How to model the Ocean Circulation Eddies using Artificial Neural Networks (OCEANN)

Iuri Gorenstein
Ilana Wainer

C²
Laboratório de Oceanografia Física, Clima e Criosfera
Instituto Oceanográfico da Universidade de São Paulo (IO-USP)

Summary

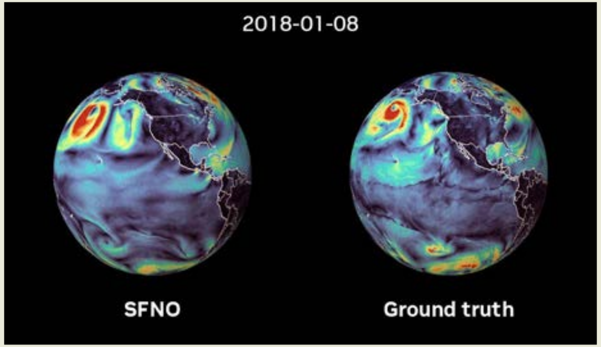
- Neural Networks and Climate Modelling
- Regional Model Barotropic Integration
- Architecture and Loss Functions
- Results

Introduction

Neural networks (NNs) are mostly used alongside traditional ocean models to improve simulations and predictions. The Spherical Fourier Neural Operator (bottom right figure) has been successful in emulating long-term atmospheric and ocean circulation dynamics.

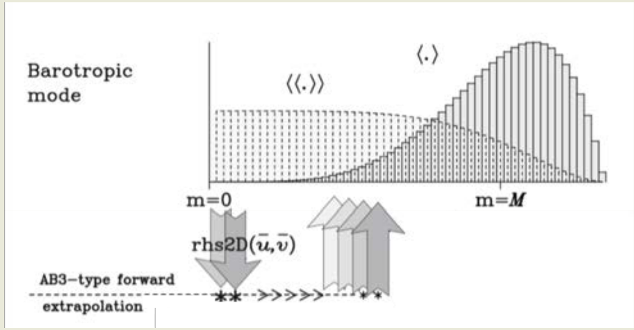
However, if you wish to apply it to a different domain with constraints different from those used during its training, it will need to be retrained.

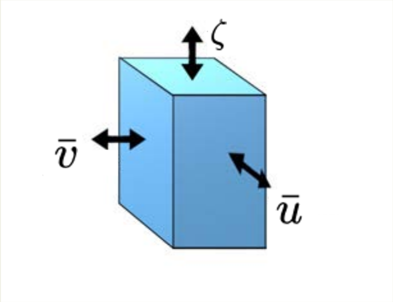
We propose using regional ocean model simulations to train a NN with inherently integrated physical principles to evolve the discrete Navier-Stokes equations across different spatial resolutions and domains.



Regional Ocean Model (ROMS/CROCO)

The Barotropic approximation refer to systems where the pressure depends only on height, ignoring vertical pressure gradients and variations.





In this approximation, the state of the system is fully determined by the barotropic momentum, and the geopotential height or pressure horizontal fields

Regional Ocean Model (ROMS/CROCO)

Generalized forward-backward (predictor-corrector)

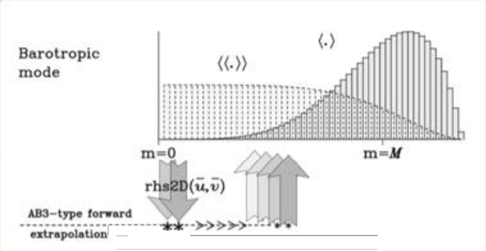
1. AB3-type extrapolation

$$D^{m+\frac{1}{2}} = H + \left(\frac{3}{2} + \beta\right)\zeta^m - \left(\frac{1}{2} + 2\beta\right)\zeta^{m-1} + \beta\zeta^{m-2}$$

$$\overline{u}^{m+\frac{1}{2}} = \left(\frac{3}{2} + \beta\right)\overline{u}^m - \left(\frac{1}{2} + 2\beta\right)\overline{u}^{m-1} + \beta\overline{u}^{m-2}$$

2. Integration of ζ

$$\zeta^{m+1} = \zeta^m - \Delta\tau\,\partial_x(D^{m+\frac{1}{2}}\overline{u}^{m+\frac{1}{2}})$$



Regional Ocean Model (ROMS/CROCO)

Generalized forward-backward (predictor-corrector)

1. AB3-type extrapolation

$$
\begin{aligned}
D^{m+\frac{1}{2}} &= H + \left(\frac{3}{2}+\beta\right)\zeta^{m} - \left(\frac{1}{2}+2\beta\right)\zeta^{m-1} + \beta\zeta^{m-2} \\
\overline{u}^{m+\frac{1}{2}} &= \left(\frac{3}{2}+\beta\right)\overline{u}^{m} - \left(\frac{1}{2}+2\beta\right)\overline{u}^{m-1} + \beta\overline{u}^{m-2}
\end{aligned}
$$

