

Bridging Physics and Learning: application to ocean dynamics

R. Fablet et al.

ronan.fablet@imt-atlantique.fr

web: rfablet.github.io

Webinar IMT Data & AI, July 2020



AI & Ocean Science: General context

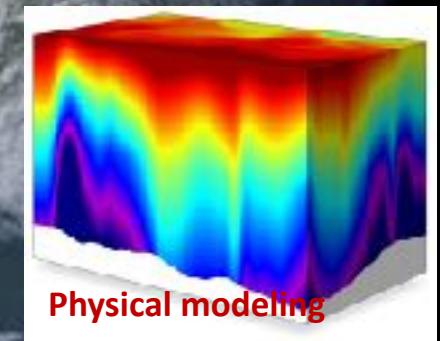
Bridging physics paradigms and deep learning

Focus on DL & Inverse Problems

Some illustrations

General question

How to solve sampling gaps and extract high-level information for ocean monitoring and surveillance ?



Context: No observation / simulation system to resolve all scales and processes simultaneously

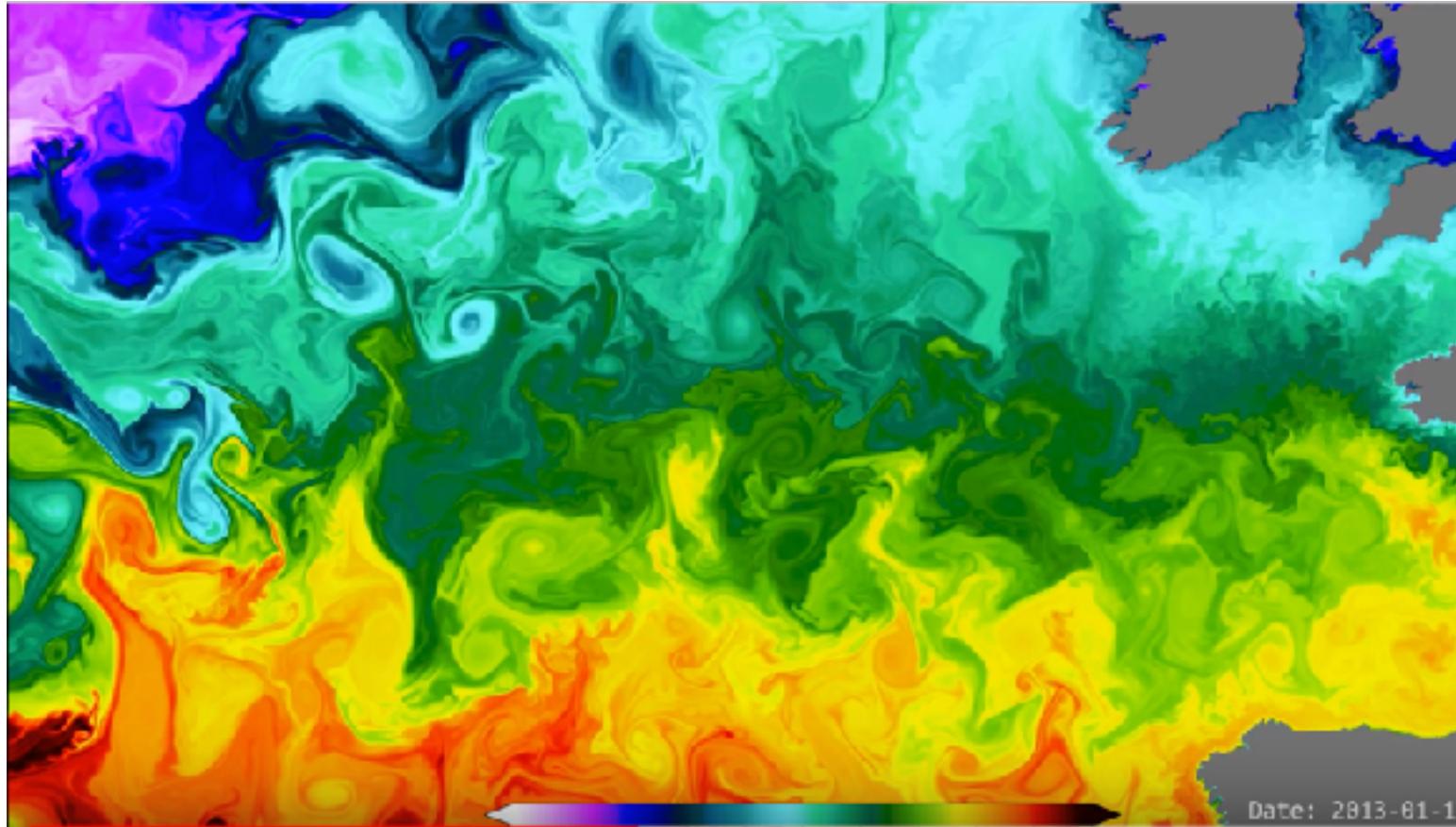
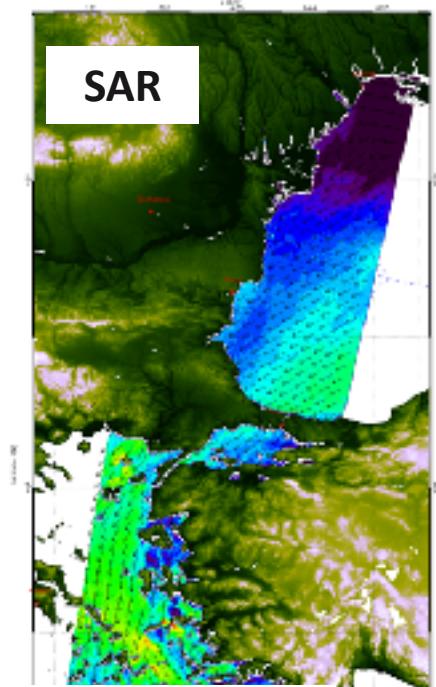
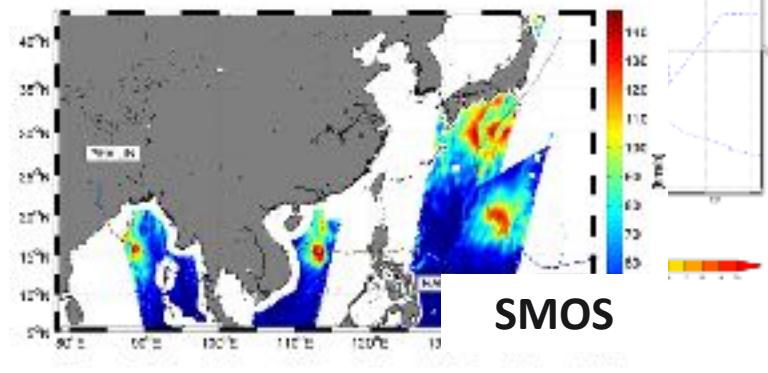
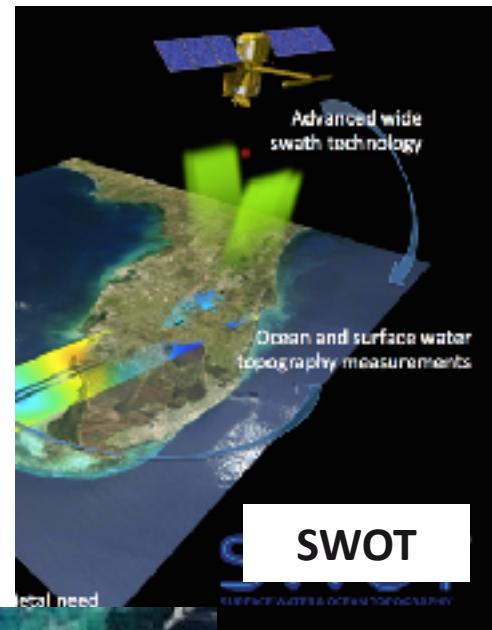
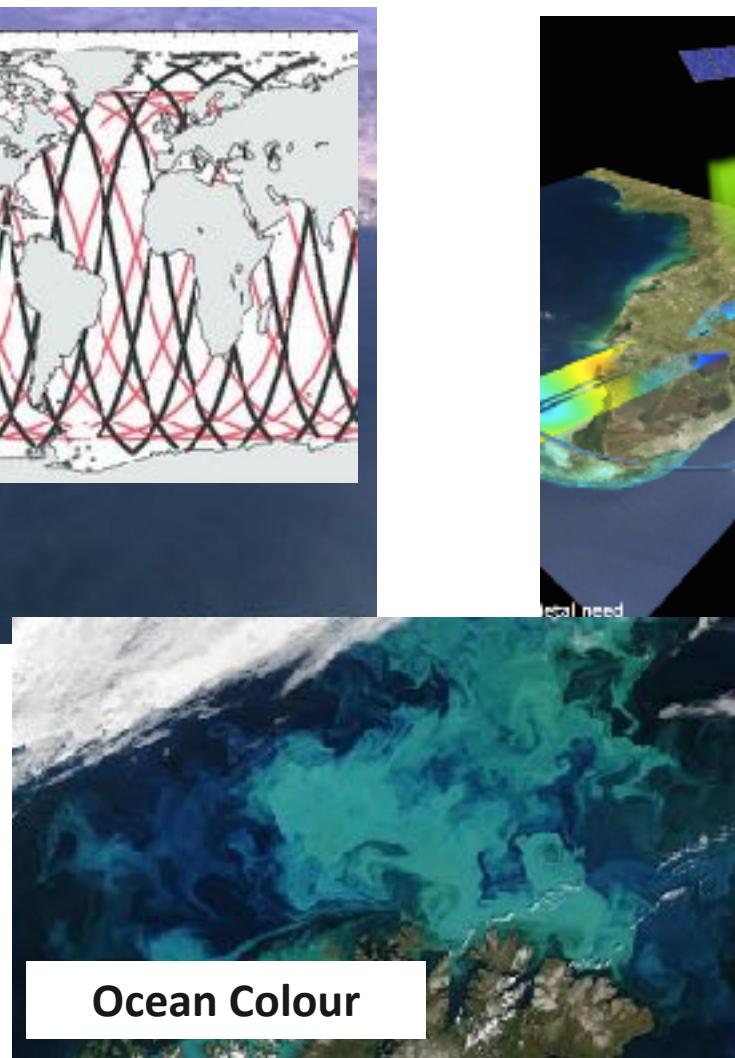
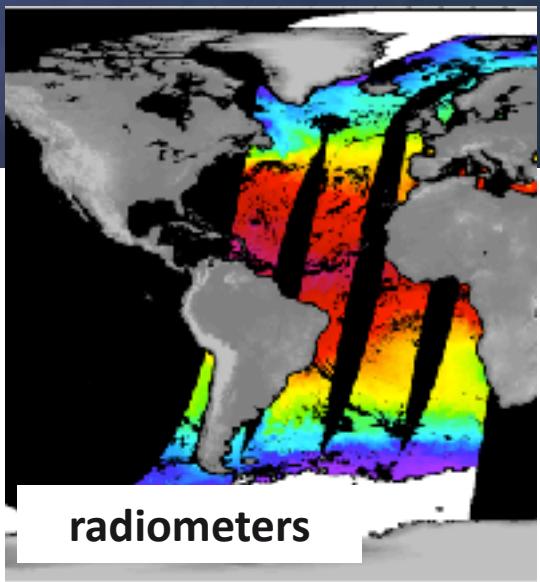
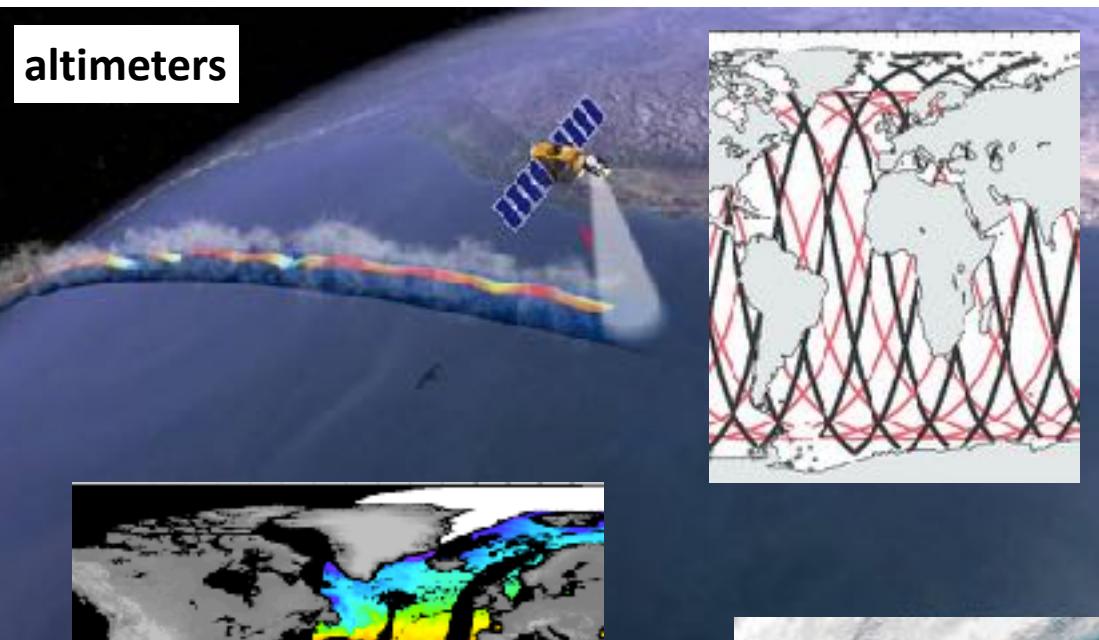
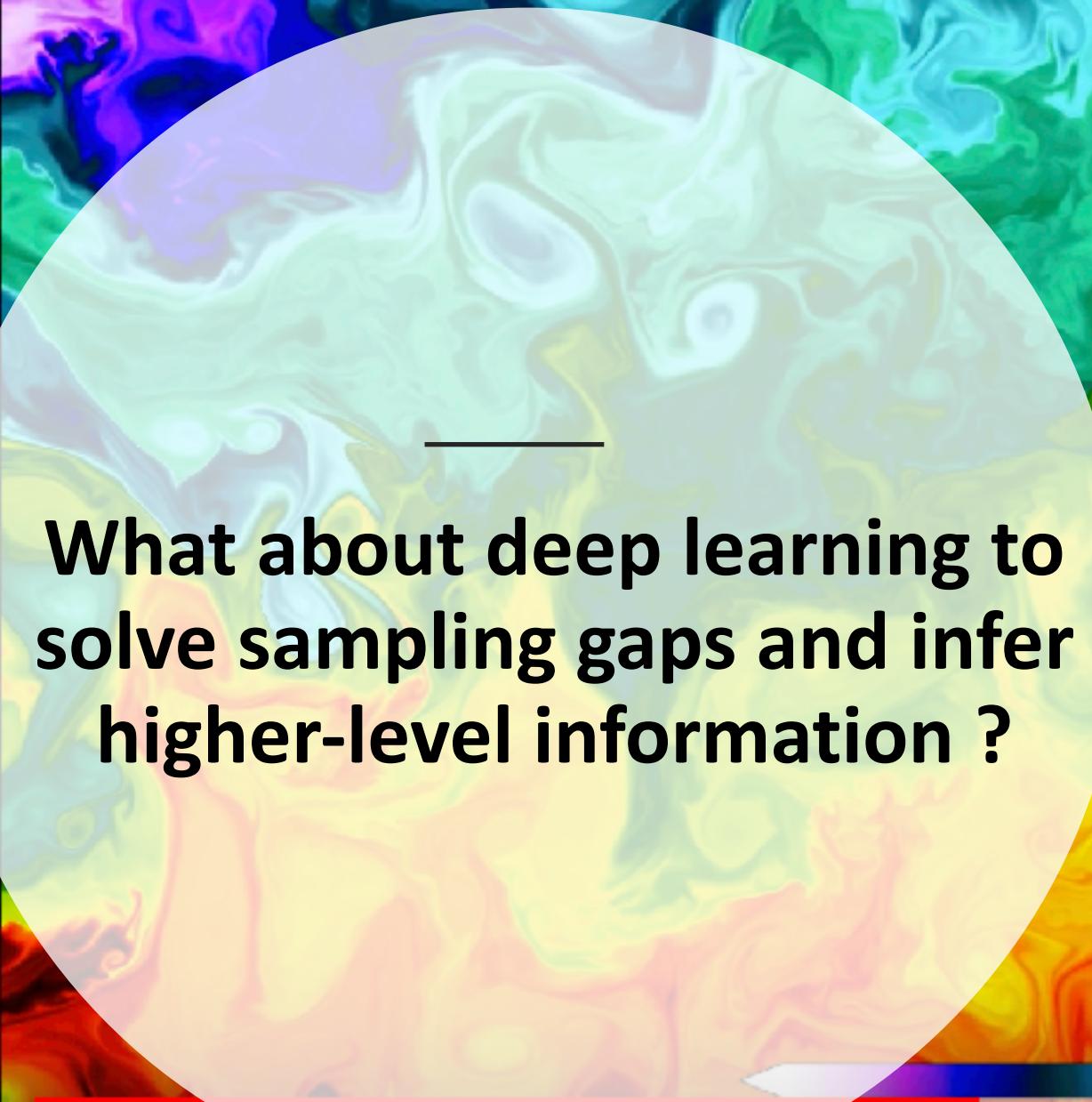


Illustration of satellite-derived sea surface observations

28-JUL-2014 08:30:00 (GLT (UT)) CLD

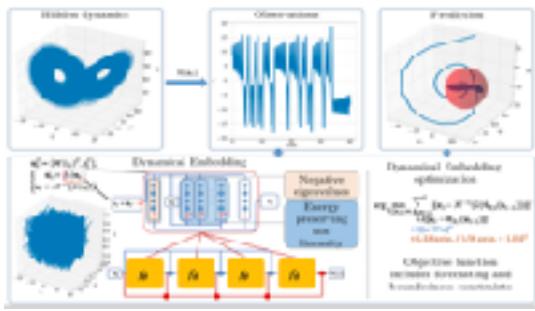




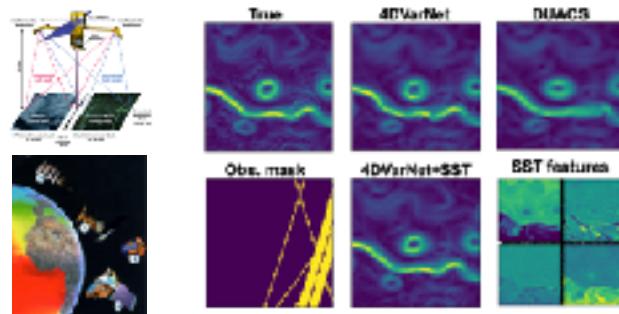
What about deep learning to
solve sampling gaps and infer
higher-level information ?

