# Deep learning for sea surface height reconstruction from multi-variate satellite observations

*Ph.D. defense of*

Théo **ARCHAMBAULT**

*directed and Co-directed by*

Dominique **BÉRÉZIAT** and Anastase **CHARANTONIS**

*at*

LIP6, LOCEAN, Sorbonne Université

Jury:

Alexander **BARTH**, Maître de recherches FNRS, Université de Liège, Rapporteur

Emmanuel **COSME**, Professeur assistant, Université Grenoble Alpes, Rapporteur

Claire **MONTELEONI**, Directrice de recherche, INRIA Paris, Examinatrice

Cécile **MALLET**, Professeure, LATMOS, Examinatrice

Alexandre **STEGNER**, Professeur, CNRS - Ecole Polytechnique - Amphitrite, Examinateur

Maxime **BALLAROTTA**, Docteur, CLS-Groupe, Invité

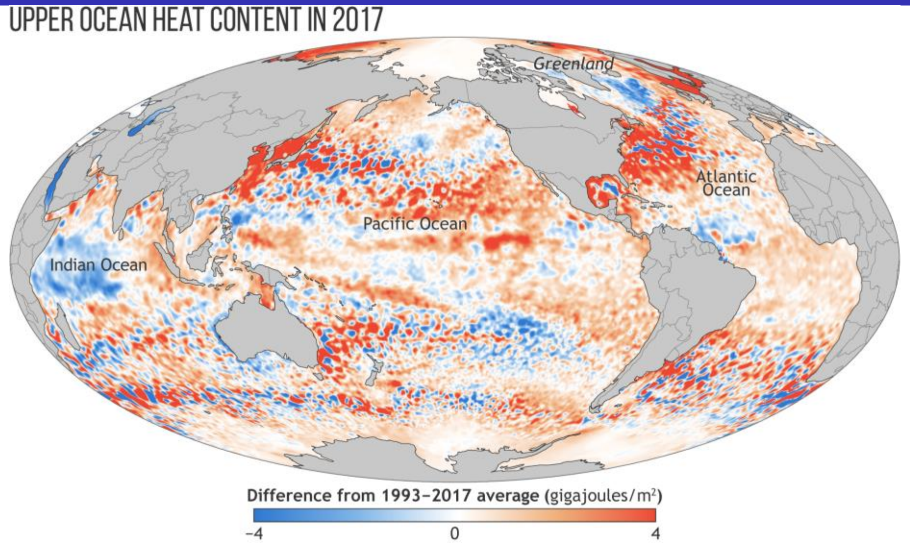
04/10/2024

# Introduction: observations of the oceans

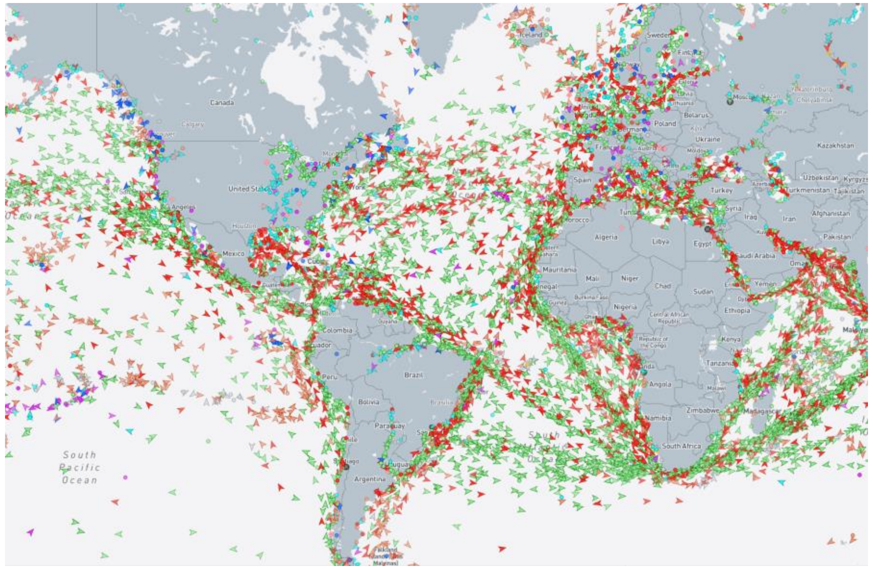
The oceans:

- Climate regulation: absorb and regulate heat
- Meteorology

- Marine applications
  - Commercial exchanges




Source: NOAA Climate.gov adapted from the Climate 2017



Source: https://www.marinetraffic.com/

# Introduction: observations of the oceans

The oceans:

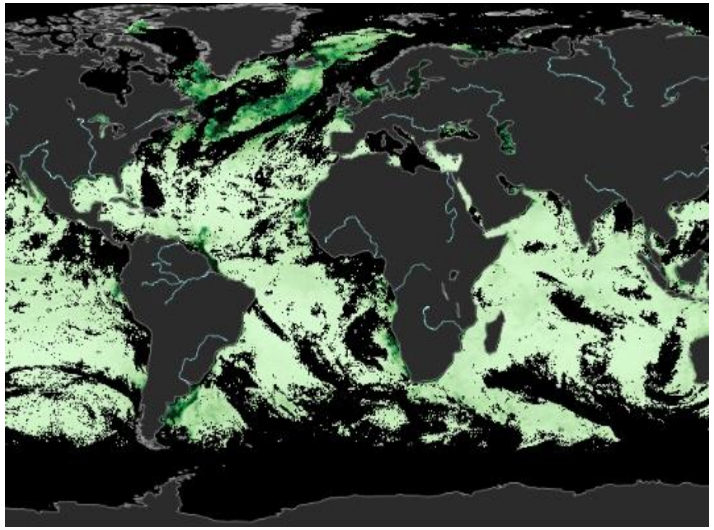
- Climate regulation: absorb and regulate heat
- Meteorology

- Marine applications
  - Commercial exchanges
- Their observation is challenging:
  - Many physical variables
  - Wide areas
  - Many layers

Satellite remote sensing

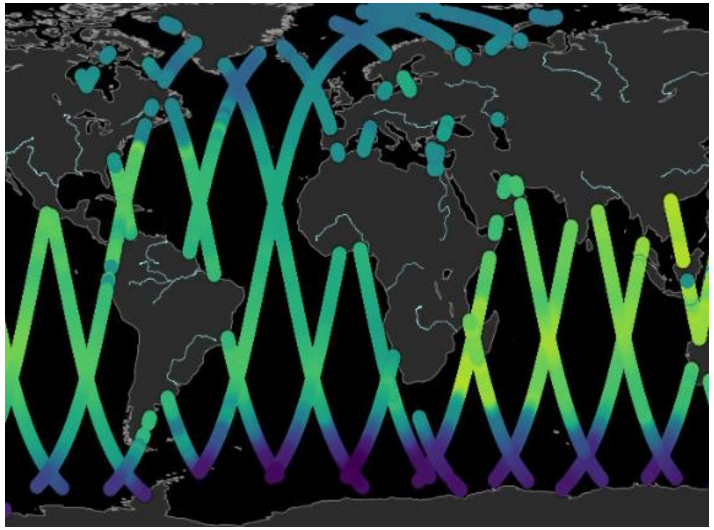
- Multi-variate observations
- High-dimension data

Chlorophyll-A



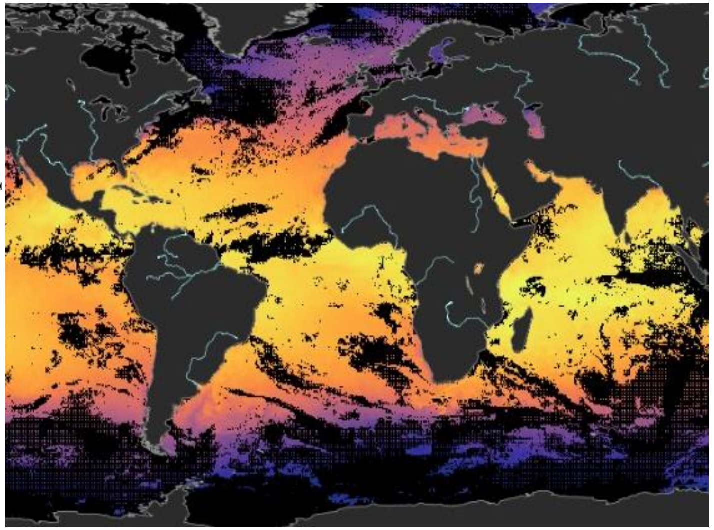
Source: CMEMS, doi: 10.48670/moi-00278

Sea Surface Height



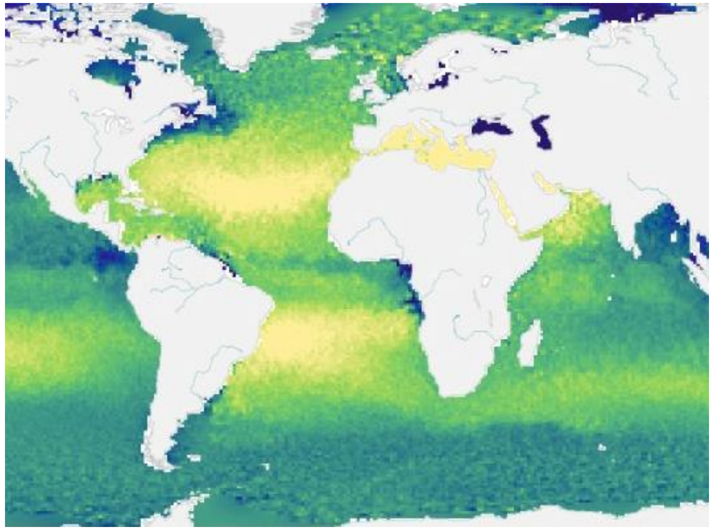
Source: CMEMS, doi: 10.48670/moi-00278

Sea Surface Temperature



Source: CMEMS, doi: 10.48670/moi-00164

Salinity



Source: CMEMS, doi: 10.1175/JTECH-D-20-0093.1

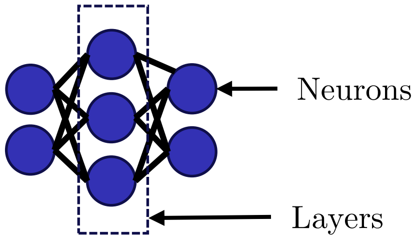
# Introduction : Deep Learning

Data Assimilation: Combining data and physical knowledge

- Optimal interpolation, Kalman filter, variational methods, nudging, ...