2. Integration of ζ

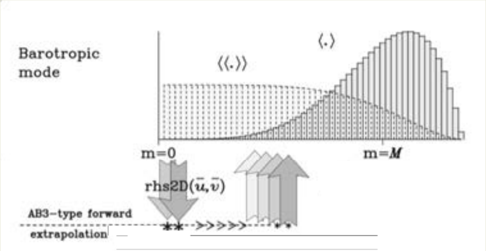
$$\zeta^{m+1} = \zeta^{m} - \Delta\tau\, \partial_x (D^{m+\frac{1}{2}}\overline{u}^{m+\frac{1}{2}})$$

3. AM4 interpolation

$$\zeta^{\star} = \left(\frac{1}{2}+\gamma+2\varepsilon\right)\zeta^{m+1} + \left(\frac{1}{2}-2\gamma-3\varepsilon\right)\zeta^{m} + \gamma\zeta^{m-1} + \varepsilon\zeta^{m-2}$$

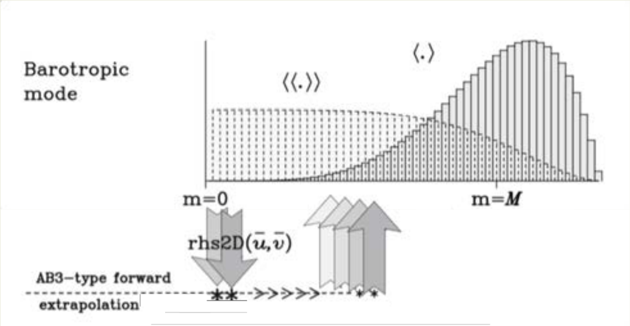
4. Integration of $\bar{u}$

$$\overline{u}^{m+1} = \frac{1}{D^{m+1}}\left[D^{m}\overline{u}^{m} + \Delta\tau\ \mathrm{RHS2D}(D^{m+\frac{1}{2}}, \overline{u}^{m+\frac{1}{2}}, \zeta^{\star})\right]$$



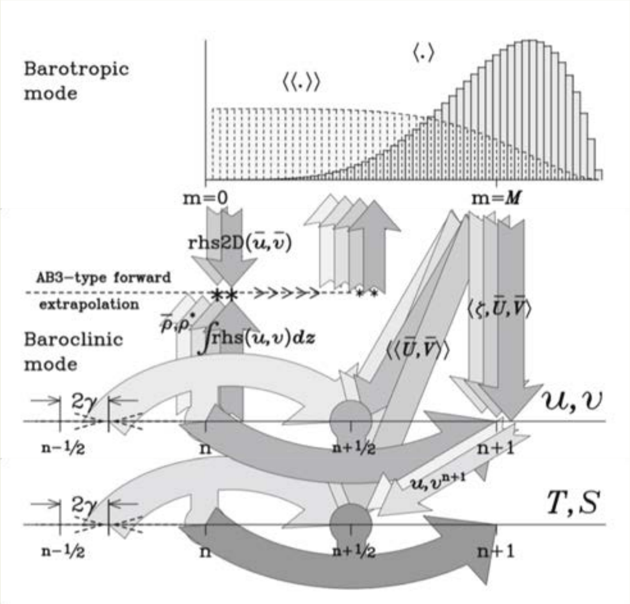
Regional Ocean Model (ROMS/CROCO)

Fast step



Regional Ocean Model (ROMS/CROCO)

Slow step



CROCO Barotropic Simulation

To train a network to simulate the dynamics of the ocean system similarly to a regional model, we must start with the barotropic dynamics.

A barotropic simulation of dt=15s is used to train the neural network. At this stage, the simulation used to train and validate the network is not realistic. The goal is simply to enable the network to learn and embed the dynamics of the regional model.

The training set used is the South Atlantic Circulation System at low resolution (⅓ of degree resolution) and the cross validation is done with the Benguela region at high resolution (1/12 of degree resolution).

Input Vectors

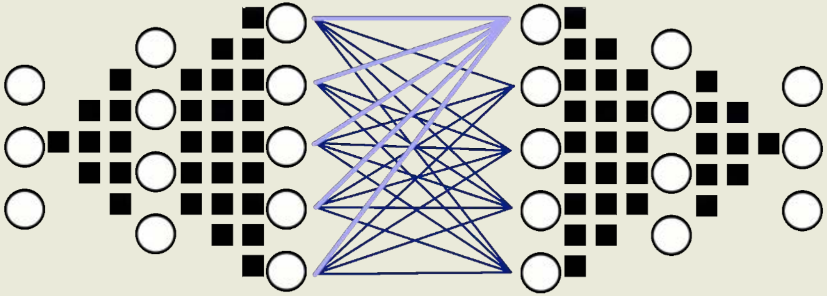
$$x_t = x_t(\zeta_t, \bar{u}_t, \bar{v}_t)$$

At each grid point, we use the simulations' barotropic variables, Coriolis parameter, and topography, both present and past, to define a vector ($\boldsymbol{x}_t$) that represents that grid point and its neighbors at that time.

This vector will be used to train the neural network to model the evolution of barotropic sea surface height.