Topics of interest with emphasis of DL approaches

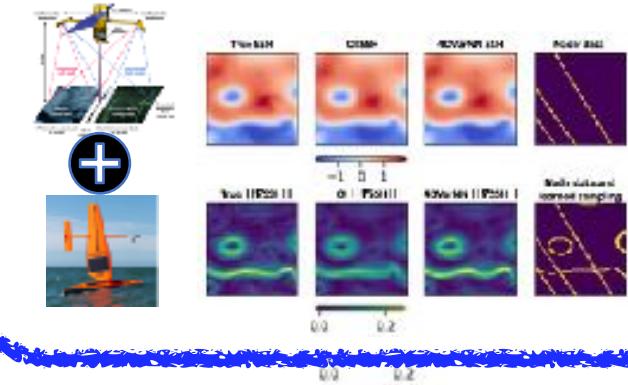
Observation-driven forecasting



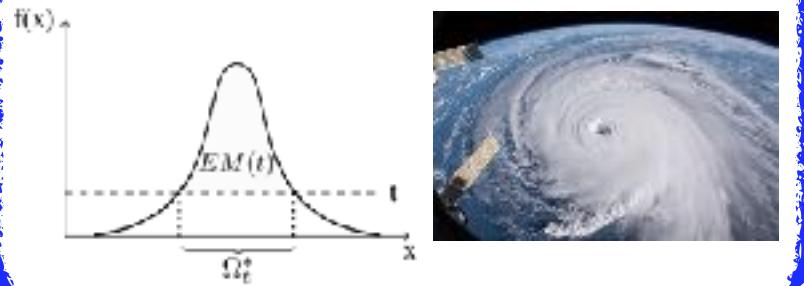
Multimodal reconstruction



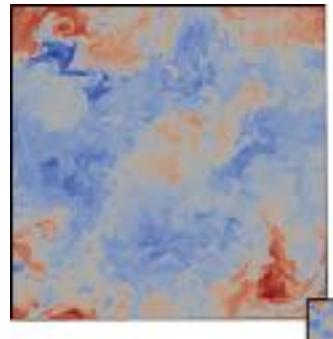
Learning where to sample ?



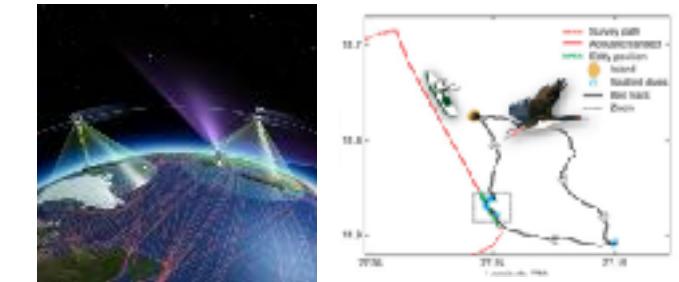
Predicting extremes



Ocean modeling



Trajectory data analysis and modeling



Learning & Geoscience: an old story ?

Empirical Orthogonal Functions and Statistical Weather Prediction

By
EDWARD N. LORENZ

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF METEOROLOGY
Cambridge, Massachusetts

DECEMBER 1956

Scientific Report No. 1
STATISTICAL FORECASTING PROJECT

EDWARD N. LORENZ

EOF/PCA

THE RESEARCH FUNDING FOR THIS DOCUMENT WAS PROVIDED BY THE UNIDENTIFIED RESEARCH DIRECTORATE OF THE AIR FORCE CAMPAIGN FOR SEARCH, SYSTEMS, RESEARCH AND DEVELOPMENT COMMAND, UNDER CONTRACT NO. AF33(657)-56.

Deterministic Nonperiodic Flow*

EDWARD N. LORENZ

Massachusetts Institute of Technology

(Manuscript received 10 December 1961, in revised form 1 January 1962)

ABSTRACT

Finite systems of deterministic ordinary nonlinear differential equations may be designed to represent known chaotic hydrodynamic flows. Solutions of these equations can be identified with trajectories in phase space. For those systems with bounded solutions, it is found that nonperiodic solutions are uniformly unstable with respect to small modifications, so that slightly differing initial states can evolve into considerably different states. Systems with bounded solutions have the property that finite numerical solutions A single system representing regular convection is solved numerically. All of the solutions are found to be unstable, and almost all of them are nonperiodic.

The feasibility of very-long-range weather prediction is discussed in the light of these results.

1. Introduction

Completely hydrodynamic systems exhibit steady-state flow patterns, while others oscillate in a regular periodic fashion. Still others may be irregular, seemingly haphazard motion, such as was observed for long periods of time, do not appear to repeat their previous history.

These random motions which cannot be predicted by familiar statistical methods are often called "chaotic." Lorenz et al. (1963) have shown that the motion of a fluid in a rotating rectangular container, with free boundaries, is chaotic, and is bounded within the container boundaries. In a similar numerical simulation, Ueda (1968) concluded that the resulting flow was asymmetric and steady in the heating with geyser-like. Under different conditions a system of regularity spaces were developed, and progressed at a uniform speed without changing its shape. Ueda still believes that an irregular flow is more complex and more interesting than a regular flow, and that it is more numerically interesting.

Lack of periodicity has been observed in some systems, and one of the chief difficulties in dealing with turbulent flow. Some instabilities resemble flow patterns so irregular, however, it often masking the steady-state flow, which is difficult to discern in the details of turbulence, often behave in a regular, well-organized manner. The shear waves, another writer, however, is faced with the problem of the nature of the flow, and turbulent flow, and the way in which they interact, which controls the development of the flow.

* This contribution is based on the author's research at the Massachusetts Institute of Technology, Cambridge, Massachusetts. This work was supported by the Air Force Research Directorate of the Air Force Campanile for Search, System, Research and Development Command, under contract no. AF33(657)-56.

Analogs / Nearest- neighbors

Then there are occasions when more than the statistics of incipient flow are of very real concern.

In this study we shall work with systems of deterministic equations which are imitations of hydrodynamical systems. We shall be interested principally in nonperiodic solutions, i.e., solutions which do not repeat their past history exactly, and where approximate repetitions are of finite duration. They will be represented by sets of points in solution, in which the points are usually associated with particular times.

Such systems of equations may be called "chaotic" if they exhibit the behavior of a finite collection of points—usually a very large finite collection—in which case the governing laws are expressible as a finite set of ordinary differential equations. These equations are generally highly interrelated, and the use of resonance is usually appropriate by a coordinate transformation. These are then expressed in terms of a finite set of variables, containing such variables as time, position, and pressure or density.

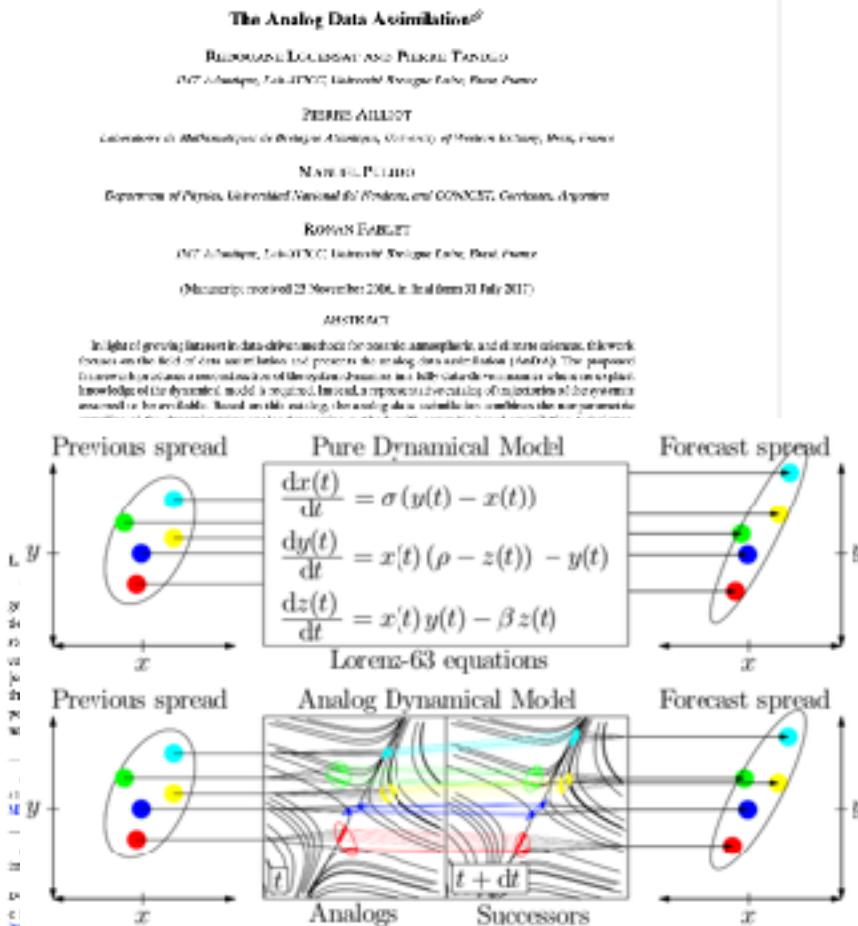
In following this paper, we shall find particular solutions of these equations analytically, especially when the solutions are periodic or invariant with time, and, in addition, much work has been devoted to examining such solutions by one means or another. Generally, however, nonperiodic solutions cannot readily be characterized by a finite set of variables, and, therefore, a procedure must be used to obtain a representation of a local neighborhood of a point in the solution. This is done by a set of vectors of these variables in terms of orthogonal functions. The governing laws then become a finite set of ordinary differential

Learning & Geoscience: Data-driven approaches for data assimilation

DOI:10.1002/2017JD027742

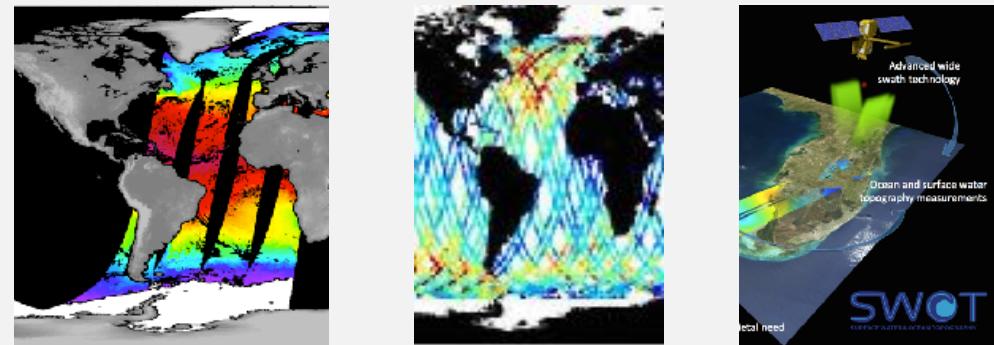
LUGUENSAT ET AL.

463



The analog data assimilation [Luguensat et al., 2017]

- Combination of analog forecasting strategies and EnKF assimilation schemes
- Extension to 2D+t geophysical dynamics



Open questions

- Bridging model-driven and data-driven paradigms
- Learning data-driven representations from real observation data

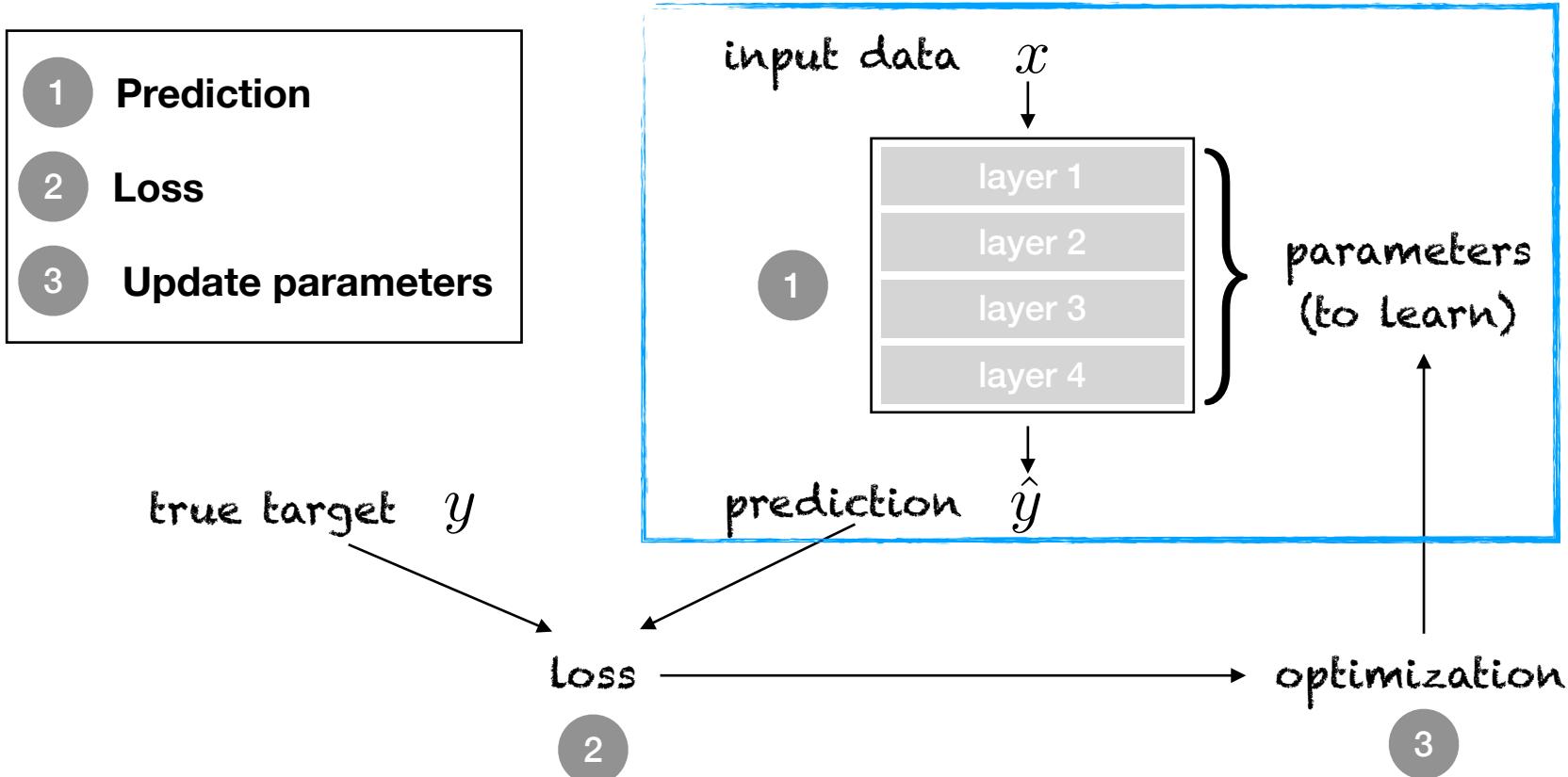
General context

Short reminder on NNs and Automatic Differentiation

Bridging physics paradigms and deep learning

Beyond Ocean Dynamics

What's learning (for a computer) ?