Machine learning: solve complex tasks using numerous examples.

- Deep Learning: Artificial Neural Networks



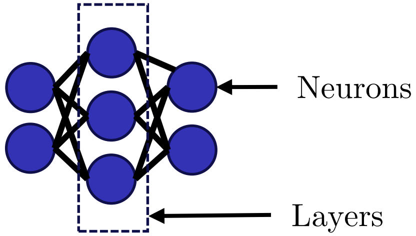
# Introduction : Deep Learning

Data Assimilation: Combining data and physical knowledge

- Optimal interpolation, Kalman filter, variational methods, nudging, ...

Machine learning: solve complex tasks using numerous examples.

- Deep Learning: Artificial Neural Networks




- Useful in many task involving high-dimension data

**Text generation**
*"Deep learning is a type of machine learning that uses layered neural networks to automatically learn patterns from data. Each layer processes information to form increasingly complex representations, enabling tasks like image recognition and natural language processing without manual feature design."*, ChatGPT, OpenAI

**Image segmentation**
source: https://blogs.nvidia.com/blog/drive-labs-panoptic-segmentation/

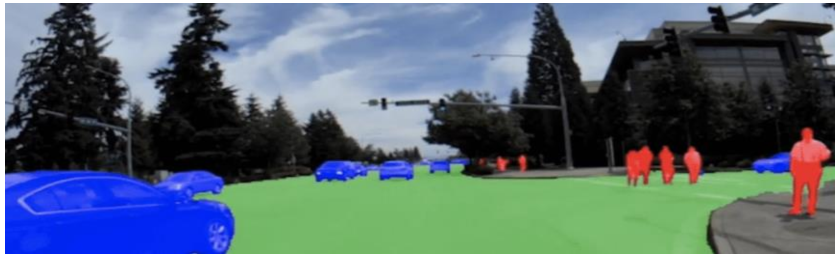


**Image reconstruction**
source: https://research.nvidia.com/publication/2017-10_learning-super-resolve-blurry-face-and-text-images

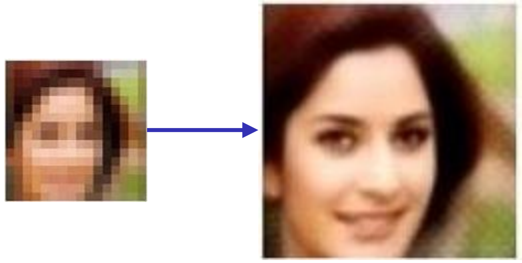


**Image generation:** *"Four pirates and a dragon on a boat"*, source: midjourney.com

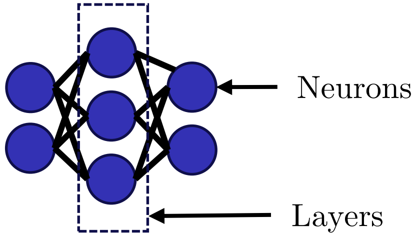
# Introduction : Deep Learning

Data Assimilation: Combining data and physical knowledge

- Optimal interpolation, Kalman filter, variational methods, nudging, ...

Machine learning: solve complex tasks using numerous examples.

- Deep Learning: Artificial Neural Networks




- Useful in many task involving high-dimension data
- A growing interest in geosciences
  - High dimension data
  - Complex relationships

**Text generation**
*"Deep learning is a type of machine learning that uses layered neural networks to automatically learn patterns from data. Each layer processes information to form increasingly complex representations, enabling tasks like image recognition and natural language processing without manual feature design."*, ChatGPT, OpenAI

**Image segmentation**
source: https://blogs.nvidia.com/blog/drive-labs-panoptic-segmentation/

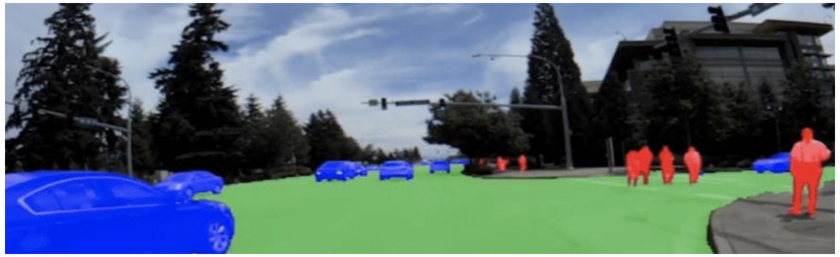


**Image reconstruction**
source: https://research.nvidia.com/publication/2017-10_learning-super-resolve-blurry-face-and-text-images

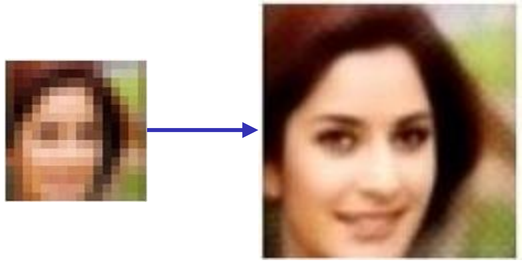


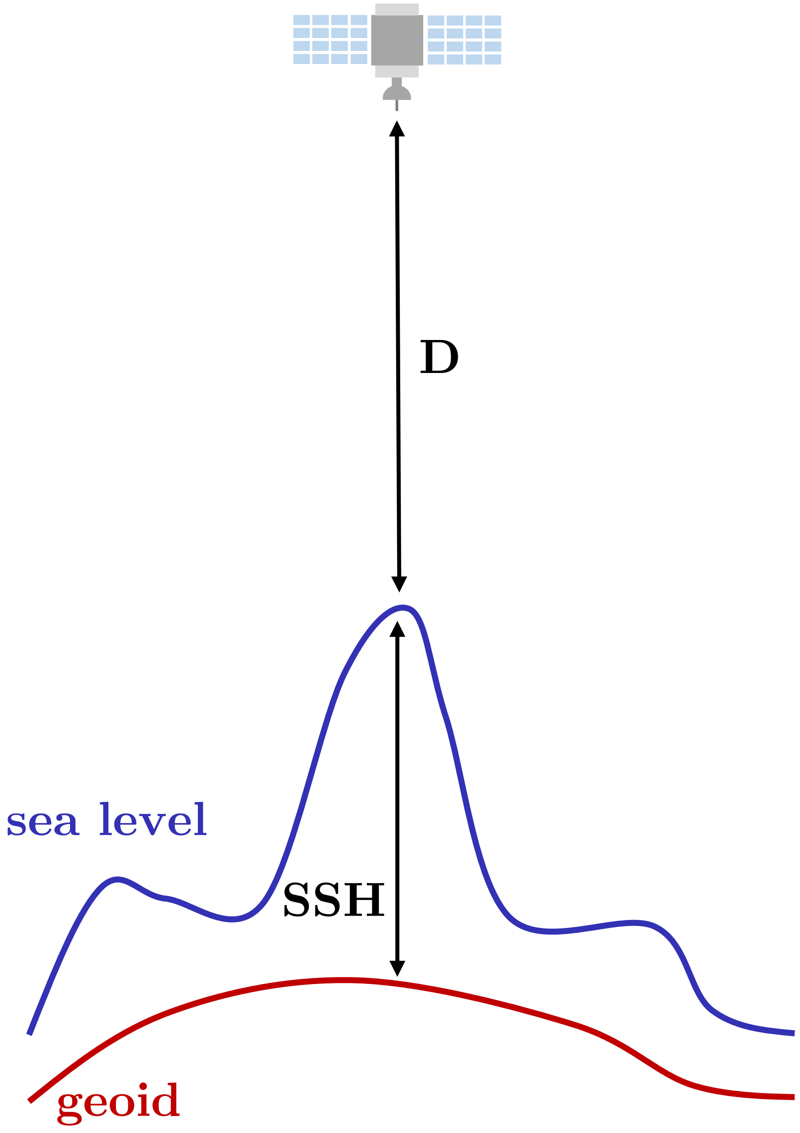
**Image generation:** *"Four pirates and a dragon on a boat"*, source: midjourney.com

# Outline

1. Introduction
2. **Satellite observations of height and temperature**
3. Reconstruction using deep neural network
4. An example of downscaling
5. An example of interpolation
6. Conclusions and perspectives

# Sea Surface Height (SSH) remote sensing

- SSH is an important variable, as it is used to estimate surface currents
- The SSH is the height of the surface above the geoid
- Measured from space by **satellite altimetry**

