Workshop LEFE-CYBER / ILICO / ODATIS

Introduction à l'intelligence artificielle dans l'assimilation des données géophysiques

Said Ouala

IMT Atlantique, Lab-STICC, Brest, France;

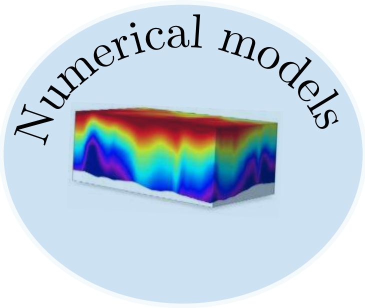
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Outline

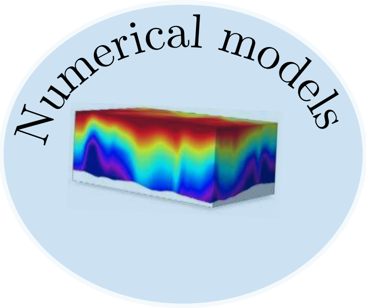
- Geophysical state estimation: models vs observations
- IA in geophysical state estimation and DA:
  - Higher resolution interpolation: general framework and applications
  - Generative models and data assimilation for modeling dynamical systems
  - End-to-end (online) Learning in Hybrid Modeling Systems

Geophysical state estimation: models vs observations

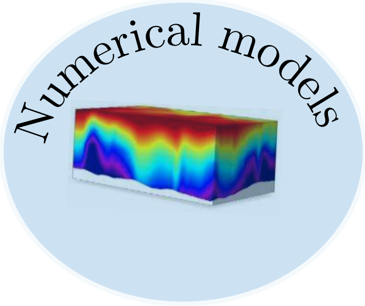
Geophysical state estimation: models vs observations



Geophysical state estimation: models vs observations

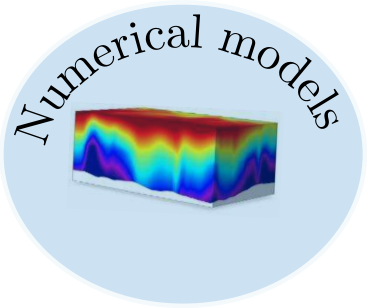


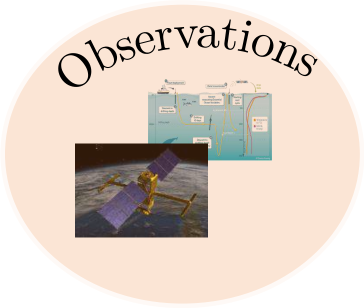
Geophysical state estimation: models vs observations



- Forecast
- Long term simulations
- Understanding of physical processes

Geophysical state estimation: models vs observations

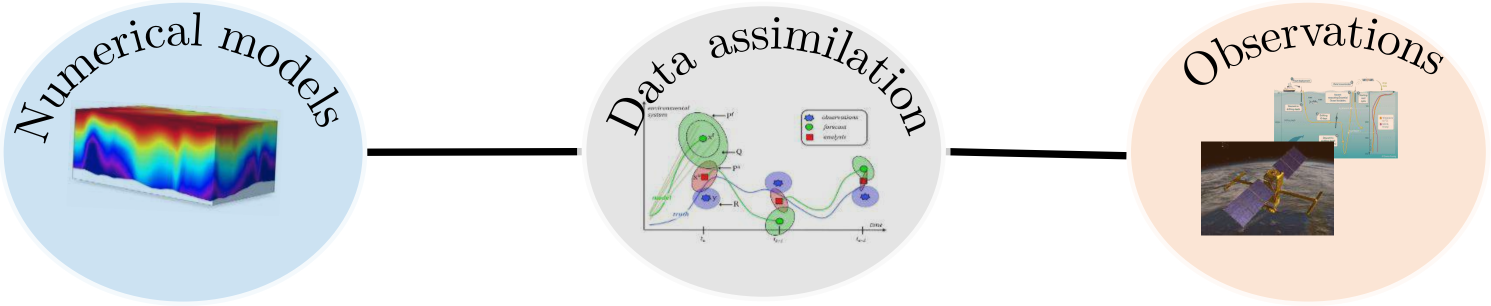




- Forecast
- Long term simulations
- Understanding of physical processes

- State monitoring
- Model validation

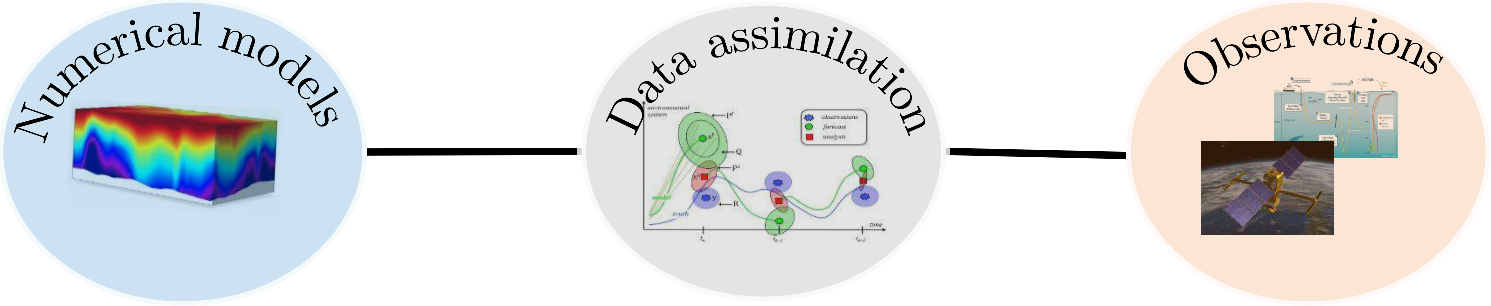
Geophysical state estimation: models vs observations



- Forecast
- Long term simulations
- Understanding of physical processes

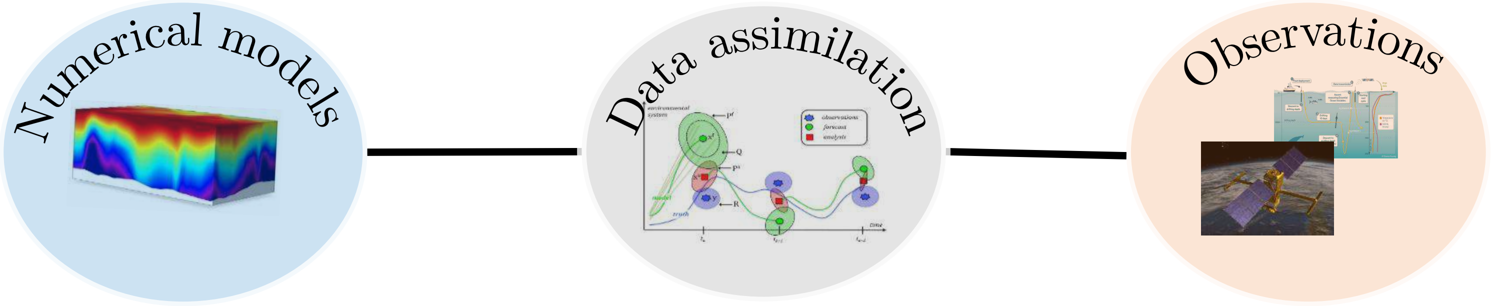
- State monitoring
- Model validation

Geophysical state estimation: models vs observations



- Initialization of the models

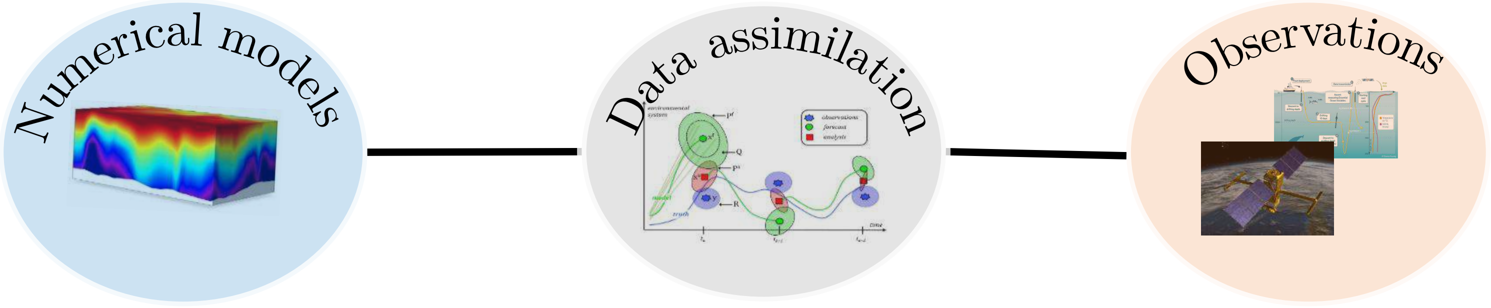
Geophysical state estimation: models vs observations



- Initialization of the models

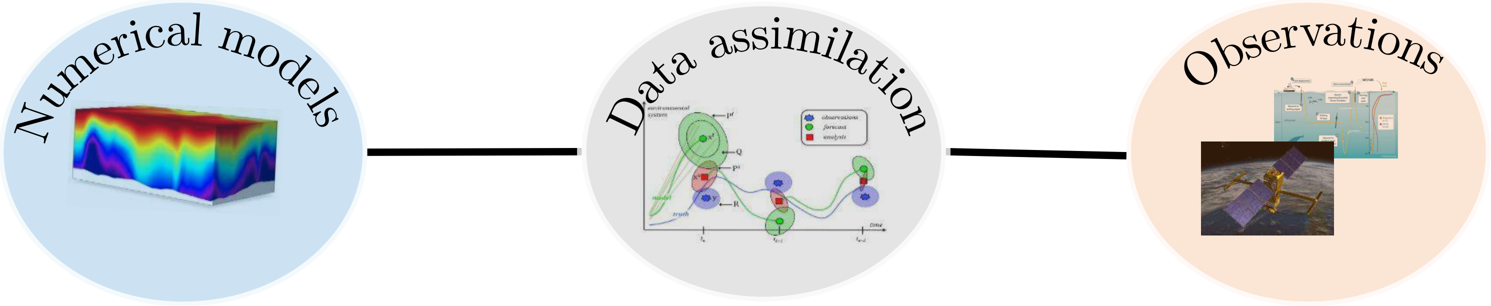
- Data reconstruction and interpolation

Geophysical state estimation: models vs observations



- Numerical discretization errors
- Model bias correction
- Choice of some parameterizations
- How to increase the predictability
- How to model a subset of variables

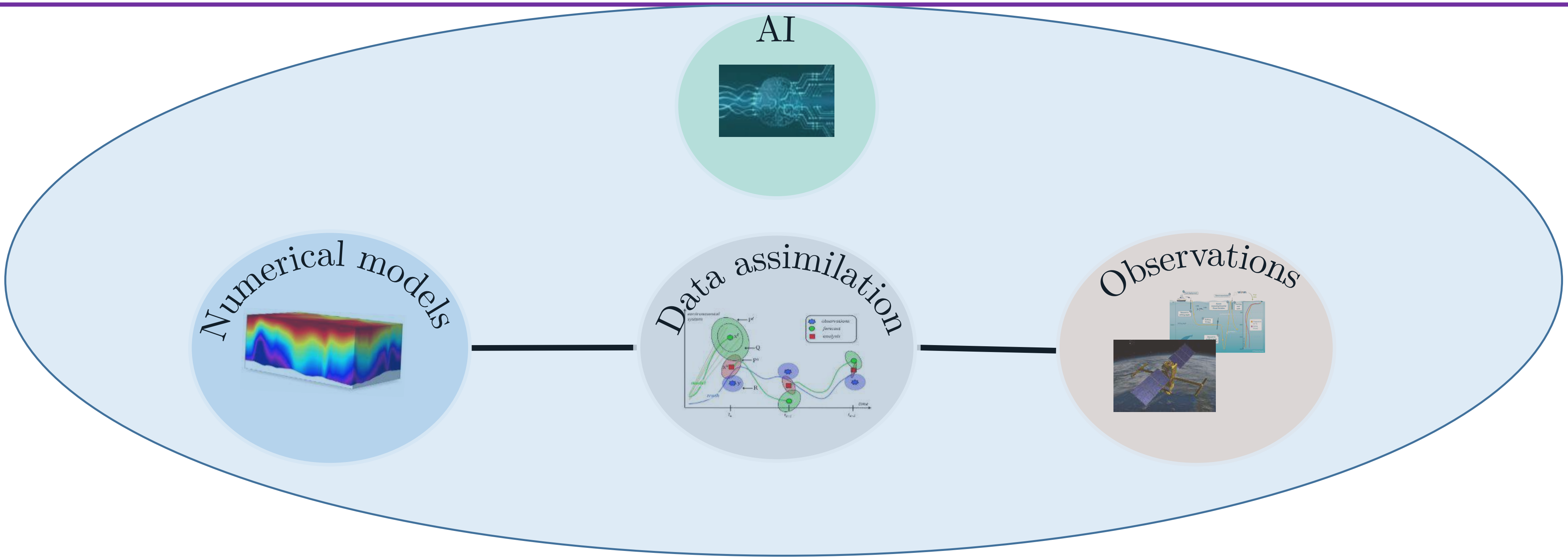
Geophysical state estimation: models vs observations



- Numerical discretization errors
- Model bias correction
- Choice of some parameterizations
- How to increase the predictability
- How to model a subset of variables

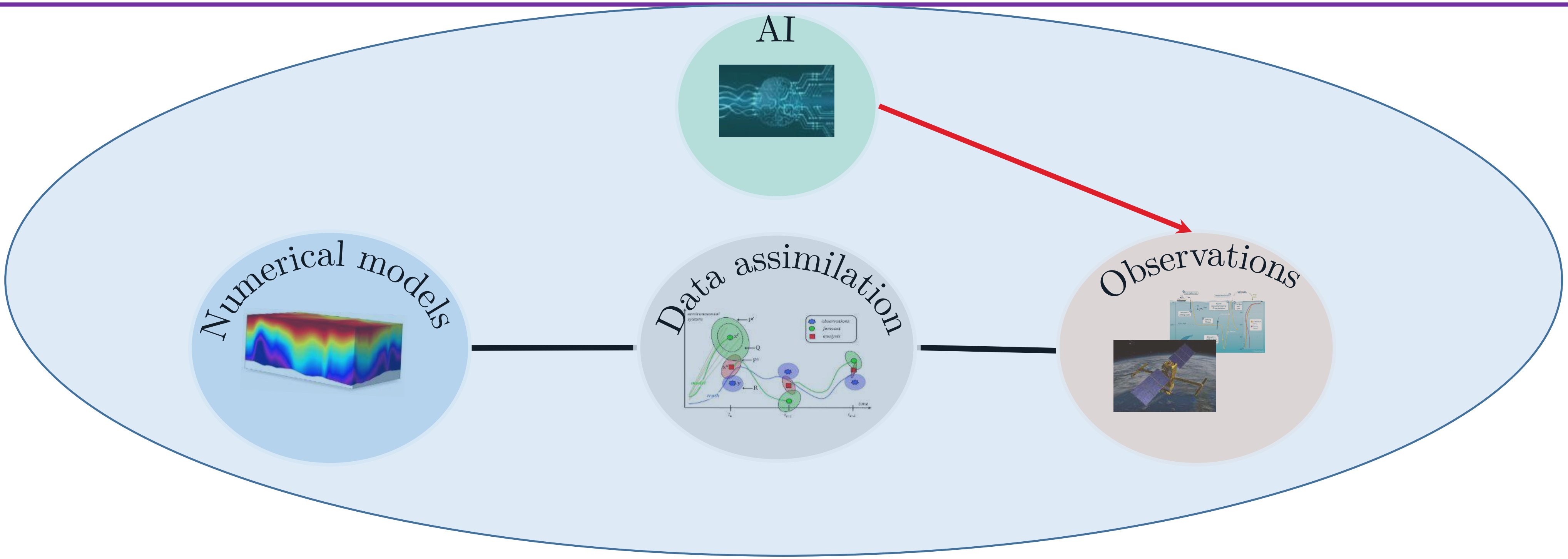
- How to explore these big amounts data
- How to design new sensing missions

