

Bridging Physics and Learning: application to ocean dynamics

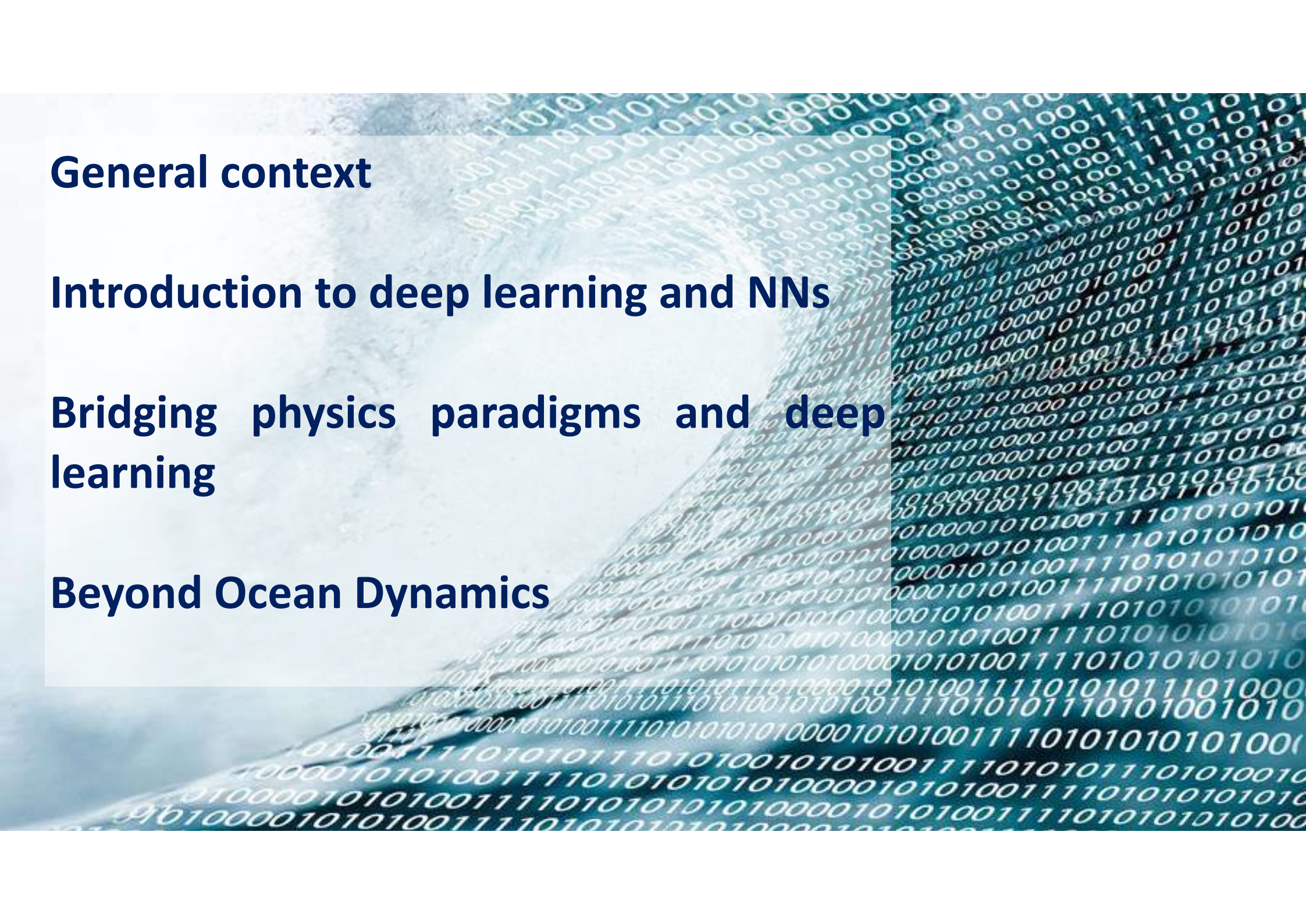
R. Fablet et al.

ronan.fablet@imt-atlantique.fr

web: rfablet.github.io

Webinar IMT Data & AI, July 2020





General context

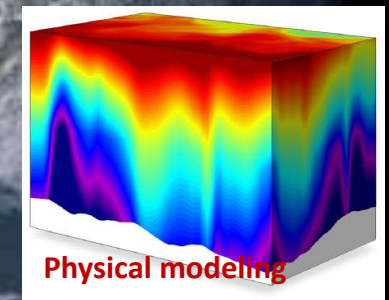
Introduction to deep learning and NNs

Bridging physics paradigms and deep learning

Beyond Ocean Dynamics

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

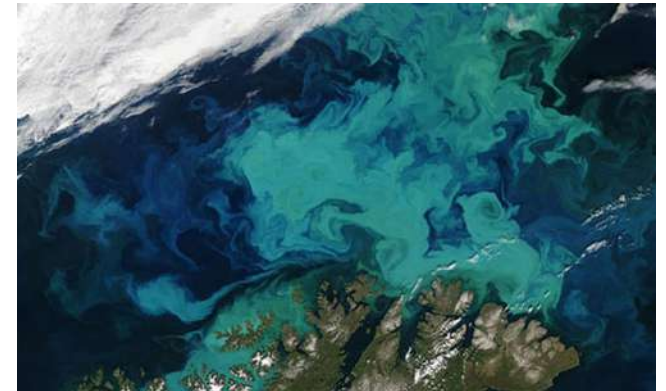
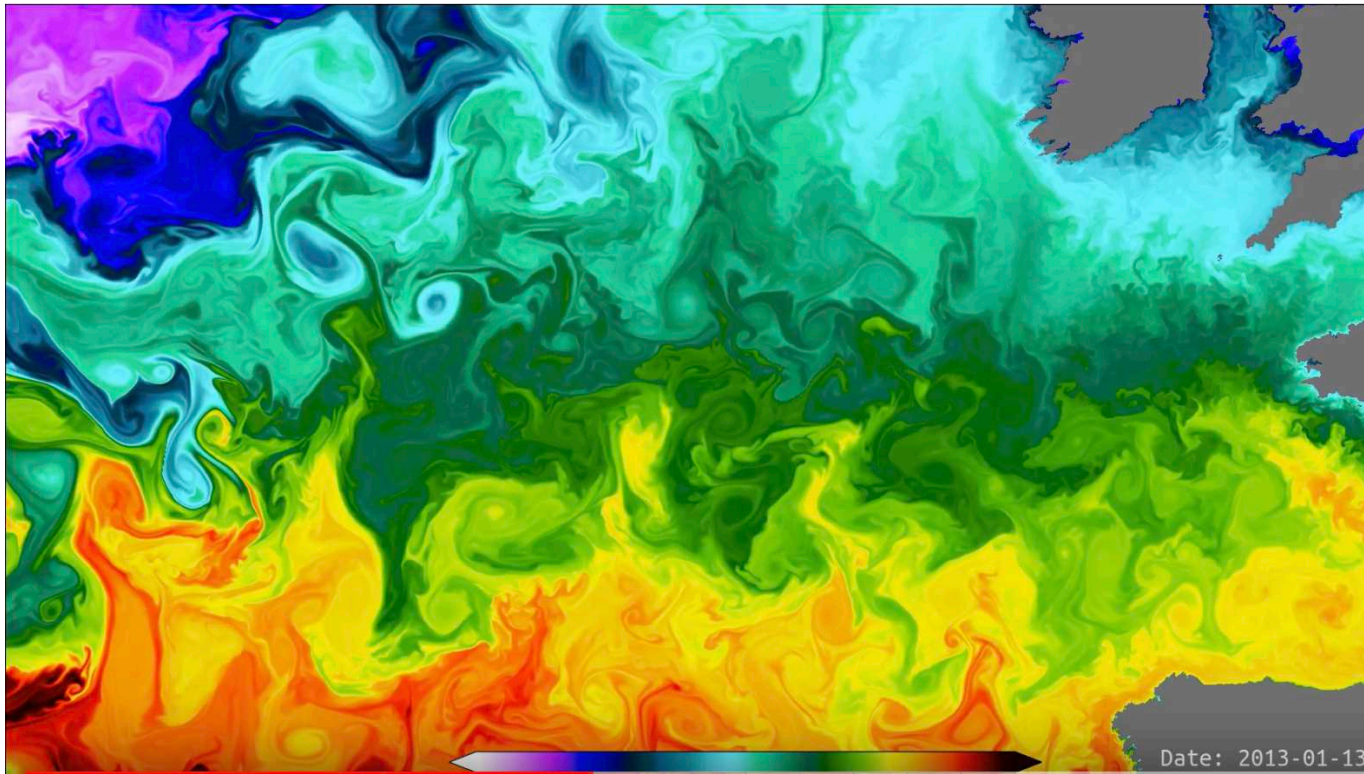
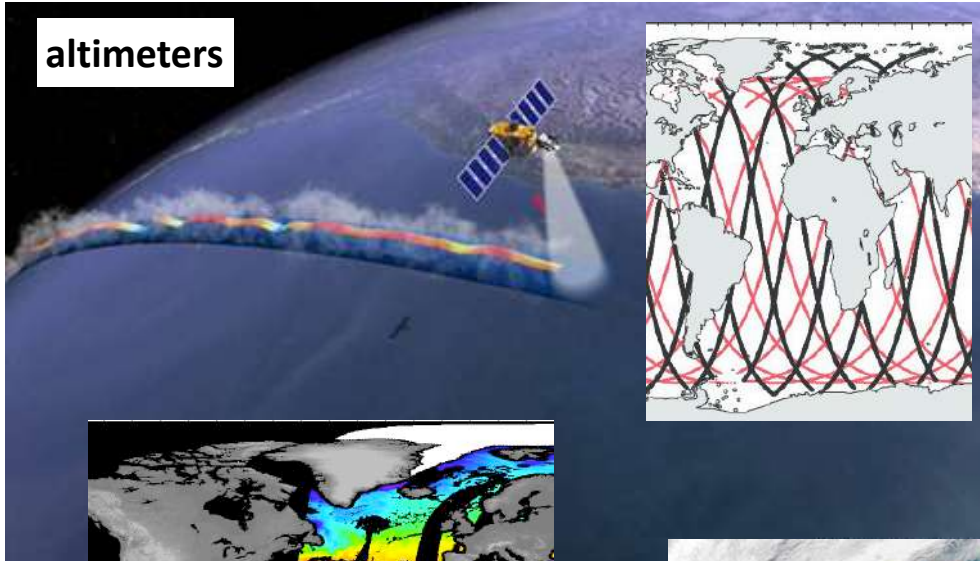
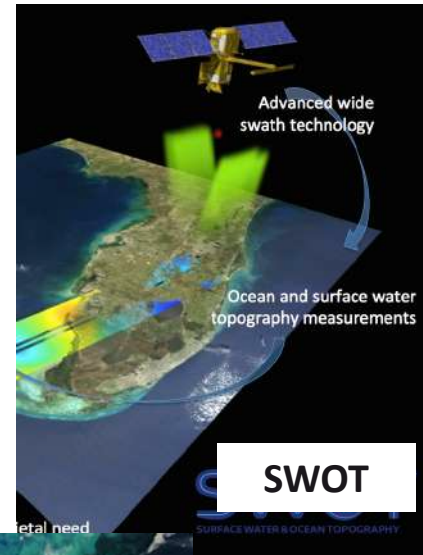


Illustration of satellite-derived sea surface observations

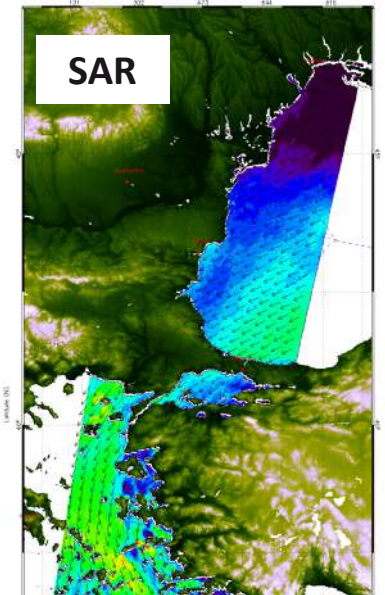
altimeters



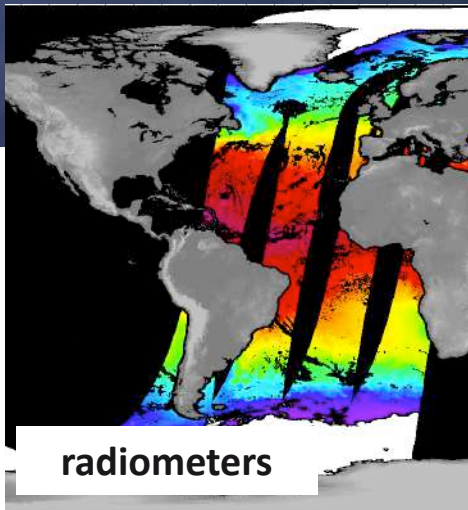
SWOT



SAR



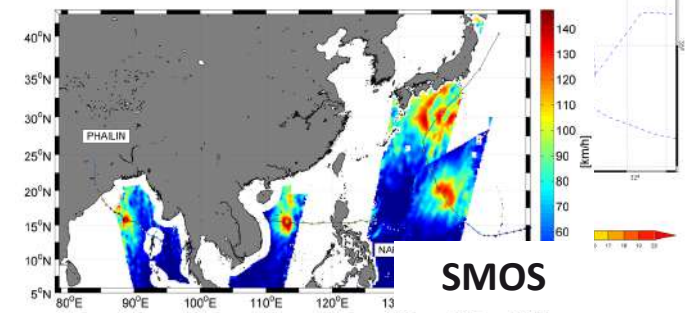
radiometers

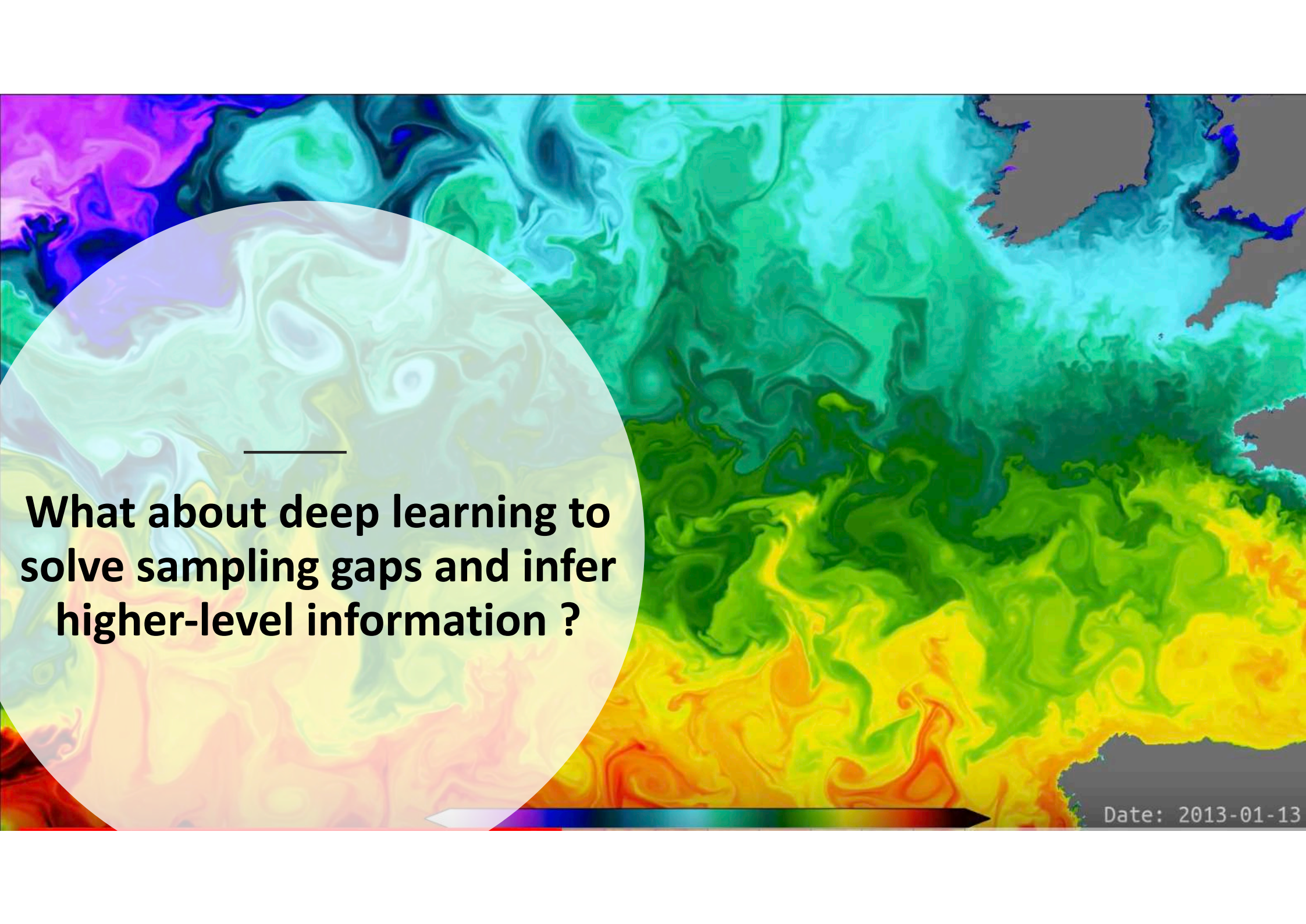


Ocean Colour



SMOS





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

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 REPORTS IN THIS DOCUMENT HAVE BEEN SPONSORED BY THE GEOPHYSICS RESEARCH DIRECTORATE OF THE AIR FORCE CAMBRIDGE RESEARCH CENTER, AIR RESEARCH AND DEVELOPMENT COMMAND, UNDER CONTRACT NO. AF19(604)1566.

Deterministic Nonperiodic Flow^{*}

EDWARD N. LORENZ

Massachusetts Institute of Technology

(Manuscript received 18 November 1962, in revised form January 1963)

ABSTRACT

Finite systems of deterministic ordinary nonlinear differential equations may be designed to represent forced dissipative hydrodynamic flow. 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 ordinarily unstable with respect to small modifications, so that slightly differing initial states can evolve into considerably different states. Systems with bounded solutions are shown to possess bounded numerical solutions. A simple system representing cellular 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 examined in the light of these results.