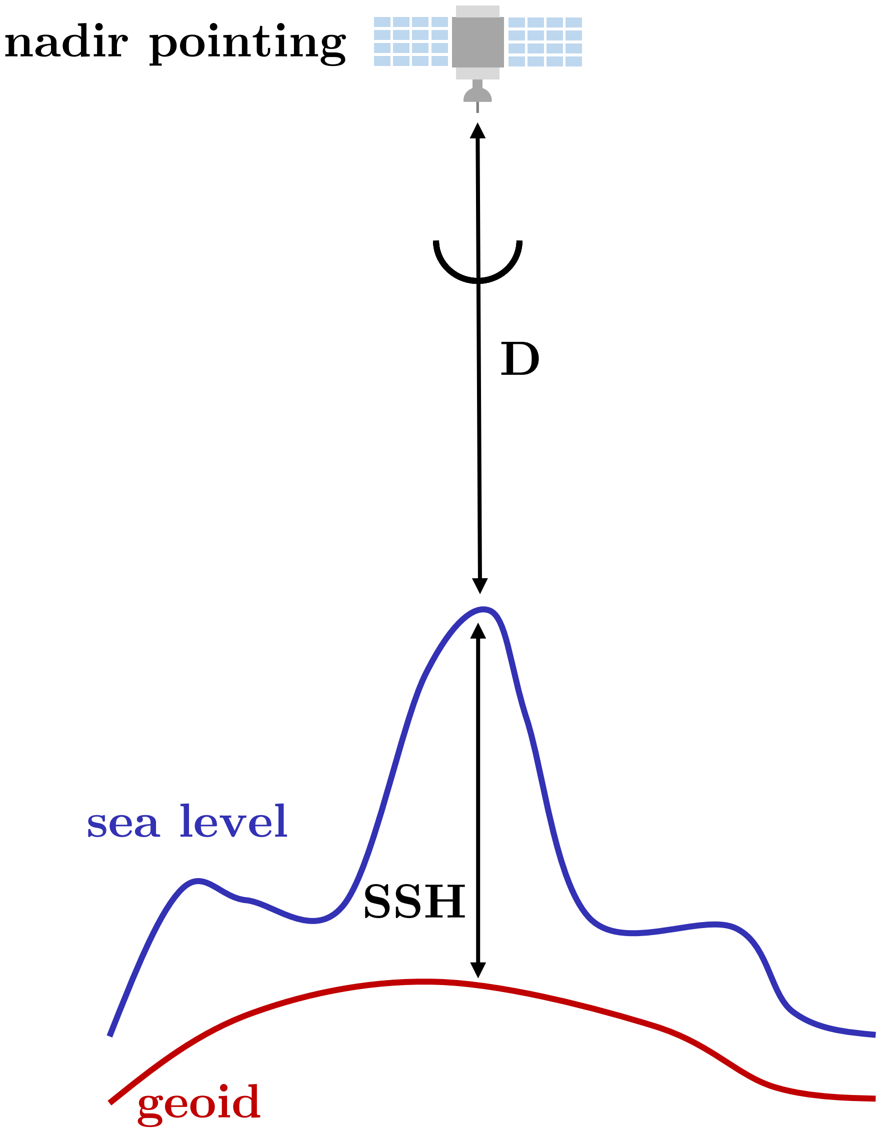


# Sea Surface Height (SSH) remote sensing

- SSH is an important variable, as it is used to estimate surface currents
- The SSH is the height of the surface above the geoid
- Measured from space by **satellite altimetry**

Two different remote sensing technology:

- Historically: nadir-pointing altimeters
  - Measure the return time of a radar pulse and deduce **D**
  - Only vertical measurements

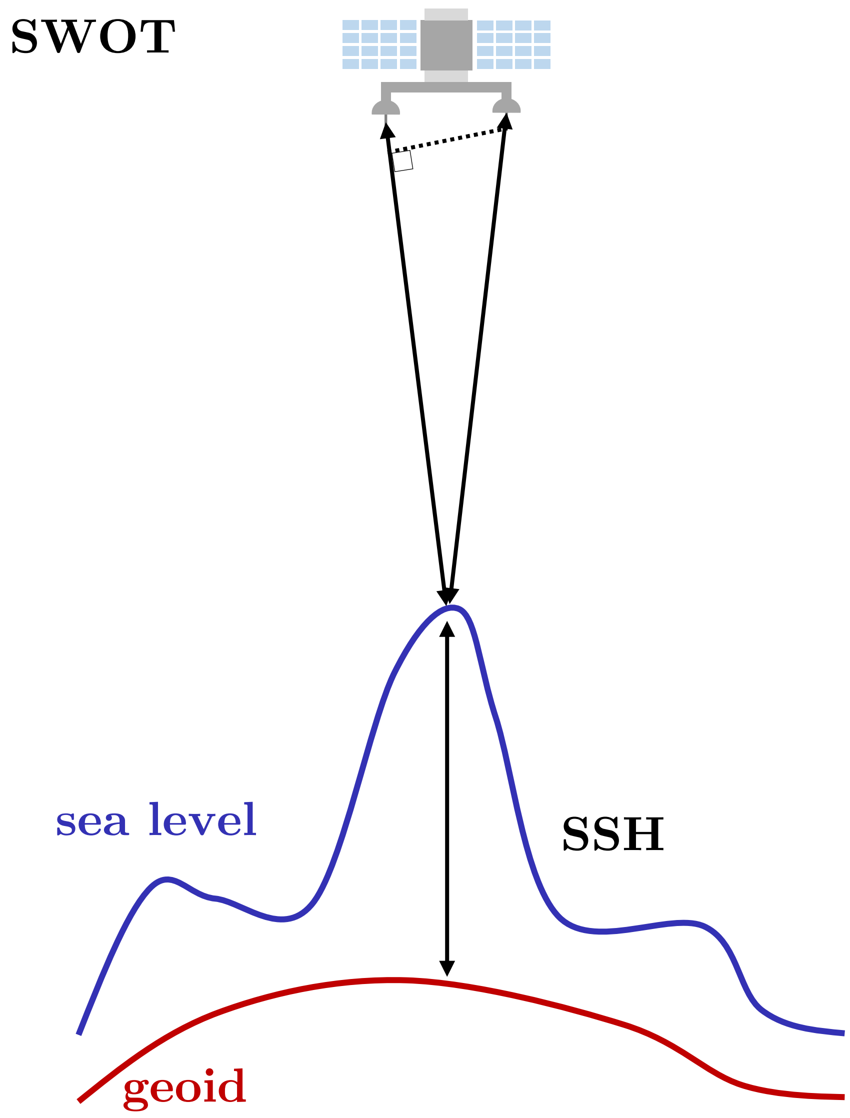


# Sea Surface Height (SSH) remote sensing

- SSH is an important variable, as it is used to estimate surface currents
- The SSH is the height of the surface above the geoid
- Measured from space by **remote sensing altimetry**

Two different remote sensing technology:

- Historically: nadir-pointing altimeters
  - Measure the return time of a radar pulse and deduce **D**
  - Only vertical measurements
- Since 2022: Interferometry altimeter (SWOT satellite)
  - Allows SSH measure at different locations
  - Wider and better resolved swath (2 swath of 60 km) with high spatial resolution (up to 250 m per pixel).

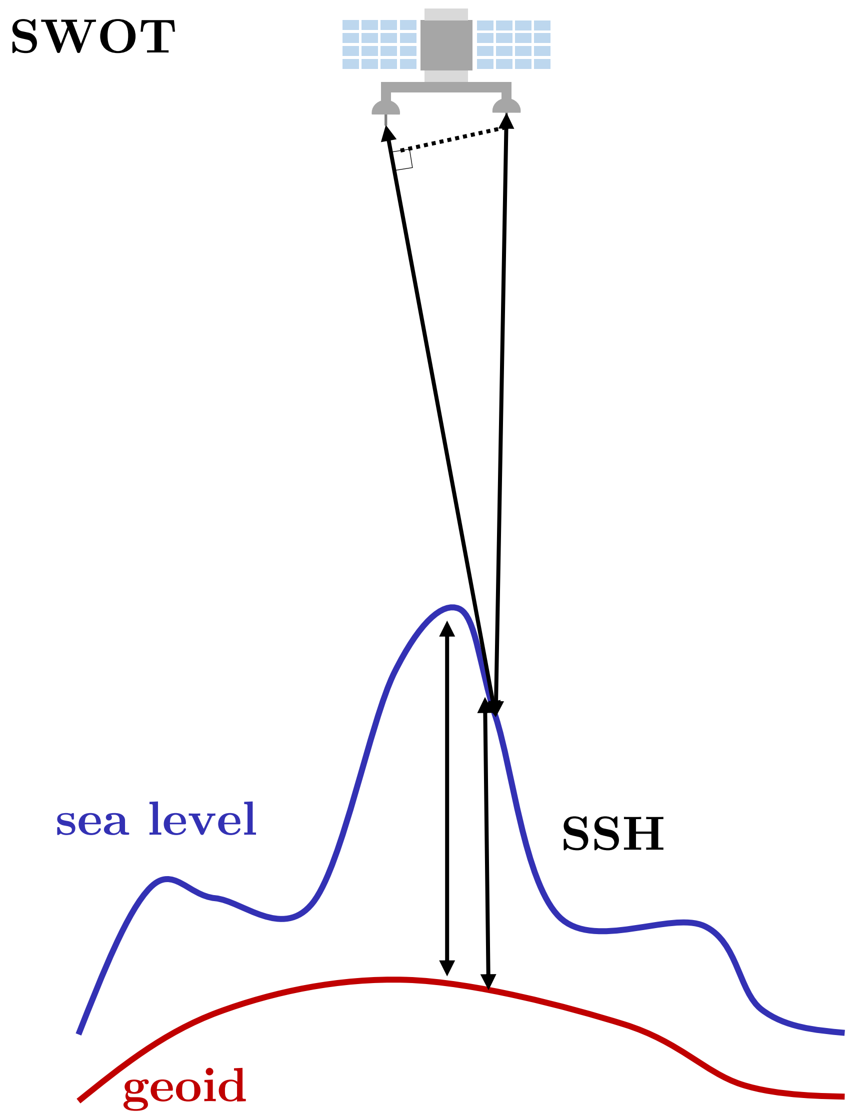


# Sea Surface Height (SSH) remote sensing

- SSH is an important variable, as it is used to estimate surface currents
- The SSH is the height of the surface above the geoid
- Measured from space by **remote sensing altimetry**

Two different remote sensing technology:

- Historically: nadir-pointing altimeters
  - Measure the return time of a radar pulse and deduce **D**
  - Only vertical measurements
- Since 2022: Interferometry altimeter (SWOT satellite)
  - Allows SSH measure at different locations
  - Wider and better resolved swath (2 swath of 60 km) with high spatial resolution (up to 250 m per pixel).