What's learning (for a computer) ?

Mathematical formulation: Learning comes to minimising some loss function given w.r.t. model parameters and training data

$$\hat{\theta} = \arg \min_{\theta} \mathcal{L} (\{x_i, y_i\}_{i \in \{1, \dots, N\}}; f_{\theta})$$

Key questions:

- Which parameterisation for model f ?
- Which loss function ?

Neural networks: composition idea

- Approximation through the composition of (simple) elementary functions:

$$f_{\theta}(x) = f_{\theta_N} \circ \dots \circ f_{\theta_2} \circ f_{\theta_1}(x)$$

- Key features:
 - Any continuous function can be approximated as the composition of elementary functions
 - Analytical/exact computation of the derivative of f with respect to parameters and input variables
 - Direct exploitation of gradient-based optimisation schemes for learning

Neural networks: automatic differentiation

- Given the general composition idea:

$$f_{\theta}(x) = f_{\theta_N} \circ \dots \circ f_{\theta_2} \circ f_{\theta_1}(x)$$

- NNs can implement automatic differentiation knowing the symbolic differentiation for elementary functions:

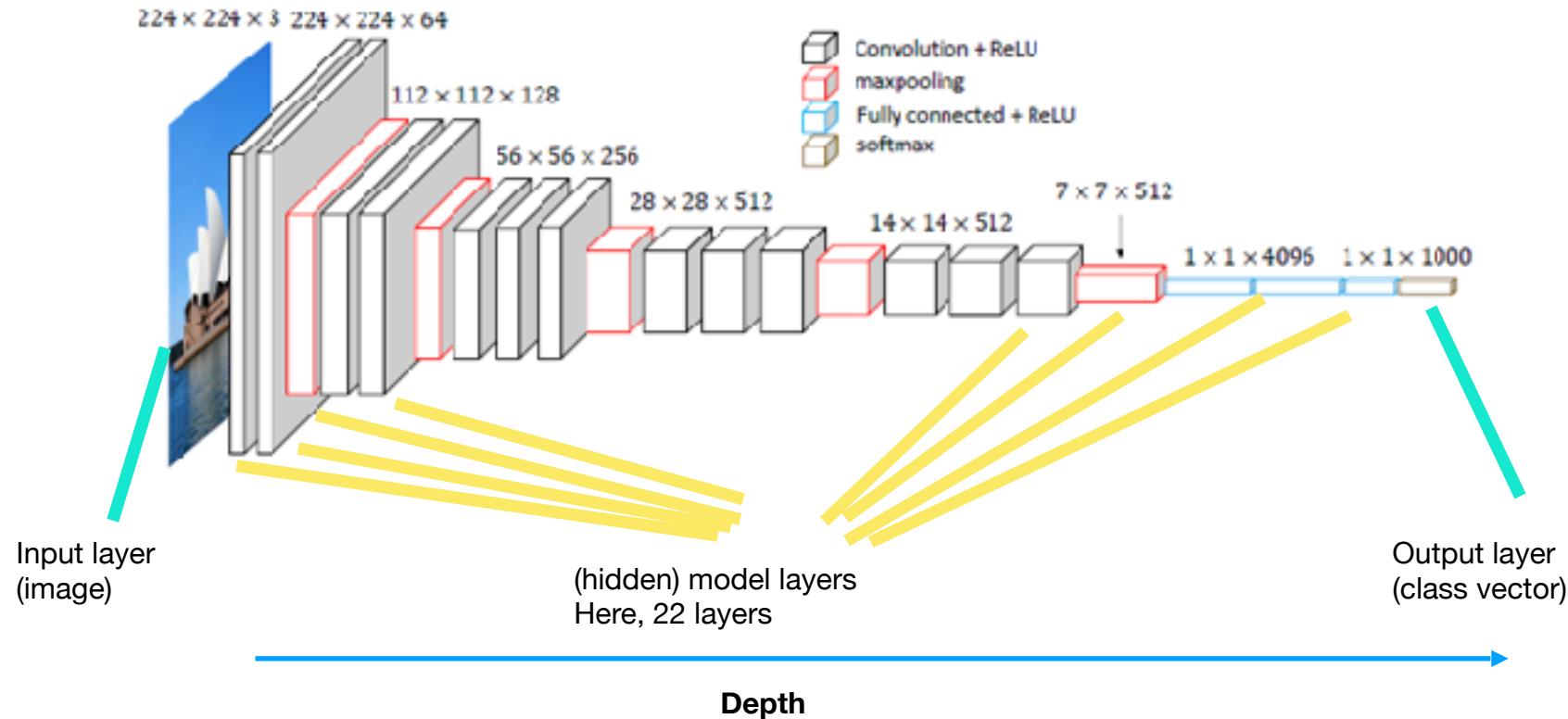
- AD example: https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html
- Basis of the backpropagation algorithm (similar to adjoint method)

- Resulting gradient descent for learning model parameters:

$$\widehat{\theta} = \arg \min_{\theta} \mathcal{L}(\{x_i, y_i\}_i; f_{\theta}) \quad \longrightarrow \quad \theta^{(k+1)} = \theta^{(k+1)} + \lambda_k \nabla_{\theta} \mathcal{L}(\{x_i, y_i\}_i; f_{\theta})$$

Deep learning models

DL models are (in general) feedforward models. VGG16 as an illustration



The more layers, the deeper..... Some models may have up to several hundreds to thousands of layers.

General context

Introduction to deep learning and NNs

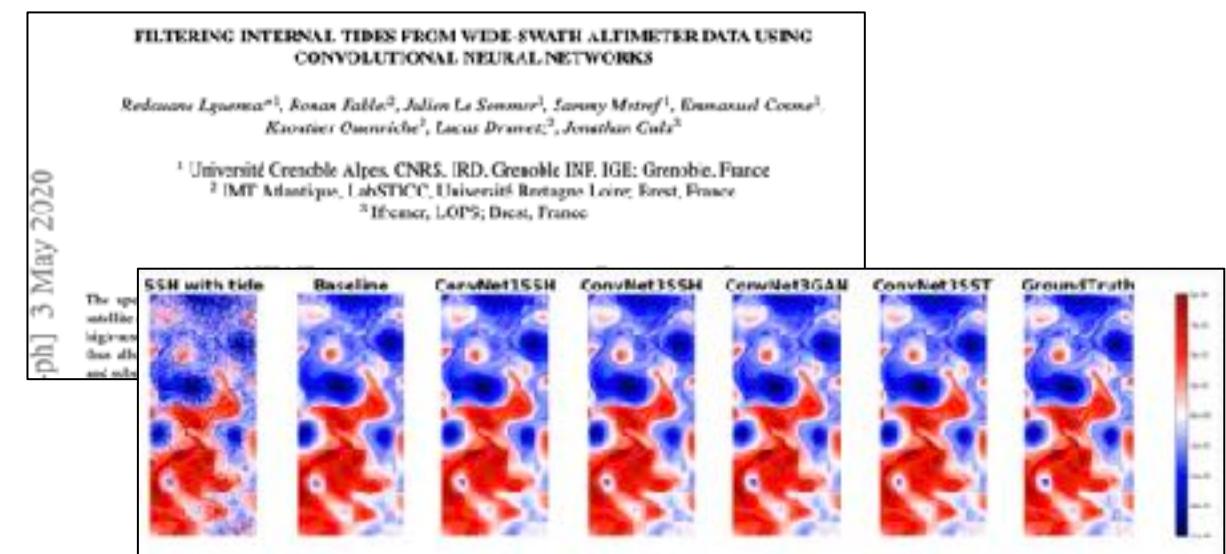
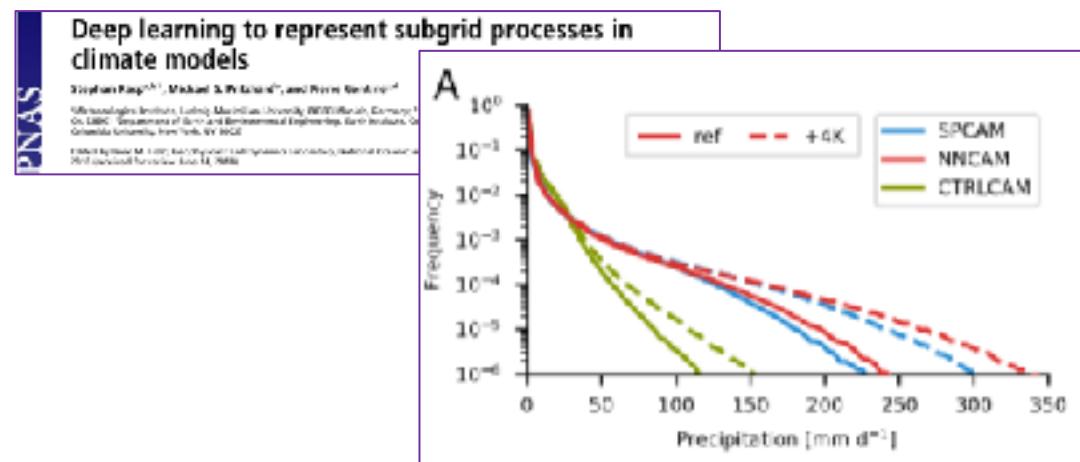
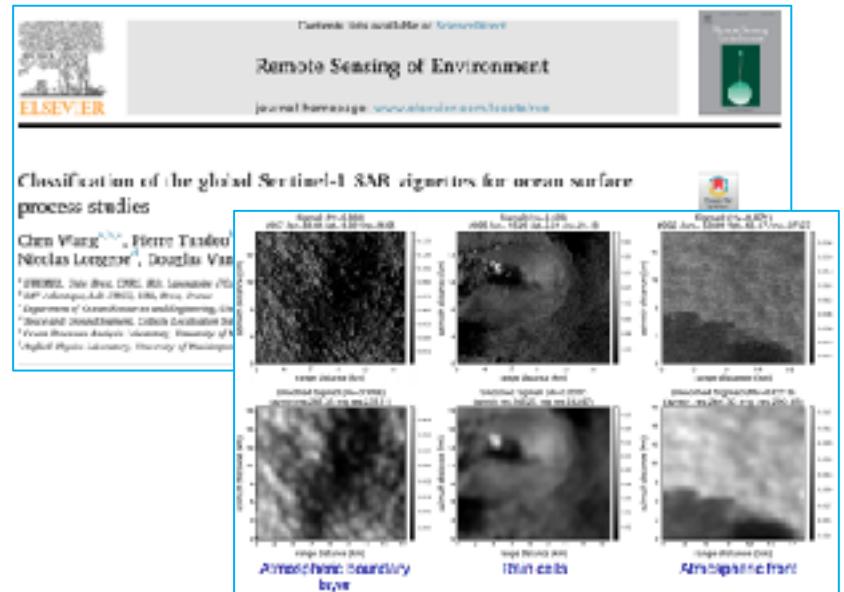
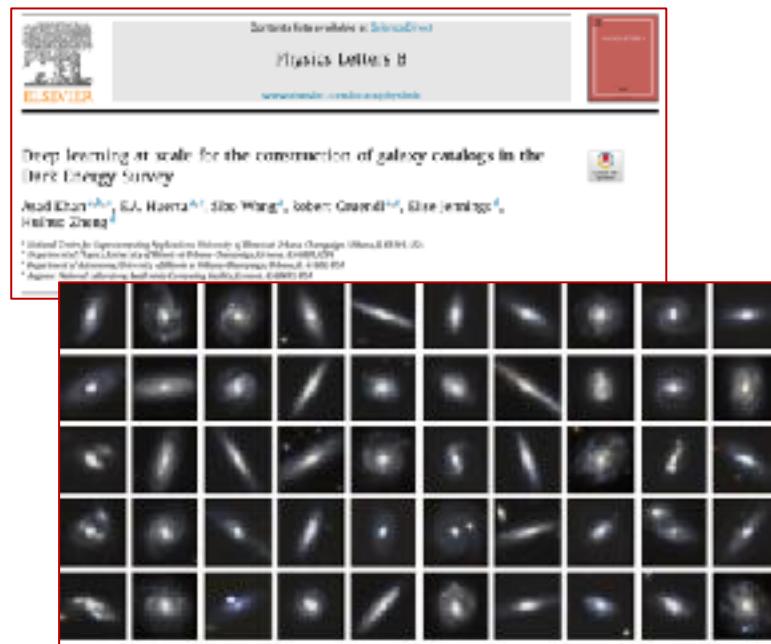
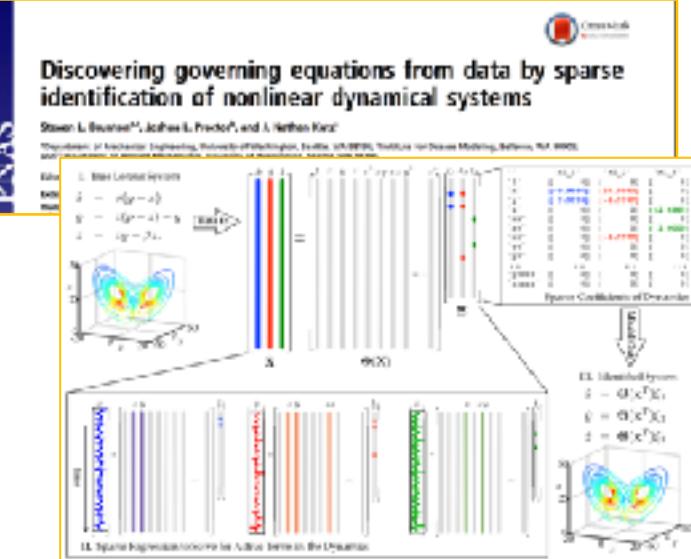
Bridging physics paradigms and deep learning

Beyond Ocean Dynamics

Deep Learning applied to physics

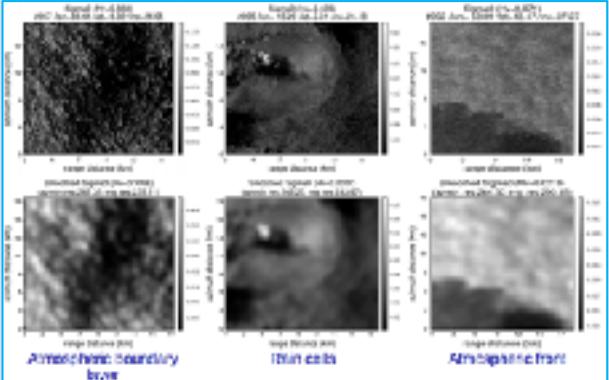
Lab-STICC

Direct applications of DL schemes to physics-related issues

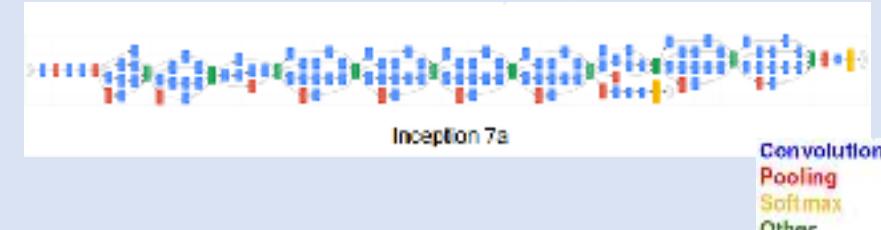


How to (directly) apply DL to physics-related issues ?

- 1 Build a groundtrued dataset



- 2 Design or choose an architecture



Many architectures available online
(VGG, ResNet, U-Net,....)