1. Introduction

Certain hydrodynamical systems exhibit steady-state flow patterns, while others oscillate in a regular periodic fashion. Still others vary in an irregular, seemingly haphazard manner, and, even when observed for long periods of time, do not appear to repeat their previous history.

These modes of behavior may be identified with the familiar rotational, oscillatory, and irregular motions of a cylindrical vortex, a pendulum, and a turbulent flow, respectively. In the case of a finite mass may be represented mathematically as a finite collection of molecules—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 intractable, and the set of molecules is usually approximated by a continuous distribution. The laws are then expressed as partial differential equations, containing such quantities as density, pressure, and velocity.

Lack of periodicity is a characteristic of turbulent flow. Because instantaneous turbulent flow patterns are so irregular, attention is often confined to the statistics of turbulence, which, in contrast to the details of turbulence, often behave in a regular well-organized manner. The short-range weather forecast, however, is forced with difficulty to predict the details of the flow, which continually change in time and space.

Thus there are occasions when more than the statistics of irregular flow are of very real concern.

In this study we shall work with systems of deterministic equations which are idealizations of hydrodynamical systems. We shall be interested principally in nonperiodic solutions, i.e., solutions which never repeat their past history exactly, and where approximate repetitions are of finite duration. These solutions, as we shall see, are associated with behavior that is irregular and nonperiodic. The solutions, as we shall see, are associated with behavior that is irregular and nonperiodic. The solutions, as we shall see, are associated with behavior that is irregular and nonperiodic.

It is sometimes possible to obtain particular solutions of these equations analytically, especially when the solutions are periodic or invariant with time, and, indeed, much work has been devoted to obtaining such solutions by one scheme or another. Ordinarily, however, nonperiodic solutions cannot readily be determined by analytical procedures. Such procedures are usually replaced by numerical procedures, which are often carried out on a computer. The solutions are then expressed in series of orthogonal functions. The governing laws then become a finite set of ordinary differential

^{*} The research reported in this paper was conducted at the Geophysics Research Directorate of the Air Force Cambridge Research Center, under Contract No. AF 19(604)1566.

Learning & Geoscience: Data-driven approaches for data assimilation

OCTOBER 2017

LGUENSAT ET AL.

4093

The Analog Data Assimilation[®]

REDOUANE LGUENSAT AND PIERRE TANDEO

IMT Atlantique, Lab-STICC, Université Bretagne Loire, Brest, France

PIERRE AILLIOT

Laboratoire de Mathématiques de Bretagne Atlantique, University of Western Brittany, Brest, France

MANUEL PULIDO

Department of Physics, Universidad Nacional del Nordeste, and CONICET, Corrientes, Argentina

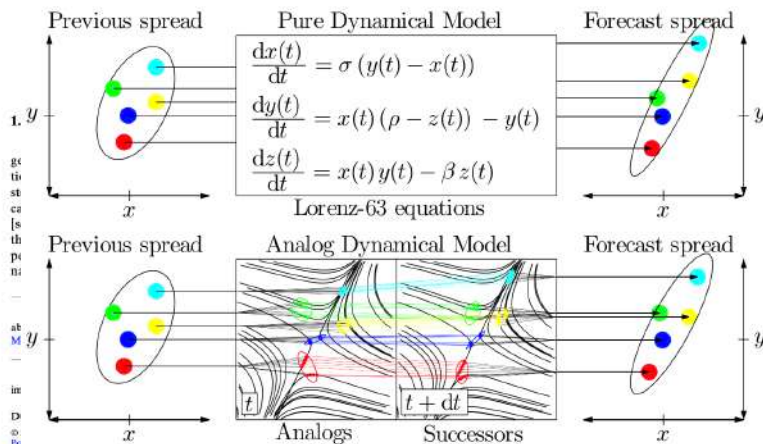
RONAN FABLET

IMT Atlantique, Lab-STICC, Université Bretagne Loire, Brest, France

(Manuscript received 23 November 2016, in final form 31 July 2017)

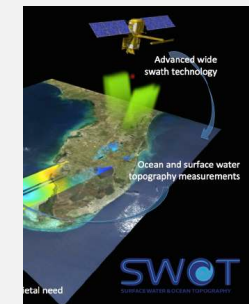
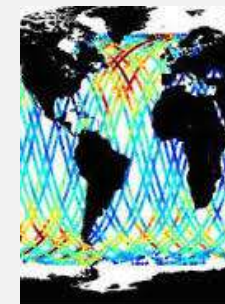
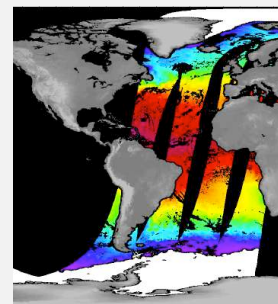
ABSTRACT

In light of growing interest in data-driven methods for oceanic, atmospheric, and climate sciences, this work focuses on the field of data assimilation and presents the analog data assimilation (AnDA). The proposed framework produces a reconstruction of the system dynamics in a fully data-driven manner where no explicit knowledge of the dynamical model is required. Instead, a representative catalog of trajectories of the system is assumed to be available. Based on this catalog, the analog data assimilation combines the nonparametric



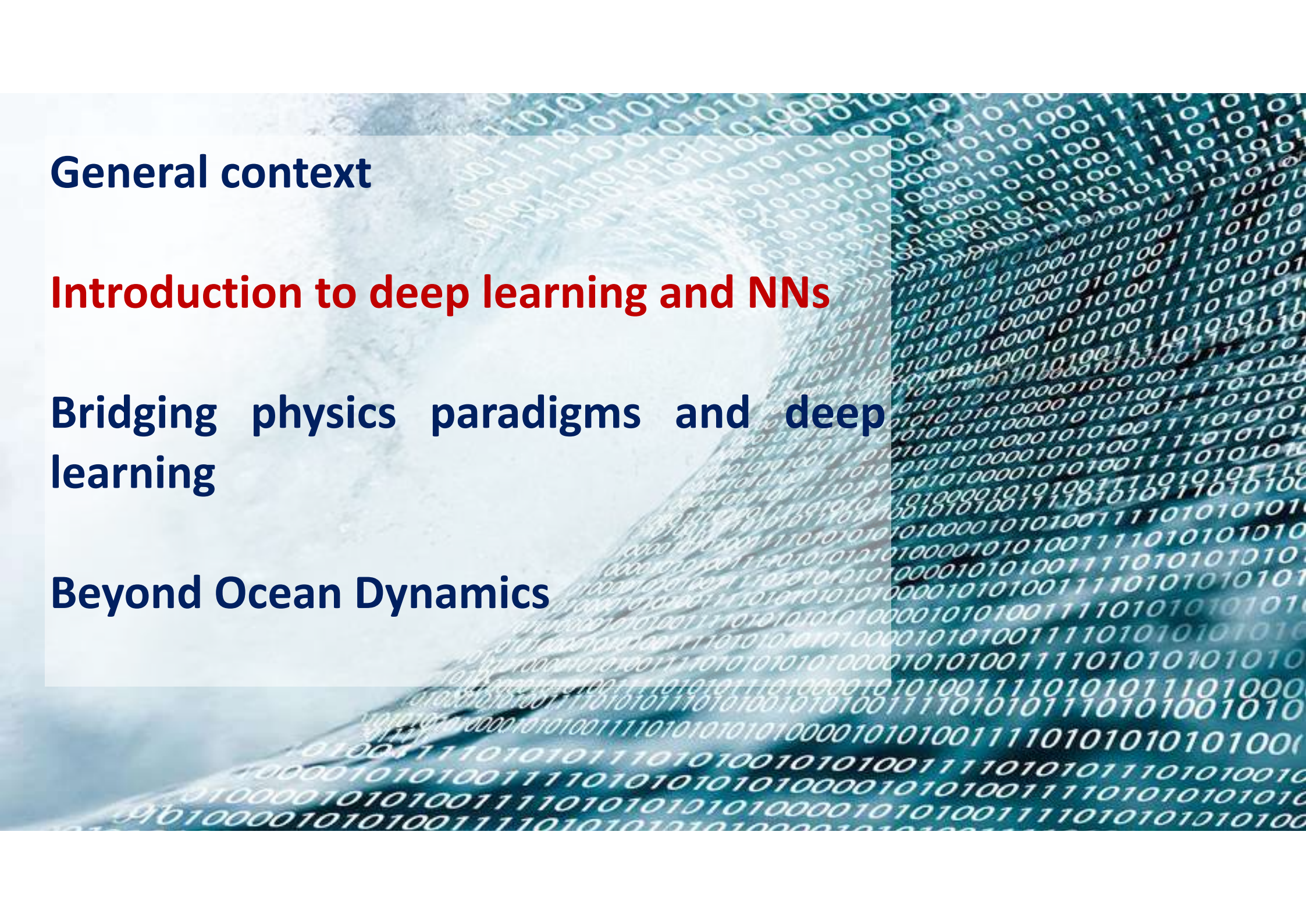
The analog data assimilation [Lguensat et al., 2017]

- 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

Introduction to deep learning and NNs

Bridging physics paradigms and deep learning

Beyond Ocean Dynamics

Introduction to Deep Learning

Resources

- **Book: Deep Learning**

Goodfellow, Bengio, Courville, MIT Press

Online version <http://www.deeplearningbook.org/>

- **Online course by Andrew Ng (Stanford/Baidu)**

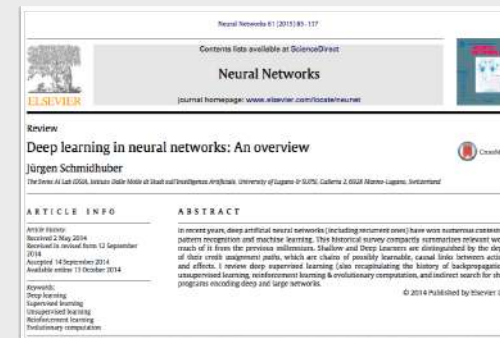
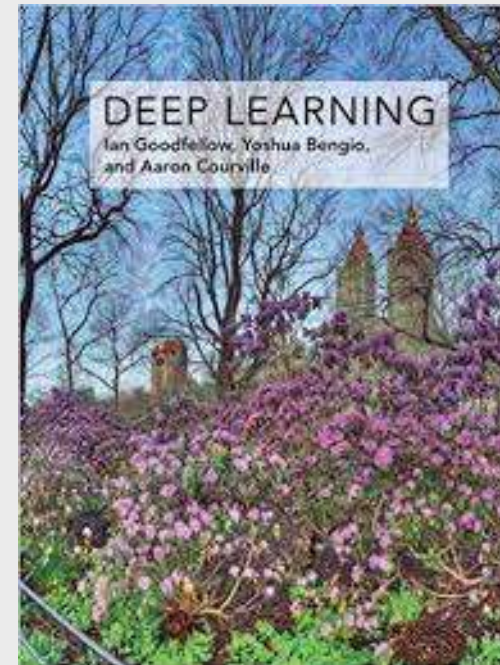
Youtube: [link](#)

Online course on Coursera: [link](#)

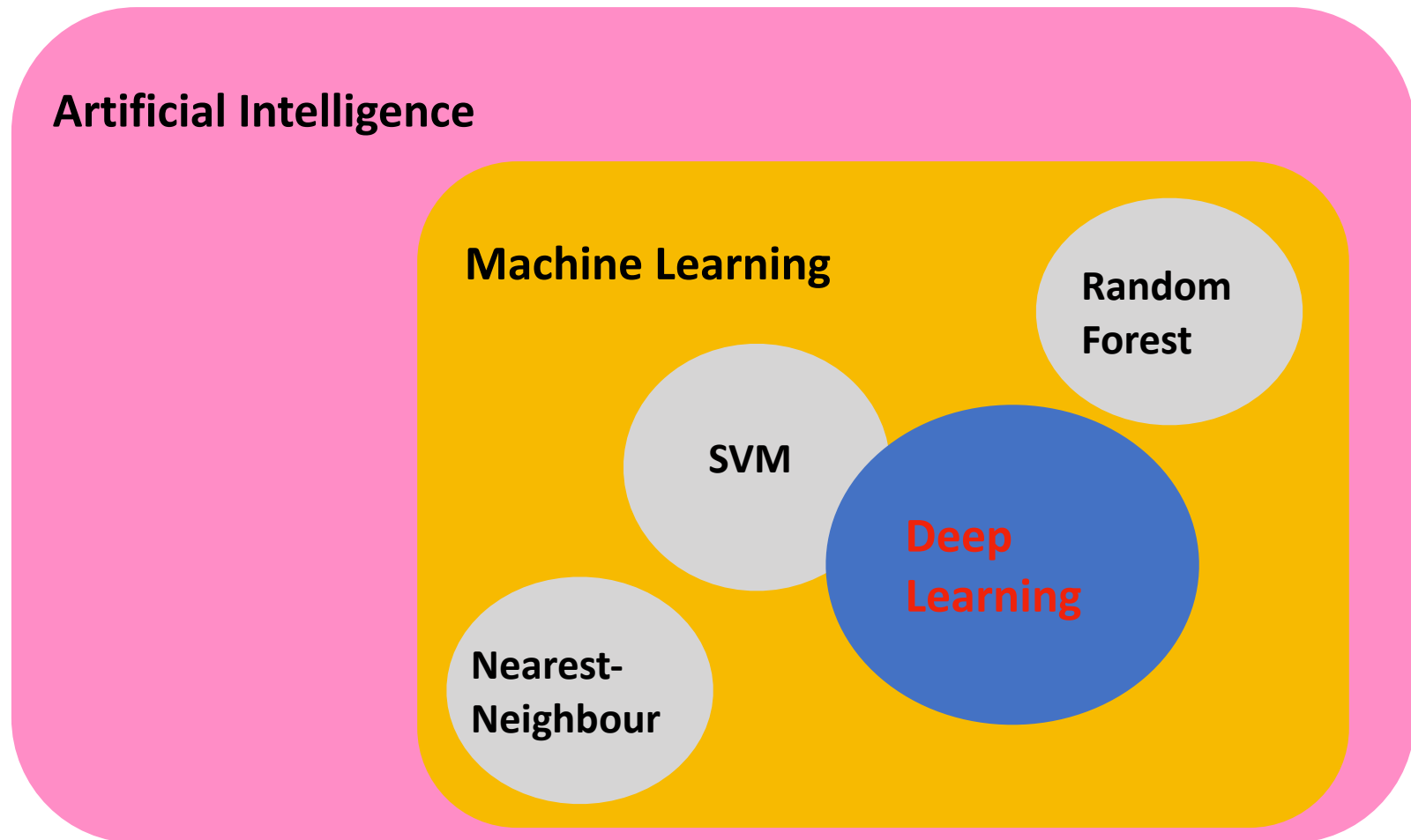
- **Review paper: Deep learning in neural networks by**

J. Schmidhuber

pdf: [link](#)



What's Deep Learning?



From Image to Text (image captioning)



A woman is throwing a **frisbee** in a park.



A **dog** is standing on a hardwood floor.

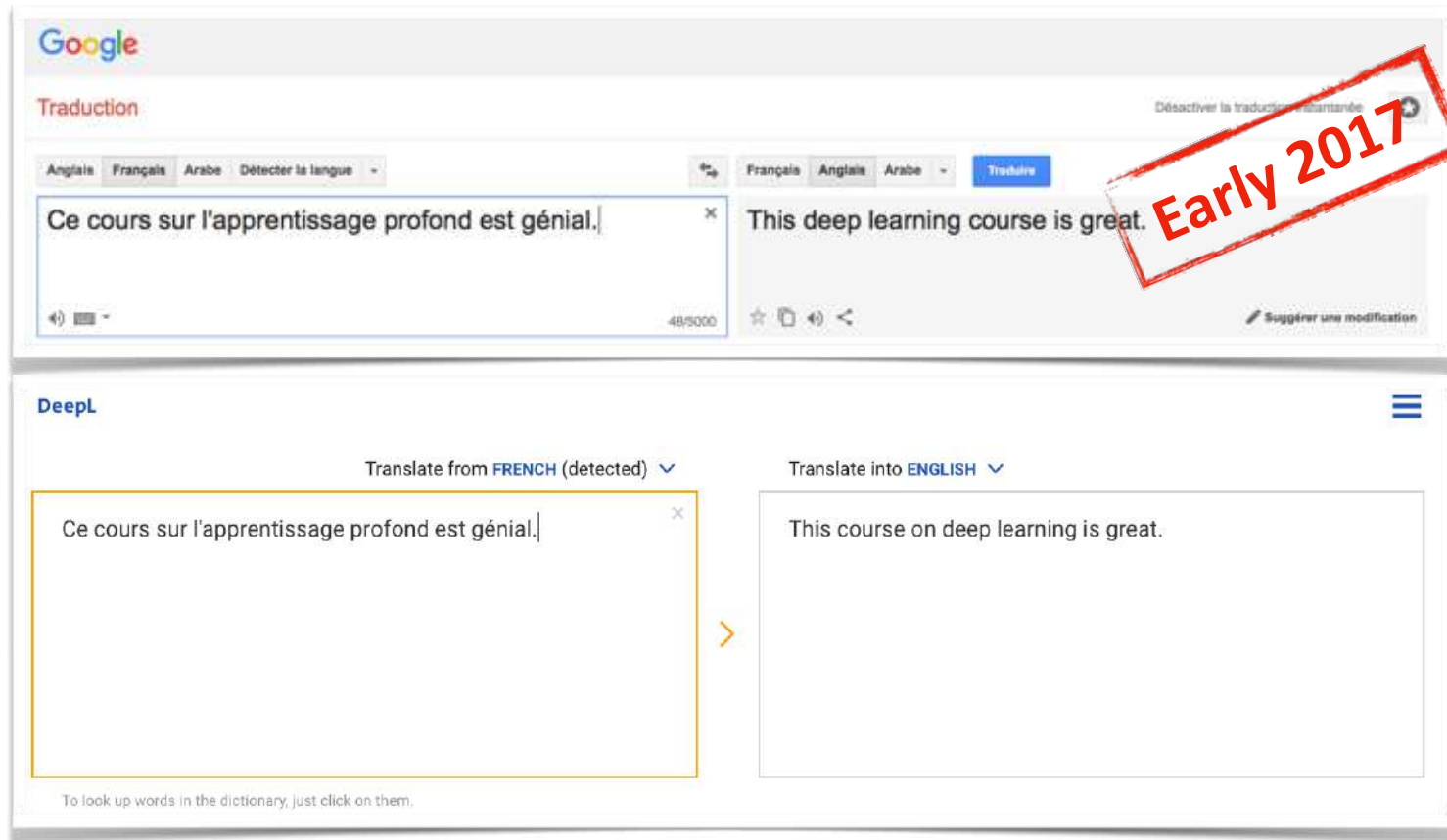


A little **girl** sitting on a bed with a teddy bear.