Consistent Koopman AutoEncoder

The input vector enters the network through the encoder

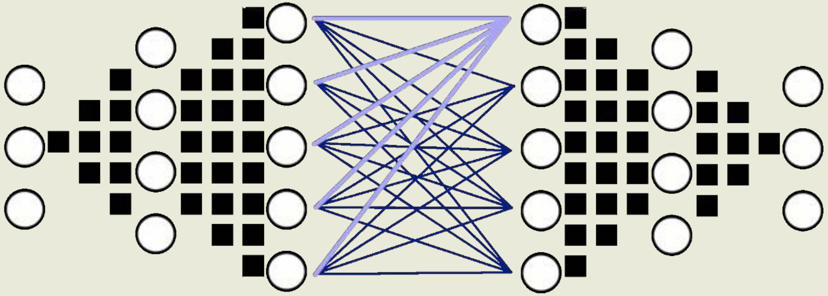


$x_t \quad \longrightarrow \quad z_t$

encoder

Consistent Koopman AutoEncoder

The dynamics is evolved in the high dimensional latent space

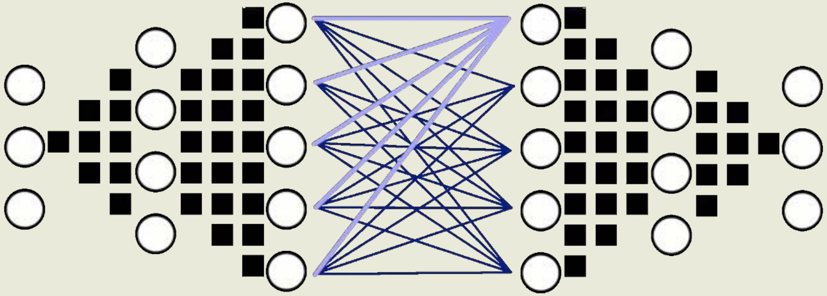


$$z_t \longrightarrow z_{t+1}$$

linear
operator

Consistent Koopman AutoEncoder

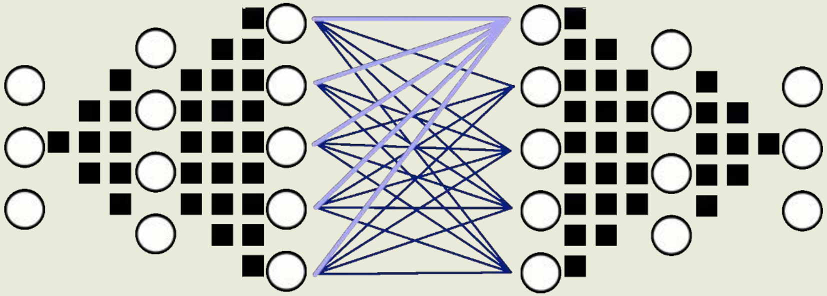
The modeled system is recovered through the decoder



$z_{t+1} \longrightarrow x_{t+1}$

decoder

Consistent Koopman AutoEncoder

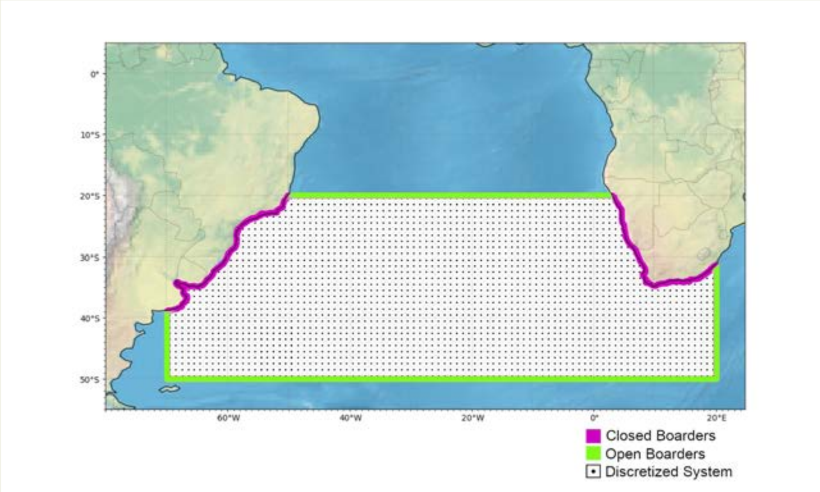


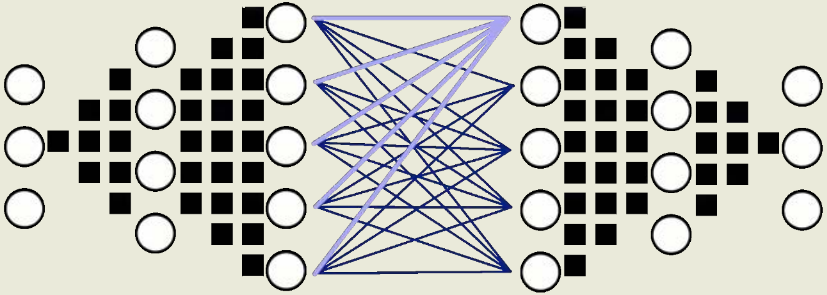
$$x_t \longrightarrow z_t \longrightarrow z_{t+1} \longrightarrow x_{t+1}$$

encoder   linear operator   decoder

Consistent Koopman AutoEncoder

This network evolves each grid point in parallel





$$x_t \quad \longrightarrow \quad z_t \quad \longrightarrow \quad z_{t+1} \quad \longrightarrow \quad x_{t+1}$$

