Variational assimilation of geophysical images leveraging deep learning tools

A. Filoche[1], D. Béréziat[1], A.Charantonis[3,4], and J.Brajard[2,3]

[1] Sorbonne Université, CNRS, LIP6, 75005 Paris, France
[2] Nansen Environmental and Remote Sensing Center (NERSC), Bergen, Norway
[3] Laboratoire d'Océanographie et du Climat (LOCEAN), Paris, France
[4] École nationale supérieure d'informatique pour l'industrie et l'entreprise (ENSIEE), Evry, France

SORBONNE UNIVERSITÉ

github

Many applications in Earth Sciences require the estimation of a physical system state. Data assimilation provides a strong framework to do so when knowledge about the governing dynamics and observations of such system are available. If this kind of inverse problem is solved in a variational form, the optimization process involves the adjoint state method for efficient gradient computing. From a computational perspective, this method is equivalent to the backpropagation algorithm and is here implemented using automatic differentiation. The ongoing development of deep learning tools allows flexible implementation of such methods and opens the door to hybrid data-knowledge driven modeling. This is illustrated on geophysical images assimilation problems.

Data Assimilation Framework

A state vector $\mathbf{X}$ evolves over a discrete time $t \in [0:T]$ according to a partially-known dynamics $\mathbb{M}$, see Eq. (1). Partial and noisy observations $\mathbf{Y}$ are available through a linear observation operator $\mathbb{H}$, Eq. (2). A background $\mathbf{X}_b$ gives prior information about the initial state, Eq. (3).

Evolution: $$\mathbf{X}_{t+1} = \mathbb{M}_t(\mathbf{X}_t) + \varepsilon_{m_t} \quad (1)$$

Observation: $$\mathbf{Y}_t = \mathbb{H}_t\mathbf{X}_t + \varepsilon_{R_t} \quad (2)$$

Background: $$\mathbf{X}_0 = \mathbf{X}_b + \varepsilon_b \quad (3)$$

Data assimilation aims at optimally combining these various sources of information to **estimate** the system state $\mathbf{X}$.

Variational Data Assimilation: 4D-Var

- **Cost function**
  In a variational formalism this estimation is done over a temporal window $[0:T]$ via the minimization of a cost function denoted J in Eq. (4).

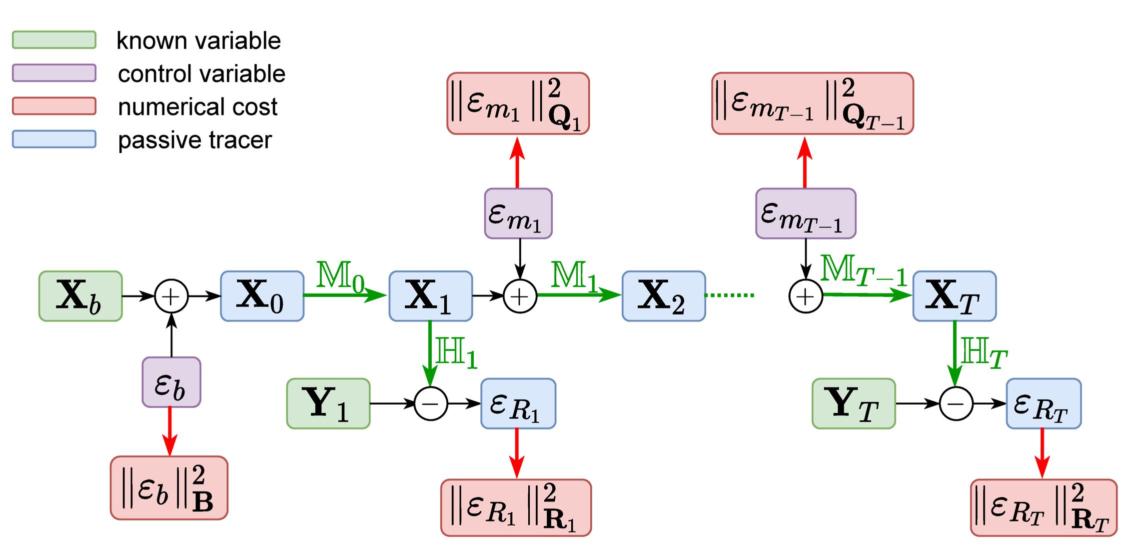
$$J(\varepsilon_b, \varepsilon_m) = \frac{1}{2}\|\varepsilon_b\|^2_{\mathbf{B}} + \frac{1}{2}\sum_{t=0}^{T}\|\varepsilon_{R_t}\|^2_{\mathbf{R}_t} + \frac{1}{2}\sum_{t=1}^{T-1}\|\varepsilon_{m_t}\|^2_{\mathbf{Q}_t} \quad (4)$$

