Lecture #5: Deep Learning and Inverse Problems in Geoscience

R. Fablet et al.
ronan.fablet@imt-atlantique.fr
web: cia-oceanix.github.io

Advanced course on Deep Learning and Geophysical Dynamics

Inverse Problems in Geoscience

Mathematical formulation for inverse Problems

Inverse problems & Deep learning

Applications to geophysical dynamics

Inverse Problems in Geoscience

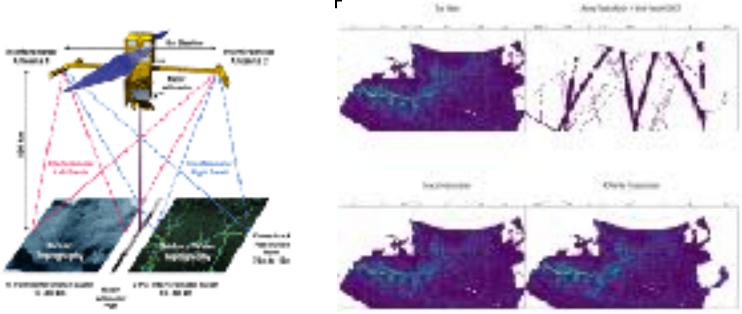
Mathematical formulations for inverse Problems

Inverse problems as learning problems

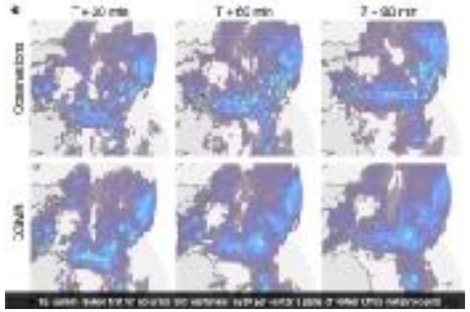
Applications to geophysical dynamics

Inverse Problems in Geoscience: some examples

Interpolation

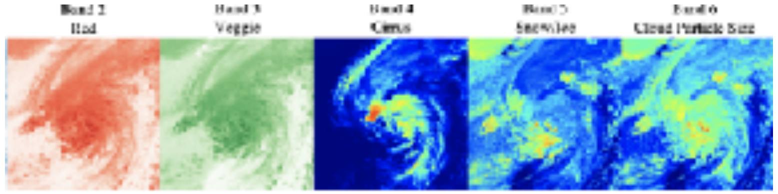


Obs.-driven Forecasting



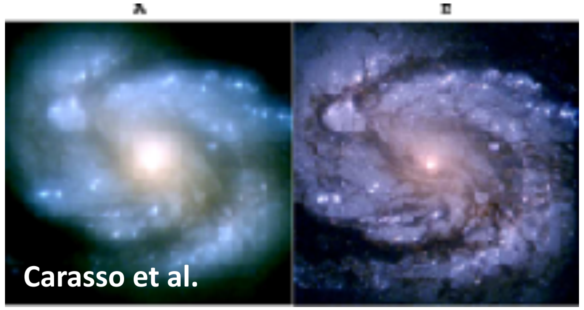
Deepmind

Multimodal fusion



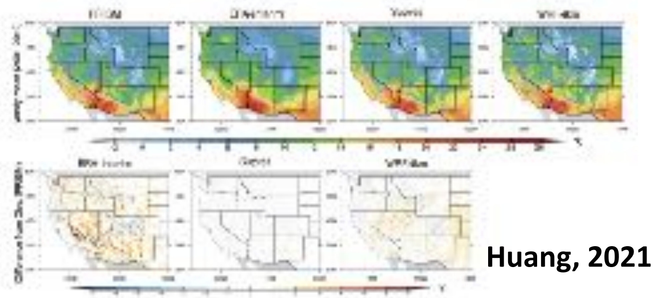
Vandal et al.

Deconvolution



Carasso et al.

Downscaling

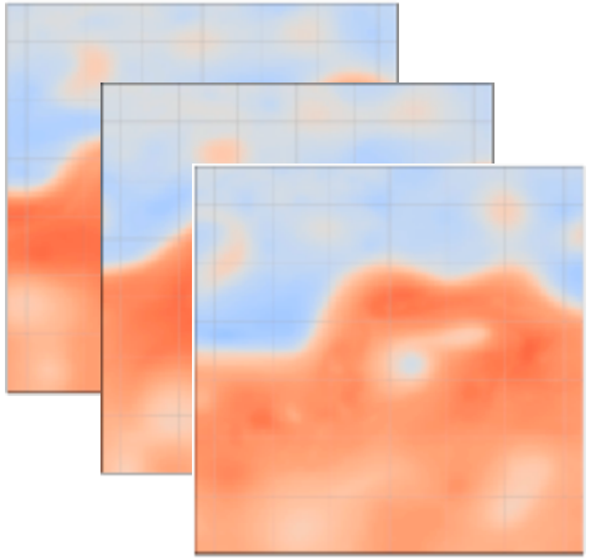


Huang, 2021

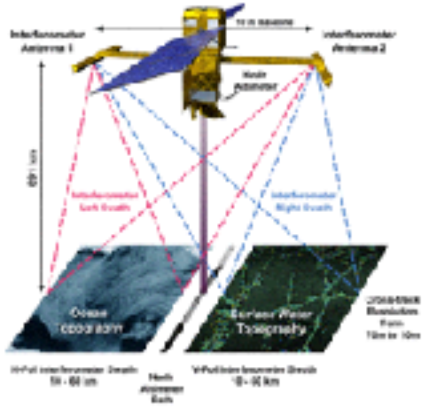
....

Inverse Problems in Geoscience: Generic formulation

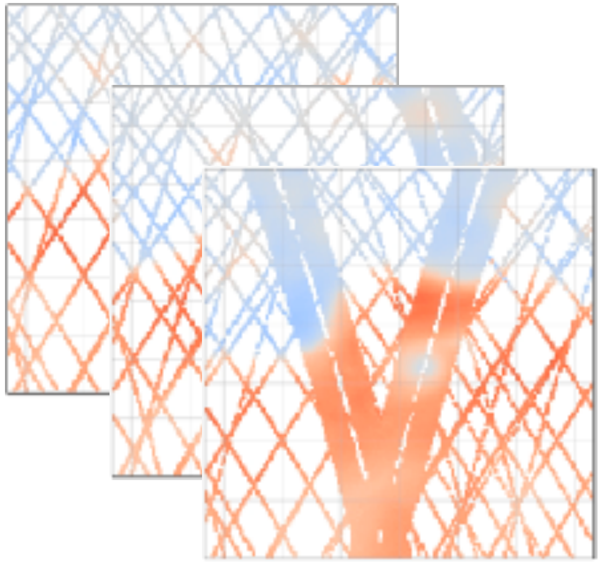
State



Observing system

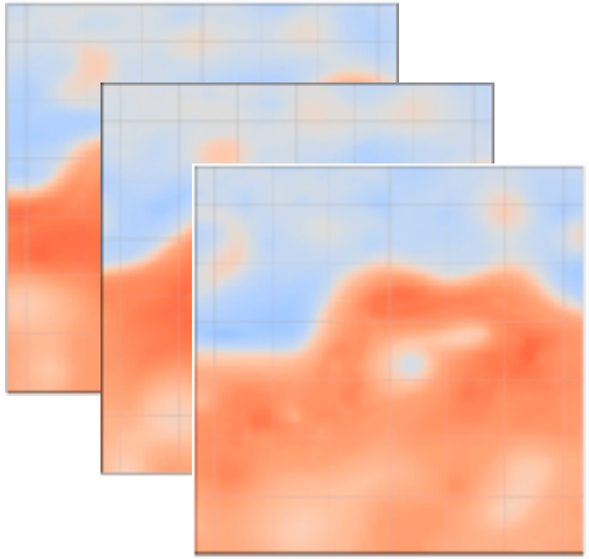


Observation



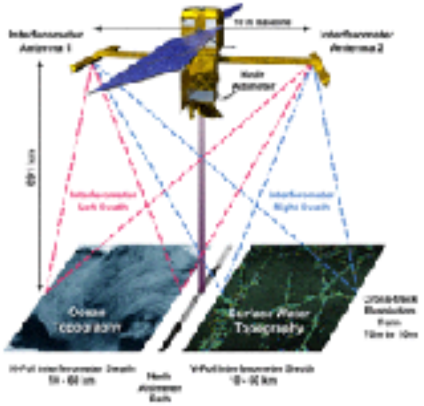
Inverse Problems in Geoscience: Generic formulation

State



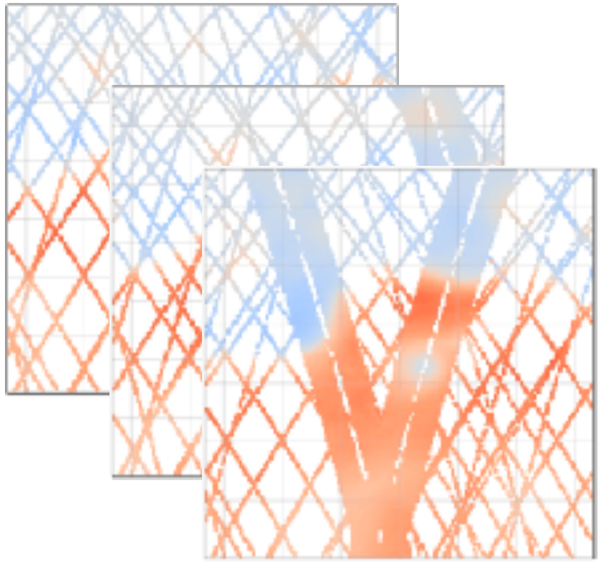
X

Observing system



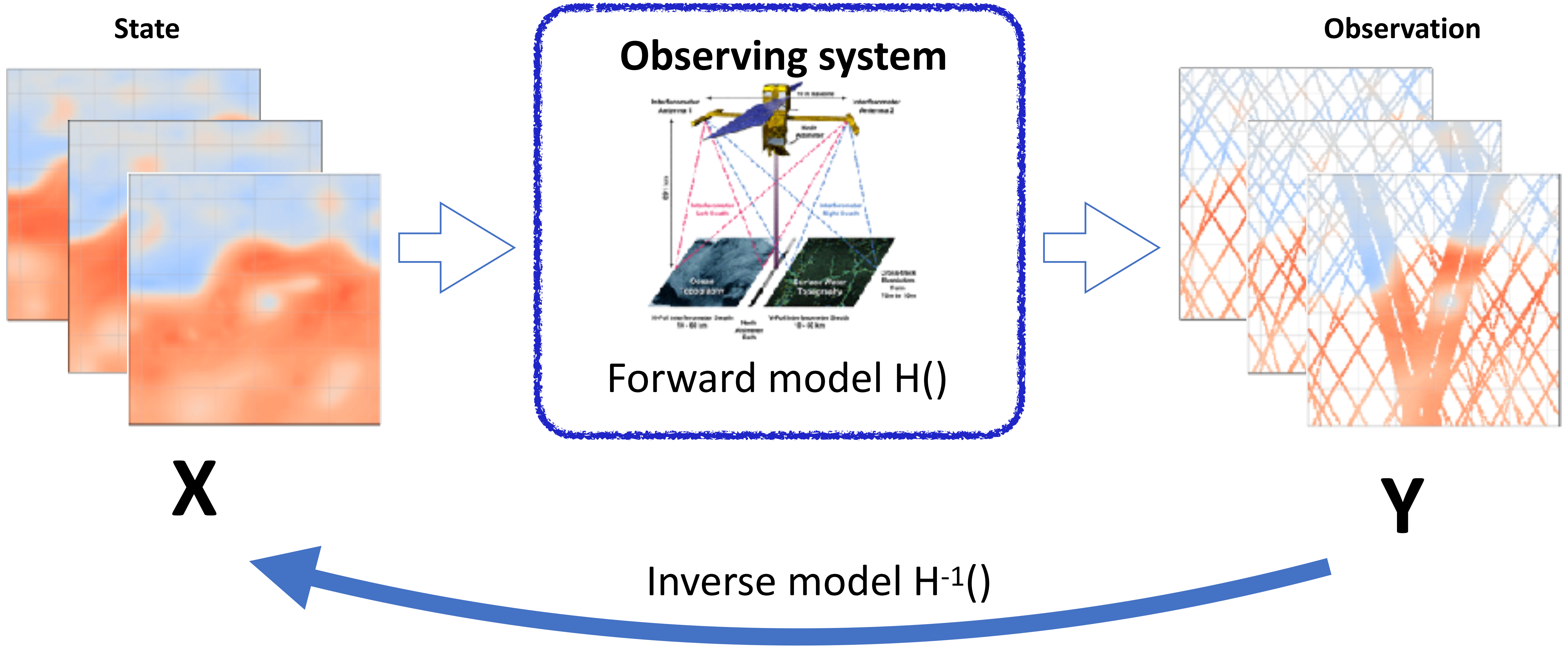
Forward model H()

Observation



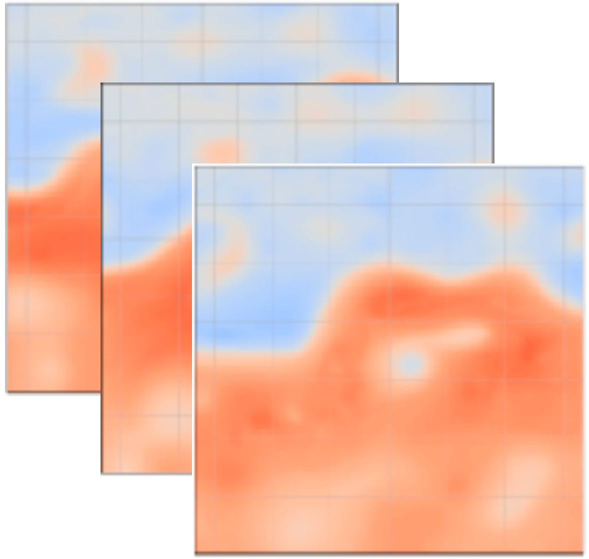
Y

Inverse Problems in Geoscience: Generic formulation

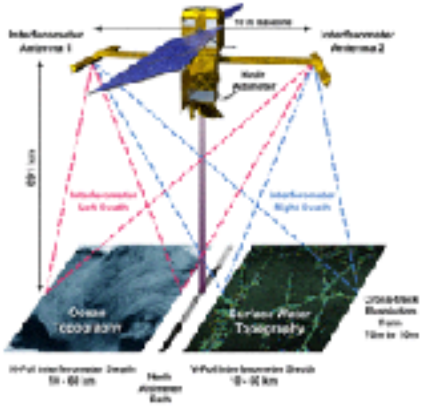


Inverse Problems in Geoscience: Examples of forward model

State

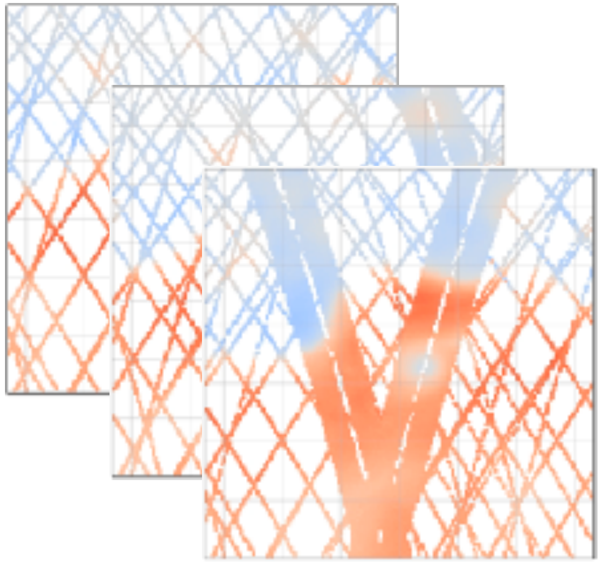


Observing system



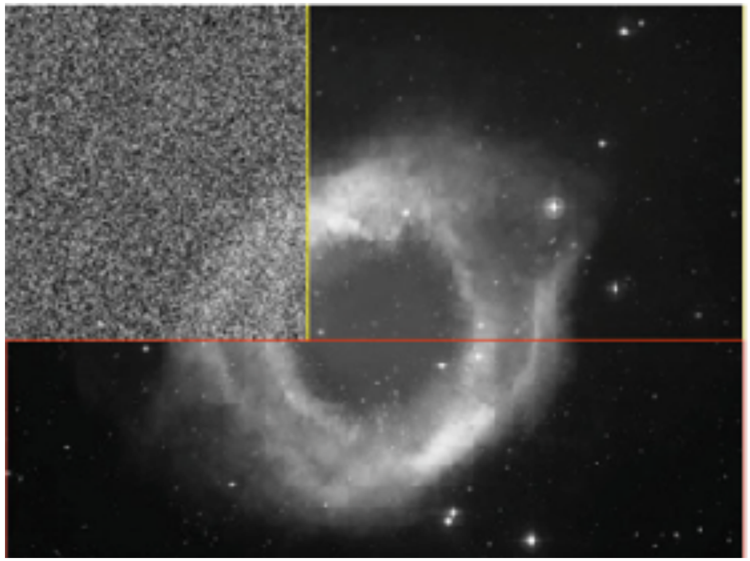
Forward model H()

Observation

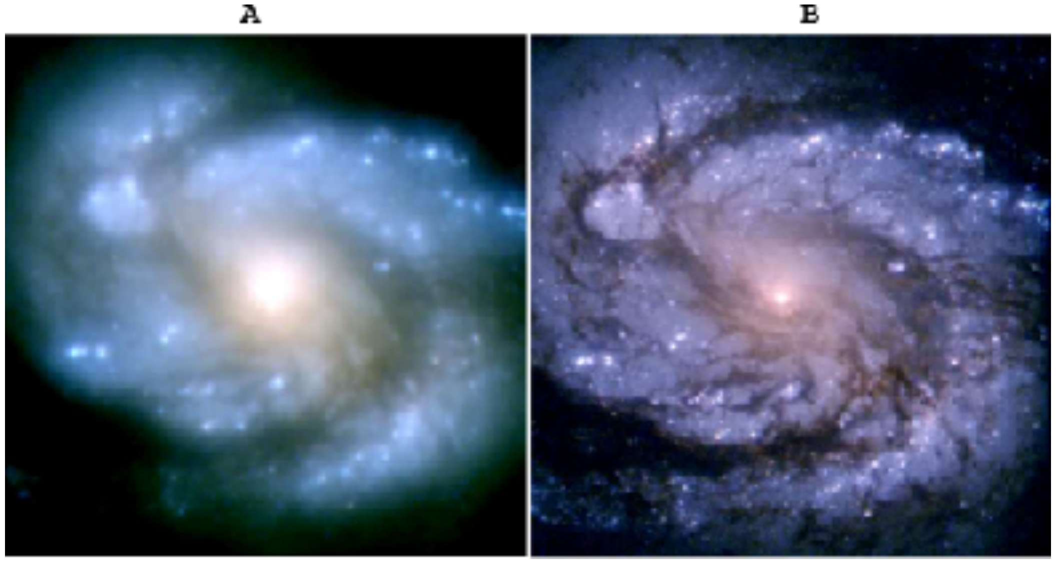


Inverse Problems in Geoscience: Examples of forward model

Denoising problem

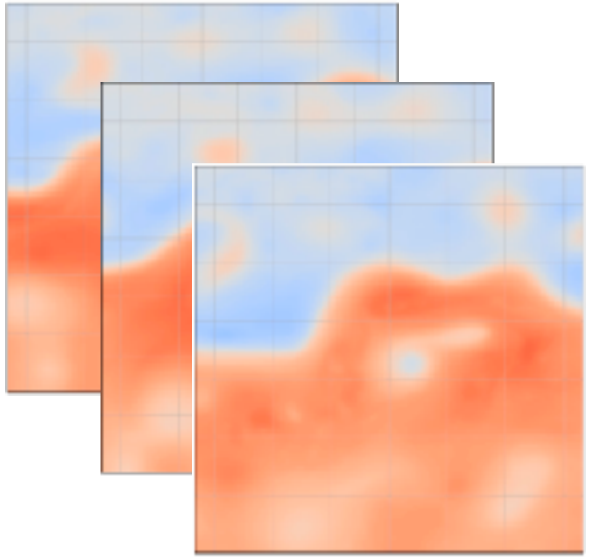


Downscaling problem

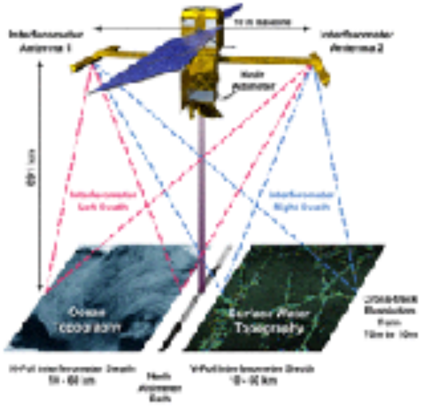


Inverse Problems and ill-posedness

State

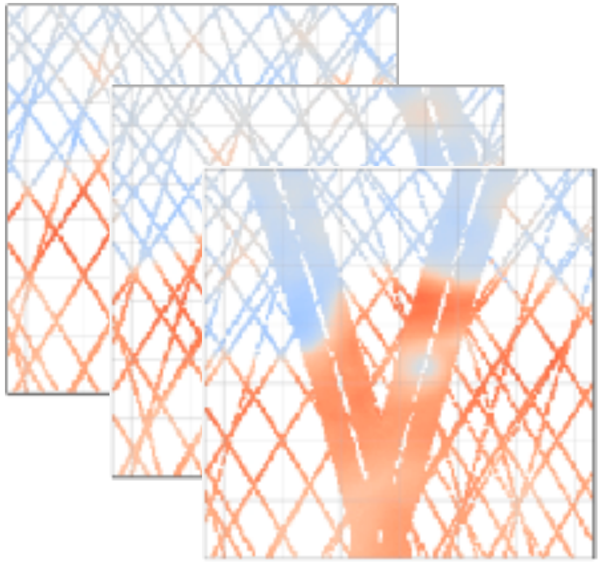


Observing system



Forward model H()

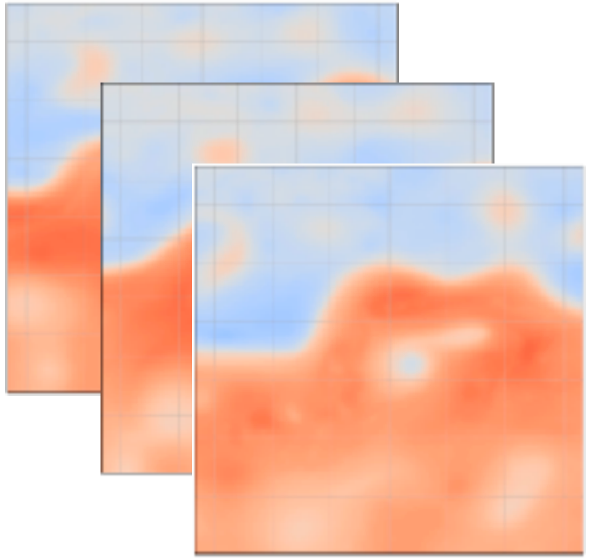
Observation



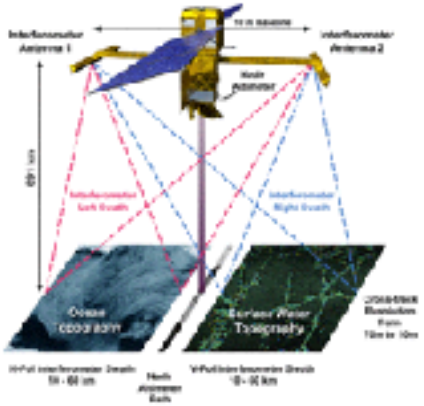
Why is space-time interpolation an ill-posed problem ?