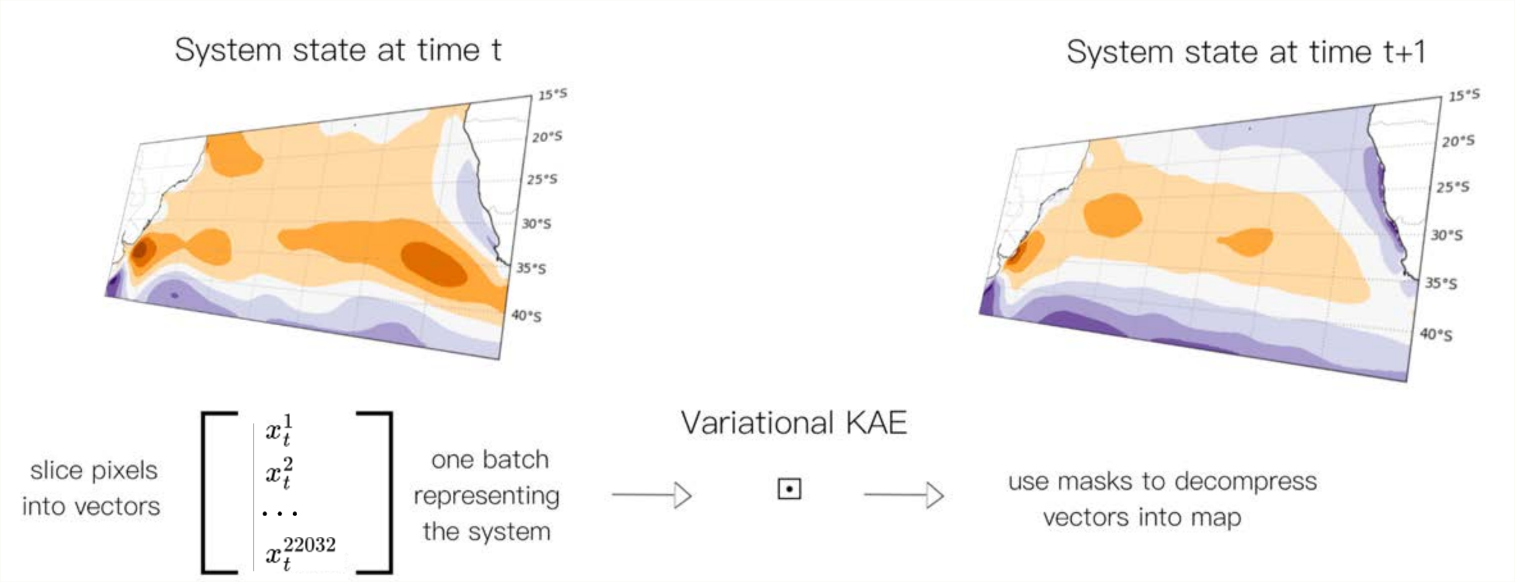
encoder

linear operator

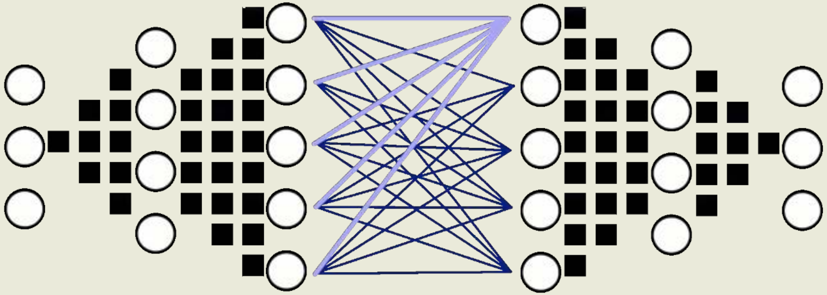
decoder

Consistent Koopman AutoEncoder

South Atlantic at low resolution (204x108 pixels) - Training Set
Benguela Region at High Resolution (362x461 pixels) - Validation Set



Consistent Koopman AutoEncoder



Forward operator

$z_{t+1} \longrightarrow x_{t+1}$

$\nearrow$

decoder

$x_t \longrightarrow z_t$

encoder

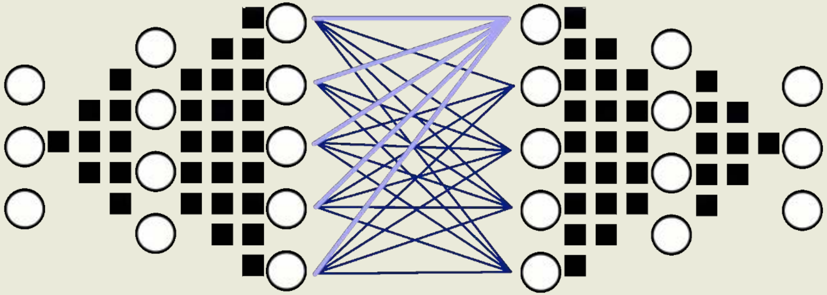
$\searrow$

$z_{t-1} \longrightarrow x_{t-1}$

Backward operator

decoder

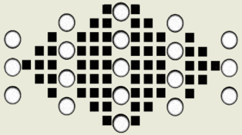
Consistent Koopman AutoEncoder



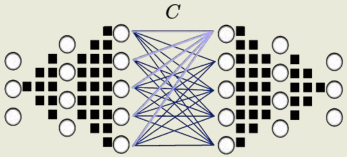
$$x_t \xrightarrow{\text{encoder}} z_t \quad \begin{array}{l} \overset{\text{Forward operator}}{\nearrow} \quad z_{t+1} \xrightarrow{\text{decoder}} x_{t+1} \\[2em] \underset{\text{Backward operator}}{\searrow} \quad z_{t-1} \xrightarrow{\text{decoder}} x_{t-1} \end{array}$$