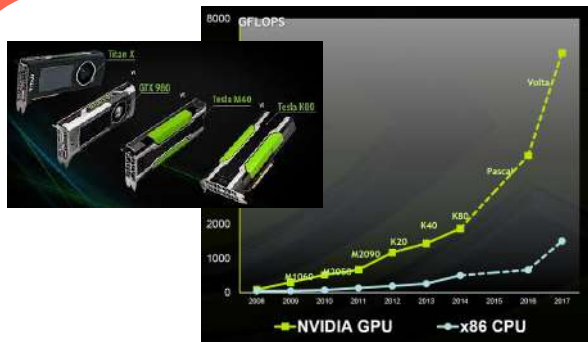
A group of **people** sitting on a boat in the water.

Automatic Translation



and many more.....

Key reasons for the emergence of DL



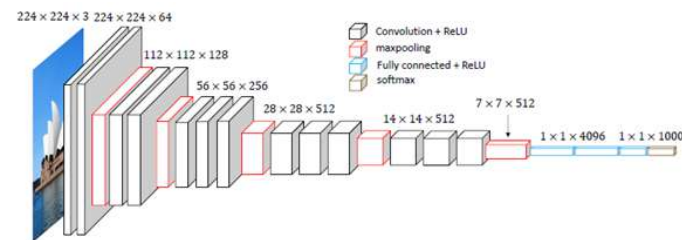
High-performance computing (GPU)



Large annotated dataset (> 1M)

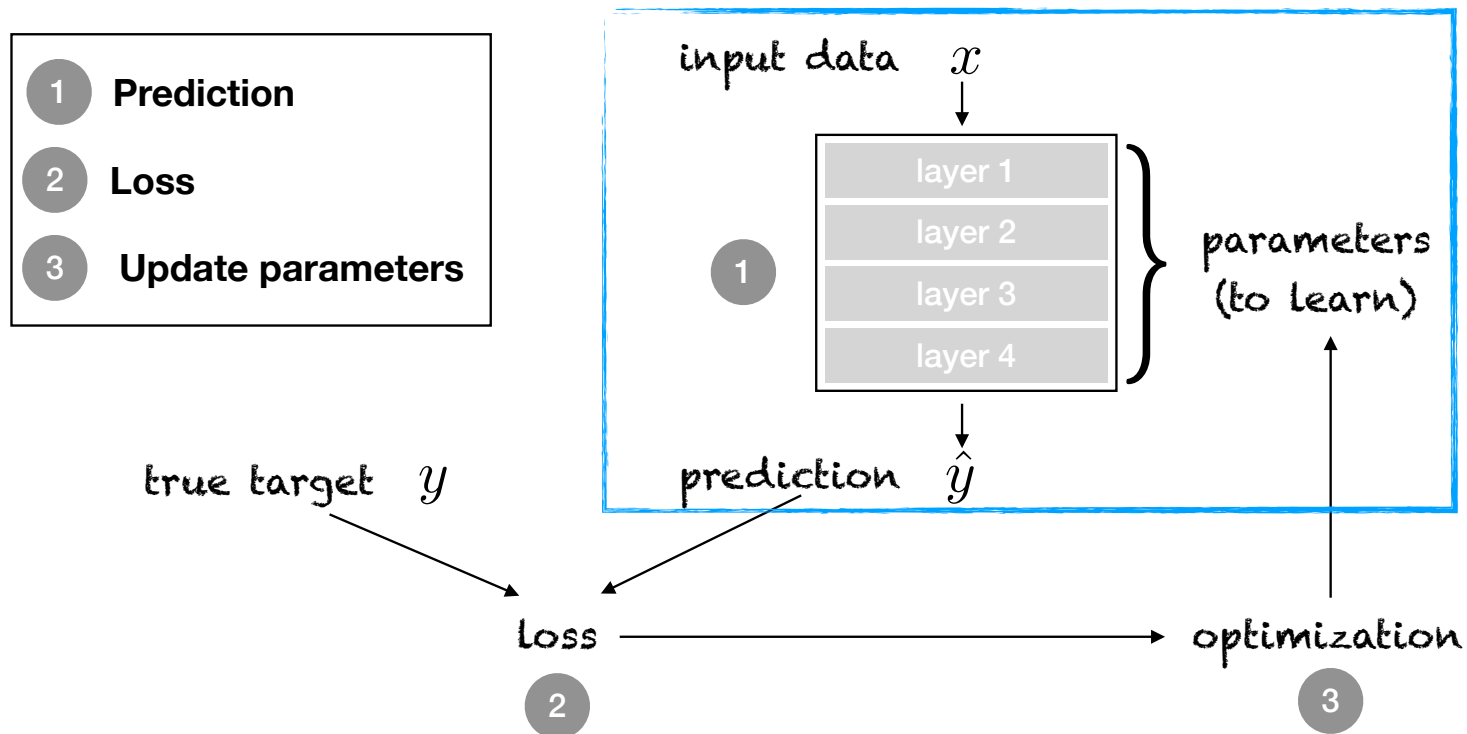


Efficient & easy-to-use frameworks



End-to-end learning

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

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

Key questions:

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

Deep learning models

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:**

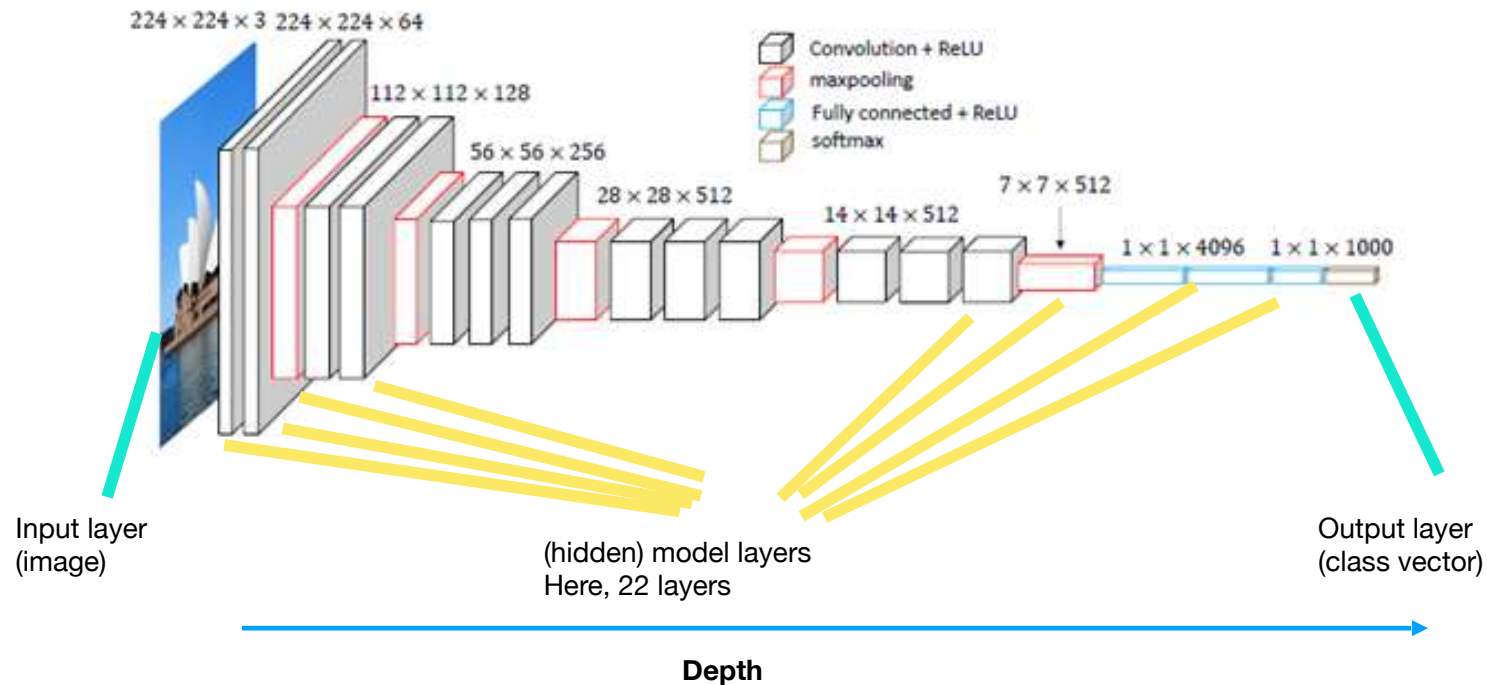
- AD example: https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html
- Basis of the backpropagation algorithm (similar to adjoint method)

- **Resulting gradient descent for learning model parameters:**

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

Deep learning models

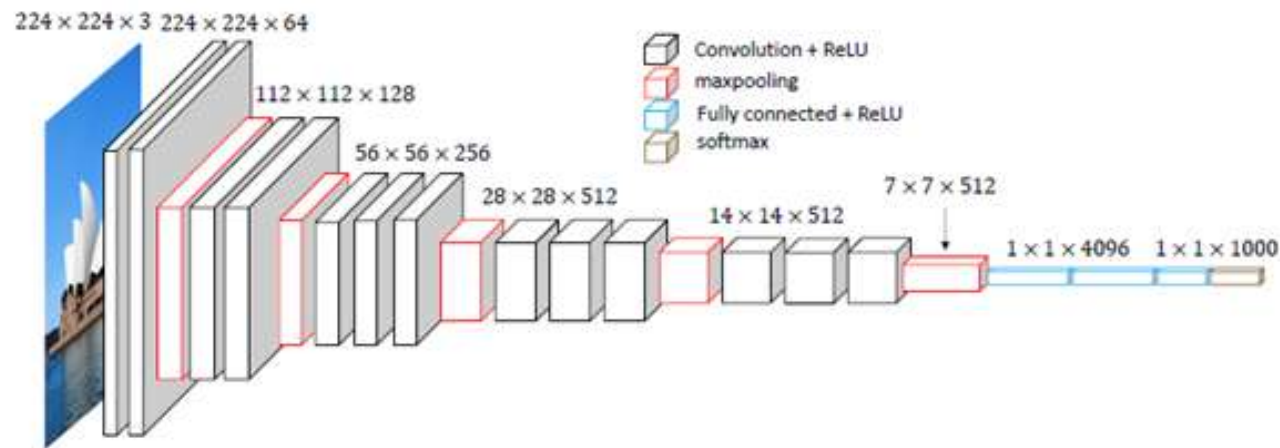
DL models are (in general) feedforward models. VGG16 as an illustration



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

Basics of DL models

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



Elementary
components

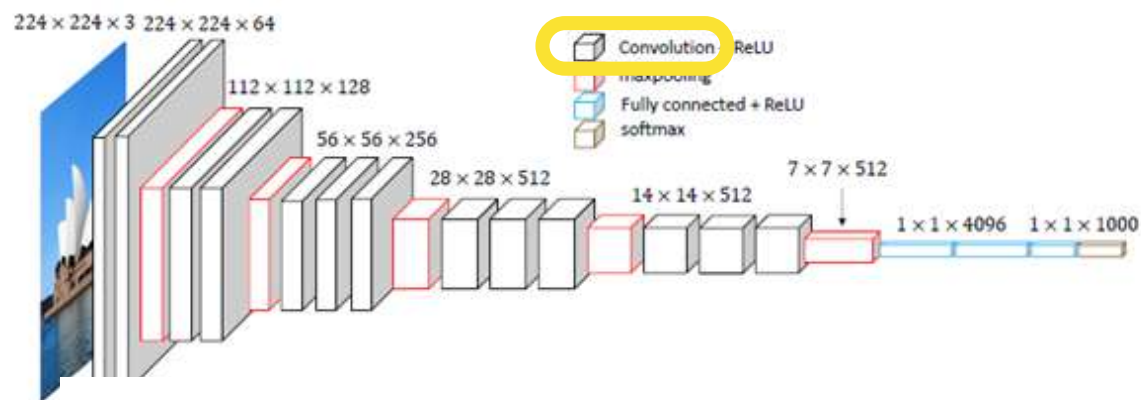
Convolution layers

Activation layers

Pooling layers

Dense layers

Basics of DL models



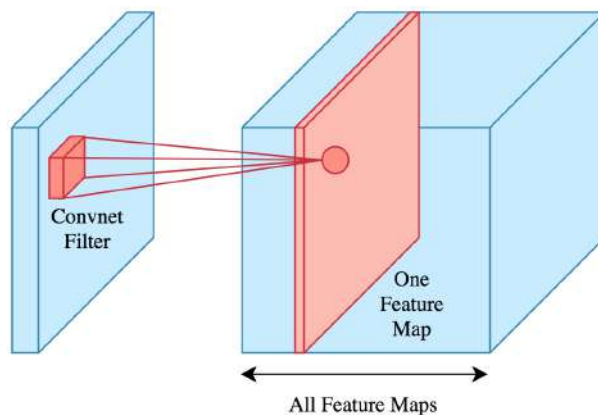
Elementary
components

Convolution layers

Activation layers

Pooling layers

Dense layers

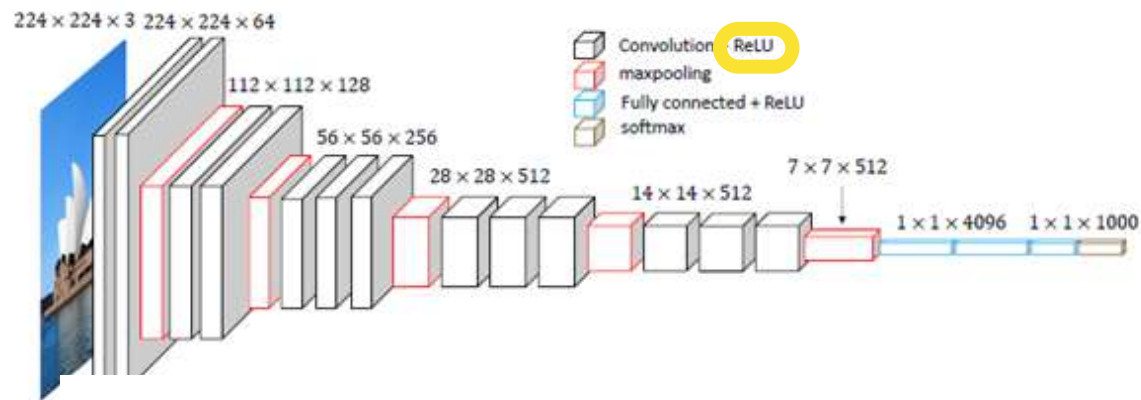


Number of parameters:
Filter size x Number of filters

e.g. $3 \times 3 \times K \times N_{\text{filt}}$

**Independent on the sizes of the input
and output layer**

Basics of DL models



Elementary components

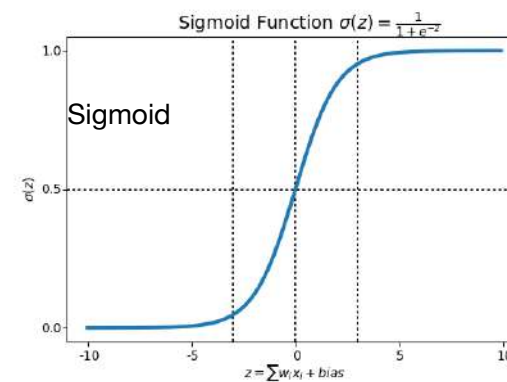
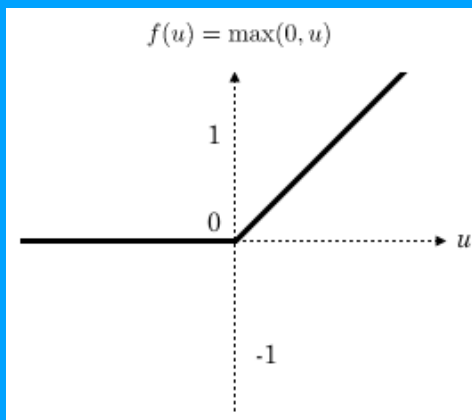
Convolution layers

Activation layers

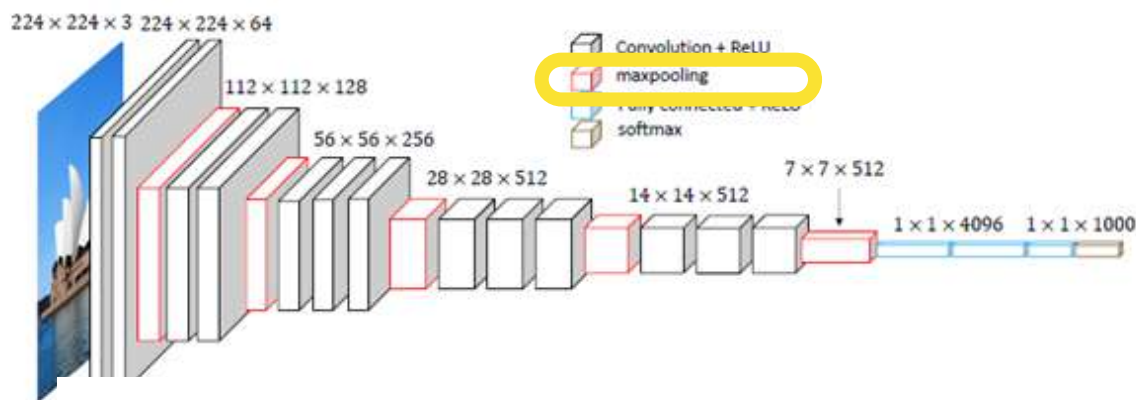
Pooling layers

Dense layers

ReLU
(Rectified Linear Unit)