Inverse Problems and ill-posedness

State

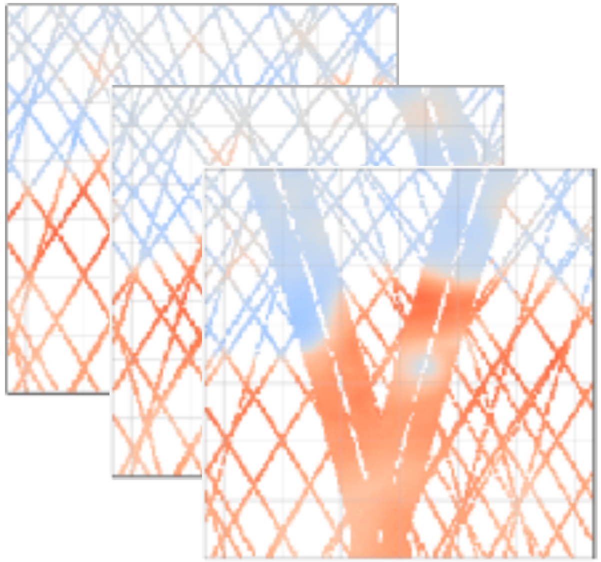


Observing system



Forward model H()

Observation



X

Y

Inverse model $H^{-1}()$

We need some additional information (prior) to make the problem invertible

Inverse Problems in Geoscience

Inverse problems as learning problems

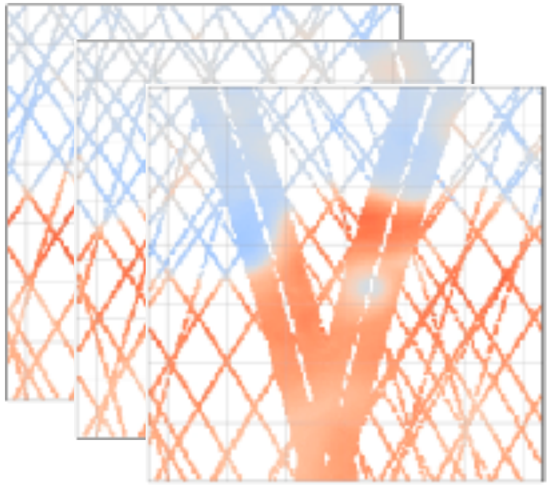
Mathematical formulations for inverse Problems

Inverse problems as learning problems

Applications to geophysical dynamics

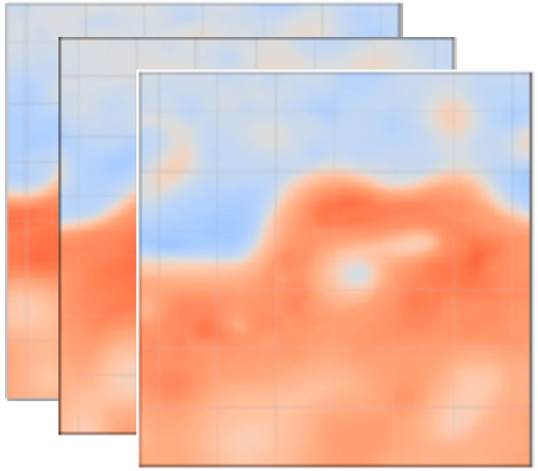
End-to-end learning for inverse problems

End-to-end architecture



Partial observations y

Deep learning Models



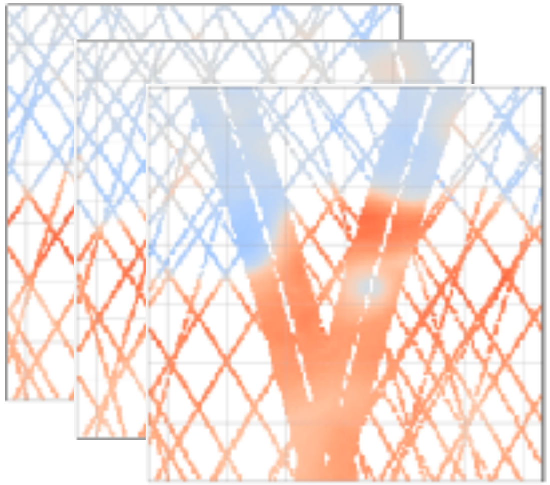
True states x

Assuming a dataset of pairs of true states and partial observations is available

Which training loss ? **Which models / architectures ?**

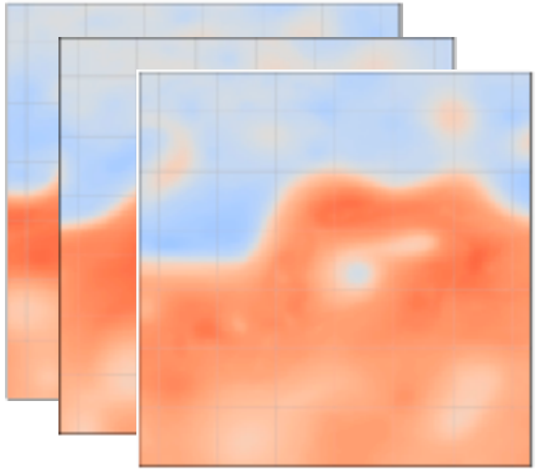
End-to-end learning for inverse problems

End-to-end architecture



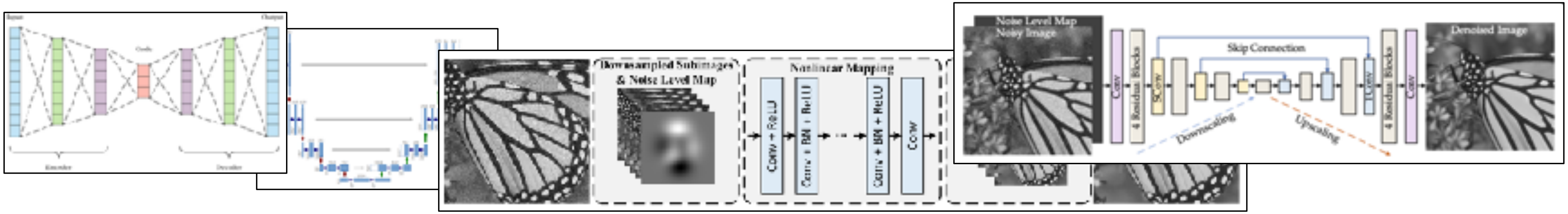
Partial observations y

Deep learning Models



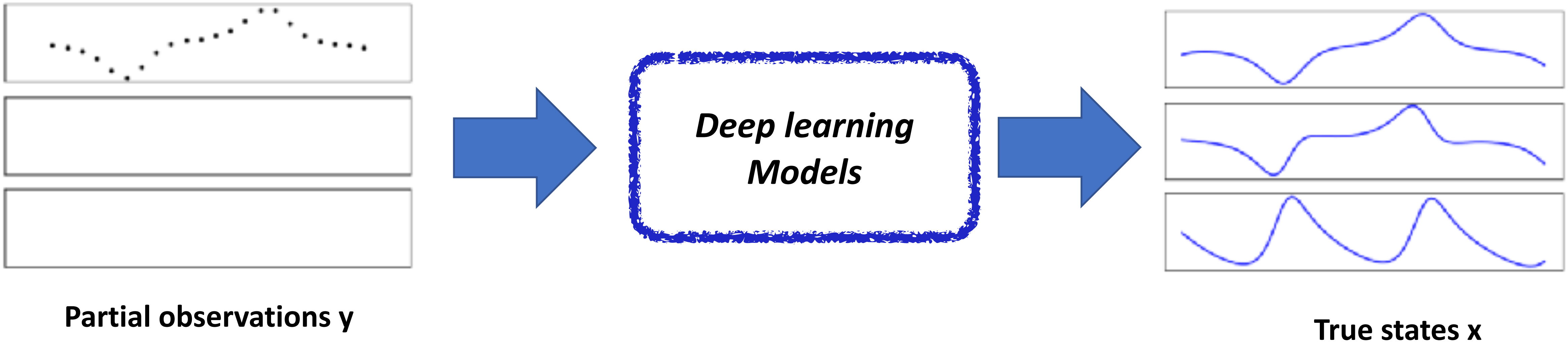
True states x

Which architectures? State-of-the-art CNN architectures?



End-to-end learning for inverse problems

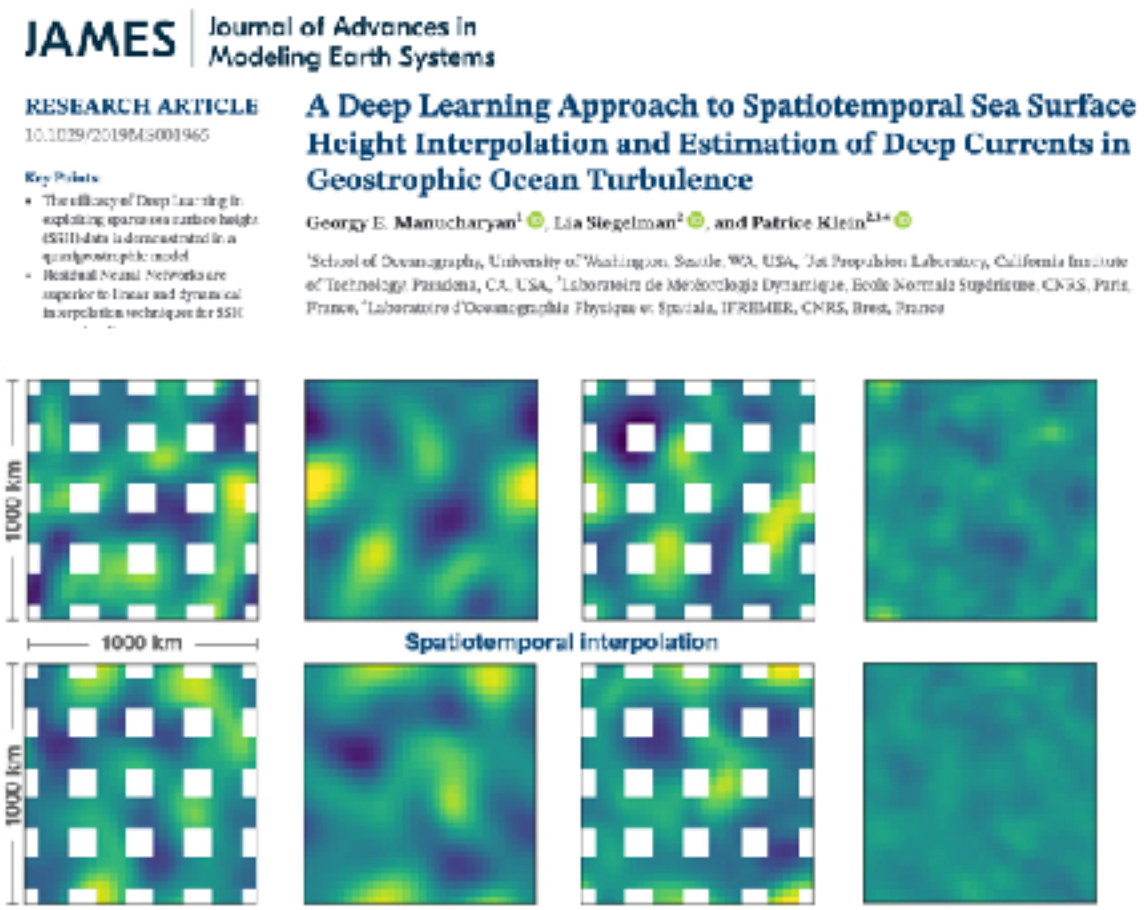
An illustration for Lorenz-63 dynamics



Colab notebook:

https://github.com/CIA-Oceanix/DLGD2022/blob/main/lecture-4-dl-oi-da/notebooks/notebookPyTorch_InvProb_LearningBased_UnrollingFixedPoint_L63.ipynb

Example for geoscience problems



JAMES | Journal of Advances in Modeling Earth Systems

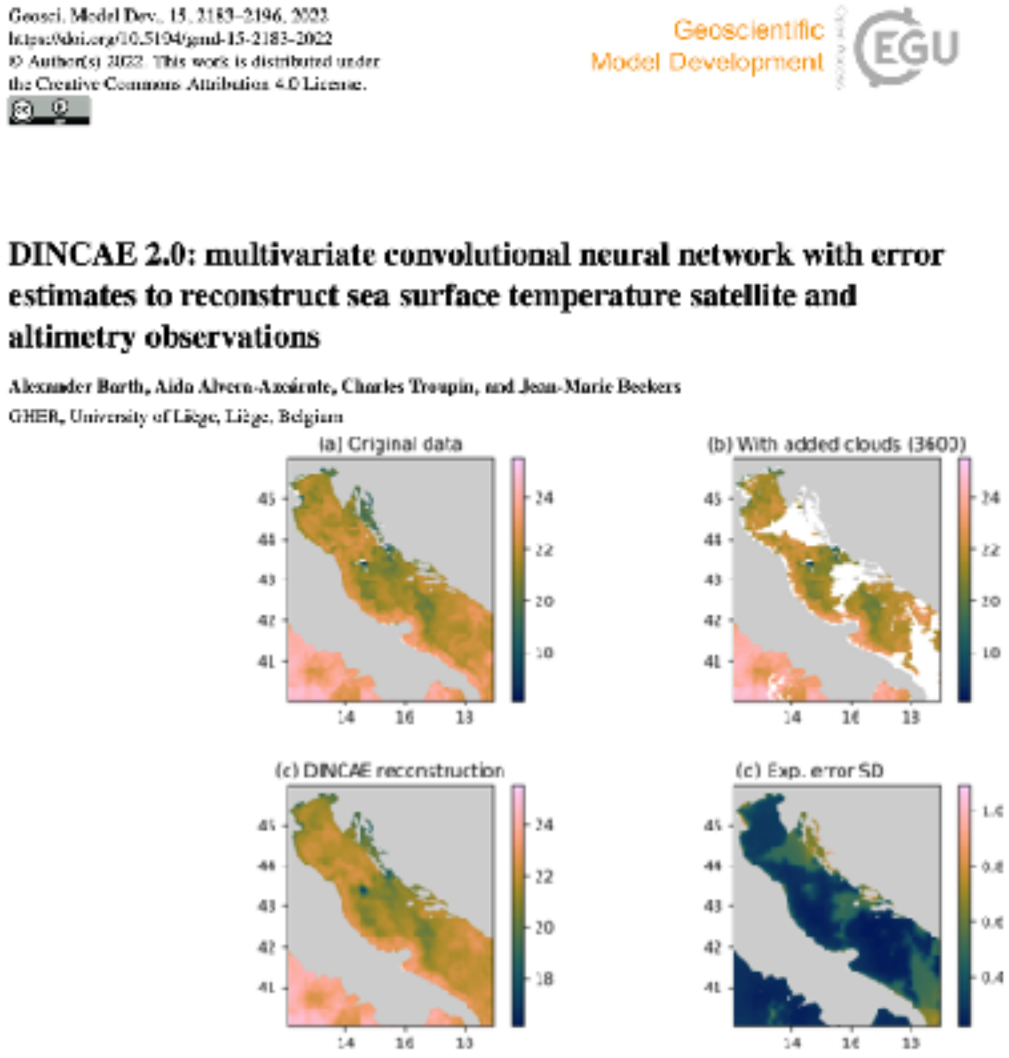
RESEARCH ARTICLE

10.1029/2019MS001965

A Deep Learning Approach to Spatiotemporal Sea Surface Height Interpolation and Estimation of Deep Currents in Geostrophic Ocean Turbulence

Georgy E. Manucharyan[1], Lia Siegelman[2], and Patrice Klein[2,3,4]

[1]School of Oceanography, University of Washington, Seattle, WA, USA, [2]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, [3]Laboratoire de Météorologie Dynamique, Ecole Normale Supérieure, CNRS, Paris, France, [4]Laboratoire d'Océanographie Physique et Spatiale, IFREMER, CNRS, Brest, France



Geosci. Model Dev., 15, 2183–2196, 2022
https://doi.org/10.5194/gmd-15-2183-2022
© Author(s) 2022. This work is distributed under the Creative Commons Attribution 4.0 License.

Geoscientific Model Development

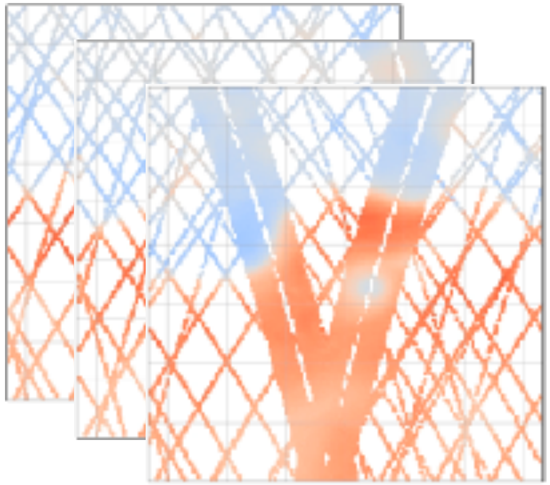
DINCAE 2.0: multivariate convolutional neural network with error estimates to reconstruct sea surface temperature satellite and altimetry observations

Alexander Barth, Aida Alvera-Azcárate, Charles Troupin, and Jean-Marie Beckers

GHER, University of Liège, Liège, Belgium

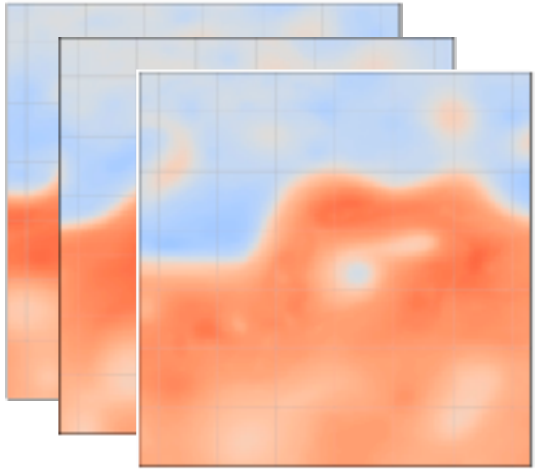
Deep learning and inverse problems

End-to-end architecture



Partial observations y

Deep learning Models



True states x

Should we reinvent the wheel ? Or can we benefit from more than 50 years of knowledge and research in signal processing, optimisation, applied math…. ?