One autoencoder
Two linear operators

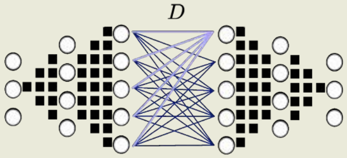
Loss Functions



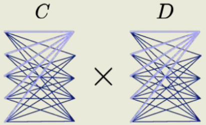
$$\varepsilon_{id} = \frac{1}{2n}\sum_{t=1}^{n} ||x_t - \hat{x}_t||^2$$



$$\varepsilon_{fwd} = \frac{1}{2n}\sum_{t=1}^{n} ||x_{t+1} - \hat{x}_{t+1}||^2$$



$$\varepsilon_{bwd} = \frac{1}{2n}\sum_{t=1}^{n} ||x_{t-1} - \hat{x}_{t-1}||^2$$



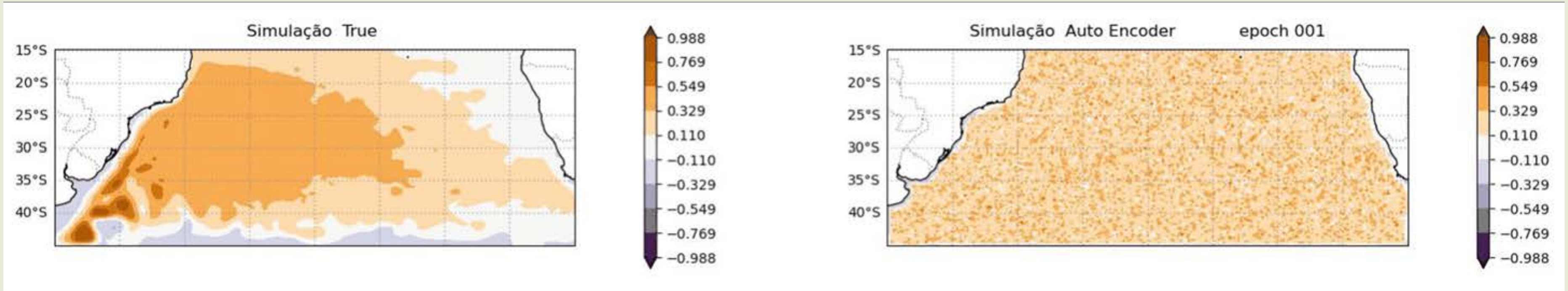
$$\varepsilon_{con} = \sum_{k=1}^{n} \frac{1}{2k}||D_{k*}C_{*k} - Id||_F^2 - \frac{1}{2k}||C_{k*}D_{*k} - Id||_F^2$$

Results

The results presented here are from a network with approximately 120,000 parameters. The encoder and decoder each have 2 hidden layers with 128 neurons, while the linear operator has 100 neurons. The model was trained for 300 epochs, which took 10 hours on a single GPU.

