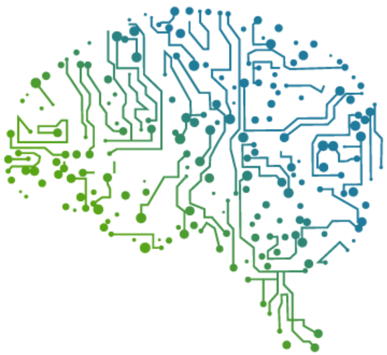
Boosting performance in Machine Learning of Turbulent and Geophysical Flows via scale separation

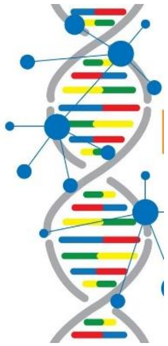
Davide Faranda, Mathieu Vrac, Pascal Yiou, Flavio Maria Emanuele Pons, Adnane Hamid, Giulia Carella, Cedric Gacial Ngoungue Langue, Soulivanh Thao, and Valerie Gautard

EGU2020-7569

MACHINE LEARNING IN SCIENCE



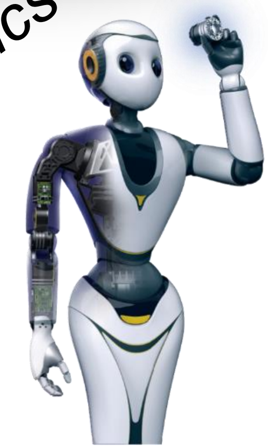
Neurosciences



Genomics

Traffic

Robotics



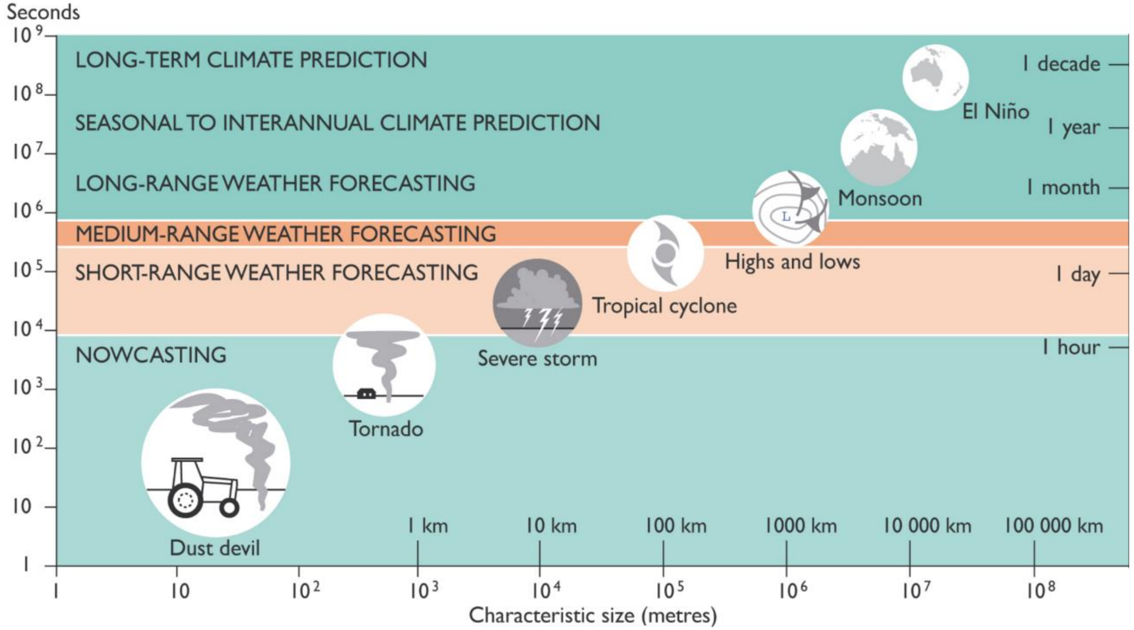
-**Complex** Systems

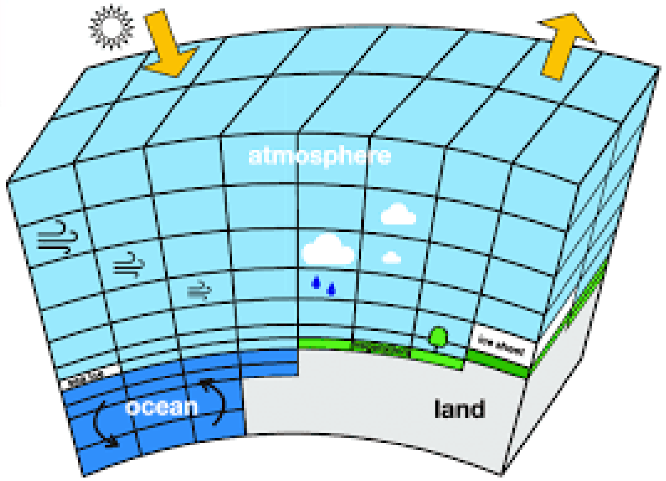
-Multiple **Spatial and time Scales**

-Large **Availability of Training Data**

-Missing Equations of State