Basics of DL models



Elementary components

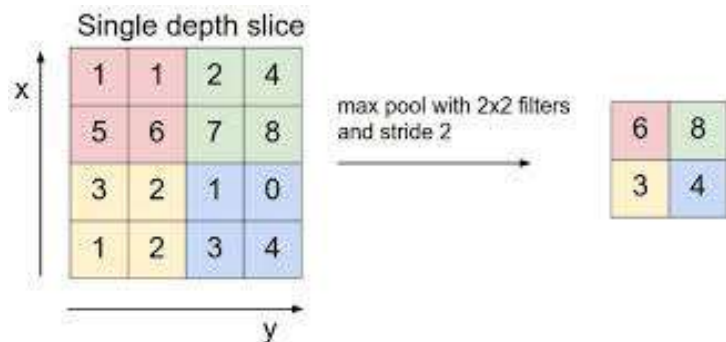
Convolution layers

Activation layers

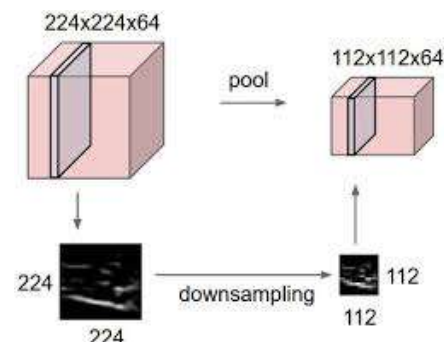
Pooling layers

Dense layers

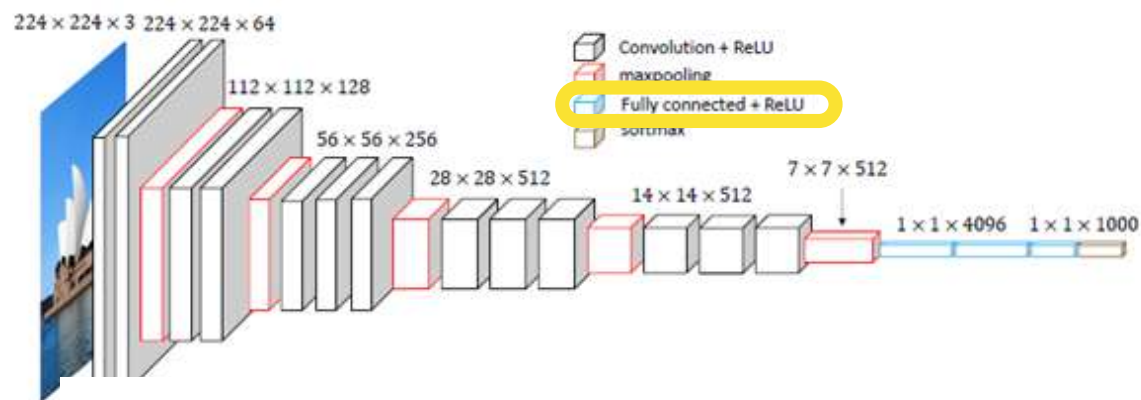
An example of max-pooling operator



Pooling downsamples the input layer



Basics of DL models



Elementary components

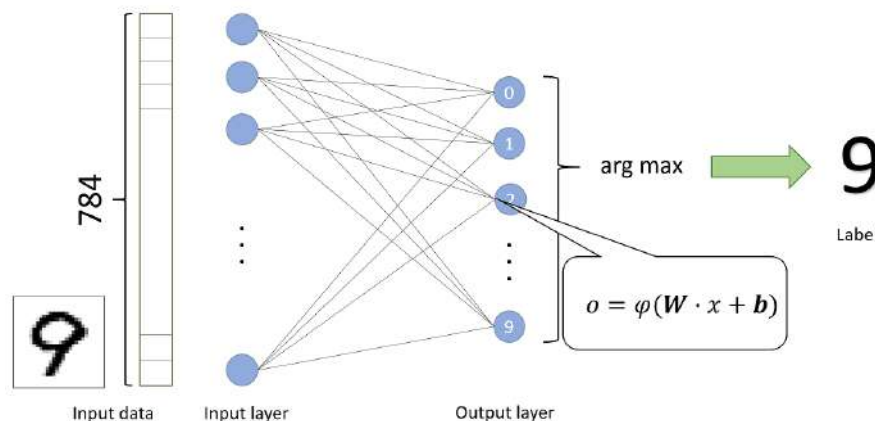
Convolution layers

Activation layers

Pooling layers

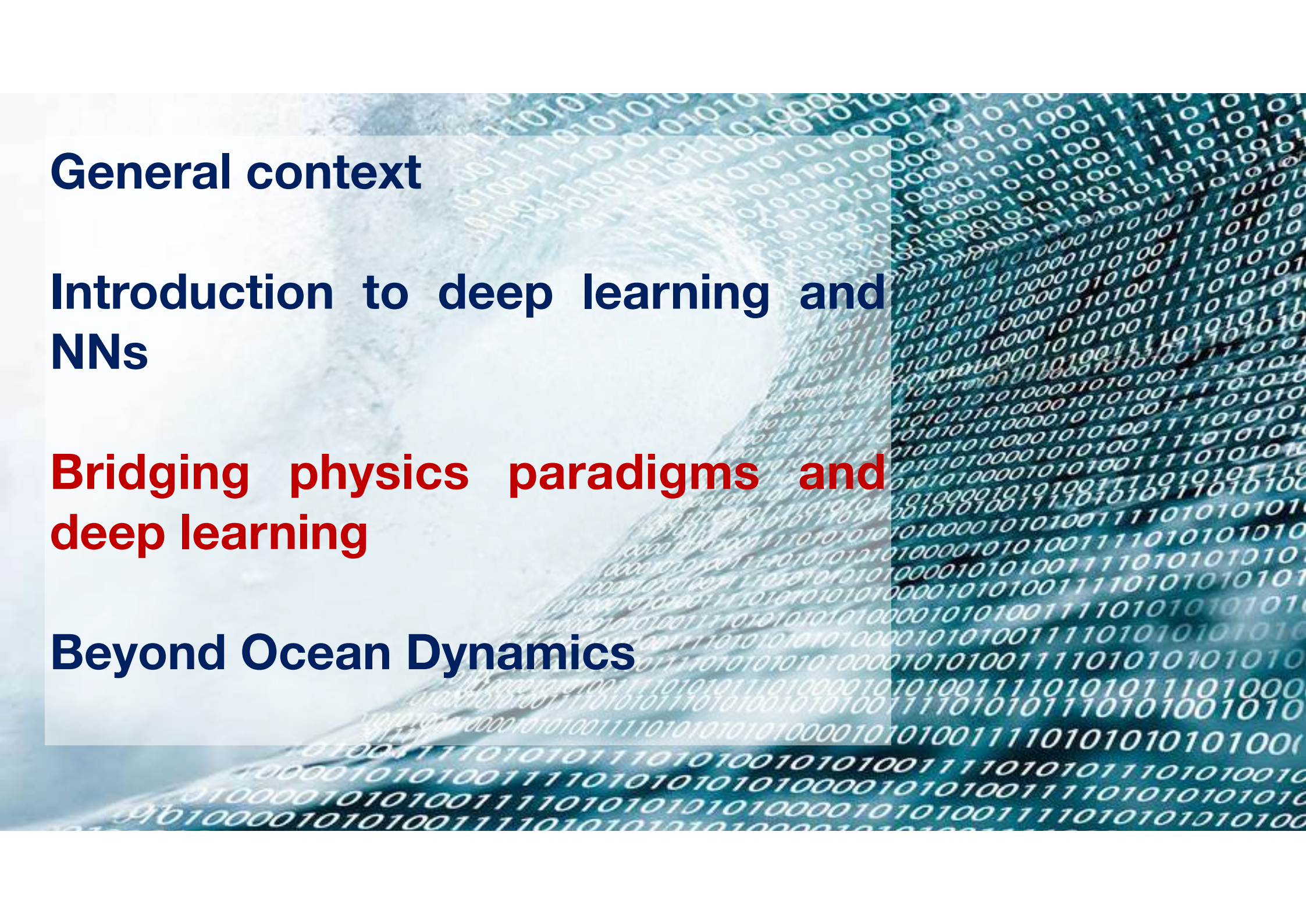
Dense layers

Dense layers
or
Fully-connected (FC) layer
as in a classic MLP



**Let's play with MLPs using
tensorflow playground**

<http://playground.tensorflow.org/>



General context

**Introduction to deep learning and
NNs**

**Bridging physics paradigms and
deep learning**

Beyond Ocean Dynamics

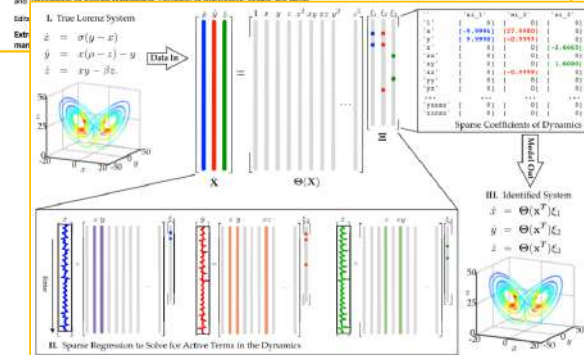
Deep Learning applied to physics

Direct applications of DL schemes to physics-related issues

Discovering governing equations from data by sparse identification of nonlinear dynamical systems

Steven L. Brunton^{1,*}, Joshua L. Proctor², and J. Nathan Kutz³

¹Department of Mechanical Engineering, University of Washington, Seattle, WA 98195; ²Institute for Disease Modeling, Bellevue, WA 98005; and ³Department of Aeronautics, University of Washington, Seattle, WA 98195



Deep learning at scale for the construction of galaxy catalogs in the Dark Energy Survey

Asad Khan^{1,2,3,*}, E.A. Huerta^{4,5}, Sibor Wang¹, Robert Gruendl^{1,2,3}, Elise Jennings¹, Huihuo Zheng¹

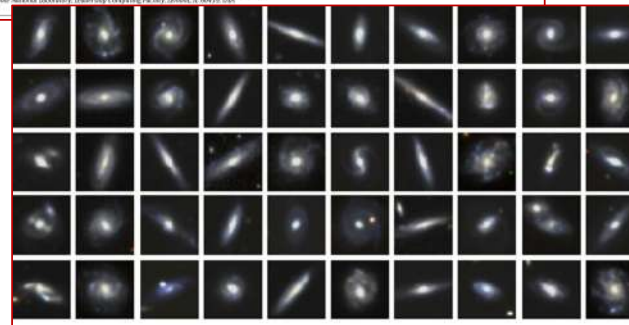
¹National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

²Department of Physics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

³Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

⁴Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

⁵Argonne National Laboratory, Leadership Computing Facility, Lemont, IL 60439, USA



Classification of the global Sentinel-1 SAR vignettes for ocean surface process studies

Chen Wang^{1,2,*}, Pierre Tandoe Nicolas Longepe³, Douglas Van

¹IFREMER, Univ. Brest, CNRS, IRD, Collaborative d'OS

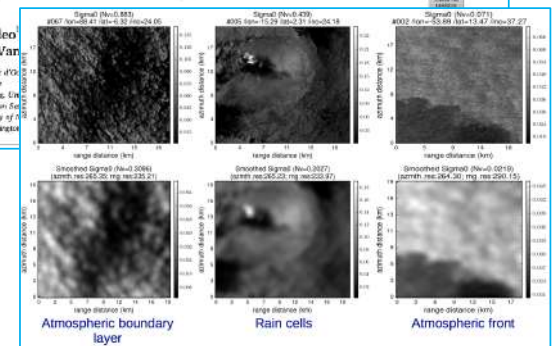
²IMT Atlantique, Lab-STICC, Brest, France

³Department of Coastal Resources and Engineering, Brest

⁴Space and Ground Segment, Calsonic Development, Brest

⁵Ocean Processus Analysis Laboratory, University of

⁶Applied Physics Laboratory, University of Washington



Deep learning to represent subgrid processes in climate models

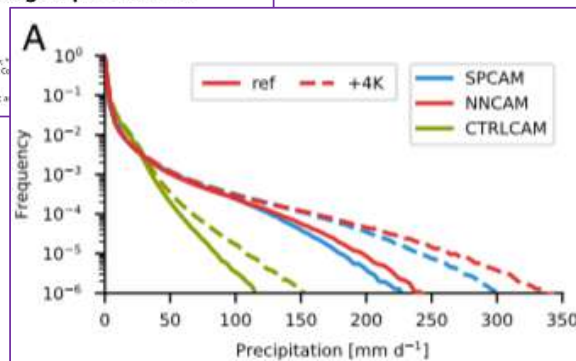
Stephan Rasp^{1,2,*}, Michael S. Pritchard³, and Pierre Gentine^{1,2}

¹Meteorological Institute, Ludwig-Maximilians-University, 80333 Munich, Germany; ²CA 93097; ³Department of Earth and Environmental Engineering, Earth Institute, Co

Columbia University, New York, NY 10027

Edited by Isaac M. Held, Geophysical Fluid Dynamics Laboratory, National Oceanic and

2018 (received for review June 16, 2018)



FILTERING INTERNAL TIDES FROM WIDE-SWATH ALTIMETER DATA USING CONVOLUTIONAL NEURAL NETWORKS

Redouane Lguensat¹, Ronan Fabre², Julien Le Sommer³, Sammy Metref¹, Emmanuel Cosme¹, Kaouther Ouenniche², Lucas Drumetz², Jonathan Gula³

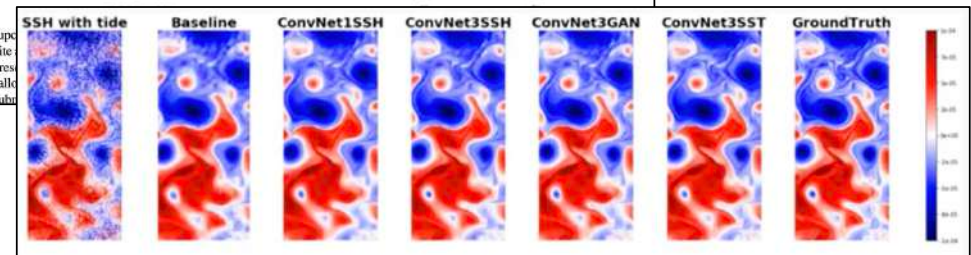
¹ Université Grenoble Alpes, CNRS, IRD, Grenoble INP, IGE, Grenoble, France

² IMT Atlantique, LabSTICC, Université Bretagne Loire, Brest, France

³ Ifremer, LOPS, Brest, France

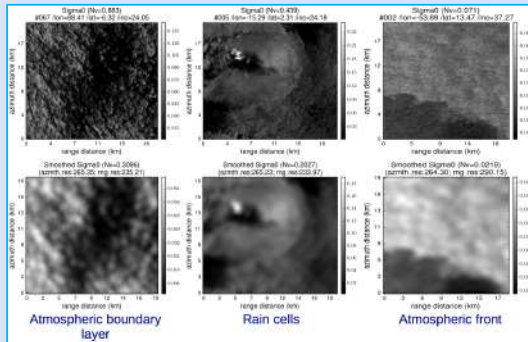
ph] 3 May 2020

The upc
satellite
high-res
thus alle
and sub



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

1 Build a groundtruthed dataset



2 Design or choose an architecture



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

3 Choose a DL framework

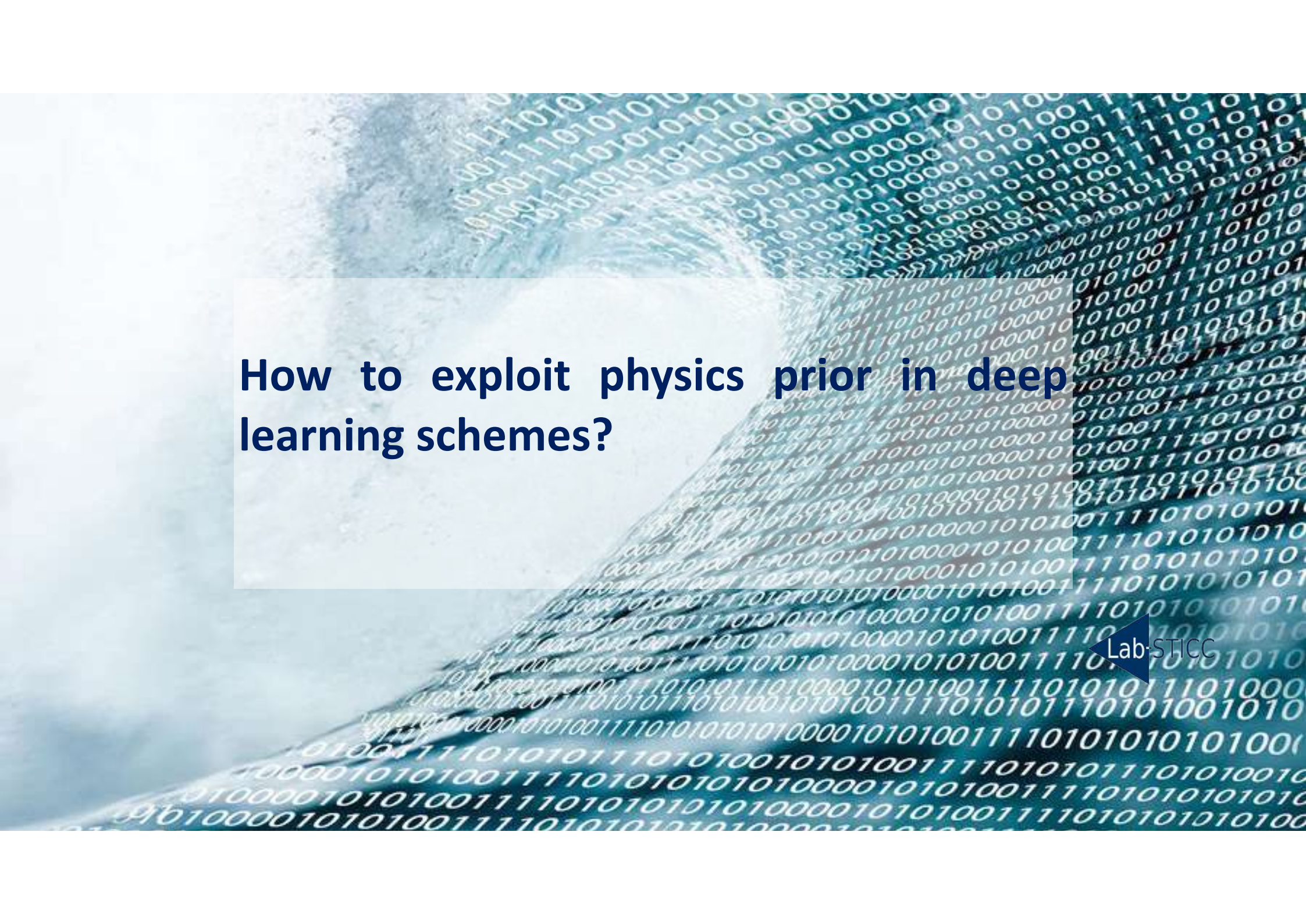


4 Learn the model

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](https://www.tensorflow.org/tutorials/keras/classification)



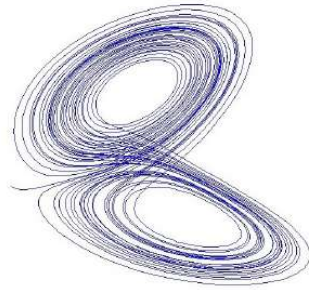
How to exploit physics prior in deep learning schemes?

