The Alan Turing Institute

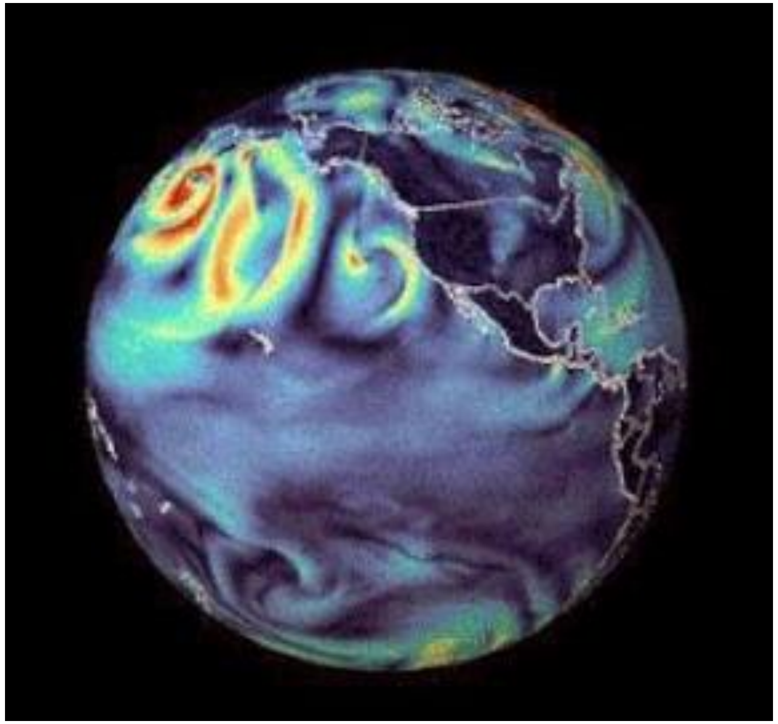
Explainable forecasting from big weather data: rapid and sustainable solutions

FOSDEM’25 – HPC, Big Data & Data Science Devroom

David Salvador-Jasin
Research Data Scientist

Data-driven weather prediction

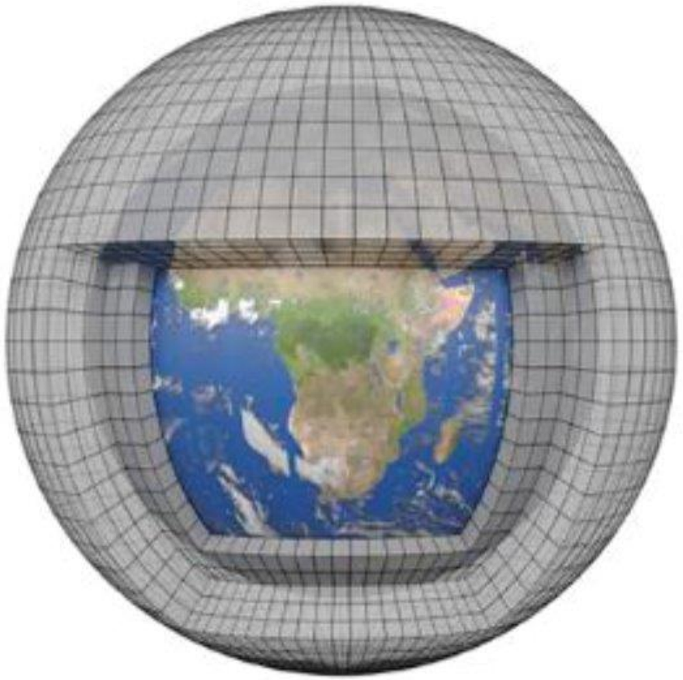
- Traditional physics-based models provide accurate forecasts, but are computationally expensive
- Substantial progress in data-driven weather prediction in recent years
- Recently developed purely data-driven models outperform physics-based models in many standard forecast scores



Data-driven weather prediction

Diverse set of deep-learning architectures being employed:

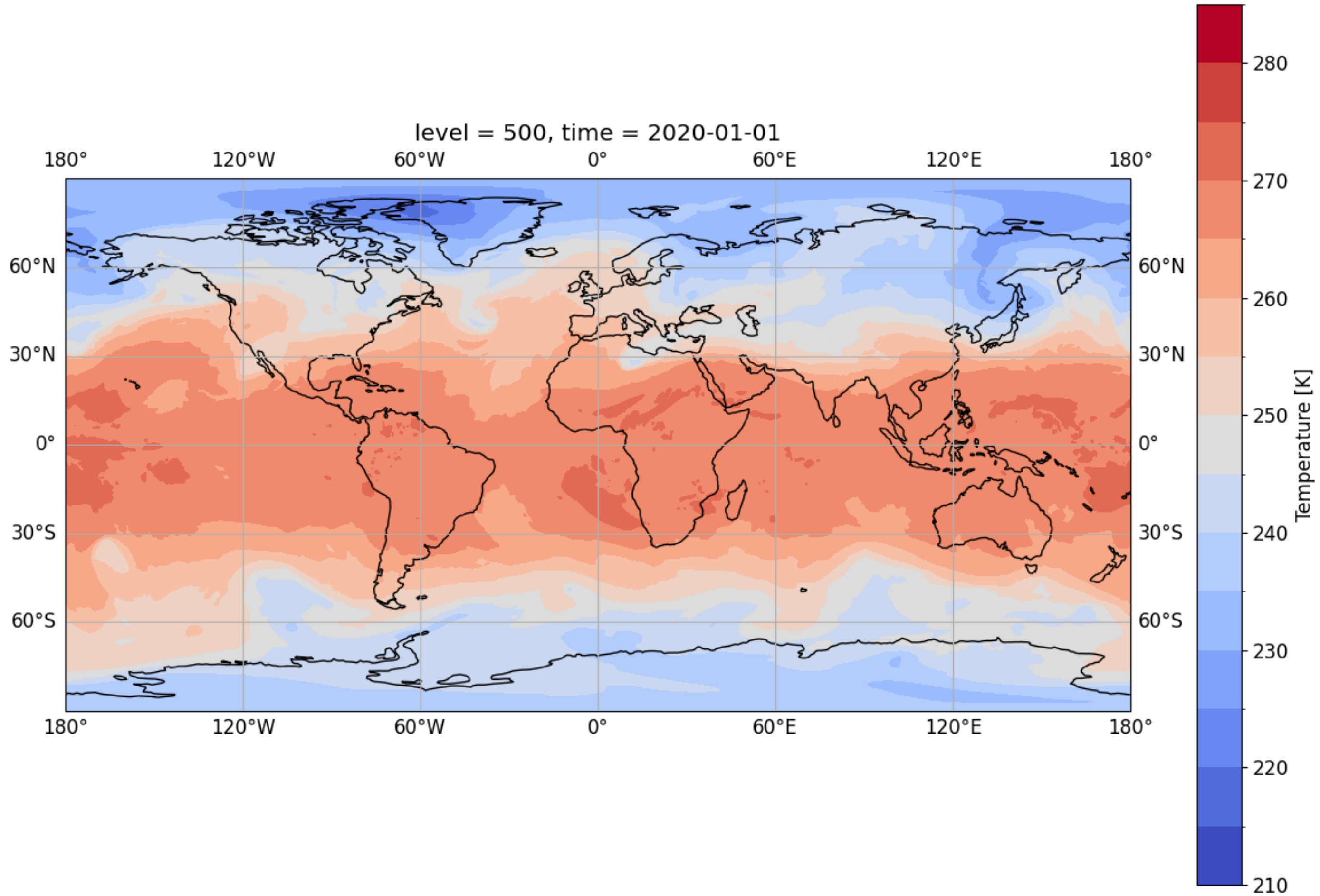
- Vision transformers
- Neural operators
- Graph Neural Networks (GNNs)
- Et al.



The ERA5 dataset

- ECMWF atmospheric reanalysis of the global climate covering the period from mid 20th century to present
- Hourly estimates of many variables on a lat/lon grid at multiple pressure levels
- Huge dataset:
    - 0.25 degree lat/lon res
    - Temperature
    - Level=500 hPa
    - 2000/01/01 – 2020/01/01

$\left.\right\}$ 728 GB



Our goal

- Develop an **inexpensive**, **data-driven forecasting model** that can serve as a **baseline for comparison**, having a similar role to:
  - **Persistence forecasting**
  - **Climatology**
- Gain deeper understanding of the **underlying physics** from the data

Dynamic Mode Decomposition (DMD):
- Purely data-driven
- Computationally efficient
- Explainable
- Can approximate non-linear dynamics through a linear approximation

The team

The Alan Turing Institute

- David Salvador Jasin
- Lydia France
- Louisa Van Zeeland
- Oliver Strickson
- Peter Yatsyshin

University of Washington

- Nathan Kutz

Barcelona Supercomputing Centre

- Benet Eiximeno

Dynamic Mode Decomposition

- Seeks the leading spectral decomposition (**eigenvalues and eigenvectors**) of the **best-fit linear operator** $\mathbf{A}$ that relates two snapshot matrices in time

- Provides a **best-fit, linear characterization of a non-linear dynamical system** from data alone

- Connection with **Koopman theory** for dynamical systems

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ x(t_1) & x(t_2) & \dots & x(t_m) \\ | & | & & | \end{bmatrix}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ x(t_2) & x(t_3) & \dots & x(t_{m+1}) \\ | & | & & | \end{bmatrix}$$