- 3 Choose a DL framework



K Keras

PYTORCH

- 4 Learn the model

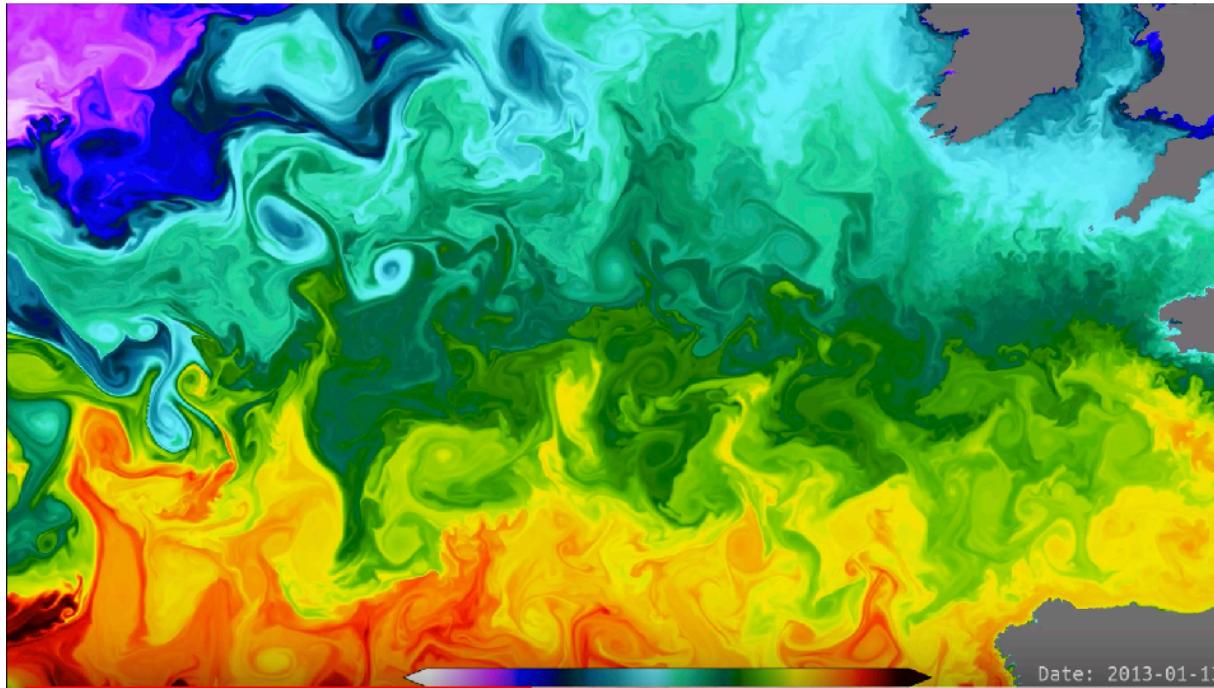
Select a training loss

Optimize model parameters using optimizers embedded in DL frameworks

A toy example using Tensorflow-Keras: <https://www.tensorflow.org/tutorials/keras/classification>

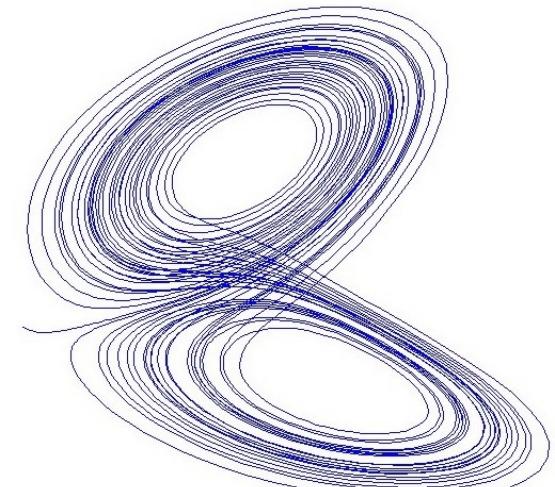
How to exploit physics prior in deep learning schemes?

How to model Geophysical Dynamics?



$$\begin{aligned}\frac{dx(t)}{dt} &= \sigma (y(t) - x(t)) \\ \frac{dy(t)}{dt} &= x(t) (\rho - z(t)) - y(t) \\ \frac{dz(t)}{dt} &= x(t) y(t) - \beta z(t)\end{aligned}$$

Lorenz-63 equations



What are chaotic dynamics?

https://github.com/CIA-Oceanix/DLGD2021/blob/main/lecture-5-inverse-problems/notebookPyTorch_InvProb_ModelBased_L63_Students.ipynb

Bridging physics & AI: a broader picture

Physical model

$$\frac{\partial u}{\partial t} + \langle \nabla u, v \rangle = \kappa \Delta u$$

Representation learning

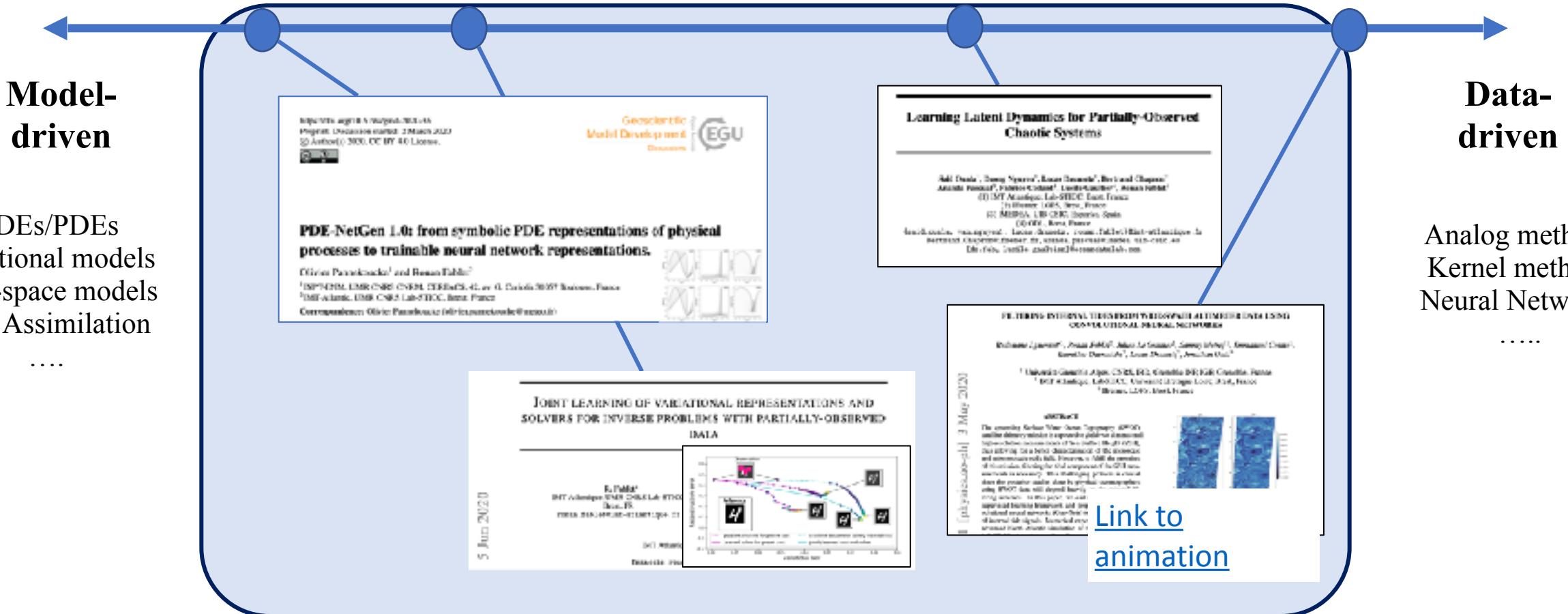
Trainable representation



Making the most of AI and Physics Theory

- Model-Driven/Theory-Guided & Data-Constrained schemes
- Data-Driven & Physics-Aware schemes (eg, Ouala et al., 2019)

Bridging Physics & AI: a broader picture



Physics-informed &
Data-constrained

Data-driven &
Physics-aware

Bridging physics & AI: Expected breakthroughs

Physical model

$$\frac{\partial u}{\partial t} + \langle \nabla u, v \rangle = \kappa \Delta u$$

Representation learning

Data-driven representation



Making the most of AI and Physics Theory

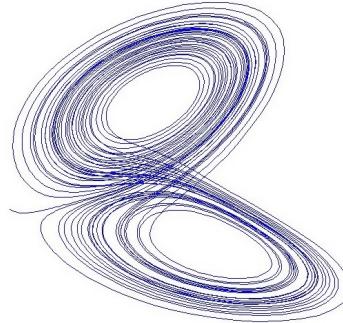
- **Model-Driven/Theory-Guided & Data-Constrained schemes**
- Data-Driven & Physically-Sound schemes (eg, Ouala et al., 2019)

How to embed physics-driven priors in DL models ?

An illustration through neural ODE for Lorenz-63 dynamics (Fablet et al., 2018)

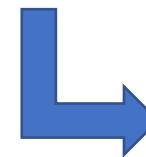
$$\begin{aligned}\frac{dx(t)}{dt} &= \sigma (y(t) - x(t)) \\ \frac{dy(t)}{dt} &= x(t) (\rho - z(t)) - y(t) \\ \frac{dz(t)}{dt} &= x(t) y(t) - \beta z(t)\end{aligned}$$

Lorenz-63 equations



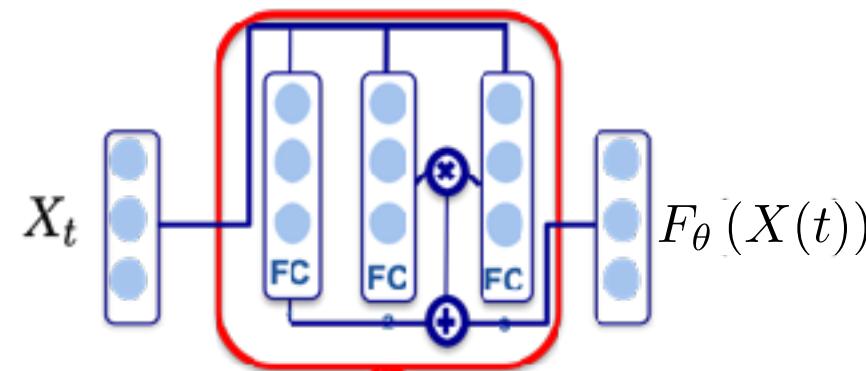
Associated Euler integration scheme

$$d_t X(t) = F_\theta (X(t))$$

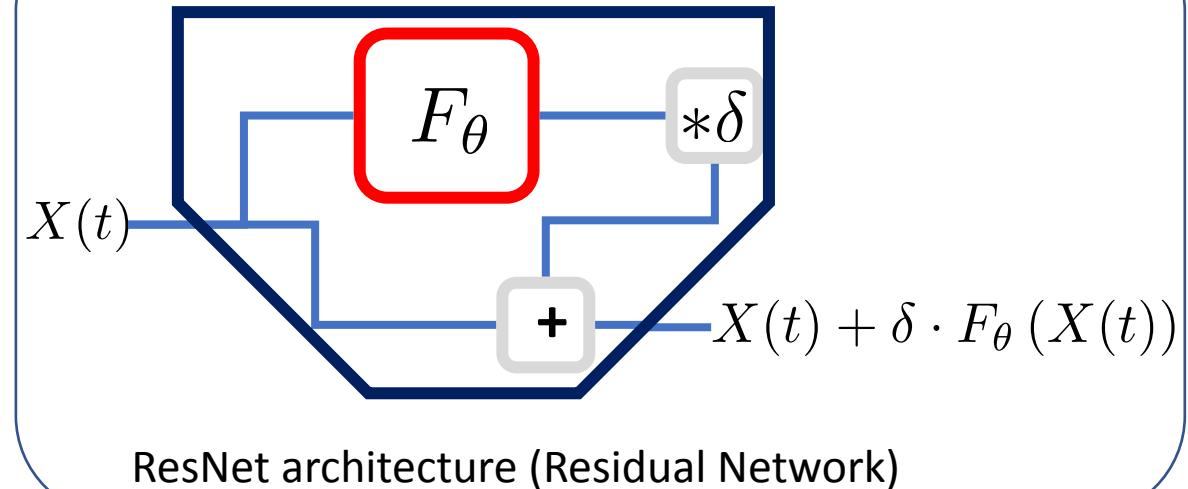


$$X(t + \delta) = X(t) + \delta \cdot F_\theta (X(t))$$

NN architecture for differential operator



NN architecture for integration scheme



How to embed physics-driven priors in DL models ?

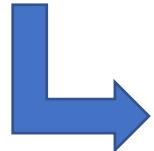
An illustration through neural ODE for Lorenz-63 dynamics (Fablet et al., 2018)

$$\begin{aligned}\frac{dx(t)}{dt} &= \sigma (y(t) - x(t)) \\ \frac{dy(t)}{dt} &= x(t) (\rho - z(t)) - y(t) \\ \frac{dz(t)}{dt} &= x(t) y(t) - \beta z(t)\end{aligned}$$

Lorenz-63 equations

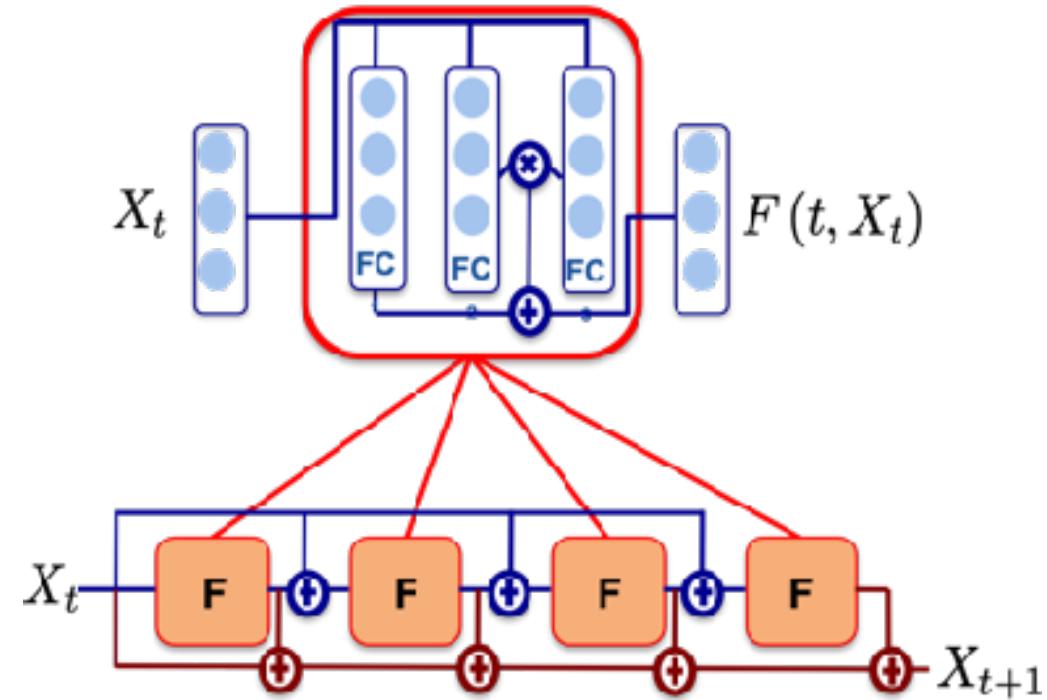
Generalization to higher-order integration schemes (eg, RK4)

$$d_t X(t) = F_\theta (X(t))$$



$$X(t + \delta) = X(t) + \sum_i \beta_i k_i$$

$$\text{with } k_i = F_\theta (X(t) + \delta \alpha_i k_{i-1})$$



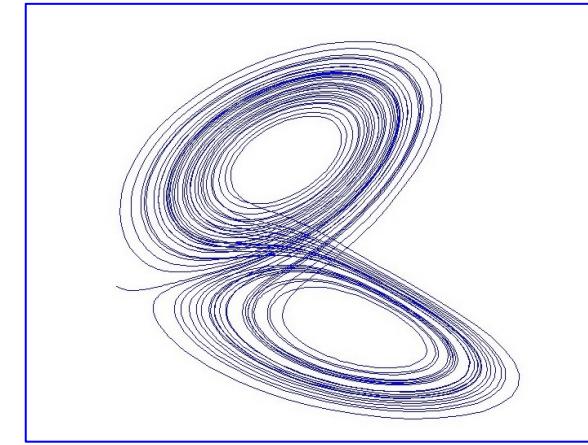
NB: Same number of trainable model parameters as the Euler-based architecture

How to embed physics-driven priors in DL models ?