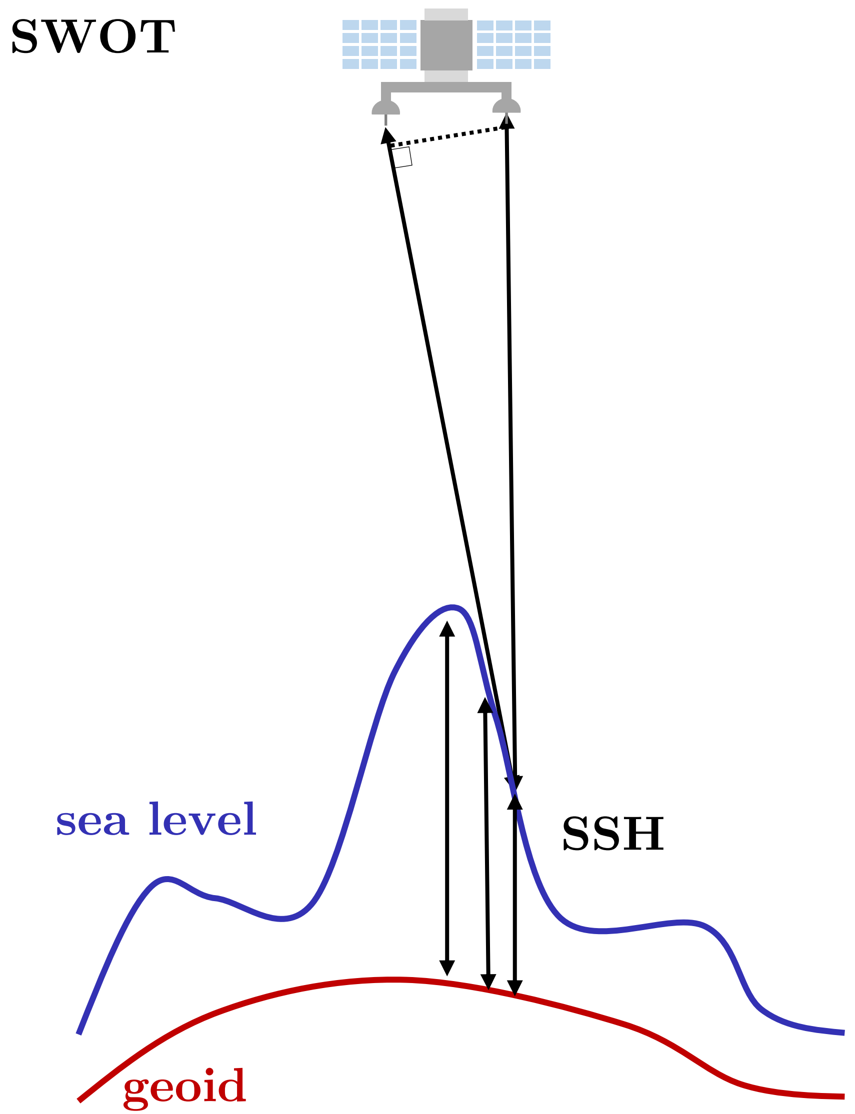


# Sea Surface Height (SSH) remote sensing

- SSH is an important variable, as it is used to estimate surface currents
- The SSH is the height of the surface above the geoid
- Measured from space by **remote sensing altimetry**

Two different remote sensing technology:

- Historically: nadir-pointing altimeters
  - Measure the return time of a radar pulse and deduce **D**
  - Only vertical measurements
- Since 2022: Interferometry altimeter (SWOT satellite)
  - Allows SSH measure at different locations
  - Wider and better resolved swath (2 swath of 60 km) with high spatial resolution (up to 250 m per pixel).

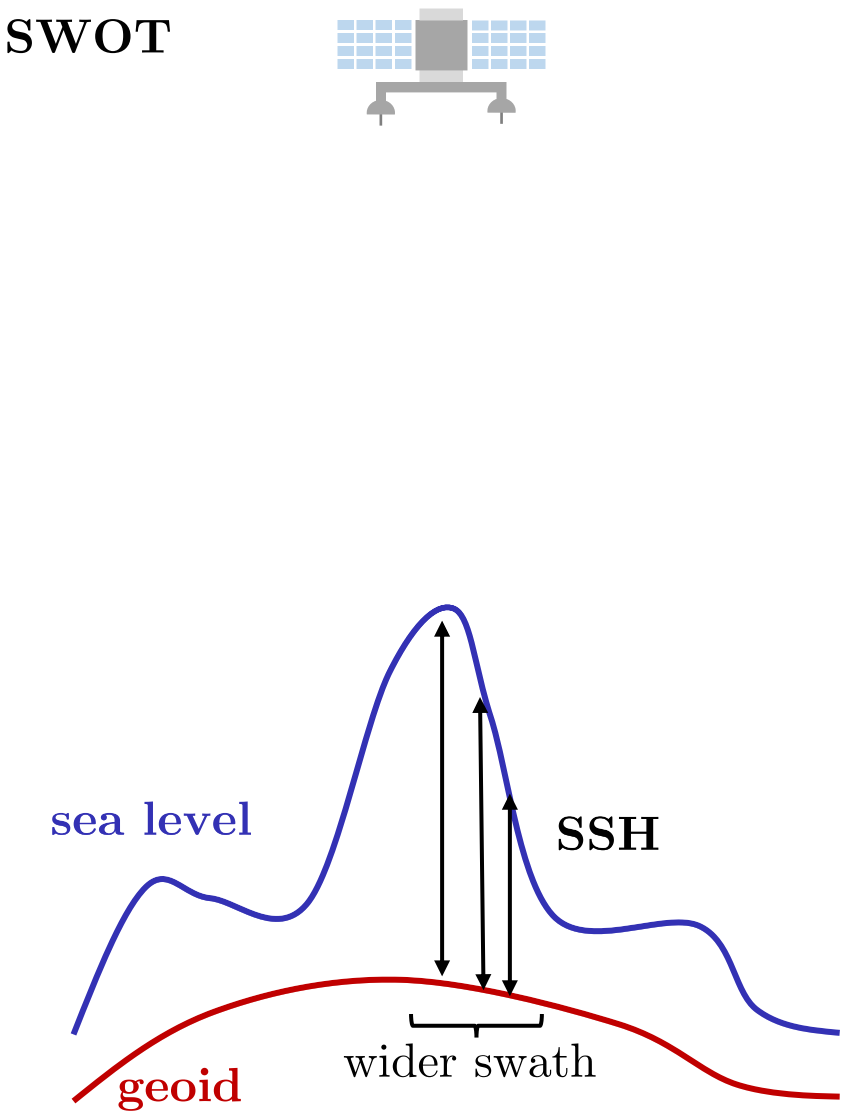


# Sea Surface Height (SSH) remote sensing

- SSH is an important variable, as it is used to estimate surface currents
- The SSH is the height of the surface above the geoid
- Measured from space by **remote sensing altimetry**

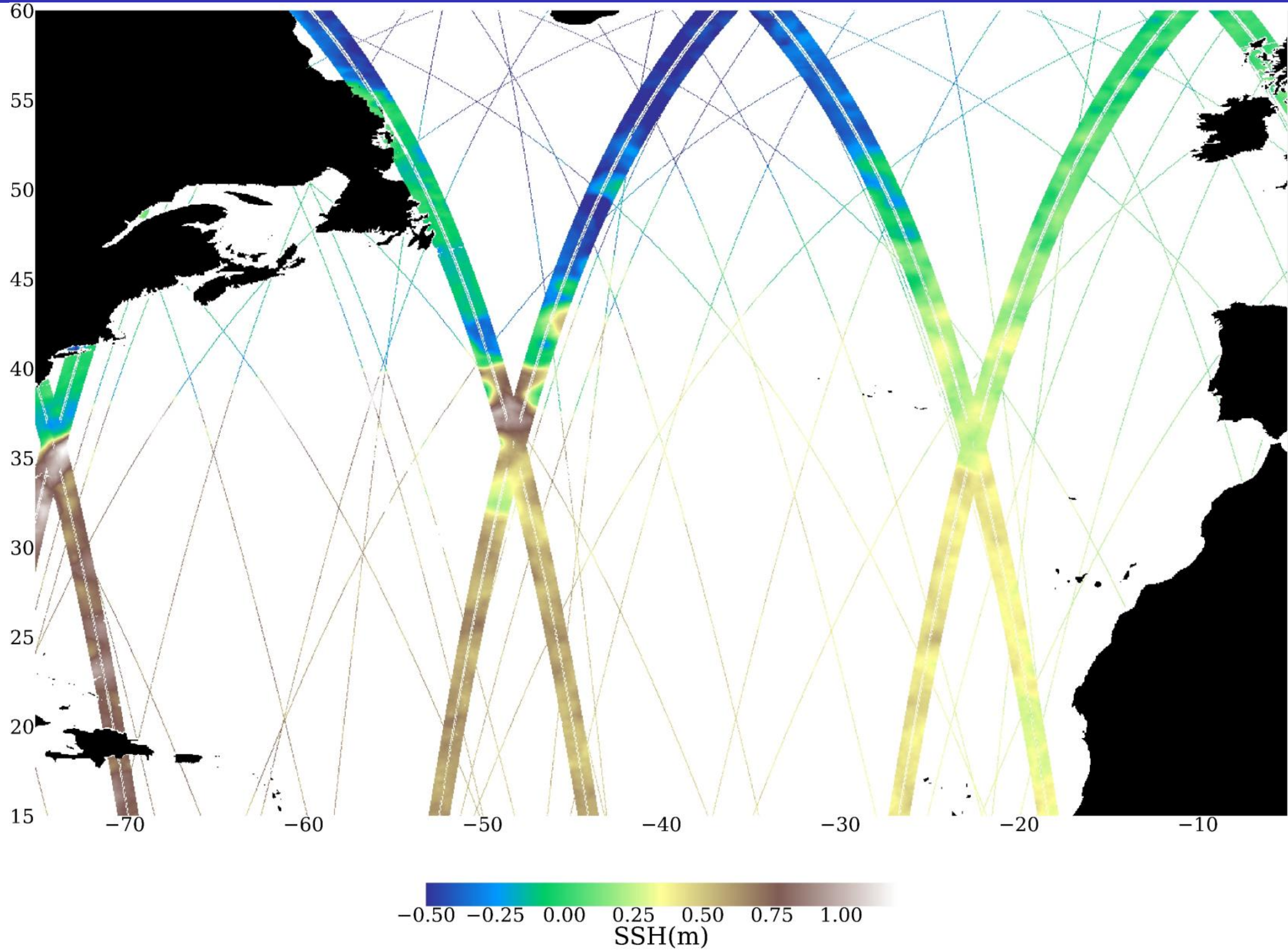
Two different remote sensing technology:

- Historically: nadir-pointing altimeters
  - Measure the return time of a radar pulse and deduce **D**
  - Only vertical measurements
- Since 2022: Interferometry altimeter (SWOT satellite)
  - Allows SSH measure at different locations
  - Wider and better resolved swath (2 swath of 60 km) with high spatial resolution (up to 250 m per pixel).



# SSH mapping

- Daily observations other the north Atlantic (2023/05/14)
    - 6 nadir-pointing altimeters (along-track data)
    - SWOT (120 km wide swath)

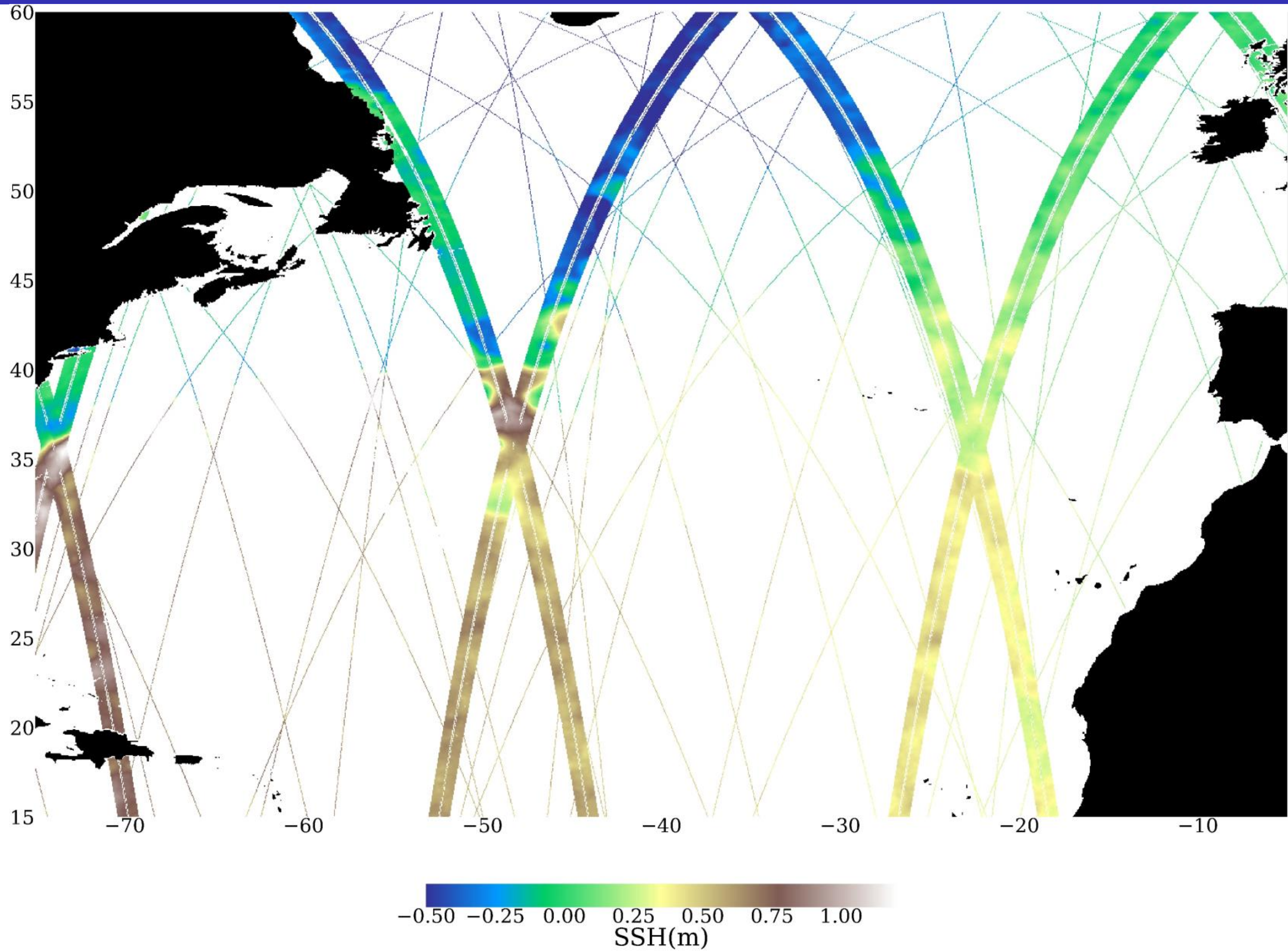


# SSH mapping

- Daily observations other the north Atlantic (2023/05/14)
  - 6 nadir-pointing altimeters (along-track data)
  - SWOT (120 km wide swath)

Even with 7 satellites significant gaps remain in the data.

- Spatio-temporal interpolation

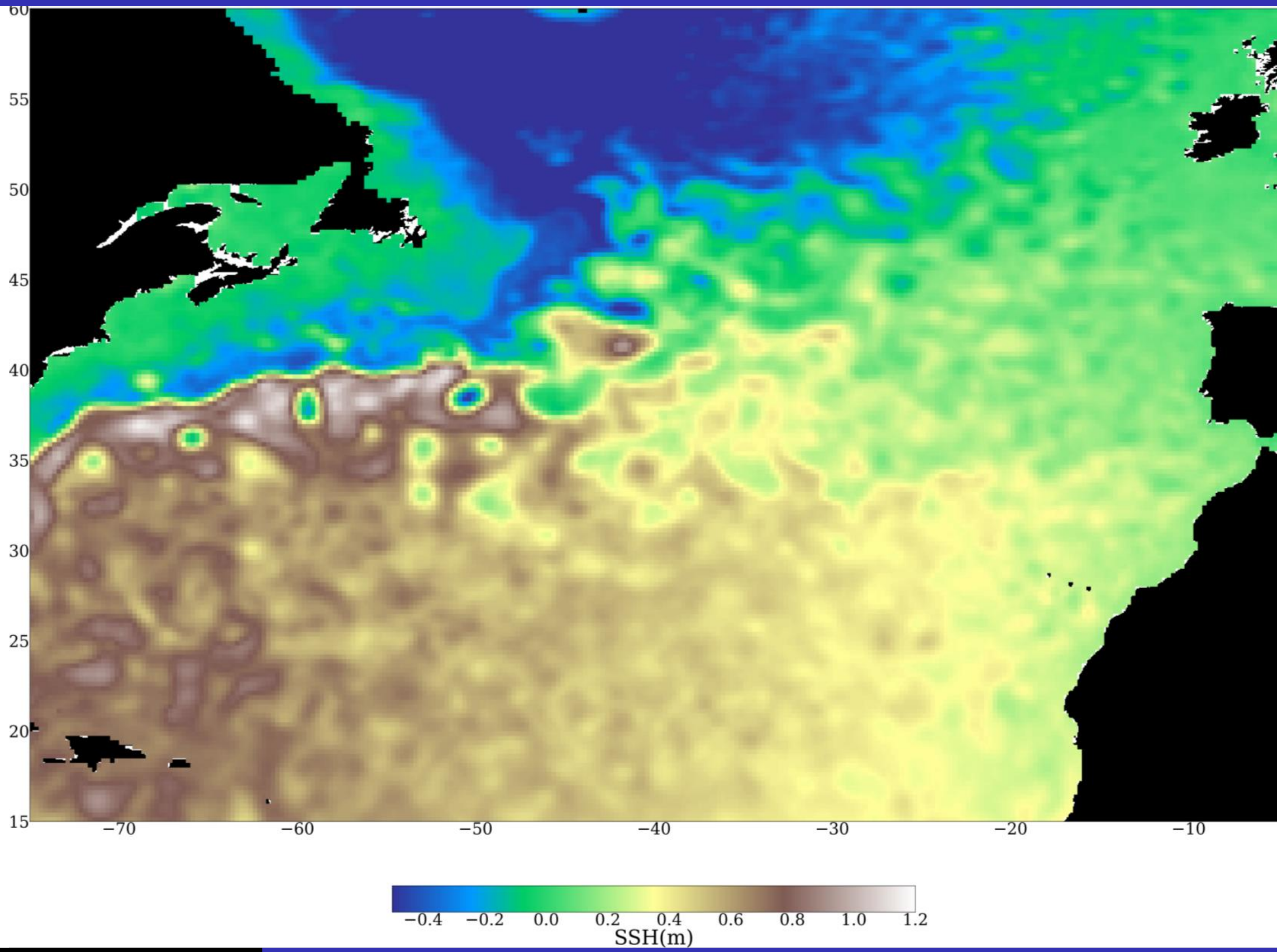


# SSH mapping

- Daily observations other the north Atlantic (2023/05/14)
  - 6 nadir-pointing altimeters (along-track data)
  - SWOT (120 km wide swath)

Even with 7 satellites significant gaps remain in the data.