MACHINE LEARNING IN CLIMATE SCIENCE





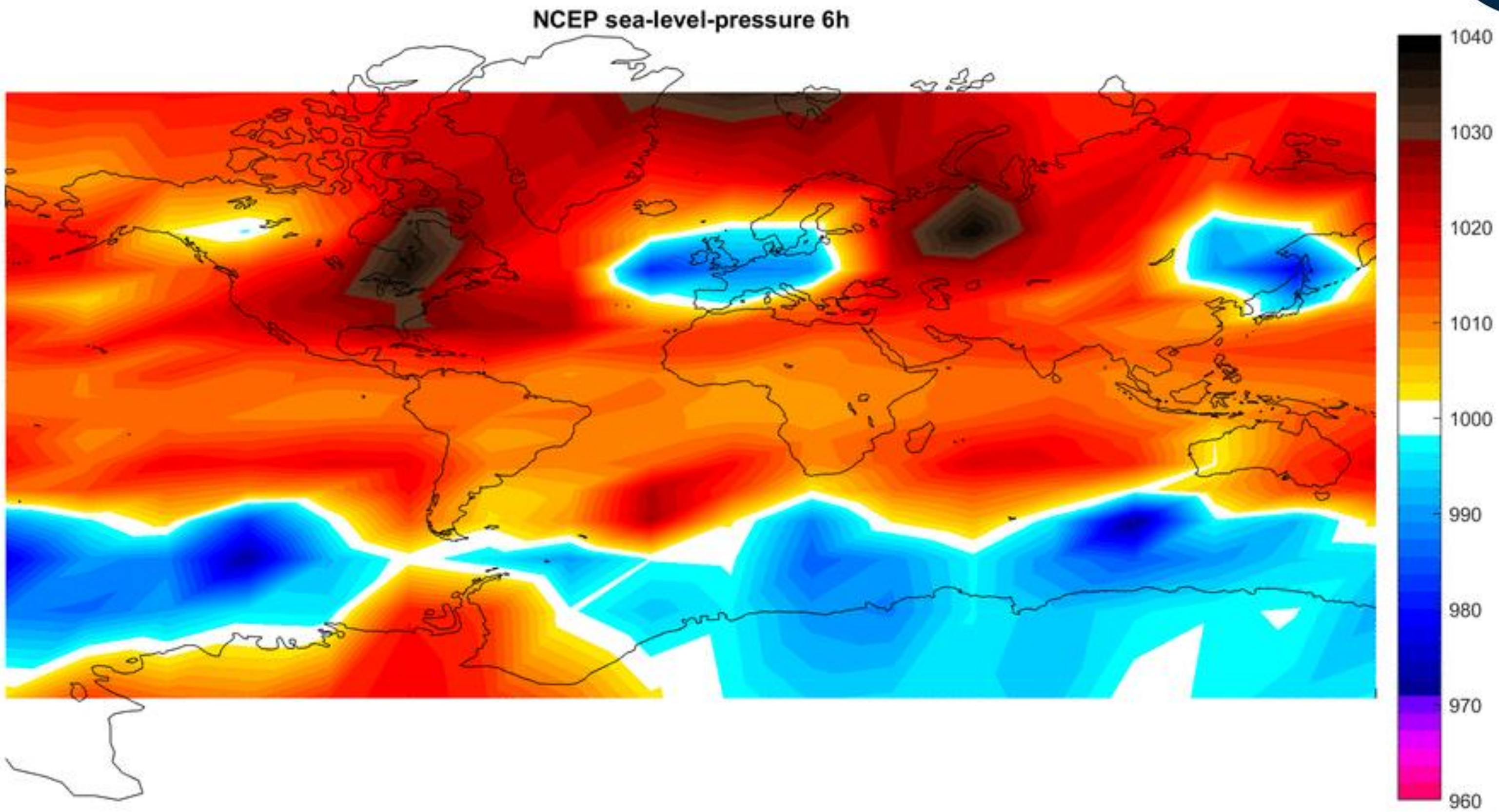
-**Complex** Systems

-Multiple **Spatial and time Scales**

-Large **Availability of Training Data**

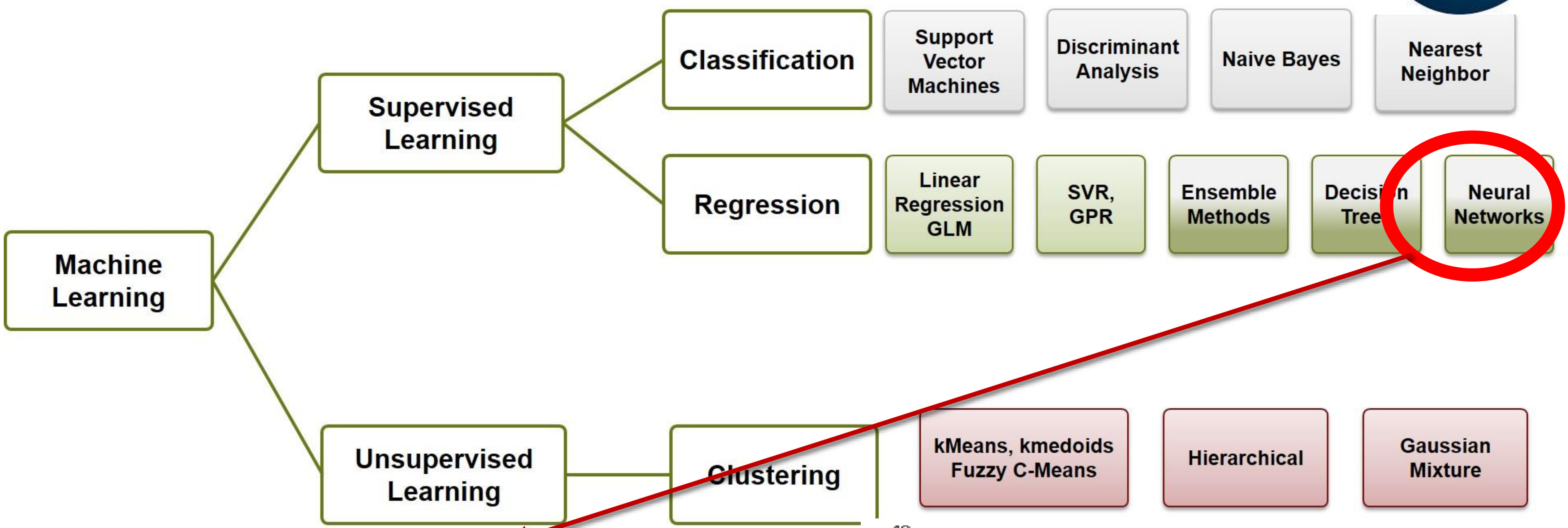
~~-Missing Equations of State~~ **(we have Navier-Stokes eqs.)**

WHICH SCIENTIFIC PROBLEM?



Task: forecast and generate a sea-level pressure forecast and its long term statistics to mimic that of the NCEP reanalysis.

WHICH TECHNIQUE?

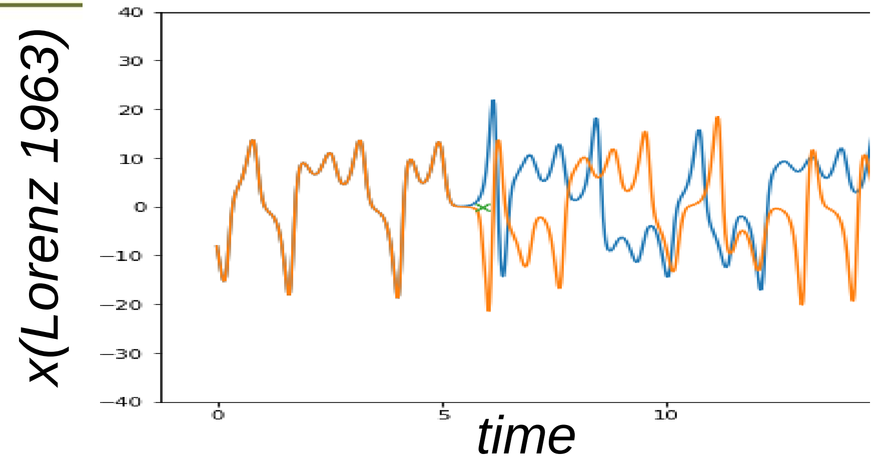


Pathak et al.– Phys. Rev. Lett. 2018

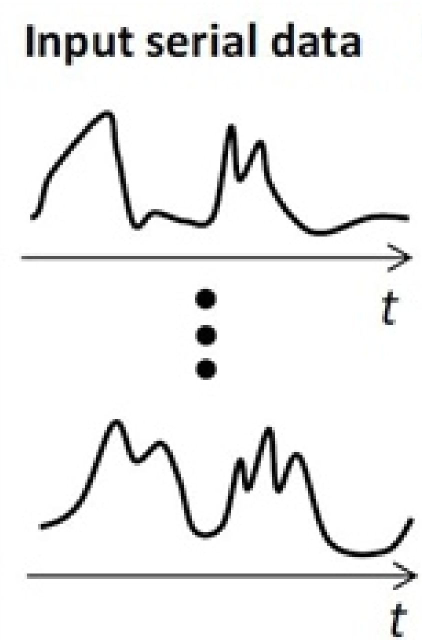
-Echo State Network for chaotic Systems

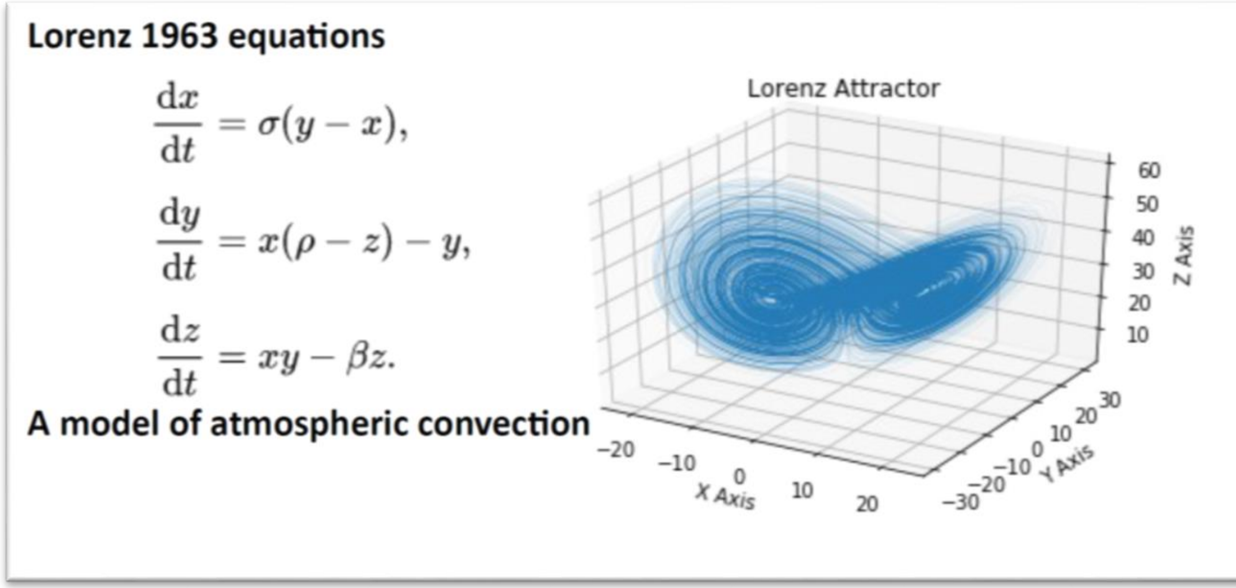
-Forecasts beyond the Lyapunov time!