- **Gradients**
  Usually, PDE-constrained optimization in inverse problems is solved with adjoint state methods. In our case of variational data assimilation, calculus of variations provides an analytical expression of ∇J in Eq. (5,6).

$$\nabla_{\varepsilon_b}J = \mathbf{B}^{-1}\varepsilon_b - \sum_{t=0}^{T}\left[\mathbb{H}_t\frac{\partial \mathbb{M}_{0\to t}}{\partial \mathbf{X}}\right]^{\top}\mathbf{R}_t^{-1}\varepsilon_{R_t} \quad (5)$$

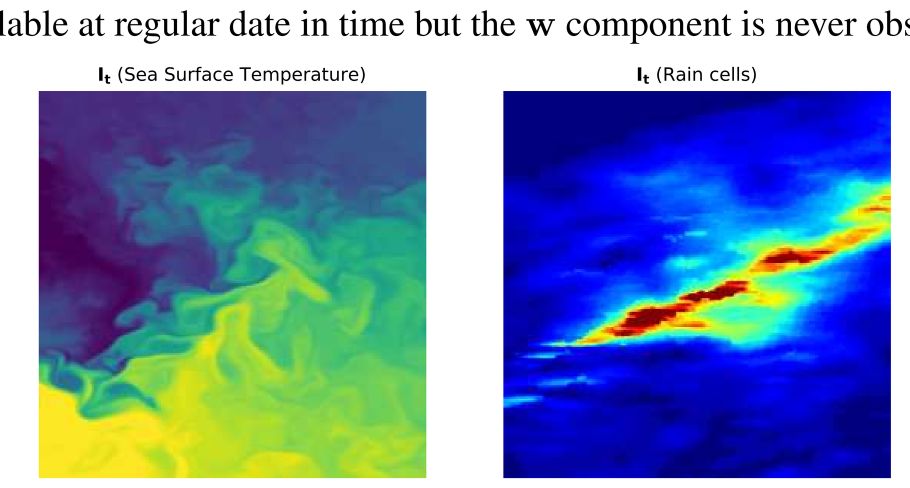
$$\nabla_{\varepsilon_{m_t}}J = \mathbf{Q}^{-1}\varepsilon_{m_t} - \sum_{t'=t+1}^{T}\left[\mathbb{H}_{t'}\frac{\partial \mathbb{M}_{t\to t'}}{\partial \mathbf{X}}\right]^{\top}\mathbf{R}_{t'}^{-1}\varepsilon_{R_{t'}} \quad (6)$$

- **Computational Graph**
  In the following figure, we present a schematic view of the forward integration leading to the calculation of J. Designing such forward mapping is very similar to designing a neural network architecture and the associated cost function.



Geophysical system

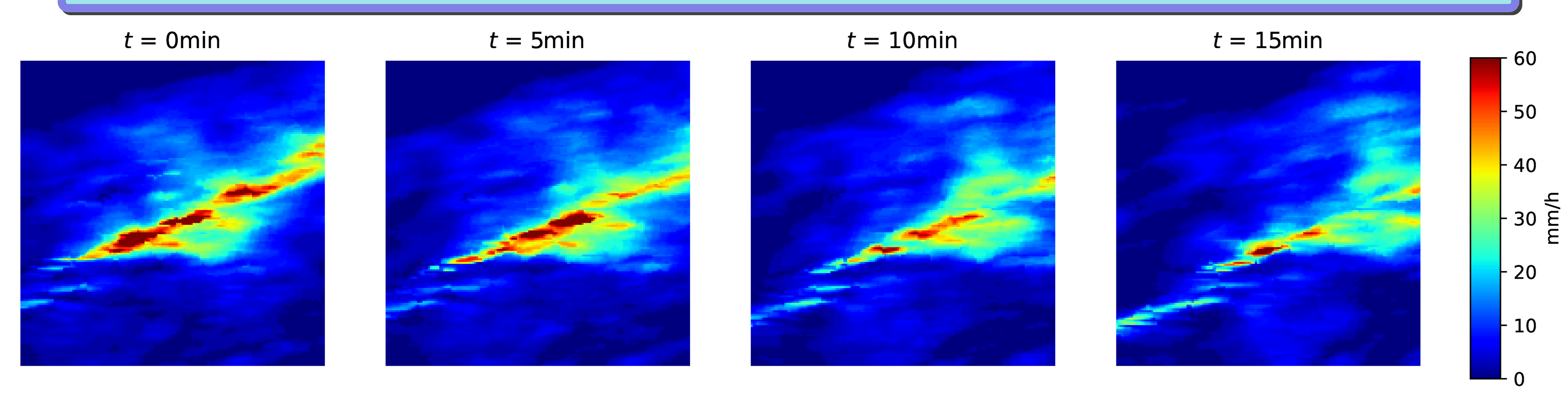
- **State vector** $\mathbf{X}_t = \begin{pmatrix}\mathbf{w}_t & I_t\end{pmatrix}^{\top}$ is composed of a geophysical image tracer I_t and the associated motion field $\mathbf{w}_t$.
- **Observation** on I_t are available at regular date in time but the $\mathbf{w}$ component is never observed.