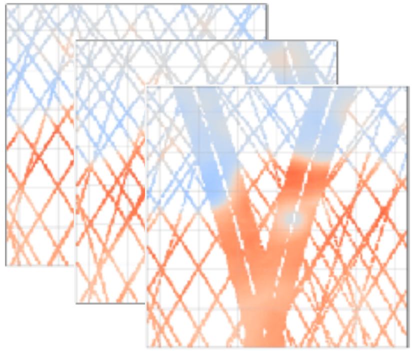
Inverse Problems in Geoscience

Mathematical formulations for inverse Problems

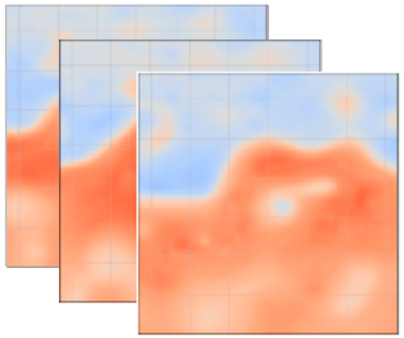
Inverse problems as learning problems

Applications to geophysical dynamics

(Variational) Data Assimilation: (Weak-Constraint) 4DVar DA



Partial observations y



True states x

State-space formulation:

$$\begin{cases} \dfrac{\partial x(t)}{\partial t} & = & \mathcal{M}\left(x(t)\right) + \eta(t) \\ \\ y(t,p) & = & x(t,p) \quad \forall t, \forall p \in \Omega_t \end{cases}$$

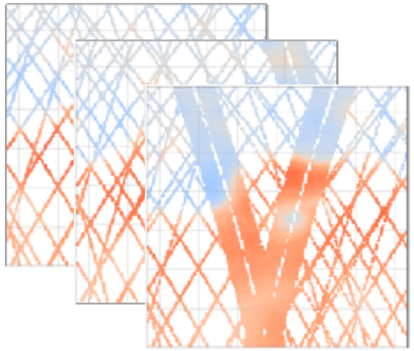
Associated variational formulation:

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

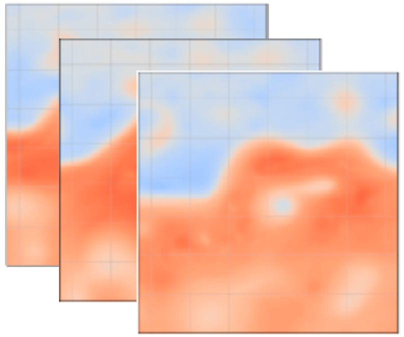
with $\quad \Phi(x)(t) = x(t-\Delta) + \int_{t-\Delta}^{t} \mathcal{M}\left(x(u)\right) du$

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

(Variational) Data Assimilation: (Weak-Constraint) 4DVar DA



Partial observations y

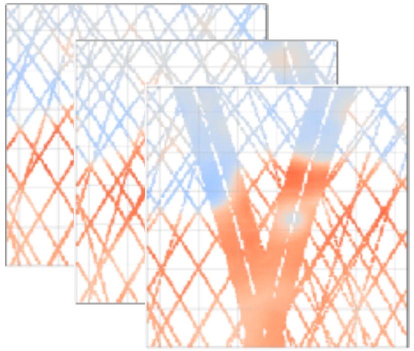


True states x

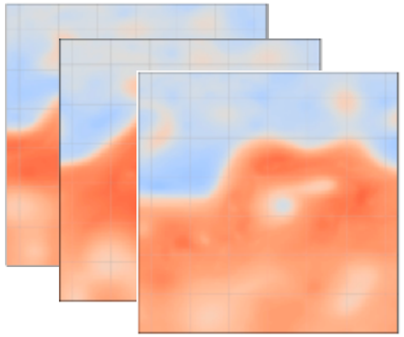
State-space formulation:

$$\left\{\begin{array}{ccl} \dfrac{\partial x(t)}{\partial t} & = & \mathcal{M}\left(x(t)\right) + \eta(t) \\ \\ y(t,p) & = & x(t,p) \quad \forall t, \forall p \in \Omega_t \end{array}\right.$$

Inverse problems stated as minimisation problems



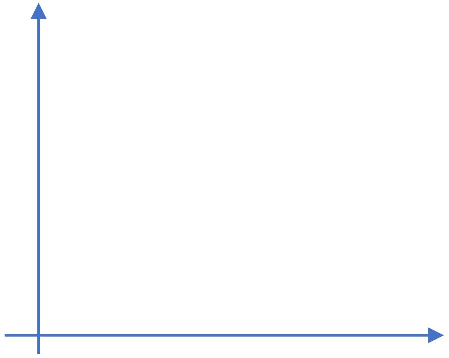
Partial observations y



True states x

Minimization problem

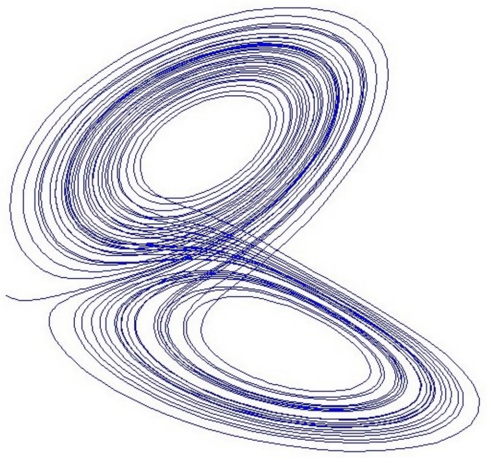
$$X = \arg\min_{X} \|Y - H(X)\|^2 + \lambda U_{reg}(X)$$



How to solve the minimization ?

Can we use Pytorch to implement the minimization ?

(Variational) Data Assimilation: (Weak-Constraint) 4DVar DA



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

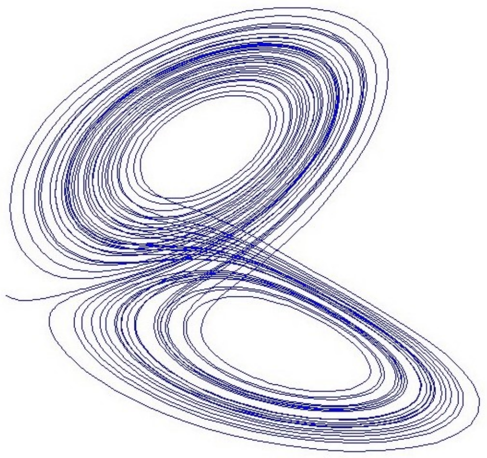
Lorenz-63 equations

Minimization problem

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

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

(Variational) Data Assimilation: (Weak-Constraint) 4DVar DA



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

Lorenz-63 equations

Minimization problem

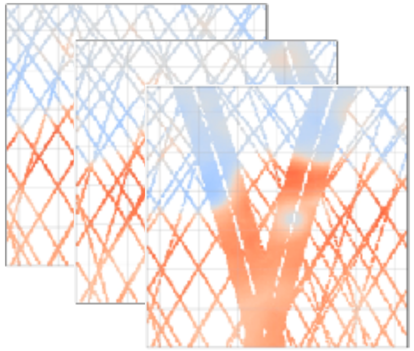
$$\arg\min_x \lambda_1 \|x - y\|_\Omega^2 + \lambda_2 \|x - \Phi(x)\|^2$$

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

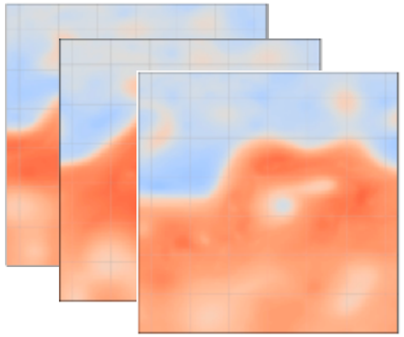
Let's try to implement this minimisation with Pytorch.

https://github.com/CIA-Oceanix/DLGD2022/blob/main/lecture-4-dl-oi-da/notebooks/notebookPyTorch_InvProb_ModelBased_L63.ipynb

4DVar Data Assimilation and Optimal Interpolation



Partial observations y

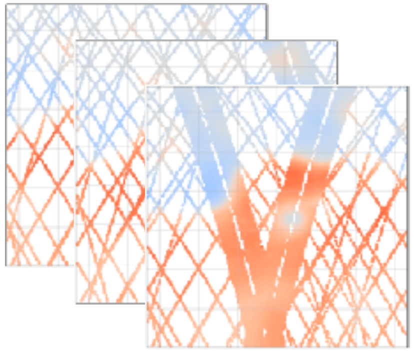


True states x

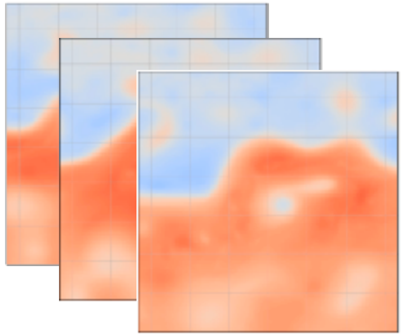
$$\left\{\begin{array}{lcl} x & \propto & \mathcal{N}\left(\mu, B\right) \\ y(p,t) & = & x(p,t) + \epsilon(t,p)\ \forall t, p \text{ with } p \in \Omega_t \\ \epsilon & \propto & \mathcal{N}\left(0, R\right) \end{array}\right.$$

$$\widehat{x} = \arg\min_{x} \log P\left(y|x\right) + \log P\left(x\right)$$

4DVar Data Assimilation and Optimal Interpolation



Partial observations y



True states x

(Weak-Constraint) 4DVar formulation

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

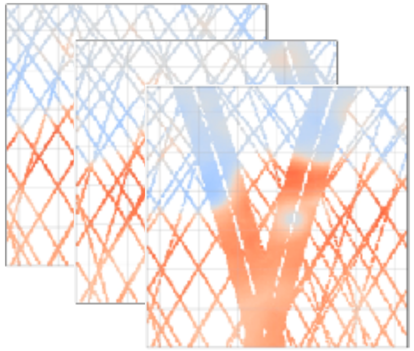
Optimal interpolation formulation

$$\arg\min_x \|y - H_\Omega \cdot x\|_R^2 + x^t B^{-1} x$$

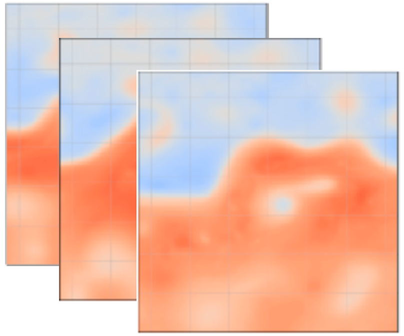
$$\arg\min_x \|y - H_\Omega \cdot x\|_R^2 + \|x - (\mathbb{I} - L) \cdot x\|^2 \text{ with } B = L^t L$$

OI solution $\widehat{x} = \mu + \mathbf{K} \cdot y \text{ with } \mathbf{K} = B H_\Omega \left(H_\Omega B H_\Omega^t + R \right)^{-1}$

Wrap-up on 4DVar DA and OI



Partial observations y



True states x

Weak-Constraint 4DVar formulation

$$\arg\min_{x} \|y - H_\Omega \cdot x\|^2 + \lambda \|x - \Phi(x)\|^2 + \nu \|x_0 - \mu_0\|^2_{B_0}$$

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

Strong-Constraint 4DVar formulation

$$\arg\min_{x_0} \|y - H_\Omega \cdot x\|^2 + \nu \|x_0 - \mu_0\|^2_{B_0} \text{ s.t. } \left\{ \begin{array}{lcl} x(t_0) & = & x_0 \\ \frac{dx}{dt} & = & \mathcal{M}\left(x(t)\right) \end{array} \right.$$

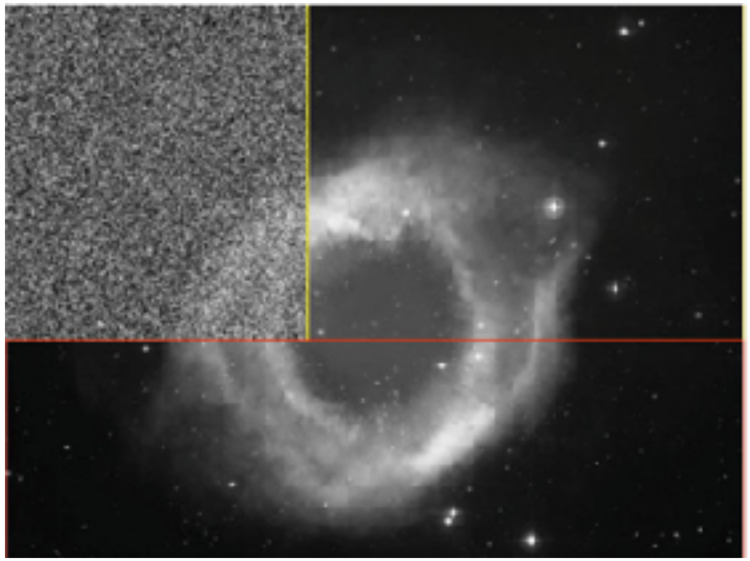
OI solution

$$\arg\min_{x} \|y - H_\Omega \cdot x\|^2_R + \|x - (\mathbb{I} - L) \cdot x\|^2 \text{ with } B = L^t L$$

Inverse problems stated as minimisation problems

Denoising problem

$$Y = X + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$



Probablistic prior

$$X \sim P_X$$

$$X = \arg\min_X \lambda \|X - Y\|^2 - log P_X(X)$$

Dictionary-based prior

$$X = D.\alpha$$

$$\widehat{x} = \arg\min_{x,\alpha} \|y - x\|^2 + \lambda \|x - D.\alpha\|^2$$

Norm-based prior $$\widehat{x} = \arg\min_x \|y - x\|^2 + \lambda \|\nabla x\|^2$$

Generic formulation $$\widehat{x} = \arg\min_x \|y - H(x)\|^2 + \lambda U_{Reg}(x)$$

Key message for (variational) Model-based Methods

- ***Solution stated as the minimisation of a variational cost***
- ***Explicit knowledge of observation model and prior***
- ***Implementation of associated minimisation algorithm***
- ***Analytical derivation of gradient operator (***e.g., Euler-Lagrange equations, Adjoint Methods***)***
- ***Implementation using DL schemes (***eg, Pytorch***) do not require the analytical gradient operator***

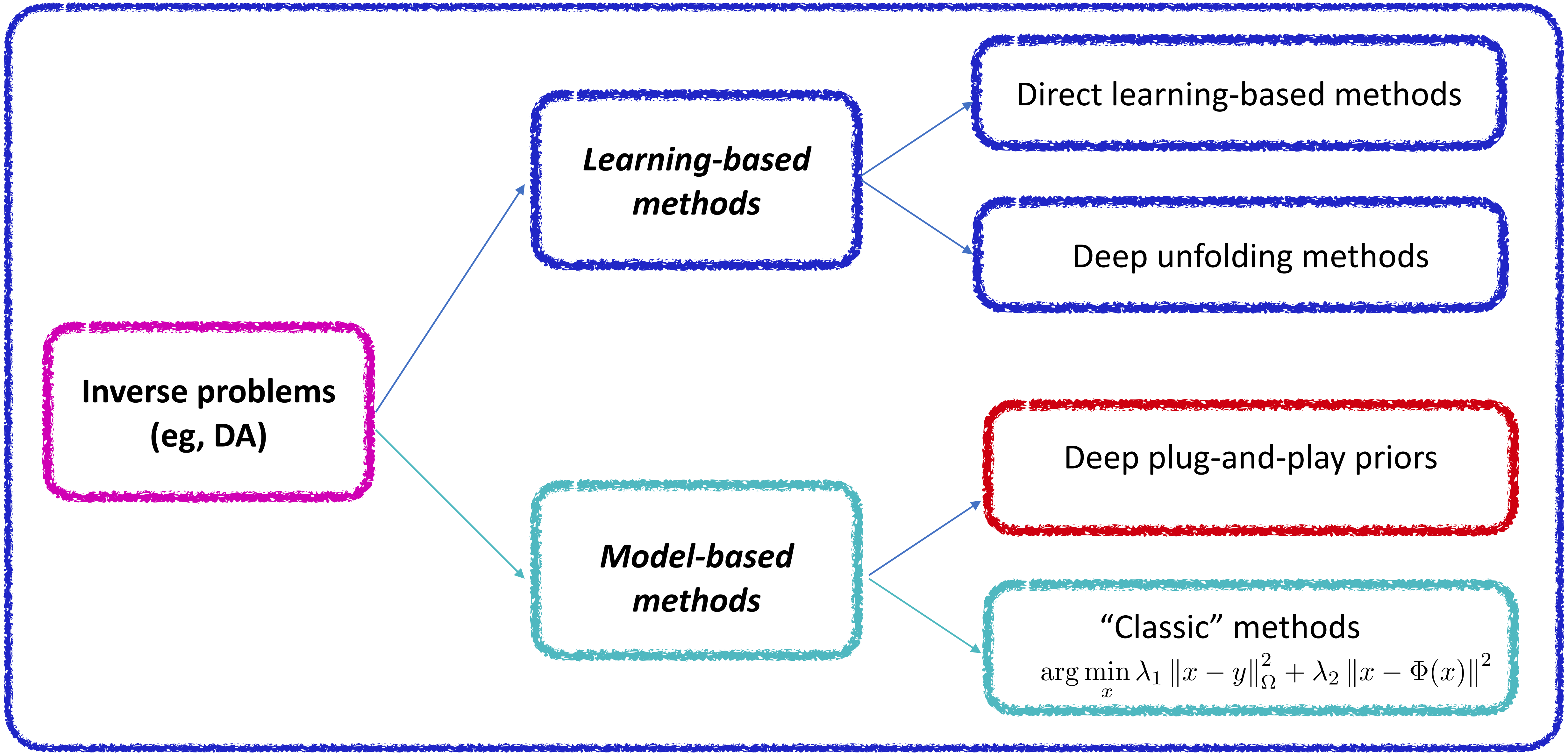
Inverse Problems in Geoscience

Mathematical formulations for inverse Problems

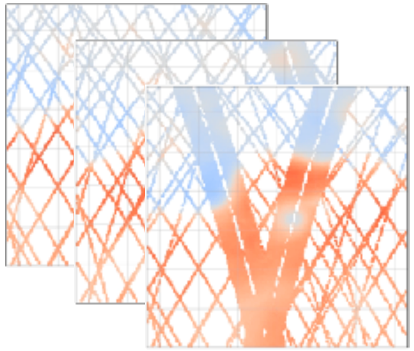
Inverse problems & Deep learning

Applications to geophysical dynamics

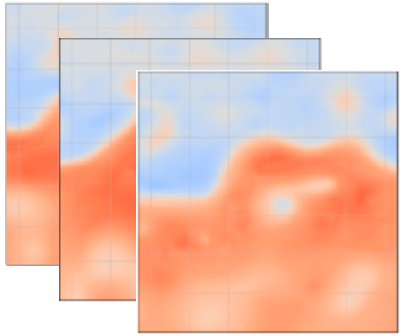
Model-driven vs. Learning-based approaches



Inverse problems using Deep plug-and-play priors



Partial observations y

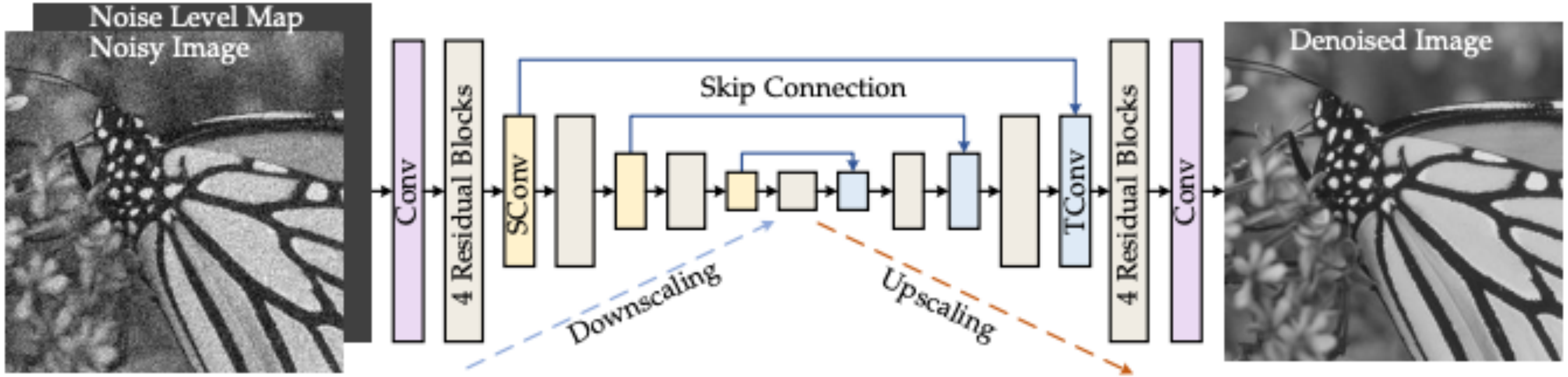


True states x

Model-based formulation with a (deep) learning-based prior

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

Trainable plug-and-play prior

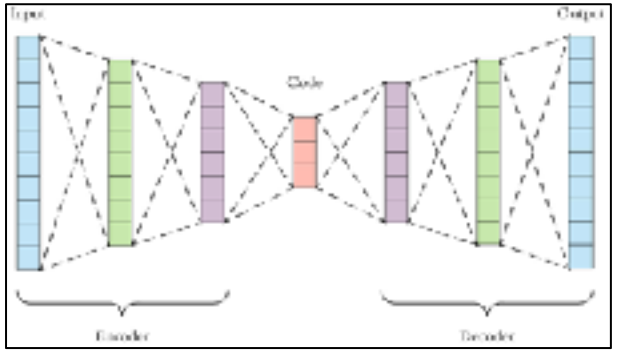


x $\qquad$ $\Phi(x)$

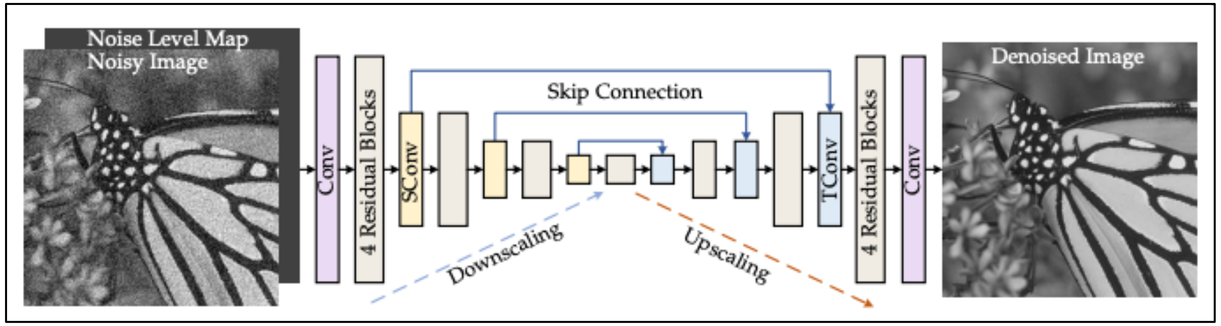
Zhang et al. Plug-and-Play Image Restoration with Deep Denoiser Prior. IEEE TPAMI, 2021. https://arxiv.org/pdf/2008.13751.pdf