An illustration through L63 dynamics: numerical experiments (Fablet et al., 2018)

Forecasting experiments			
Noise-free training data			
Forecasting time step	t_0+h	t_0+4h	t_0+8h
Analog forecasting	$<10^{-6}$	0.002	0.005
Sparse regression	$<10^{-6}$	0.002	0.006
MLP	$<10^{-6}$	0.018	0.044
<i>Bi-NN(4)</i>	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$

Noisy training data ($\sigma=0.5$)						
Forecasting time step	t_0+h	t_0+4h	t_0+8h	0	0.25	1
Analog forecasting	$<10^{-6}$	2.01	2.2			
<i>Bi-NN(4)</i>	$<10^{-6}$	0.054	0.14			



Assimilation experiment (1 obs. every 8 time steps)						
Noise standard deviation in training data	0	0.25	1			
<i>True model</i>	<u>0.50</u>	-	-			
Analog forecasting	0.65	1.17	1.81			
<i>Bi-NN(4)</i>	0.60	0.75	0.86			

NN Generator from Symbolic PDEs (Pannekoucke et al., 2020)

$$\partial_t u + u \partial_x u = \kappa \partial_x^2 u$$



Symbolic calculus
(*Sympy*)



PDE-GenNet
(*keras*)

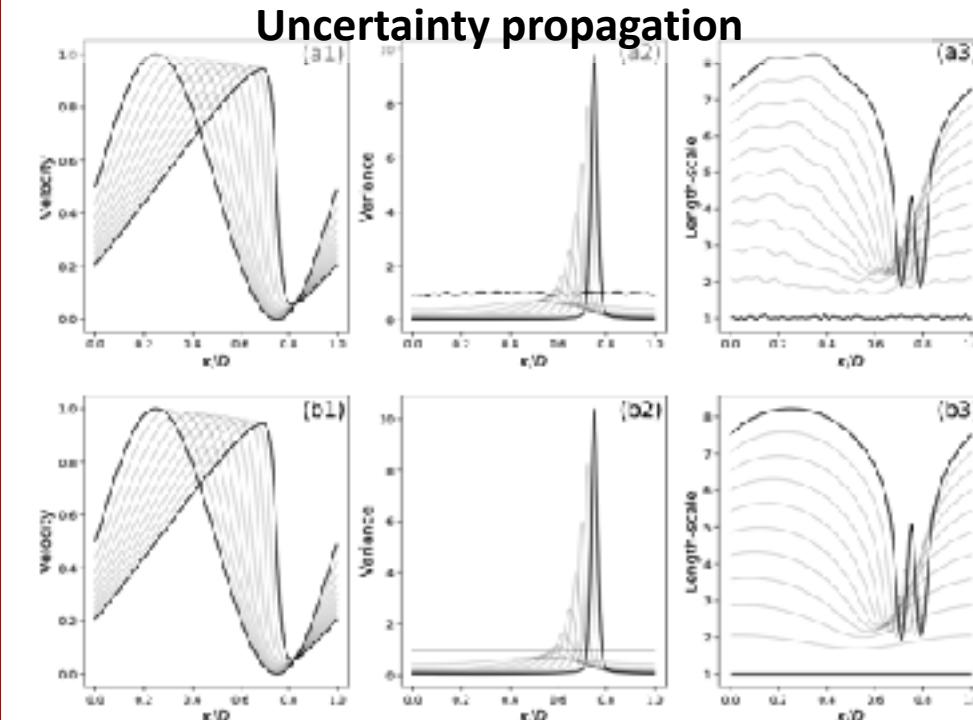


$$\left(\begin{array}{c} \mu_u(t) \\ \Sigma_u(t) \end{array} \right) \rightarrow \boxed{\text{ResNet}} \rightarrow \left(\begin{array}{c} \mu_u(t+1) \\ \Sigma_u(t+1) \end{array} \right)$$

```
# Example of computation of a derivative
kernel_Du_x_ol = np.asarray([[0.0, 0.0, 0.0], [0.0, 0.1/(2*self.dx), self.coordinates.index('x'))], [0.0, 0.0]]).reshape((3, 3)*4(1, 1))
Du_x_ol = DerivativeFactory((3, 3), kernel=kernel_Du_x_ol, name='Du_x_ol')(u)

# Computation of trend_u
mul_1 = keras.layers.multiply([Dkappa_11_x_ol, Du_x_ol], name='MulLayer_1')
mul_2 = keras.layers.multiply([Dkappa_12_x_ol, Du_y_ol], name='MulLayer_2')
mul_3 = keras.layers.multiply([Dkappa_12_y_ol, Du_x_ol], name='MulLayer_3')
mul_4 = keras.layers.multiply([Dkappa_21_y_ol, Du_y_ol], name='MulLayer_4')
mul_5 = keras.layers.multiply([Du_x_ol, kappa_11], name='MulLayer_5')
mul_6 = keras.layers.multiply([Du_y_ol, kappa_22], name='MulLayer_6')
mul_7 = keras.layers.multiply([Du_x_ol, kappa_12], name='MulLayer_7')
sc_mul_1 = keras.layers.Lambda(lambda x: 2.8*x, name='ScalarMulLayer_1')(mul_7)
trend_u = K
```

Generated code



Ensemble-based prediction

NN prediction

Bridging physics & AI: Expected breakthroughs

Physical model

$$\frac{\partial u}{\partial t} + \langle \nabla u, v \rangle = \kappa \Delta u$$

Representation learning

Data-driven representation



Making the most of AI and Physics Theory

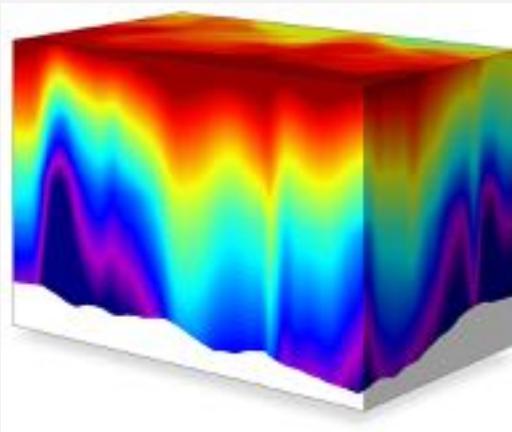
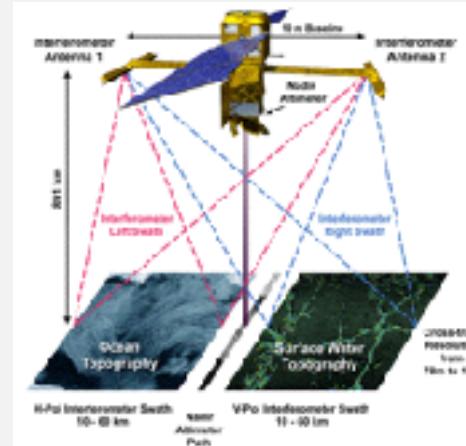
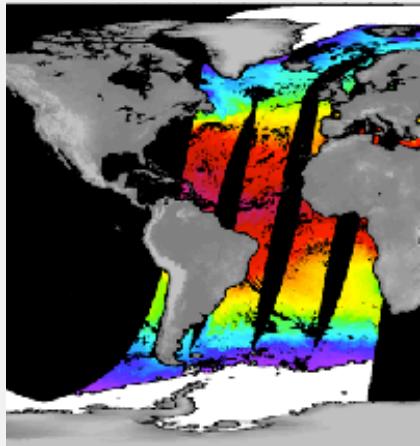
- Model-Driven/Theory-Guided & Data-Constrained schemes
- Data-Driven & Physically-Sound schemes (eg, Ouala et al., 2019)

Focus on Inverse Problems and Deep Learning

Deep learning for irregularly-sampled and partially-observed systems

Dealing with partially-observed and irregularly-sampled ocean dynamics

Can we learn directly from observation data ?



*Generic issue:
Joint identification and inversion*

Dynamical model

$$X_t \xrightarrow{\quad} \partial_t X = F(X, \xi, t, \theta) \xrightarrow{\quad} X_{t+1}$$



Observation model

$$Y_t = H(X, \zeta, t, \phi)$$

Fablet et al. 4DVarNet: Trainable Data Assimilation for Sea Surface dynamics

<https://cia-oceanix.github.io/>

Method

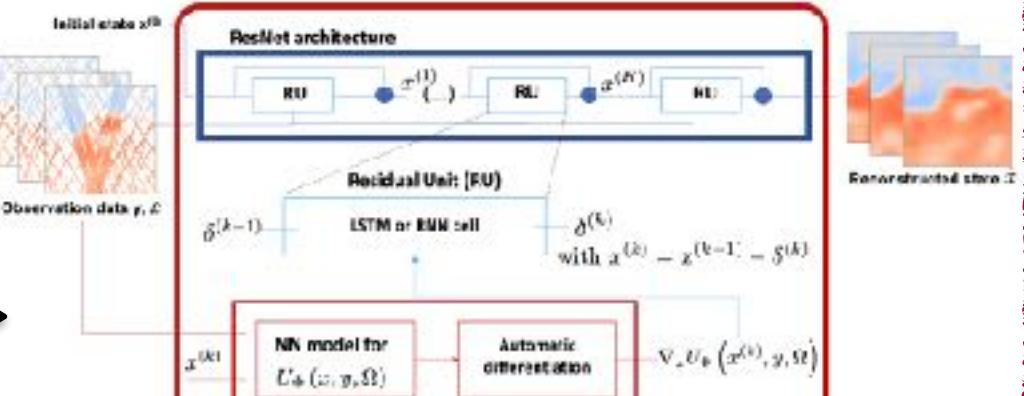
From a Variational DA formulation

$$\hat{x} = \arg \min_x \|x - y\|^2 + \lambda \|x - \phi(x)\|^2$$

Trainable variational model

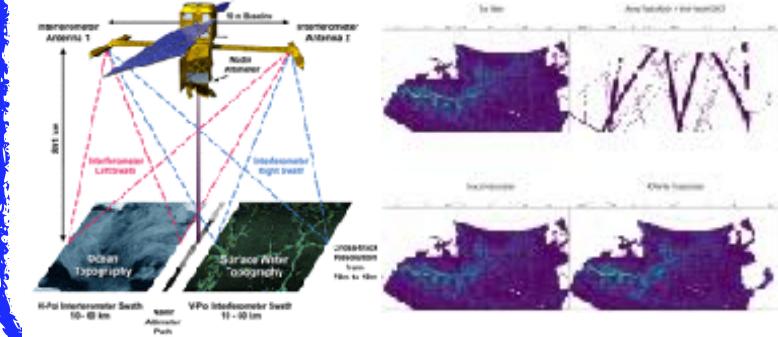
Trainable gradient-based solver

Associated end-to-end scheme

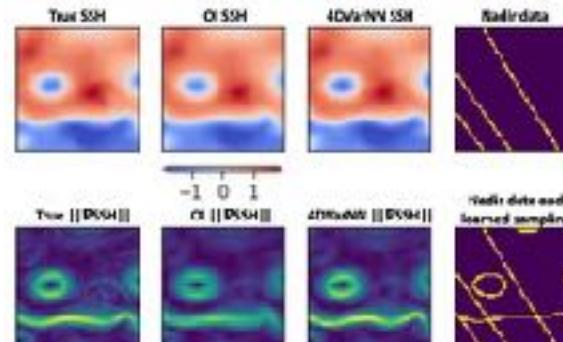


Applications

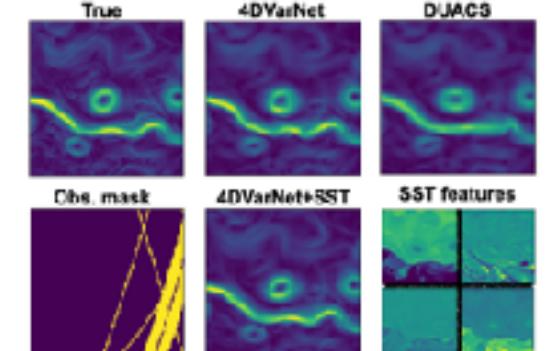
Interpolation & Forecasting



Learning where to sample ?

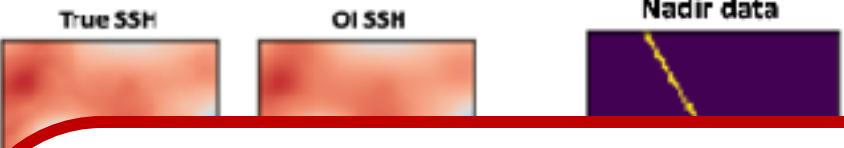


Multimodal learning

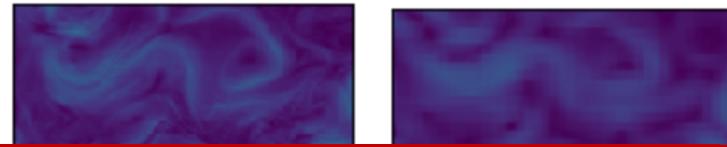


Trainable observation operators

Learning where to sample ?



Learning what to measure ?



4DVarNet models with trainable observation models

$$\hat{x} = \arg \min_x \|x - y\|^2 + \lambda \|x - \phi(x)\|^2$$

Sparse sampling operator

$$\|H(z) * (x - y)\|^2$$

$$\text{s.t. } \forall z, \|H(z)\|_1 < \epsilon$$

Multimodal observation

$$\begin{aligned} & \|x - y\|^2 \\ & + \alpha \|G * x - F * z\|^2 \end{aligned}$$

Trainable observation operators

4DVarNet models with trainable observation models

Spase sampling operator

$$\|H(z) * (x - y)\|^2$$

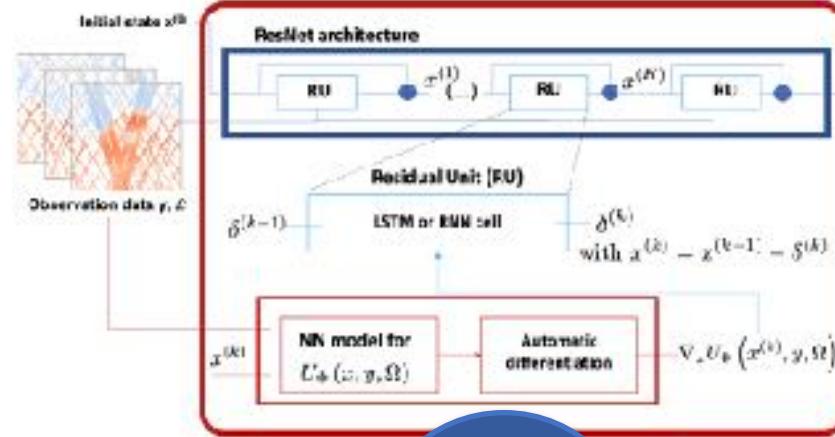
$$\text{s.t. } \forall z, \|H(z)\|_1 < \epsilon$$

Multimodal observation

$$\|x - y\|^2$$

$$+ \alpha \|G * x - F * z\|^2$$

End-to-end 4DVarNet



Supervised training loss

(under sparsity constraint for the optimal sampling case)

Multimodal data assimilation

SSH-SST case-study

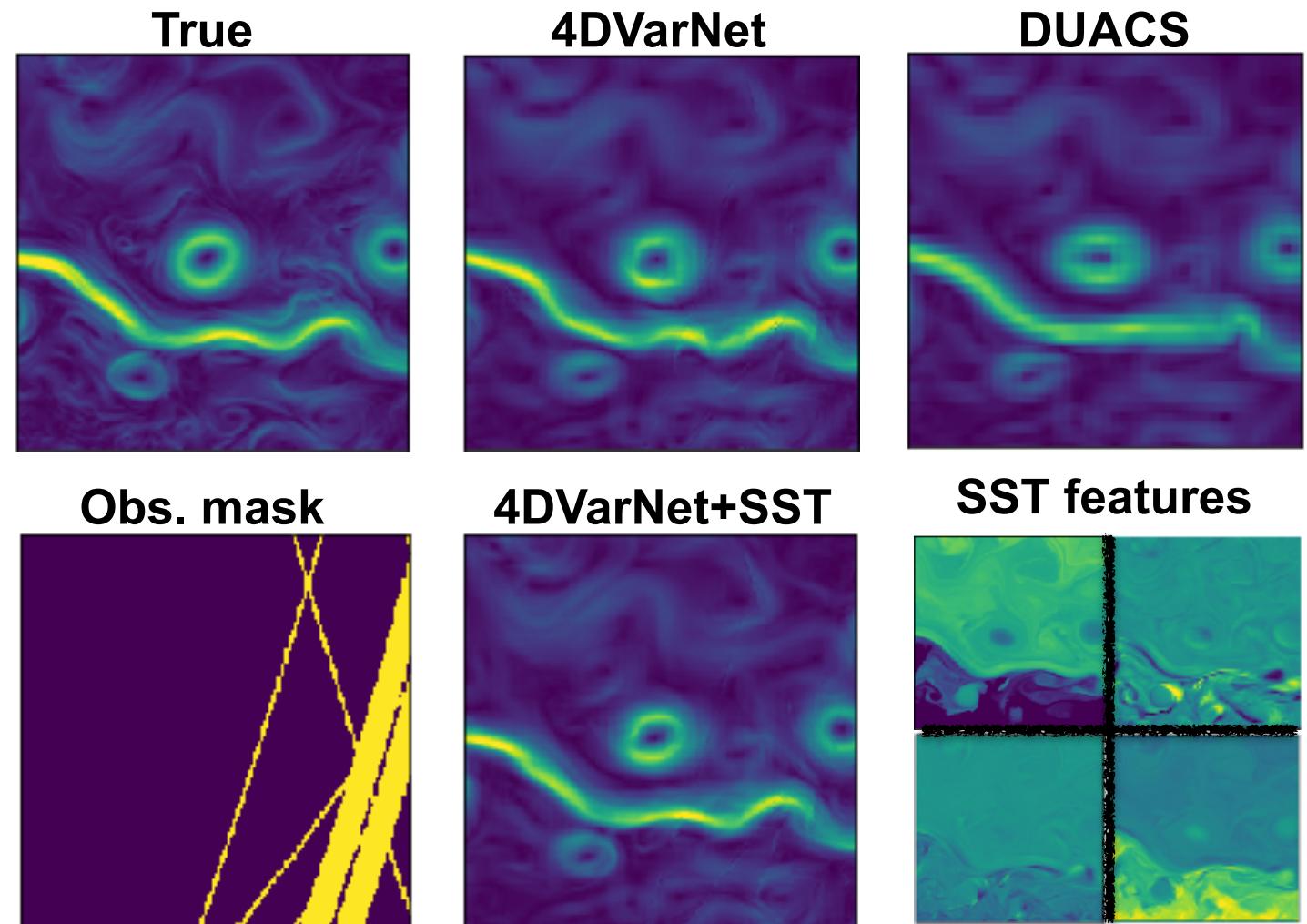
OSSE with NATL60 data

4-nadir-altimeter + SWOT +
DUACS baseline

Gulf Stream area ($10^\circ \times 10^\circ$)