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



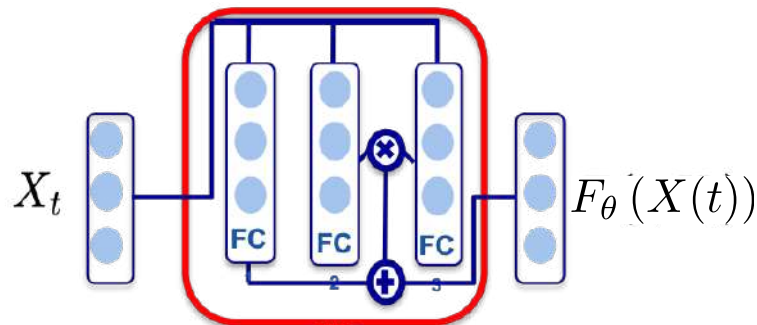
Associated Euler integration scheme

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



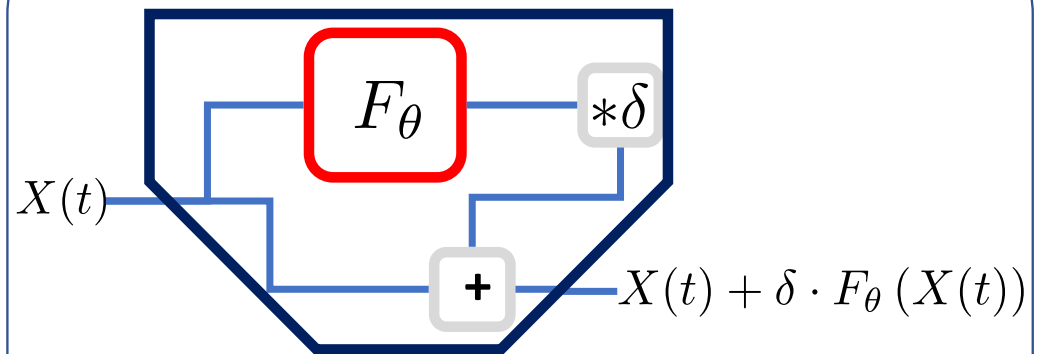
$$X(t + \delta) = X(t) + \delta \cdot F_\theta (X(t))$$

NN architecture for differential operator



Bilinear architecture

NN architecture for integration scheme



ResNet architecture (Residual Network)

How to embed physics-driven priors in DL models ?

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

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

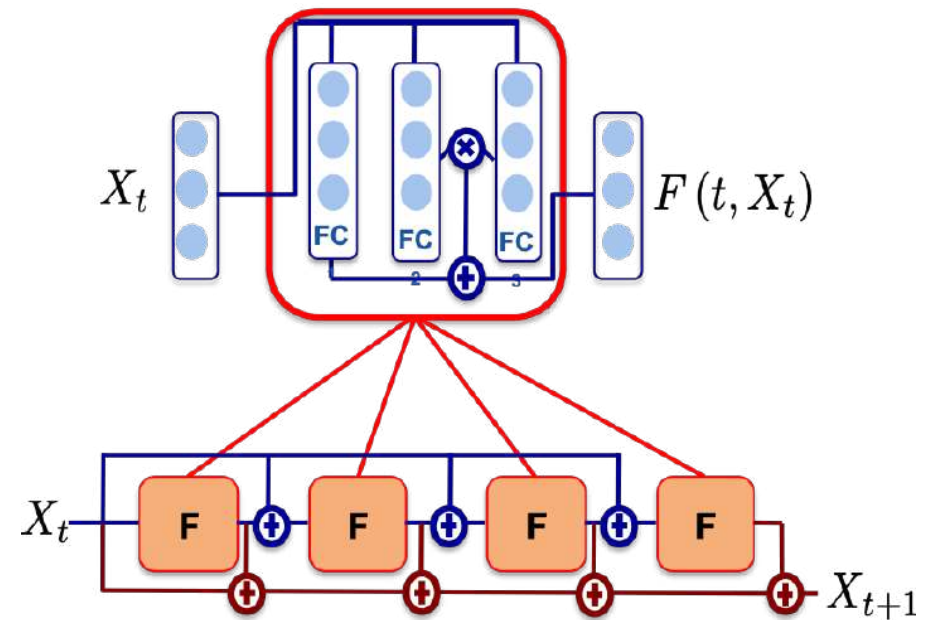
Lorenz-63 equations

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

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

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

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)

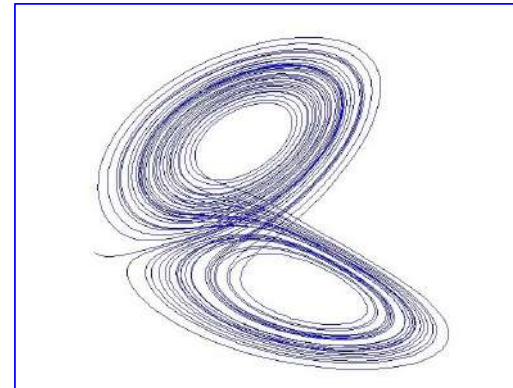
Forecasting experiments

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>	<i>$<10^{-6}$</i>	<i>$<10^{-6}$</i>	<i>$<10^{-6}$</i>

Noisy training data ($\sigma=0.5$)

Forecasting time step	t_0+h	t_0+4h	t_0+8h
Analog forecasting	$<10^{-6}$	2.01	2.2
<i>Bi-NN(4)</i>	<i>$<10^{-6}$</i>	<i>0.054</i>	<i>0.14</i>



Assimilation experiment

(1 obs. every 8 time steps)

Noise standard deviation in training data	0	0.25	1
<u>True model</u>	<u>0.50</u>	-	-
Analog forecasting	0.65	1.17	1.81
<i>Bi-NN(4)</i>	<i>0.60</i>	<i>0.75</i>	<i>0.86</i>

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



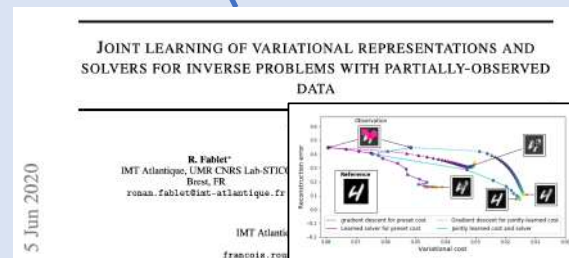
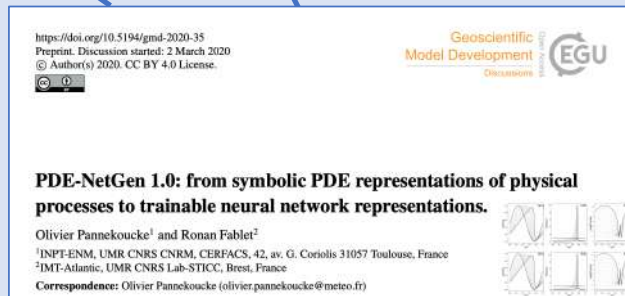
Making the most of AI and Physics Theory

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

Bridging physics & AI: a broader picture

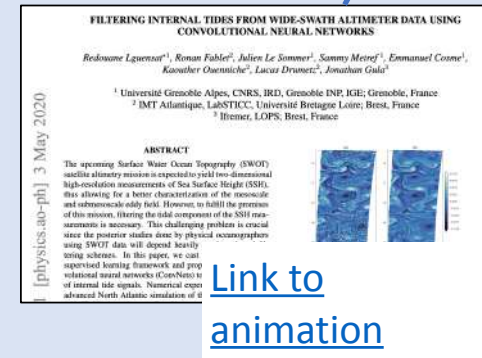
Model-driven

ODEs/PDEs
Variational models
State-space models
Data Assimilation
.....



Learning Latent Dynamics for Partially-Observed Chaotic Systems

Said Ouala¹, Duong Nguyen¹, Lucas Drumetz², Bertrand Chapron³
Ananda Pascual³, Fabrice Collard⁴, Lucile Gaultier⁴, Ronan Fablet²
(1) IMT Atlantique, Lab-STICC, Brest, France
(2) Ifremer, LOPS, Brest, France
(3) IMEDEA, UIB-CSIC, Esporles, Spain
(4) ODL, Brest, France
{said.ouala, duong.nguyen, lucas.drumetz, ronan.fablet}@imt-atlantique.fr
Bertrand.Chapron@ifremer.fr, ananda.pascual@imedea.uib-csic.es
{dr.fab, lucile.gaultier}@oceandatalab.com



Data-driven

Analog methods
Kernel methods
Neural Networks
.....

Physics-informed & Data-constrained

Data-driven & Physics-aware

Bridging physics & AI: Expected breakthroughs



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)

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)**

ResNet

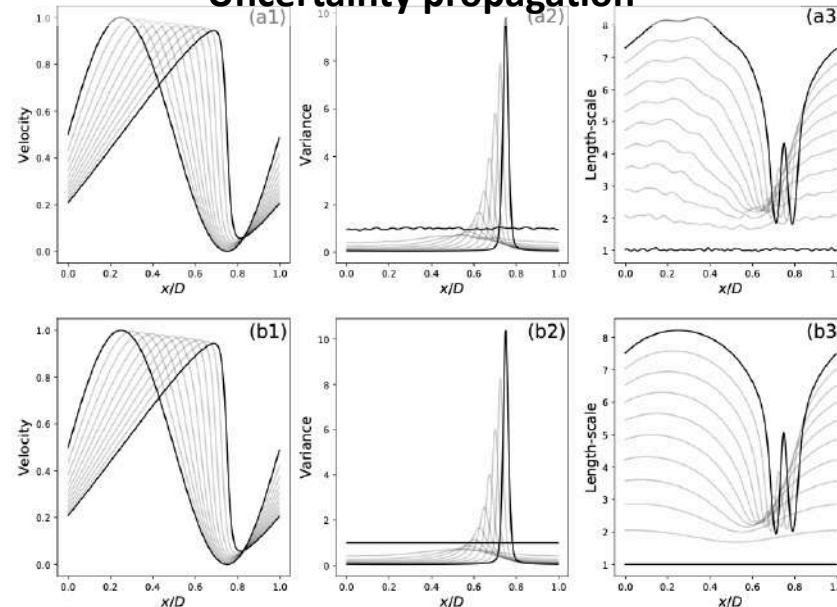
$$\begin{pmatrix} \mu_u(t) \\ \Sigma_u(t) \end{pmatrix} \rightarrow \begin{pmatrix} \mu_u(t+1) \\ \Sigma_u(t+1) \end{pmatrix}$$

Generated code

```
# Example of computation of a derivative
kernel_Du_x_o1 = np.asarray([[0.0, 0.0, 0.0, 0.0],
                             [0.0, 0.0, 0.0, 0.0],
                             [0.0, 1/(2*self.dx[self.coordinates.index('x')]), 0.0]], dtype=float).reshape((3, 3)+(1,1))
Du_x_o1 = DerivativeFactory((3, 3), kernel=kernel_Du_x_o1, name='Du_x_o1')(u)

# Computation of trend_u
mul_1 = keras.layers.multiply([Dkappa_11_x_o1, Du_x_o1], name='MulLayer_1')
mul_2 = keras.layers.multiply([Dkappa_12_x_o1, Du_y_o1], name='MulLayer_2')
mul_3 = keras.layers.multiply([Dkappa_12_y_o1, Du_x_o1], name='MulLayer_3')
mul_4 = keras.layers.multiply([Dkappa_22_y_o1, Du_y_o1], name='MulLayer_4')
mul_5 = keras.layers.multiply([Du_x_o2, kappa_11], name='MulLayer_5')
mul_6 = keras.layers.multiply([Du_y_o2, kappa_22], name='MulLayer_6')
mul_7 = keras.layers.multiply([Du_x_o1_y_o1, kappa_12], name='MulLayer_7')
sc_mul_1 = keras.layers.Lambda(lambda x: 2.0*x, name='ScalarMulLayer_1')(mul_7)
trend_u = mul_1 + mul_2 + mul_3 + mul_4 + mul_5 + mul_6 + sc_mul_1
```

Uncertainty propagation



**Ensemble-based
prediction**

NN prediction

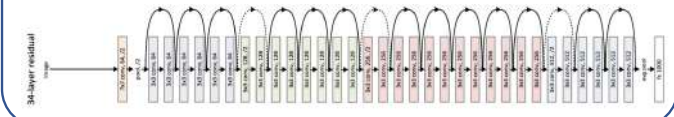
Bridging physics & AI: Expected breakthroughs

Physical model

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

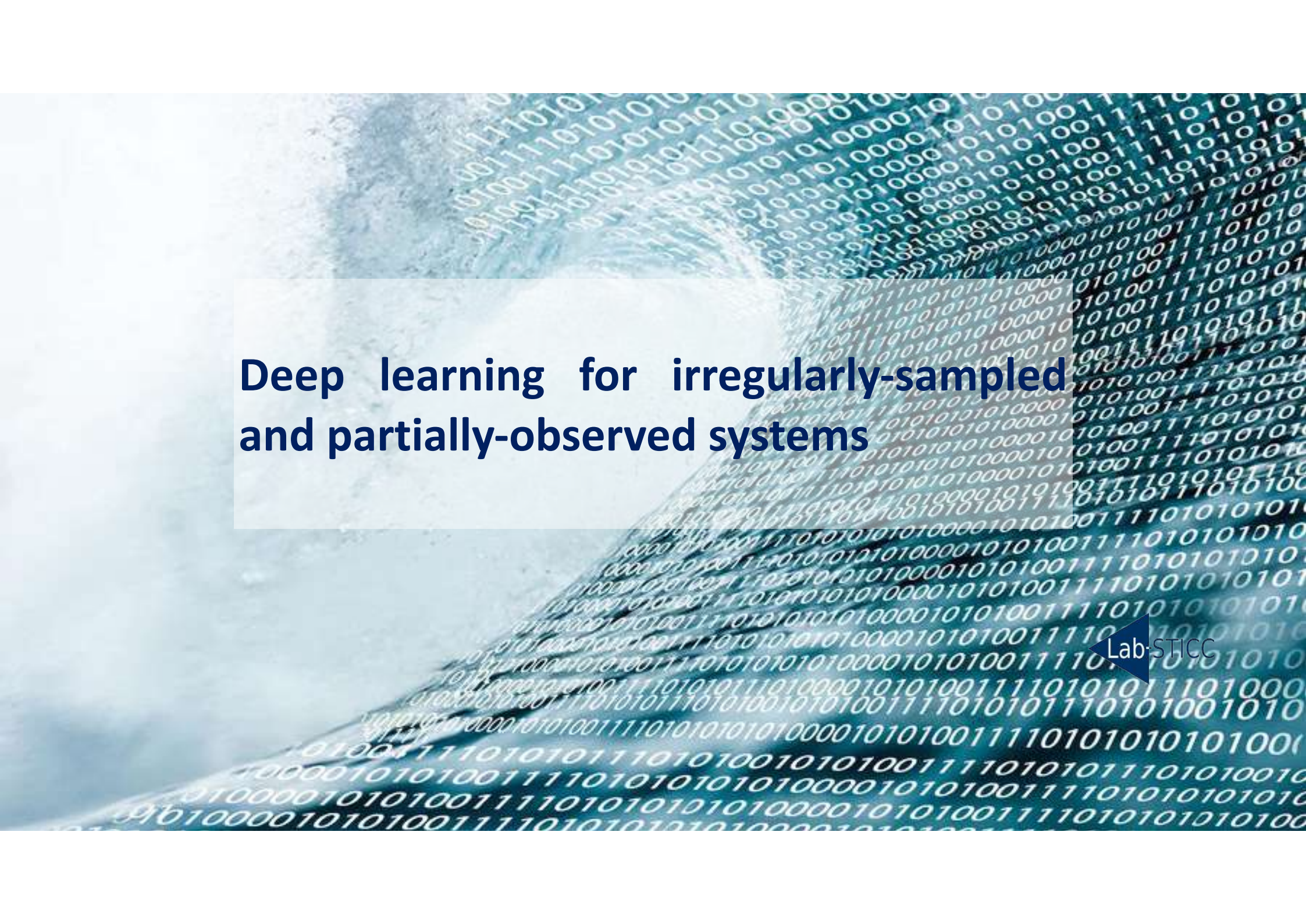
**Representation
learning**

Data-driven representation



Making the most of AI and Physics Theory

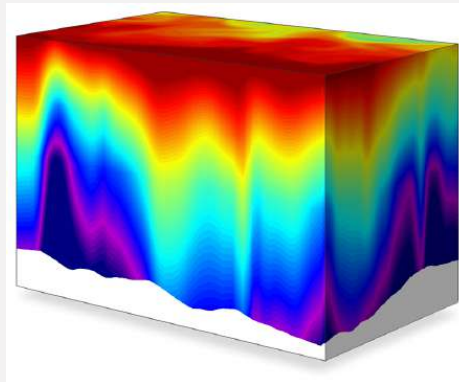
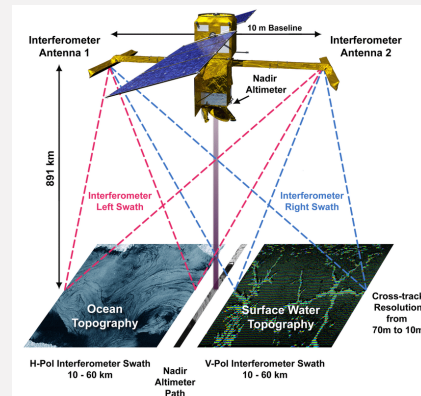
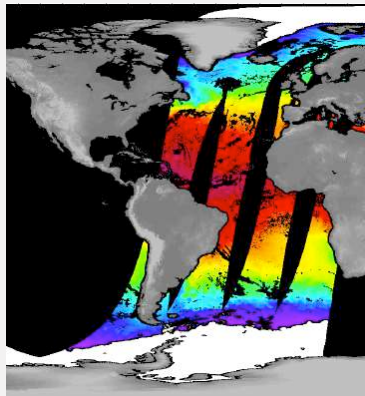
- Model-Driven/Theory-Guided & Data-Constrained schemes
- Data-Driven & Physically-Sound schemes (eg, Ouala et al., 2019)

The background of the slide features a stylized representation of binary code (0s and 1s) in a light blue and white color scheme. The code is arranged in a way that suggests it is flowing or being projected, with some parts appearing more prominent than others. The overall effect is a digital, high-tech aesthetic. A semi-transparent white rectangular box is centered on the slide, containing the title text.