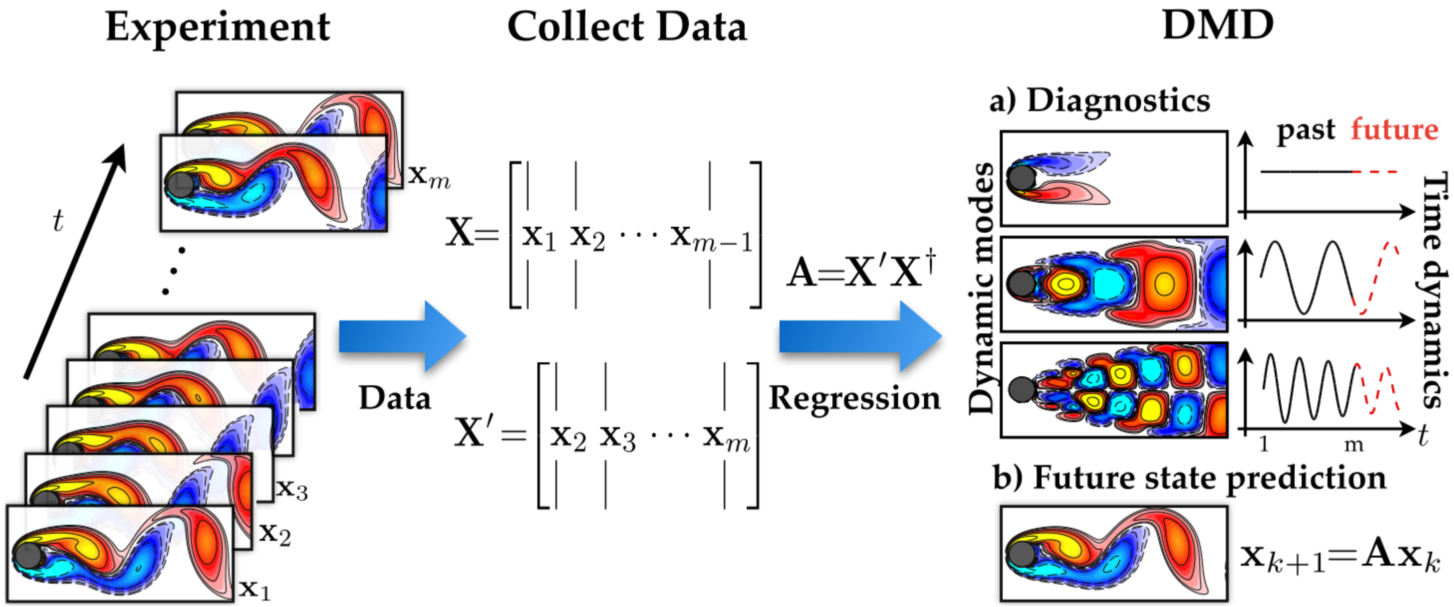
$$\mathbf{X}' \approx \mathbf{A}\mathbf{X}$$

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F = \mathbf{X}'\mathbf{X}^\dagger$$

$$\mathbf{X} \approx \underbrace{\mathbf{\Phi}}_{\text{eigenvectors}} \operatorname{diag}(\underbrace{\mathbf{b}}_{\text{amplitudes}}) \mathbf{T}(\underbrace{\boldsymbol{\omega}}_{\text{eigenvalues}})$$

Dynamic Mode Decomposition

Connects the favorable aspects of the **SVD for spatial dimensionality reduction** and the **FFT for temporal frequency identification**



Reproduced from Kutz et al. (2016)

Optimized DMD (optDMD)

- Original DMD strongly **affected by the presence of noise**
- optDMD (*Askham & Kutz, 2018*) is a **non-linear optimization** of DMD **enabled by variable projection** methods
- **Avoids much of the bias of exact DMD**
- Can be viewed as **a postprocessing step of the original DMD algorithm**

$$\mathbf{X} \approx \mathbf{\Phi}\,\mathrm{diag}(\mathbf{b})\mathbf{T}(\boldsymbol{\omega}) =$$

$$\begin{bmatrix} | & \cdots & | \\ \phi_1 & \cdots & \phi_r \\ | & \cdots & | \end{bmatrix} \begin{bmatrix} b_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_r \end{bmatrix} \begin{bmatrix} e^{\omega_1 t_1} & \cdots & e^{\omega_1 t_m} \\ \vdots & \ddots & \vdots \\ e^{\omega_r t_1} & \cdots & e^{\omega_r t_m} \end{bmatrix}$$

optDMD solves:

$$\underset{\boldsymbol{\omega},\mathbf{\Phi}_\mathrm{b}}{\mathrm{argmin}} \|\mathbf{X} - \mathbf{\Phi}_\mathrm{b}\mathbf{T}(\boldsymbol{\omega})\|_F\,,$$

where $\mathbf{\Phi}_\mathrm{b} = \mathbf{\Phi}\,\mathrm{diag}(\mathbf{b})$

Approx optDMD:

$$\mathrm{SVD} \rightarrow \mathbf{X}_r = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^*$$

$$\underset{\boldsymbol{\omega},\mathbf{\Phi}_\mathrm{b}}{\mathrm{argmin}} \|\mathbf{\Sigma}_r\mathbf{V}_r - \mathbf{\Phi}_\mathrm{b}\mathbf{T}(\boldsymbol{\omega})\|_F$$

