# Autoencoders vs EOF

ML Journal Club / August 6, 2025

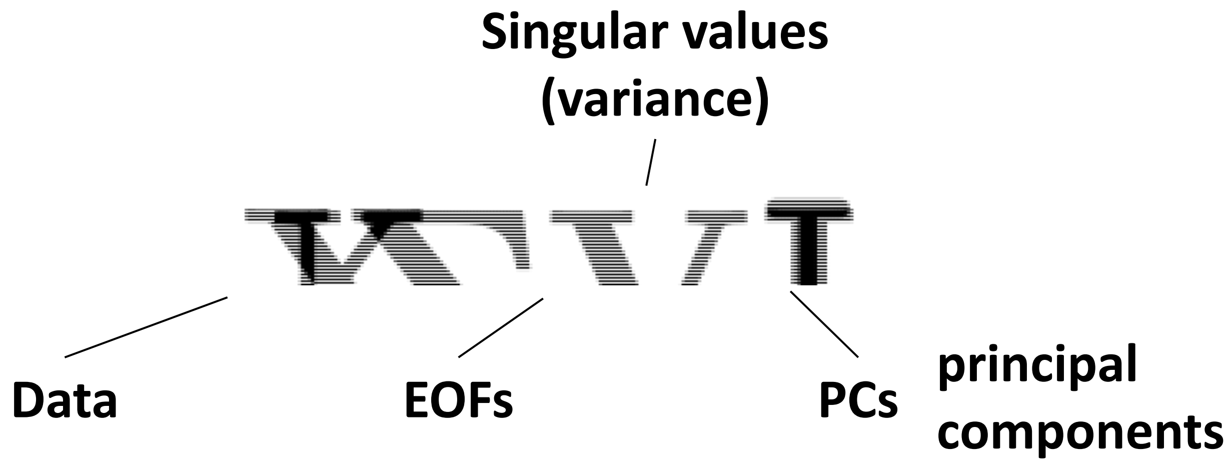
## Why reduce dimensionality?

Gridded field at 5x5 deg → 2592 grid points for each timestep and variable

For practical part, we want to find the most efficient way to encode the **2592-dimensional** information with just **5 dimensions**

Climate information is redundant since fields often follow coherent patterns and have high spatial autocorrelation (seasonal cycle, ENSO, PDO, …)

# Empirical orthogonal functions (EOFs), aka SVD, PCA, POP, etc.



# Empirical orthogonal functions (EOFs), aka SVD, PCA, POP, etc.

**Singular values (variance)**

$$X = U \Sigma V^T$$

**Data**

**EOFs**

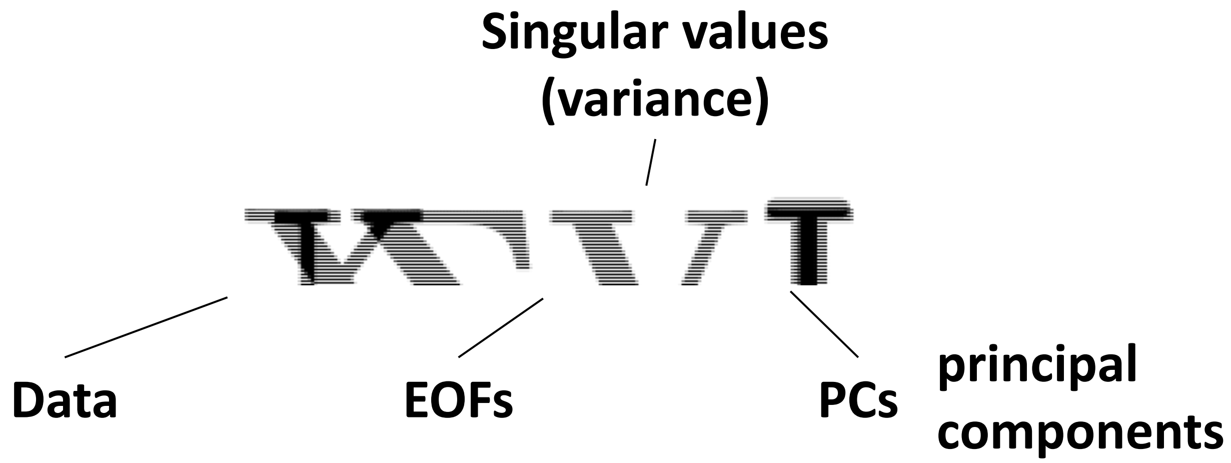
**PCs** **principal components**

$\color{blue}{m}$ x $\color{red}{n}$

**$\color{blue}{m}$-dim.**

**over $\color{red}{n}$ times**

# Empirical orthogonal functions (EOFs), aka SVD, PCA, POP, etc.



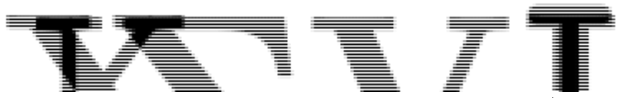
m x n

2592 grid points
over 1849 times

# Empirical orthogonal functions (EOFs), aka SVD, PCA, POP, etc.

m x n

Singular values (variance)



Data

m x n

2592 grid points over 1849 times

EOFs

m x m

PCs

n x n

principal components

## EOFs under the hood

$X = U\Sigma V^T$

[illegible]