- **Dynamical model** is described by the linear and non-linear advection equations in Eq. (7).

$$\begin{cases}\dfrac{\partial I}{\partial t} + \mathbf{w}.\nabla I = 0\\ \dfrac{\partial \mathbf{w}}{\partial t} + \mathbf{w}.\nabla\mathbf{w} = 0\end{cases} \quad (7)$$

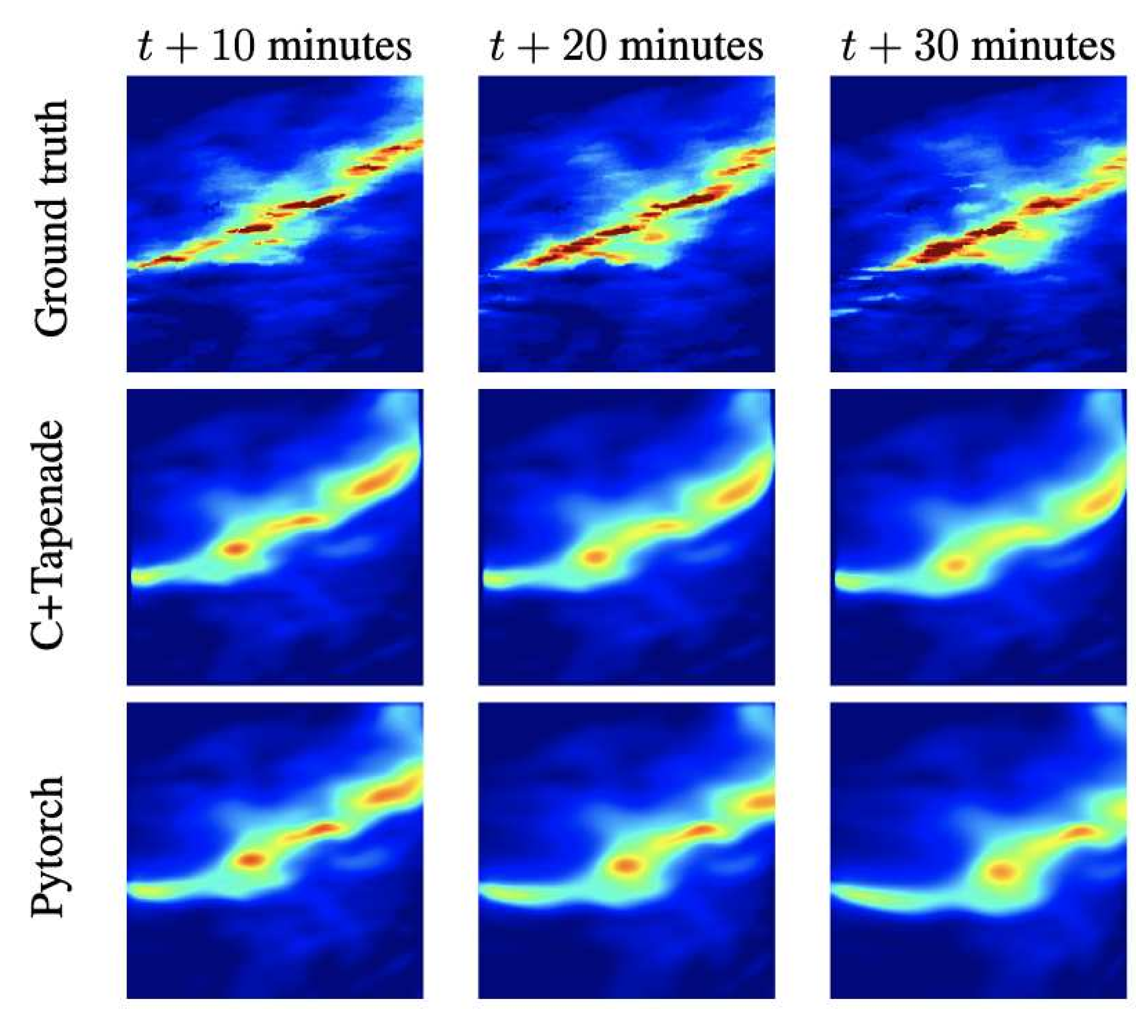
Observational window of rain radar images