The PyDMD package

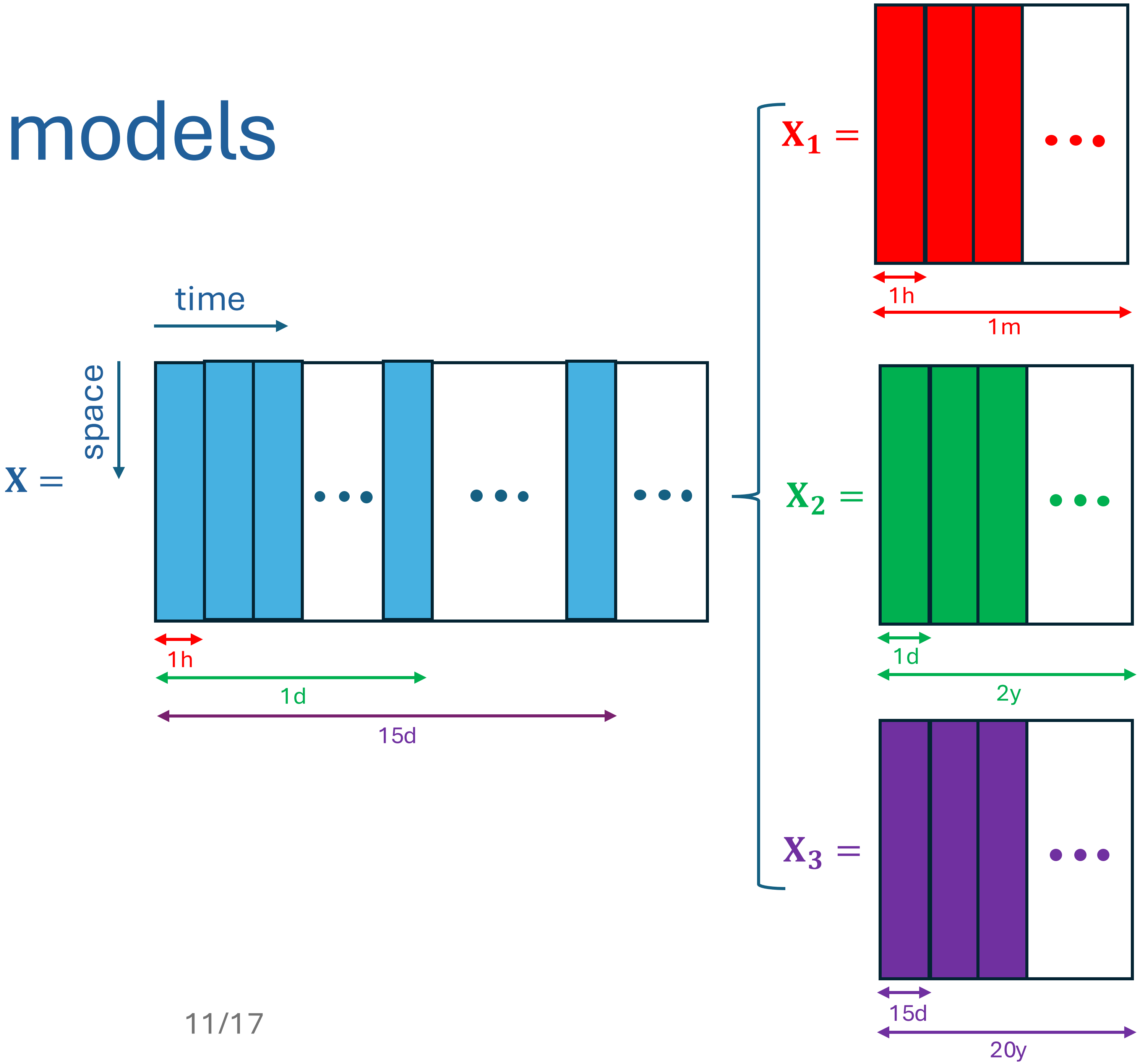
- A Python package for performing DMD: https://github.com/PyDMD/PyDMD
- The optDMD algorithm of Askham & Kutz (2018) is implemented in the BOPDMD class of PyDMD
- We have implemented a new `fit_econ` method for a much cheaper DMD fit: https://github.com/PyDMD/PyDMD/pull/568

```
from pydmd import BOPDMD

bopdmd = BOPDMD(svd_rank=12, proj_basis=U)
bopdmd.fit(X, t)  # intractable if X is very large!
bopdmd.fit_econ(s, V, t)  # can run on a laptop in seconds
```

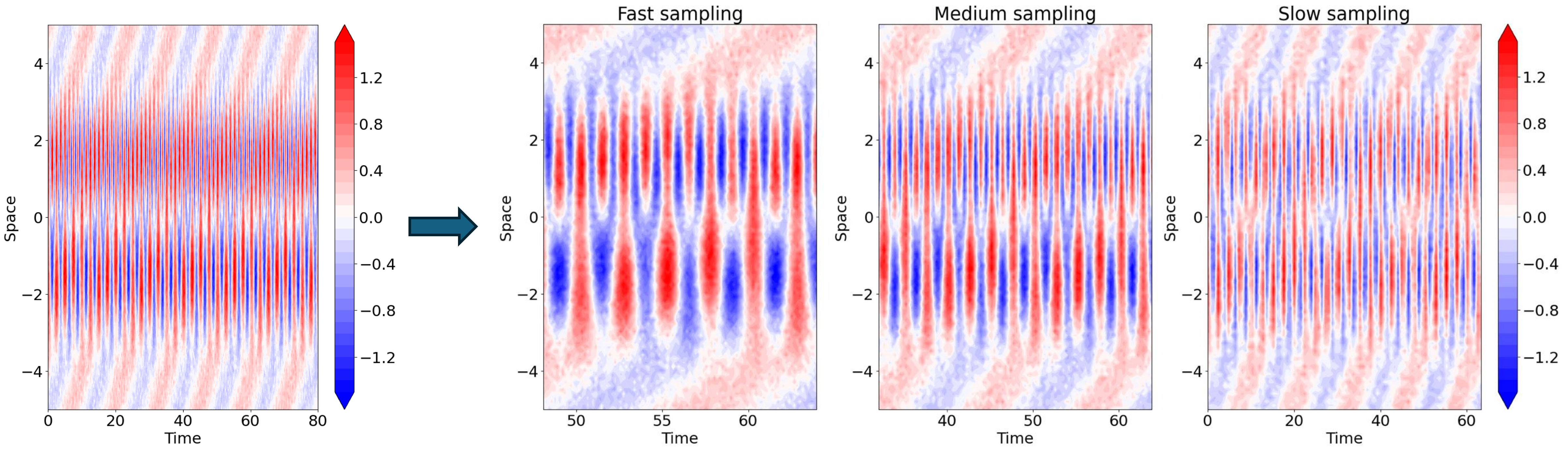
Combining DMD models

- Although we can apply optDMD to the rank r approximation of $\mathbf{X}$, we still need to compute the SVD of $\mathbf{X}$
- Can we build separate DMD models for much-smaller subsamples of $\mathbf{X}$ and combine them together to produce a forecast?