Deep learning for irregularly-sampled and partially-observed systems

learning for partially-observed systems / from irregularly-sampled data [Ouala et al., 2020; Fablet et al., 2020]

Can we learn directly from observation data ?



Generic issue:
Joint identification and inversion

Dynamical model

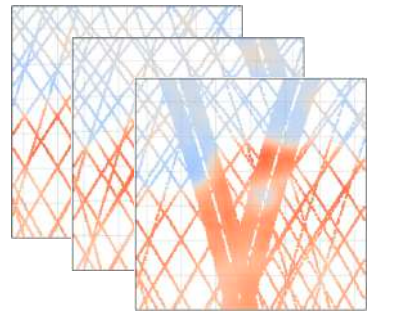
$$X_t \rightarrow \partial_t X = F(X, \xi, t, \theta) \rightarrow X_{t+1}$$

+

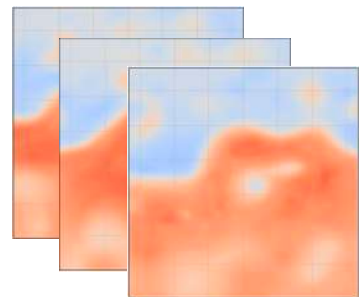
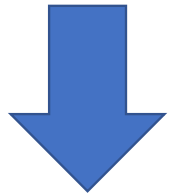
Observation model

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

(Weak constraint) 4DVar Data Assimilation (DA) formulation



Partial observations y



True states x

State-space formulation:

$$\begin{cases} \frac{\partial x(t)}{\partial t} = \mathcal{M}(x(t)) \\ y(t) = x(t) + \epsilon(t), \forall t \in \{t_0, t_0 + \Delta t, \dots, t_0 + N\Delta t\} \end{cases}$$

Associated variational formulation:

$$\arg \min_x \lambda_1 \sum_i \|x(t_i) - y(t_i)\|_{\Omega_{t_i}}^2 + \lambda_2 \sum_n \|x(t_i) - \Phi(x)(t_i)\|^2$$

$$\text{with } \Phi(x)(t) = x(t - \Delta) + \int_{t-\Delta}^t \mathcal{M}(x(u)) du$$

➔ $\boxed{\arg \min_x \lambda_1 \|x - y\|_{\Omega}^2 + \lambda_2 \|x - \Phi(x)\|^2}$

4DVar Data Assimilation (DA) formulation

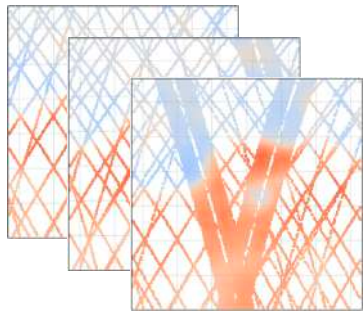
Model-driven schemes: $\arg \min_x \underbrace{\lambda_1 \|x - y\|_\Omega^2 + \lambda_2 \|x - \Phi(x)\|^2}_{U_\Phi(x, y, \Omega)}$

Gradient-based solver (adjoint/Euler-Lagrange method): $U_\Phi(x, y, \Omega)$

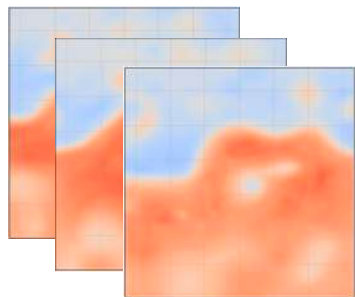
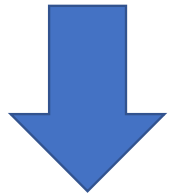
$$x^{(k+1)} = x^{(k)} - \alpha \nabla_x U_\Phi(x^{(k)}, y, \Omega)$$

Implementation of 4DVar Data Assimilation using Deep Learning frameworks

- Implement operator Φ as a neural network (e.g., neural ODE architecture w.r.t. a known dynamical operator)
- Use embedded automatic differentiation tool to implement the above gradient descent without deriving analytically the adjoint operator
- Link to an example for L63 dynamics: https://colab.research.google.com/drive/1SyU6PzOAzgBz4_U_5WbPMriwT4bLXKvh?usp=sharing



Partial observations y



True states x

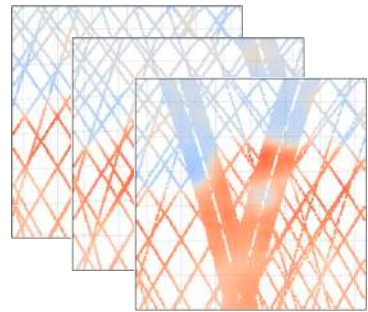
End-to-end learning for inverse problems (Fablet et al., 2020)

Model-driven schemes: $\hat{x} = \arg \min_x \underbrace{\lambda_1 \|x - y\|_\Omega^2 + \lambda_2 \|x - \Phi(x)\|_\Omega^2}_{U_\Phi(x^{(k)}, y, \Omega)}$

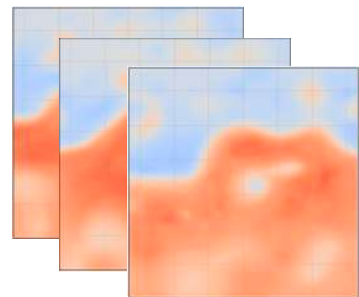
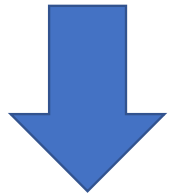
Gradient-based solver (adjoint/Euler-Lagrange method):

$$x^{(k+1)} = x^{(k)} - \alpha \nabla_x U_\Phi(x^{(k)}, y, \Omega)$$

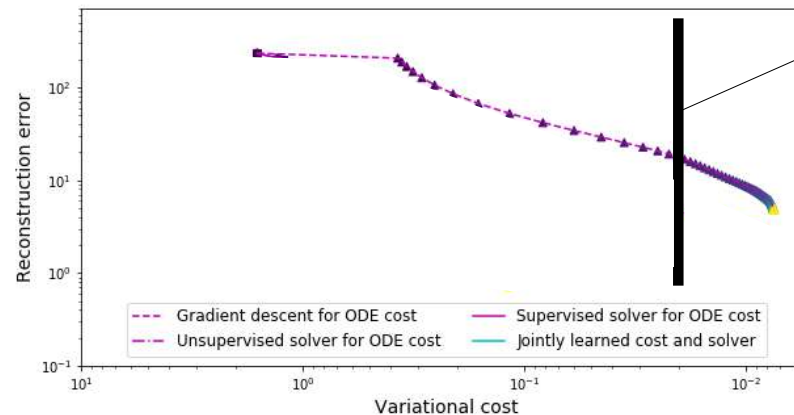
No control on the reconstruction error $x^{true} \neq \arg \min_x U_\Phi(x^{(k)}, y, \Omega)$



Partial observations y



True states x

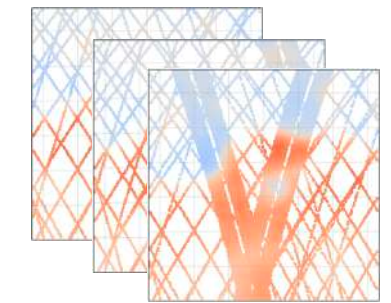


Variational cost for the true state

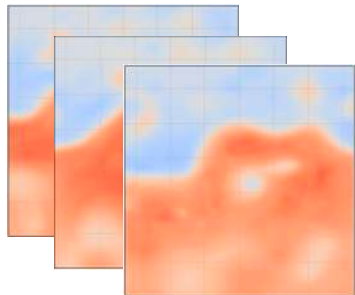
End-to-end learning for inverse problems (Fablet et al., 2020)

Model-driven schemes: $\hat{x} = \arg \min_x \lambda_1 \|x - y\|_{\Omega}^2 + \lambda_2 \Phi(x)$

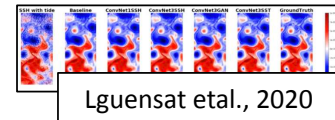
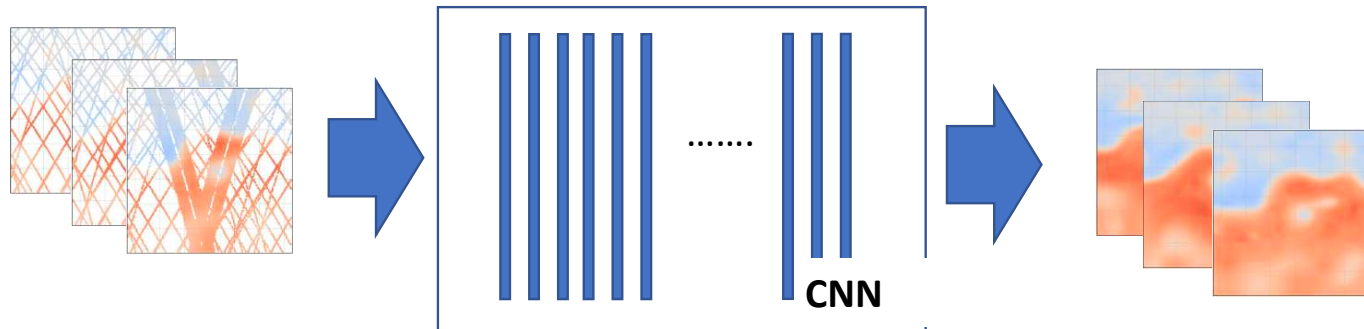
Direct learning for inverse problems: $\hat{x} = \Psi(y)$ $y \rightarrow \text{CNN} \rightarrow x$



Partial observations y



True states x

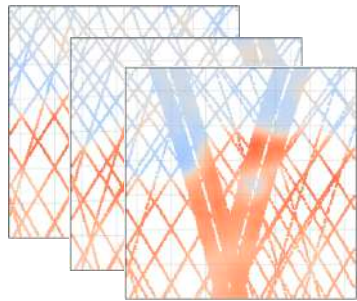


Lguensat et al., 2020

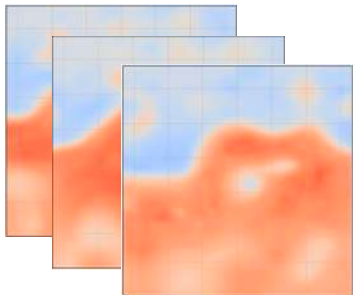
Examples of CNN architectures: Reaction-Diffusion architectures, ADMM-inspired architectures,...

Good performance but possibly weak interpretability/generalization capacities of the solution beyond the training cases

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



Partial observations y



True states x

Model-driven schemes: $\hat{x} = \arg \min_x \lambda_1 \|x - y\|_{\Omega}^2 + \lambda_2 \Phi(x)$

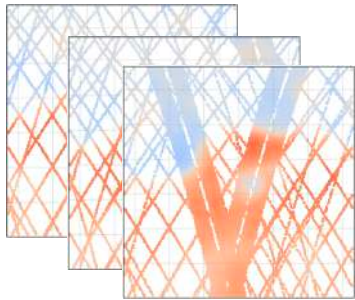
Direct learning for inverse problems: $\hat{x} = \Psi(y)$ $y \rightarrow \text{CNN} \rightarrow x$

Proposed scheme: joint learning of the variational model and solver

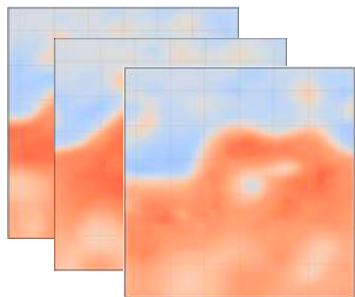
- Theoretical bi-level optimization

$$\arg \min_{\Phi} \sum_n \|x_n - \tilde{x}_n\|^2 \text{ s.t. } \tilde{x}_n = \arg \min_{x_n} U_{\Phi}(x_n, y_n, \Omega_n)$$

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



Partial observations y



True states x

Model-driven schemes: $\hat{x} = \arg \min_x \lambda_1 \|x - y\|_{\Omega}^2 + \lambda_2 \Phi(x)$

Direct learning for inverse problems: $\hat{x} = \Psi(y)$ $y \rightarrow \text{CNN} \rightarrow x$

Proposed scheme: joint learning of the variational model and solver

- Theoretical bi-level optimization

$$\arg \min_{\Phi} \sum_n \|x_n - \tilde{x}_n\|^2 \text{ s.t. } \tilde{x}_n = \arg \min_{x_n} U_{\Phi}(x_n, y_n, \Omega_n)$$

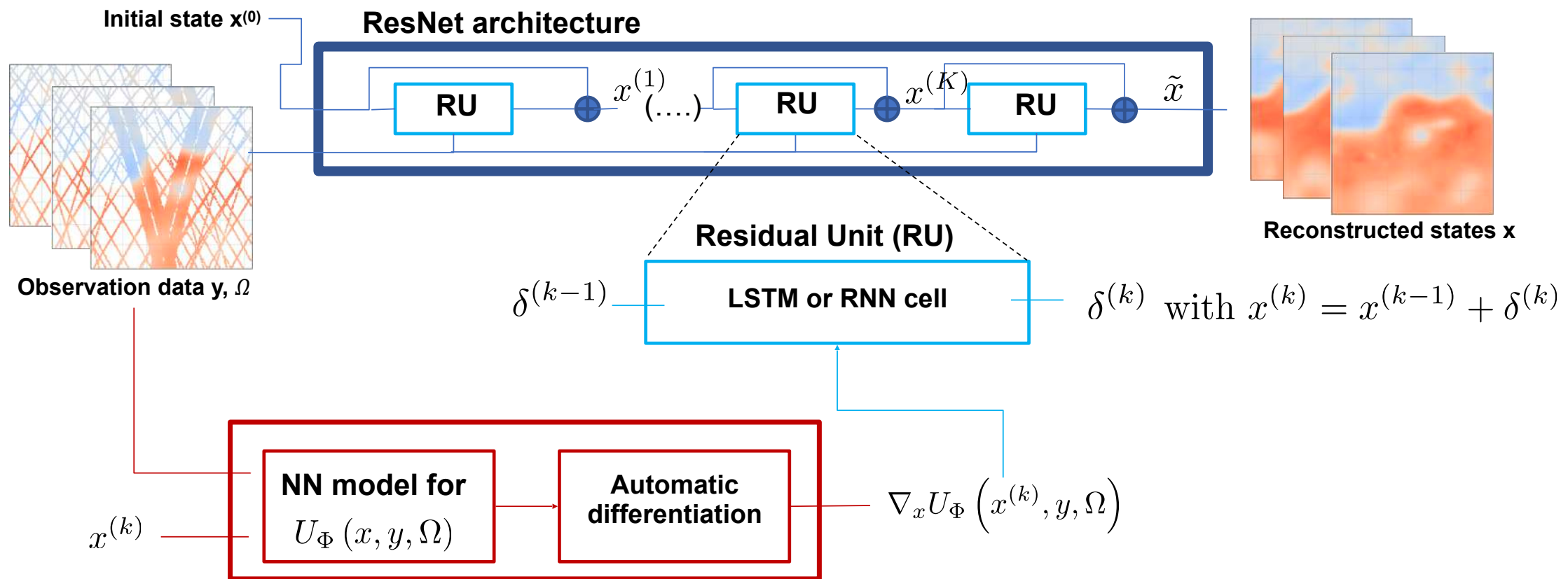
- Restated with a gradient-based NN solver for inner minimization

$$\arg \min_{\Phi, \Gamma} \sum_n \|x_n - \tilde{x}_n\|^2 \text{ s.t. } \tilde{x}_n = \Psi_{\Phi, \Gamma}(x_n^{(0)}, y_n, \Omega_n)$$

Iterative NN solver using automatic differentiation to compute gradient $\nabla_x U_{\Phi}(x^{(k)}, y, \Omega)$