Inverse problems using Deep plug-and-play priors

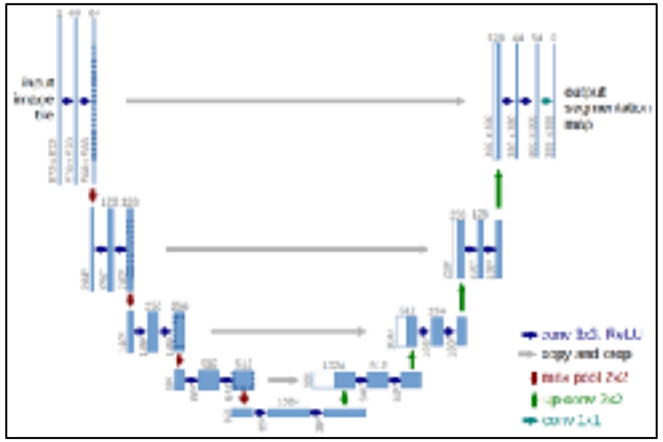
Examples of plug-and-play priors (denoiser architecture)



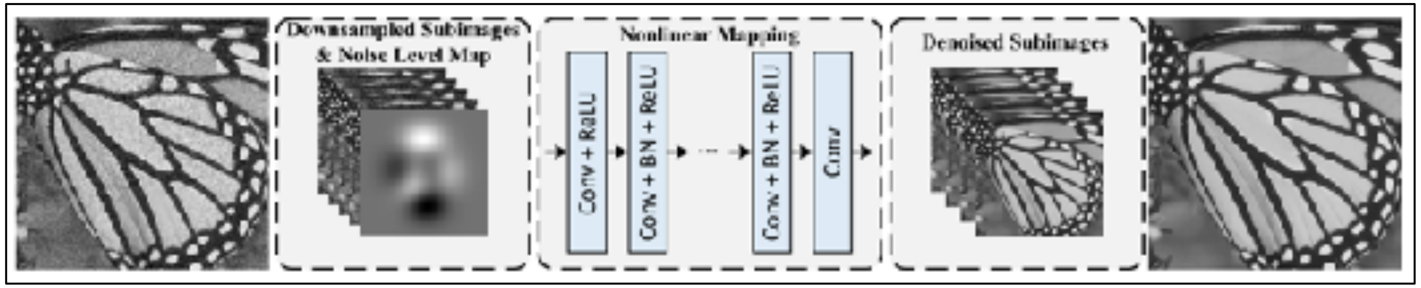
Auto-encoder



DRUNet https://arxiv.org/pdf/2008.13751.pdf

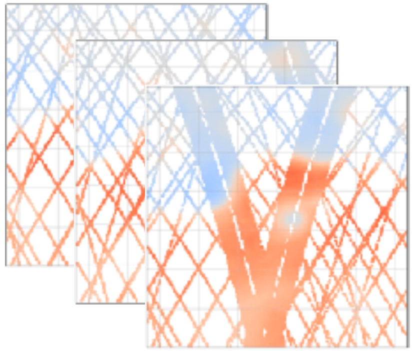


UNet https://arxiv.org/abs/1505.04597

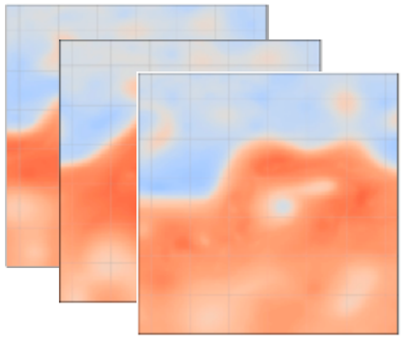


FFDNet https://arxiv.org/pdf/1710.04026.pdf

Inverse problems using Deep plug-and-play priors



Partial observations y



True states x

Model-based formulation

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

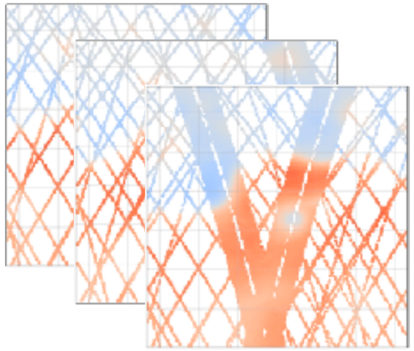
Trainable plug-and-play prior

Use of trainable priors but no actual learning specifically designed for the targeted inverse problem

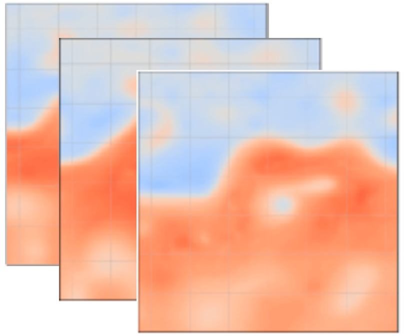
Let’s go and test it using a PCA-based prior

https://github.com/CIA-Oceanix/DLGD2022/blob/main/lecture-4-dl-oi-da/notebooks/notebookPyTorch_InvProb_ModelBased_L63.ipynb

Inverse problems: from plug-and-play to implicit priors



Partial observations y



True states x

Inverse problem with plug-and-play priors

Trainable plug-and-play prior

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

Inverse problem with a prior in latent space

Pre-trained decoder

$$\arg\min_{z} \lambda_1 \|y - x\|_{\Omega}^{2} \text{ s.t. } x = \mathcal{D}(z)$$

Inverse problem with deep image prior [*Ulyanov'17*]

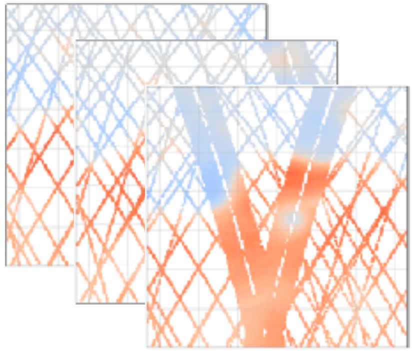
$$\arg\min_{\theta} \lambda_1 \|y - x\|_{\Omega}^{2} \text{ s.t. } x = \mathcal{D}_{\theta}(z) \text{ and } z \sim \mathcal{N}(0, \mathbb{I})$$

https://arxiv.org/abs/1711.10925

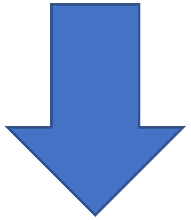
Inverse problem with implicit neural representation

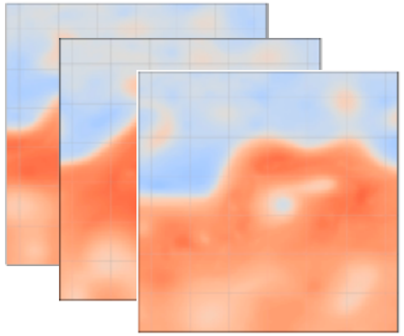
$$\arg\min_{\theta} \lambda_1 \|y - x\|_{\Omega}^{2} \;\text{ s.t. } \forall p,\; x(p) = \mathcal{D}_{\theta}(p)$$

Inverse problems: from plug-and-play to implicit priors



Partial observations y





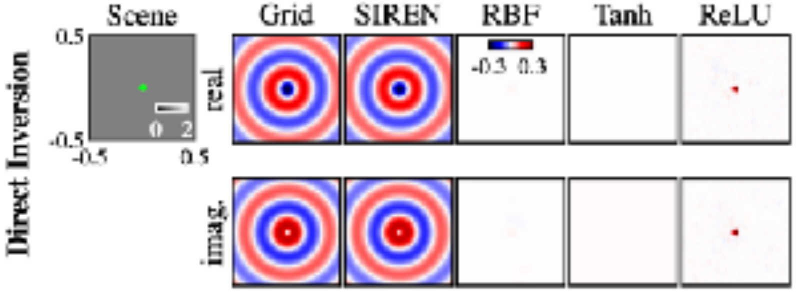
True states x

Inverse problem with implicit neural representations

$$\forall p,\ x(p) = \mathcal{D}_\theta(p)$$

Space-continuous (grid-free) representations of nD tensors

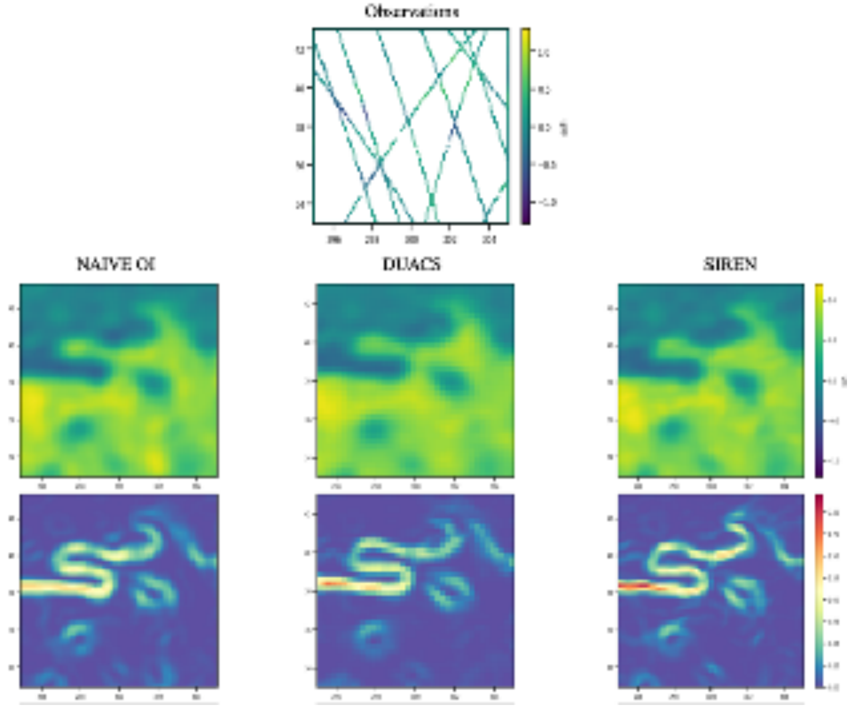
SIREN: implicit representations with periodic activations



https://www.vincentsitzmann.com/siren/

Application to SSH mapping

SIREN architecture / unsupervised learning



Jhonson et lal., ML4PS'22

https://arxiv.org/abs/2211.10444

Key messages for (variational) Model-based Methods

- ***Solution stated as the minimisation of a variational cost***
- ***Explicit knowledge of observation model and prior***
- ***Implementation of associated minimisation algorithm***
- ***Analytical derivation of gradient operator (***e.g., Euler-Lagrange equations, Adjoint Methods***)***
- ***Implementation using DL schemes (***eg, Pytorch***) do not require the analytical gradient operator***
- ***Possible extension to pre-trained plug-and-play and implicit priors***

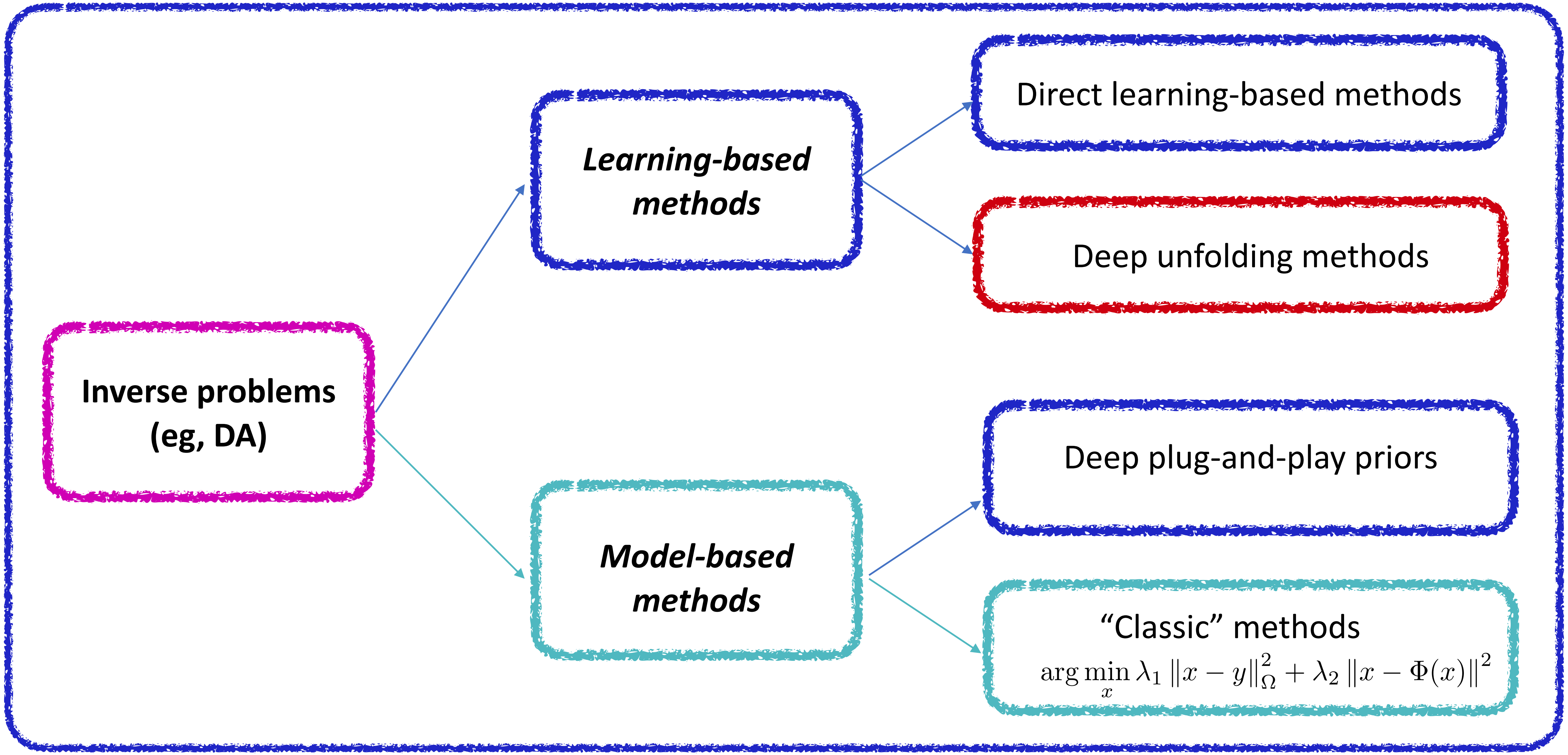
Inverse Problems in Geoscience

Mathematical formulations for inverse Problems

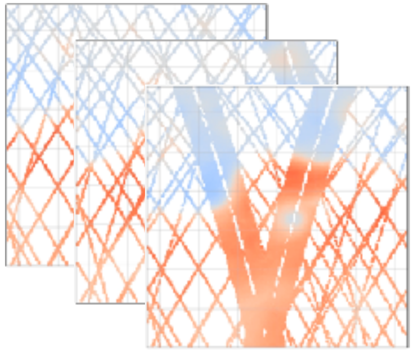
Inverse problems & Deep learning

Applications to geophysical dynamics

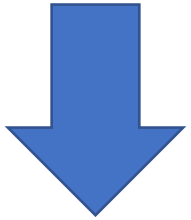
Model-driven vs. Learning-based approaches

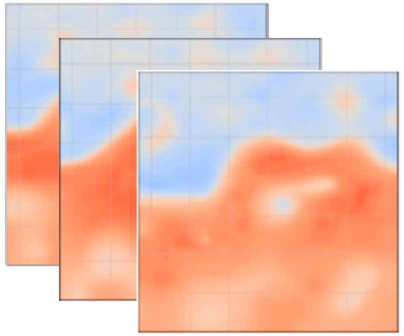


Can we relate end-to-end learning and model-based schemes?



Partial observations y





True states x

The example of the Optimal Interpolation

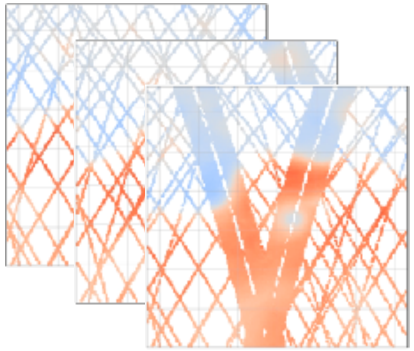
$$\arg\min_{x} \|y - H_\Omega \cdot x\|_R^2 + \|x - (\mathbb{I} - L) \cdot x\|^2 \text{ with } B = L^t L$$

Associated optimality criterion (bi-level formulation)

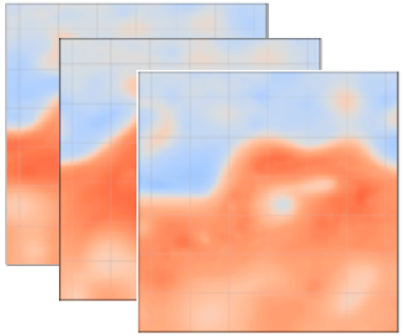
$$\min_{L} \mathbb{E}\left(\|\widehat{x} - x^{true}\|^2\right)$$

$$\text{s.t.} \quad \widehat{x} = \arg\min_{x} \|y - H_\Omega \cdot x\|^2 + \|x - \Phi_L(x)\|^2$$

Can we relate end-to-end learning and model-based schemes?