- *Equations* *VS* *machine learning*



ECHO STATE NETWORKS + RECURRENCE



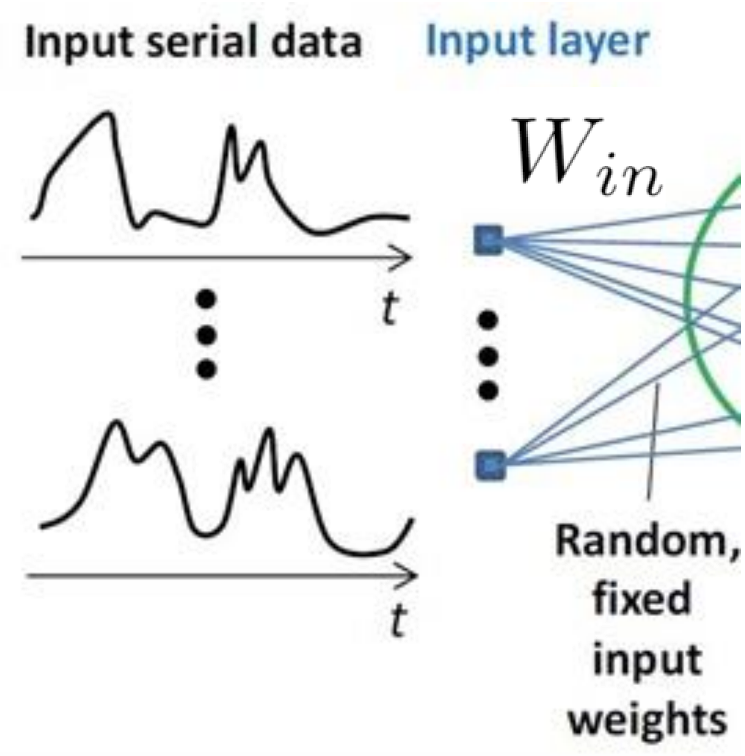


X(t) is a L dimensional vector

-Variables need to be standardized

$$x(t) \xrightarrow{\text{ECHO STATE NETWORK}} x(t+dt)$$

ECHO STATE NETWORKS + RECURRENCE



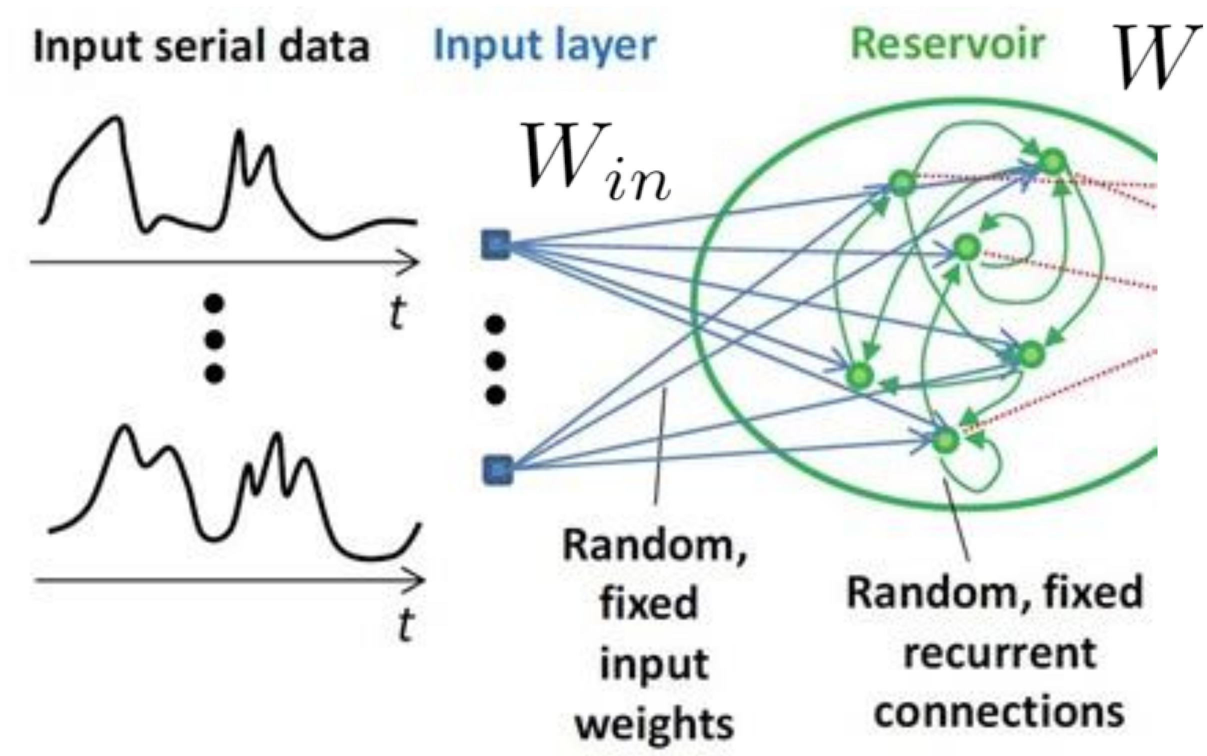
W_{in} is a matrix LxN

-*L* is the number of variables.
-*N* is the network size

Each lines consists of random weights uniform in [0.5 0.5]

$x(t)$ —ECHO STATE NETWORK→ $x(t+dt)$

ECHO STATE NETWORKS + RECURRENCE



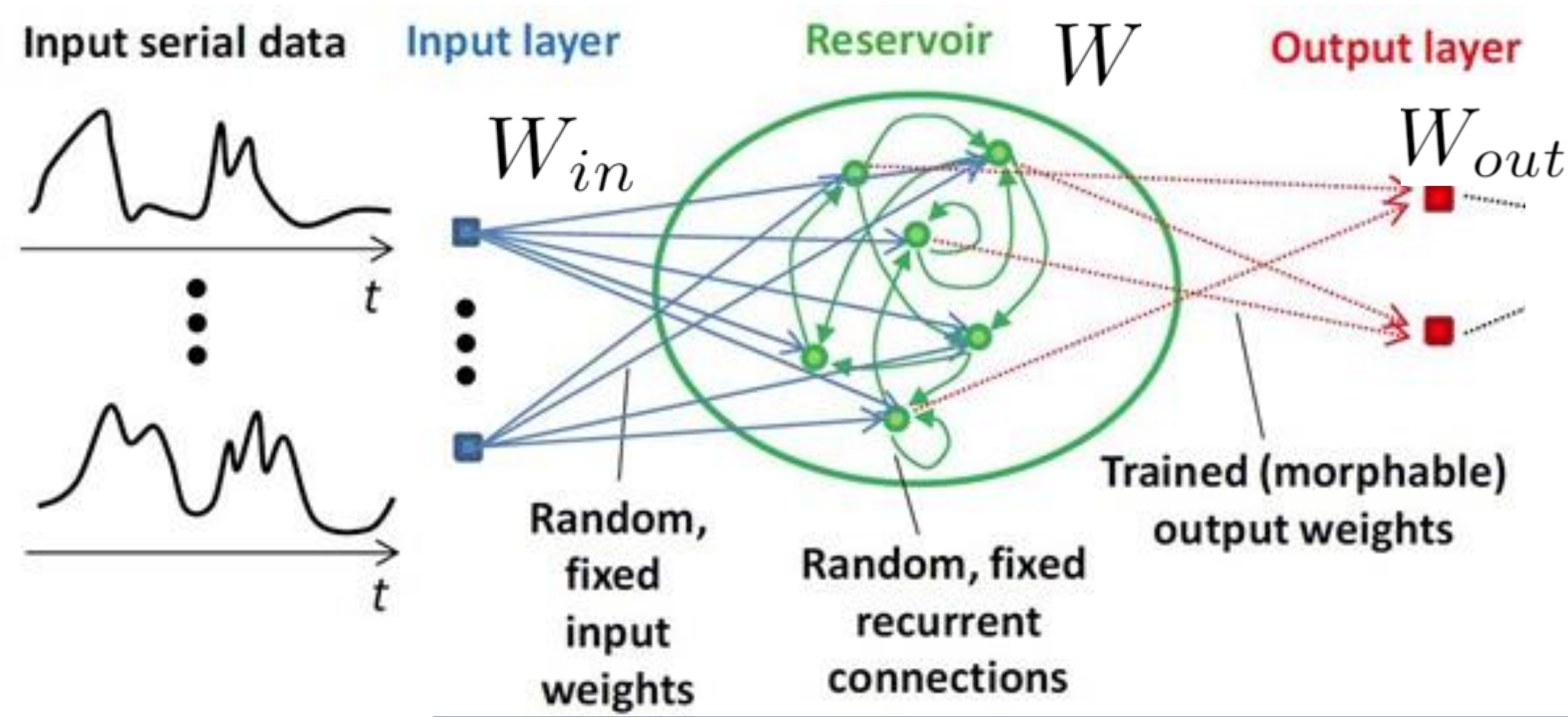
W is NxN matrix

-*N* is the network size
-Activation function is tanh

Each lines consists of random weights uniform in [0.5 0.5]

