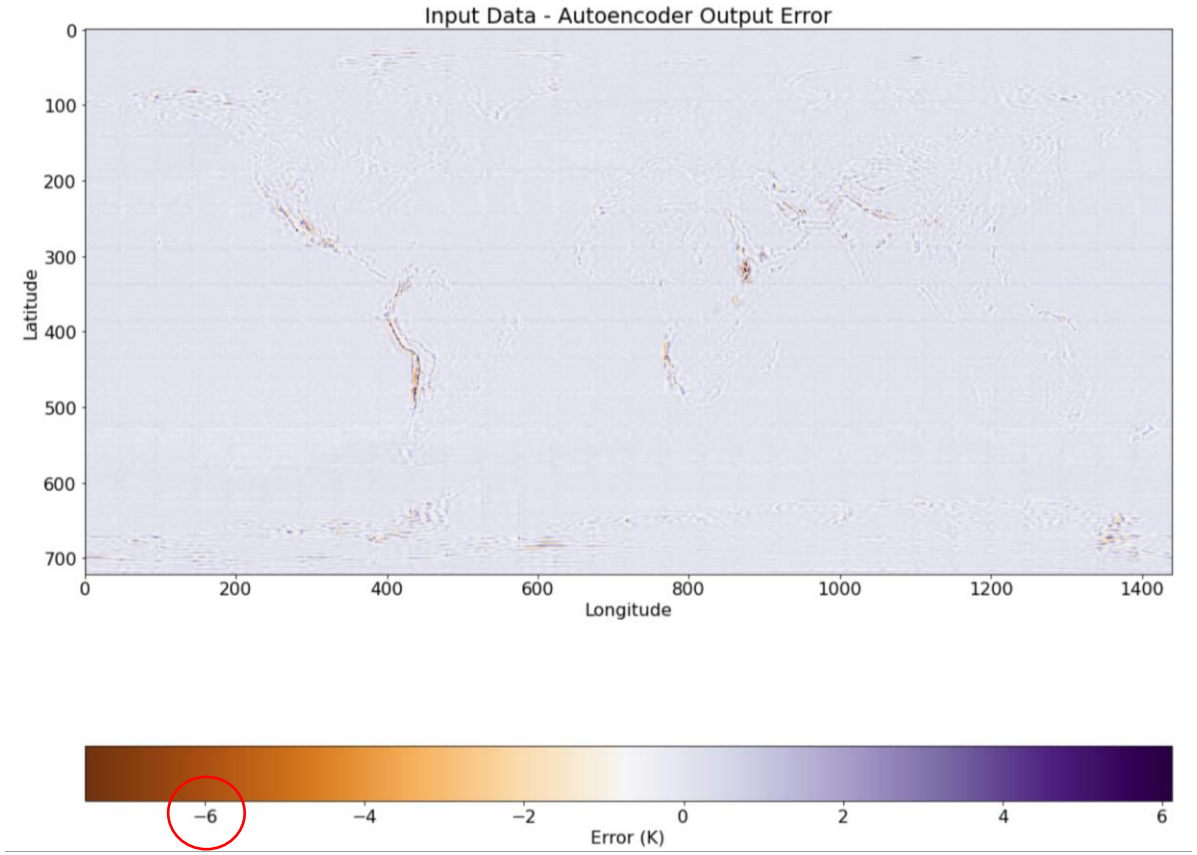
Model (AE + Residuals) Output

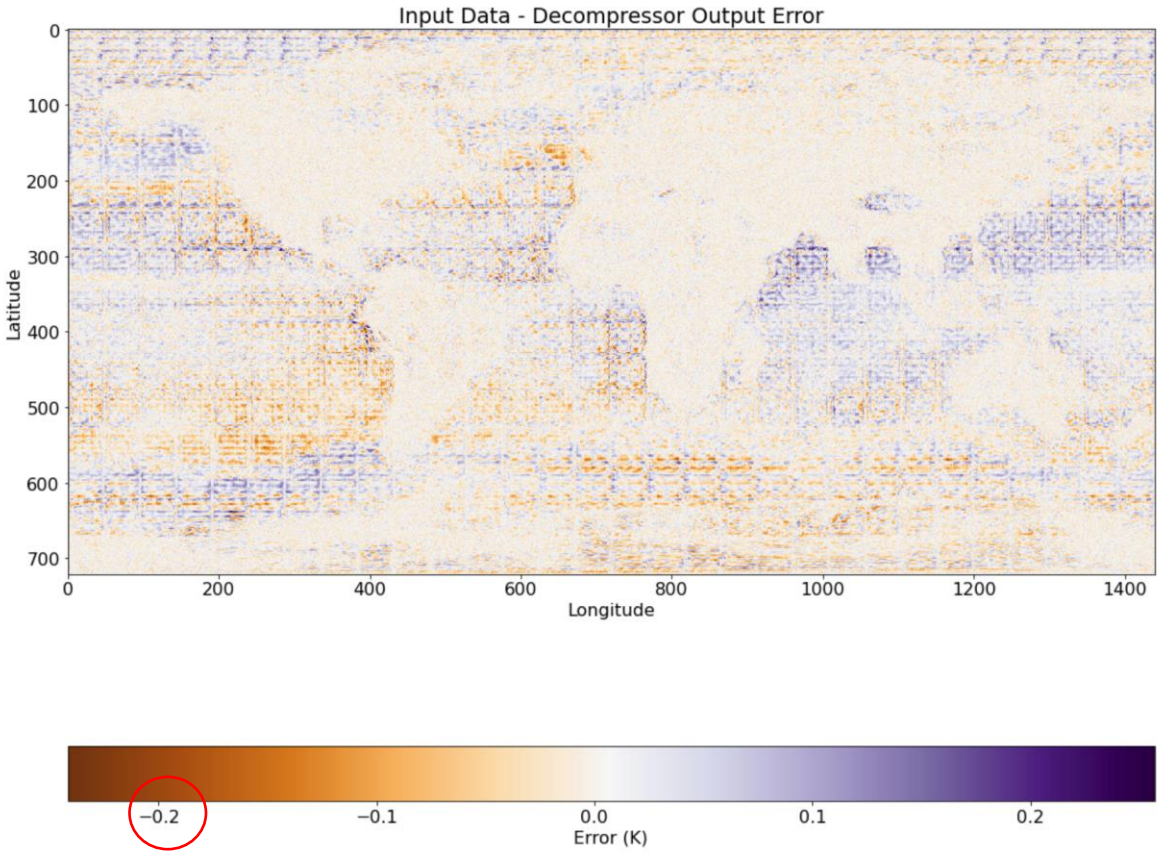


1980-04-12