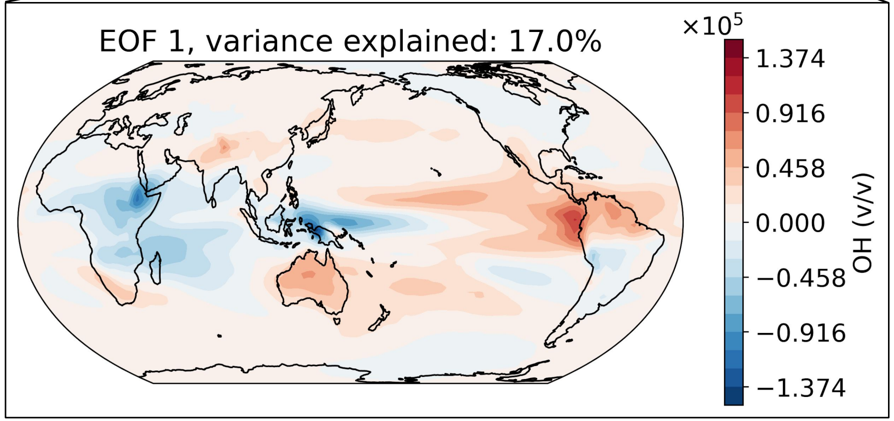
m-dim. vector

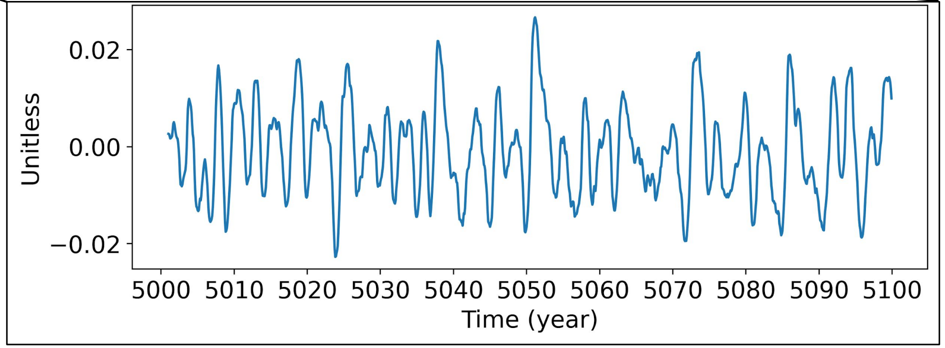
over n times

# EOFs under the hood

[illegible]

[illegible]



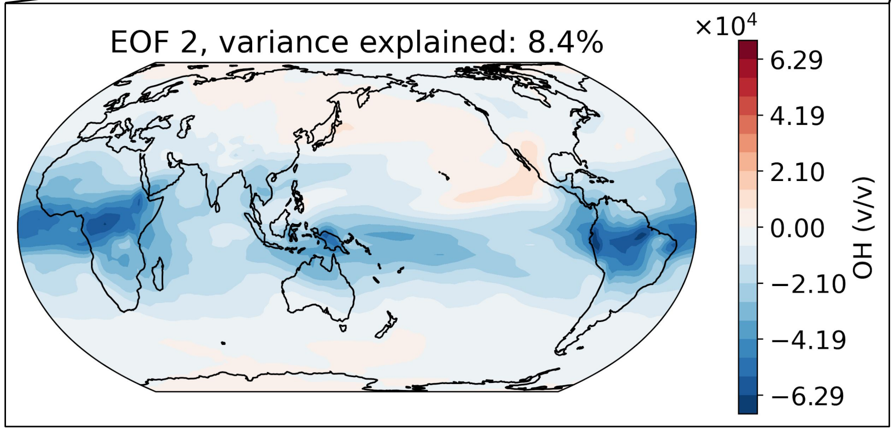


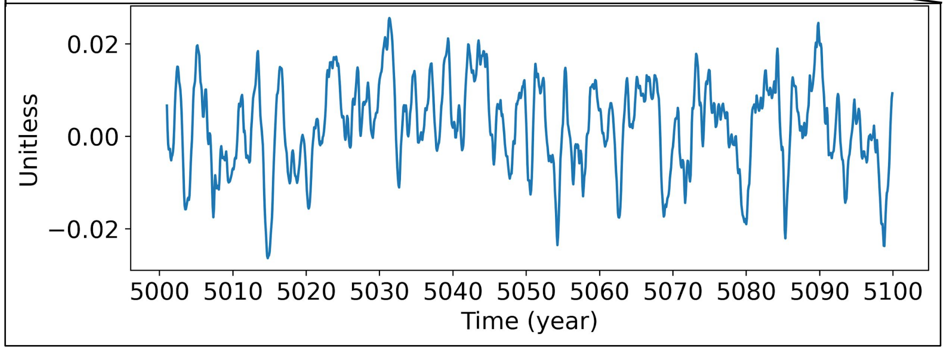
**m-dim. vector**

**over n times**

## EOFs under the hood

[illegible]



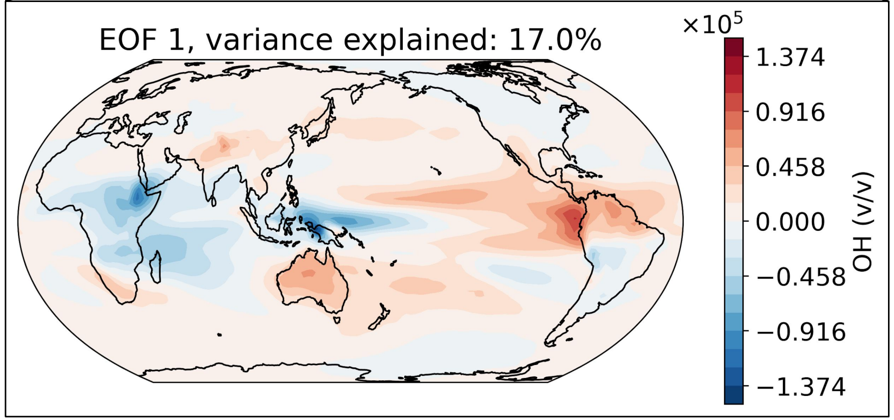


**m-dim. vector**

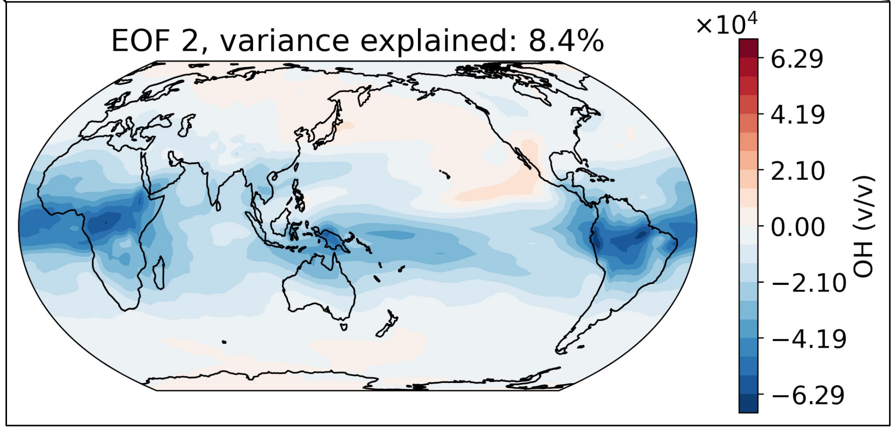
**over n times**

## EOFs under the hood

[illegible]