IA in geophysical state estimation and DA



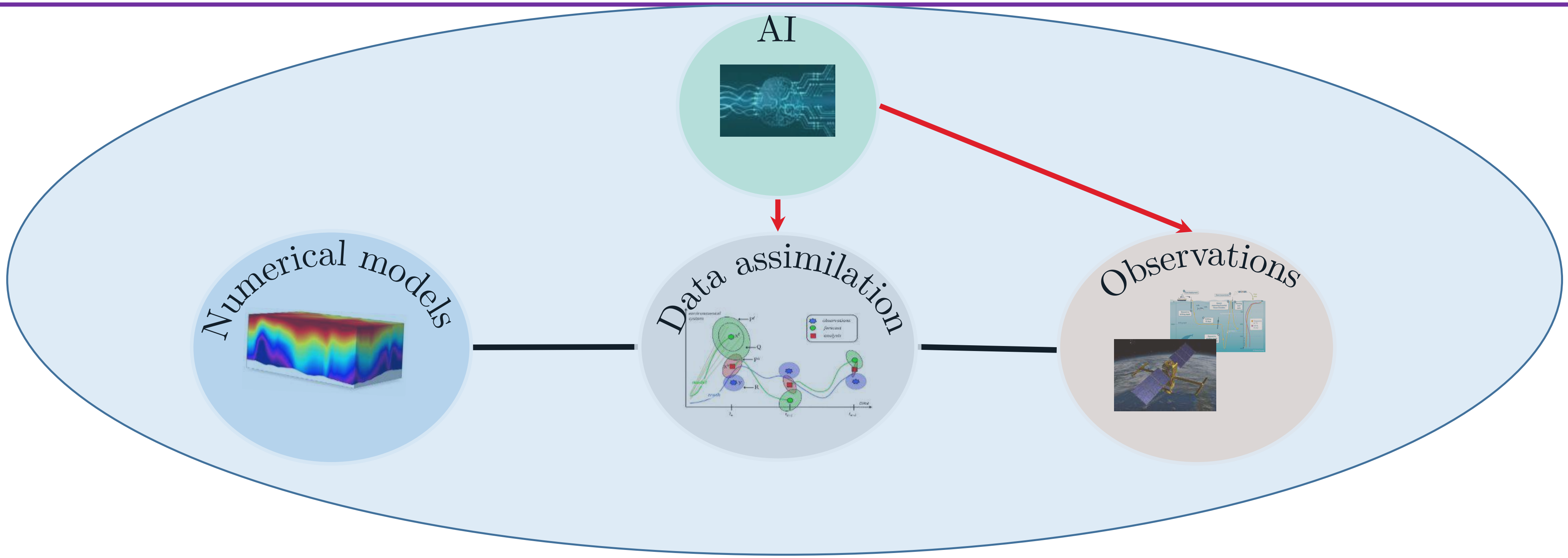
Improving geophysical state estimation using machine learning and AI

IA in geophysical state estimation and DA



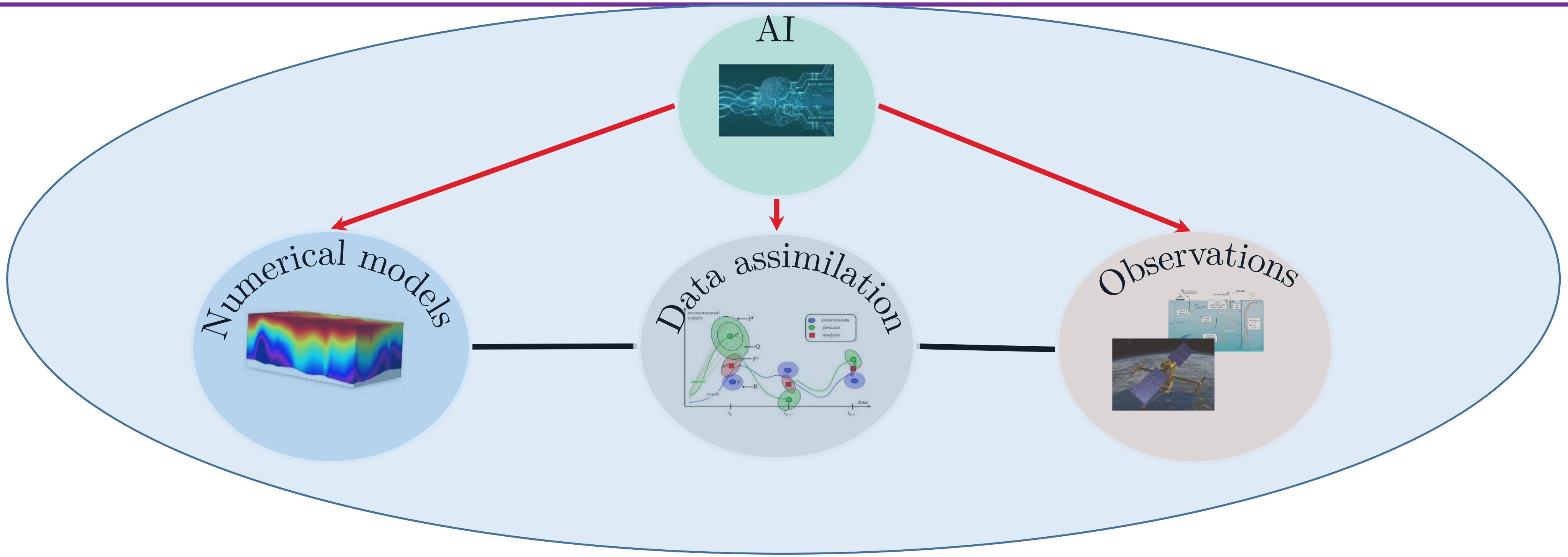
- Higher resolution interpolation
- Data driven synergy, emulators

IA in geophysical state estimation and DA



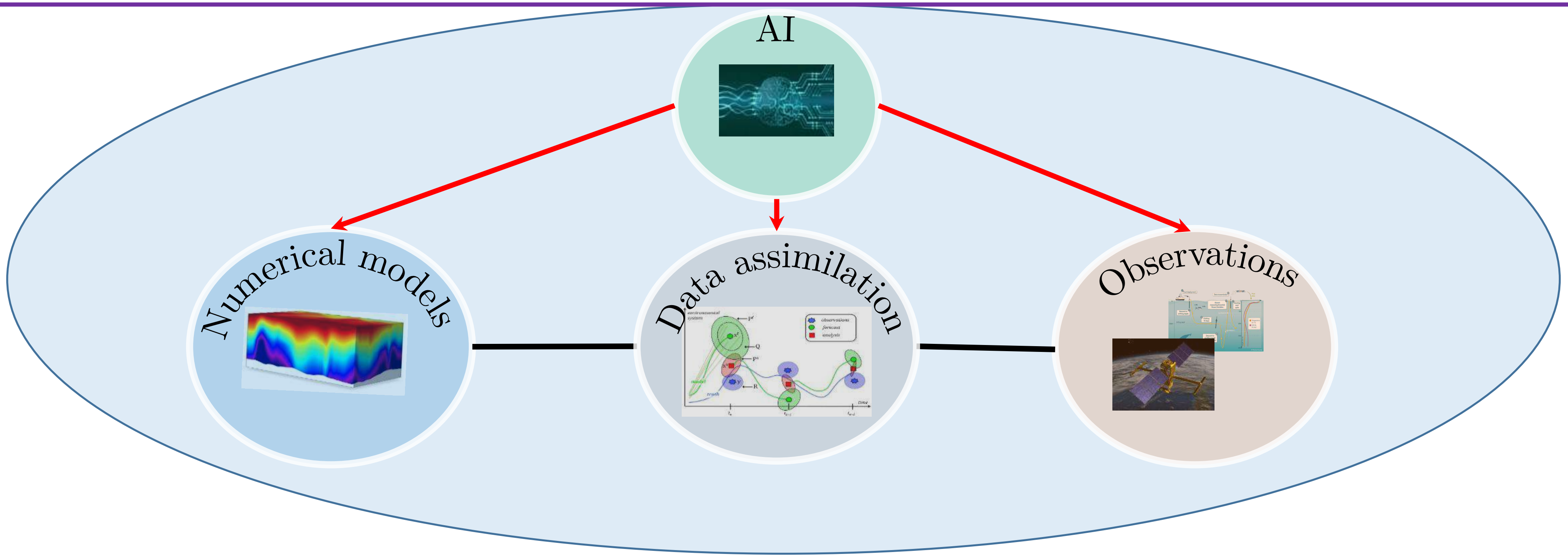
- Point of view from both AI generative models/standard DA schemes for surrogate modeling
  Machine learning, data assimilation and uncertainty quantification

IA in geophysical state estimation and DA



- Surrogate modeling
- Accelerating model resolution
- Model tuning and parameterization, hybrid models

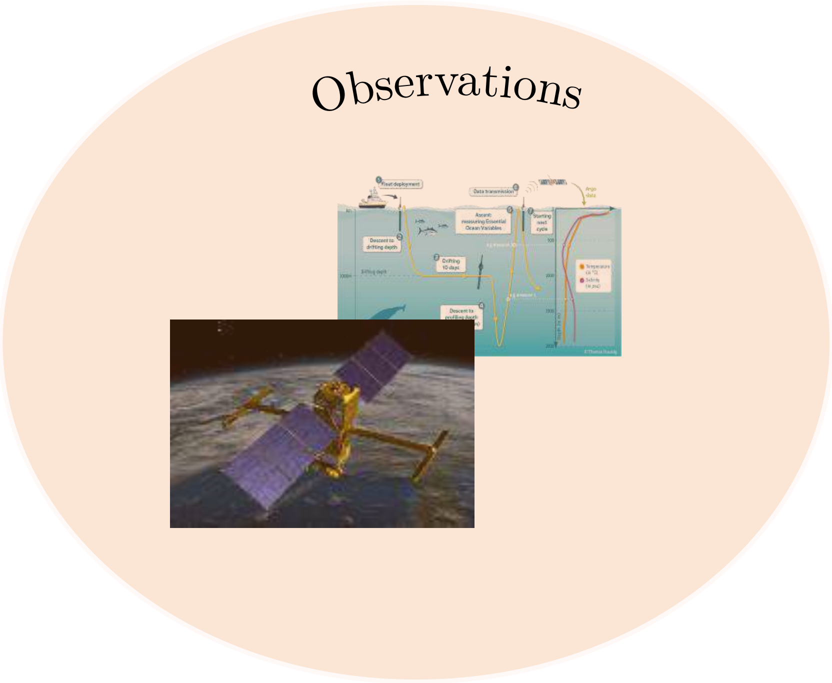
IA in geophysical state estimation and DA



- ➡ **Formulation of higher resolution interpolation with examples**
- ➡ **Generative models and data assimilation for modeling dynamical systems**
- ➡ **End-to-end Learning of sub-models in Hybrid Modeling Systems**