Combining DMD models

$x(t)$ $x_1(t)$ $x_2(t)$ $x_3(t)$



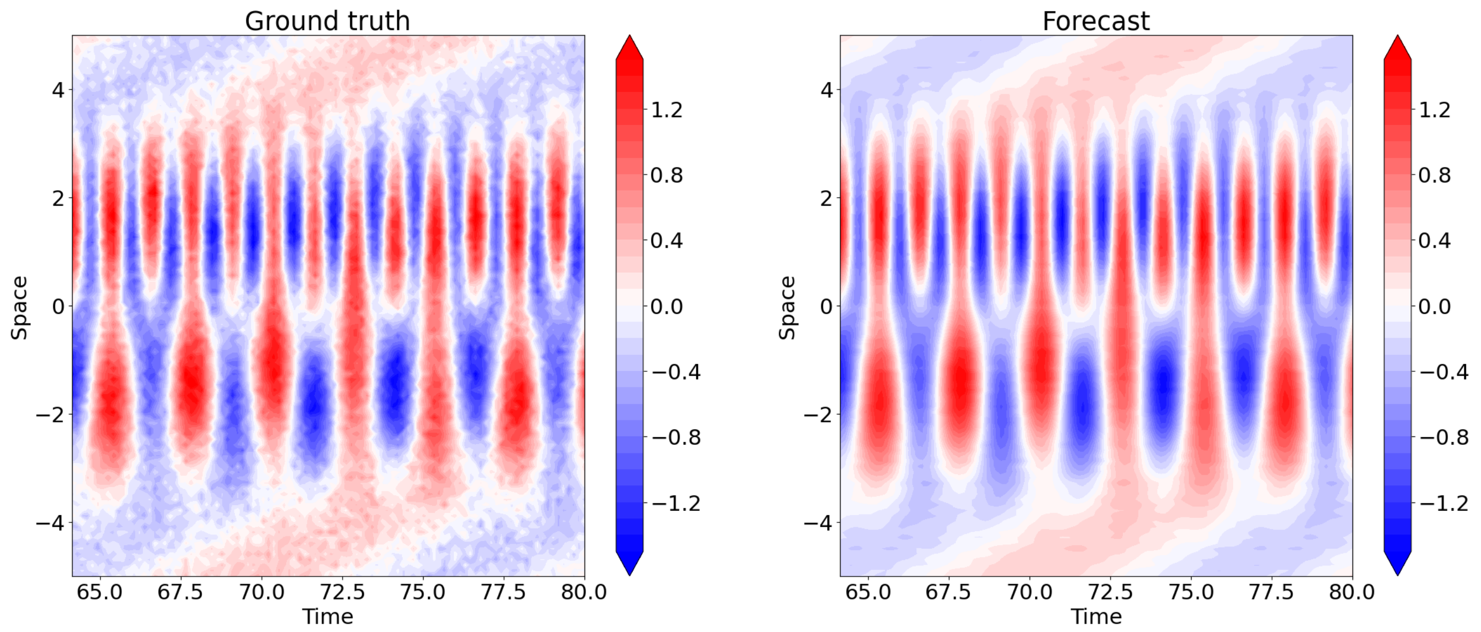
Duration: 16 s	Duration: 32 s	Duration: 63 s
Sampling period: 0.16 s	Sampling period: 0.32 s	Sampling period: 0.8 s

Combining DMD models

$$x_1(t) \approx DMD_1 = \sum_{j=1}^{r_1} \boldsymbol{\phi}_{(1,j)} e^{t\omega_{(1,j)}} b_{(1,j)}$$

$$x_2(t) \approx DMD_2 = \sum_{j=1}^{r_2} \boldsymbol{\phi}_{(2,j)} e^{t\omega_{(2,j)}} b_{(2,j)}$$

$$x_3(t) \approx DMD_3 = \sum_{j=1}^{r_3} \boldsymbol{\phi}_{(3,j)} e^{t\omega_{(3,j)}} b_{(3,j)}$$