- Spatio-temporal interpolation
  - DUACS : linear Optimal Interpolation (OI) **[Taburet et al., 2019]**
  - Global operational product
  - The effective resolution of DUACS is low
  - It misses small structures, i.e. $<$ 100 km [**Amores et al., 2018, Stegner et al., 2021**]



# SSH mapping

- Daily observations other the north Atlantic (2023/05/14)
  - 6 nadir-pointing altimeters (along-track data)
  - SWOT (120 km wide swath)

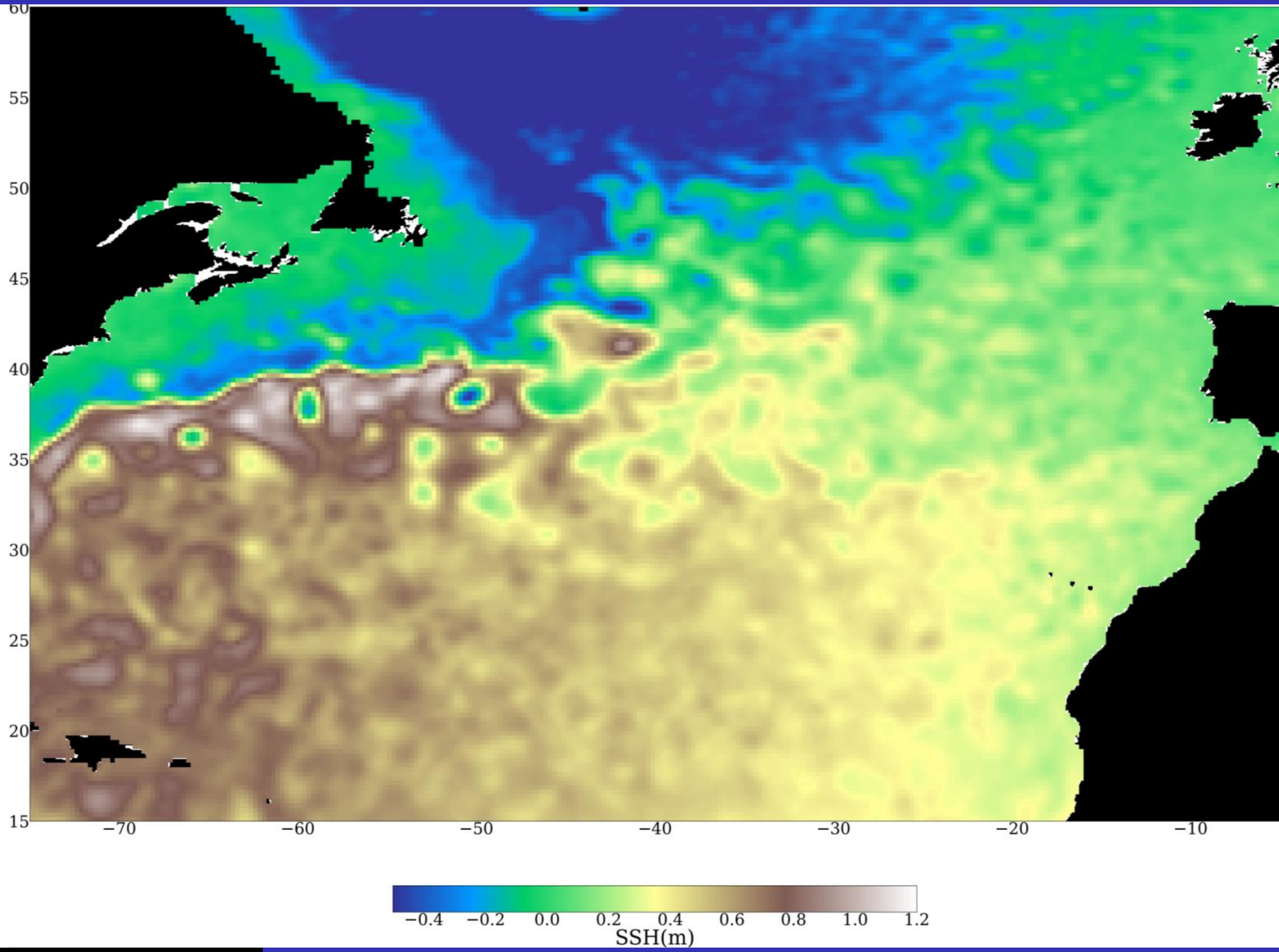
Even with 7 satellites significant gaps remain in the data.

- Spatio-temporal interpolation
  - DUACS : linear Optimal Interpolation (OI) **[Taburet et al., 2019]**
  - Global operational product
  - The effective resolution of DUACS is low
  - It misses small structures, i.e. $<$ 100 km [**Amores et al., 2018, Stegner et al., 2021**]

SSH reconstruction is an important challenge

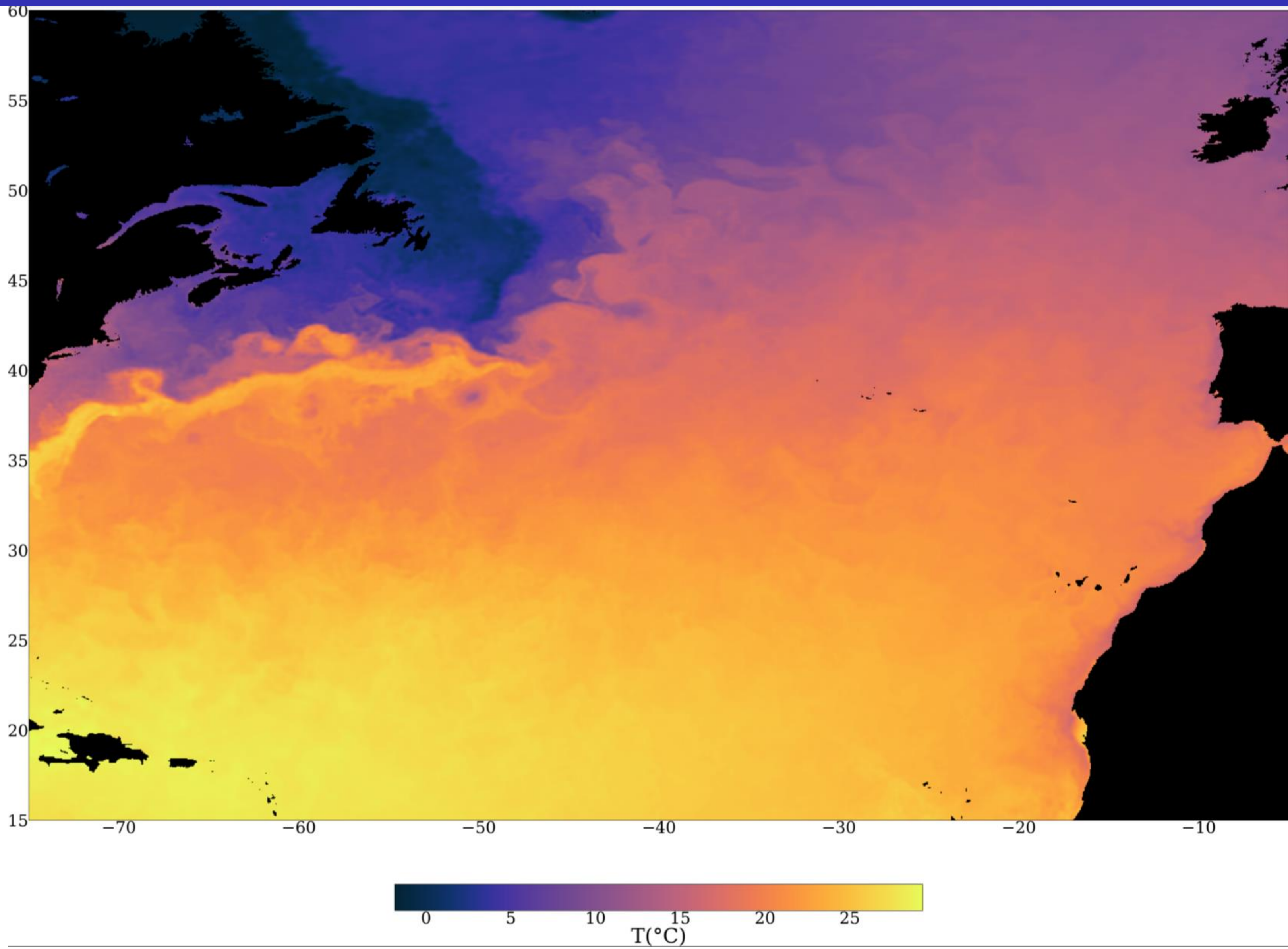
OCEAN DATA CHALLENGES

https://ocean-data-challenges.github.io/



# Sea Surface Temperature (SST) remote sensing

- Sea Surface Temperature (SST) is measured by direct imaging.