Autoencoder and Model Error

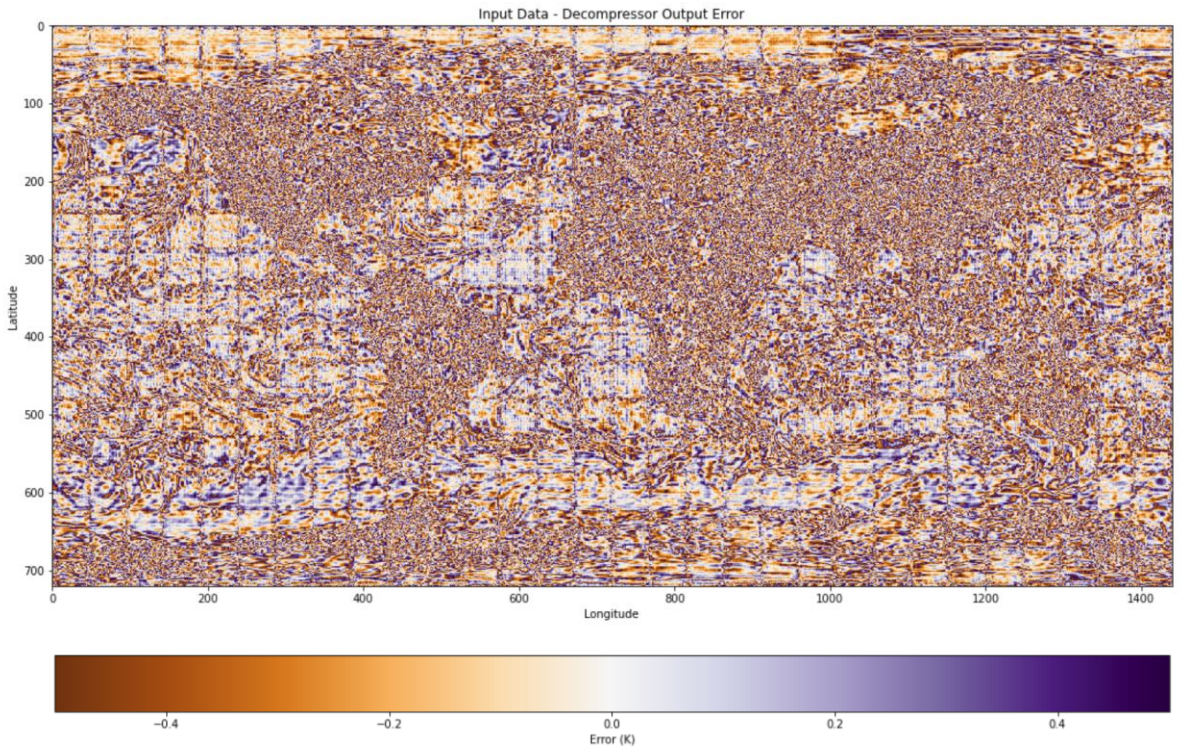
$Threshold = 0.5\ K$ $Compression\ Factor = 29$