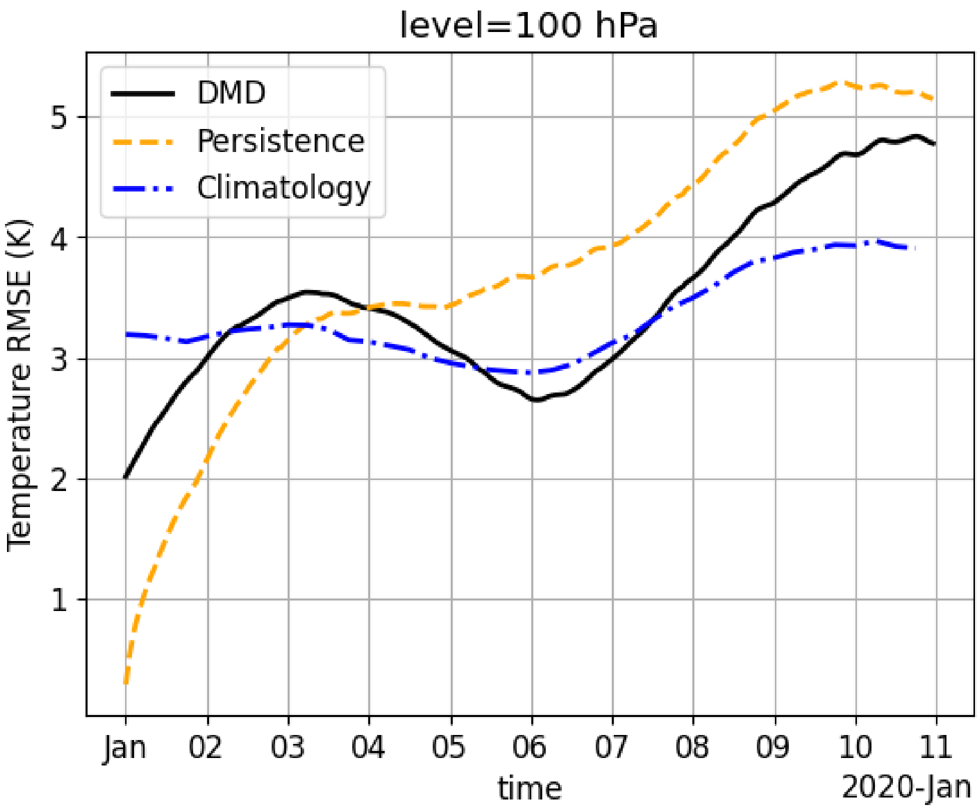


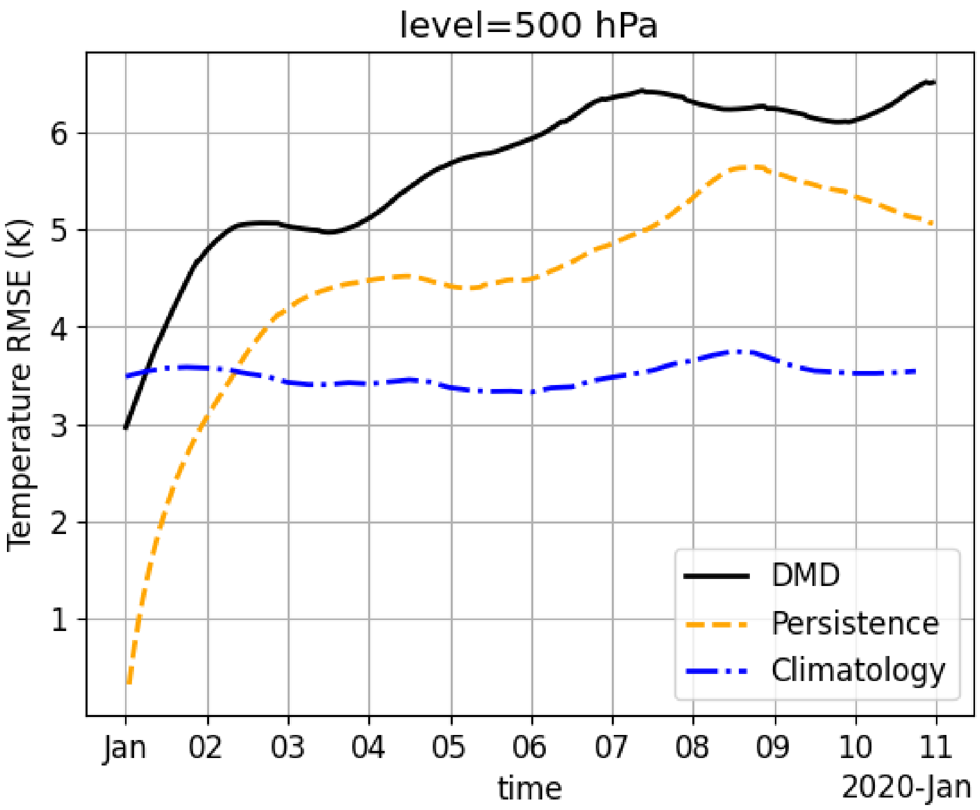
A model for $x(t)$ can be obtained by cherry-picking modes from DMD_1, DMD_2 and DMD_3.