Partial observations y



True states x

The example of the Optimal Interpolation

$$\arg\min_x \|y - H_\Omega \cdot x\|_R^2 + \|x - (\mathbb{I} - L) \cdot x\|^2 \text{ with } B = L^t L$$

Associated optimality criterion (bi-level formulation)

$$\min_L \mathbb{E}\left(\|\widehat{x} - x^{true}\|^2\right)$$

$$\text{s.t.} \quad \widehat{x} = \arg\min_x \|y - H_\Omega \cdot x\|^2 + \|x - \Phi_L(x)\|^2$$

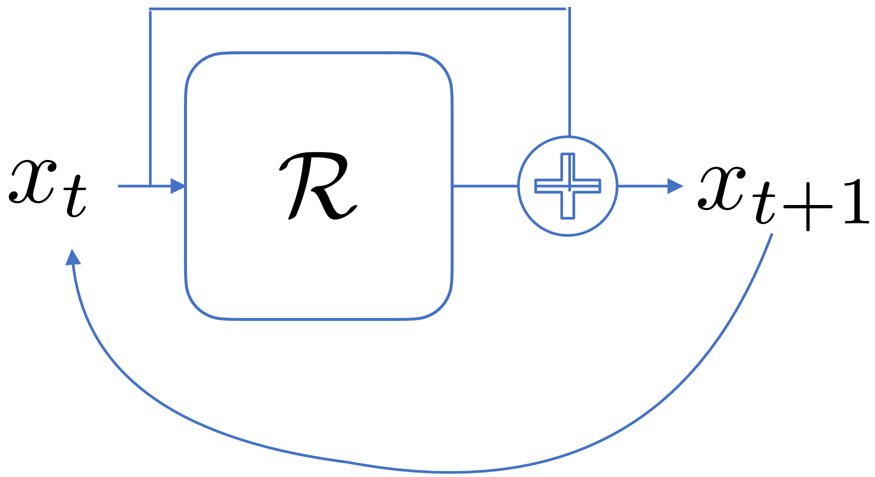
No similar property in general for non-linear/non-quadratic formulations

Folded vs. Unfolded Representations

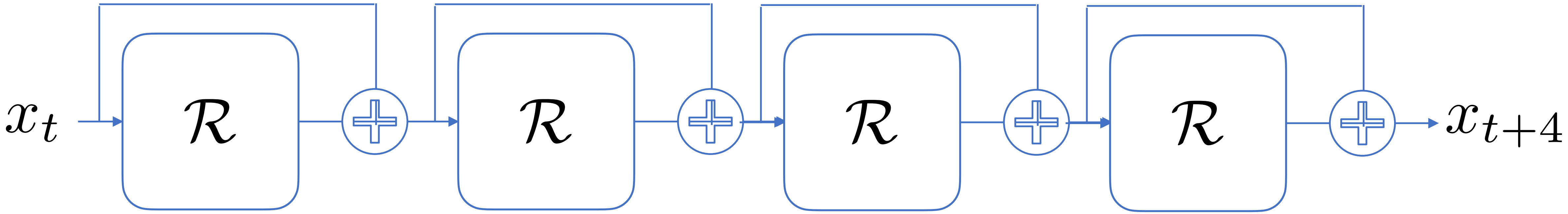
An example with a ResNet

$$x_{t+1} = x_t + \mathcal{R}\left(x_t\right)$$

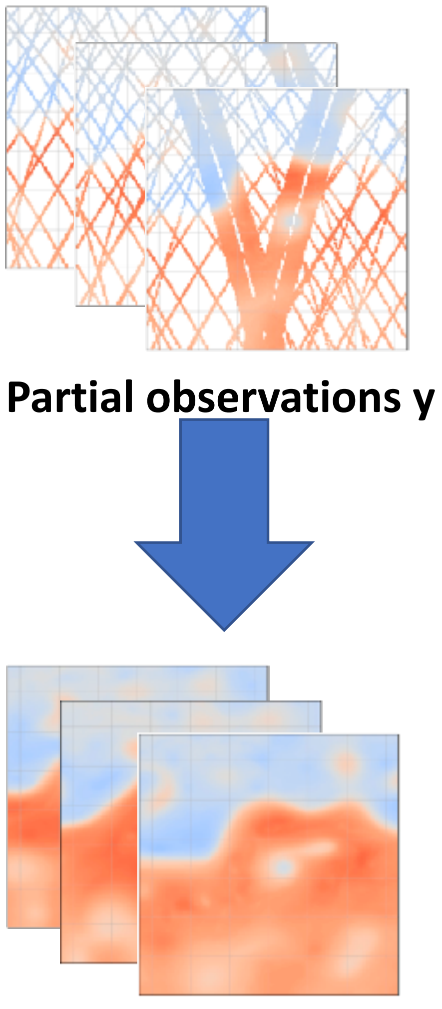
Folded Representation



Unfolded Representation



Inverse problems using Deep unfolding schemes



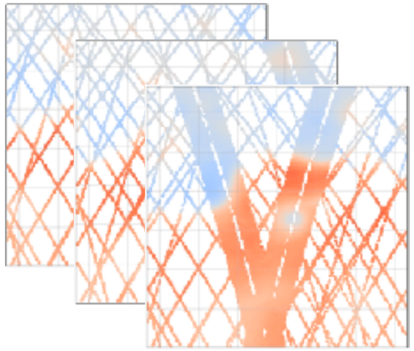
Basic idea: exploit knowledge on optimisation algorithms for inverse problems

- **Many schemes involve iterative algorithms**
- **On may unfold an Iterative procedure to define a deep learning architecture**

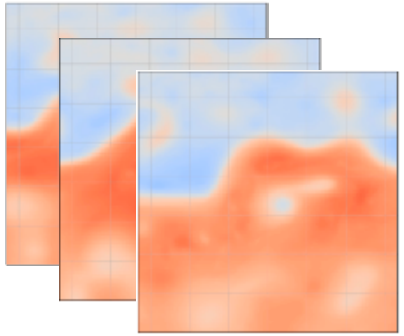
Examples

- **Image denoising/Unfolding of reaction-diffusion schemes** (e.g., Chen et al., 2015)
- **Medical imaging/Unfolding of ADMM schemes** (e.g., Yang et al., 2016)
- **Interpolation/Unfolding of fixed-point algorithms** (e.g., Fablet et al., 2020)
- **DA/Deep unfolding of sequential DA algorithms** (e.g., Boudier et al., 2020)

Data Assimilation using Deep unfolding schemes



Partial observations y



True states x

An example using fixed-point algorithms for interpolation problems

$$\arg\min_x \|x - \Phi(x)\|^2 \ \text{ subject to } y_\Omega = x_\Omega$$

Iterative step
$$\begin{cases} \tilde{x}^{(k)} & = & \Phi\left(x^{(k)}\right) \\ x^{(k+1)}\left(\Omega\right) & = & y^{(k)}\left(\Omega\right) \\ x^{(k+1)}\left(\overline{\Omega}\right) & = & \tilde{x}^{(k)}\left(\overline{\Omega}\right) \end{cases}$$

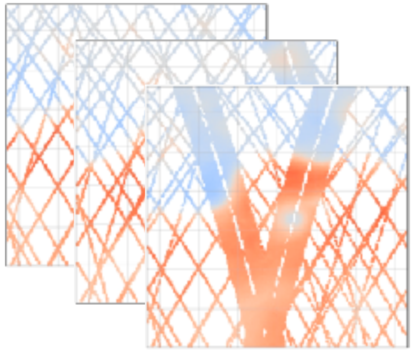
Projection using Φ

Keep observed data

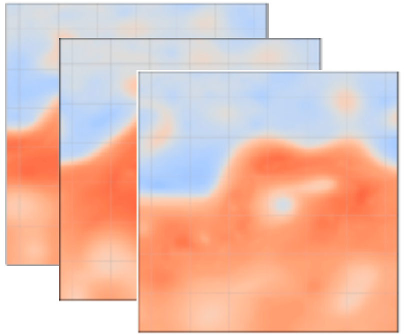
Update unobserved data

Associated Deep unfolding scheme

Inverse problems using Deep unfolding schemes



Partial observations y



True states x

An example using fixed-point algorithms for interpolation problems

$$\arg\min_x \|x - \Phi(x)\|^2 \text{ subject to } y_\Omega = x_\Omega$$

Iterative step

$$\begin{cases} \tilde{x}^{(k)} = \Phi\left(x^{(k)}\right) & \textbf{Projection using } \Phi \\ x^{(k+1)}(\Omega) = y^{(k)}(\Omega) & \textbf{Keep observed data} \\ x^{(k+1)}(\overline{\Omega}) = \tilde{x}^{(k)}(\overline{\Omega}) & \textbf{Update unobserved data} \end{cases}$$

Associated Deep unfolding scheme

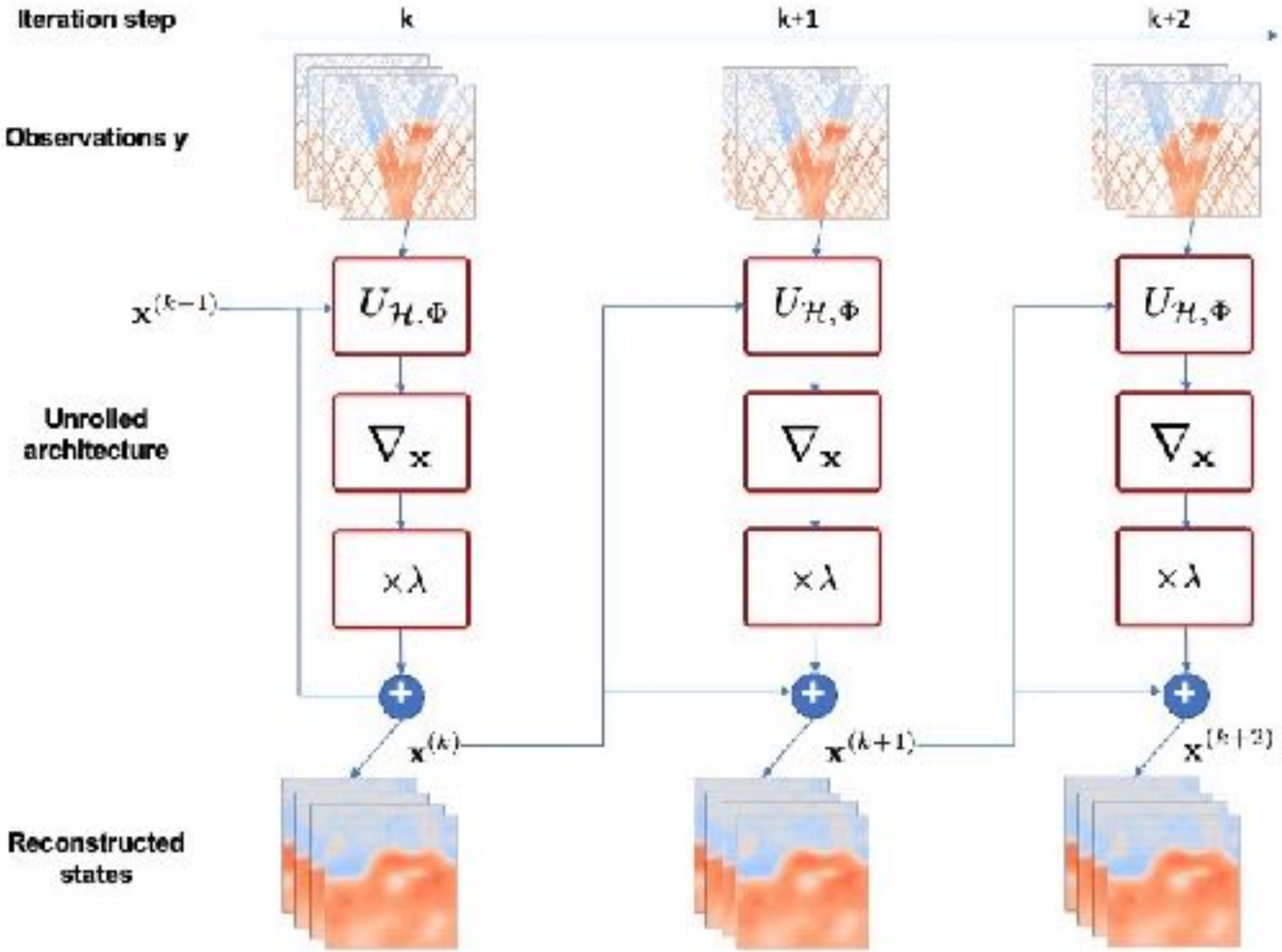
https://github.com/CIA-Oceanix/DLGD2021/edit/main/lecture-5-inverse-problems/notebookPyTorch_InvProb_LearningBased_UnrollingFixedPoint_L63_Students.ipynb

Data Assimilation using Deep unfolding schemes

Deep unfolding of 4DVar DA schemes (e.g., 4DVar-WC) [Fablet et al., 2021]

Unfolded architecture of a 4DVar-WC scheme

$$\min_{\mathbf{x}} \underbrace{\|\mathbf{x} - \mathcal{H}(\mathbf{x})\|^2 + \nu \|\mathbf{x} - \Phi(\mathbf{x})\|^2}_{U_{\mathcal{H},\Phi}}$$



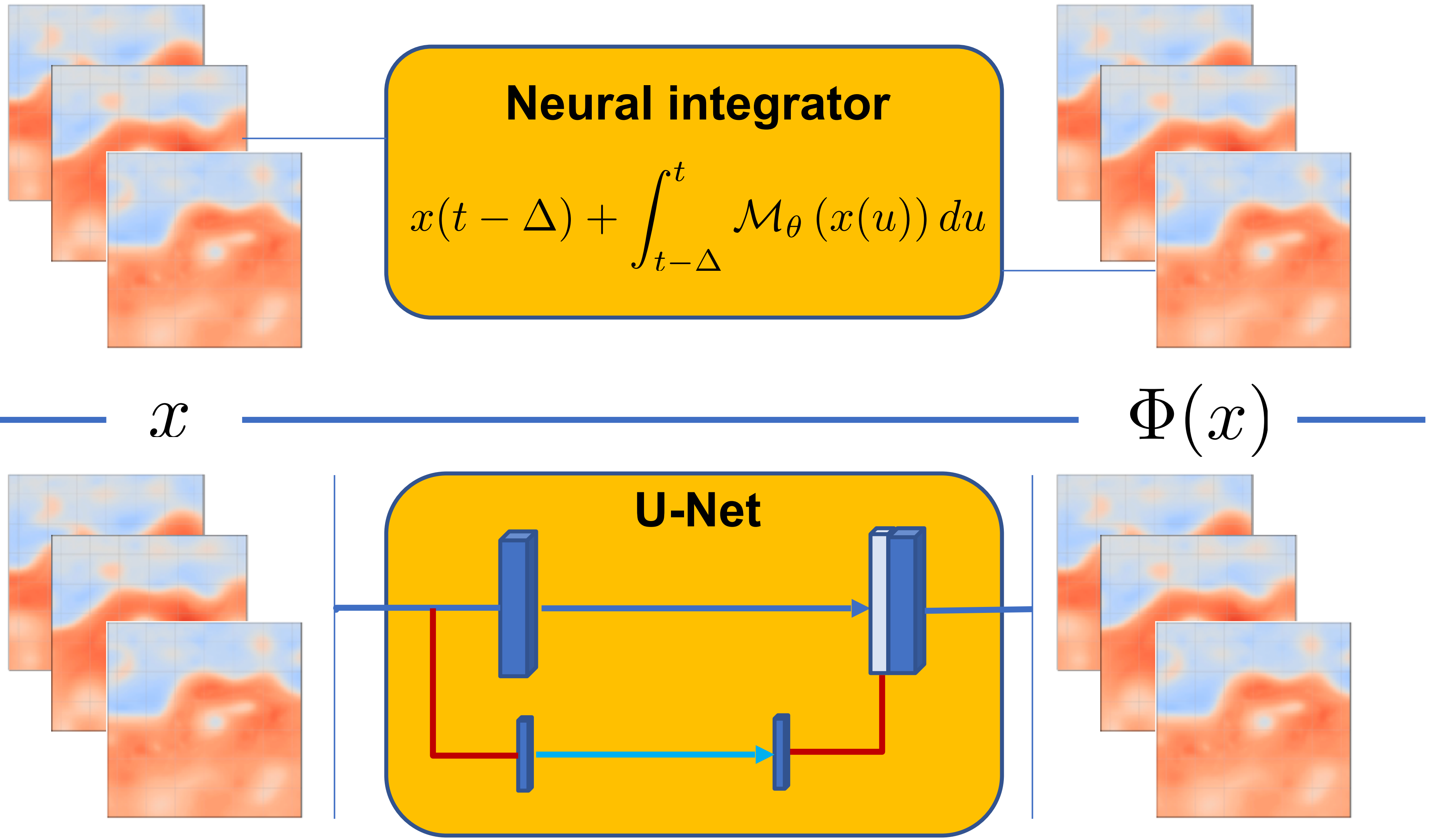
https://doi.org/10.1029/2021MS002572

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

Parameterization using (learnable) ODE operator

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

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



Data Assimilation using Deep unfolding schemes

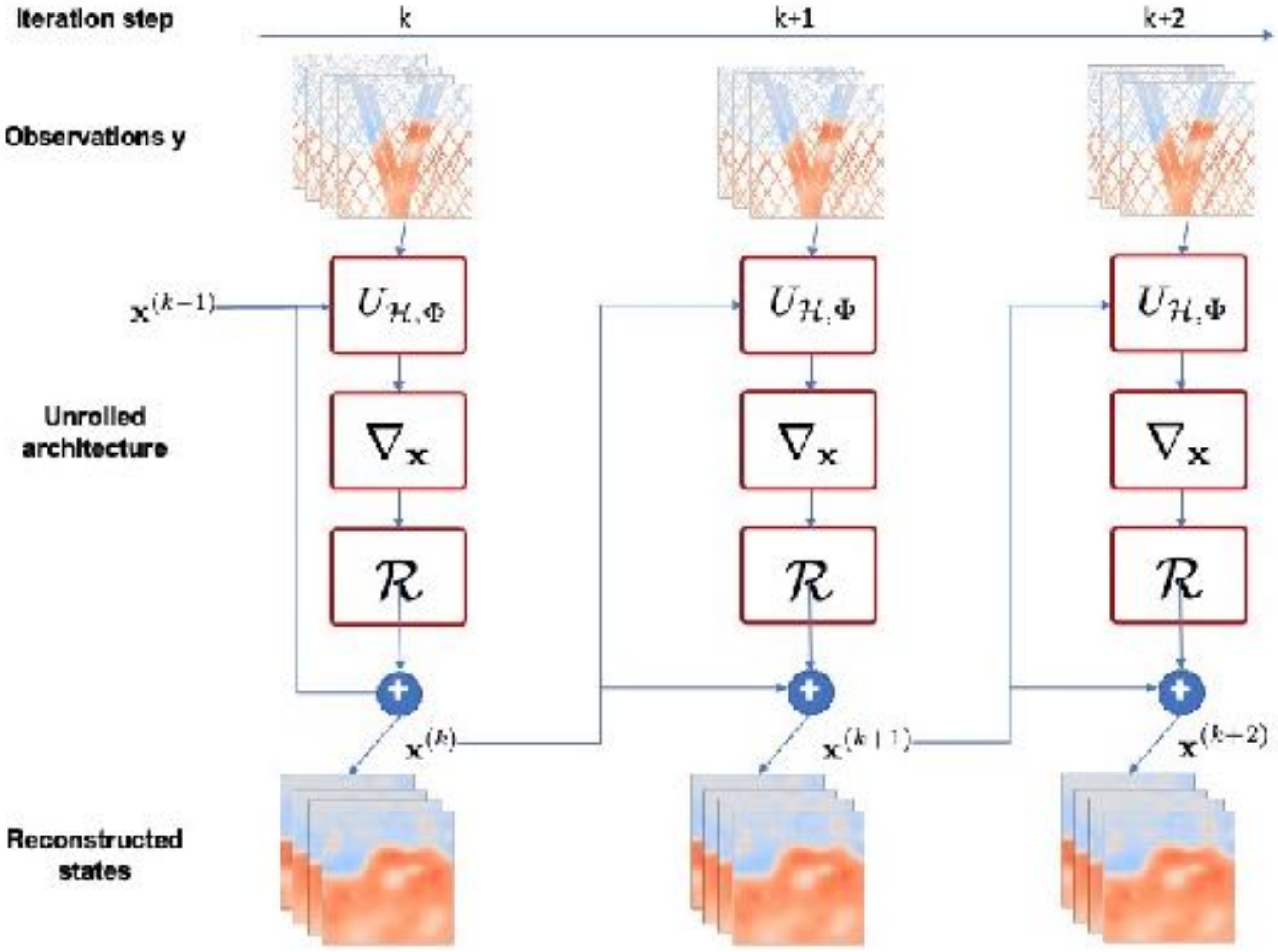
Deep unfolding of 4DVar DA schemes (e.g., 4DVar-WC) [Fablet et al., 2021]

Unfolded architecture of a 4DVarNet scheme

$$\min_{\mathbf{x}} \underbrace{\|\mathbf{x} - \mathcal{H}(\mathbf{x})\|^2 + \nu \|\mathbf{x} - \Phi(\mathbf{x})\|^2}_{U_{\mathcal{H},\Phi}}$$