$$x(t) \xrightarrow{\text{ECHO STATE NETWORK}} x(t+dt)$$

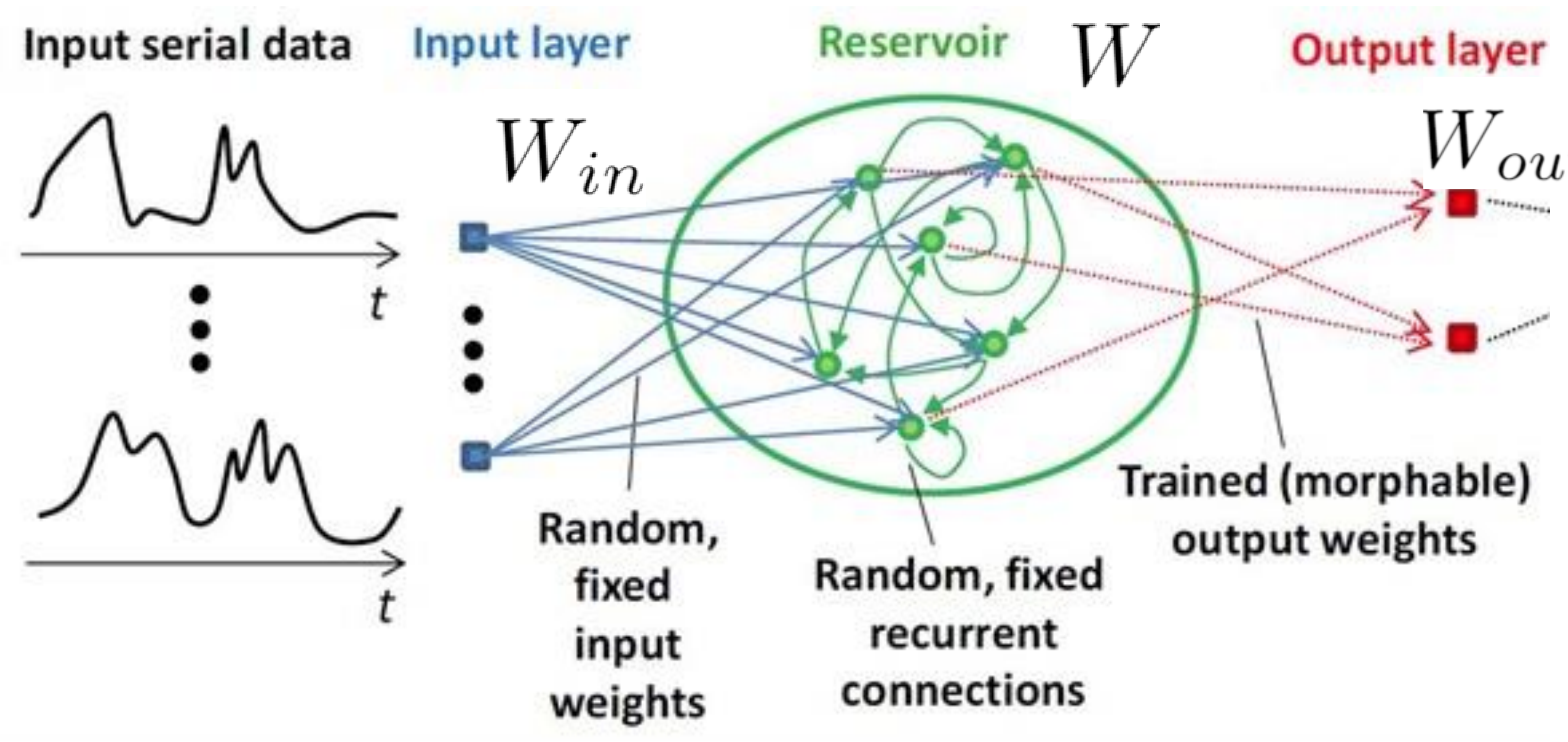
ECHO STATE NETWORKS + RECURRENCE



W_{out} is a matrix NxL

-Optimized during the training with a ***Ridge regression*** *so that the output matches x(t+dt)*

ECHO STATE NETWORKS + RECURRENCE

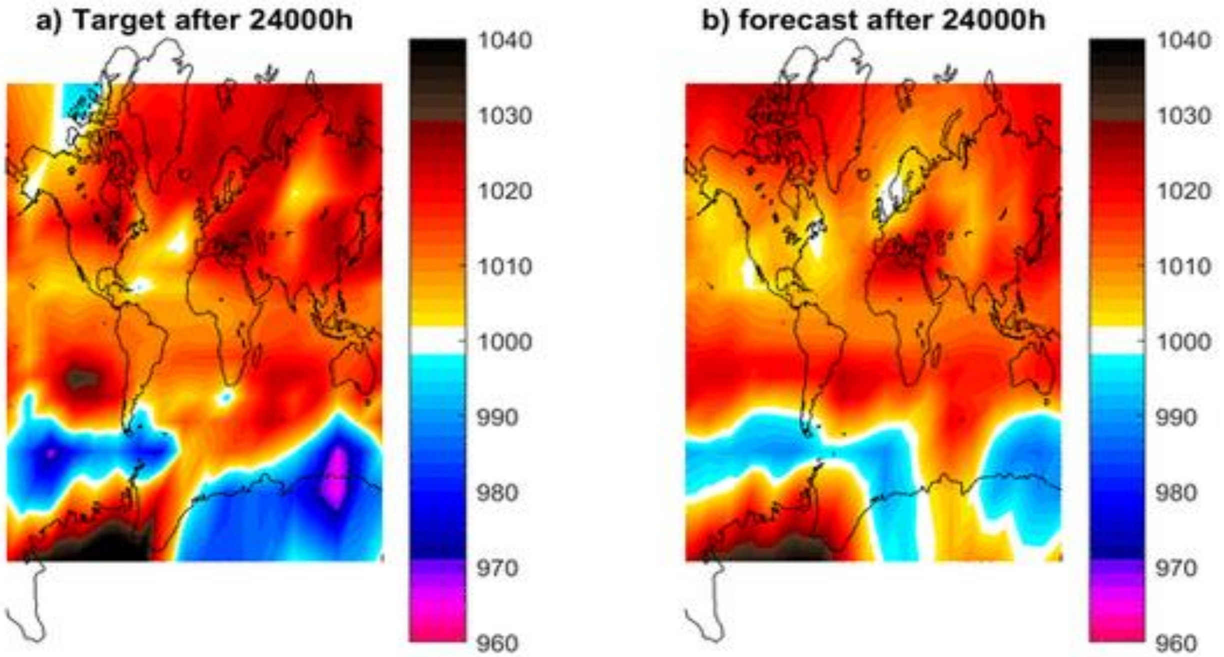


$$x(t+dt) = \tanh(Wx(t) + W_{in}W_{out}x(t))$$

FIRST TRIALS ON SEA-LEVEL PRESSURE

Network Size= 200 Neurons, Learning Time = 10 years Forecast Length = 10 years

At long time, the dynamics is stuck, it does not look realistic anymore (independently on the chosen parameters)



Similar results: Scher & Messori (2018,2019), Dueben & Bauer (2018)

=> We need to take one step back to assess what is wrong

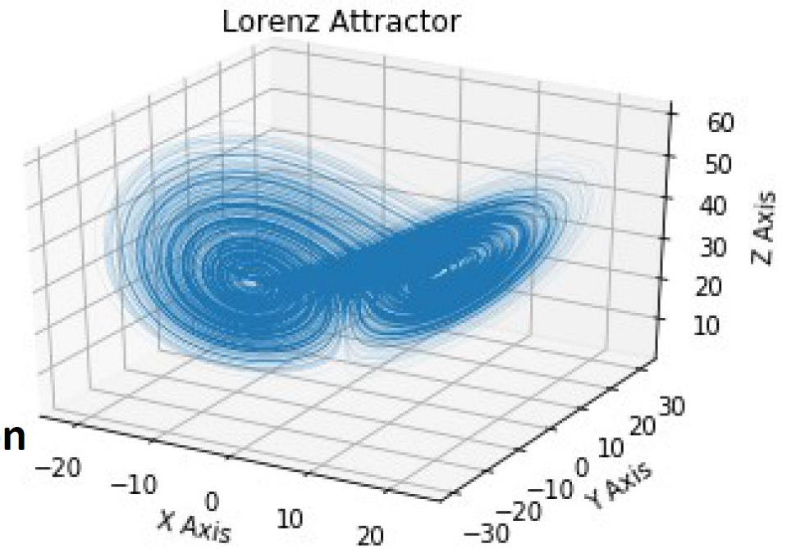
TEST SYSTEMS

Lorenz 1963 equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

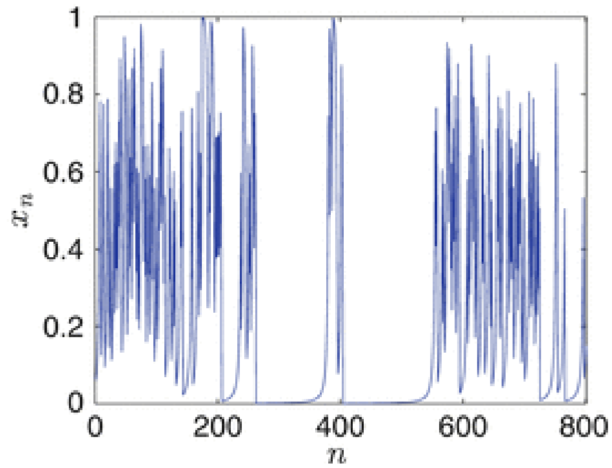
$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$