Check it out on GitHub: https://github.com/ClimeTrend/dmd-toy-dataset

Preliminary results

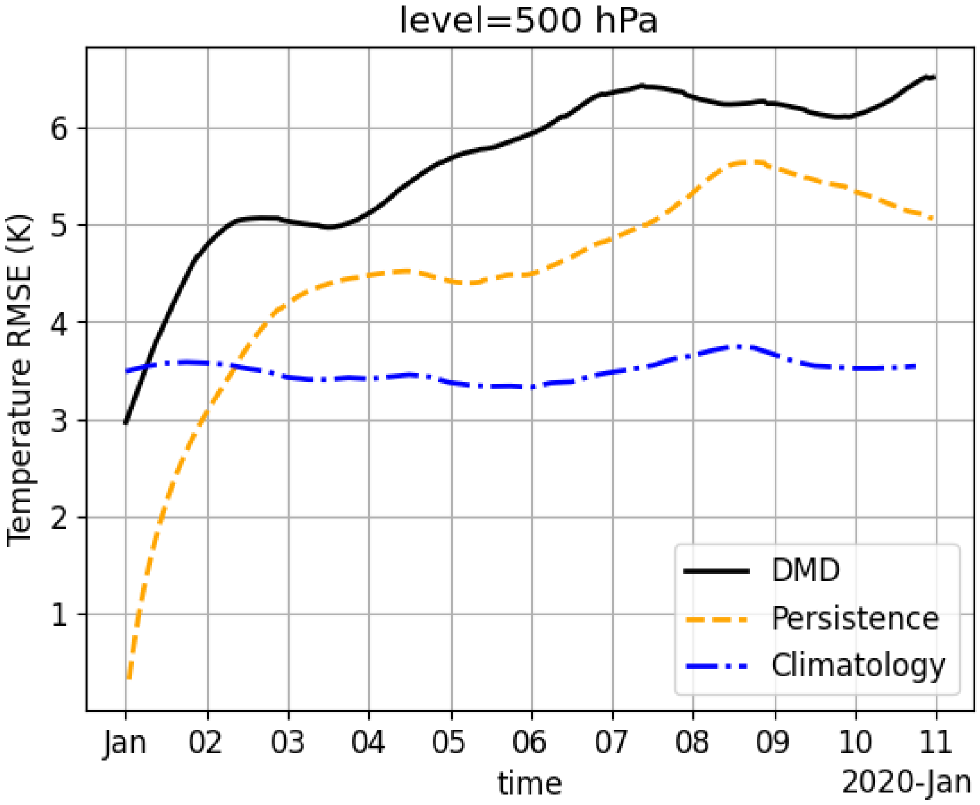
DMD model trained on Dec 2019 only, $\Delta t = 1h$
Forecasting first 10 days of Jan 2020