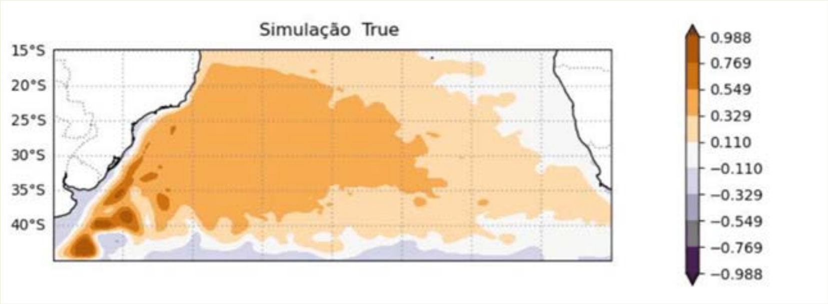
Results

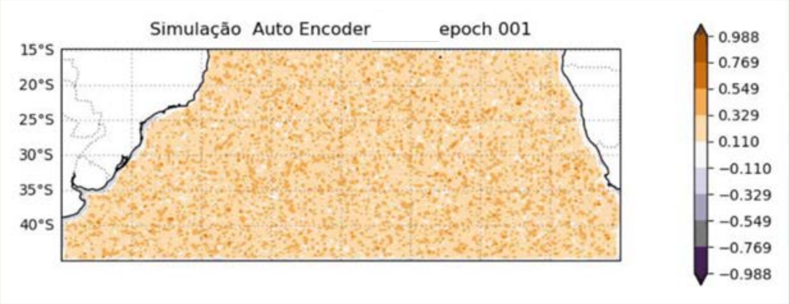
Auto-Encoder Training

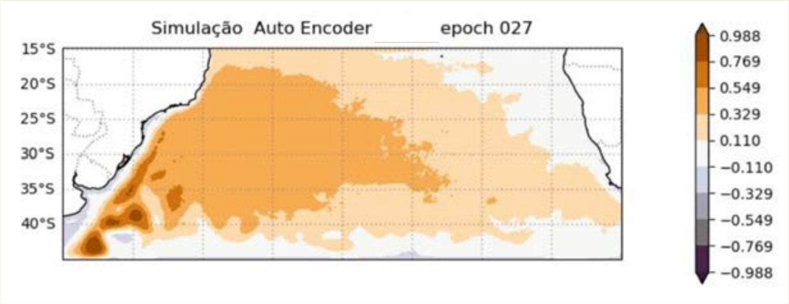


Results

Auto-Encoder Training





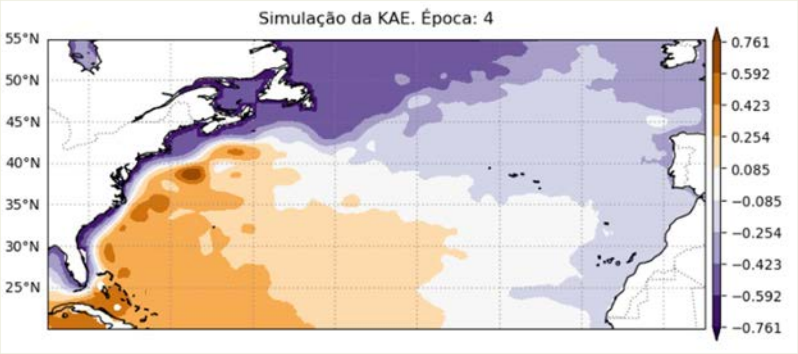


Results

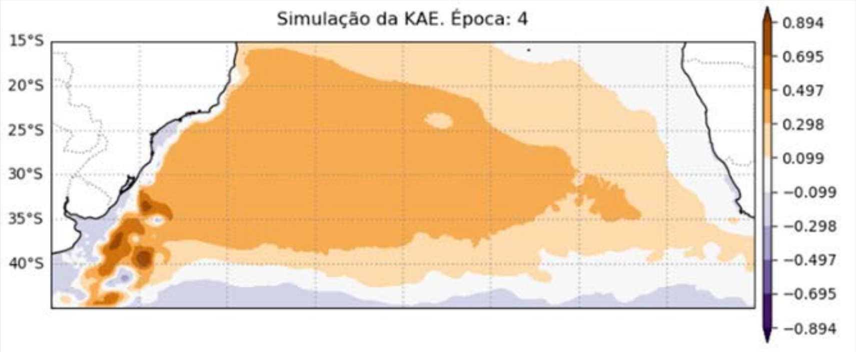
The network can encode the sea surface height at different ocean domains and resolutions.