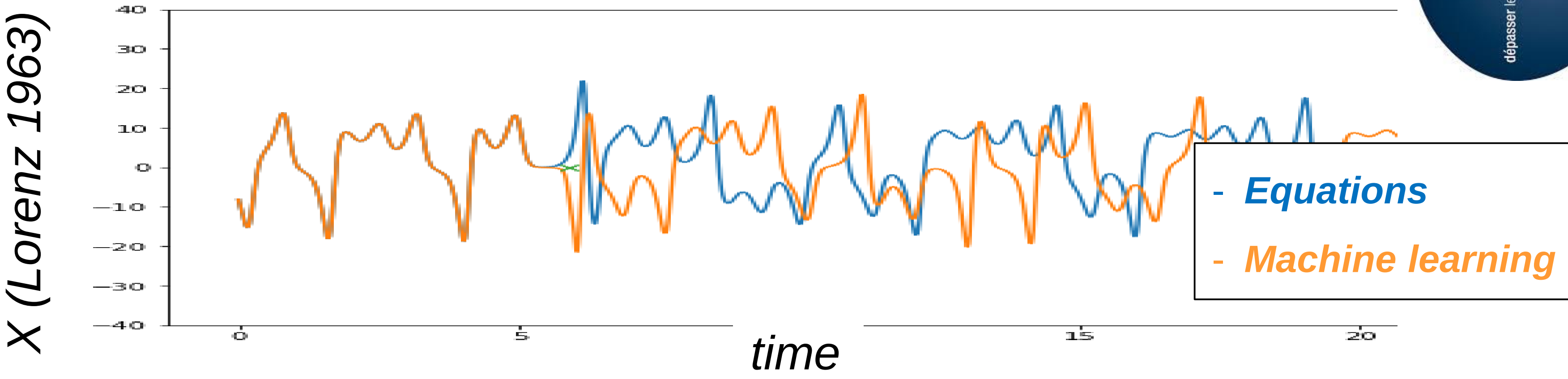
A model of atmospheric convection

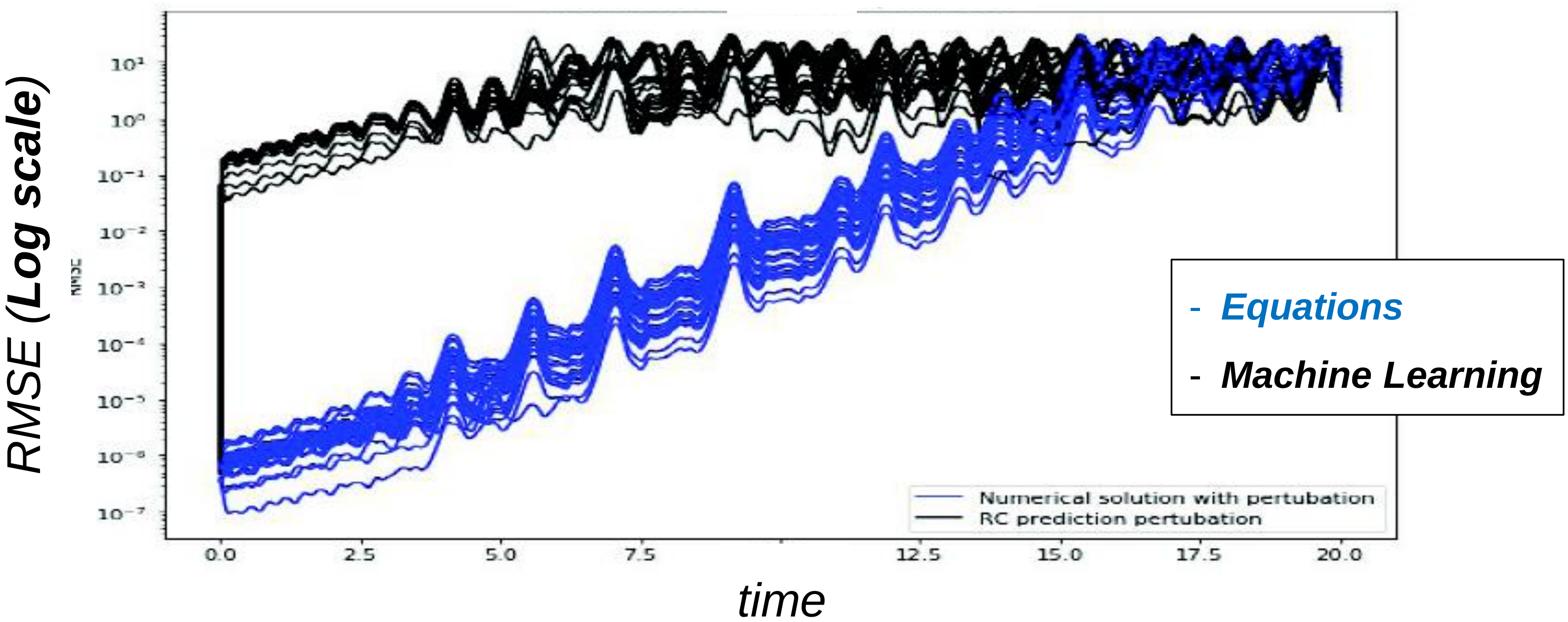
Pomeau Manneville intermittent map

$$x_n = x_{n-1}(1 + 2^{\beta} \mathrm{x}_{n-1}) \quad \text{if } x_n < .5$$
$$x_n = 2x_{n-1} - 1 \quad \text{if } x_n > .5$$