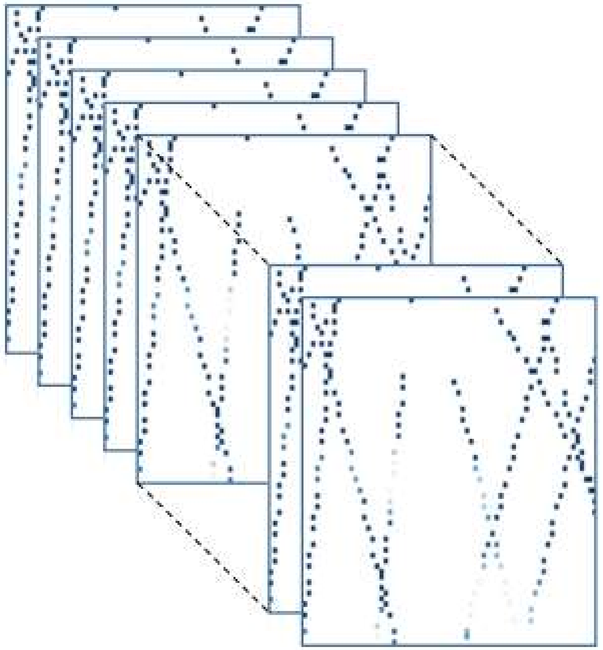
AI for observations

Higher resolution interpolation: general framework and applications



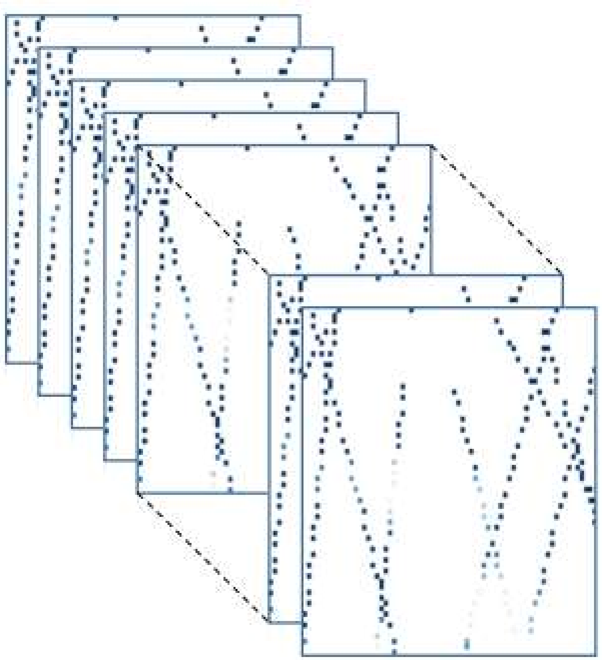
Higher resolution interpolation

Problem statement



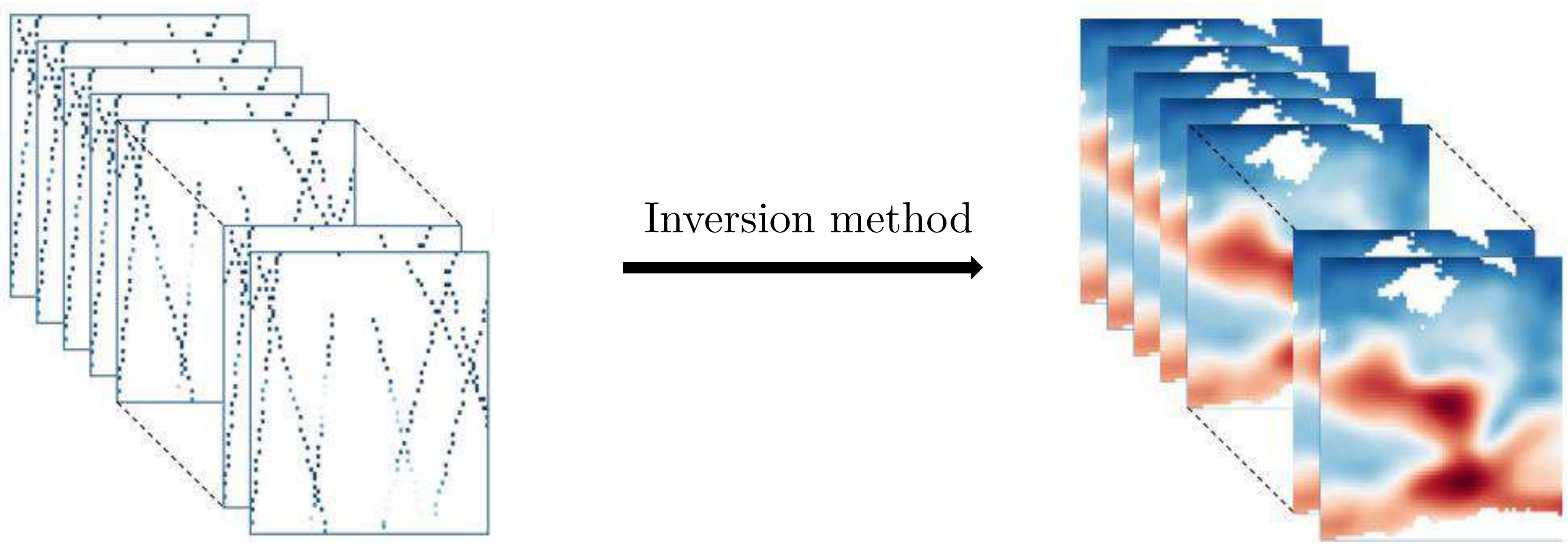
Higher resolution interpolation

Problem statement



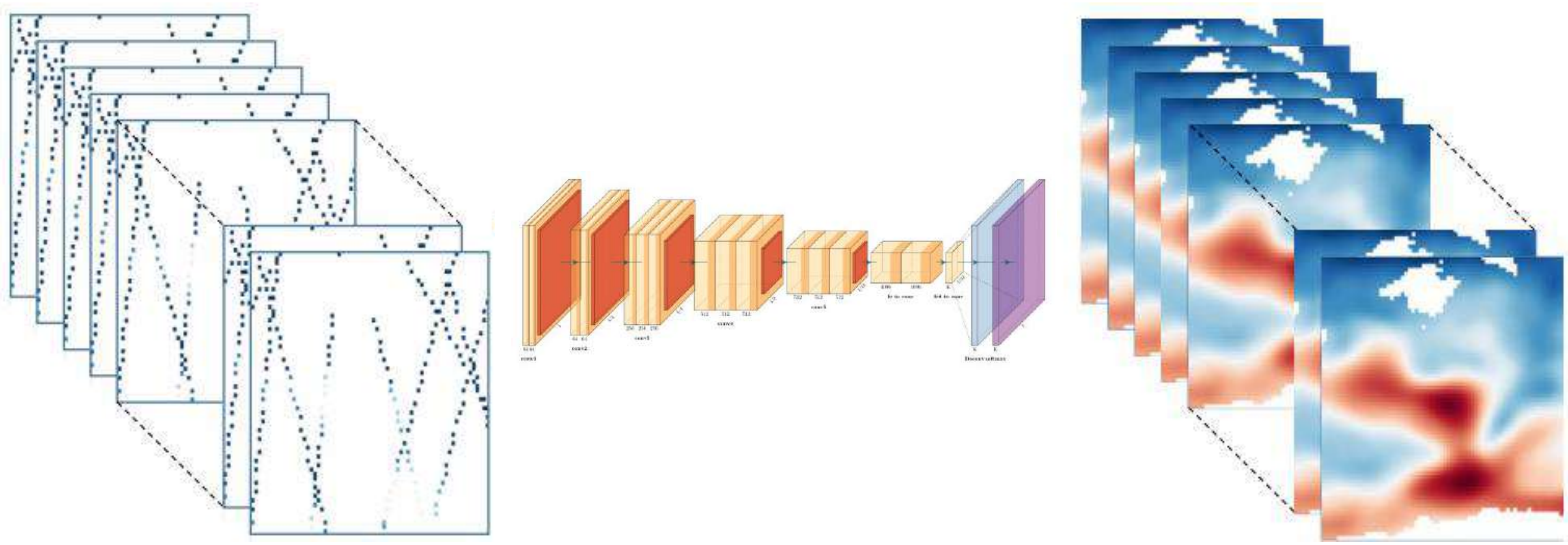
Higher resolution interpolation

Problem statement



Higher resolution interpolation

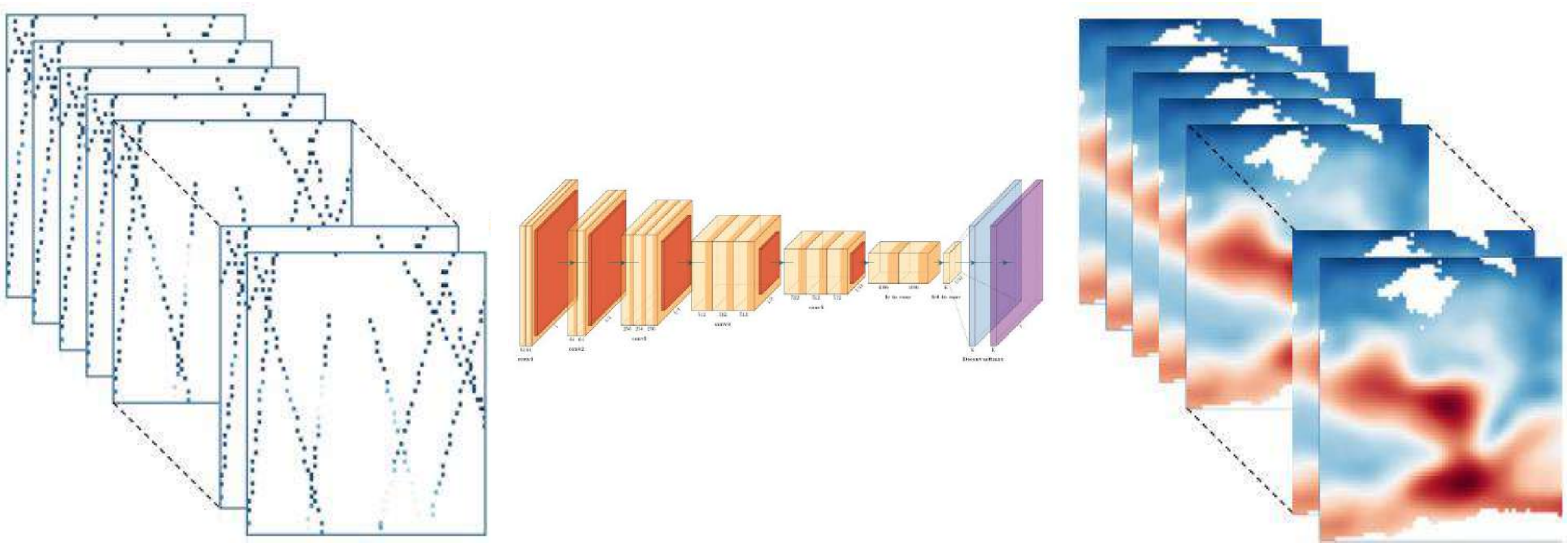
Problem statement



- Easy implementation and testing
- Takes advantage of recent developments in AI architectures

Higher resolution interpolation

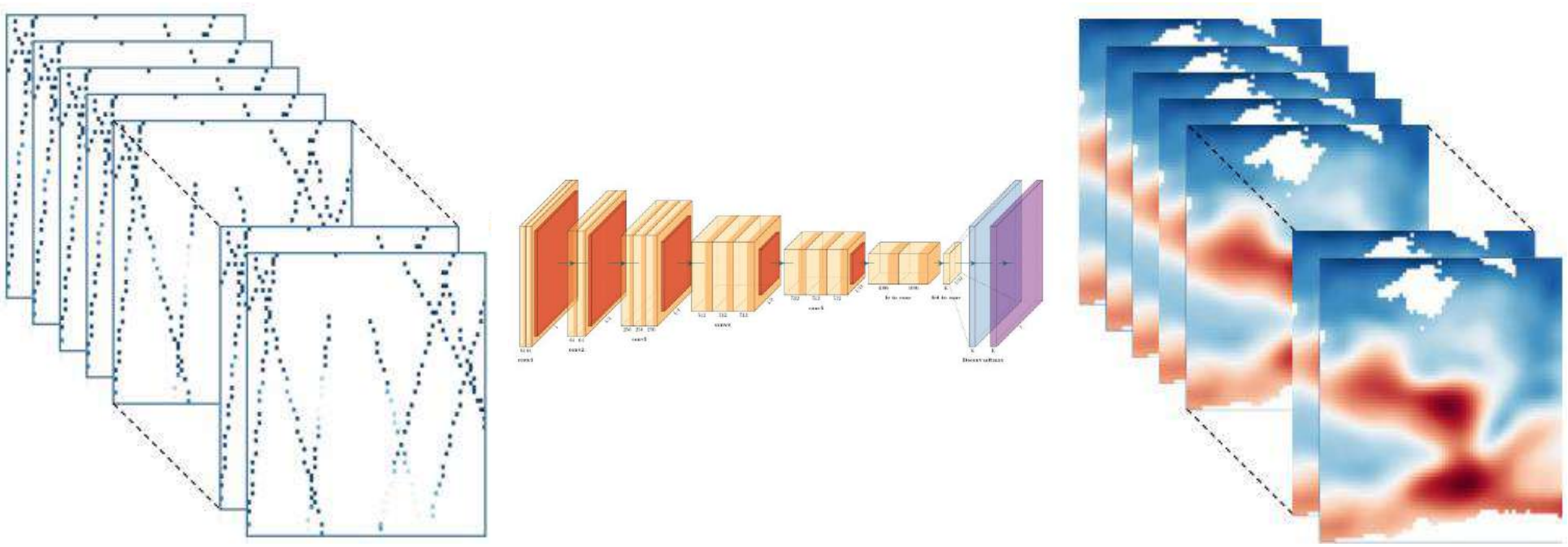
Problem statement



- Easy implementation and testing
- Takes advantage of recent developments in AI architectures
- Uncertainty quantification ?
- Forecasting applications ?

Higher resolution interpolation

Problem statement



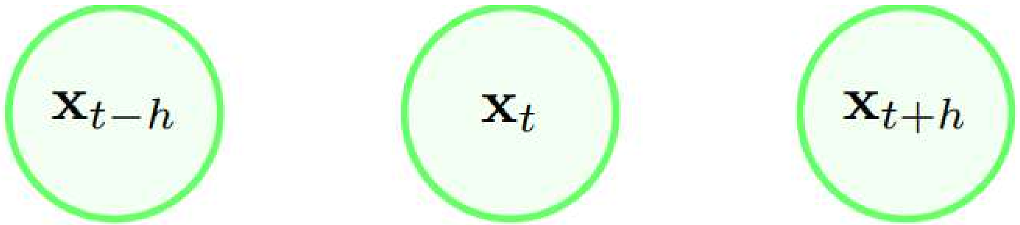
- Easy implementation and testing
- Takes advantage of recent developments in AI architectures
- Uncertainty quantification ? ⟶ Generative models
- Forecasting applications ? ⟶ Multitask learning

Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

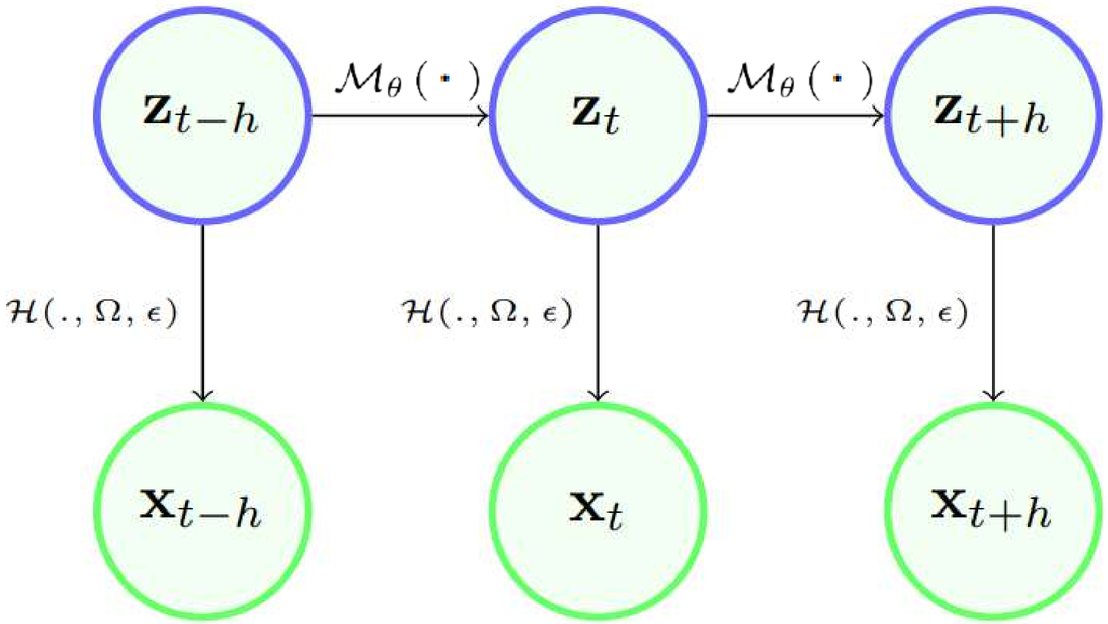
Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:



Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

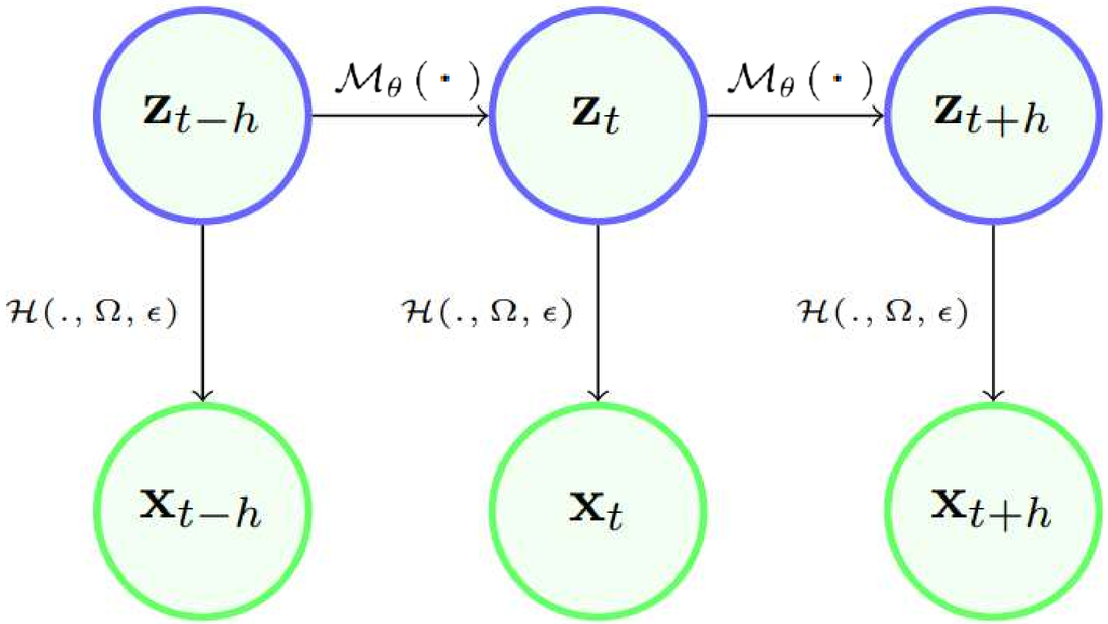


Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

- State variabl
- Observations

$$\begin{cases} \mathbf{z}_{t+1} = \mathcal{M}_\theta(\mathbf{z}_t) + \varepsilon_t \\ \mathbf{x}_t = \mathcal{H}_\theta(\mathbf{z}_t) + \mu_t \end{cases}$$

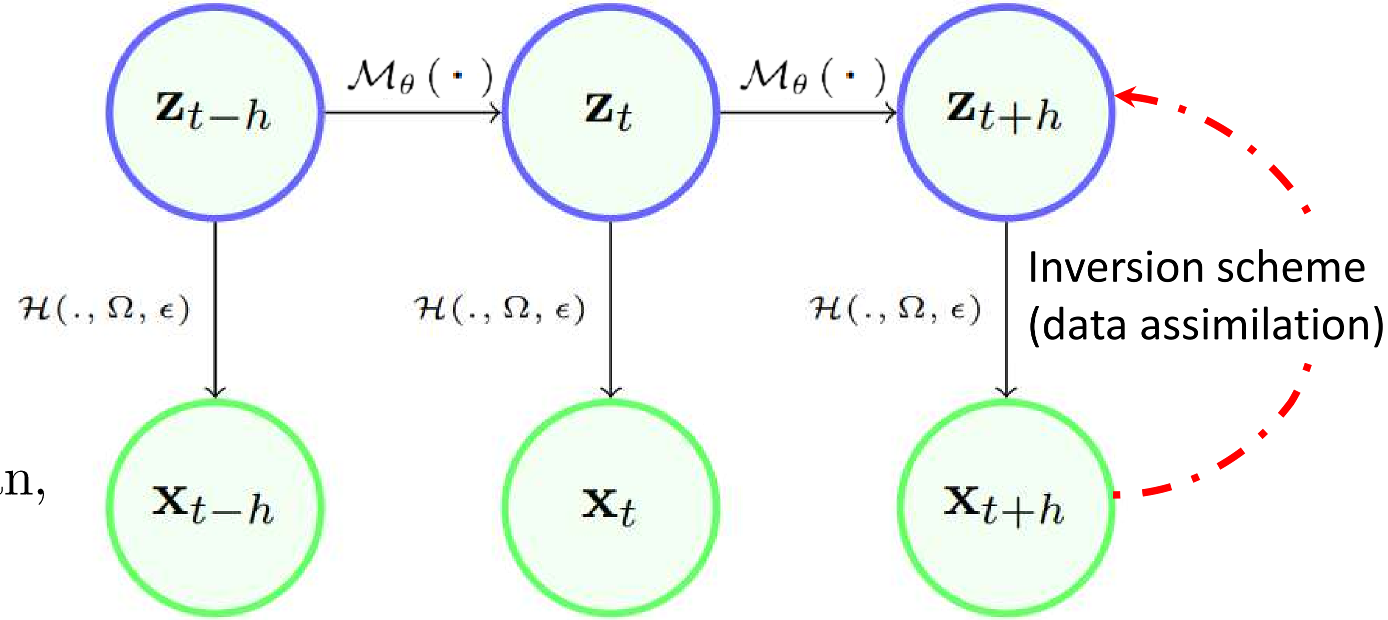


Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

- State variabl
- Observations

$$\begin{cases} \mathbf{z}_{t+1} = \mathcal{M}_\theta(\mathbf{z}_t) + \varepsilon_t \\ \mathbf{x}_t = \mathcal{H}_\theta(\mathbf{z}_t) + \mu_t \end{cases}$$



Learning steps:

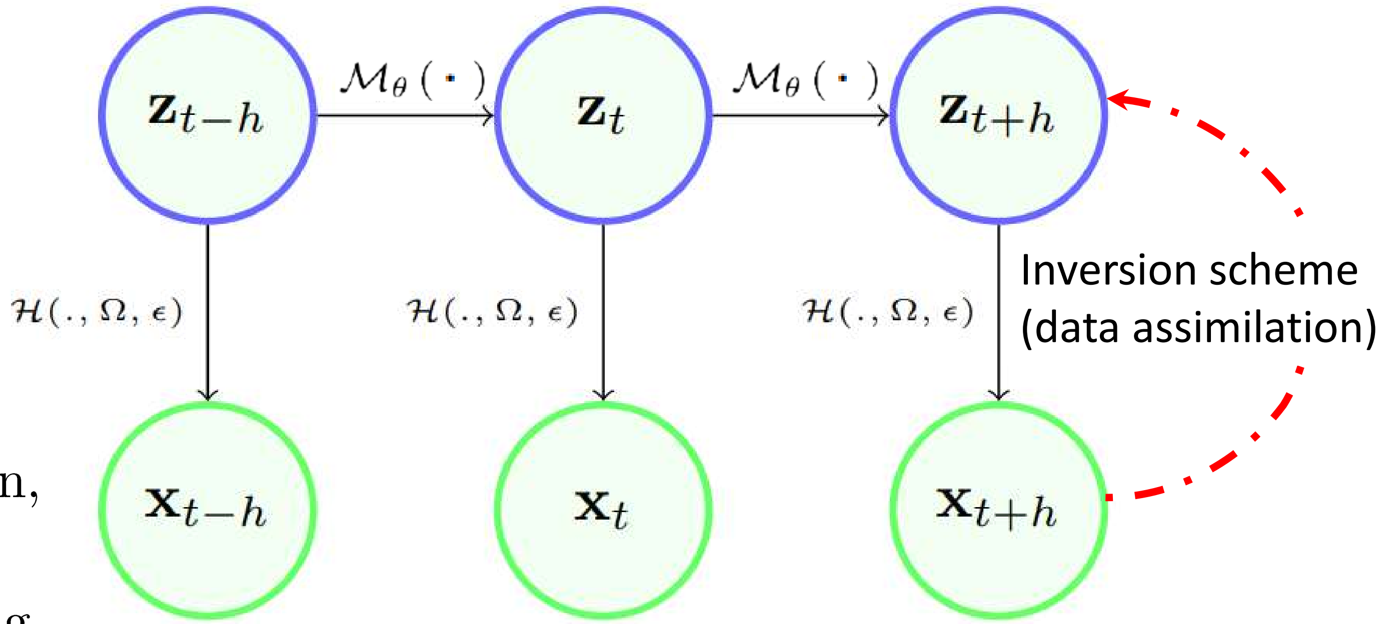
- Compute the state variable using an inversion scheme (e.g. Ensemble Kalman, 4D-Var) $\tilde{z}_t$

Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

- State variabl
- Observations

$$\begin{cases} \mathbf{z}_{t+1} = \mathcal{M}_\theta(\mathbf{z}_t) + \varepsilon_t \\ \mathbf{x}_t = \mathcal{H}_\theta(\mathbf{z}_t) + \mu_t \end{cases}$$



Learning steps:

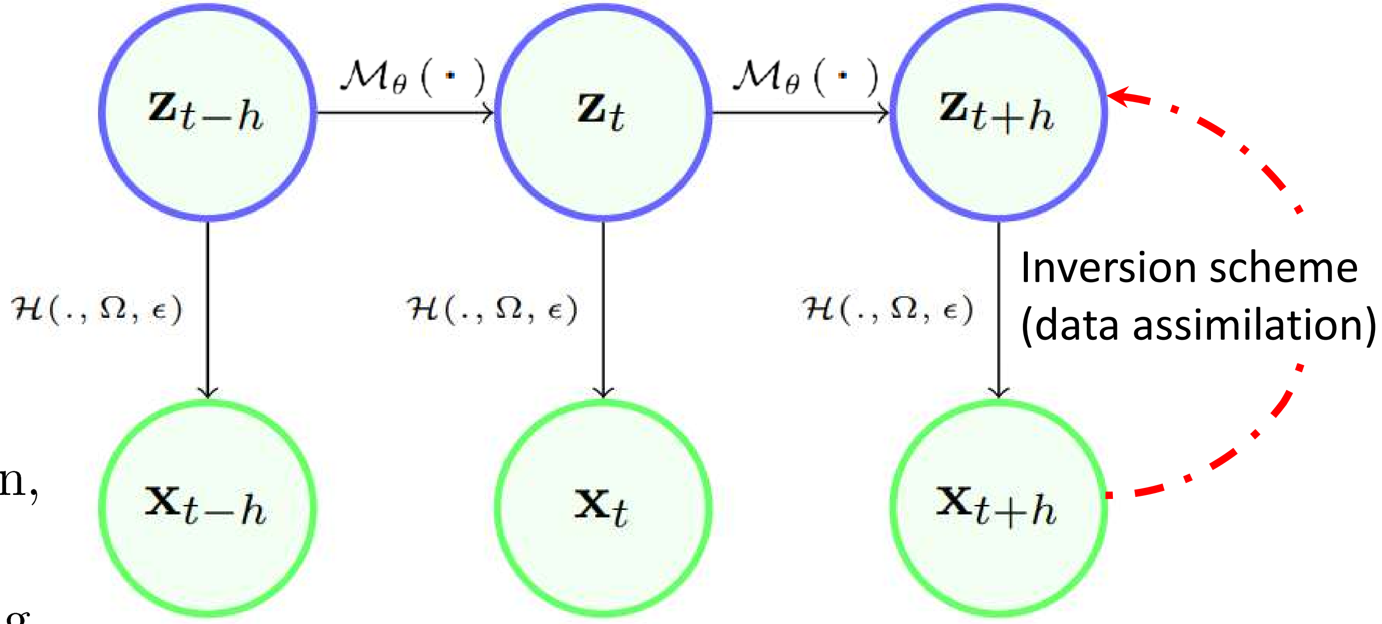
- Compute the state variable using an inversion scheme (e.g. Ensemble Kalman, 4D-Var) $\tilde{z}_t$
- Compute the gap free observations using the forward model $\tilde{x}_t = H(\tilde{z}_t)$

Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

- State variabl
- Observations

$$\begin{cases} \mathbf{z}_{t+1} = \mathcal{M}_\theta(\mathbf{z}_t) + \varepsilon_t \\ \mathbf{x}_t = \mathcal{H}_\theta(\mathbf{z}_t) + \mu_t \end{cases}$$



Learning steps:

- Compute the state variable using an inversion scheme (e.g. Ensemble Kalman, 4D-Var) $\tilde{z}_t$
- Compute the gap free observations using the forward model $\tilde{x}_t = H(\tilde{z}_t)$

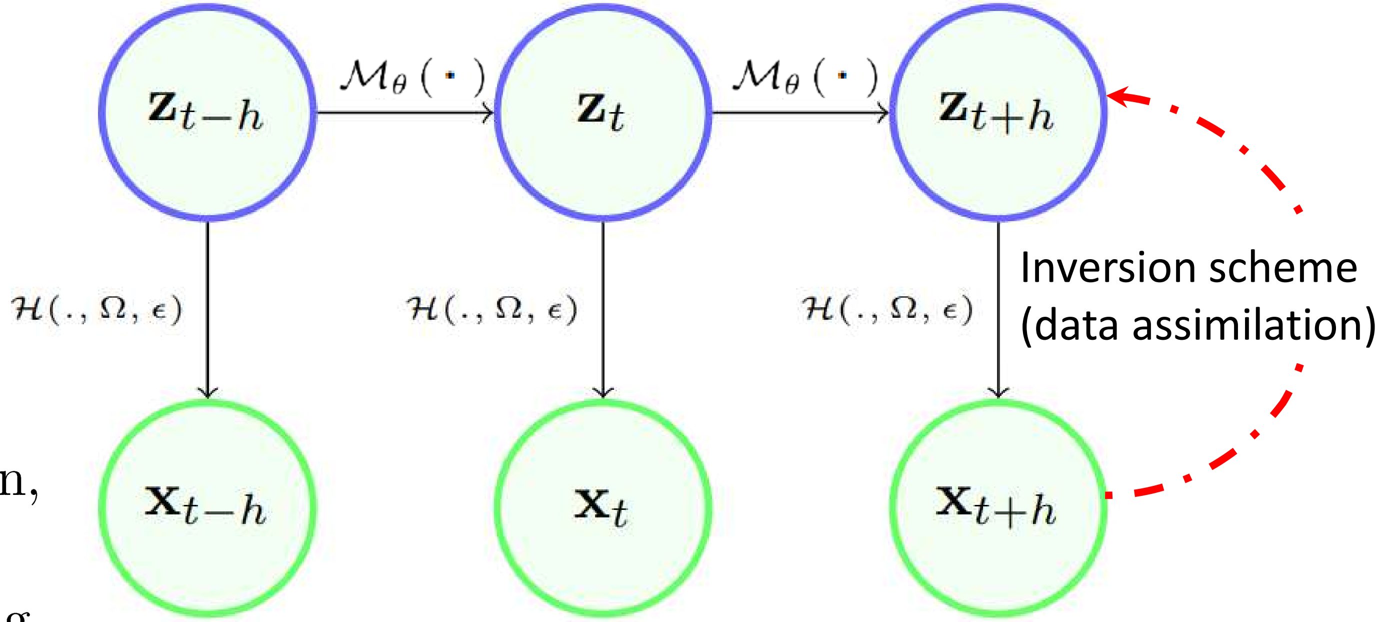
Minimize $|x_t - H(\tilde{z}_t)|$

Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

- State variabl
- Observations

$$\begin{cases} \mathbf{z}_{t+1} = \mathcal{M}_\theta(\mathbf{z}_t) + \varepsilon_t \\ \mathbf{x}_t = \mathcal{H}_\theta(\mathbf{z}_t) + \mu_t \end{cases}$$



Learning steps:

- Compute the state variable using an inversion scheme (e.g. Ensemble Kalman, 4D-Var) $\tilde{z}_t$
- Compute the gap free observations using the forward model $\tilde{x}_t = H(\tilde{z}_t)$

Minimize $|x_t - H(\tilde{z}_t)|$

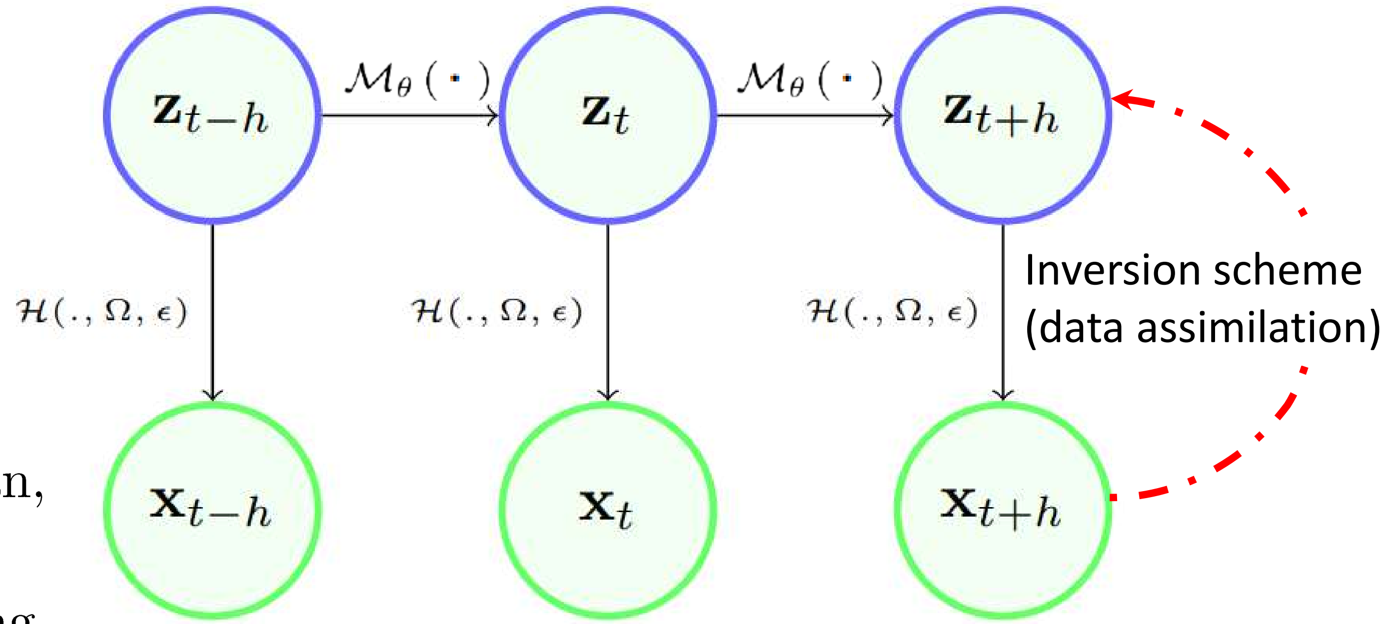
- Naturally deals with missing data
- Can do forecast

Higher resolution interpolation

Turn spatiotemporal interpolation into a Bayesian filtering problem:

- State variabl
- Observations

$$\begin{cases} \mathbf{z}_{t+1} = \mathcal{M}_\theta(\mathbf{z}_t) + \varepsilon_t \\ \mathbf{x}_t = \mathcal{H}_\theta(\mathbf{z}_t) + \mu_t \end{cases}$$



Learning steps:

- Compute the state variable using an inversion scheme (e.g. Ensemble Kalman, 4D-Var) $\tilde{z}_t$
- Compute the gap free observations using the forward model $\tilde{x}_t = H(\tilde{z}_t)$

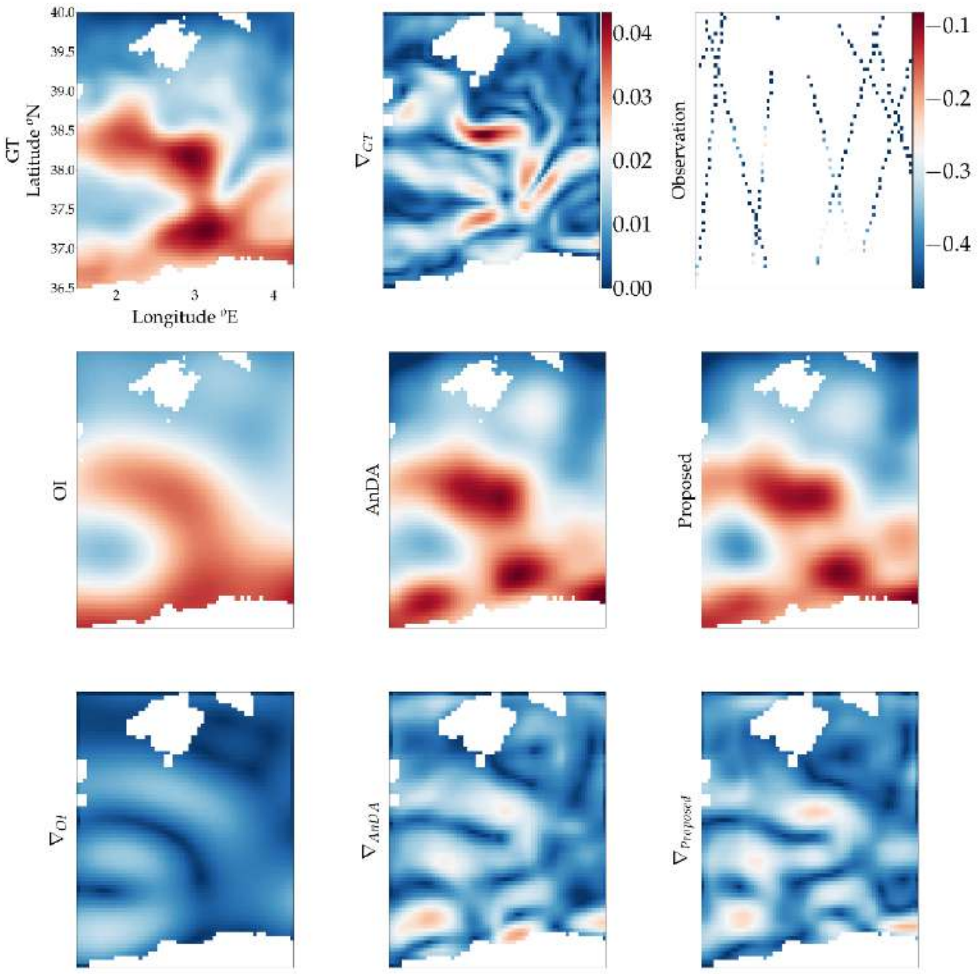
Minimize $|x_t - H(\tilde{z}_t)|$

Maximize $p(x_t)$

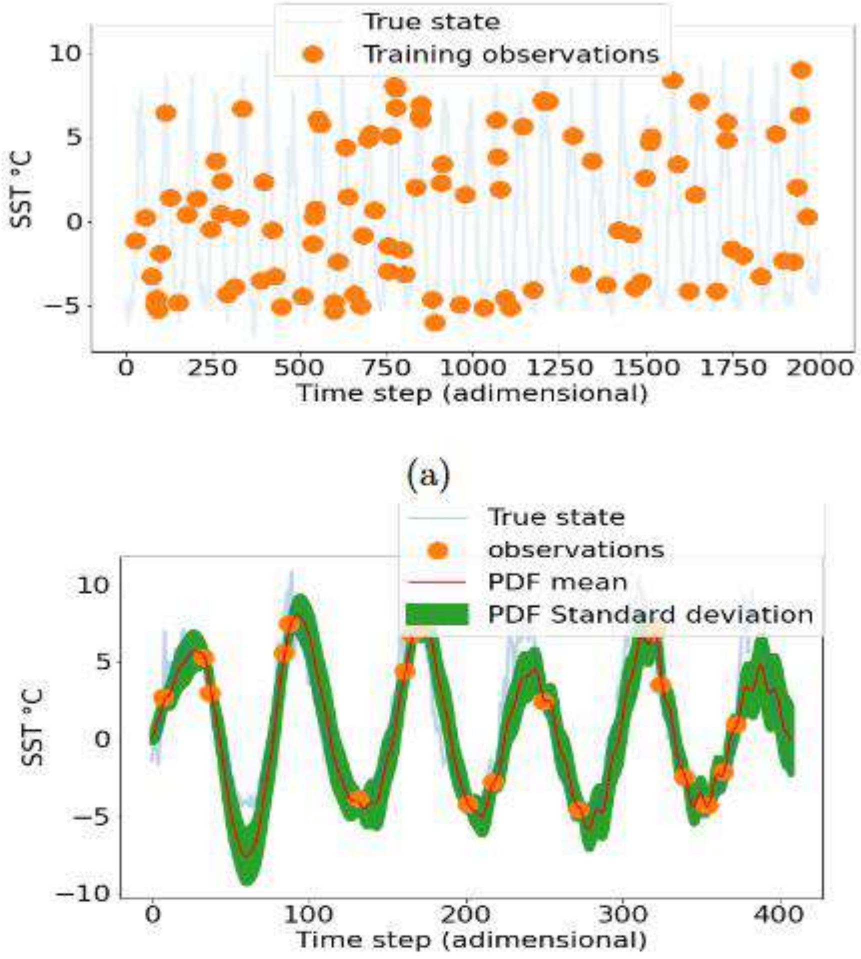
- Naturally deals with missing data
- Can do forecast
- Probabilistic formulation