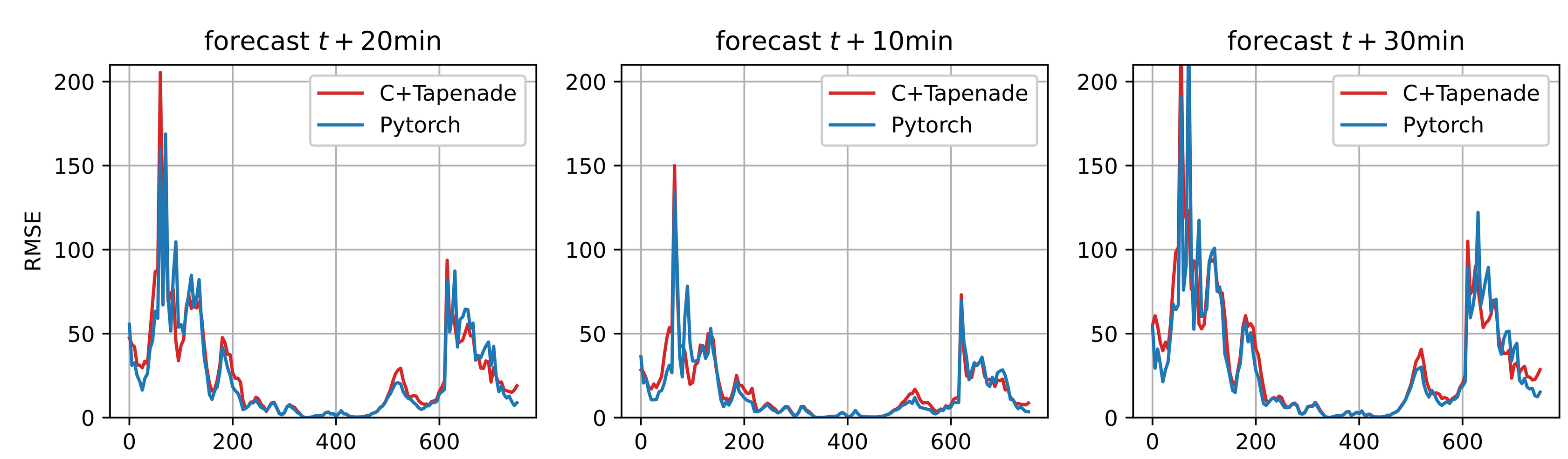
Results on rain nowcasting

After estimation of the motion field $\mathbf{w}$, we are able to produce a forecast by integrating the dynamical model.
We compare the obtained results with a reliable code coded in C and using the automatic differentiation tool Tapenade.

- **Forecasted state**



- **Forecast error**



The Pytorch code replicates reference performances closely. Explanations on differences are discussed in the paper.

- **Computation time**

Code paradigm	Rain radar	Sea surface temperature
C+Tapenade	2.0 ± 0.3 s	25.8 ± 1.1 s
Pytorch	90.0 ± 43.6 s	71.3 ± 1.1 s

Pytorch flexibility comes at a computational cost.

Conclusion

- It is possible to use deep learning tools outside the scope of neural networks
- We employ them to solve geophysical inverse problems requiring automatic differentiation for the adjoint state method
- Solving such variational problems numerically becomes much more accessible
- We compared the results we obtain with a trustworthy code to validate our approach
- Regarding Earth sciences and modeling, we believe this tools can unify data-driven and knowledge-driven methods