63% vs. 53% gain in SSH
MSE w.r.t. DUACS with/
without SST (Winter period)

50% vs. 13% gain in SSH
MSE w.r.t. DUACS using
nadir altimeter data only



Optimal sampling

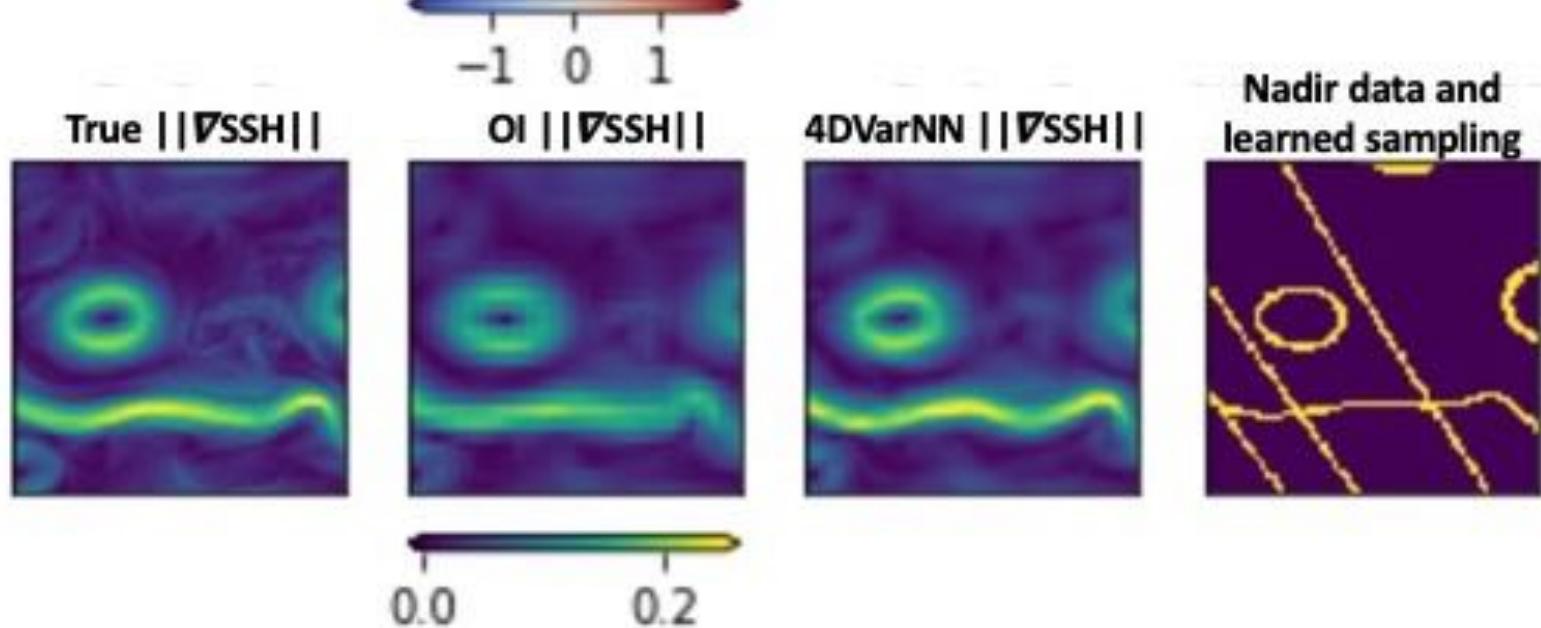
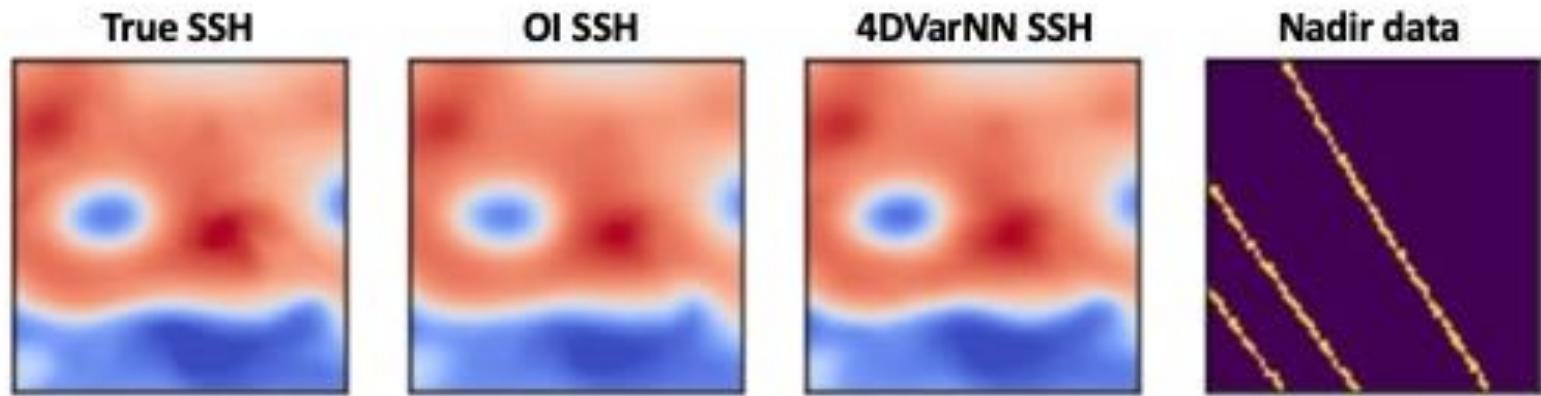
SSH case-study

OSSE with NATL60 data

4-nadir-altimeter + DUACS baseline

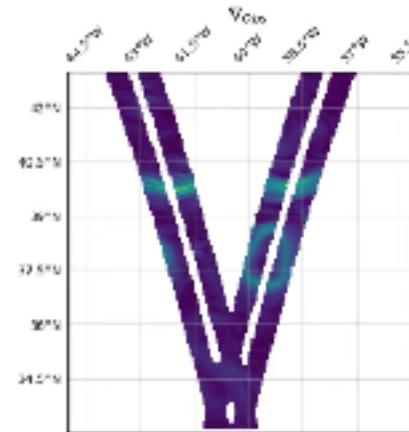
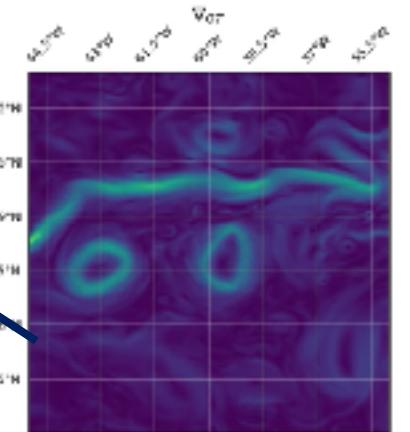
Gulf Stream area ($10^\circ \times 10^\circ$)

Mean relative gain of 60%
in the reconstruction of the
SSH using the learned
sampling (~6% of the pixels
vs. 1.3% for nadir altimeters)

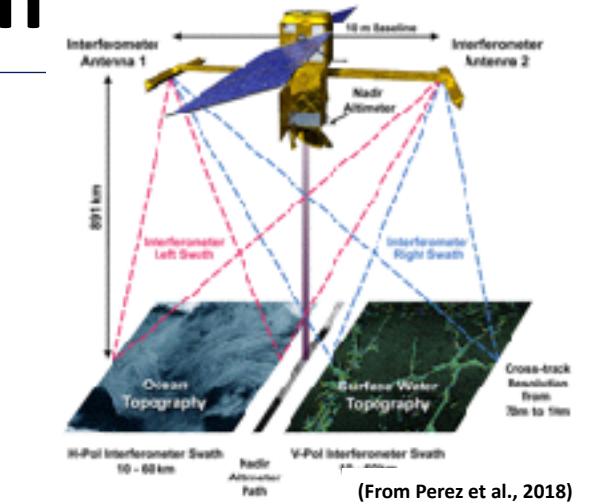
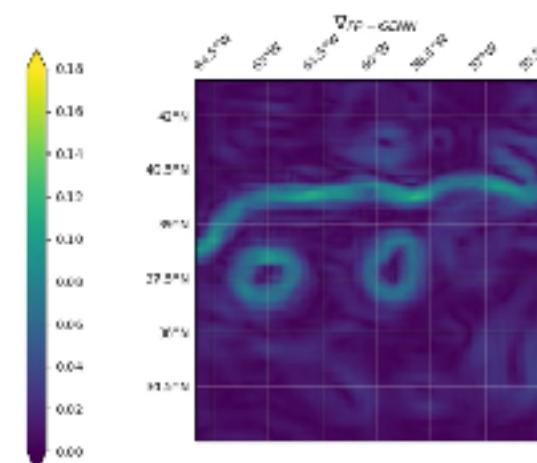
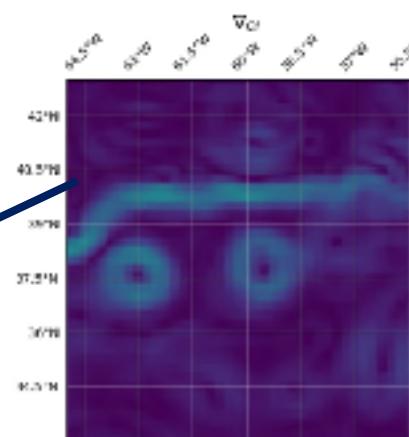


An application for upcoming SWOT mission

Groundtruth



State-of-the-art
operational processing

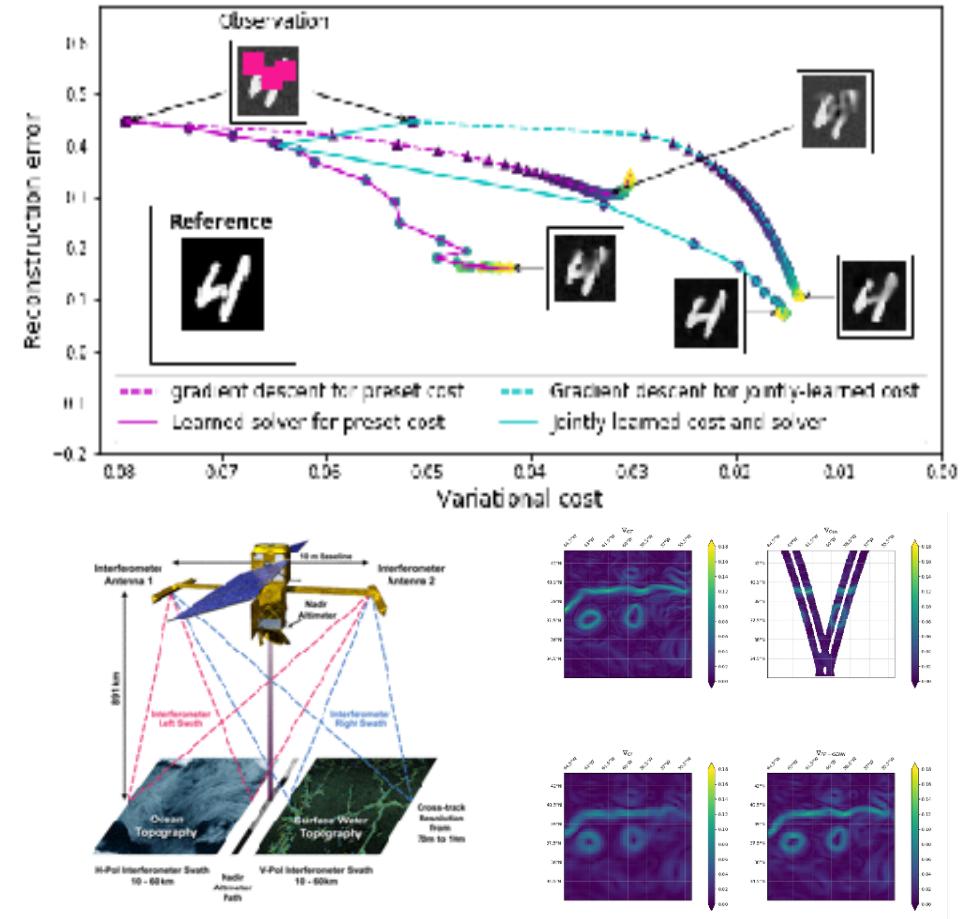


Proposed NN framework
(Fablet et al., 2019)

End-to-end learning for inverse problems (Fablet et al., 2020)

Key messages

- We can bridge DNN and variational models to solve inverse problems
- Learning both variational priors and solvers using groundtruthed (simulation) or observation-only data
- The best model may not be the TRUE one for inverse problems
- Generic formulation/architecture beyond space-time dynamics

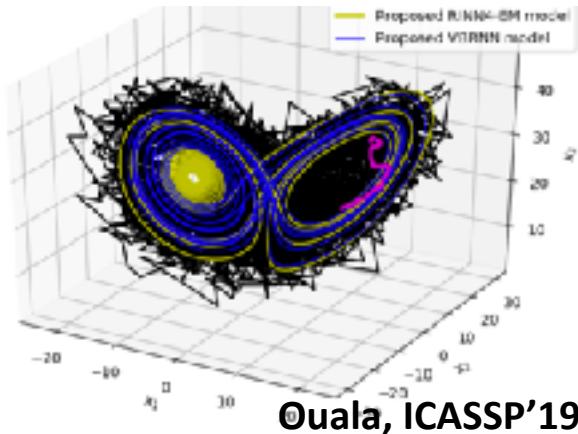


Preprint: <https://arxiv.org/abs/2006.03653>

Code: <https://github.com/CIA-Oceanix>

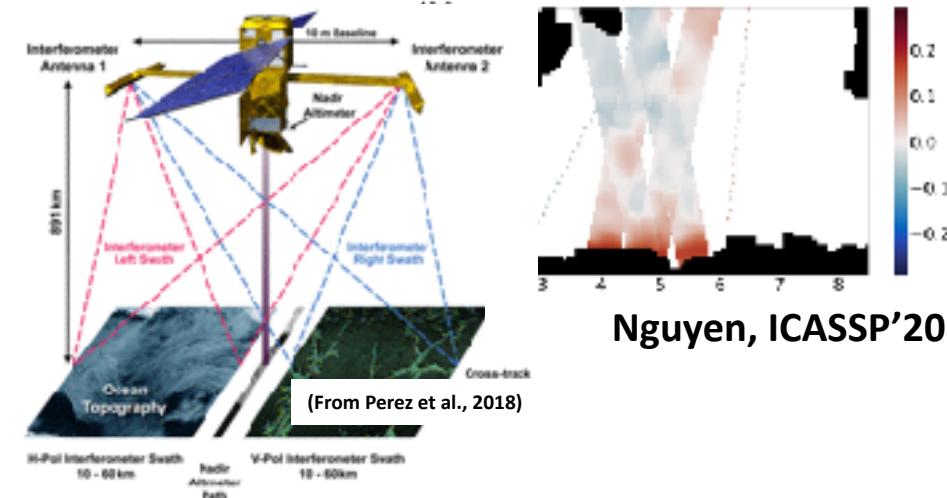
End-to-end learning from real observation data ?