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

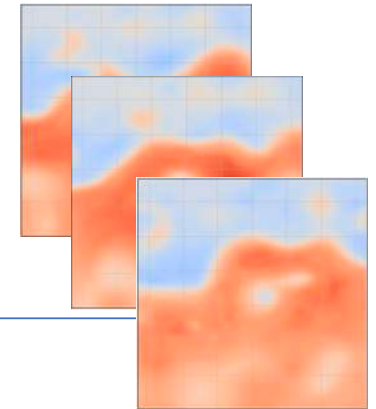
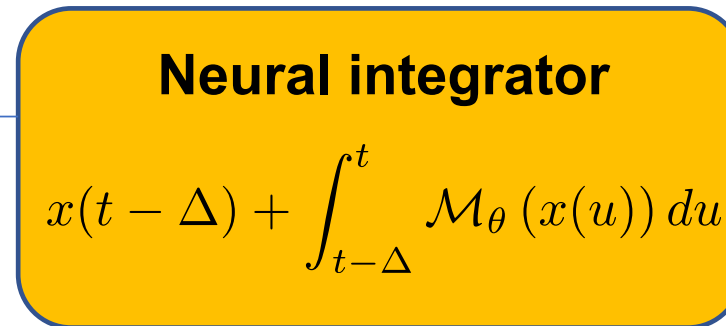
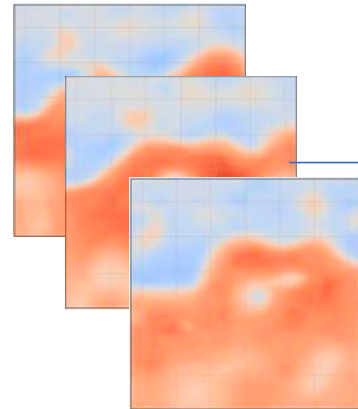
Proposed scheme: **associated NN architecture**



End-to-end learning for 4DVar DA: projection operator Φ

Parameterization
using (learnable)
ODE operator

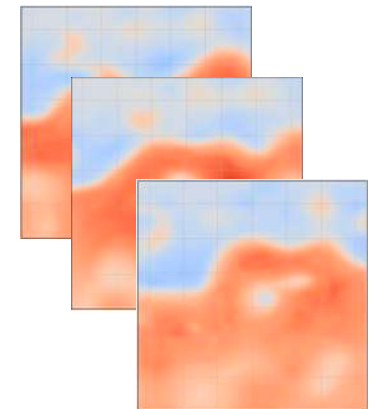
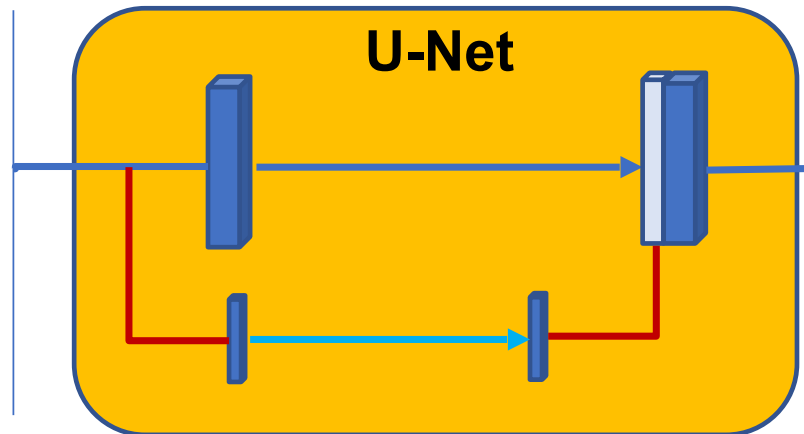
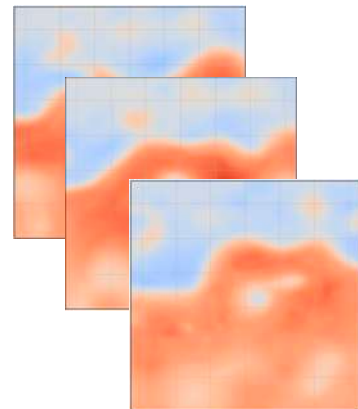
$$\frac{\partial x(t)}{\partial t} = \mathcal{M}_\theta(x(t))$$



x

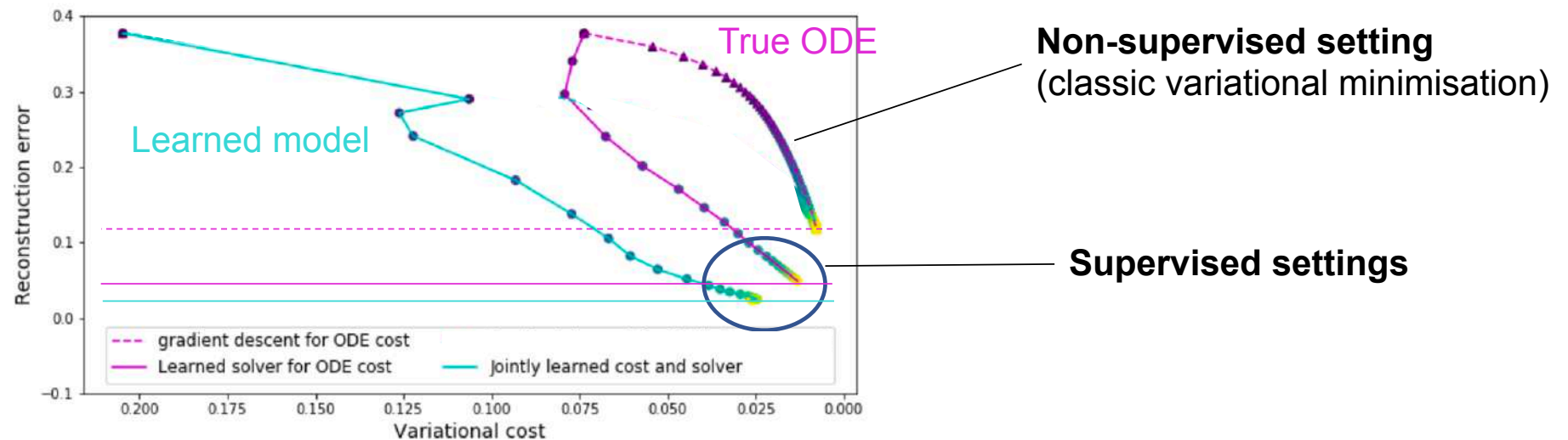
$\Phi(x)$

Two-scale
U-Net-like
Parameterization
(Gibbs Field)

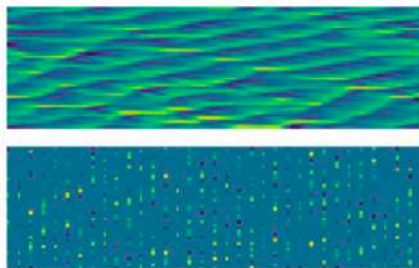


End-to-end learning for inverse problems (Fablet et al., 2020)

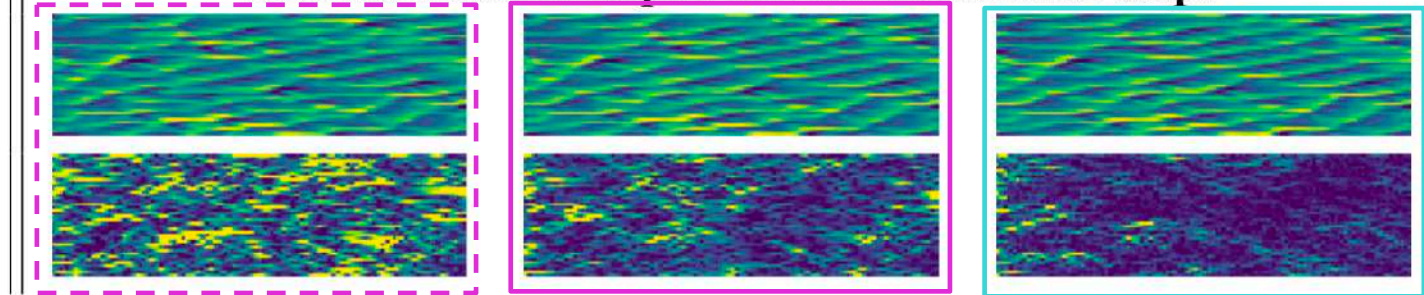
Illustration on Lorenz-96 dynamics (Bilinear ODE)



True and observed states

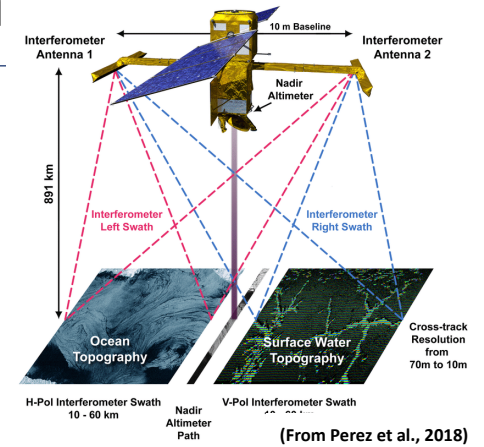
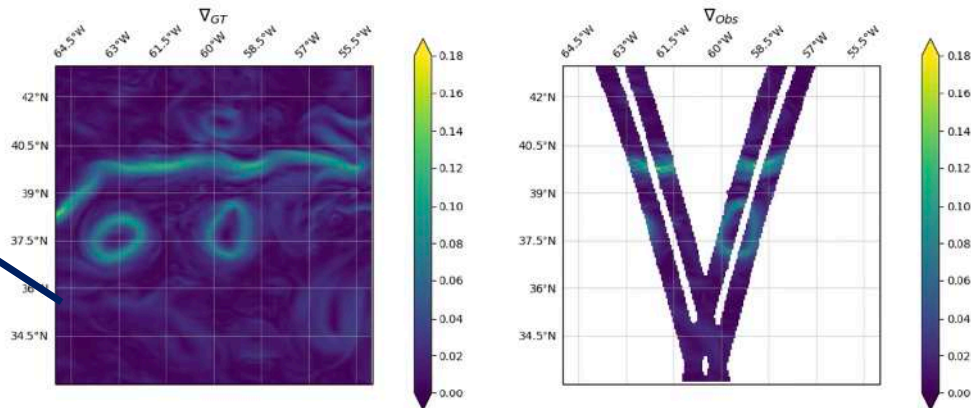


Reconstruction examples and associated error maps

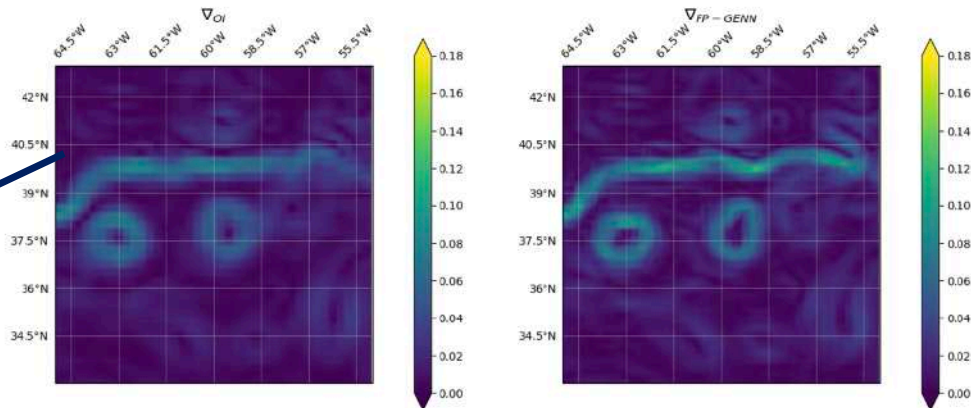


An application for upcoming SWOT mission

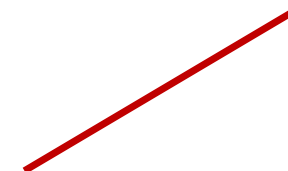
Groundtruth



State-of-the-art
operational processing



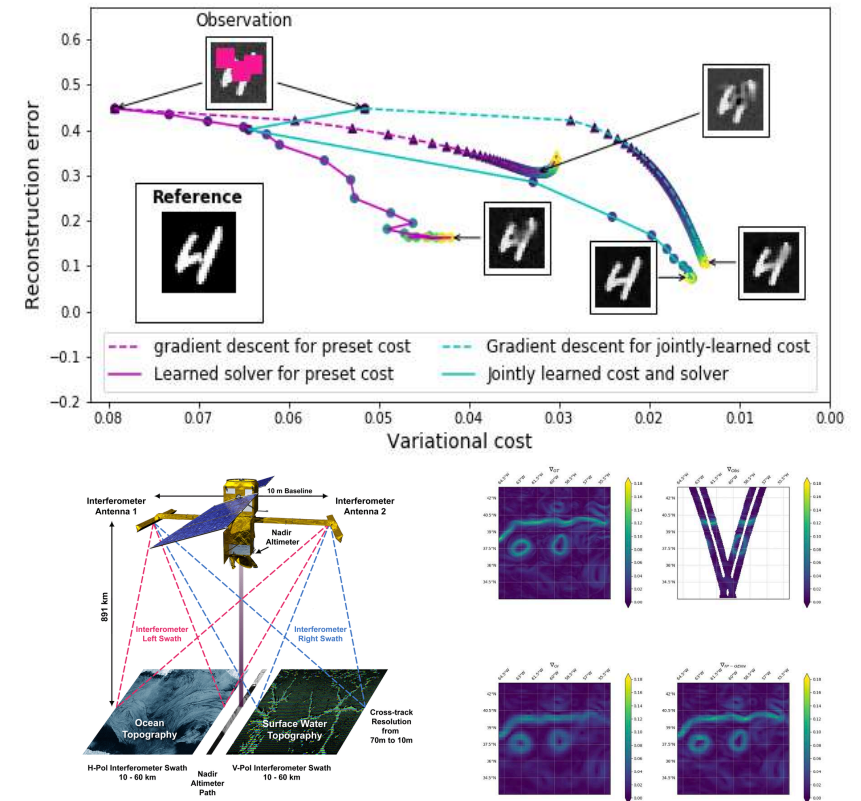
Proposed NN framework
(Fablet et al., 2019)



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**

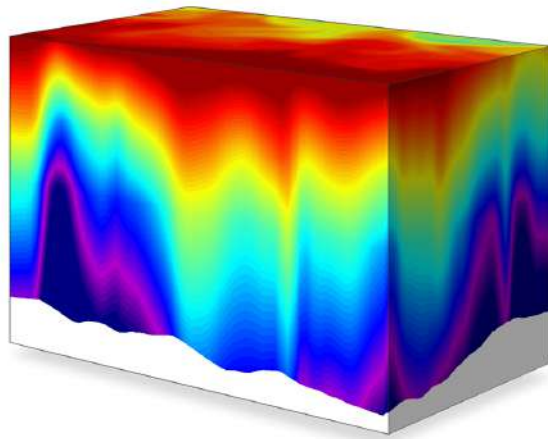
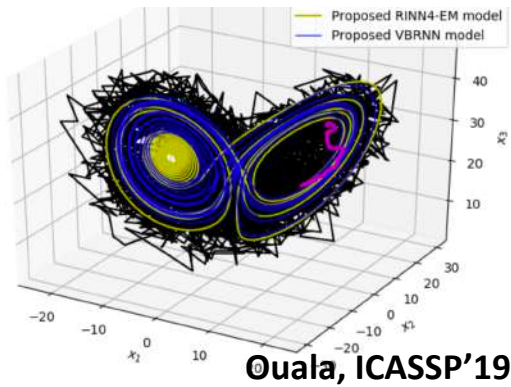


Preprint: <https://arxiv.org/abs/2006.03653>

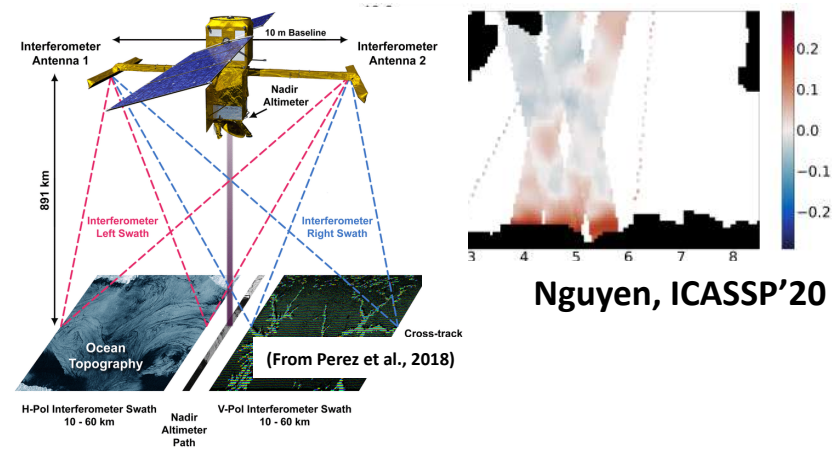
Code: <https://github.com/CIA-Oceanix>

End-to-end learning from real observation data ?