Higher resolution interpolation, examples

Interpolation results of SLA data in med sea



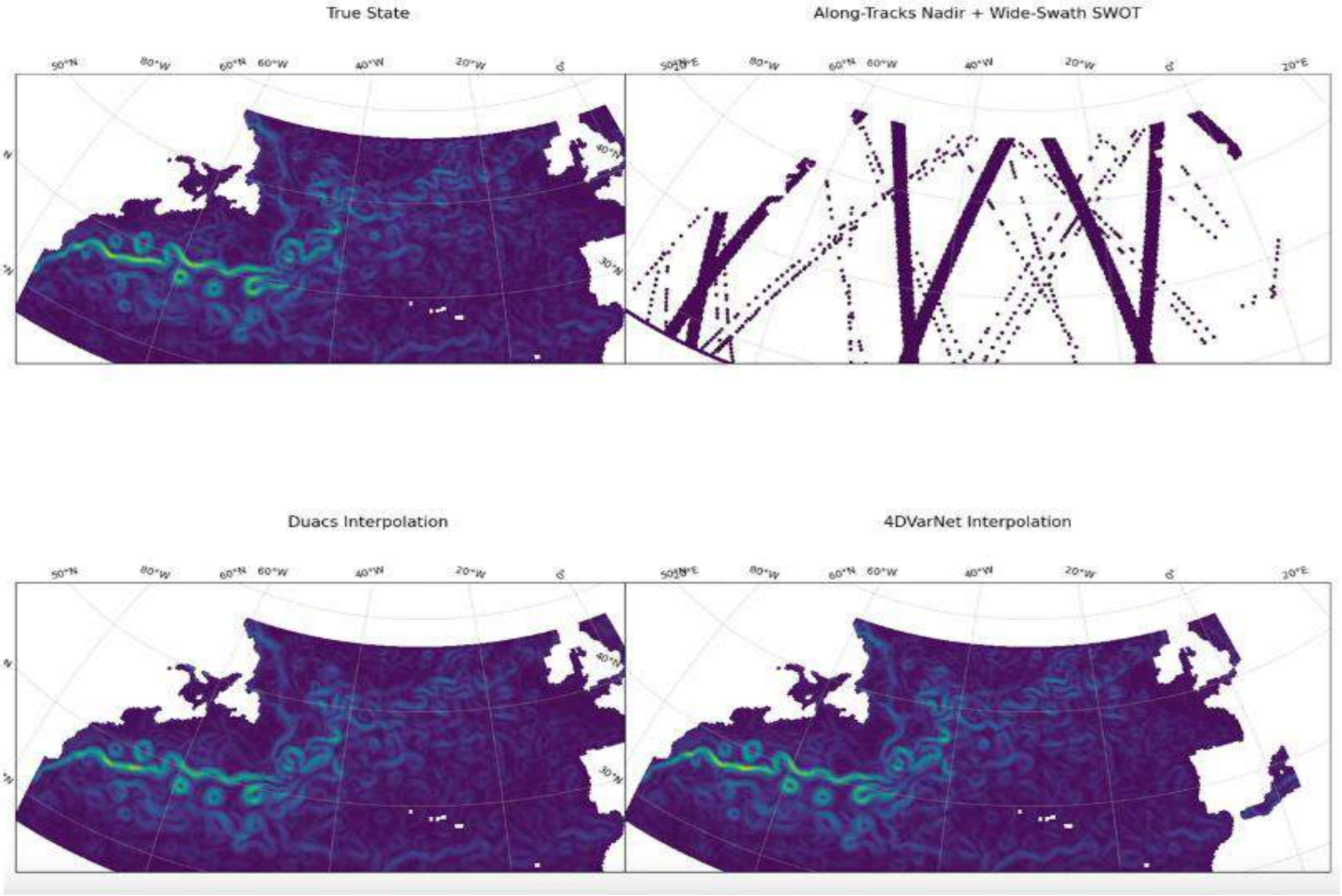
SST anomaly



(a)

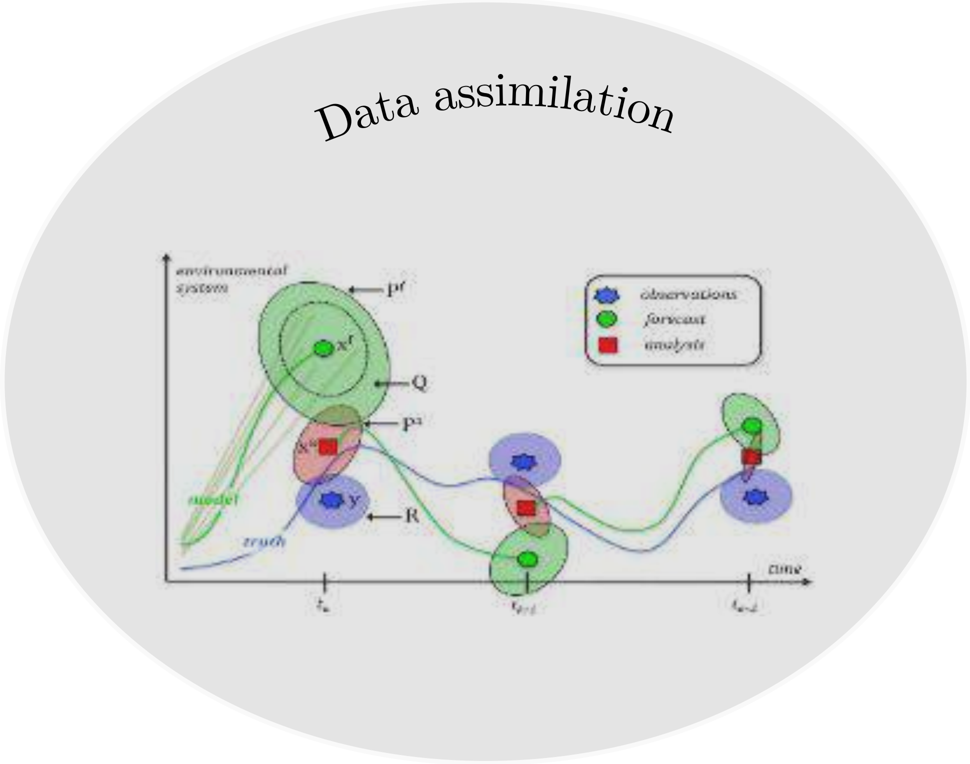
(b)

High


True State
Along-Tracks Nadir + Wide-Swath SWOT
Duacs Interpolation
4DVarNet Interpolation
50°N
80°W
60°N
60°W
40°W
20°W
0°
20°E
40°N
30°N

AI for Data Assimilation

Generative models and data assimilation for modeling dynamical systems



Generative models and data assimilation

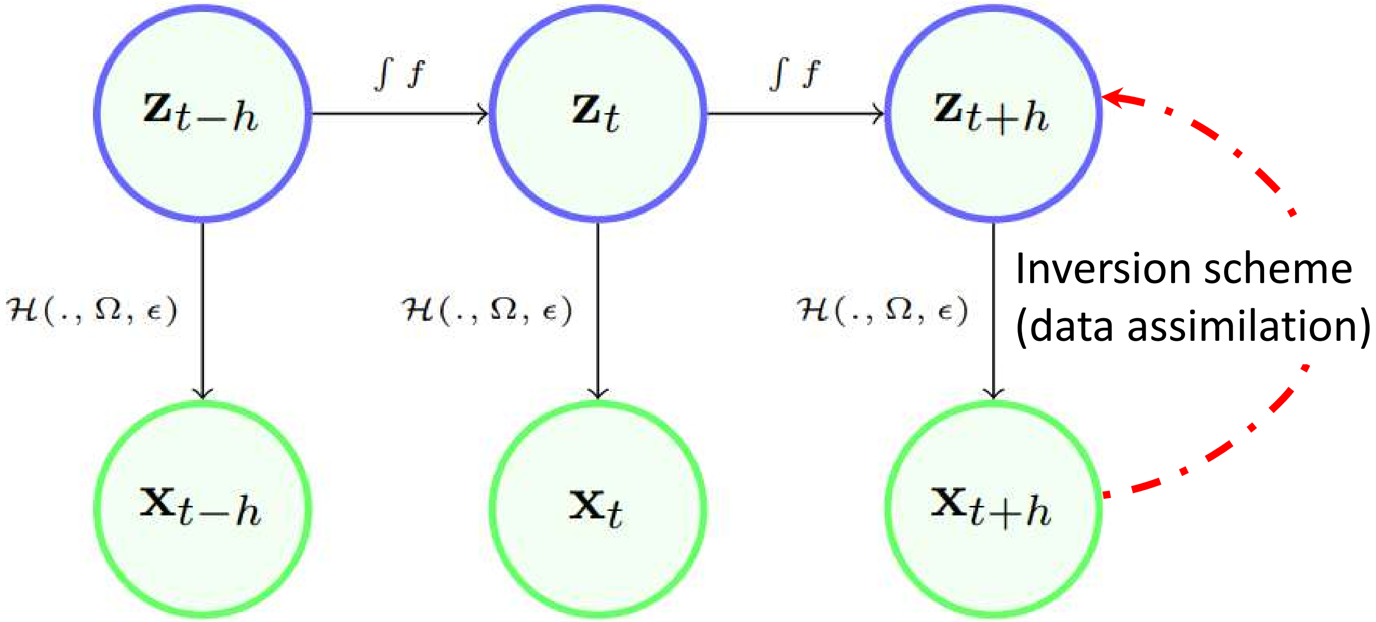
Problem statement

Generative models and data assimilation

Problem statement

- Let us start from the same state space model and assume we want to do interpolation/forecast etc.

- We showed previously how to use DA to solve interpolation/forecasting problems



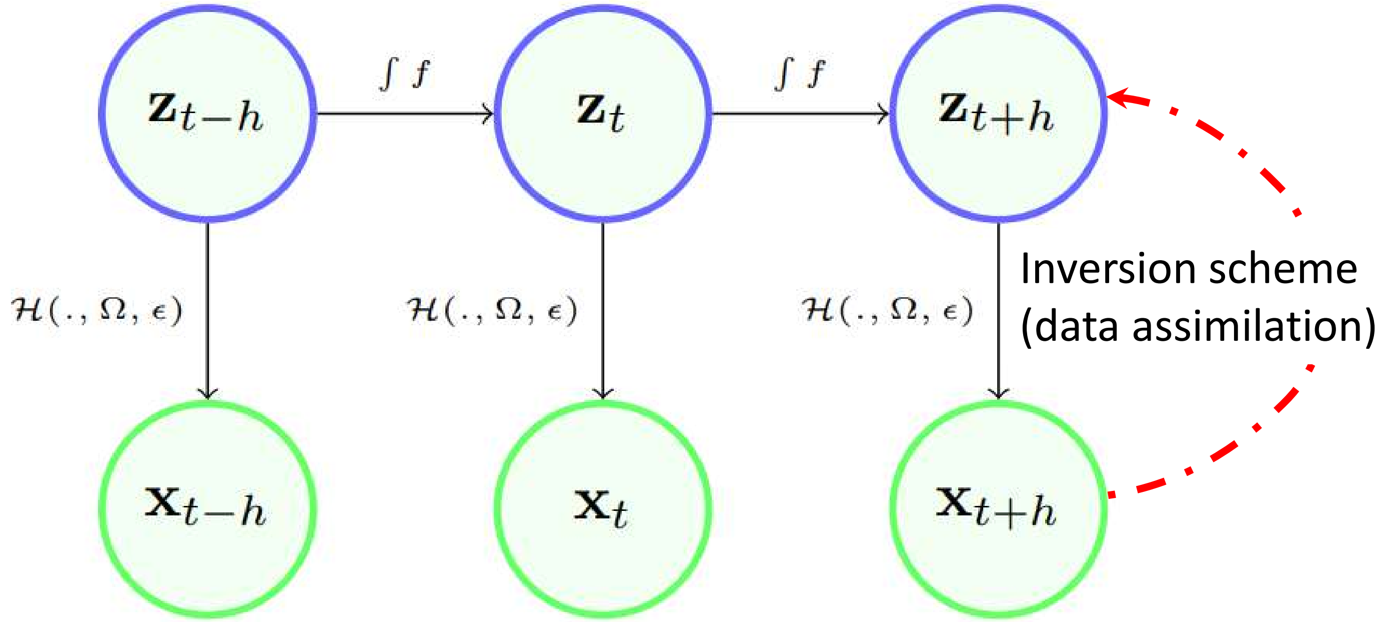
Generative models and data assimilation

Problem statement

- Let us start from the same state space model and assume we want to do interpolation/forecast etc.

- We showed previously how to use DA to solve interpolation/forecasting problems

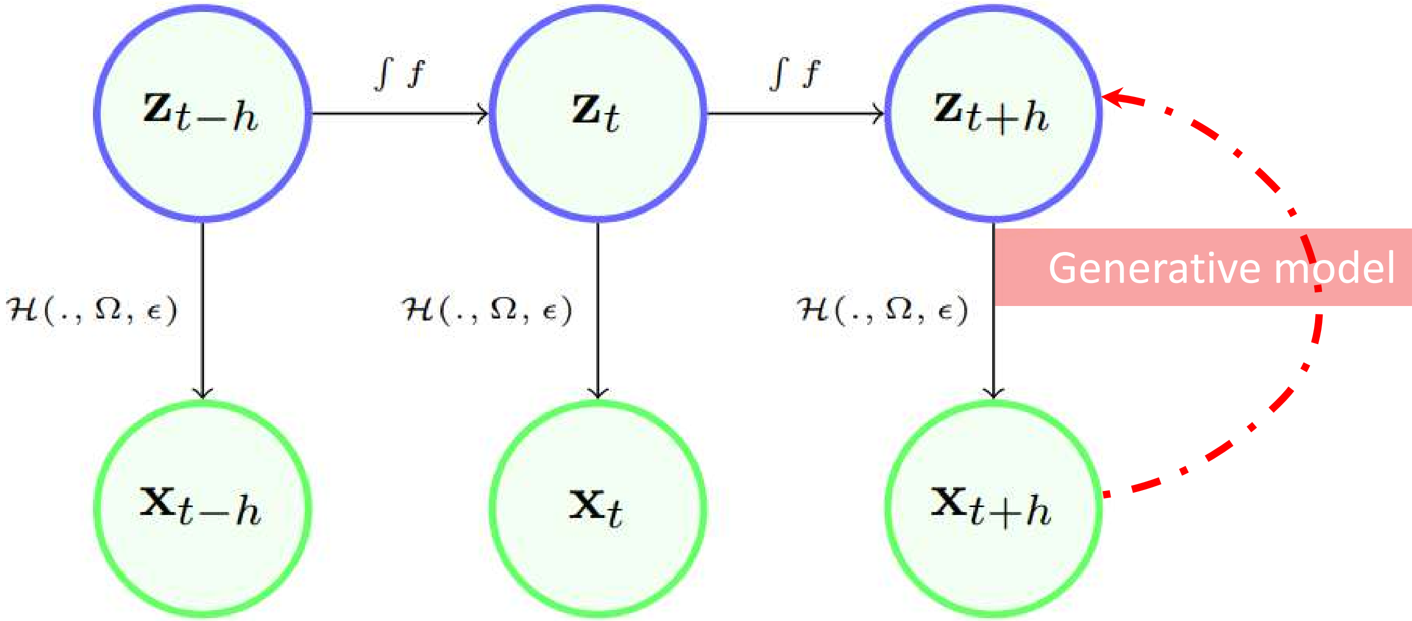
- We can use ideas from generative modeling to do the same tasks



Generative models and data assimilation

Problem statement

- Let us start from the same state space model and assume we want to do interpolation/forecast etc.