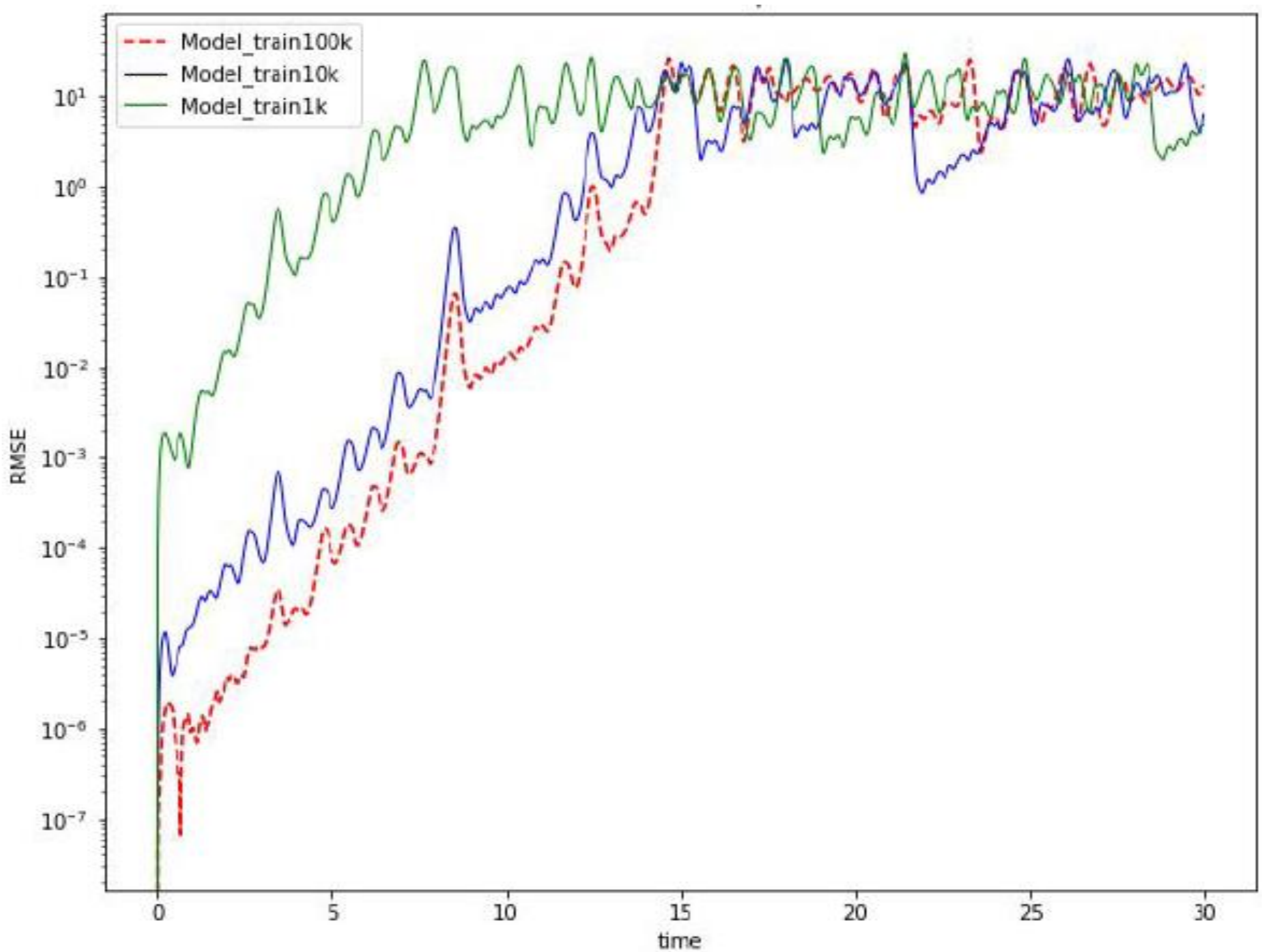


DANGER #1: LEARNING TIME





DANGER #1: LEARNING TIME

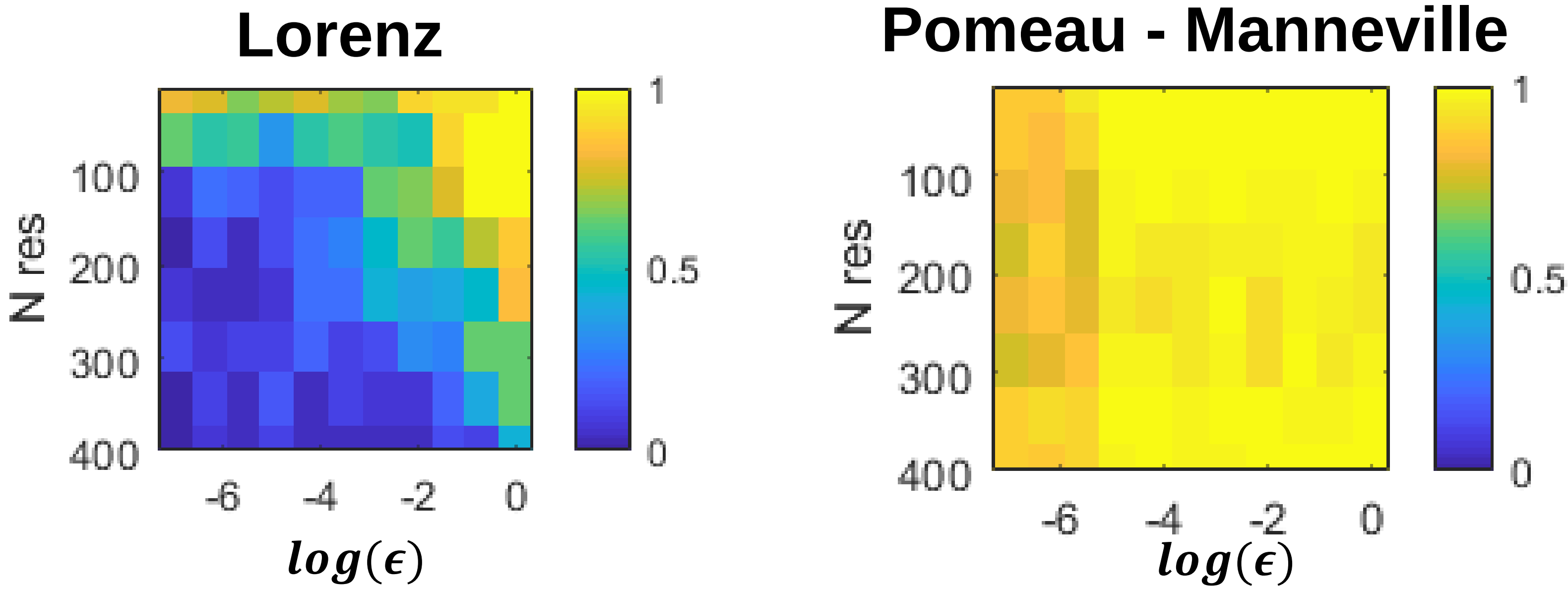


DANGER #2 NOISE & INTERMITTENCY

Additive noise to the Lorenz 1963 equations & Pomeau-Manneville Intermittent map:

$$x(t+dt) = f\big(x(t)\big) + \epsilon\xi(t)$$

where $\xi(t)$ is a random variable uniform in [-0.5 0.5]



Percentage of failure in reproducing the attractor

(0 means never fail,   1 means always fail)

POSSIBLE SOLUTION: SCALE SEPARATION

1) **Filter the noise**

There are countless methods, but we use the simplest possible one:

Moving Average filter *with window size:*

$$ws \ll \tau$$

where τ is the Lyapunov time

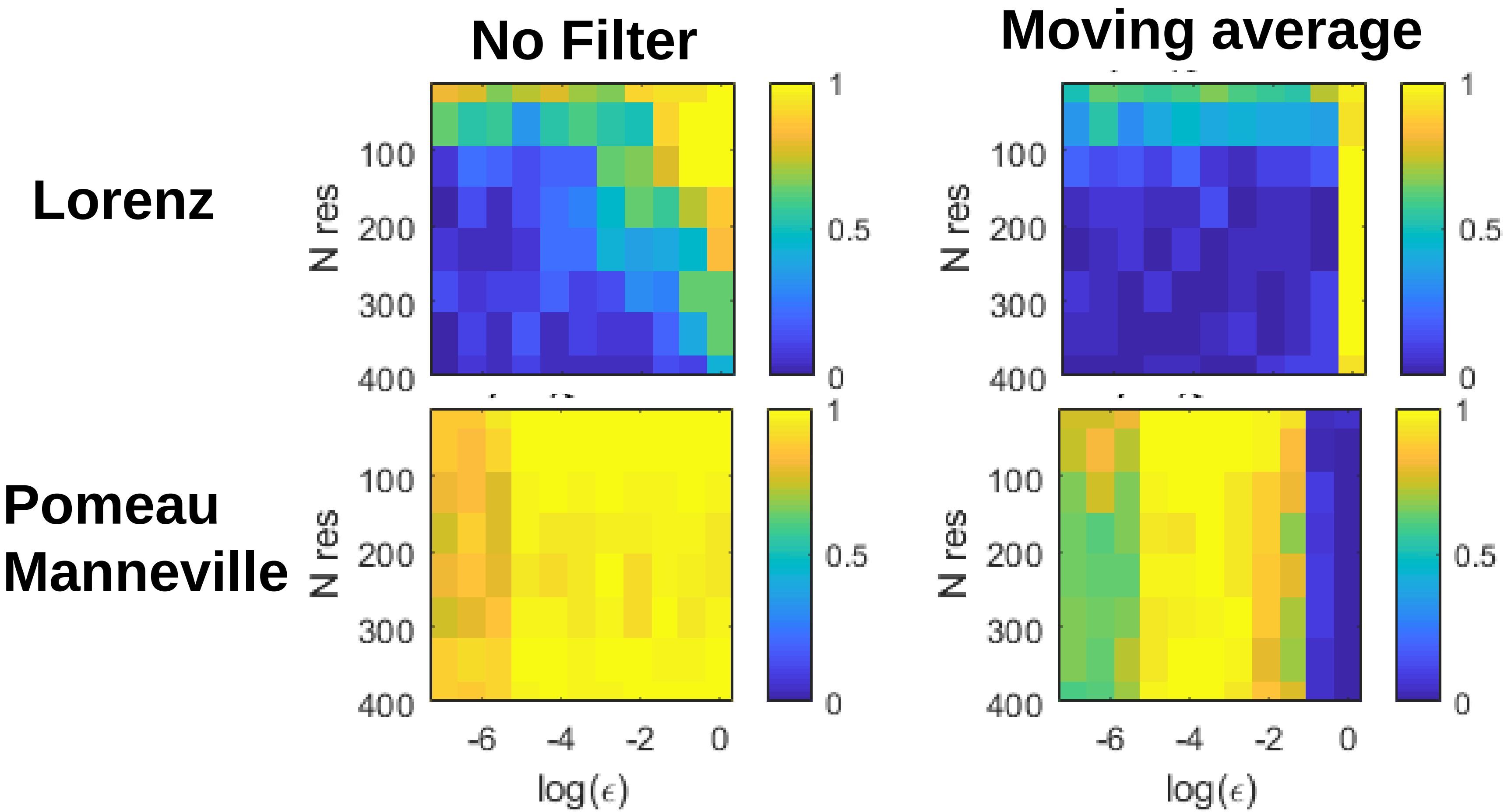
2) **Apply Echo State Network to the filtered system only**