Scarce time sampling



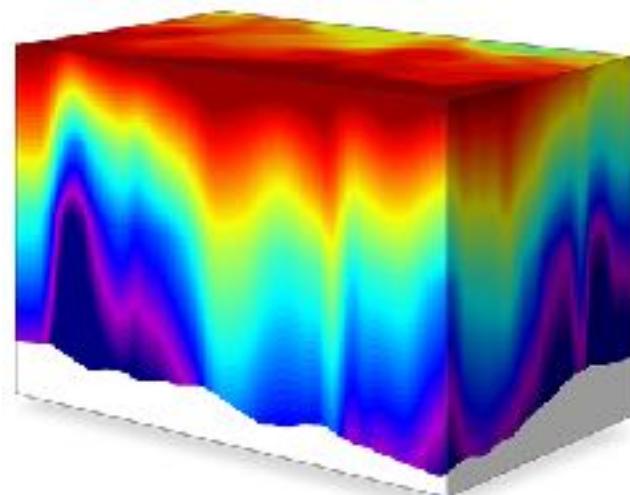
Ouala, ICASSP'19

Noisy and irregular sampling



Nguyen, ICASSP'20

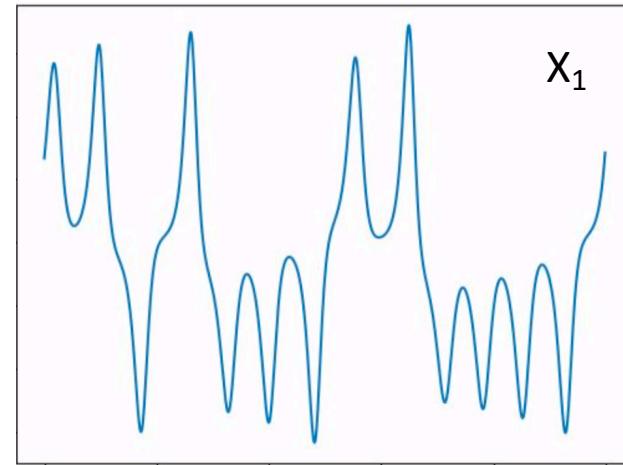
Partially-observed
system



Ouala, preprint 2019

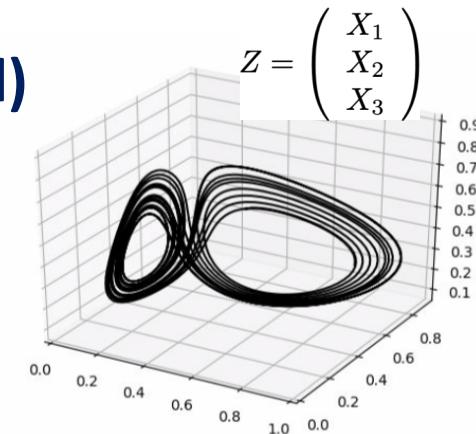
Neural ODE for partially-observed systems [Ouala et al., 2020]

Illustration for L63 assuming only the first components is observed

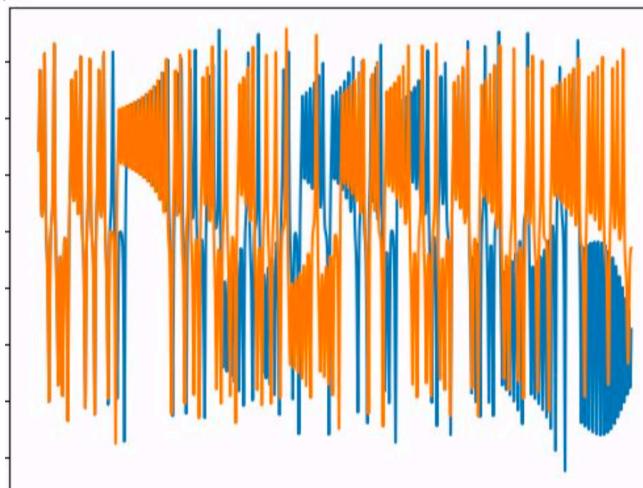


Learning Latent (unobserved) dynamics

$$d_t Z_t = \Phi_\theta(Z_t)$$



Objectives: accurate short-term forecast and realistic « long-term » patterns for X_1



Approach: trainable variational formulation with latent dynamics

Neural ODE for partially-observed systems [Ouala et al., 2019]

Problem statement: end-to-end learning of the latent (augmented) space and of the associated dynamics

$$X_t = \begin{pmatrix} x_t \\ z_t \end{pmatrix}$$

Observed variables
Unknown variables

Dynamical model in the latent space

$$\partial_t X_t = f_\theta (X_t)$$

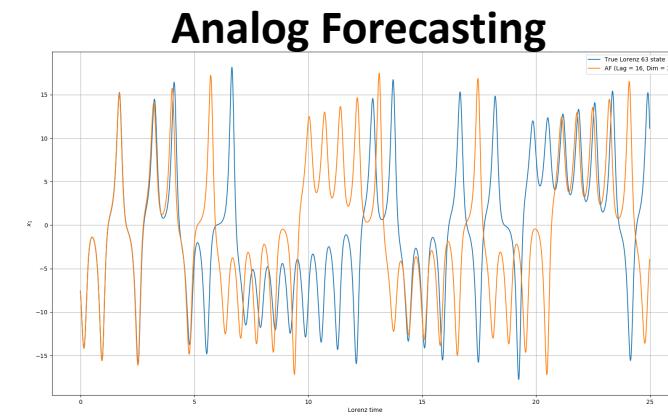
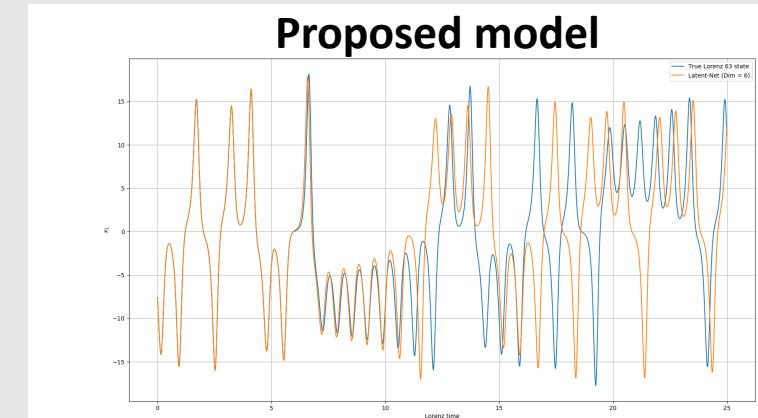
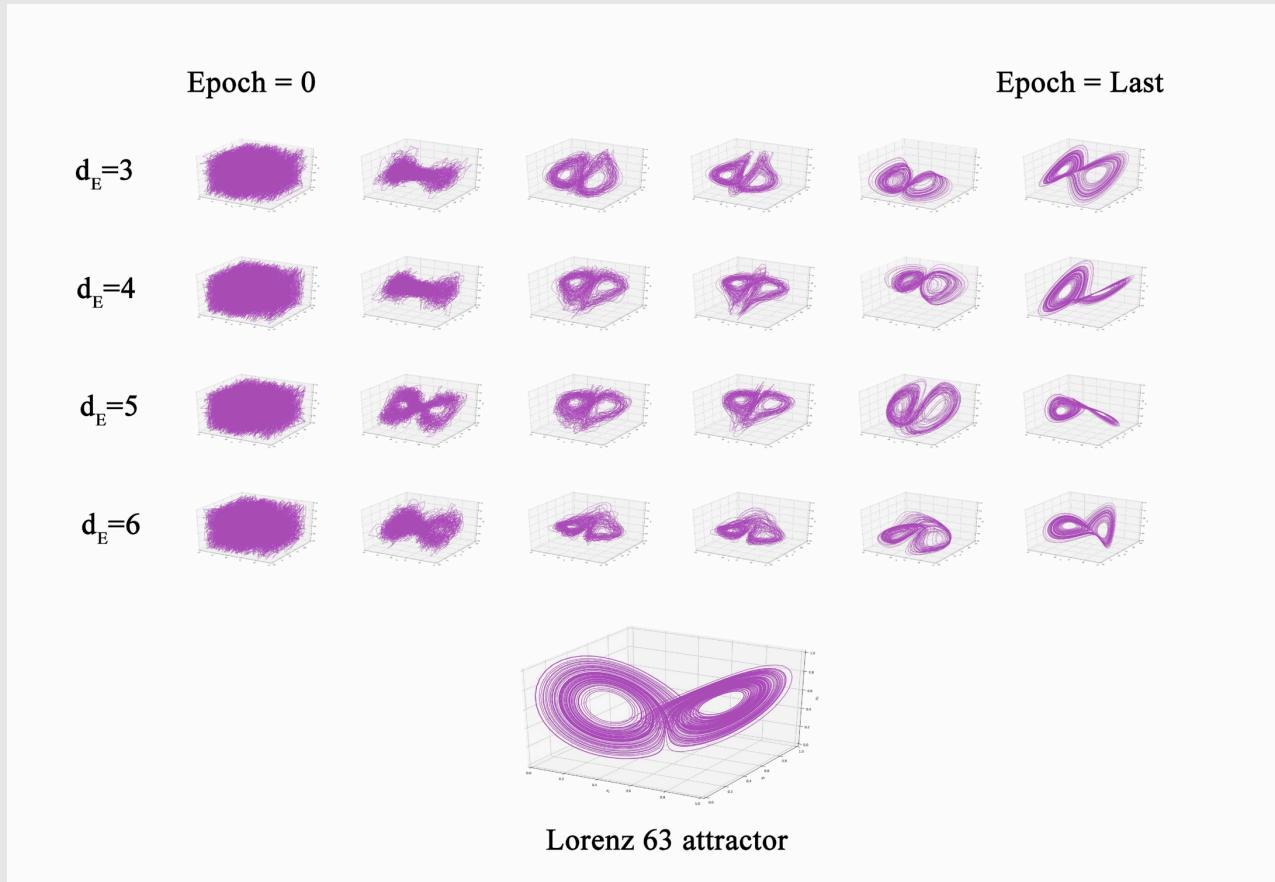
Goals:

1. Learn model parameters θ from observed time series
2. Forecast future observed states given previous ones

Proposed approach: WC 4DVar formulation with an unknown dynamical model

Neural ODE for partially-observed systems [Ouala et al., 2020]

Illustration on Lorenz-63 dynamics



Summary

- *NNs as numerical schemes for ODE/PDE/energy-based representations of geophysical flows*
- *Embedding geophysical priors in NN representations* (e.g., Lguensat et al., 2019; Ouala et al., 2019)
- *End-to-end architecture for jointly learning a representation (eg, ODE) and a solver* (e.g., Fablet et al., 2020)
- *Towards stochastic representations embedded in NN architectures* (e.g., Pannekoucke et al., 2020, Nguyen et al., 2020)

Beyond Ocean Dynamics

Learning stochastic hidden dynamics

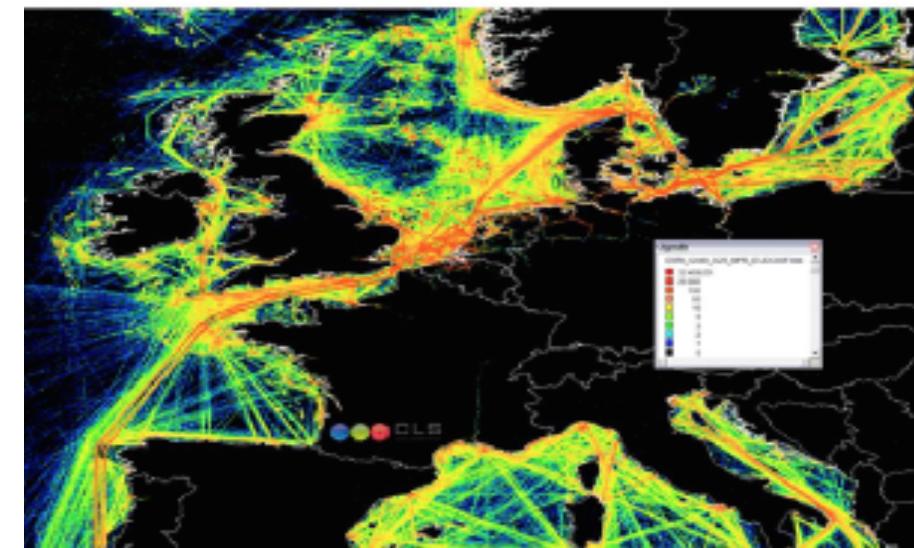
Lab-STICC

Learning stochastic hidden dynamics [Nguyen et al., 2018]

The example of AIS Vessel trajectory data

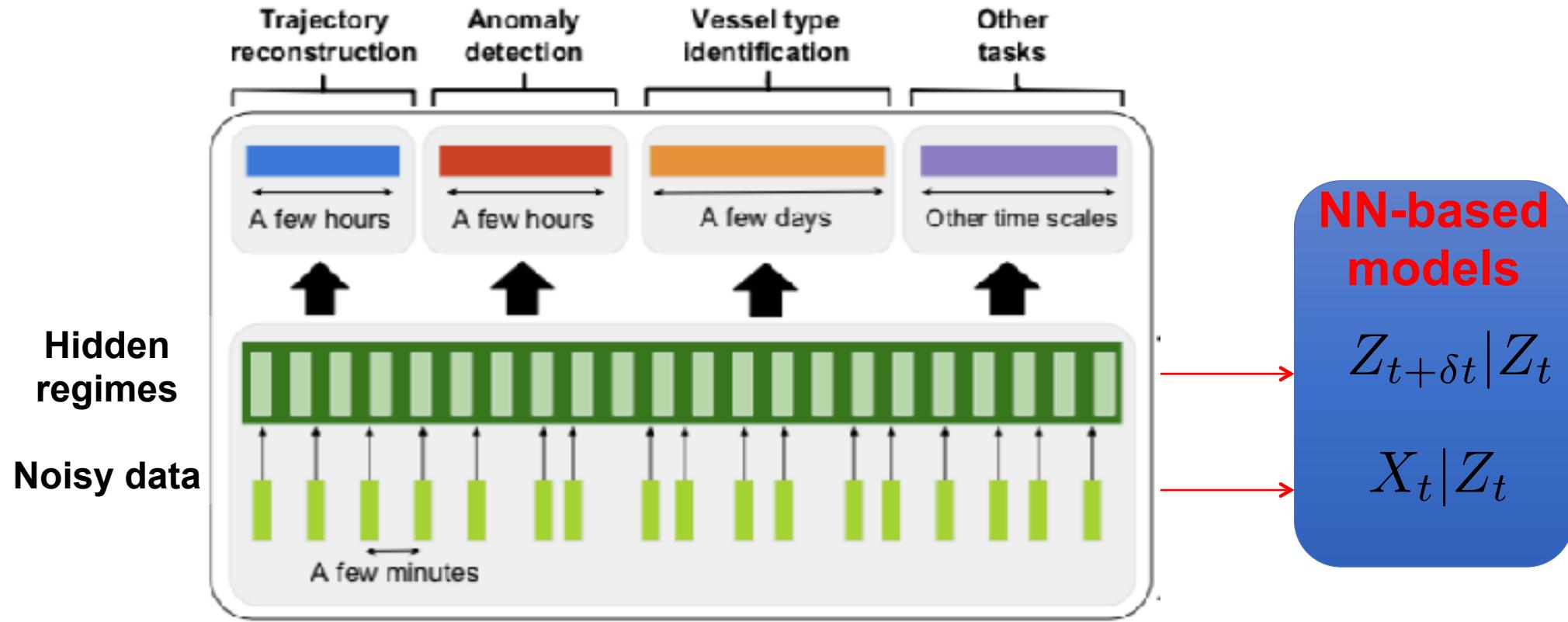


- Millions of AIS positions daily
- Noisy data: irregular sampling, corrupted data



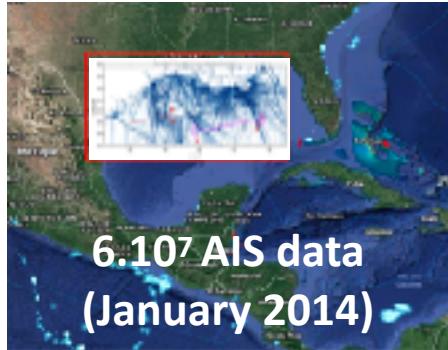
How can we learn from AIS data streams ?