**m-dim. vector**

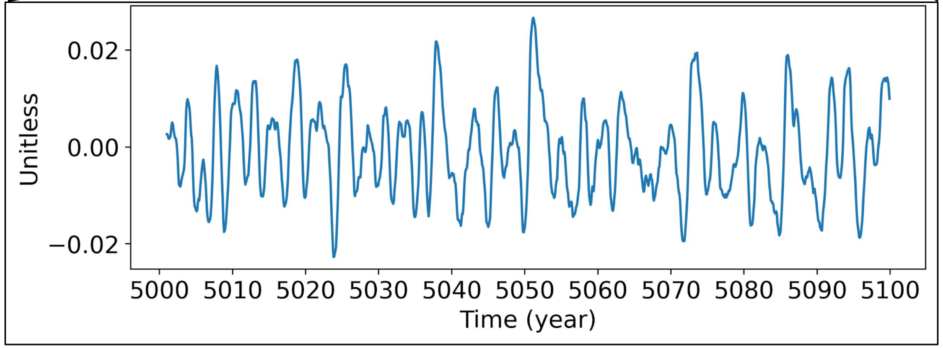


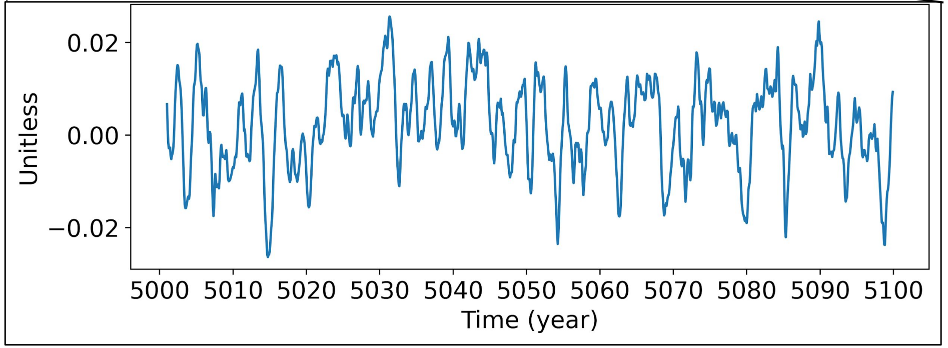
**m-dim. vector**

## EOFs under the hood

[illegible]

[illegible]





**over n times**

**over n times**

# EOFs under the hood are matrix multiplications

[illegible]

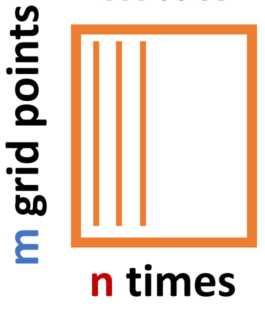
**m-dim. vector**
**over n times**

$$X = U\Sigma V^T$$

**Data**
**m x n**

**EOFs**
**m x m**

**PCs**
**n x n**



## EOFs under the hood are matrix multiplications

$x(t) =$ [illegible]

m-dim. vector
over n times

[illegible]