3) **Add back the residual to the forecast**

 davide.faranda@cea.fr

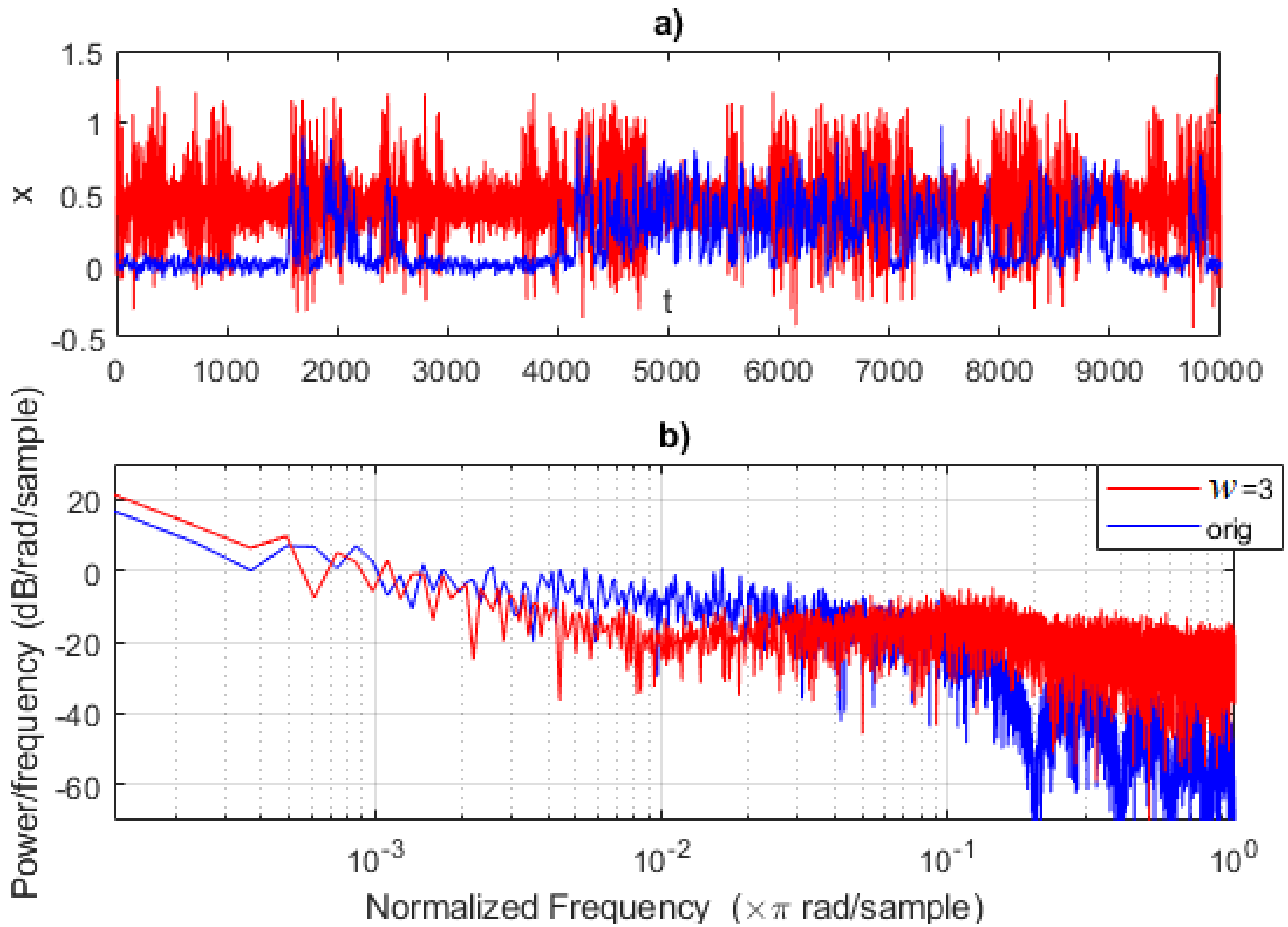
IMPROVEMENTS FOR LOW D SYSTEMS

Percentage of failure in reproducing the attractor
(0 means never fail,   1 means always fail)



IMPROVEMENTS FOR LOW D SYSTEMS

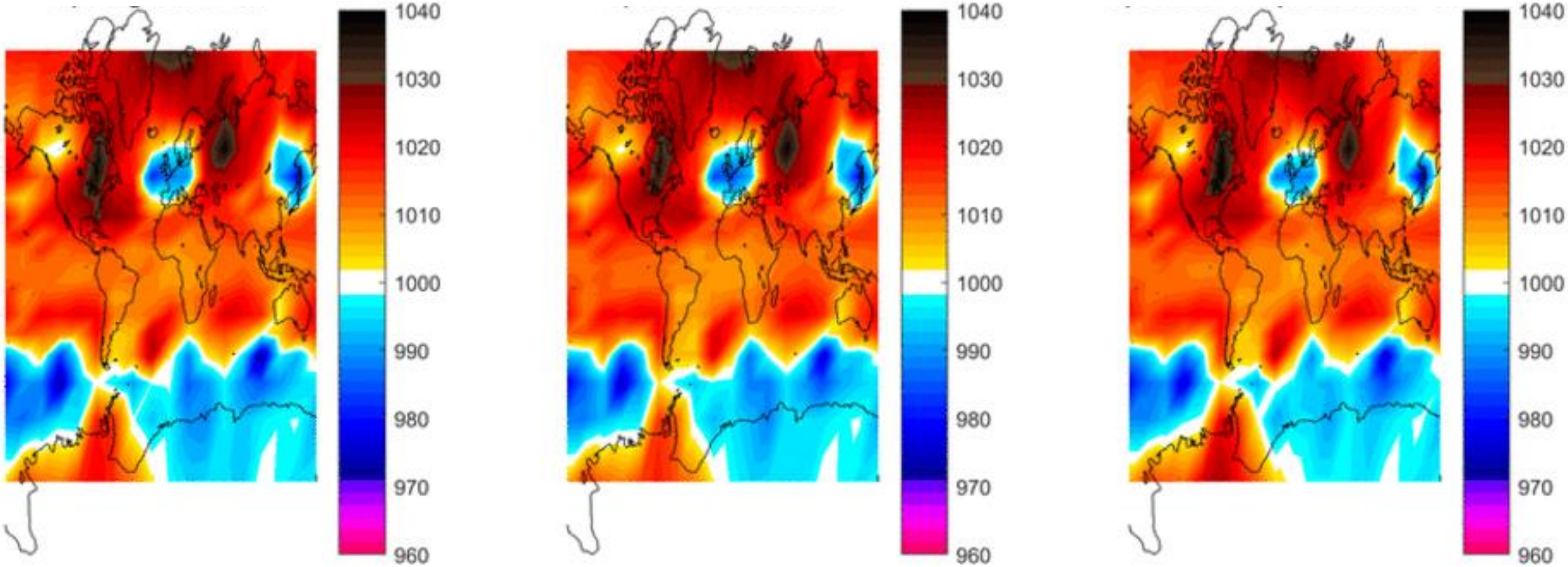
Pomeau Manneville



TEST ON NCEP SEA-LEVEL PRESSURE

Network Size= 200 Neurons, Learning Time = 10 years Forecast Length = 10 years

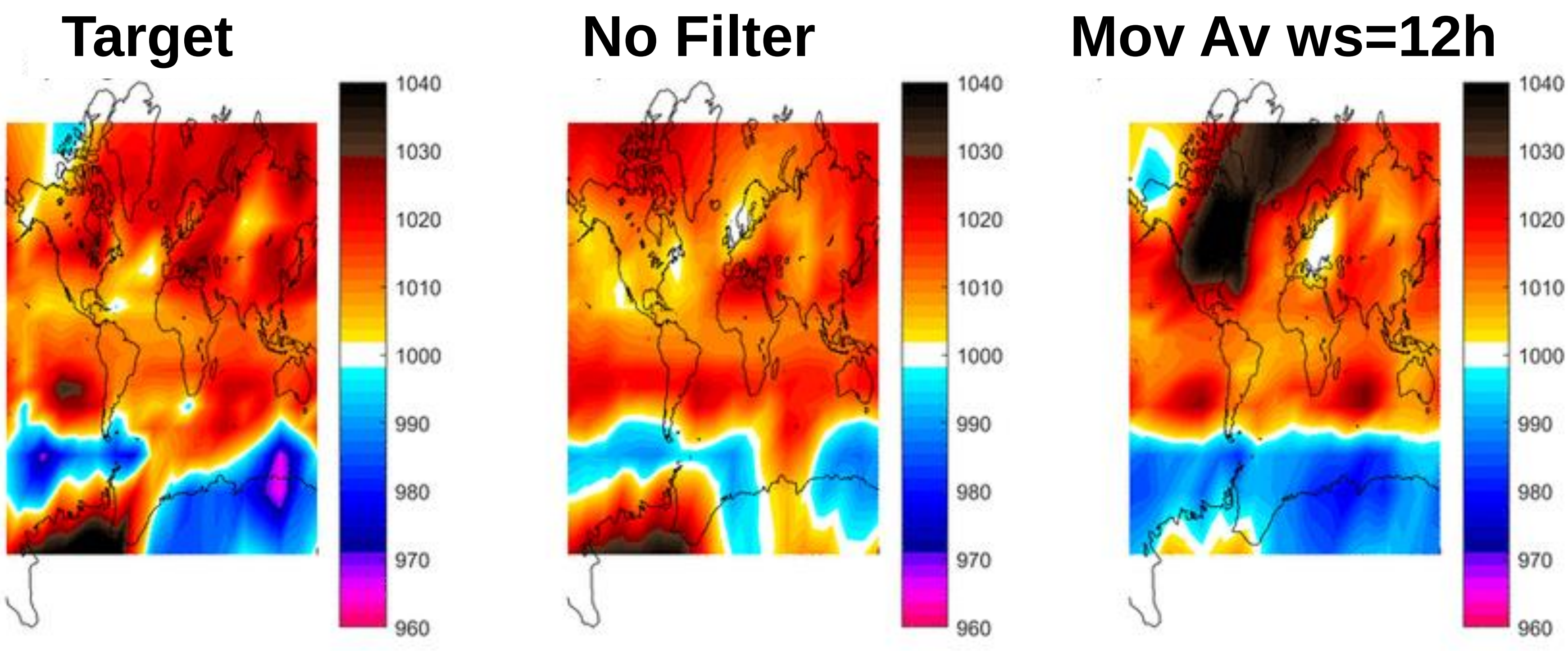
Target | **No Filter** | **Mov Av ws=12h**