Scarce time sampling



Noisy and irregular sampling

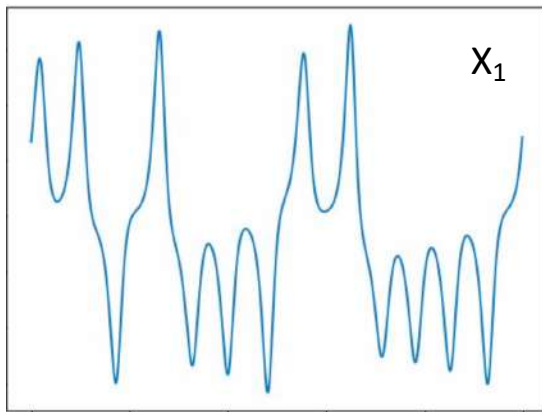


Partially-observed system

Ouala, preprint 2019

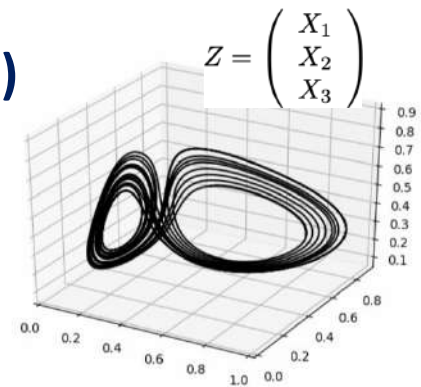
Neural ODE for partially-observed systems [Ouala et al., 2020]

Illustration for L63 assuming only the first components is observed

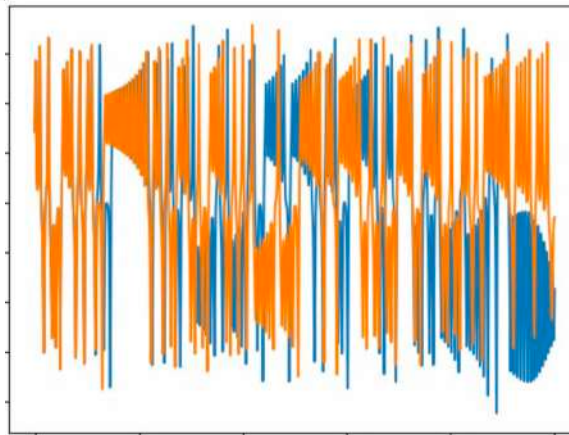


Learning Latent (unobserved)
dynamics

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



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

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

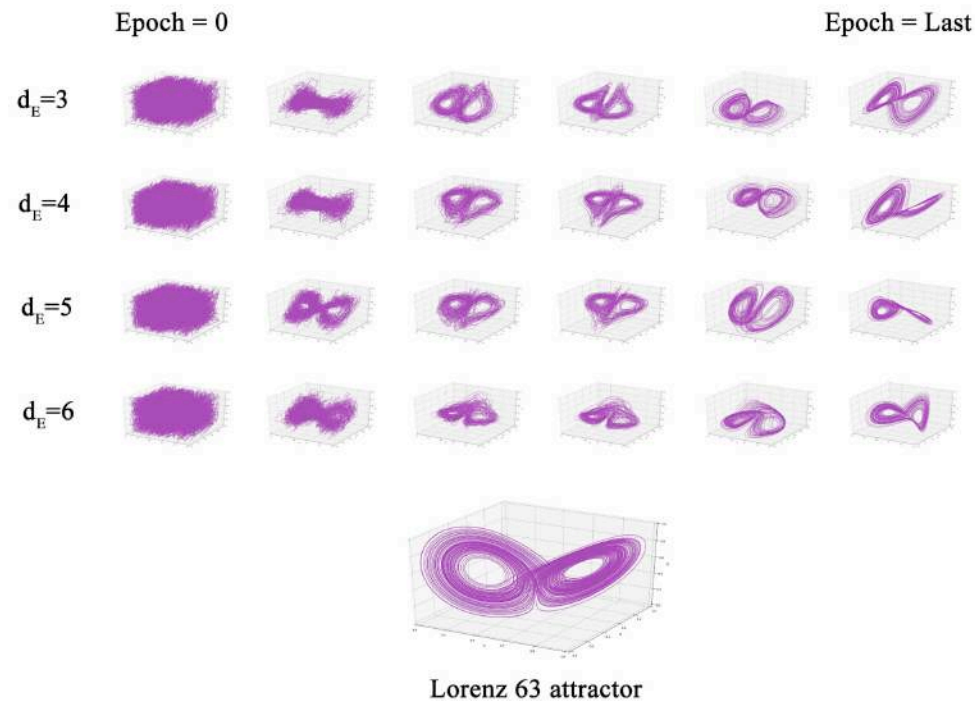
Goals:

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

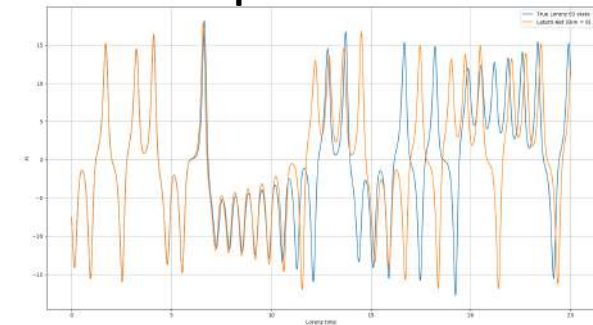
Proposed approach: WC 4DVar formulation with an unknown dynamical model

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

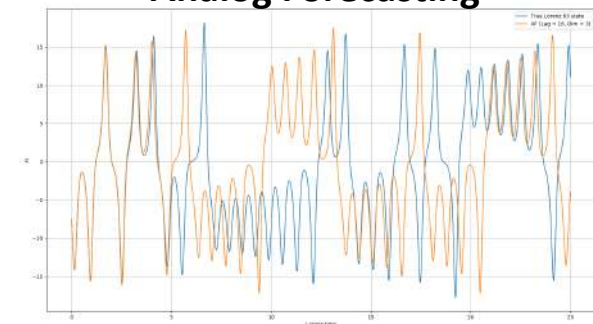
Illustration on Lorenz-63 dynamics



Proposed model



Analog Forecasting



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

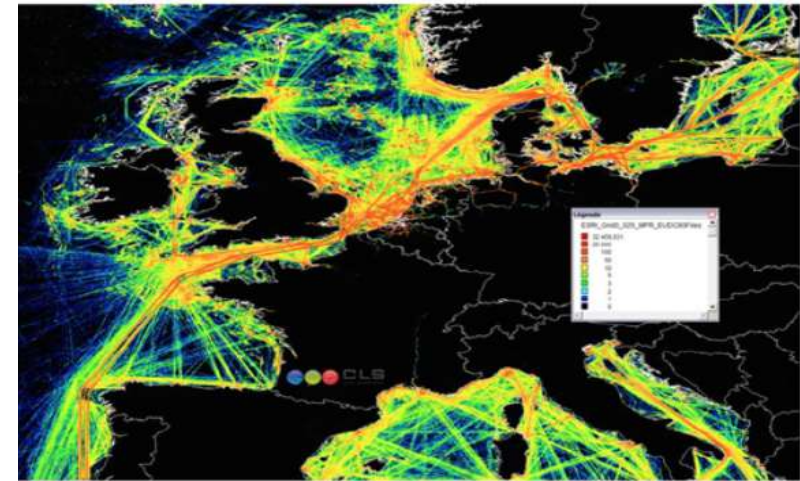
Learning stochastic hidden dynamics

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

The example of AIS Vessel trajectory data

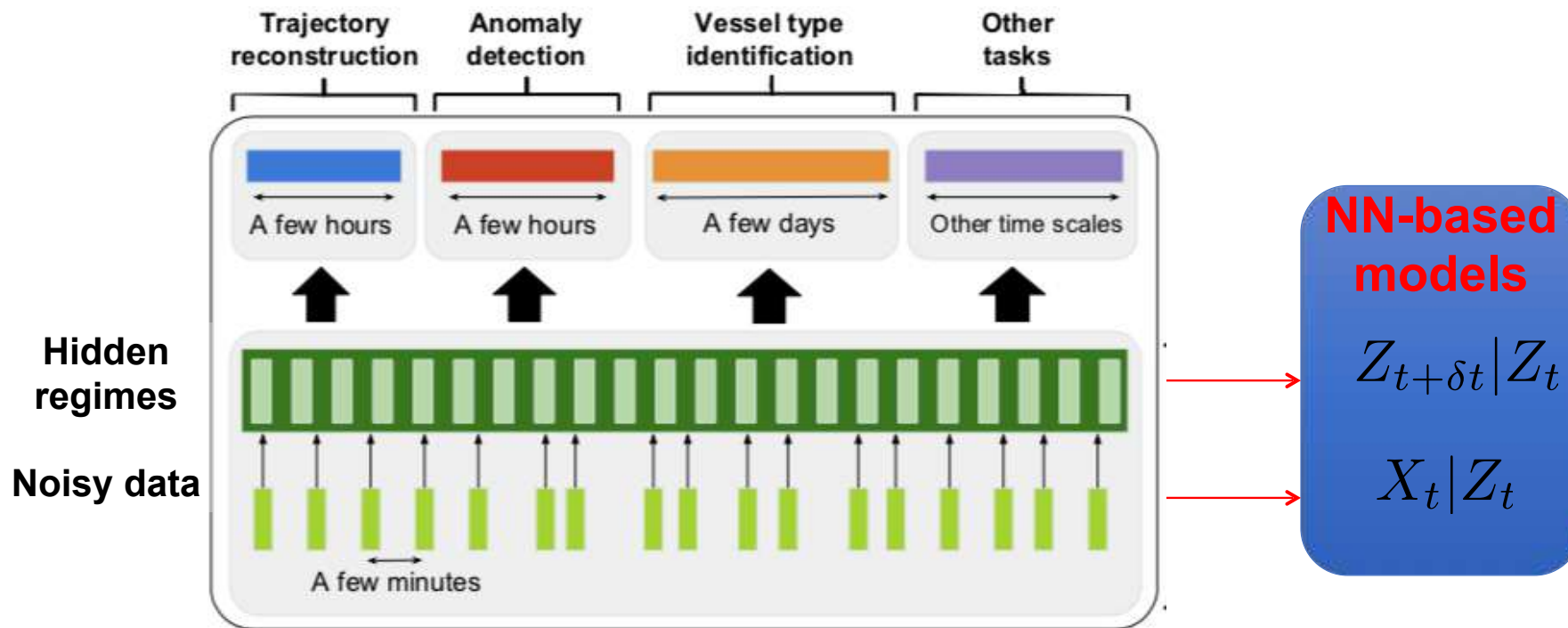


- Millions of AIS positions daily
- Noisy data: irregular sampling, corrupted data



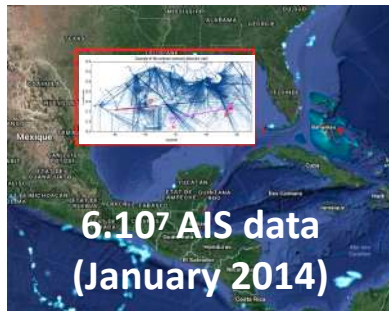
How can we learn from AIS data streams ?

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



Model training from noisy AIS streams using variational Bayesian approximation

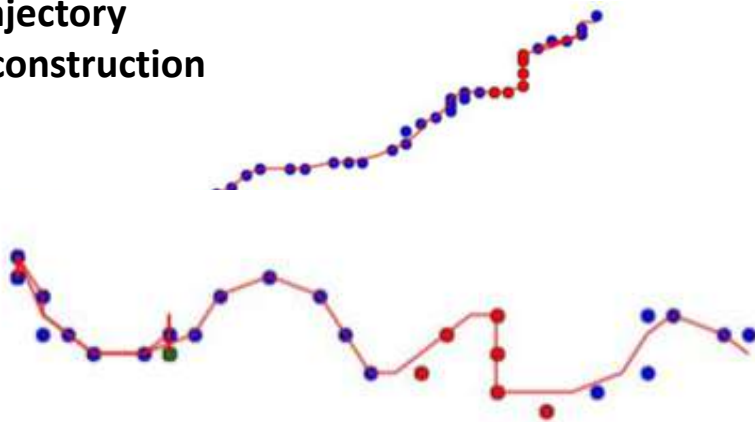
Learning stochastic hidden dynamics [Nguyen et al., 2018]



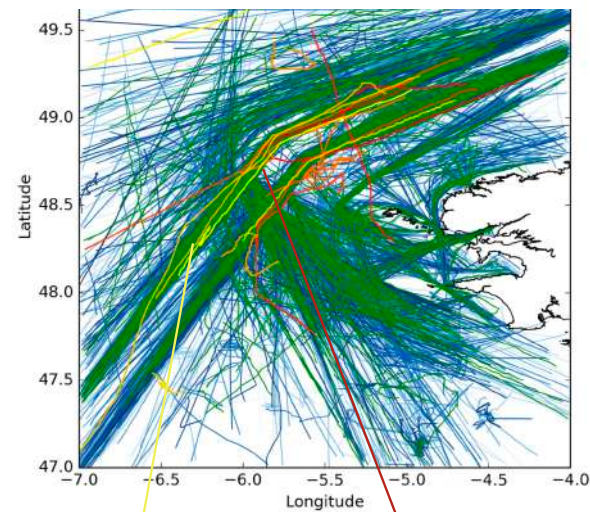
Vessel type recognition

~88% of correct
recognition

Trajectory
reconstruction



Abnormal behaviour detection



Beyond Ocean Dynamics

Dynamical System Theory for Deep Learning

Lab-STICC

Understanding DL models ?



88% **tabby cat**

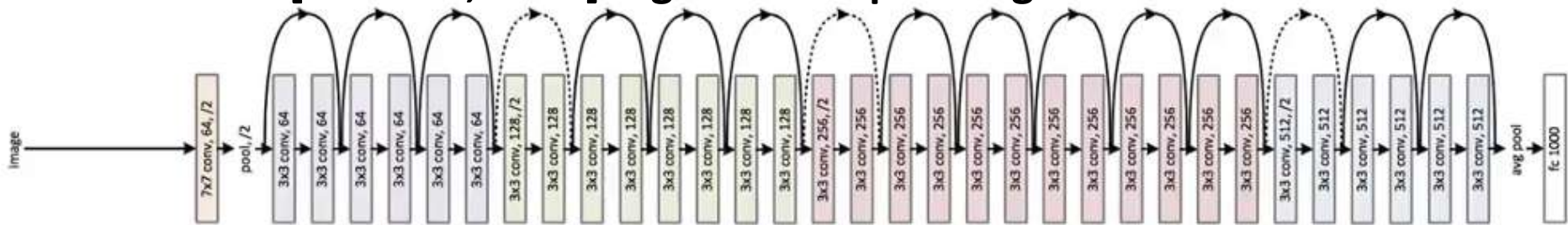
adversarial
perturbation →



99% **guacamole**

Understanding ResNets [Rousseau et al., 2019]

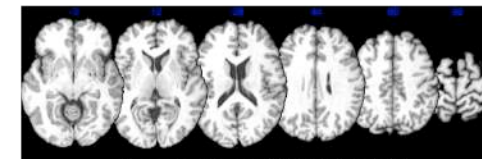
ResNet [He et al., 2015] regarded as space registration machines



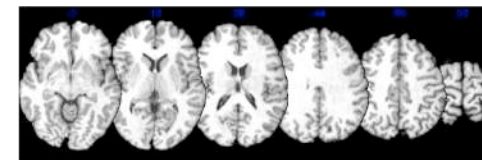
- **Image registration examples**



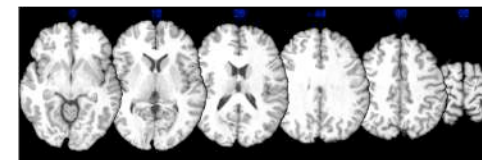
[Matlab tutorial]



a) source image (256x256x124)



b) target image (256*256*124)

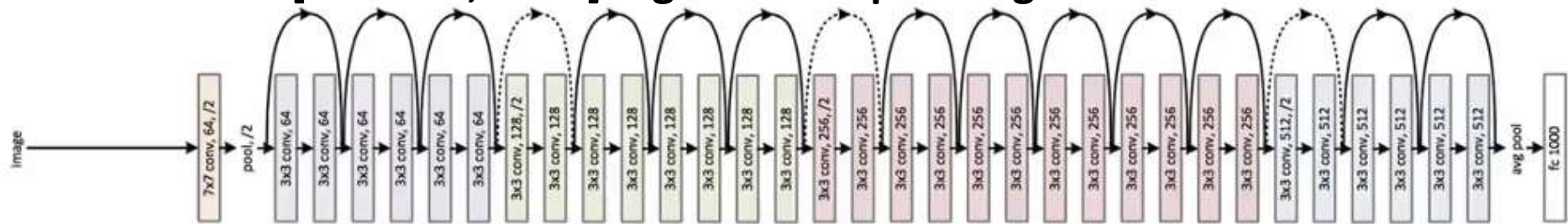


c) registered image (source to target)

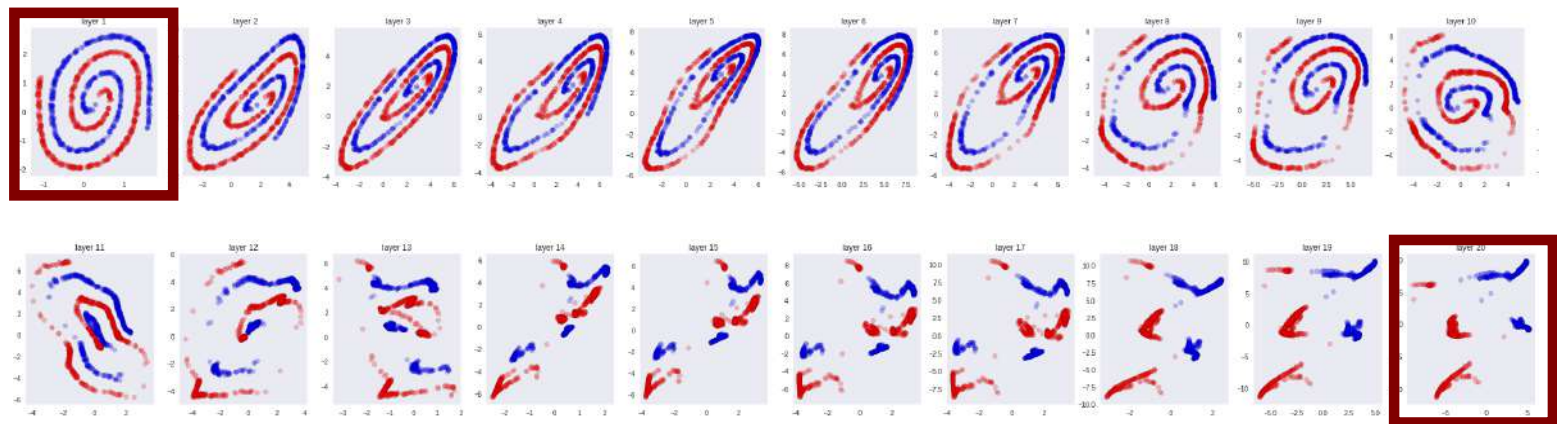
[Dramms tutorial]

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