Data
m x n

EOFs
m x m

PCs
n x n

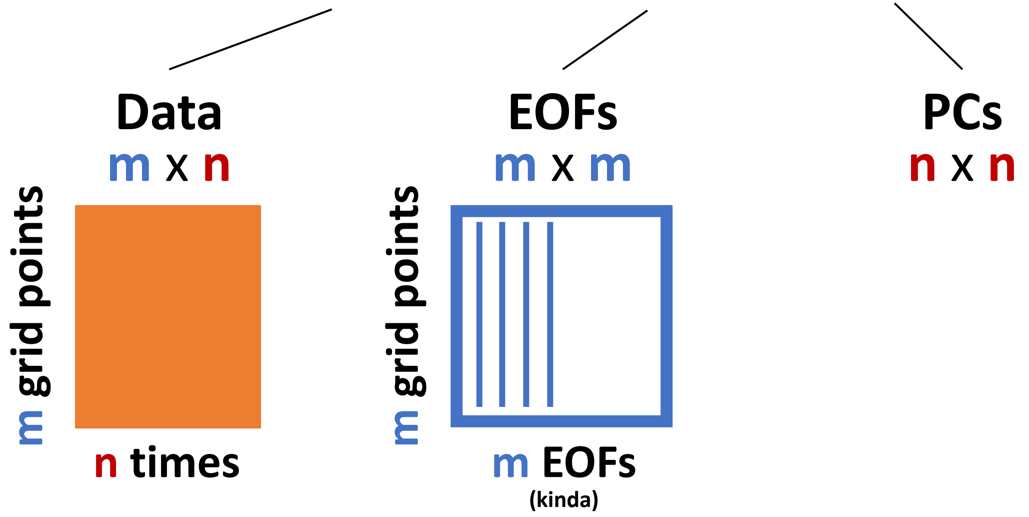
m grid points
n times

## EOFs under the hood are matrix multiplications

$$x(t) = a_1\phi_1(t) + \phi_2 a_2(t)$$

m-dim. vector
over n times

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{T}$$



# EOFs under the hood are matrix multiplications

$$x(t) = \alpha_1 \phi_1(t) + \phi_2 \alpha_2(t)$$

m-dim. vector

over n times

$$X = U \Sigma V^T$$

Data

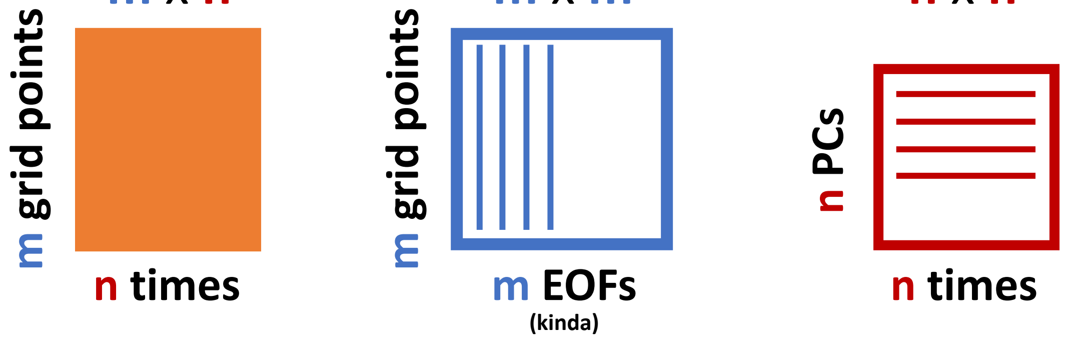
m x n

EOFs

m x m

PCs

n x n

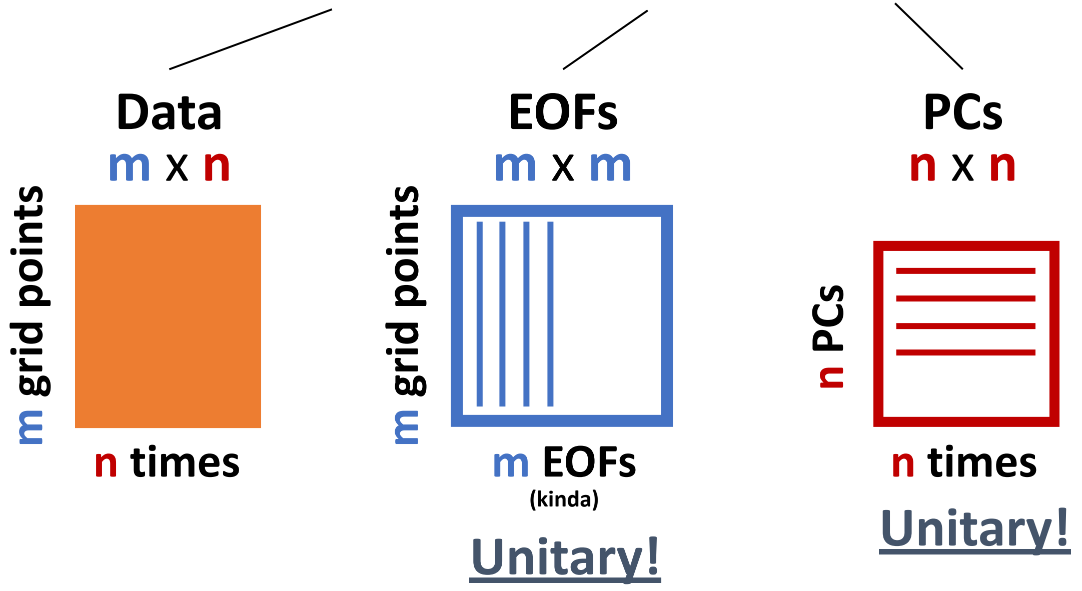


# EOF and PC matrices are unitary (orthonormal)

$x(t) =$ [illegible]

m-dim. vector
over n times

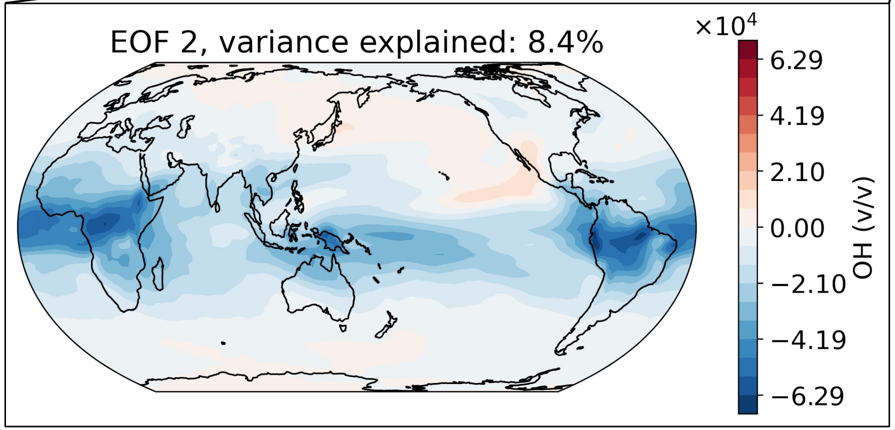
$X = U\Sigma V^T$

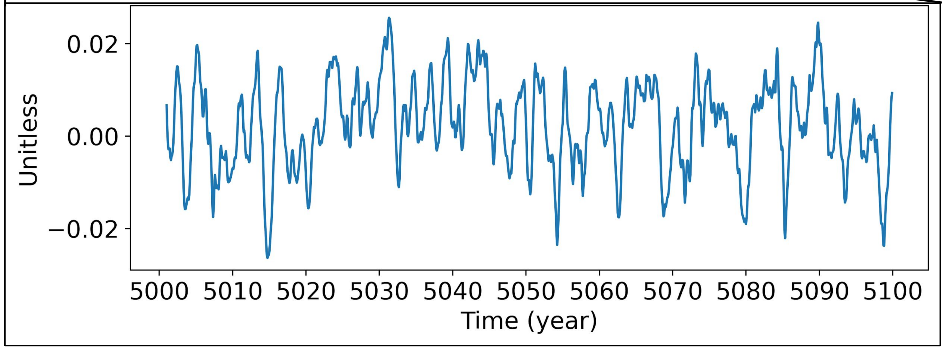


## EOF truncation

**m-dim. vector**
**over n times**

[illegible]