- We showed previously how to use DA to solve interpolation/forecasting problems

- We can use ideas from generative modeling to do the same tasks

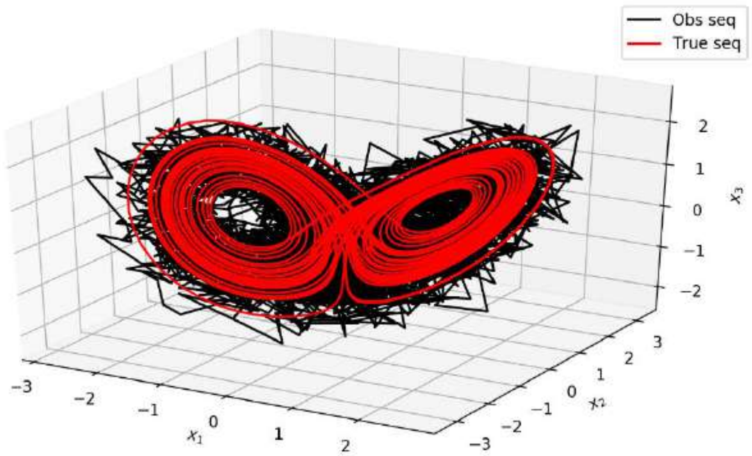
- For instance, we can maximize the evidence lower bound of the SSM:

$$\underbrace{\log p_\theta(\mathbf{x}_{t_0:t_N})}_{\text{Model evidence}} = \underbrace{-\mathbb{E}_{q_\phi}\left[\log q_\phi(\cdot \mid \mathbf{x}_{t_0:t_N})\right] + \mathbb{E}_{q_\phi}\left[\log p_\theta(\cdot, \mathbf{x}_{t_0:t_N})\right]}_{\text{Marginal log Likelihood (ELBO)}} + \underbrace{D_{KL}\left(q_\phi(\cdot \mid \mathbf{x}_{t_0:t_N}) \| p_\theta(\cdot \mid \mathbf{x}_{t_0:t_N})\right)}_{\text{Intratable, } > 0}$$

Generative models and data assimilation

Application example, Learning dynamical systems from noisy/partial observations

Lorenz 63 system



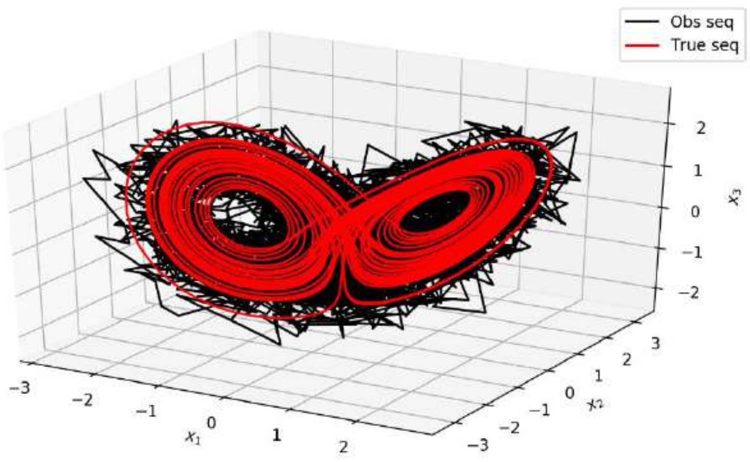
Training both the:

- Dynamical model
- The noise variances
- The Filter

Generative models and data assimilation

Application example, Learning dynamical systems from noisy/partial observations

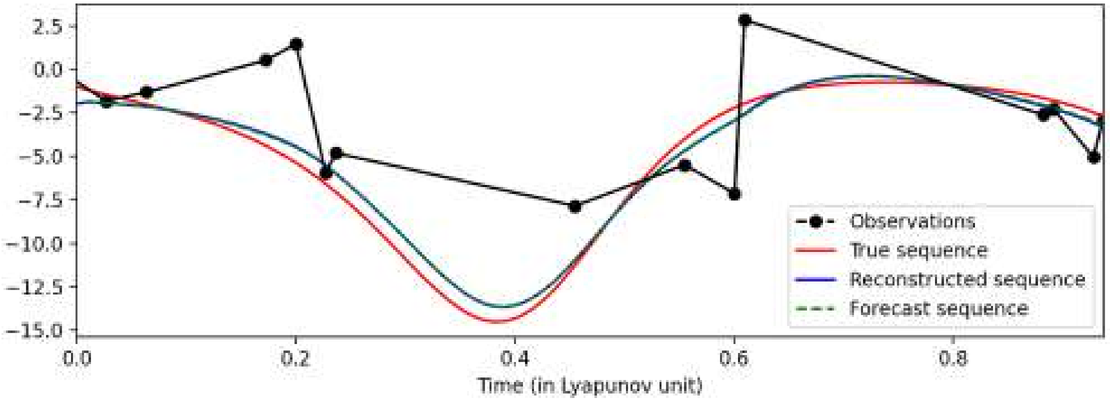
Lorenz 63 system



Training both the:

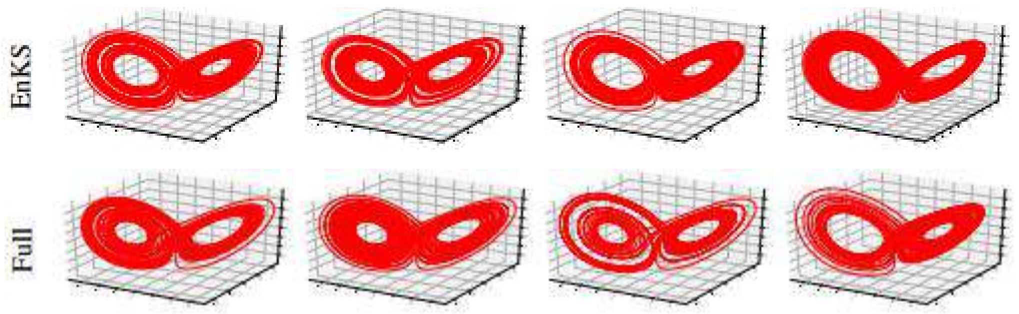
- Dynamical model
- The noise variances
- The Filter

An example of the first dimension of the L63 system reconstructed by the inference module of our model. The observations are noisy (r = 33%) and irregularly sampled

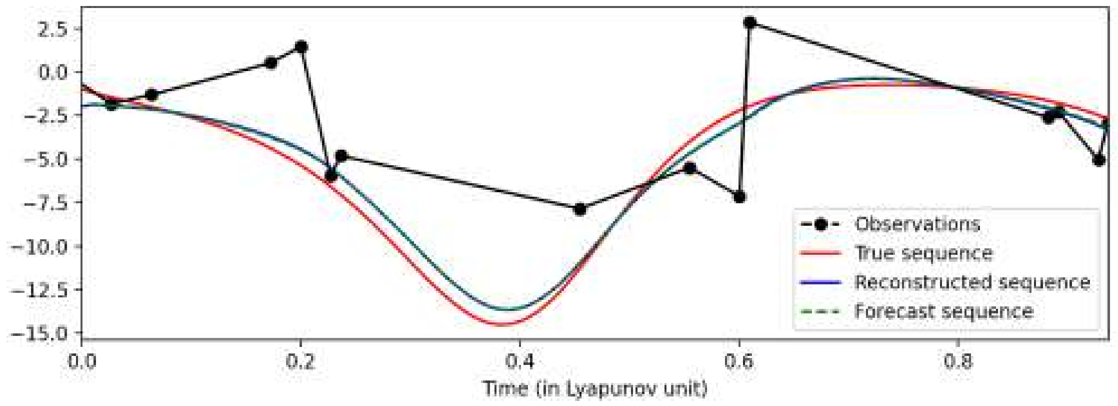


Generative models and data assimilation

Application example, Learning dynamical systems from noisy/partial observations



An example of the first dimension of the L63 system reconstructed by the inference module of our model. The observations are noisy (r = 33%) and irregularly sampled



Generative models and data assimilation

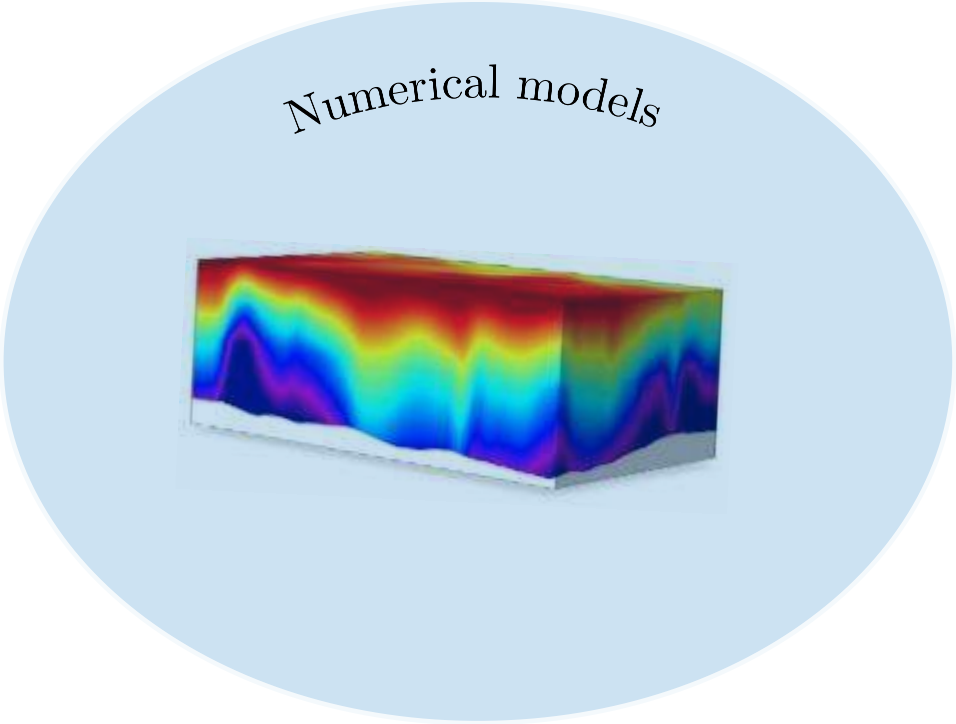
Ongoing works on further links between DA and generative models

$$\underbrace{\log p_\theta(\mathbf{x}_{t_0:t_N})}_{\text{Model evidence}} = \underbrace{-\mathbb{E}_{q_\phi}\left[\log q_\phi(\cdot \mid \mathbf{x}_{t_0:t_N})\right] + \mathbb{E}_{q_\phi}\left[\log p_\theta(\cdot, \mathbf{x}_{t_0:t_N})\right]}_{\text{Marginal log Likelihood (ELBO)}} + \underbrace{D_{KL}\left(q_\phi(\cdot \mid \mathbf{x}_{t_0:t_N}) \| p_\theta(\cdot \mid \mathbf{x}_{t_0:t_N})\right)}_{\text{Intratable, } > 0}$$

- Given the parameters of the SSM, than maximizing the ELBO gives us a filter
- If the filter converges, we even have an estimate of the model evidence (can be used for model selection)
- Benchmark this framework against standard DA for applications such as filtering, smoothing, and parameters/evidence estimation

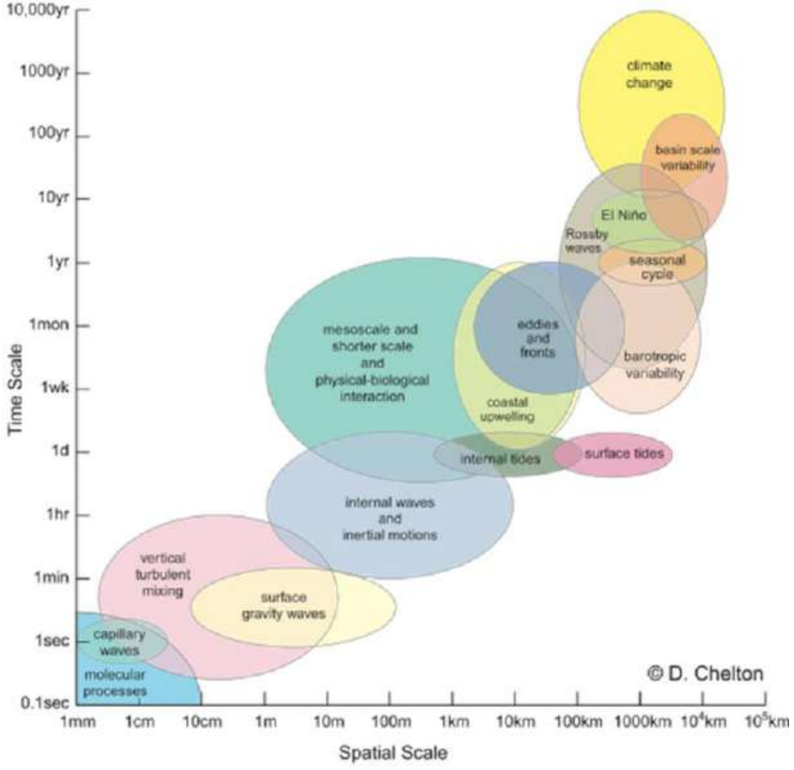
AI for Numerical Models

End-to-end Learning of sub-models in Hybrid Modeling Systems



Online learning of hybrid models
Tuning geophysical models

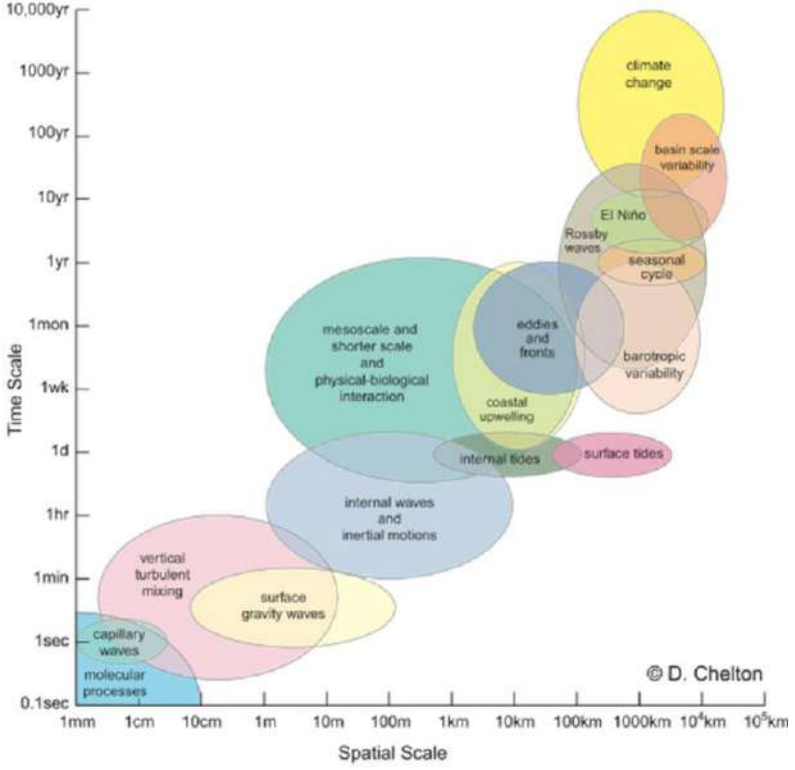
Reality



$$\begin{cases} \frac{\partial \mathbf{u}^{\dagger}{}_t}{\partial t} & = & f(\mathbf{u}_t^{\dagger}) \\ \mathbf{u}_t^{\dagger} & \in & \Omega \end{cases}$$