Results

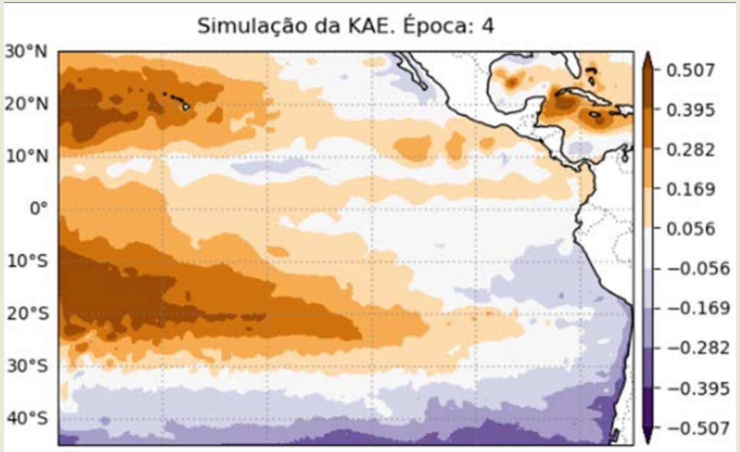
North Atlantic Circulation System



High Resolution SACS



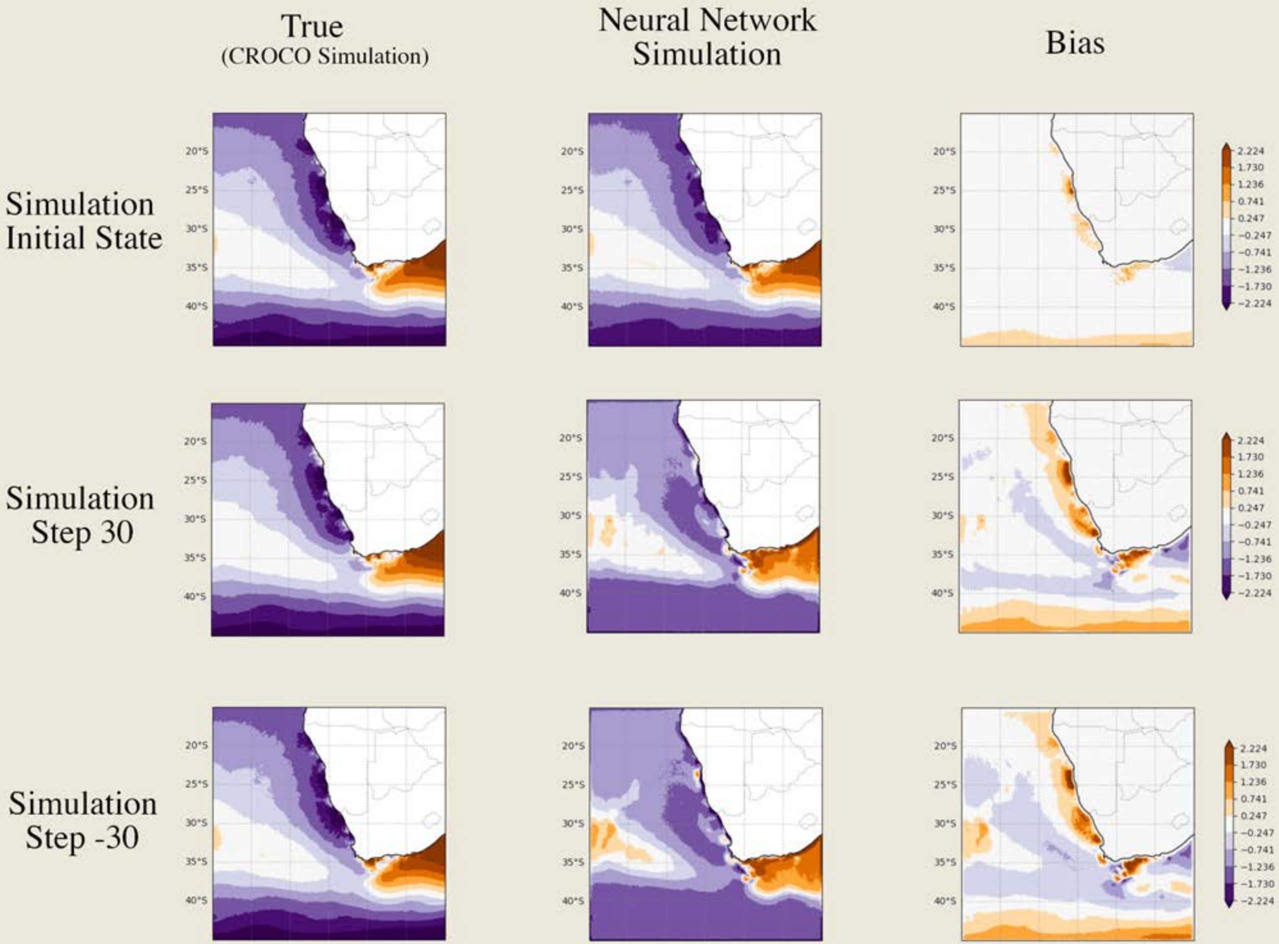
West Pacific Circulation System



Results

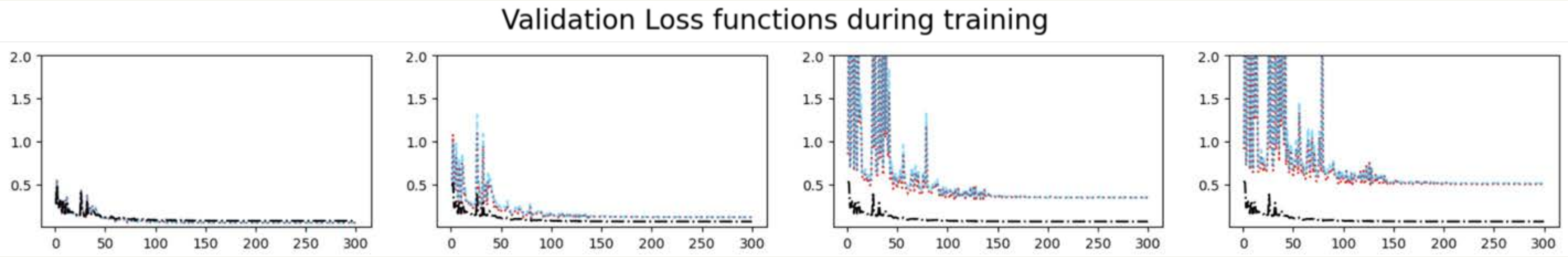
The evolution of the system's dynamics is accumulating errors at each time step.

While the error for a single step is small, after 60 iterations of the barotropic simulation, the results diverge from the true values.


True
(CROCO Simulation)
Neural Network Simulation
Bias
Simulation Initial State
Simulation Step 30
Simulation Step -30
20°S
25°S
30°S
35°S
40°S
2.224
1.730
1.236
0.741
0.247
−0.247
−0.741
−1.236
−1.730
−2.224

Results

Loss functions per epoch of Training. From left to right: loss in 1 integration step, 5 steps, 30 steps, and 60 steps. Loss identity in black, loss forward in blue, loss backward in red.