**m-dim. vector**

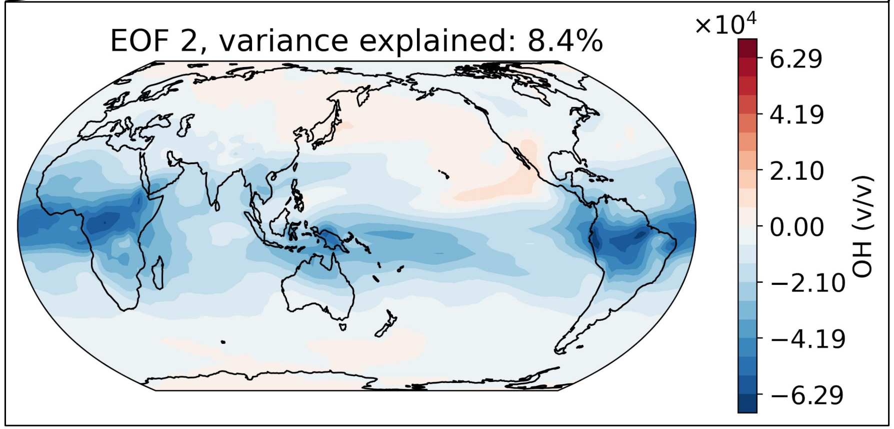
**over n times**

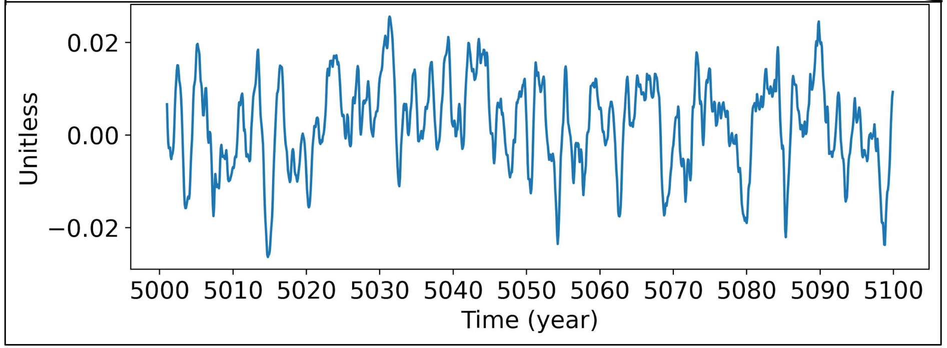
# EOF truncation

[illegible]

m-dim. vector
over n times

[illegible]





[illegible]

m-dim. vector
over n times

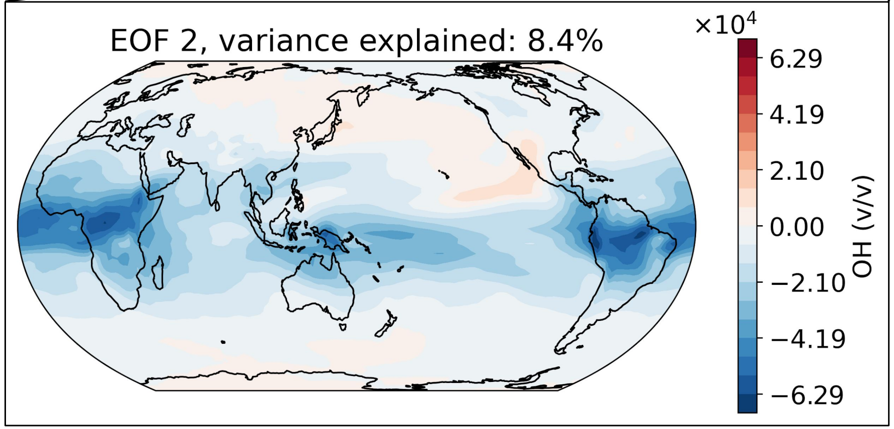
represented by m̃ vectors

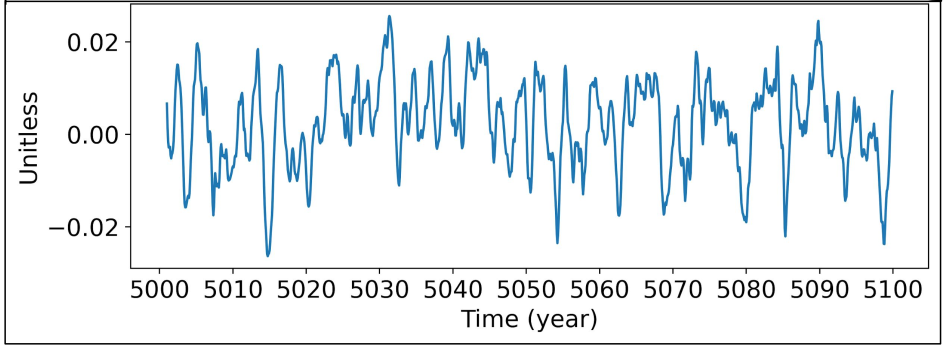
## EOF truncation

$$X \in \mathbb{R}^{m \times n}$$

m-dim. vector over n times

$$x(t) = \underbrace{e_1}\underbrace{c_1(t)} + \cdots + e_m c_m(t)$$





$$\tilde{x}(t) = e_1 c_1(t) + \cdots + e_{\tilde{m}} c_{\tilde{m}}(t)$$

m-dim. vector over n times

represented by m̃ vectors

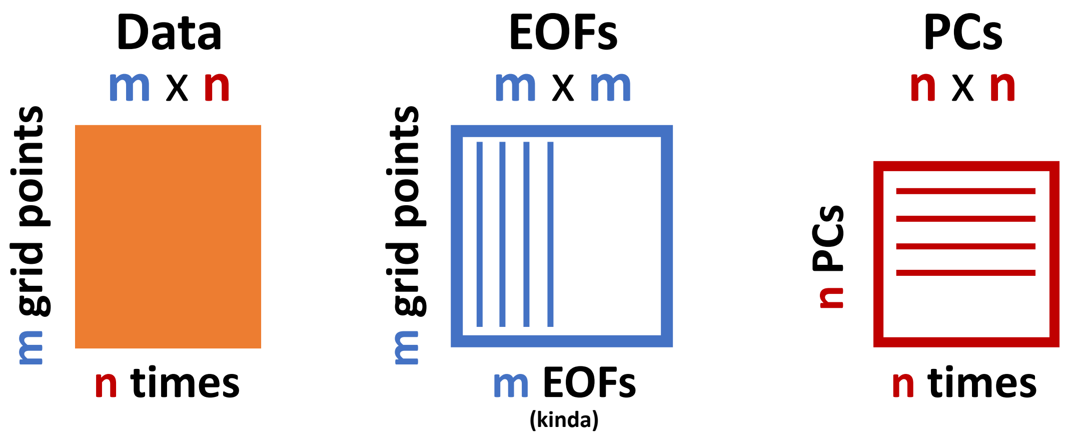
$$\tilde{X} \in \mathbb{R}^{\tilde{m} \times n}$$