Online learning of hybrid models
Tuning geophysical models

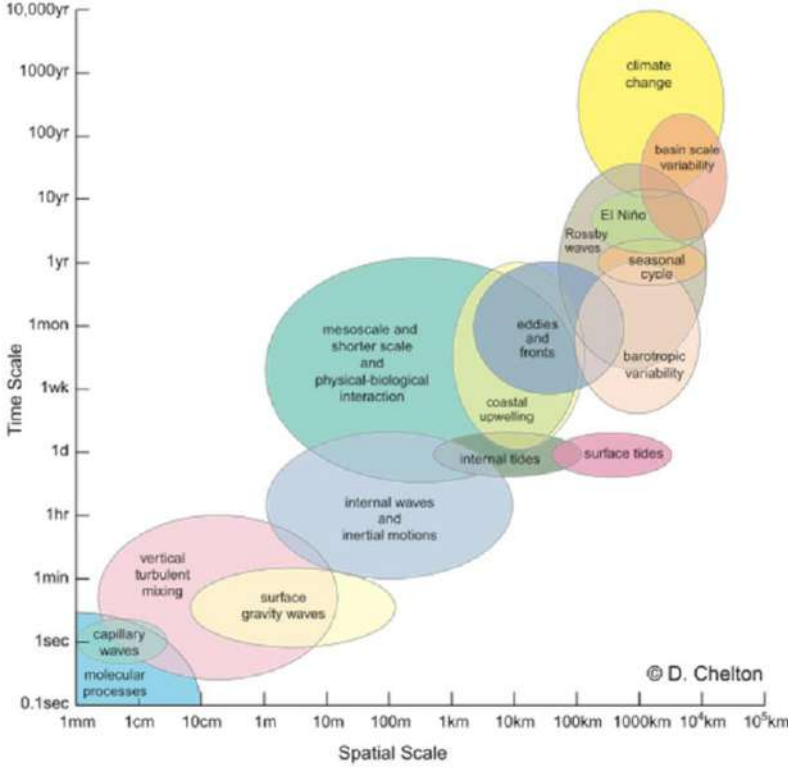
Reality



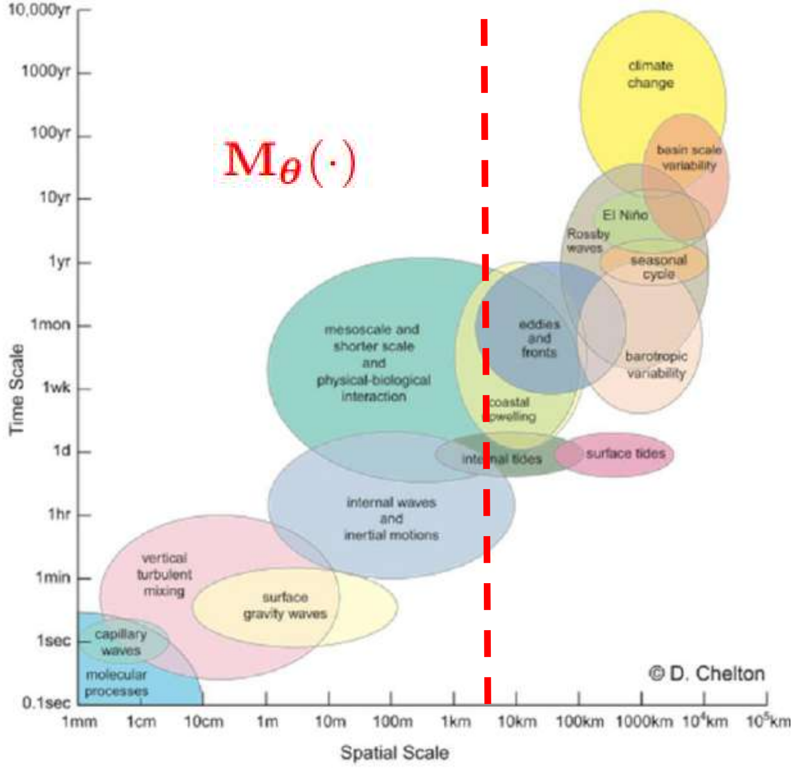
$$\begin{cases} \frac{\partial \mathbf{u}^\dagger{}_t}{\partial t} & = & f(\mathbf{u}_t^\dagger) \\ \mathbf{u}_t^\dagger & \in & \Omega \end{cases}$$

Online learning of hybrid models
Tuning geophysical models

Reality



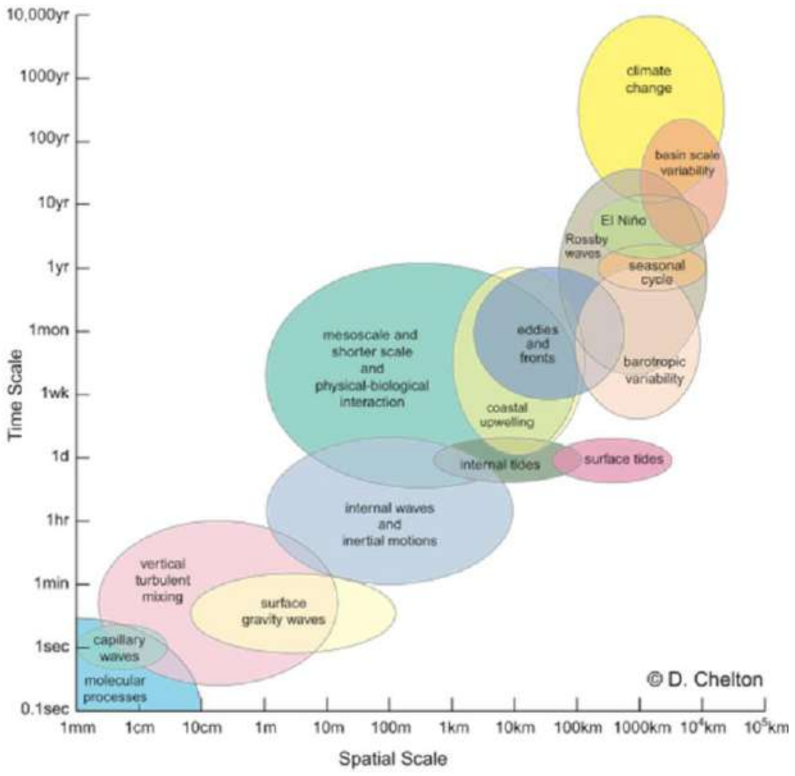
Computer



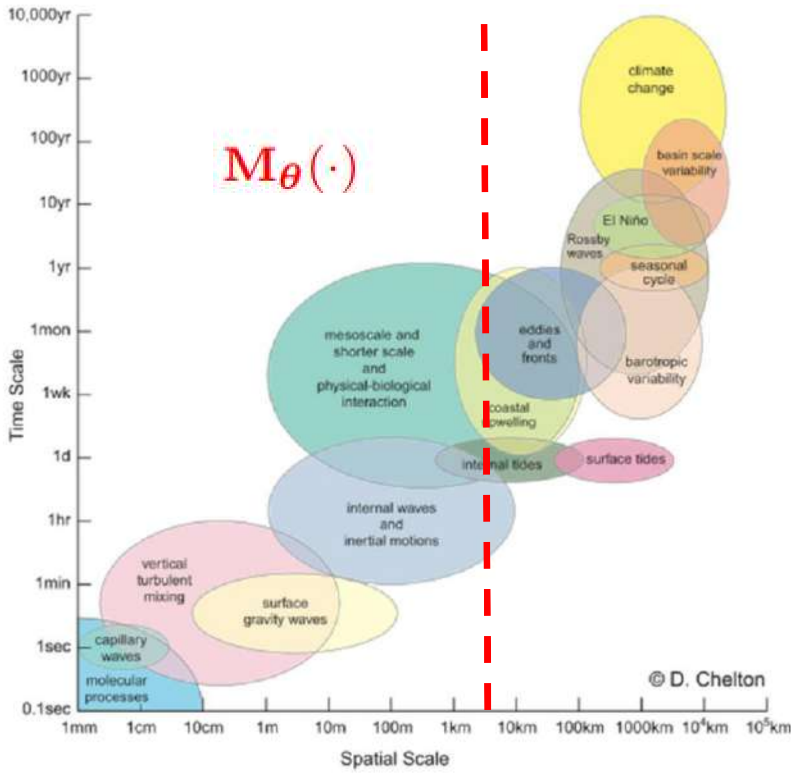
$$\begin{cases} \frac{\partial \mathbf{u}^{\dagger}{}_t}{\partial t} & = & f(\mathbf{u}^{\dagger}_t) \\ \mathbf{u}^{\dagger}_t & \in & \Omega \end{cases}$$

Online learning of hybrid models
Tuning geophysical models

Reality



Computer

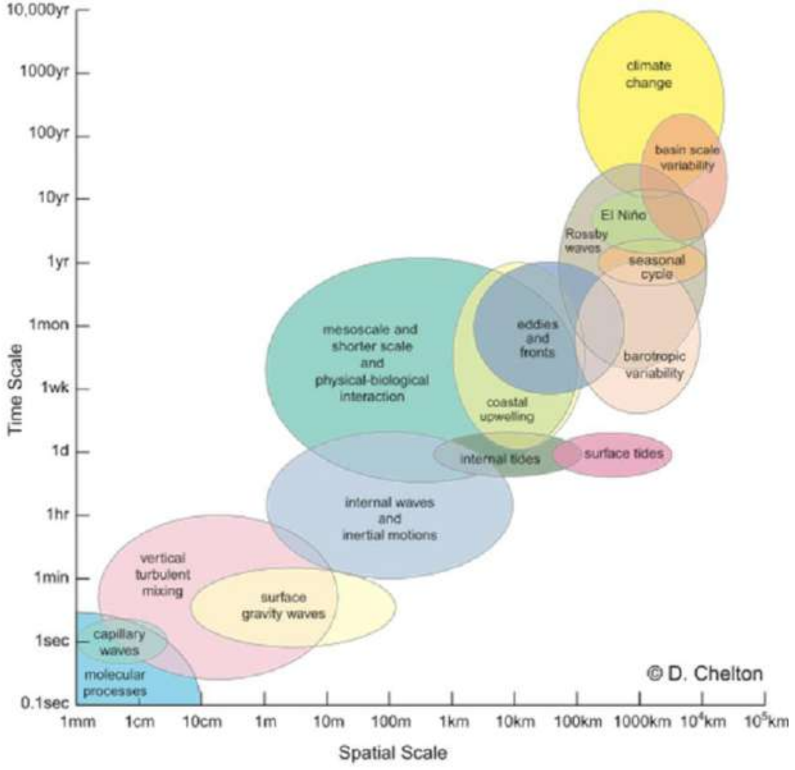


$$\begin{cases} \frac{\partial \mathbf{u}^{\dagger}{}_t}{\partial t} & = & f(\mathbf{u}_t^{\dagger}) \\ \mathbf{u}_t^{\dagger} & \in & \Omega \end{cases}$$

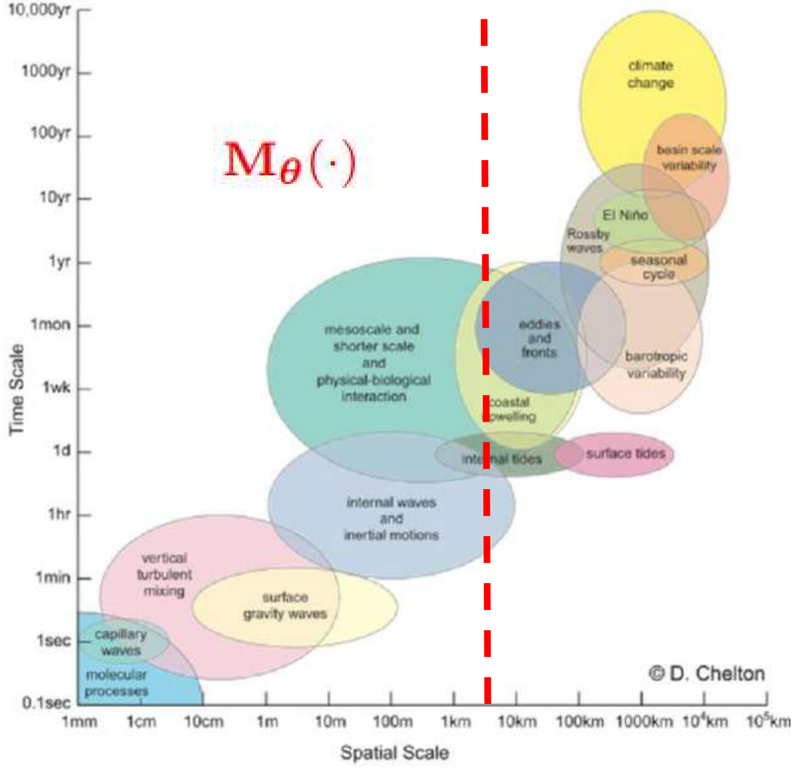
$$\begin{cases} \frac{\partial \mathbf{u}_t}{\partial t} & = & \mathbf{F}(\mathbf{u}_t) + \mathbf{M}_{\boldsymbol{\theta}}(\cdot) \\ \tau(\mathbf{u}_t^{\dagger}) & = & \mathbf{u}_t \in \bar{\Omega} \end{cases}$$

Online learning of hybrid models
Tuning geophysical models

Reality



Computer



$$\begin{cases} \dfrac{\partial \mathbf{u}^{\dagger}{}_t}{\partial t} & = & f(\mathbf{u}_t^{\dagger}) \\ \mathbf{u}_t^{\dagger} & \in & \Omega \end{cases}$$

$$\begin{cases} \dfrac{\partial \mathbf{u}_t}{\partial t} & = & \mathbf{F}(\mathbf{u}_t) + \mathbf{M}_{\boldsymbol{\theta}}(\cdot) \\ \ldots) & = & \mathbf{u}_t \in \bar{\Omega} \end{cases}$$

Online learning with hybrid models ?

Online learning of hybrid models

Training hybrid models

Recall the hybrid model:

$$\begin{cases} \frac{\partial \mathbf{u}_t}{\partial t} & = & \mathbf{F}(\mathbf{u}_t) + \mathbf{M}_{\boldsymbol{\theta}}(\cdot) \\ \tau(\mathbf{u}_t^{\dagger}) & = & \mathbf{u}_t \in \bar{\Omega} \end{cases}$$

Online learning of hybrid models

Training hybrid models

Recall the hybrid model:

Physical core

$$\begin{cases} \frac{\partial \mathbf{u}_t}{\partial t} & = \mathbf{F}(\mathbf{u}_t) + \color{red}{\mathbf{M}_{\boldsymbol{\theta}}(\cdot)} \\ \tau(\mathbf{u}_t^{\dagger}) & = \mathbf{u}_t \in \bar{\Omega} \end{cases}$$

Online learning of hybrid models

Training hybrid models

Recall the hybrid model:

Physical core — Sub-model

$$\begin{cases} \frac{\partial \mathbf{u}_t}{\partial t} & = \mathbf{F}(\mathbf{u}_t) + \color{red}{\mathbf{M}_{\boldsymbol{\theta}}(\cdot)} \\ \tau(\mathbf{u}_t^{\dagger}) & = \mathbf{u}_t \in \bar{\Omega} \end{cases}$$

Training hybrid models

Recall the hybrid model:

Physical core — Sub-model

$$
\begin{cases} \dfrac{\partial \mathbf{u}_t}{\partial t} & = \quad \mathbf{F}(\mathbf{u}_t) + \color{red}{\mathbf{M}_{\boldsymbol{\theta}}(\cdot)} \\ \tau(\mathbf{u}_t^{\dagger}) & = \quad \mathbf{u}_t \in \bar{\Omega} \end{cases}
$$

The model is solved using some appropriate numerical solver:

$$
\Psi^n(\mathbf{u}_t) = \mathbf{u}_{t+nh} \approx \mathbf{u}_t + \int_{t_0}^{t0+nh} (\mathbf{F}(\mathbf{u}_t) + \mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t))dt
$$

Online learning of hybrid models

Training hybrid models

Recall the hybrid model:

Physical core — Sub-model

$$\begin{cases} \frac{\partial \mathbf{u}_t}{\partial t} & = & \mathbf{F}(\mathbf{u}_t) + \color{red}{\mathbf{M}_{\boldsymbol{\theta}}(\cdot)} \\ \tau(\mathbf{u}_t^{\dagger}) & = & \mathbf{u}_t \in \bar{\Omega} \end{cases}$$

The model is solved using some appropriate numerical solver:

$$\Psi^n(\mathbf{u}_t) = \mathbf{u}_{t+nh} \approx \mathbf{u}_t + \int_{t_0}^{t0+nh} (\mathbf{F}(\mathbf{u}_t) + \mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t))dt$$

- How to calibrate $\boldsymbol{\theta}$ the parameters of the model ?