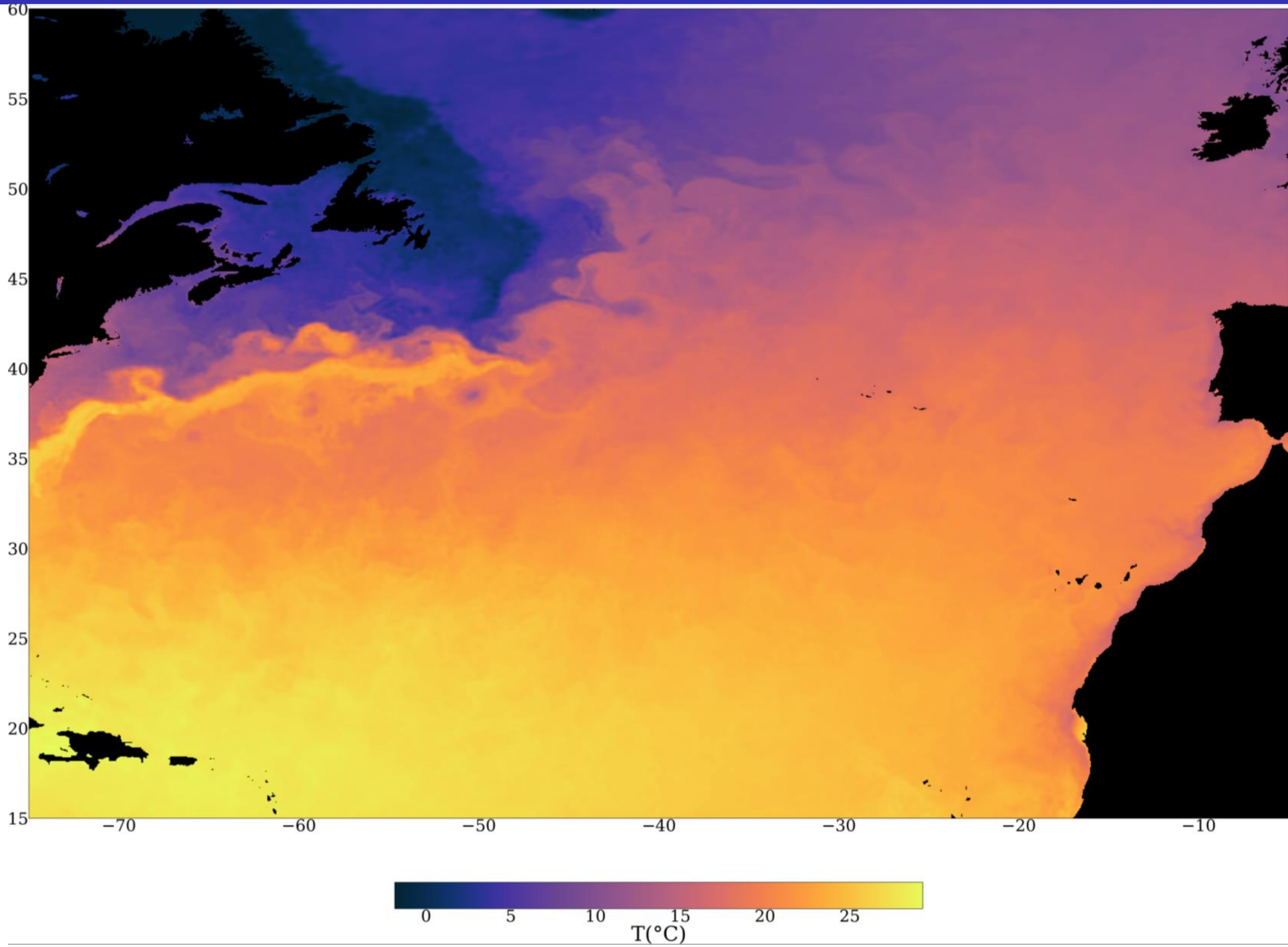


# Sea Surface Temperature (SST) remote sensing

- Sea Surface Temperature (SST) is measured by direct imaging.

Two type of sensors:

- Infrared sensors:
  - High-resolution (1 to 4 km per pixels)
  - Can not see through clouds
- Microwave sensors:
  - Lower resolution (25 km per pixel)
  - See though clouds

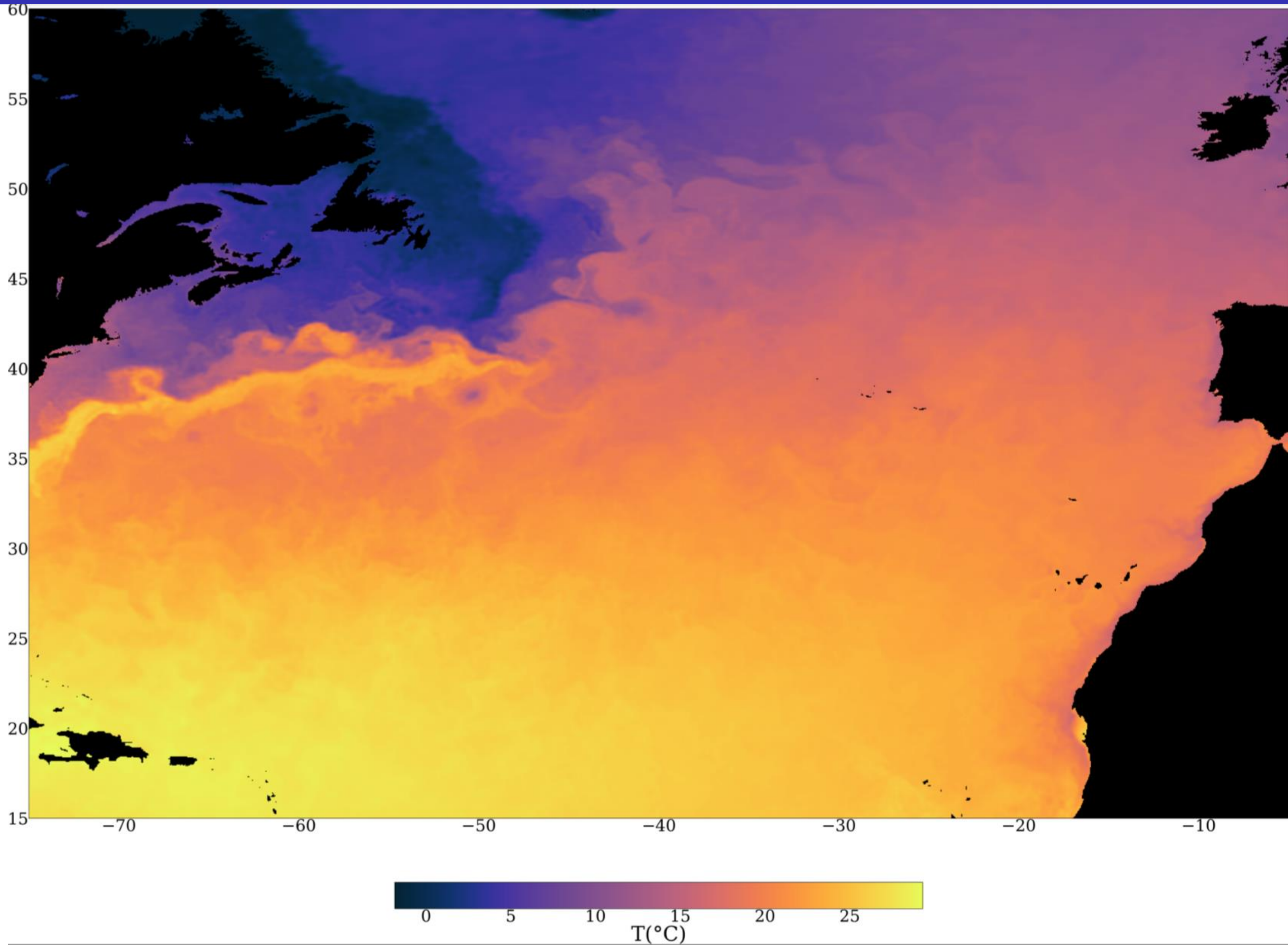


# Sea Surface Temperature (SST) remote sensing

- Sea Surface Temperature (SST) is measured by direct imaging.

Two type of sensors:

- Infrared sensors:
    - High-resolution (1 to 4 km per pixels)
    - Can not see through clouds
- Microwave sensors:
    - Lower resolution (25 km per pixel)
    - See though clouds
- Complete maps are also obtained by OI
    - Instrumental noise and blurring from the OI process