## EOF truncation

$x(t) = a_1\phi_1(t) + a_2\phi_2(t)$ [illegible]

**m-dim. vector**
**over n times**

$\mathbf{X} =$ [illegible] $\mathbf{T}$

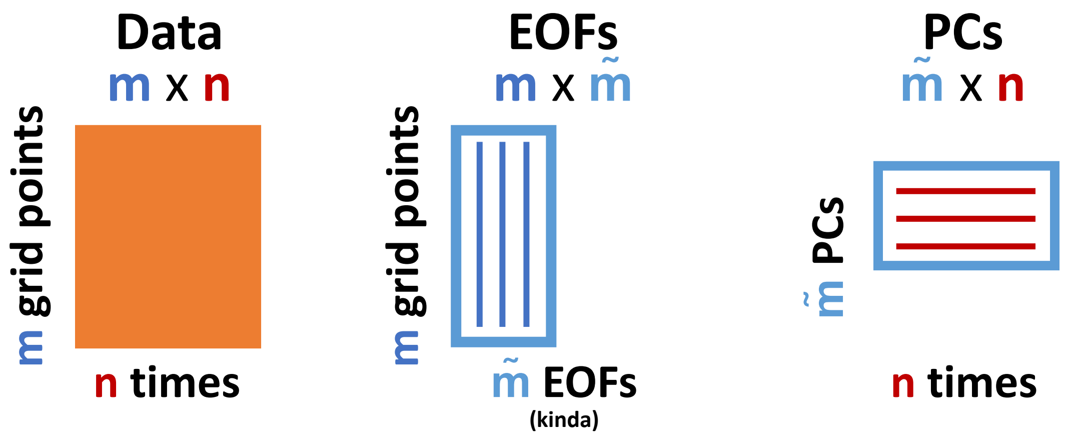


## EOF truncation

$\tilde{x}(t) =$ [illegible]

m-dim. vector
over n times

$$\tilde{X} = \tilde{E}\,\tilde{P}^T$$



# Projection to and from latent (EOF) space

[illegible]

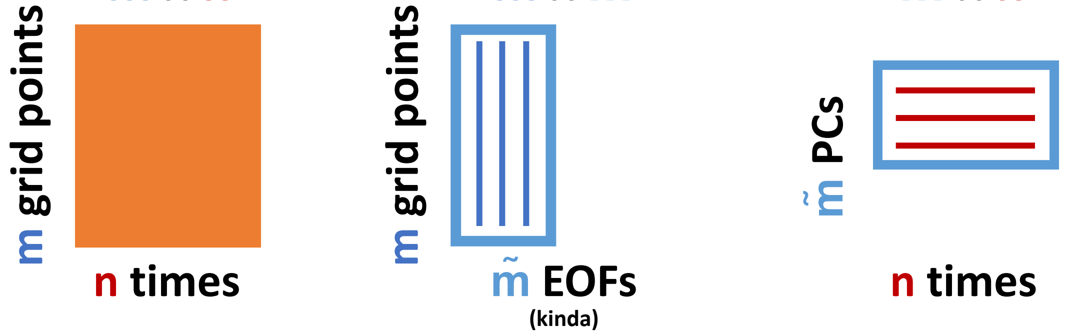
m-dim. vector
over n times

[illegible]

"state space" Data
m x n

EOFs
m x m̃

PCs
m̃ x n "latent space"



# Projection to and from latent (EOF) space

[illegible]

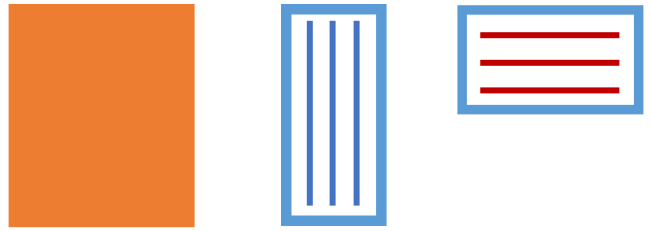
m-dim. vector

over n times

[illegible]

"state space" Data m x n EOFs m x m̃ PCs m̃ x n "latent space"

latent $\rightarrow$ state [illegible]



## Projection to and from latent (EOF) space

[illegible]

m-dim. vector
over n times

[illegible]

"state space" Data
$m \times n$

EOFs
$m \times \tilde{m}$

PCs
$\tilde{m} \times n$ "latent space"

latent $\longrightarrow$ state [illegible]

state $\longrightarrow$ latent [illegible]

## Singular values tell you the variance of each EOF/PC pair

$X = U \Sigma V^T$

Singular values
(variance)
m x n

$$\begin{bmatrix} \delta_1 & & \\ & \delta_2 & \\ & & \delta_3 \\ & & \end{bmatrix}$$