Online learning of hybrid models

Training hybrid models, offline vs online learning

Training hybrid models, offline vs online learning

Offline learning:

Training hybrid models, offline vs online learning

Offline learning:

- Define and compute the parameterization term: R_t

Training hybrid models, offline vs online learning

Offline learning:

- Define and compute the parameterization term: R_t
- $\boldsymbol{\theta}$ is estimated by matching the deep learning model $M_{\boldsymbol{\theta}}$ to R_t *i.e.*

Offline learning:

- Define and compute the parameterization term: $\boldsymbol{R_t}$
- $\boldsymbol{\theta}$ is estimated by matching the deep learning model $\boldsymbol{M_\theta}$ to $\boldsymbol{R_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t), \mathcal{R}_t, \boldsymbol{\theta})$$

Training hybrid models, offline vs online learning

Offline learning:

- Define and compute the parameterization term: R_t
- $\boldsymbol{\theta}$ is estimated by matching the deep learning model $M_{\boldsymbol{\theta}}$ to R_t *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t), \mathcal{R}_t, \boldsymbol{\theta})$$

- Once the parameterization term R_t is computed, this method is very simple to test;

Training hybrid models, offline vs online learning

Offline learning:

- Define and compute the parameterization term: R_t
- $\boldsymbol{\theta}$ is estimated by matching the deep learning model $M_{\boldsymbol{\theta}}$ to R_t *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L};\ \text{where}\ \mathcal{L} = Q(\mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t), \mathcal{R}_t, \boldsymbol{\theta})$$

- Once the parameterization term R_t is computed, this method is very simple to test;
- Used in several state-of-the-art works (example, Guan et al. (2022, 2023));

Training hybrid models, offline vs online learning

Offline learning:

- Define and compute the parameterization term: R_t
- $\boldsymbol{\theta}$ is estimated by matching the deep learning model $M_{\boldsymbol{\theta}}$ to R_t *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t), \mathcal{R}_t, \boldsymbol{\theta})$$

- Once the parameterization term R_t is computed, this method is very simple to test;
- Used in several state-of-the-art works (example, Guan et al. (2022, 2023));
- Can be subject to issues when coupling with the solver Frezat et al. (2022); Guan et al. (2022);

Training hybrid models, offline vs online learning

Offline learning:

- Define and compute the parameterization term: R_t
- $\boldsymbol{\theta}$ is estimated by matching the deep learning model $M_{\boldsymbol{\theta}}$ to R_t *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t), \mathcal{R}_t, \boldsymbol{\theta})$$

- Once the parameterization term R_t is computed, this method is very simple to test;
- Used in several state-of-the-art works (example, Guan et al. (2022, 2023));
- Can be subject to issues when coupling with the solver Frezat et al. (2022); Guan et al. (2022);
- Just an emulator of a parameterization term need to have access to R_t, do not use historical data

Training hybrid models, offline vs online learning

Training hybrid models, offline vs online learning

Online learning:

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L};\ \text{where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

$$= \underbrace{\frac{\partial Q(\cdot,\cdot,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}}_{\text{Gradient of the regularization}} + \underbrace{\frac{\partial Q(\cdot, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \cdot)}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \Psi}}_{\text{Gradient of the online cost w.r.t. } \Psi} \underbrace{\frac{\partial \Psi^n(\mathbf{u}_t)}{\partial \boldsymbol{\theta}}}_{\text{Gradient of the solver}}$$

Online learning of hybrid models

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver
  - Rewrite the solver in Pytorch ?

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver
  - Rewrite the solver in Pytorch ?
  - Differentiable emulators ?

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver
  - Rewrite the solver in Pytorch ?
  - Differentiable emulators ?
  - Adjoint sensitivity ?

Training hybrid models, offline vs online learning

Online learning:

- $\boldsymbol{\theta}$ is estimated by matching the numerical integration of the model to some observations $\boldsymbol{y_t}$ *i.e.*

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}; \text{ where } \mathcal{L} = Q(\mathbf{y}_{t+nh}, \mathbf{g}(\Psi^n(\mathbf{u}_t)), \boldsymbol{\theta})$$

- Allows for an end-to-end learning ;
- (experimental) Better stability Frezat et al. (2022)
- Need to do back-propagation through the solver
  - Rewrite the solver in Pytorch ?
  - Differentiable emulators ?
  - Adjoint sensitivity ?
  - Derivative free methods ?

Online learning of hybrid models
Euler Gradient Approximation

- Let us consider an explicit Euler solver Ψ_E, a single step integration using Ψ_E can be written as:

$$\mathbf{u}_{t+h} = \Psi(\mathbf{u}_t)$$

where

$$\Psi_E(\mathbf{u}_t) = \mathbf{u}_t + h(\mathbf{F}(\mathbf{u}_t) + \mathbf{M}_{\boldsymbol{\theta}}(\mathbf{u}_t))$$

- Assuming that the solver Ψ has order $p \geq 1$, we can write for any initial condition:

$$\begin{aligned}\mathbf{u}_{t+h} &= \Psi(\mathbf{u}_t) \\ &= \Psi_E(\mathbf{u}_t) + O(h^2)\end{aligned}$$

Online learning of hybrid models
Euler Gradient Approximation

- By using this approximation, we can show that the gradient of the solver can be decomposed as (for a fixed n):

$$\frac{\partial}{\partial \boldsymbol{\theta}}\Psi^n(\mathbf{u}_t) = \sum_{j=1}^{j=n-1}\Big(\prod_{i=1}^{i=n-j}\underbrace{\frac{\partial \Psi(\Psi^{n-i}(\mathbf{u}_t))}{\partial \Psi^{n-i}(\mathbf{u}_t)}}_{\textit{Jacobian of the flow}}\Big) h \underbrace{\frac{\partial}{\partial \boldsymbol{\theta}}\mathbf{M}_{\boldsymbol{\theta}}(\Psi^{j-1}(\mathbf{u}_t))}_{\textit{Gradient of the sub-model}} + h\frac{\partial}{\partial \boldsymbol{\theta}}\mathbf{M}_{\boldsymbol{\theta}}(\Psi^{n-1}(\mathbf{u}_t)) + O(h^2)$$

- If we approximate the Jacobian (we can use a static/ensemble approximation, a TLM if any), we can compute the gradients only using the gradient of the sub-model;
- And for n fixed, the gradients converge to the true ones quadratically in h;

QG turbulence

- The dimensionless governing equations in the vorticity (ω) and stream function (ψ) formulation in a doubly periodic square domain with length L = 2π are:

$$\frac{\partial \omega_t}{\partial t} + \mathcal{A}(\omega_t, \psi_t) = \frac{1}{\mathrm{Re}}\nabla^2 \omega_t - f - r\omega_t$$
$$\nabla^2 \psi_t = -\omega_t$$

- where $\mathcal{A}(\omega_t, \psi_t)$r represents the nonlinear advection term:

$$\mathcal{A}(\omega_t, \psi_t) = \frac{\partial \psi_t}{\partial y}\frac{\partial \omega_t}{\partial x} - \frac{\partial \psi_t}{\partial x}\frac{\partial \omega_t}{\partial y}$$

- and f represents a deterministic forcing:

$$f(x, y) = k_f\left[\cos(k_f x) + \cos(k_f y)\right]$$

QG turbulence, LES

- The dimensionless governing equations in the vorticity (ω) and stream function (ψ) formulation in a doubly periodic square domain with length L = 2π are:

$$\frac{\partial \omega_t}{\partial t} + \mathcal{A}(\omega_t, \psi_t) = \frac{1}{\mathrm{Re}}\nabla^2 \omega_t - f - r\omega_t \qquad \xrightarrow{\overline{(\cdot)}} \qquad \frac{\partial \bar{\omega}_t}{\partial t} + \mathcal{A}(\bar{\omega}_t, \bar{\psi}_t) = \frac{1}{\mathrm{Re}}\nabla^2 \bar{\omega}_t - \bar{f} - r\bar{\omega}_t + \underbrace{\mathcal{A}(\bar{\omega}_t, \bar{\psi}_t) - \overline{\mathcal{A}(\omega_t, \psi_t)}}_{\Pi_t \approx \mathbf{M}_{\boldsymbol{\theta}}}$$

$$\nabla^2 \psi_t = -\omega_t \qquad\qquad \nabla^2 \bar{\psi}_t = -\bar{\omega}_t$$

- Model the subgrid-scale term $\Pi_t \approx \mathbf{M}_{\boldsymbol{\theta}}$

QG turbulence

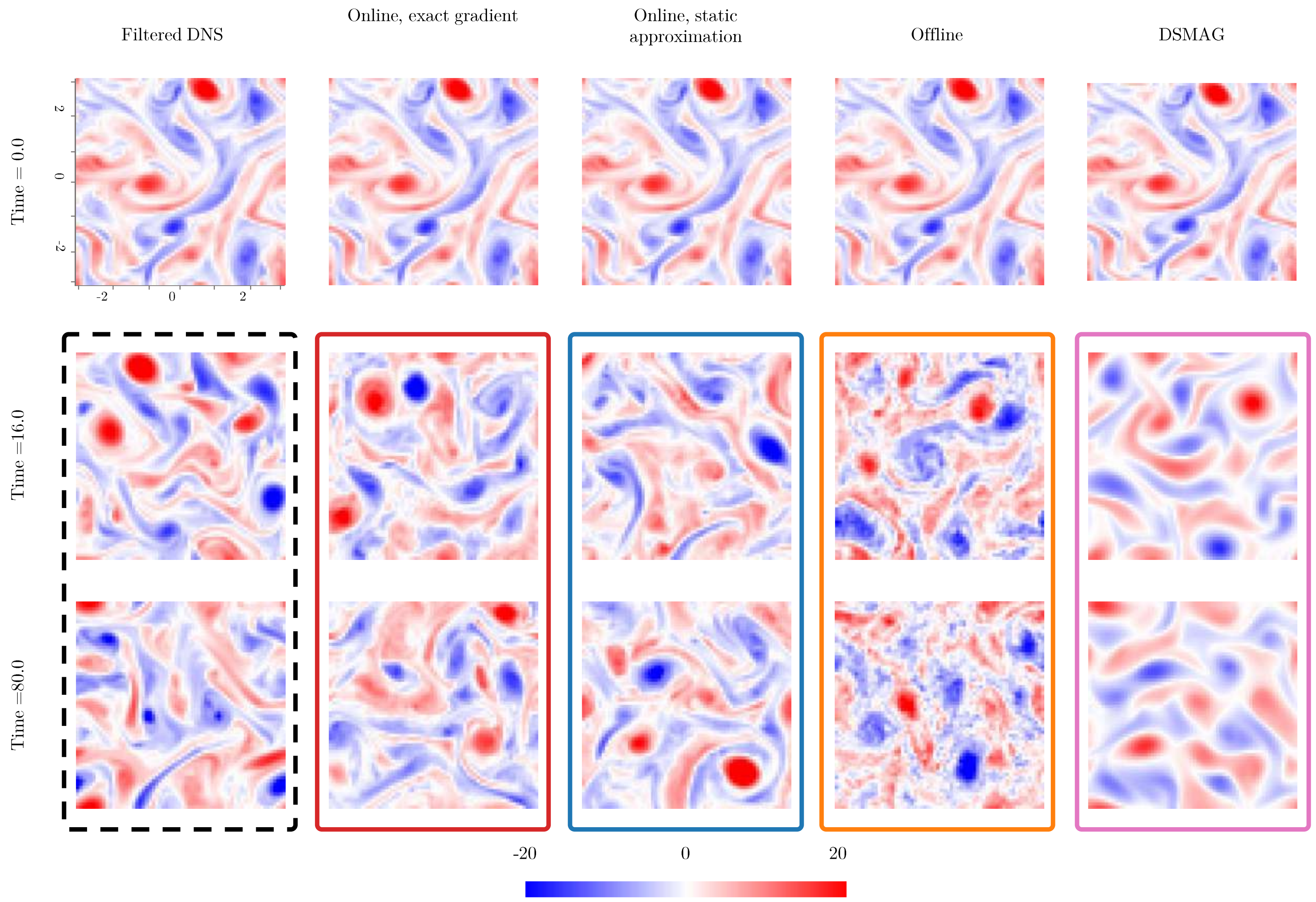
Flow configuration:

- High resolution grid : 1024×1024.
- Low resolution : 64×64.
- Re : 20000, r = 0.1, kf = 4;

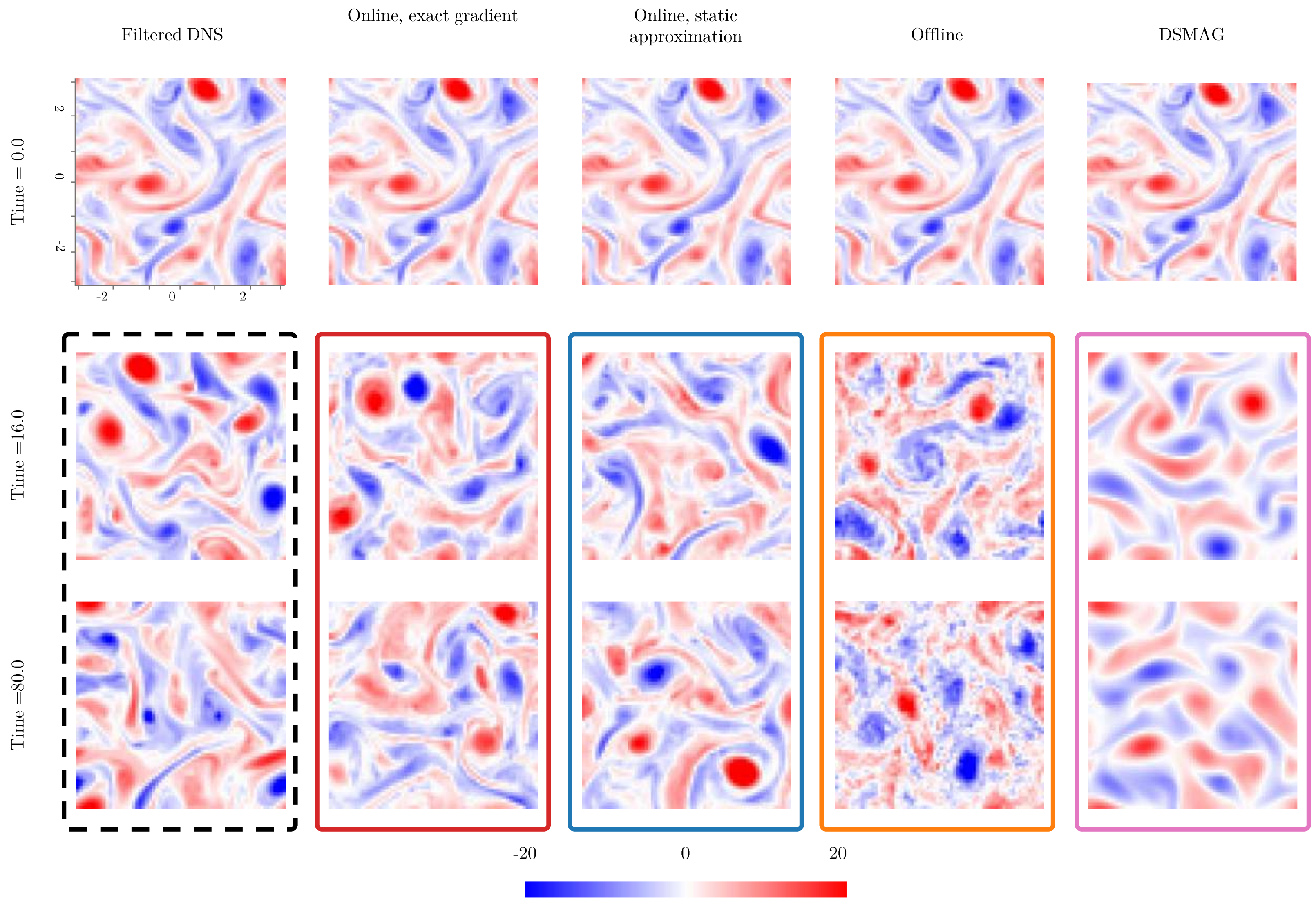
Tested models:

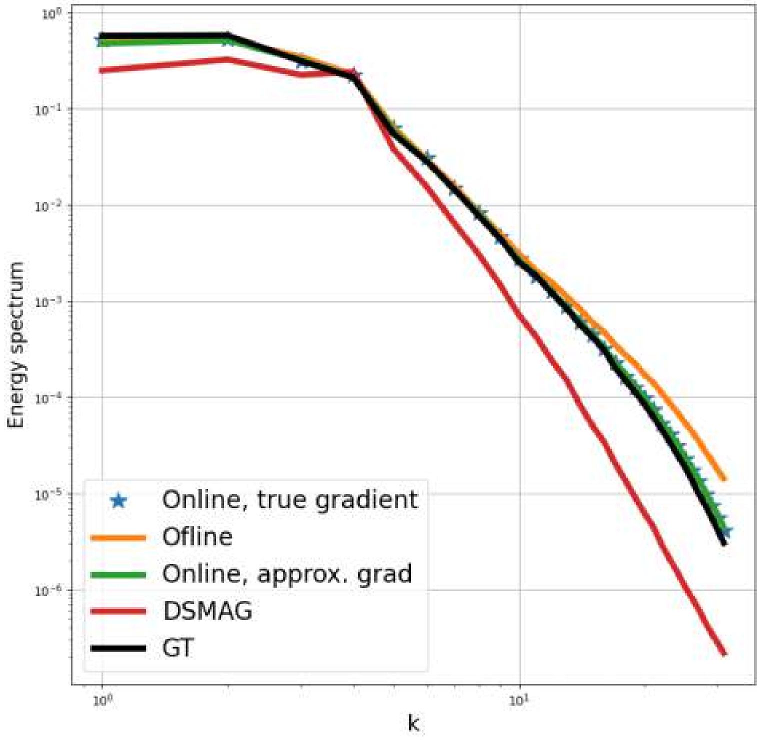
- Online learning with exact gradient;
- Online learning with approximate gradient;
- Offline learning;
- Dynamic Smagorinsky (DSMAG)

QG turbulence



QG turbulence





QG turbulence

- We proposed a simple gradient estimator for learning online hybrid models;
- Proposed methodology does not rely on a differentiable physical model, and can (in theory) be applied on non-differentiable CFD/GFD codes;
- Can use better Jacobian approximation and can be extended to definition of non additive correction terms;
- Interpretability/constraining the sub-model ?
- Multiple (stochastic) sub-models ?

Key points and perspectives

- IA can be used to improve models/data
- One of the key points is to formulate the problem we want to solve
- Towards problem standardization, benchmarks based on ocean data ?