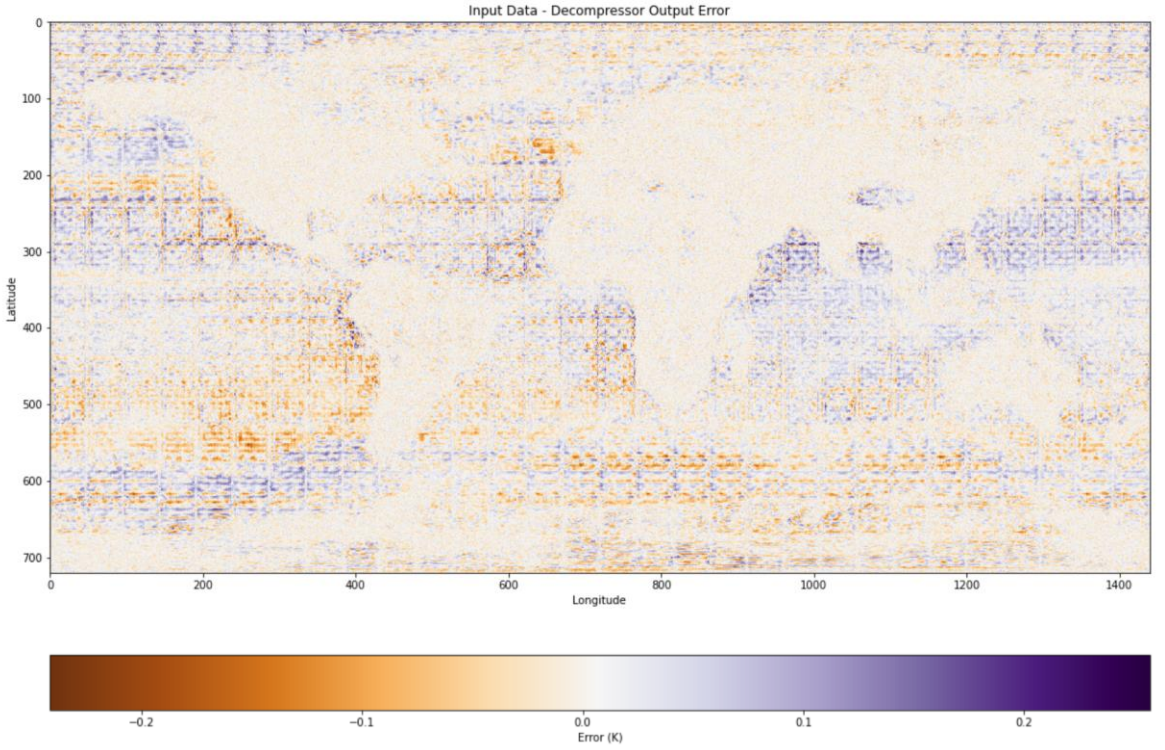


Residuals

$Threshold = 0.5\ K$



1 time step



Mean through 16 time steps

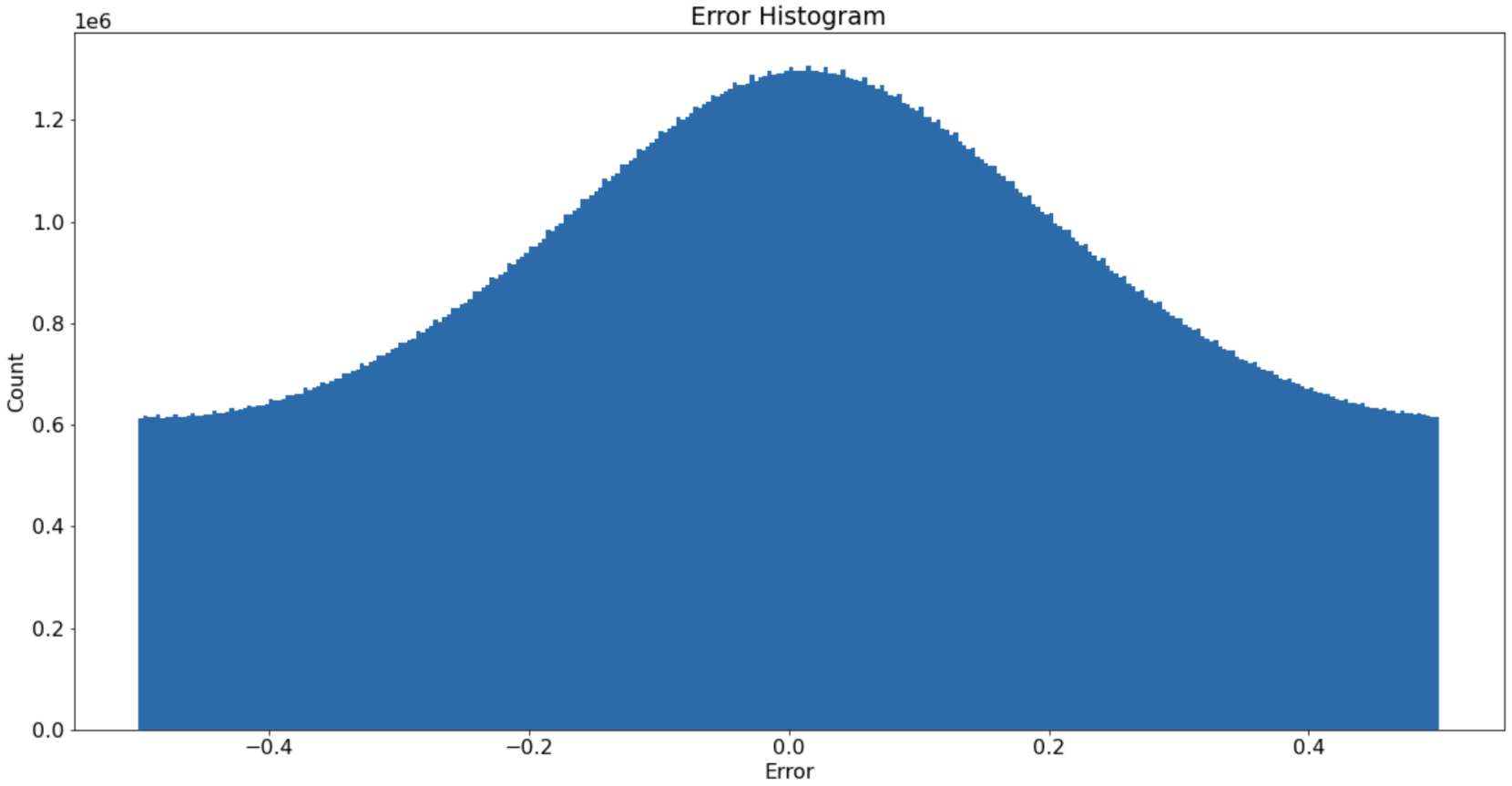
Residuals

$Threshold = 0.5\ K$

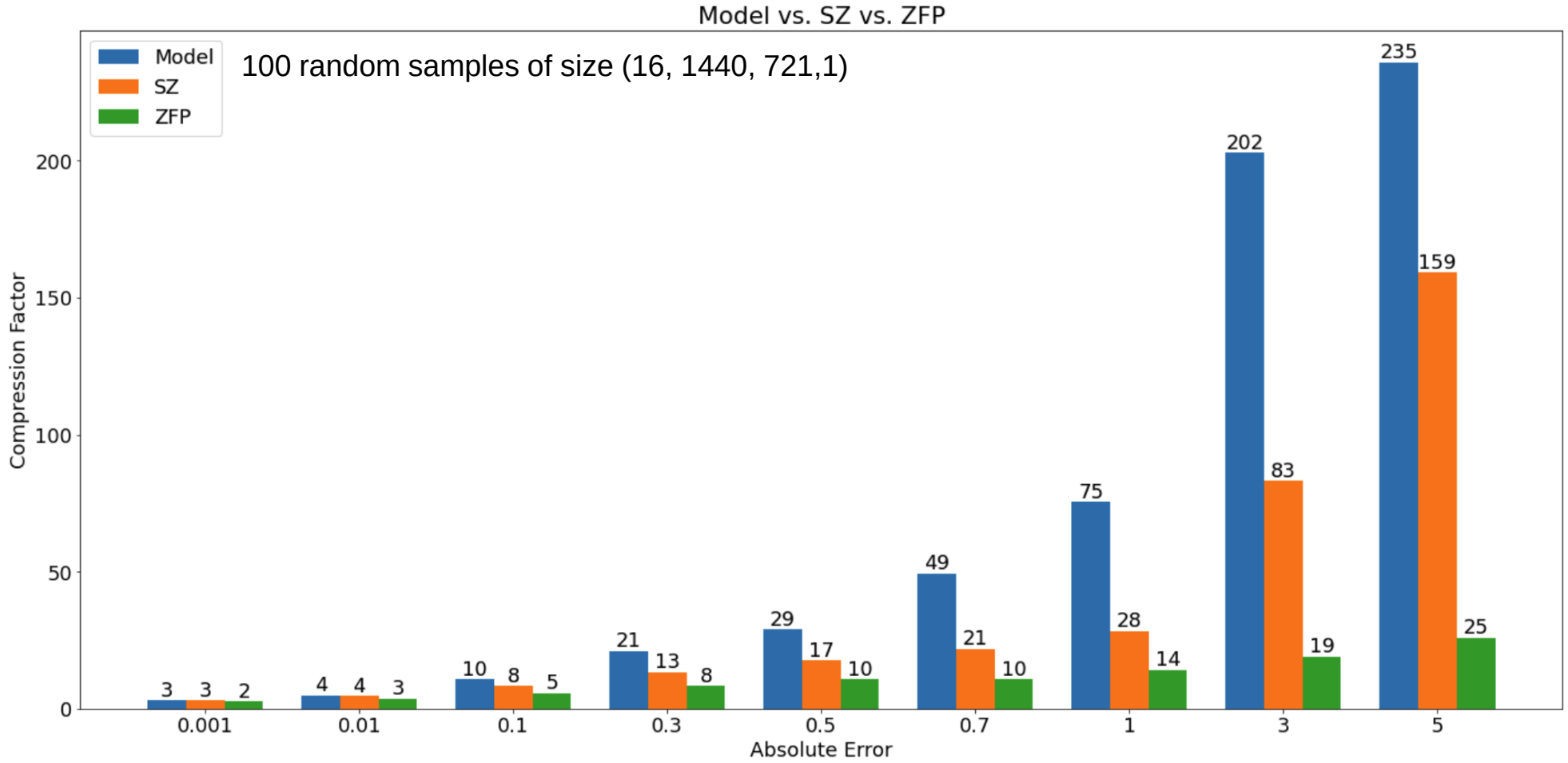
Data Original
=======
Mean: 280.13
Std: 17.70
Max Val: 317.71
Min Val: 229.97

Decompressed
=======
Mean: 280.12
Std: 17.70
Max Val: 318.09
Min Val: 229.56

Error
=======
Mean: 0.00511
Std: 0.25568
Max error: 0.5000
Min error: -0.5000



Our model vs. SZ vs. ZFP



Conclusions

- Improved compression factor with new model
- Time complexity should be taken into account
- Better results could be achieved by using all dimensions
- Try different lossless encoders
- Work with other attributes

Contact:
silke.holtz@student.kit.edu

Thank you!

Contact:
silke.holtz@student.kit.edu