$$\delta_i^2 = \sigma_i^2 n$$

Variance

## Singular values tell you the variance of each EOF/PC pair

$U \Sigma V^T$

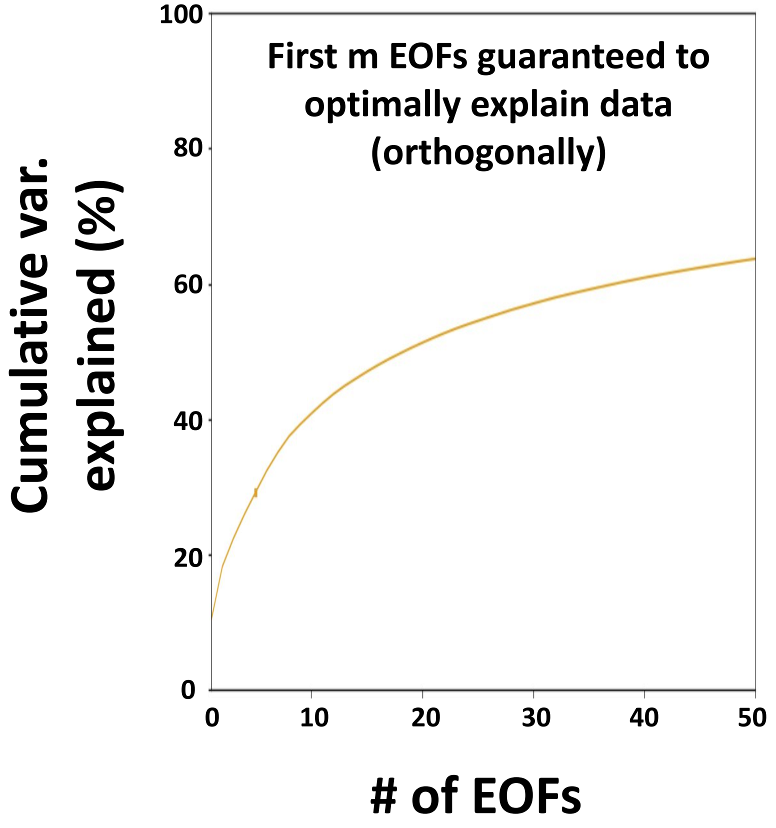
Singular values
(variance)
m x n

$$\boxed{\begin{matrix}\delta_1 & & \\ & \delta_2 & \\ & & \delta_3\end{matrix}}$$

$\delta_i^2 =$

Variance

$\sigma_i^2 n$



## EOFs do not necessarily encode “dynamically relevant” info

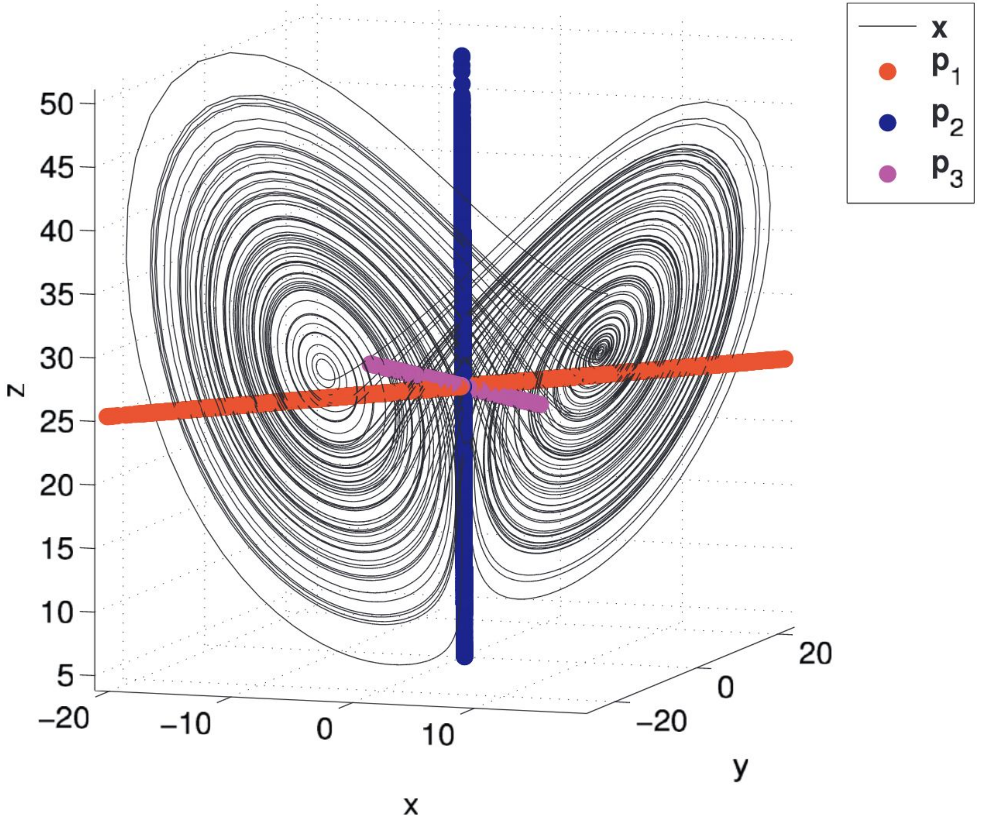
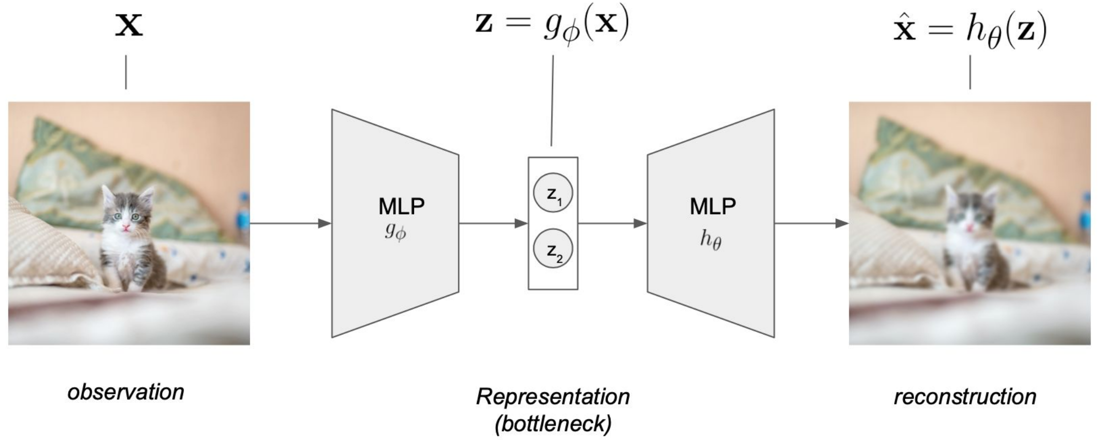


FIG. 3. Lorenz (1963) attractor $\mathbf{x}(t)$ for standard parameters producing a strange attractor. The colored lines are the projections of $\mathbf{x}(t)$ onto the three EOF modes: $\mathbf{p}_j(t) = [\mathbf{x}(t) \cdot \mathbf{e}_j]\mathbf{e}_j$. Redrafted following Mo and Ghil (1987).