Iterative gradient-based update

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k+1)} + \mathcal{R}\left(\nabla_{\mathbf{x}} U_{\mathcal{H},\Phi}(\mathbf{x},\mathbf{y})\right)$$



https://doi.org/10.1029/2021MS002572

Data Assimilation using Deep unfolding schemes

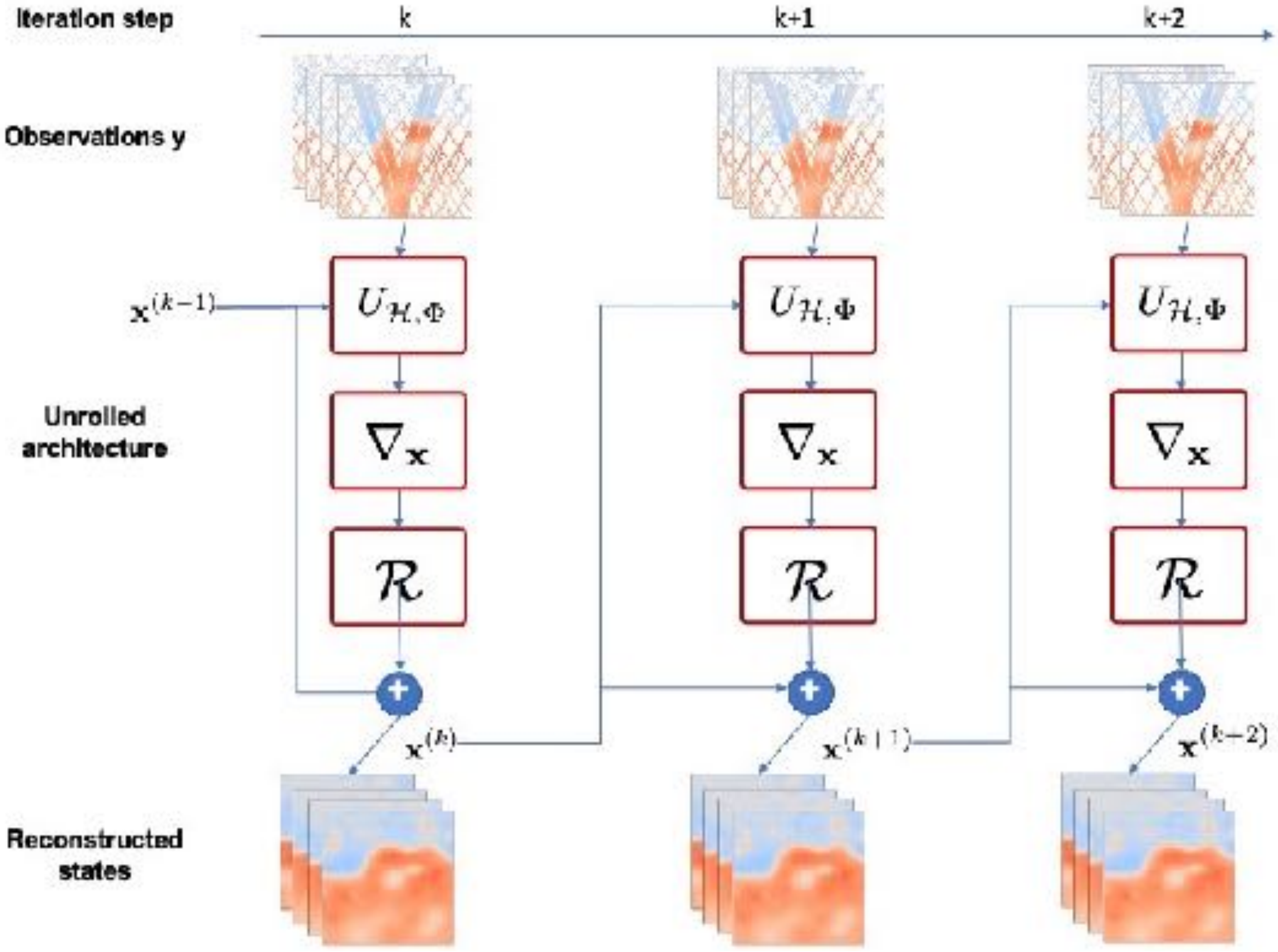
Deep unfolding of 4DVar DA schemes (e.g., 4DVar-WC) [Fablet et al., 2021]

Unfolded architecture of a 4DVarNet scheme

$$\min_{\mathbf{x}} \underbrace{\|\mathbf{x}-\mathcal{H}(\mathbf{x})\|^2+\nu\,\|\mathbf{x}-\Phi(\mathbf{x})\|^2}_{U_{\mathcal{H},\Phi}}$$

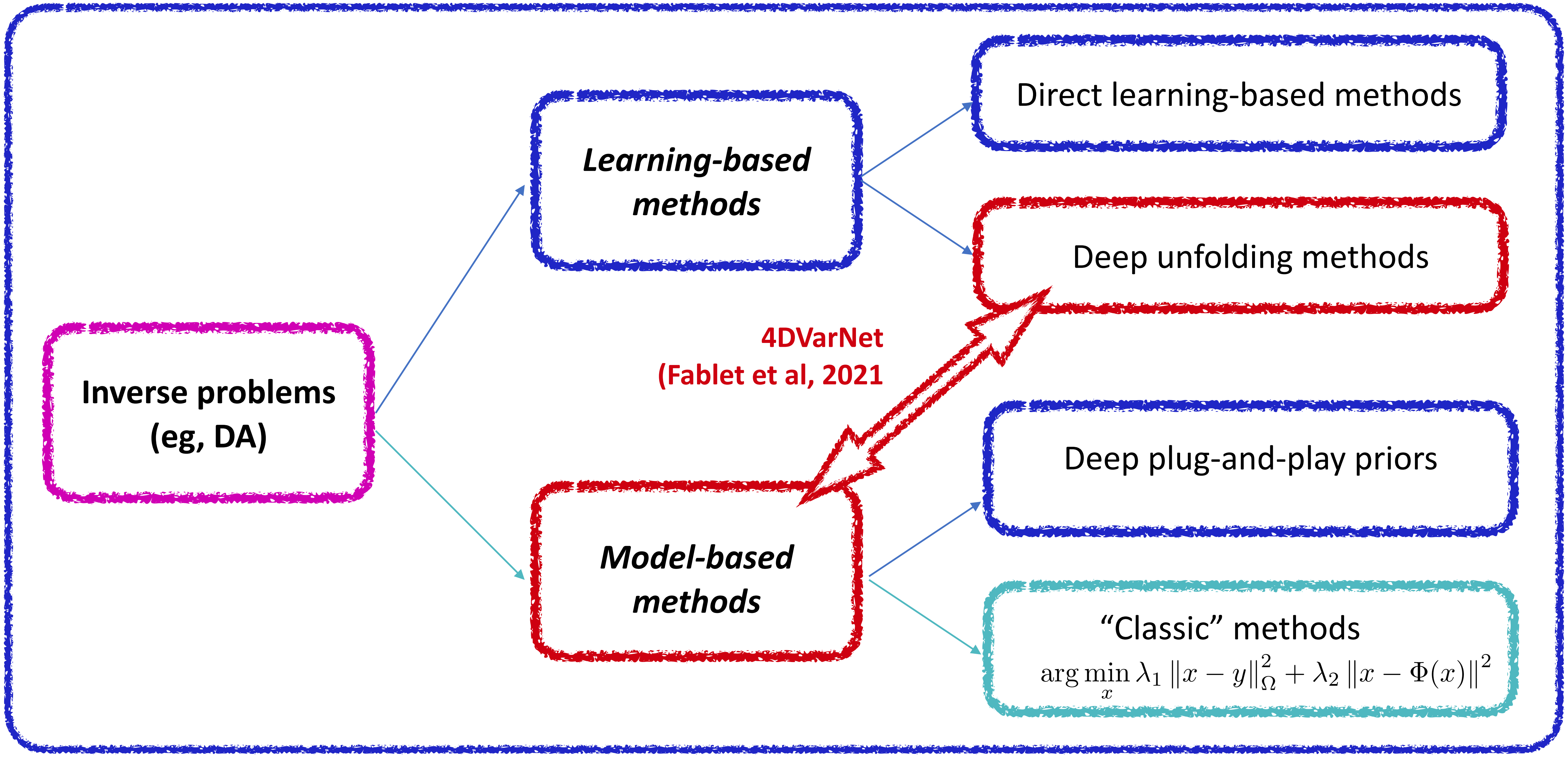
Iterative gradient-based update

$$\mathbf{x}^{(k+1)}=\mathbf{x}^{(k+1)}+\mathcal{R}\left(\nabla_{\mathbf{x}}U_{\mathcal{H},\Phi}(\mathbf{x},\mathbf{y})\right)$$



https://doi.org/10.1029/2021MS002572

Model-driven vs. Learning-based approaches



4DVarNet: Trainable 4DVar Models and Solvers

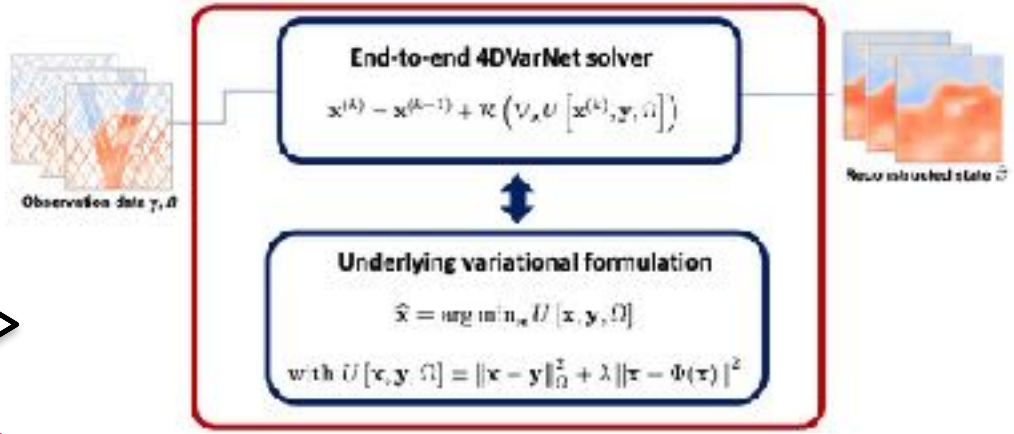
From a Variational DA formulation

$$\widehat{x} = \arg\min_x \|y - H(x)\|^2 + \lambda\|x - \Phi(x)\|^2$$

Trainable variational model

Trainable gradient-based solver

Associated end-to-end scheme



Which training loss ?

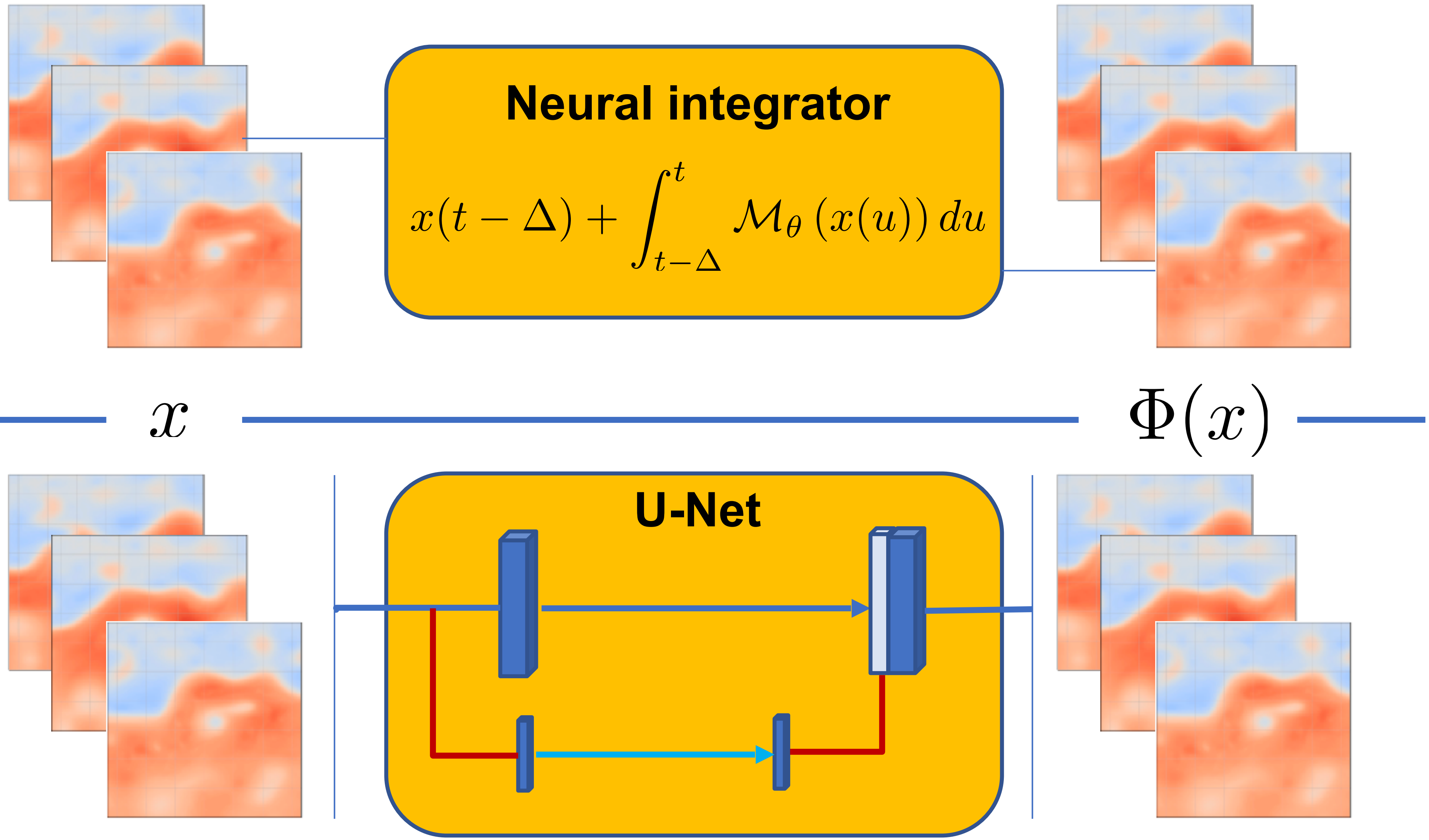
Which parameterisation for operator Φ ?

4DVarNet architecture: projection operator Φ

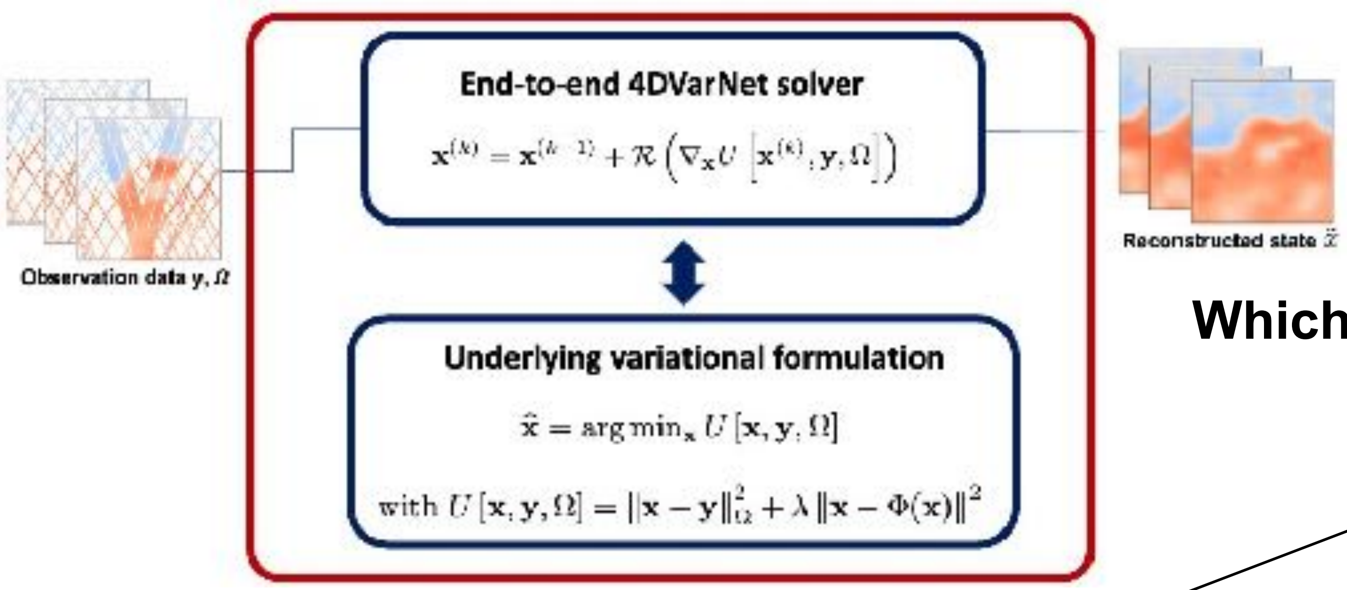
Parameterization using (learnable) ODE operator

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

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



Model-based vs. Learning-based 4DVar DA: Unsupervised vs. Supervised scheme



Which training loss for 4DVarNet scheme ?

Unsupervised loss

$$\mathcal{L}(x, y) = \|x - y\|^2 + \lambda \|x - \Phi(x)\|^2$$

Supervised loss

$$\mathcal{L}\left(x, x^{true}\right) = \left\|x - x^{true}\right\|^2$$

Regularisation loss

$$\mathcal{L}_{Reg}\left(x, x^{true}\right) = \|x - \Phi(x)\|^2 + \left\|x^{true} - \Phi(x^{true})\right\|^2$$

4DVarNet: Trainable 4DVar Models and Solvers

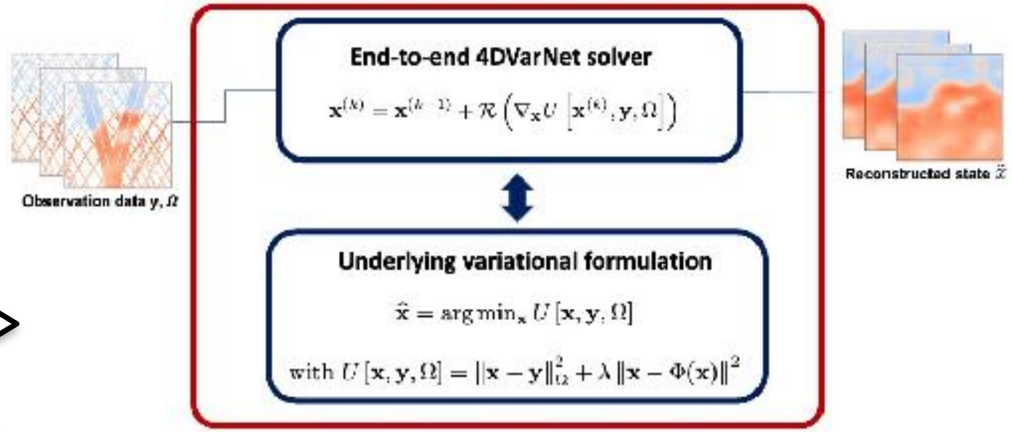
From a Variational DA formulation

$$\widehat{x} = \arg\min_{x} \|y - H(x)\|^2 + \lambda \|x - \Phi(x)\|^2$$

Trainable variational model

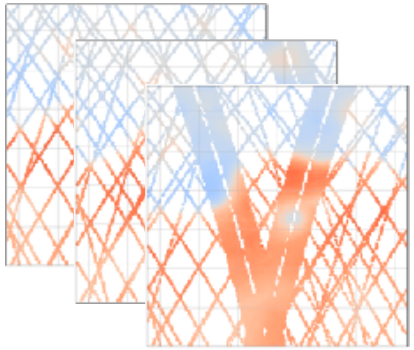
Trainable gradient-based solver

Associated end-to-end scheme

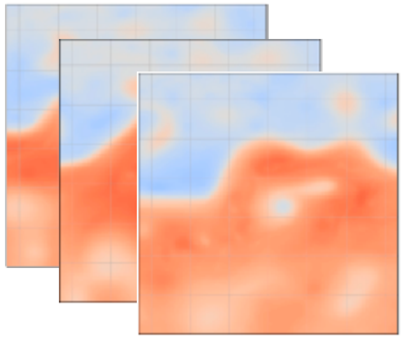


https://github.com/CIA-Oceanix/DLGD2022/blob/main/lecture-4-dl-oi-da/notebooks/notebookPyTorch_InvProb_LearningBased_4DVarNet_L63.ipynb

Model-based vs. Learning-based 4DVar DA



Partial observations y



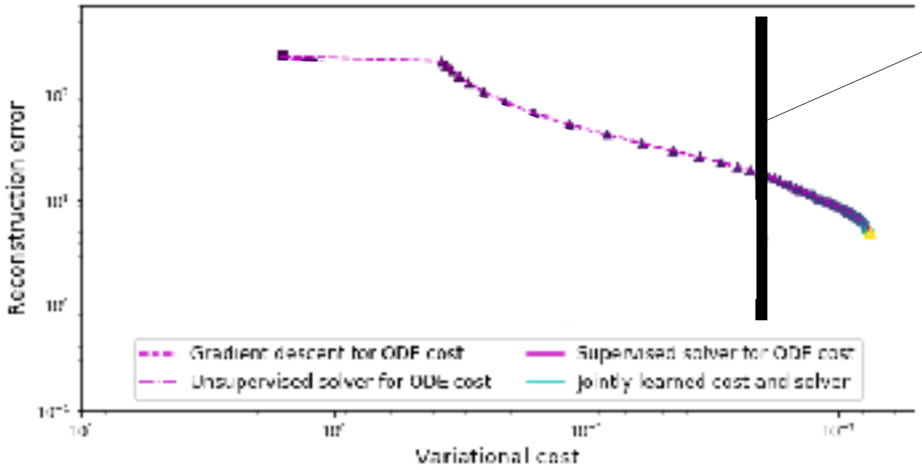
True states x

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

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

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

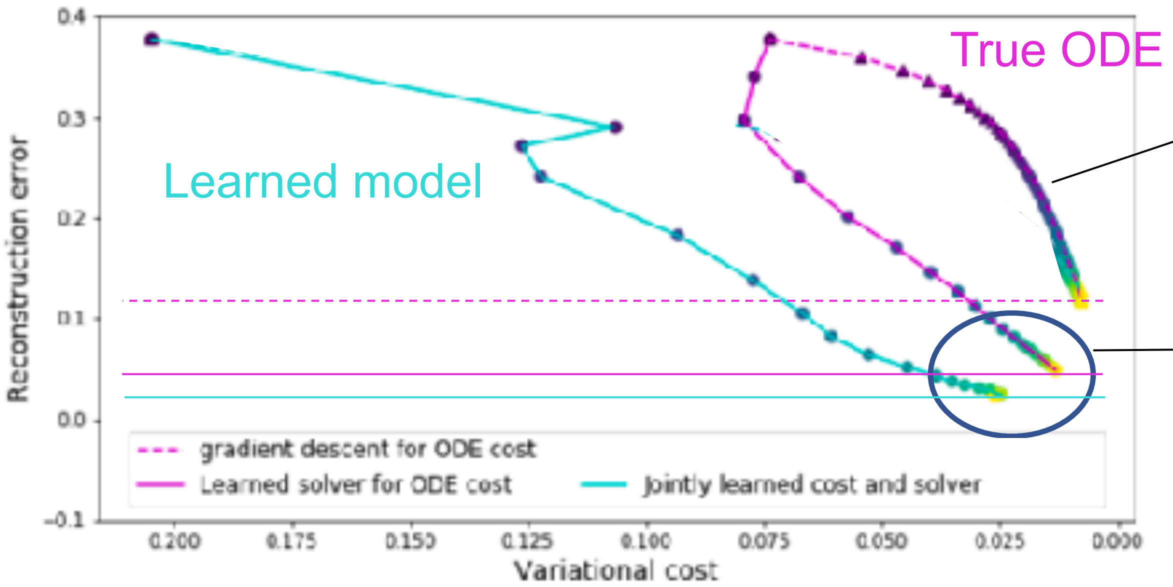
No control on the reconstruction error $x^{true} \neq \arg\min_{x} U_{\Phi}\left(x^{(k)}, y, \Omega\right)$