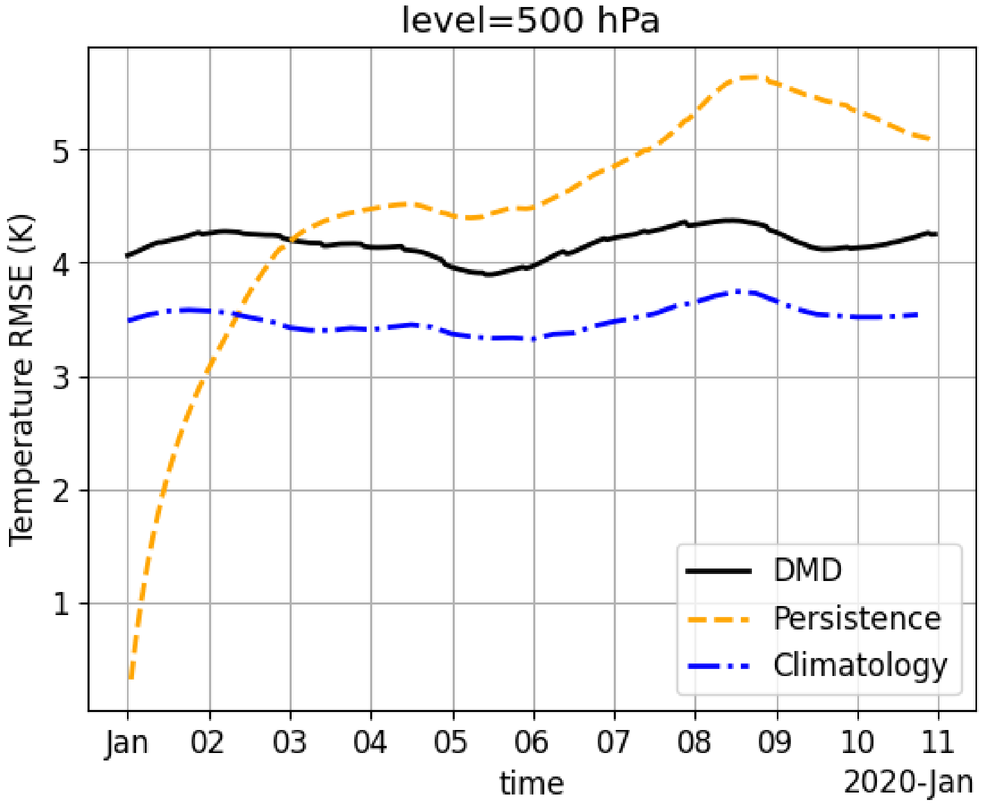
Preliminary results

DMD model trained on Dec 2019 only, Δt = 1h

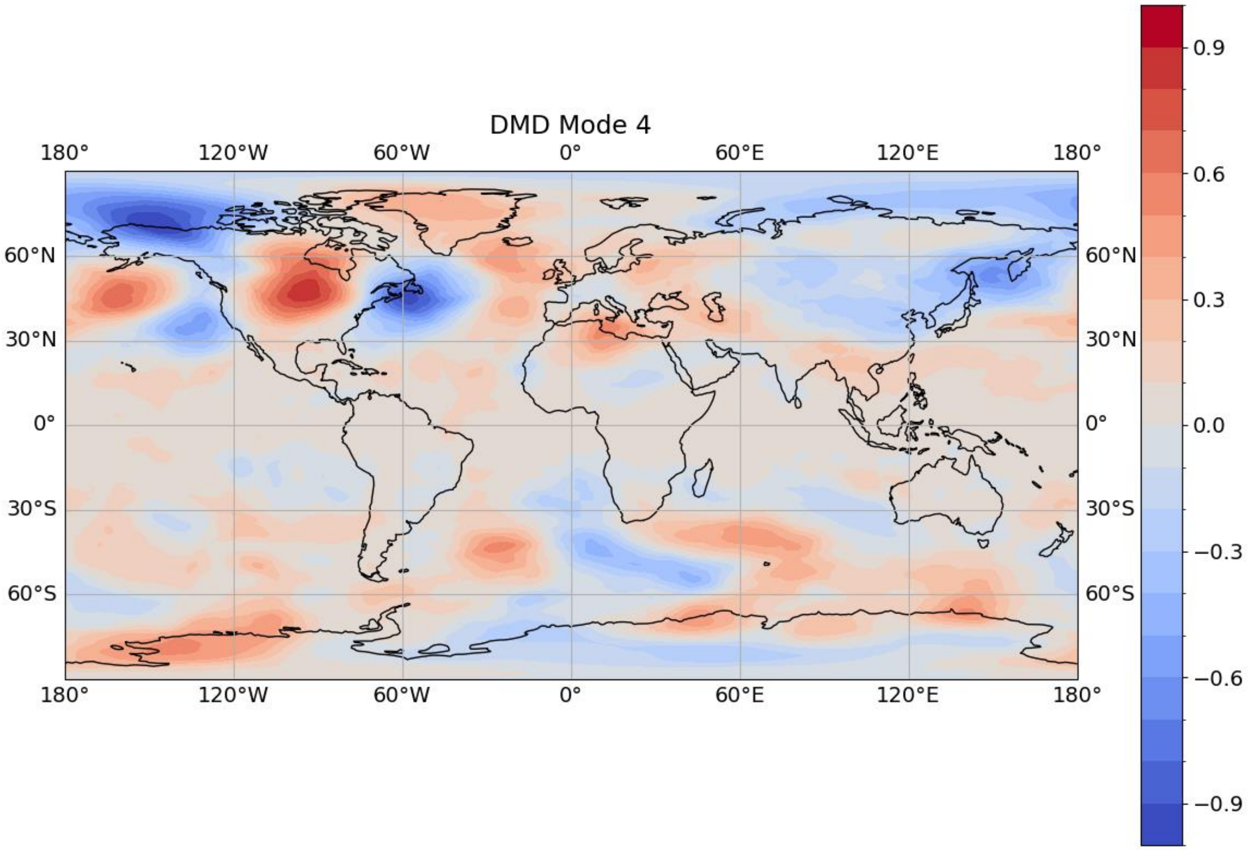


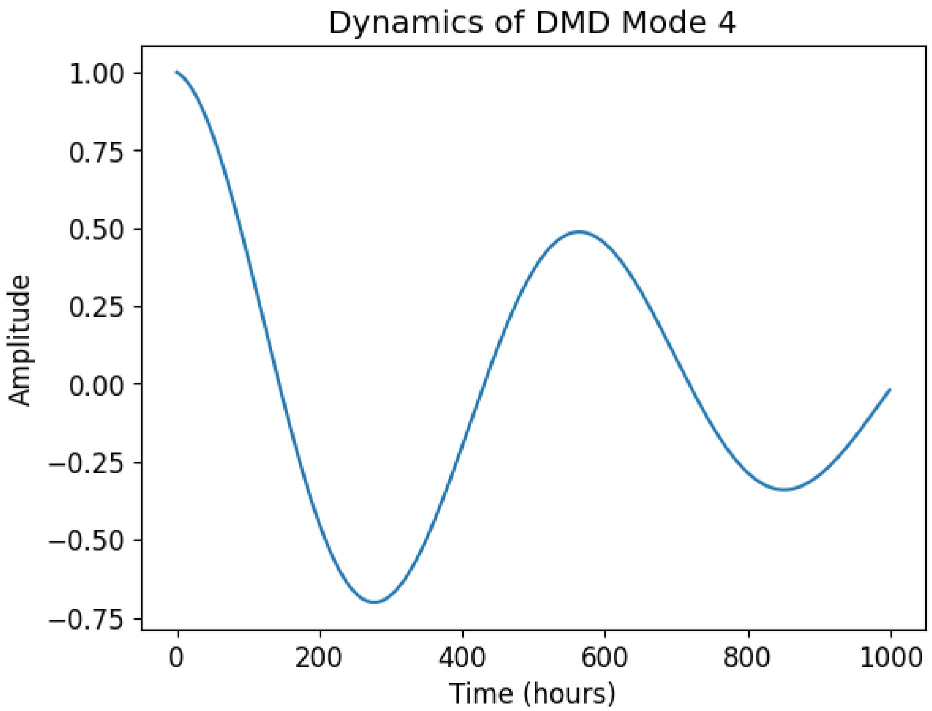
Combination of two DMD models:

- model trained on Dec 2019, Δt = 1h
- model trained on Jan 2018 – Dec 2019, Δt = 1d



Explainability in DMD





The DynaModERA (DMD-ERA5) package

- A Python package for straightforward and efficient DMD on the ERA5 dataset: https://github.com/ClimeTrend/DynaModERA
- Capabilities:
  - ERA5 download and slicing
  - Pre-processing
  - Singular Value Decomposition (either standard or randomized)
  - Integrated Data Version Control (DVC) - https://dvc.org
  - Integrated application of DMD using PyDMD (in progress)
  - Post-processing (future work)
  - Parallelized SVD using PyLOM: https://github.com/ArnauMiro/pyLowOrder (future work)
- Contributions welcome!