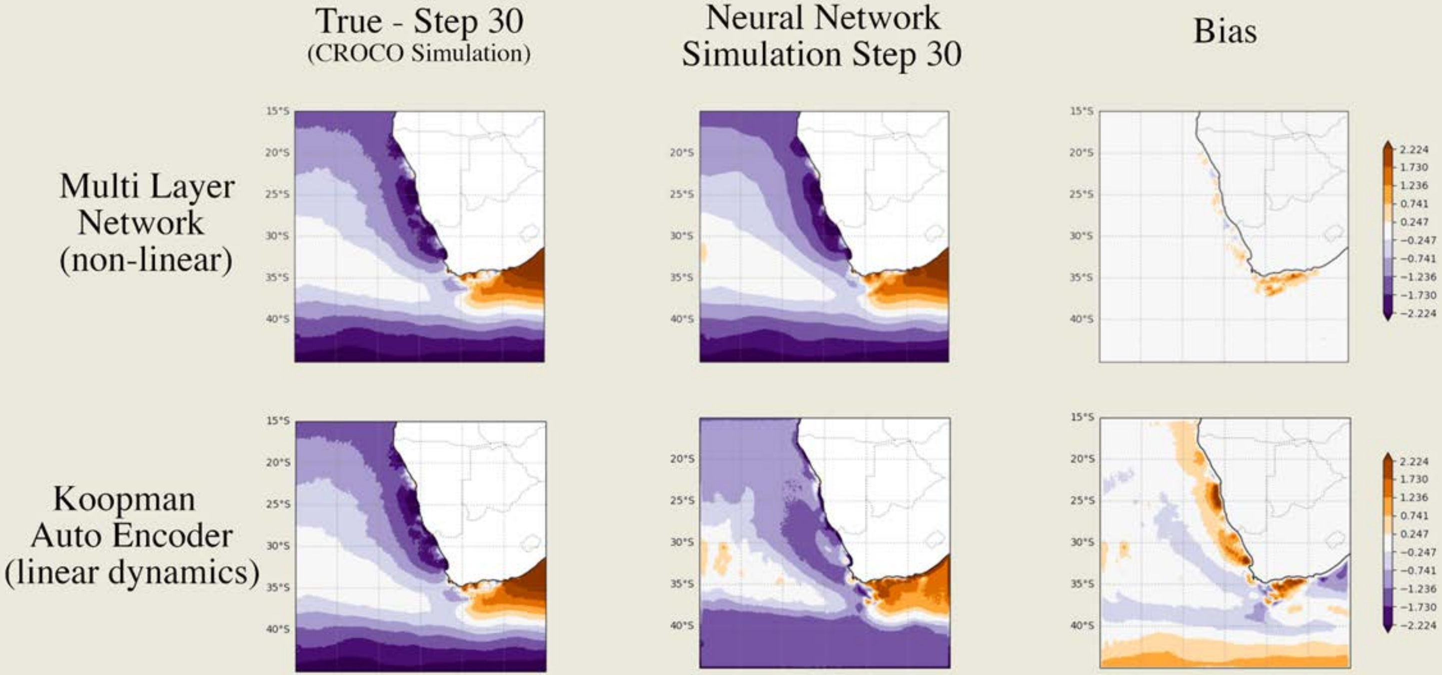


Results

Is the dimension of the linear operator too small to linearize the dynamics?

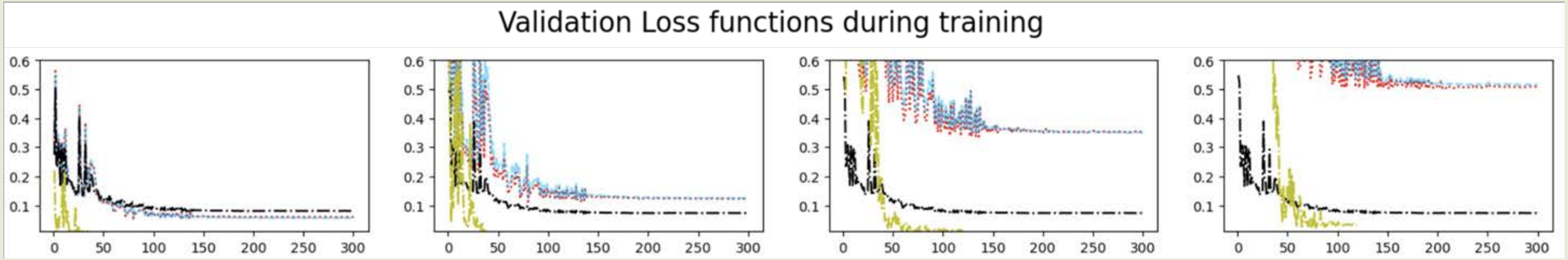
Results

The results evolving the system forward in time using a simple non-linear Multilayer neural network (three layers of 200 neurons) are better.



Results

Loss functions per epoch of training including Multilayer neural network in yellow. The non-linear network has an error one order of magnitude smaller than the linear network. This suggests that adjusting the Koopman dimension of the linear operator could significantly improve its predictions.



Conclusion

It appears feasible to integrate the barotropic dynamics of the regional oceanic model with a simple non-linear multi-layer network.

Next steps:

- Improve the physics informed Neural Network (Koopman AutoEncoder) to capture the barotropic dynamics.
- Train a network to integrate the baroclinic dynamics of the regional model.
- Fine tuning the model using observational/reanalysis data to create more realistic simulations and study eddy formation zones.

Thank you for your attention