Variational cost for the true state

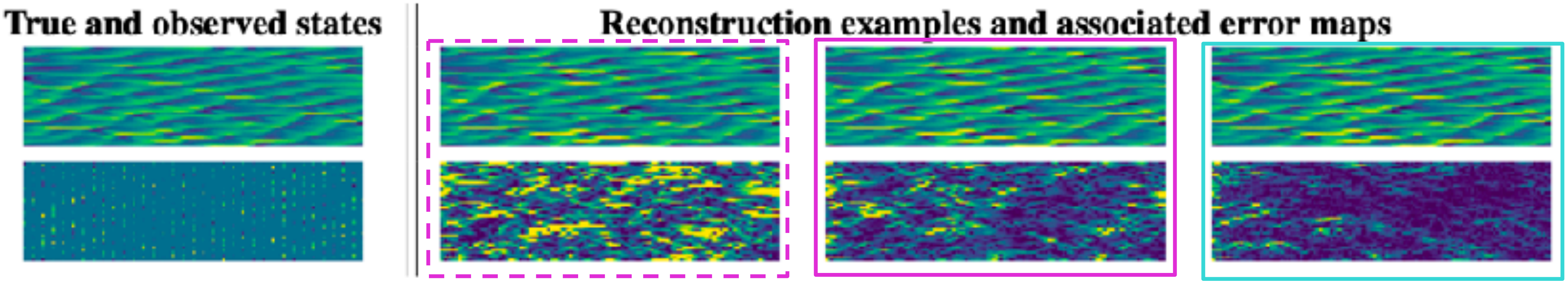
End-to-end learning for inverse problems (Fablet etal., 2020)

Illustration on Lorenz-96 dynamics (Bilinear ODE)



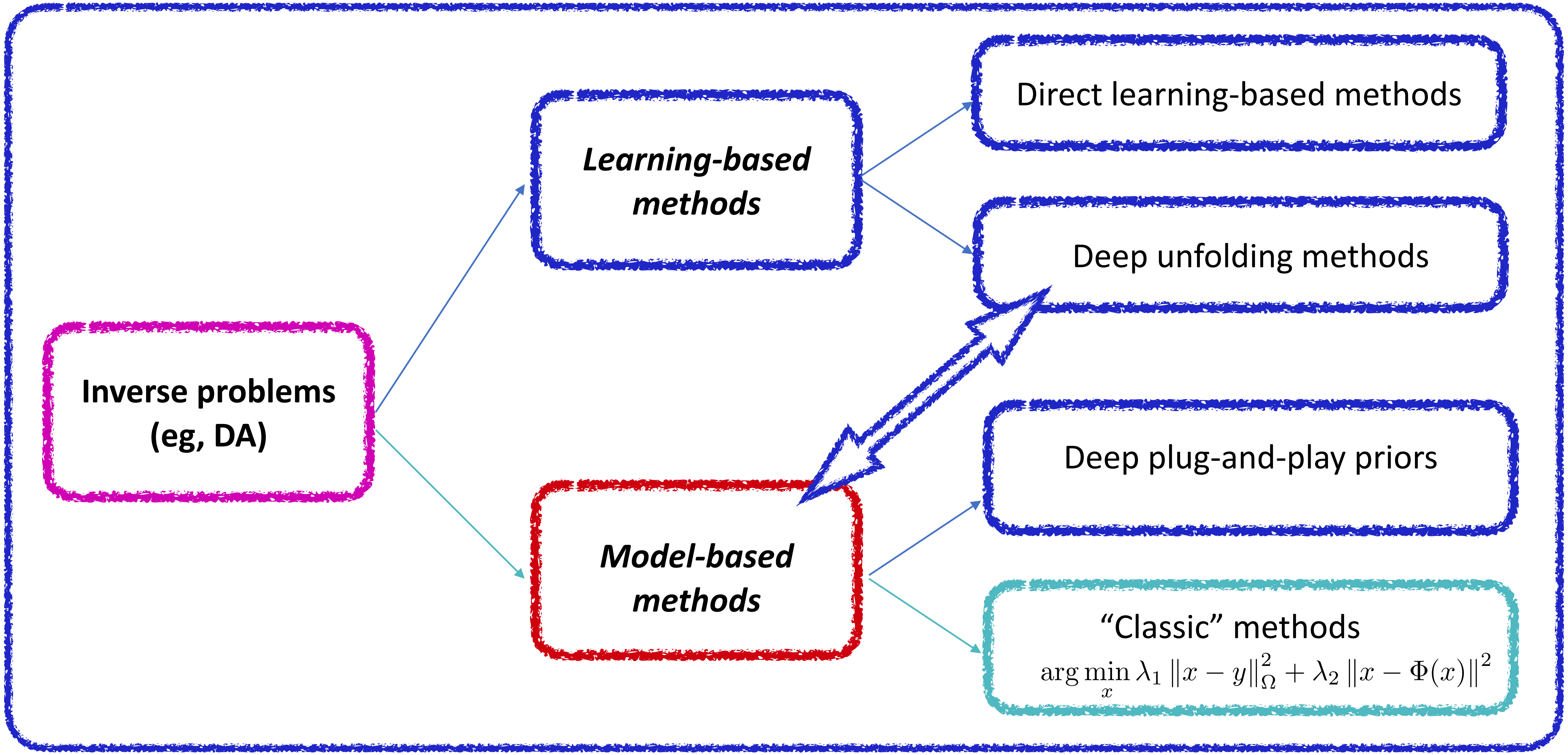
Non-supervised setting
(classic variational minimisation)

Supervised settings



Summary on Inverse Problems and Deep Learning

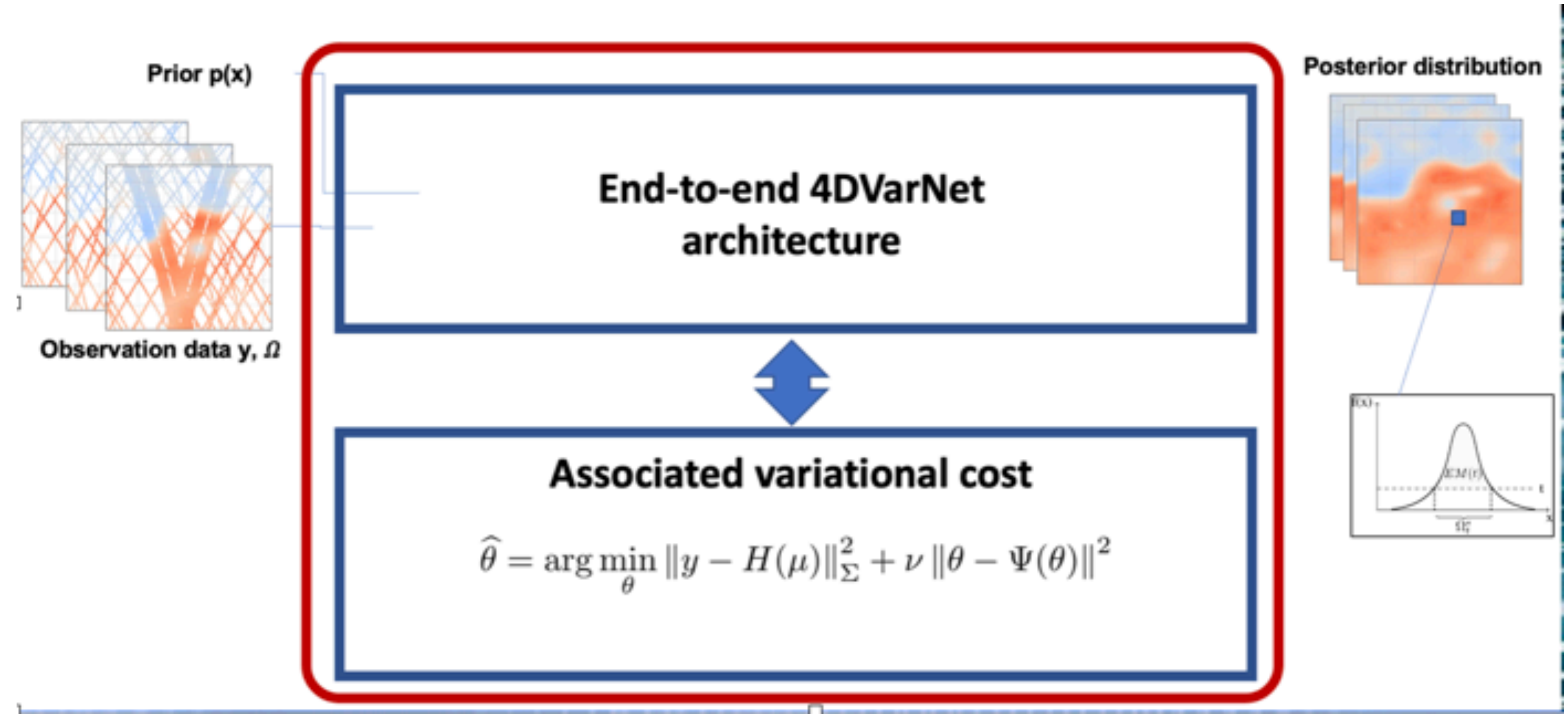
Model-driven vs. Learning-based approaches



Summary

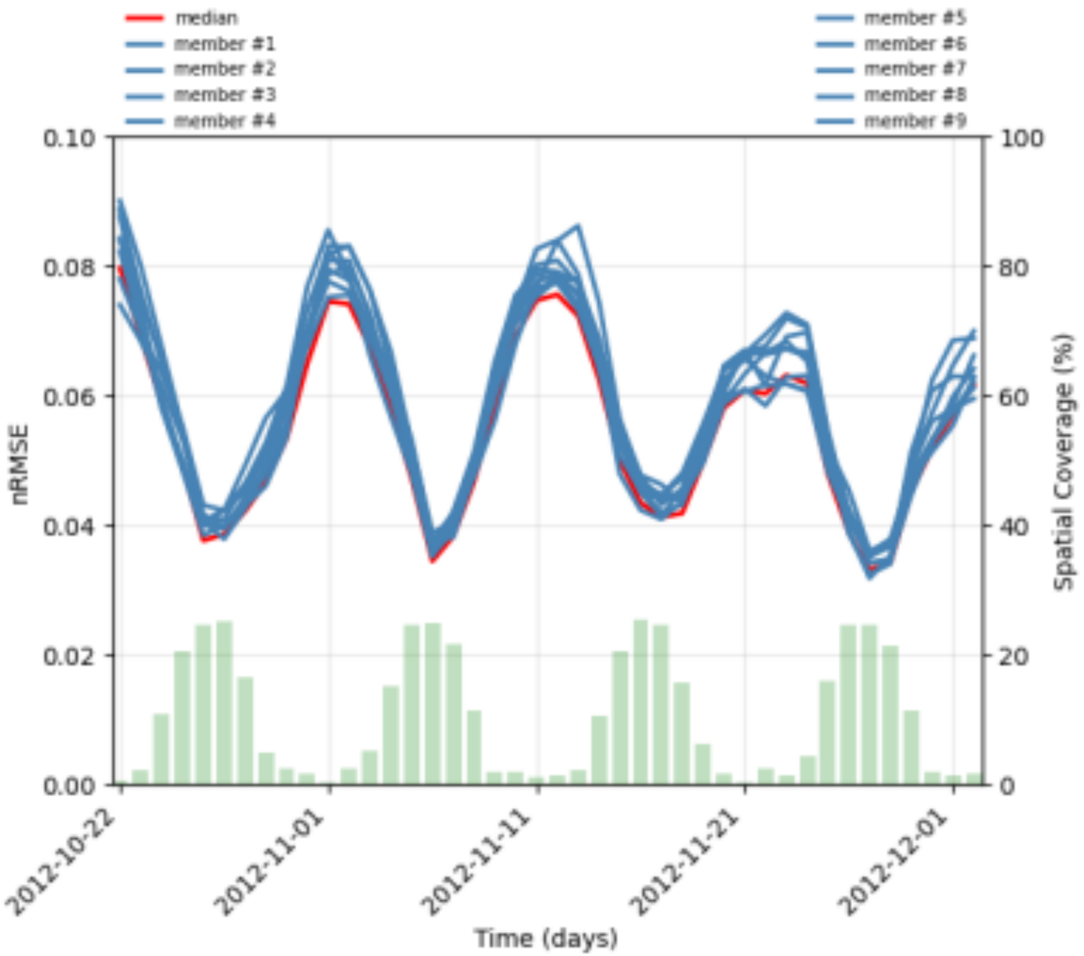
- ***NNs as numerical schemes for ODE/PDE/variational representations of geophysical dynamics***
- ***NN plug-and-play priors***
- ***End-to-end architecture for jointly learning a representation (**eg, ODE or NN prior**) and a solver***
- ***Requirement for differential implementations***
- ***The true prior might not be the optimal choice to solve inverse problems***

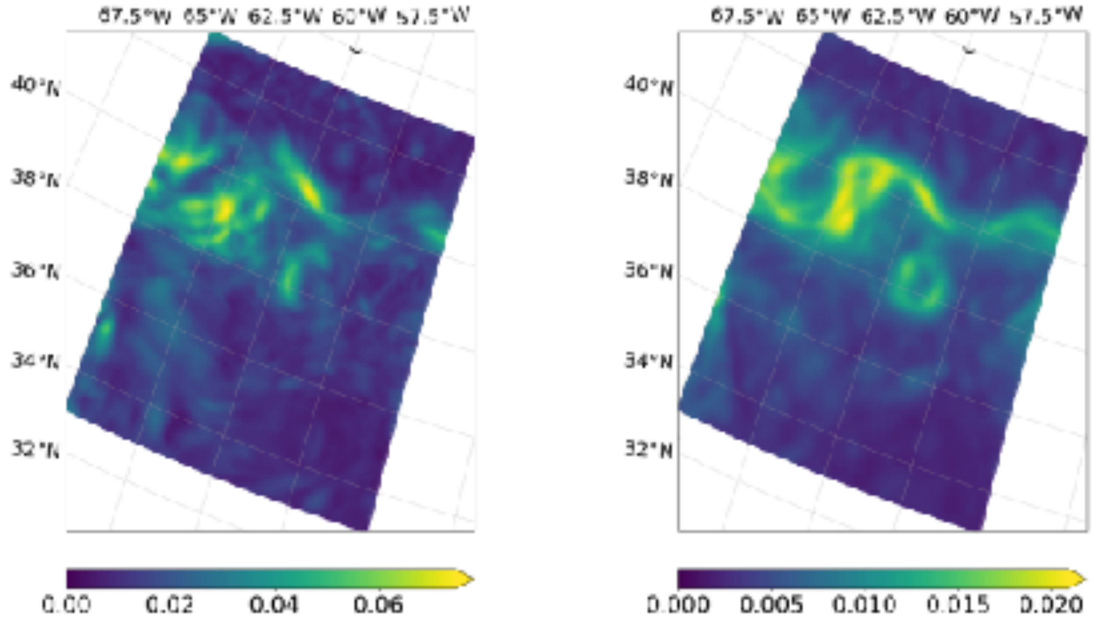
What about uncertainty quantification and end-to-end-learning?



4DVarNet: Dealing with uncertainties

Training an ensemble of models

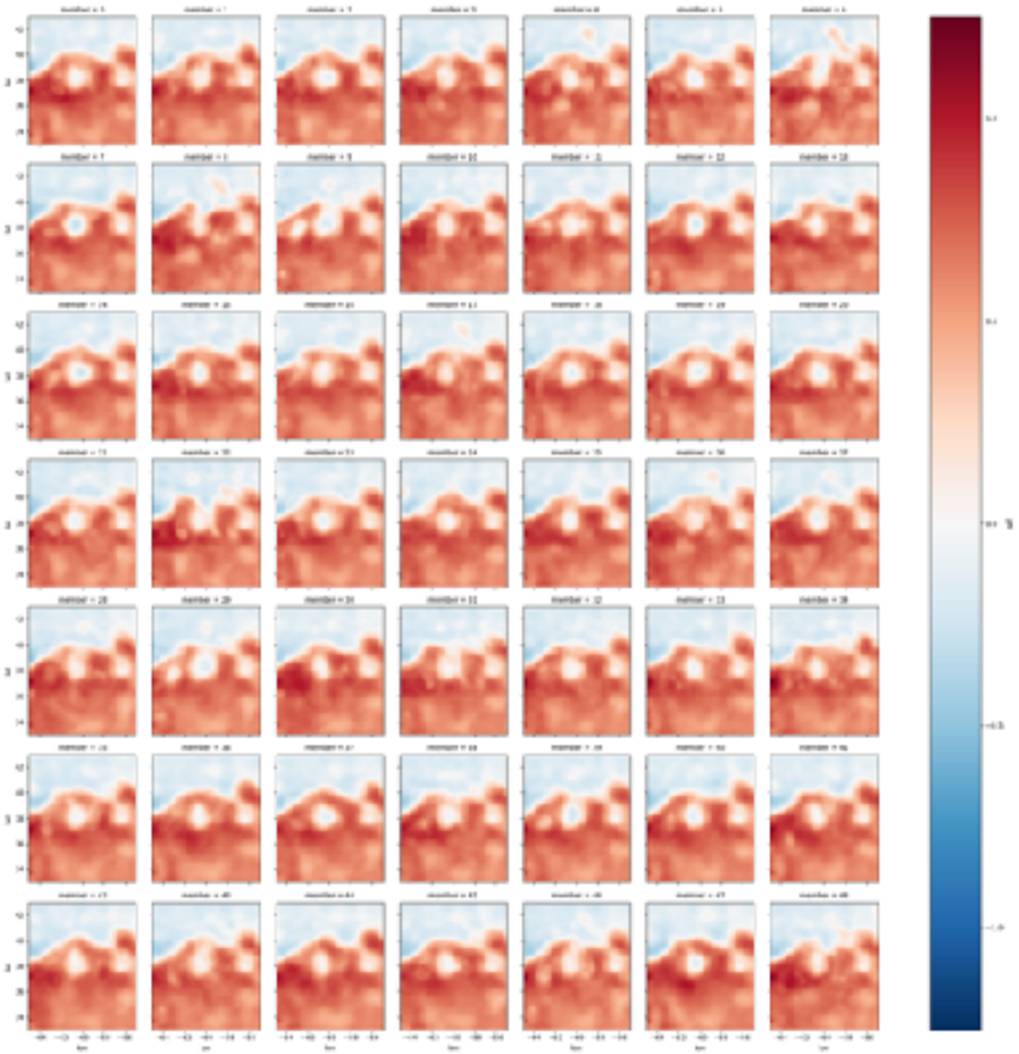




Mean interpolation RMSE vs. mean ensemble std

4DVarNet: Dealing with uncertainties

Generating ensembles of interpolated fields from randomised inputs



4DVarNet: Dealing with uncertainties [Lafon et al., in prep]