Monahan et al. (2009)

## Autoencoders = empirical (non)orthogonal functions (ENOF)

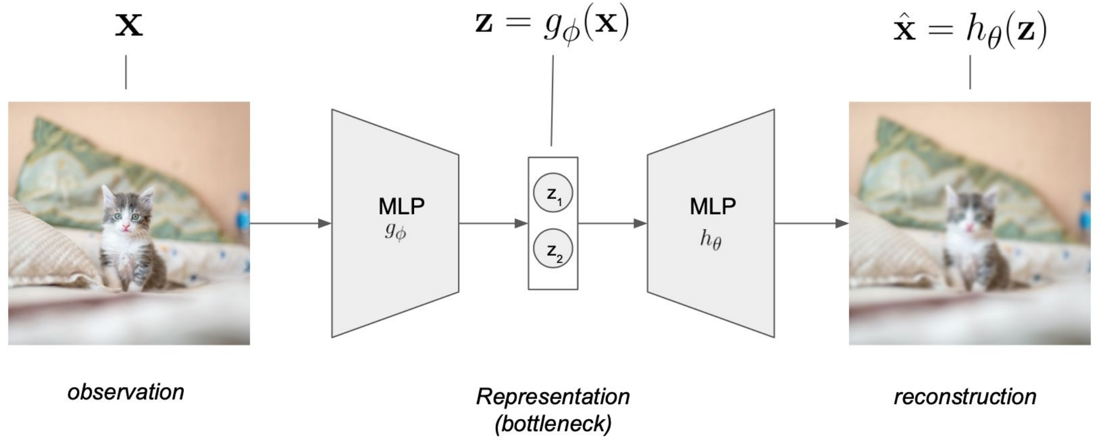

$\mathbf{x}$
$\mathbf{z} = g_\phi(\mathbf{x})$
$\hat{\mathbf{x}} = h_\theta(\mathbf{z})$
MLP
$g_\phi$
$z_1$
$z_2$
MLP
$h_\theta$
*observation*
*Representation (bottleneck)*
*reconstruction*

Reconstruct inputs with reduced latent space

Similar to EOFs, but not orthogonal or linear

# MLPs similar in spirit to EOF mappings

**state $\longrightarrow$ latent  latent $\longrightarrow$ state**


$\mathbf{x}$
$\mathbf{z} = g_\phi(\mathbf{x})$
$\hat{\mathbf{x}} = h_\theta(\mathbf{z})$
MLP $g_\phi$
$z_1$
$z_2$
MLP $h_\theta$
observation
Representation (bottleneck)
reconstruction

[illegible]  [illegible]

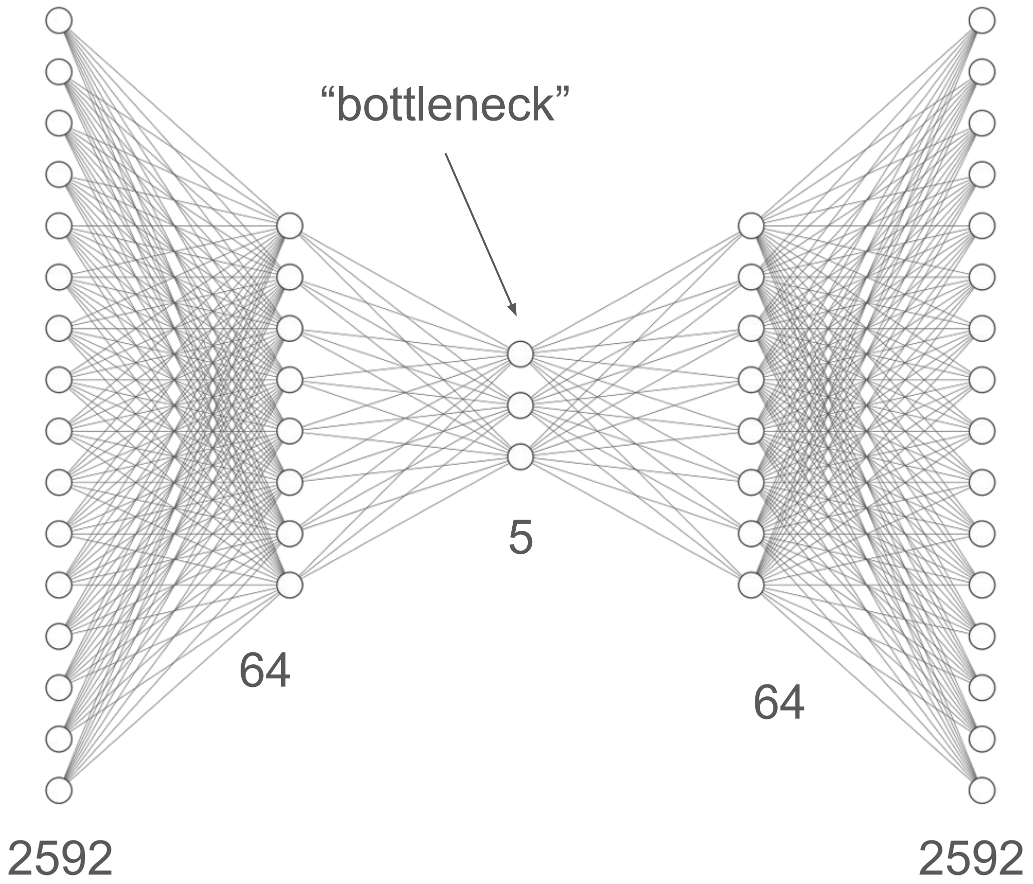
Reconstruct inputs with reduced latent space

Similar to EOFs, but not orthogonal or linear

# Architecture we will be using: Fully Connected, 4 layers

To train: MSE loss of reproducing input as output

“Latent space” is 5-dimensional “bottleneck” activations



**https://github.com/DominikStiller/mljc-autoencoder-workshop**