*For the **short term forecast**, there is no much improvement*

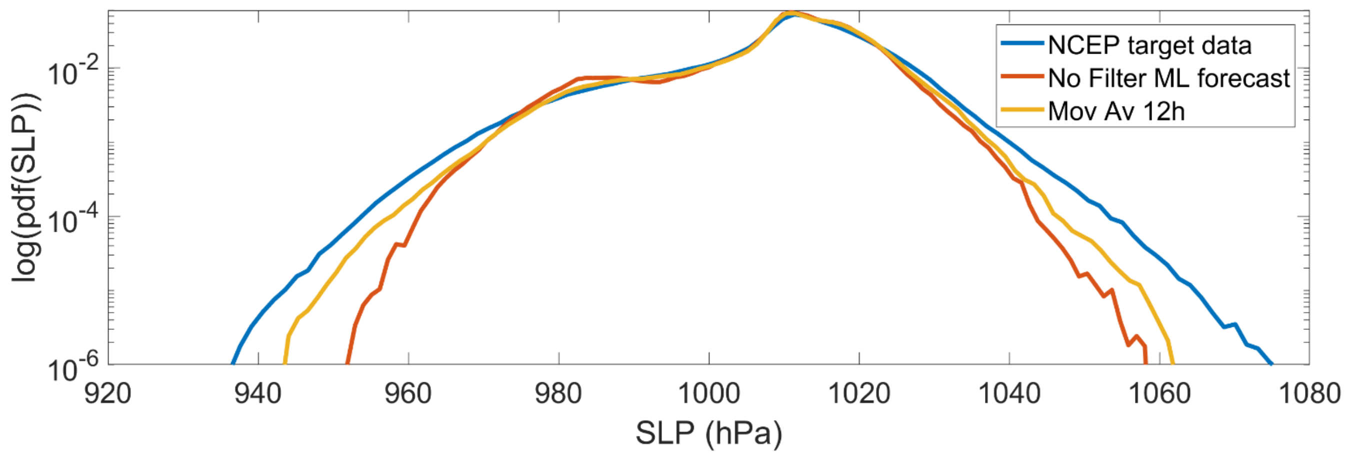
TEST ON NCEP SEA-LEVEL PRESSURE

Network Size= 200 Neurons, Learning Time = 10 years Forecast Length = 10 years

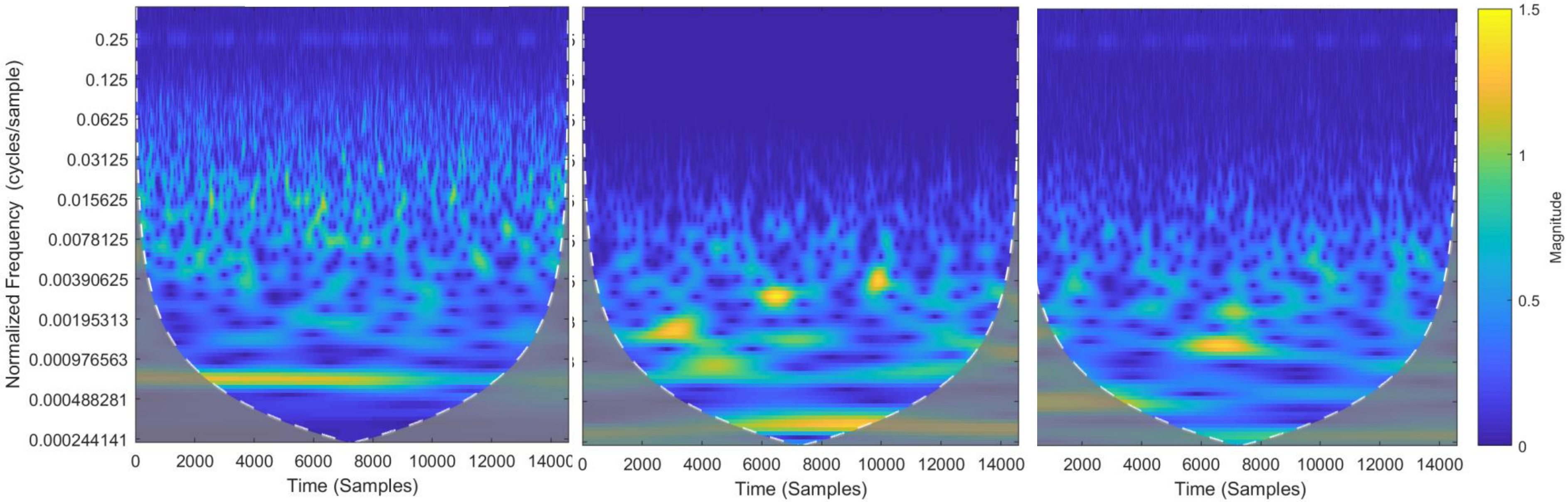


*If we look at the **long term behavior**, it is evident that the simulation with moving average is more realistic*

SPACE TIME STATISTICS



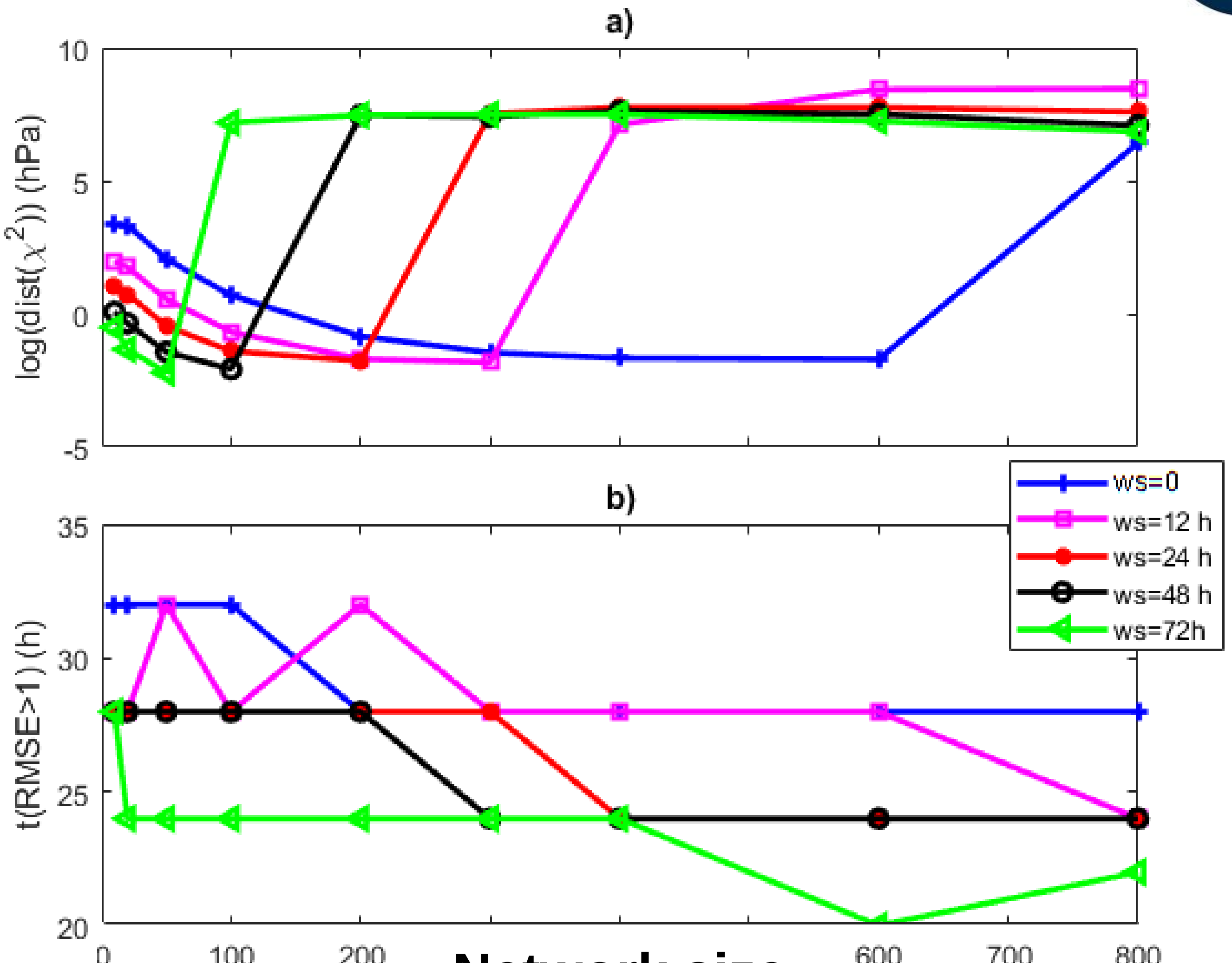
Target **No Filter** **Mov Av ws=12h**



A MORE QUANTATIVE ASSESSMENT

Distance from the NCEP data

Predictability horizon (in hours)



Network size

CONCLUSIONS

1) **It is not straightforward to apply Machine Learning techniques to geophysical flows: turbulence and intermittency worsen the performance**

2) **Partial predictability can be recovered by separating large from small scale dynamics (e.g moving average, PCA, wavelets)**

3) **Possible developments will largely benefit from interactions with the stochastic dynamical systems community**

 davide.faranda@cea.fr

REFERENCES

[1] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Physical review letters 120, 024102 (2018)

[2] S. Scher and G. Messori, Weather and climate forecasting with neural networks: using general circulation models (gcms) with different complexity as a study ground, Geoscientific Model Development 12, 2797 (2019)

[3] D. **Faranda**, M. Vrac, P. Yiou, F.M.E. Pons, A. Hamid, , G. Carella, C.G. Ngoungue Langue, S. Thao, V Gautard. Boosting performance in Machine Learning of Turbulent and Geophysical Flows via scale separation. Phys Rev Letters (in review) (2019)

Contact: davide.faranda@cea.fr

Thank You for the Attention