Learning stochastic hidden dynamics [Nguyen et al., 2018]



Model training from noisy AIS streams using variational Bayesian approximation

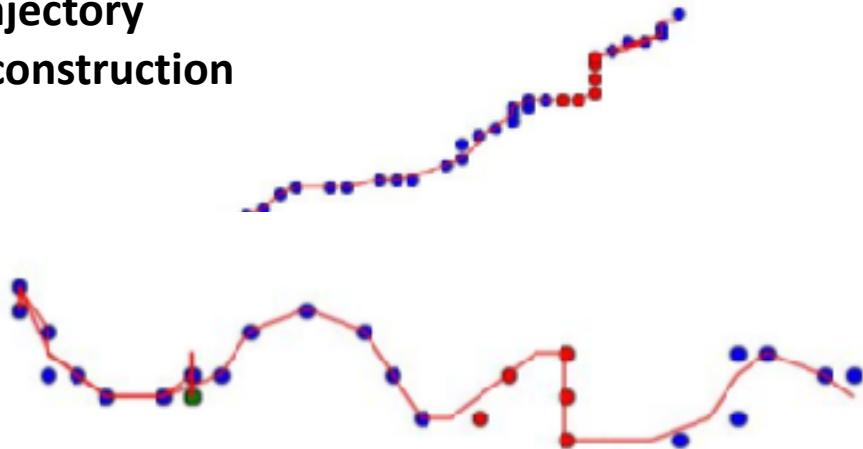
Learning stochastic hidden dynamics [Nguyen et al., 2018]



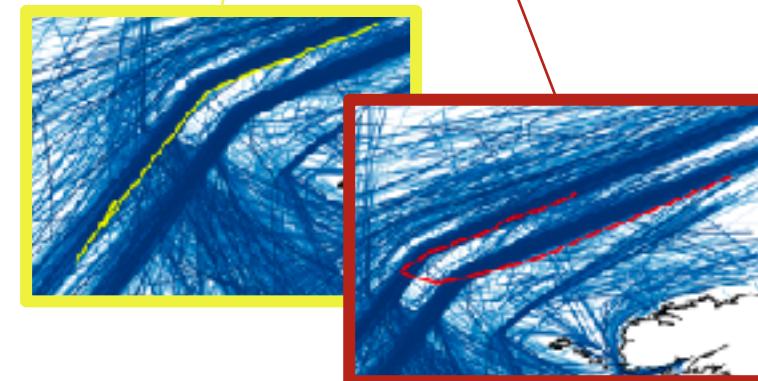
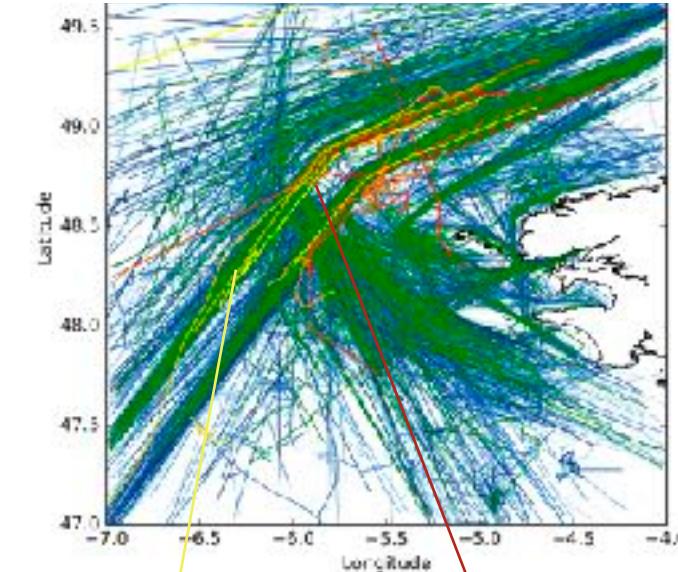
Vessel type recognition

~88% of correct
recognition

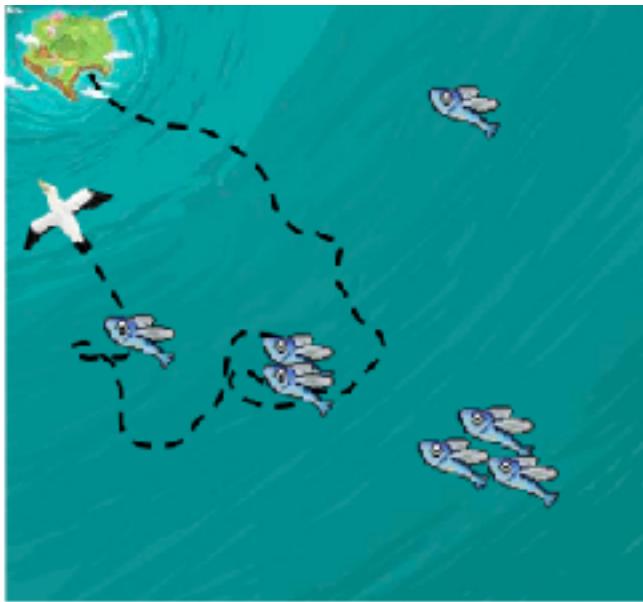
Trajectory
reconstruction



Abnormal behaviour detection



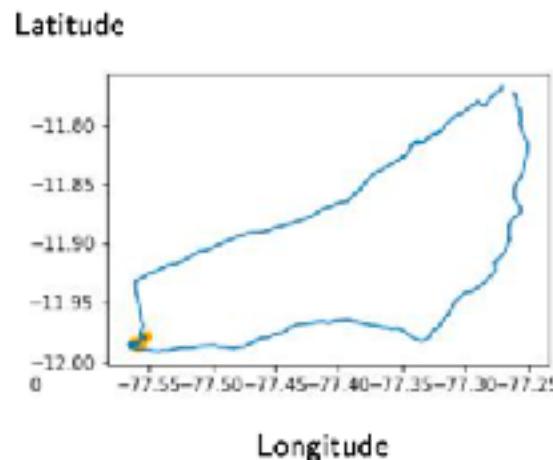
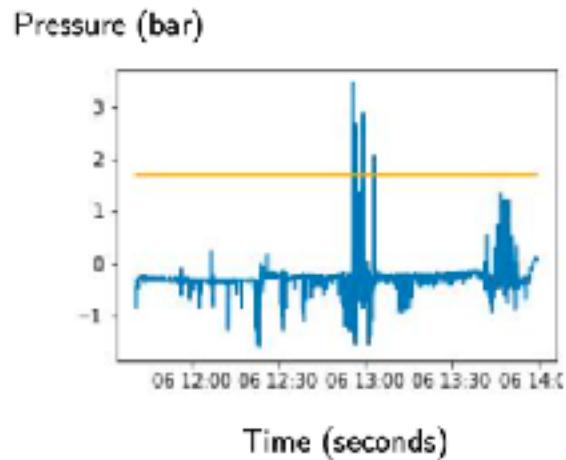
Seabird movement simulation [Roy et al., 2021]



©S.Bertrand



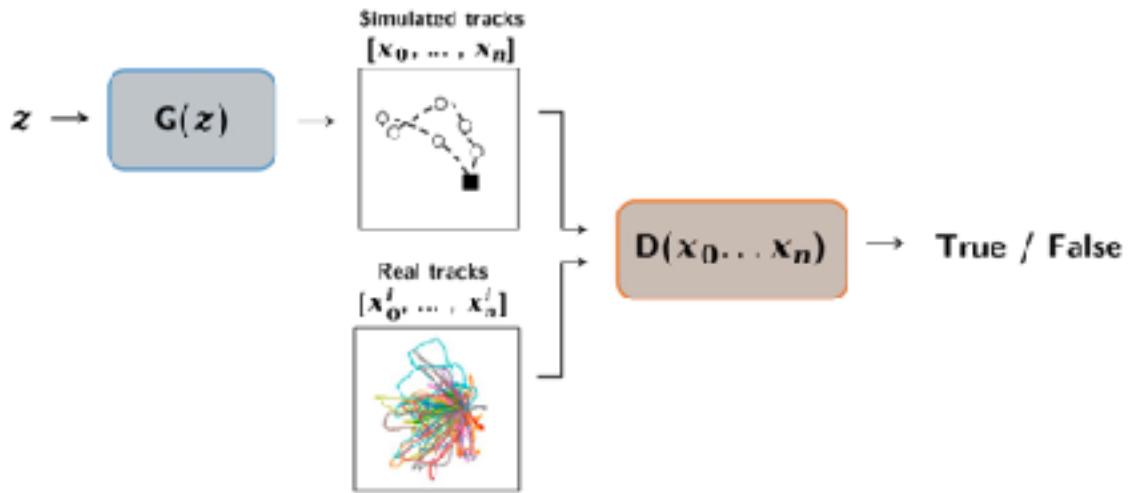
©C.Barbraud



Can we learn probabilistic models to simulate seabird trajectories ?

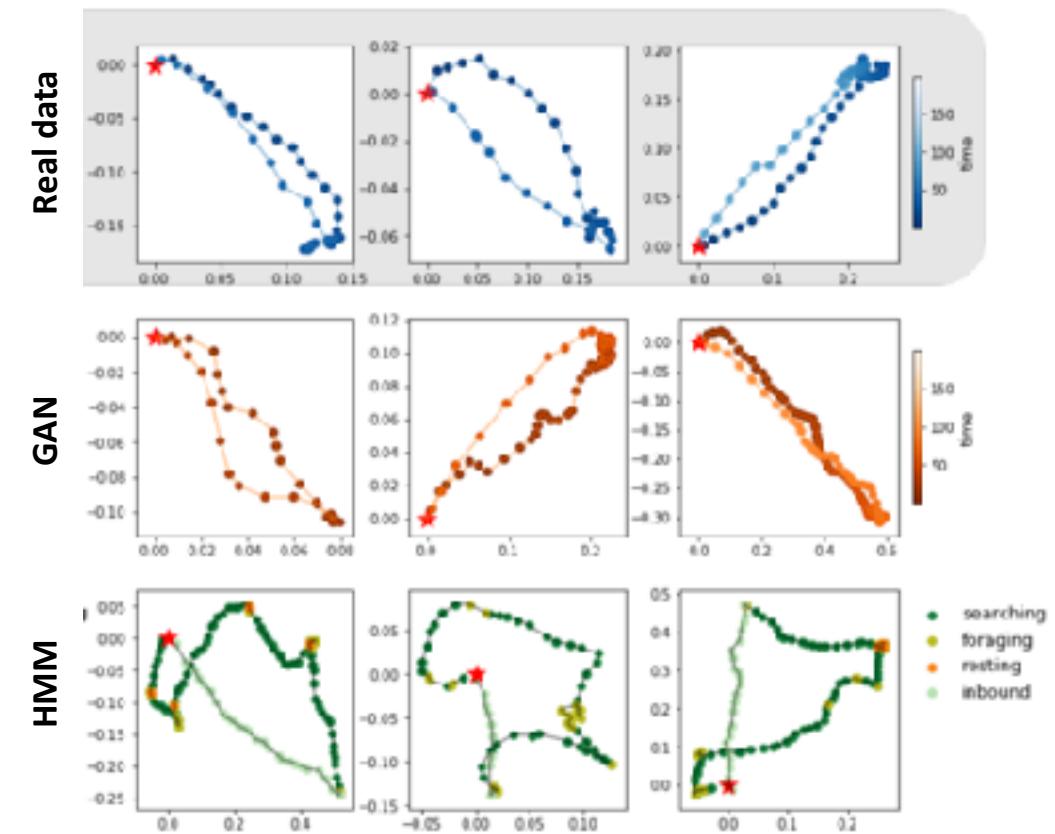
Seabird movement simulation [Roy et al., 2021]

GAN framework



Different architectures for the generator and discriminator (eg, sequential vs. Non-sequential)

Better fit than HMM-based models



Beyond Ocean Dynamics

Dynamical System Theory for Deep Learning

Lab-STICC

Understanding DL models ?



adversarial
perturbation

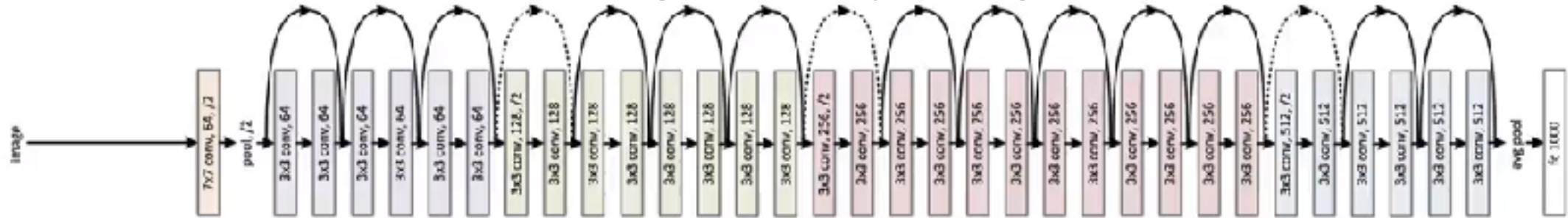


88% **tabby cat**

99% **guacamole**

Understanding ResNets [Rousseau et al., 2019]

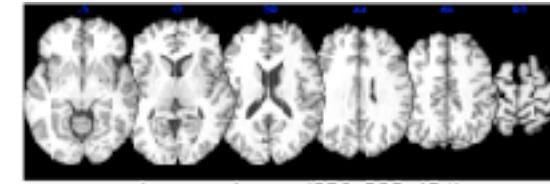
ResNet [He et al., 2015] regarded as space registration machines



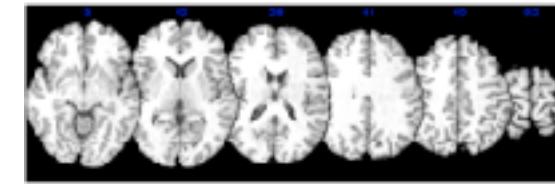
- Image registration examples



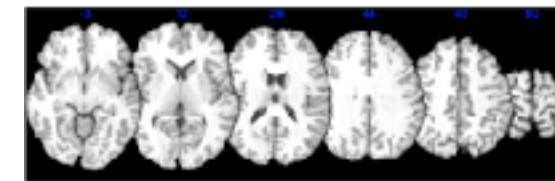
[Matlab tutorial]



a) source image (256x256x124)



b) target image (256x256x124)

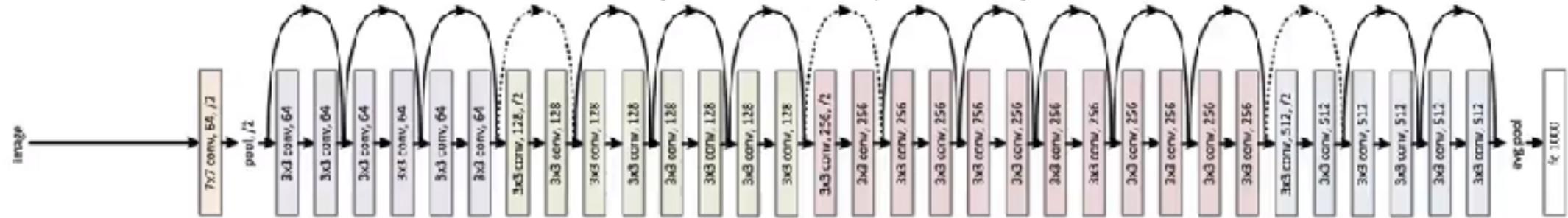


c) registered image (source to target)

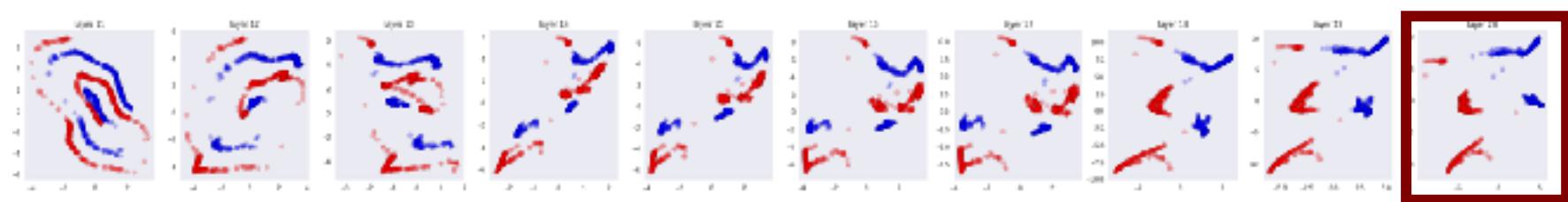
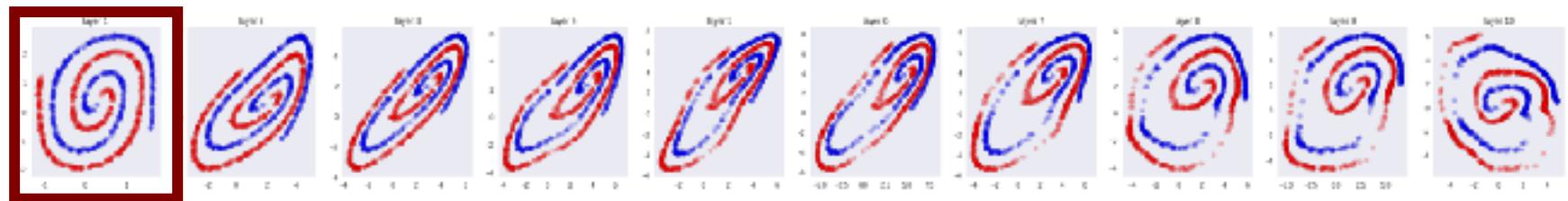
[Dramms tutorial]

Understanding ResNets [Rousseau et al., 2019]

ResNet [He et al., 2015] regarded as space registration machines



Original
feature
space



Registered space to make feasible a linear
separation between classes

Thank you.

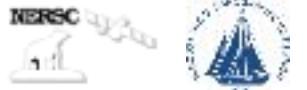
AI Chair OceaniX 2020-2024

Physics-informed AI for Observation-
Driven Ocean AnalytiX

PI: R. Fablet, Prof. IMT Atlantique, Brest

Web: <https://cia-oceanix.github.io/>

**Internship, PhD
and postdoc
opportunities**



References

General Deep Learning references

- Deep Learning Book. Goodfellow et al. Online version <http://www.deeplearningbook.org/>
- Deep Learning with Pytorch. Stevens et al. <https://pytorch.org/assets/deep-learning/Deep-Learning-with-PyTorch.pdf>
- Physics-based deep learning. Huerey et al.

References from our team

- 4DVarNet with applications on L63/L96: <https://arxiv.org/abs/2007.12941>
- Application to mapping, forecasting and optimal sampling: <https://hal.archives-ouvertes.fr/hal-03189218>
- AIS data streams and maritime traffic surveillance, <https://hal.archives-ouvertes.fr/hal-02388260v4>
- Seabird trajectory simulation, <https://www.biorxiv.org/content/10.1101/2021.04.19.438554v1.full>