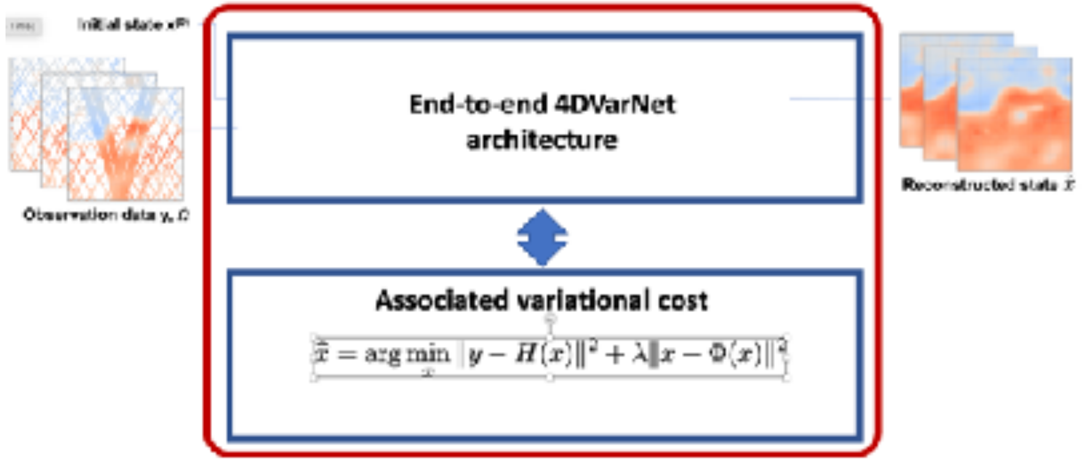
Physical state-space

$$(x, y)$$

4DVar formulation

$$\widehat{x} = \arg\min_{x} \|y - H(x)\|^2 + \lambda \|x - \Phi(x)\|^2$$

Associated end-to-end scheme



"Pdf" state-space

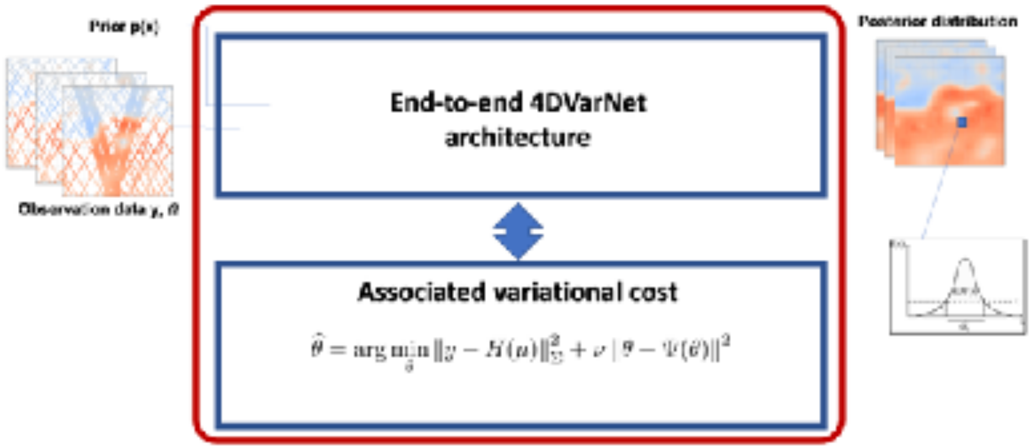
$$p(x|y) \approx q_\theta(x), y$$

4DVar formulation (ELBO)

$$\widehat{\theta} = \arg\min_{\theta} E_{q_\theta(x)}\left[p\left(y|x\right)\right] + KL\left[q_\theta(x)|p(x)\right]$$

$$\widehat{\theta} = \arg\min_{\theta} \|y - H(\mu)\|^2_\Sigma + \nu \left\|\theta - \Psi(\theta)\right\|^2$$

Associated end-to-end scheme



4DVarNet: Dealing with uncertainties [Lafon et al., in prep]

Application to the monitoring of the Danube river

- 15 stations over 31 with observations
- Reconstruction of the river flow at unobserved stations

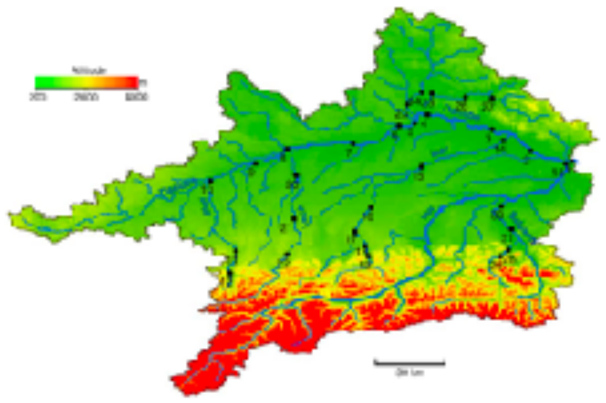
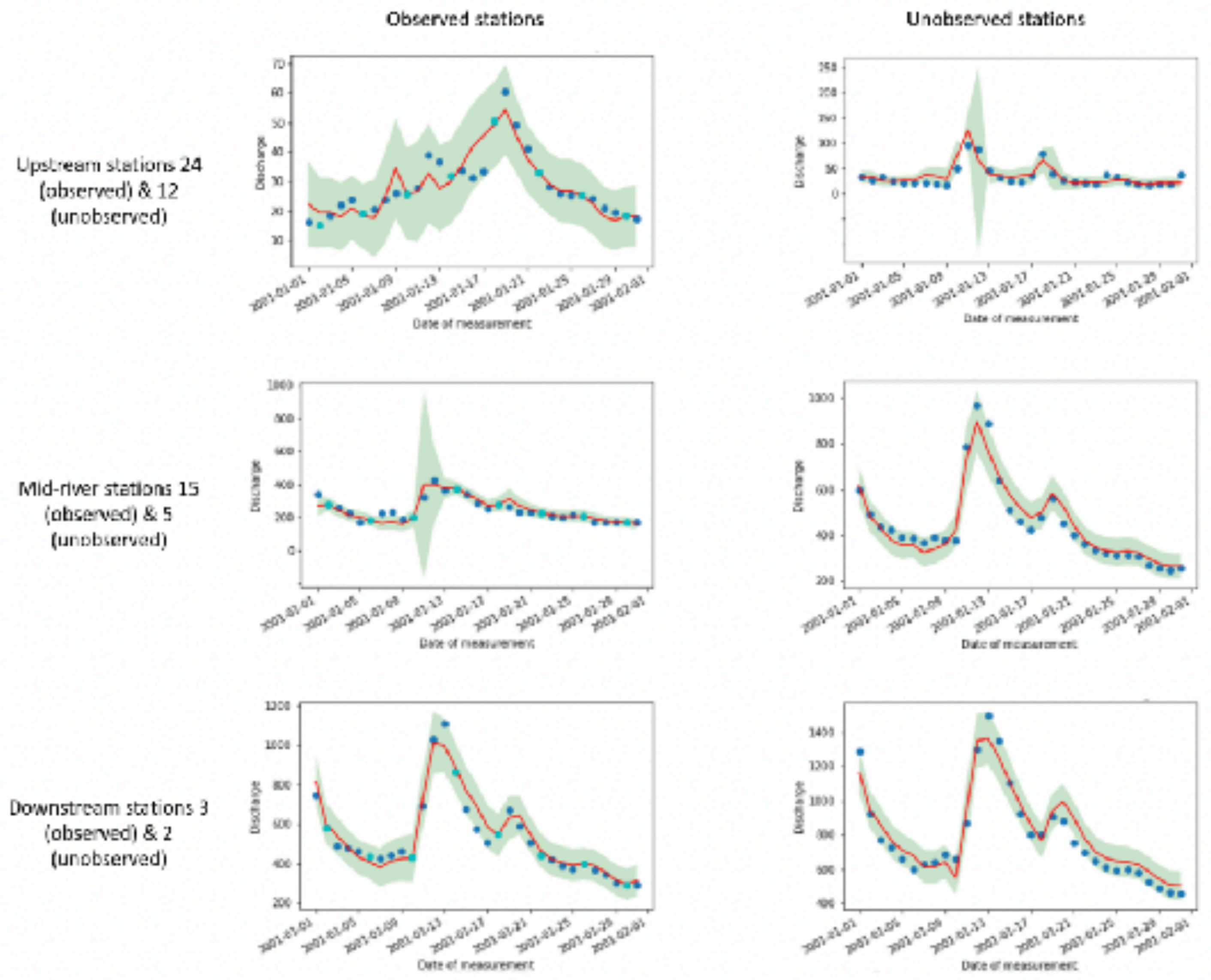


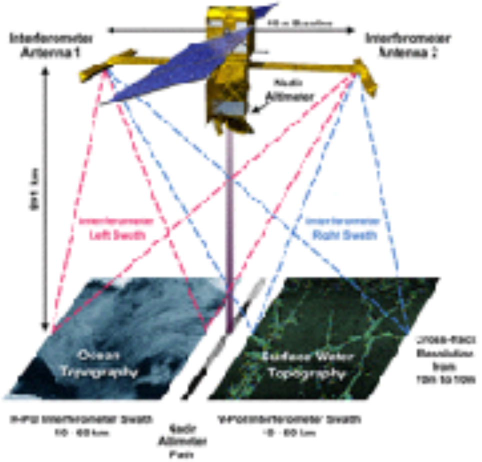
Figure: 31 gauging stations on the Danube river network (*Asadi et al., 2015*), with 50 years of daily measurements (1960-2010)



To which extent do neural DA schemes apply to real-world problems ?

4DVarNet as end-to-end DA schemes for space oceanography

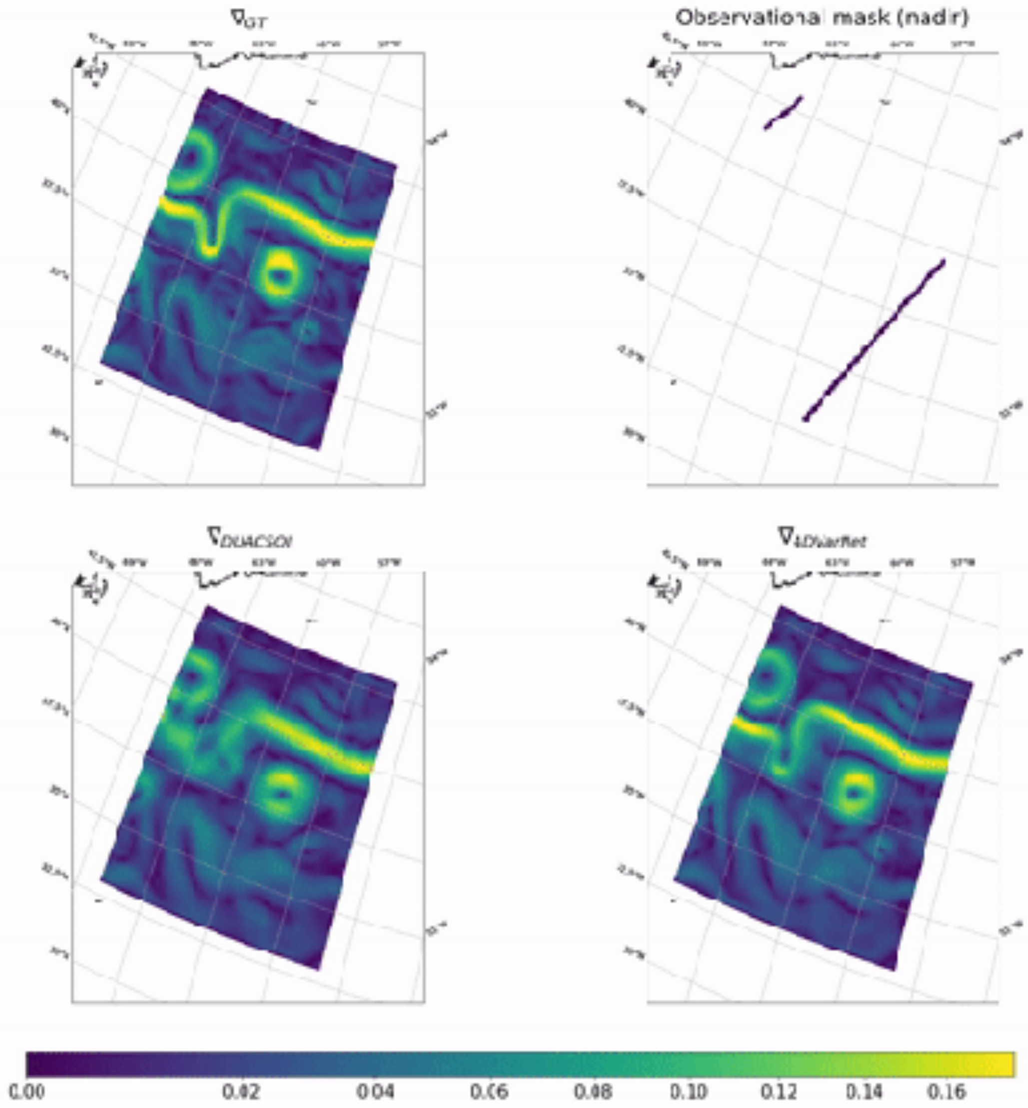
Satellite altimetry



Best score for BOOST-SWOT SLA Data Challenge

duacs 1 swot + 4 nadirs	0.92	0.02	1.22	11.15	Covariances DUACS	eval_duacs.ipynb
bfn 1 swot + 4 nadirs	0.93	0.02	0.8	10.09	CG Nudging	eval_bfn.ipynb
dymost 1 swot + 4 nadirs	0.93	0.02	1.2	10.07	Dynamic mapping	eval_dymost.ipynb
miost 1 swot + 4 nadirs	0.94	0.01	1.18	10.14	Multiscale mapping	eval_miost.ipynb
4DVarNet 1 swot + 4 nadirs 🏆	**0.96**	**0.01**	**0.70**	**4.15**	4DVarNet mapping	eval_4dvarnet.ipynb

https://github.com/ocean-data-challenges/2020a_SSH_mapping_NATL60



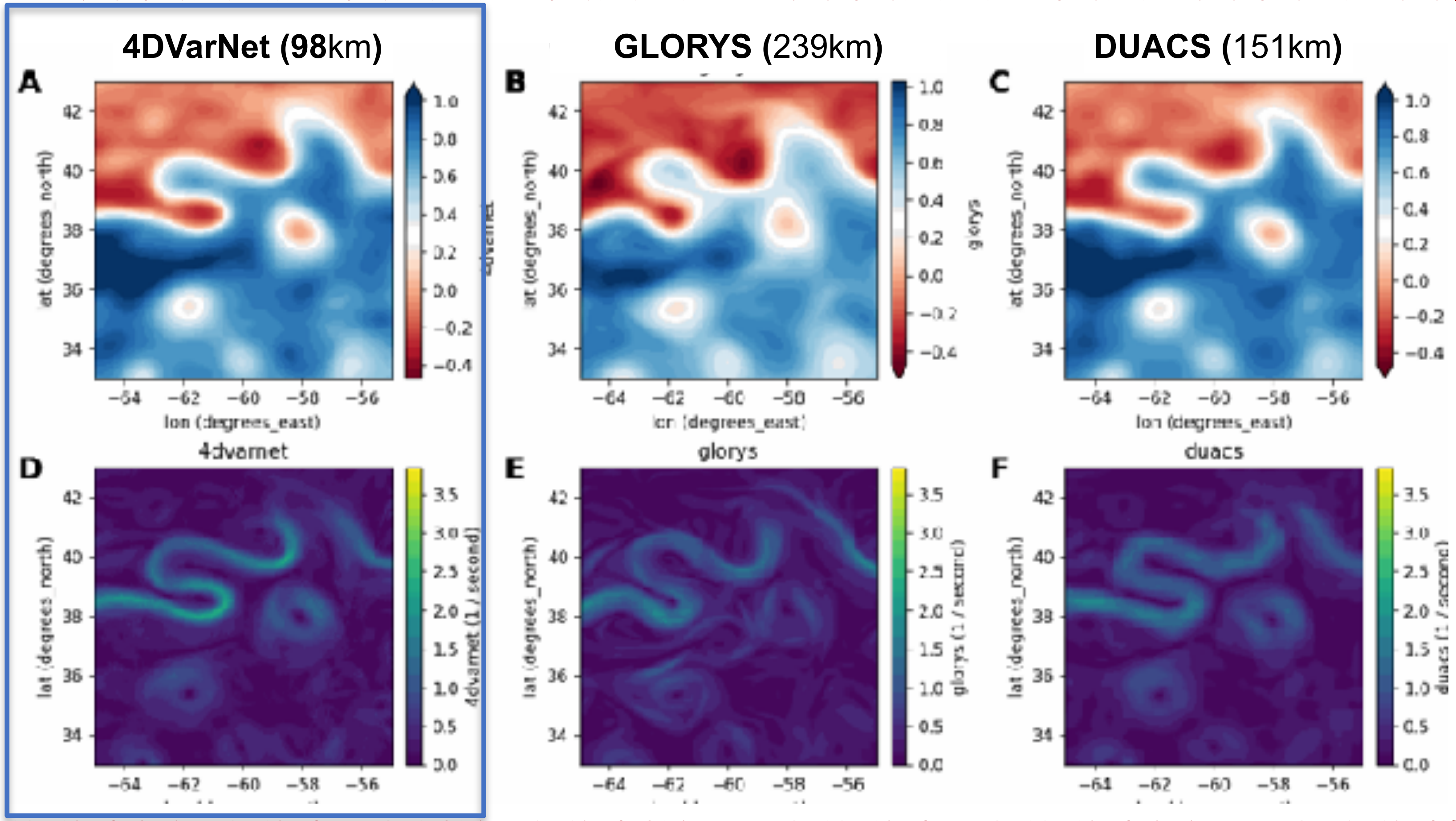
Application to SSH mapping (Real data/training from simulations)

Comparison with operational products and DA systems for real altimetry data

Gulf Stream region
4-nadir-altimeter
configuration

Git repository

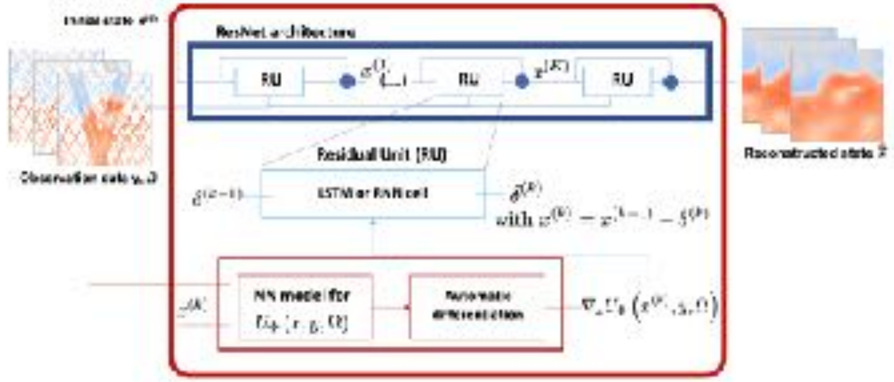
https://github.com/ocean-data-challenges/2021a_SSH_mapping_OSE

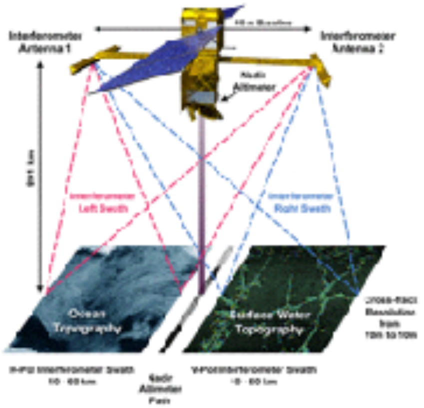


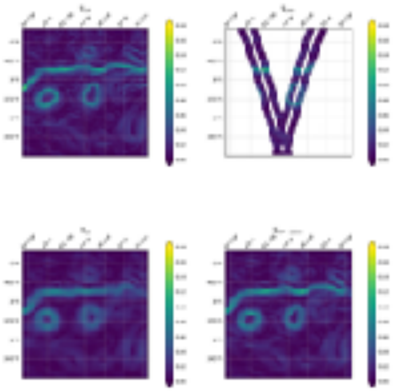
Febvre et al., in prep

Key messages

- **Physics-informed learning for satellite ocean remote sensing**
- **Trainable variational DA models** (observation model, prior, solver**)**
- **Application to interpolation, forecasting sampling and multimodal synergies**
- **Generic framework beyond space oceanography**
- **Objective-specific vs. Generic priors and DA schemes ?**







Preprint: https://doi.org/10.1029/2021MS002572
Code: https://github.com/CIA-Oceanix/4dvarnet-core

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/

References

- **Meta-learning:**
  - Andrychowicz, et al (2016). Learning to learn by gradient descent by gradient descent. NIPS, 2016
  - Hospedales et al. (2020). Meta-learning in neural networks: A survey. arXiv preprint arXiv:2004.05439.
- **End-to-end learning for Inverse Problems:**
  - Dieleman et al. End-to-end learning for music audio. IEEE ICASSP 2014. doi: 10.1109/ICASSP.2014.6854950
  - Lucas et al. (2018). Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods. IEEE SPM, 35(1), 20–36. doi: 10.1109/MSP.2017.2760358
- **Inverse problems with Plug-and-play priors:**
  - McCann et al. (2017). Convolutional Neural Networks for Inverse Problems in Imaging: A Review. IEEE SPM, 34(6), 85–95. doi: 10.1109/MSP.2017.2739299
  - Zhang et al. (2020). Plug-and-Play Image Restoration with Deep Denoiser. https://arxiv.org/abs/2008.13751
- **Deep unfolding methods:**
  - Chen et al. (2015). On Learning Optimized Reaction Diffusion Processes for Effective Image Restoration. In Proc. IEEE CVPR (pp. 5261–5269).
  - Yan et al. Deep ADMM-Net for Compressive Sensing MRI. NIPS, 2016.
  - Boudier et al. DAN -- An optimal Data Assimilation framework based on machine learning Recurrent Networks. Preprint, 2020.
  - Fablet et al. Learning Variational Data Assimilation Models and Solvers. JAMES, 2021.