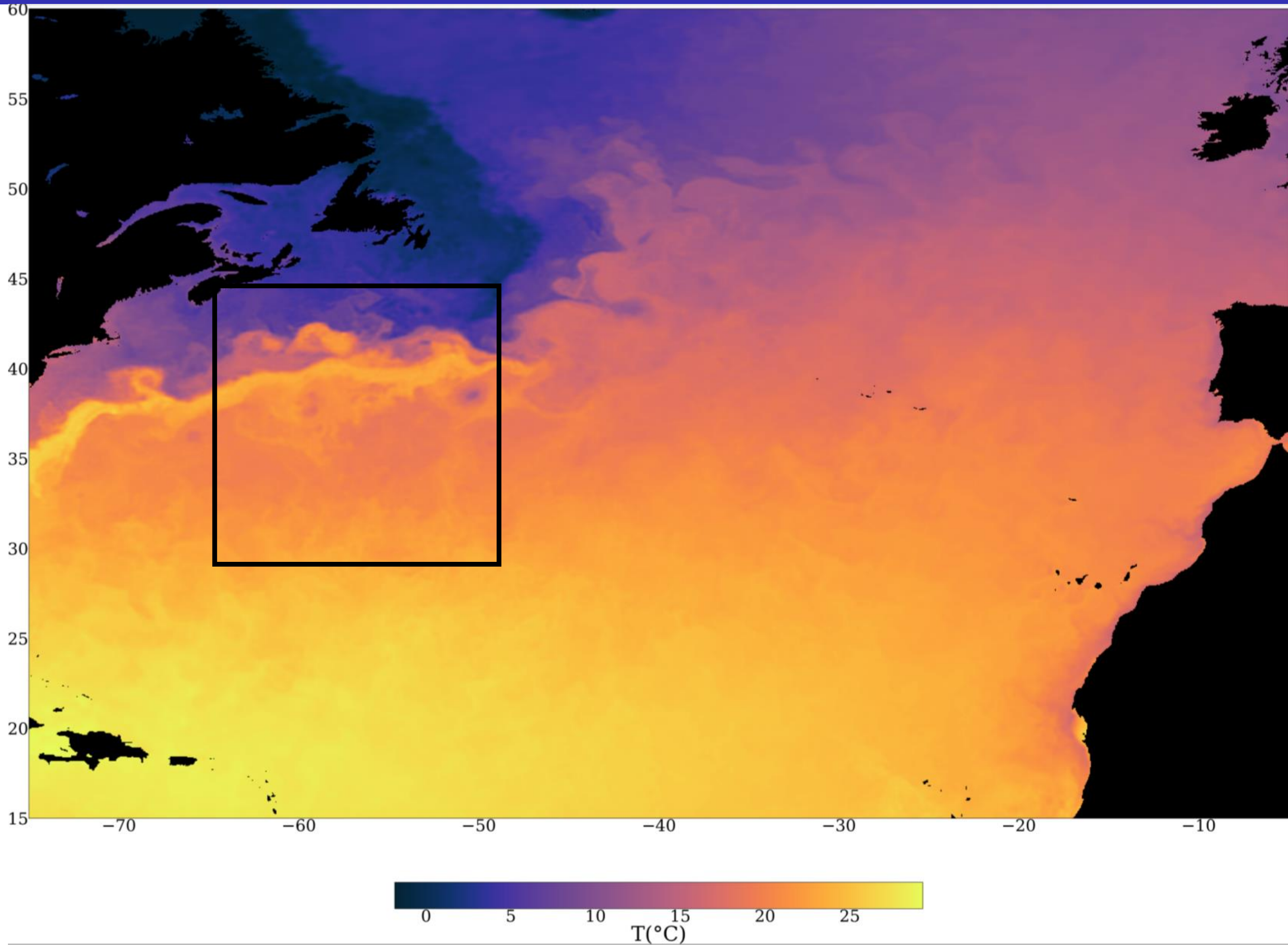


# Sea Surface Temperature (SST) remote sensing

- Sea Surface Temperature (SST) is measured by direct imaging.

Two type of sensors:

- Infrared sensors:
  - High-resolution (1 to 4 km per pixels)
  - Can not see through clouds
- Microwave sensors:
  - Lower resolution (25 km per pixel)
  - See though clouds
- Complete maps are also obtained by OI
  - Instrumental noise and blurring from the OI process

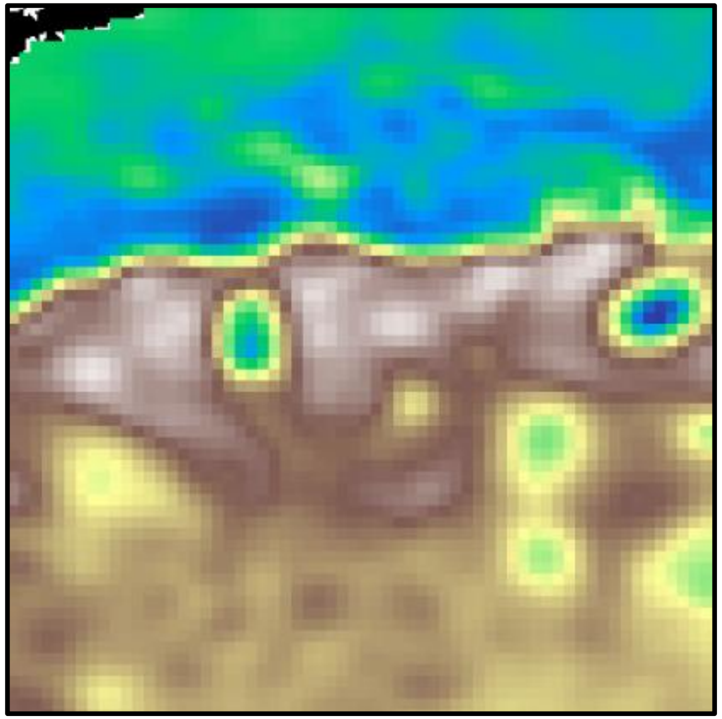


# Comparison between SSH and SST images

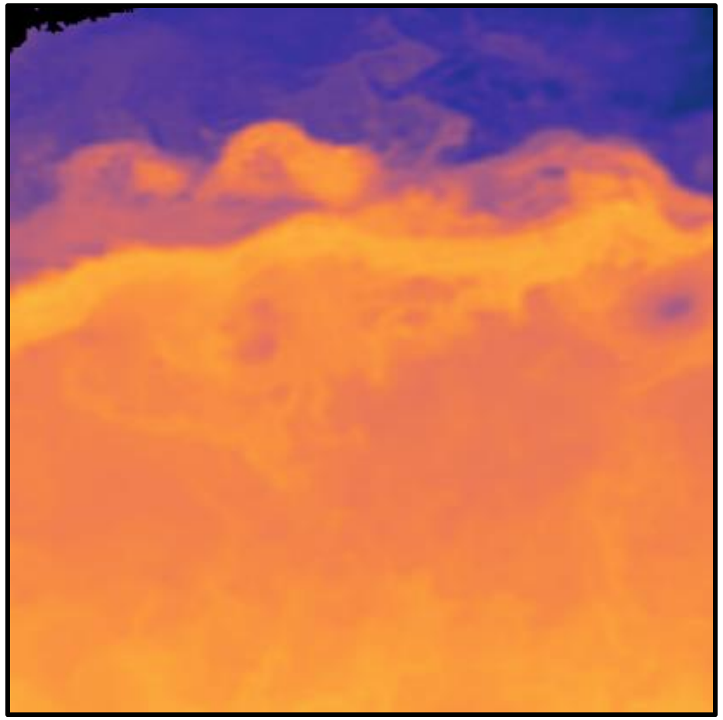
Zoom at the Gulf Stream:

- Some structures are present in the SSH and SST
- The SST image present more high-frequency details

**DUACS**



**L4 SST**



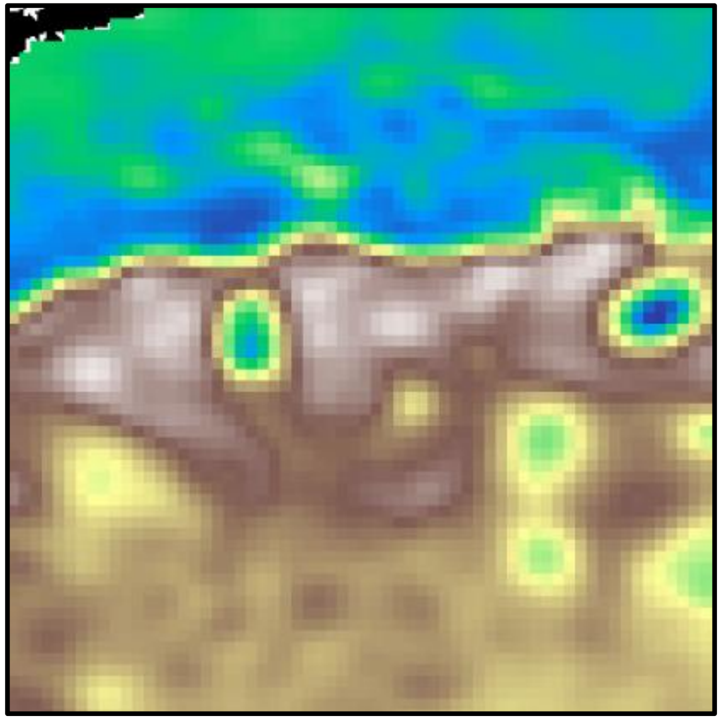
# Comparison between SSH and SST images

Zoom at the Gulf Stream:

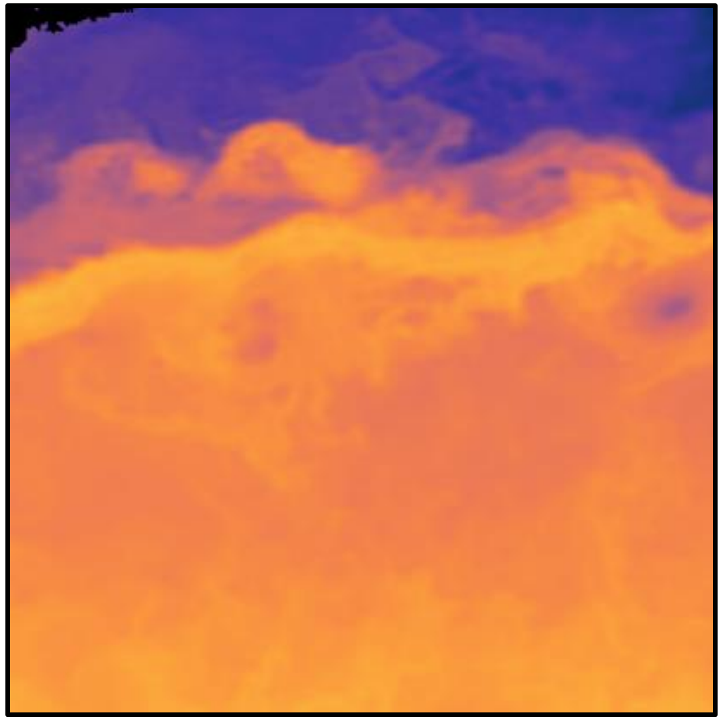
- Some structures are present in the SSH and SST
- The SST image present more high-frequency details

➤ Improve **SSH** mapping using **SST** information: **multi-variate reconstruction**

**DUACS**

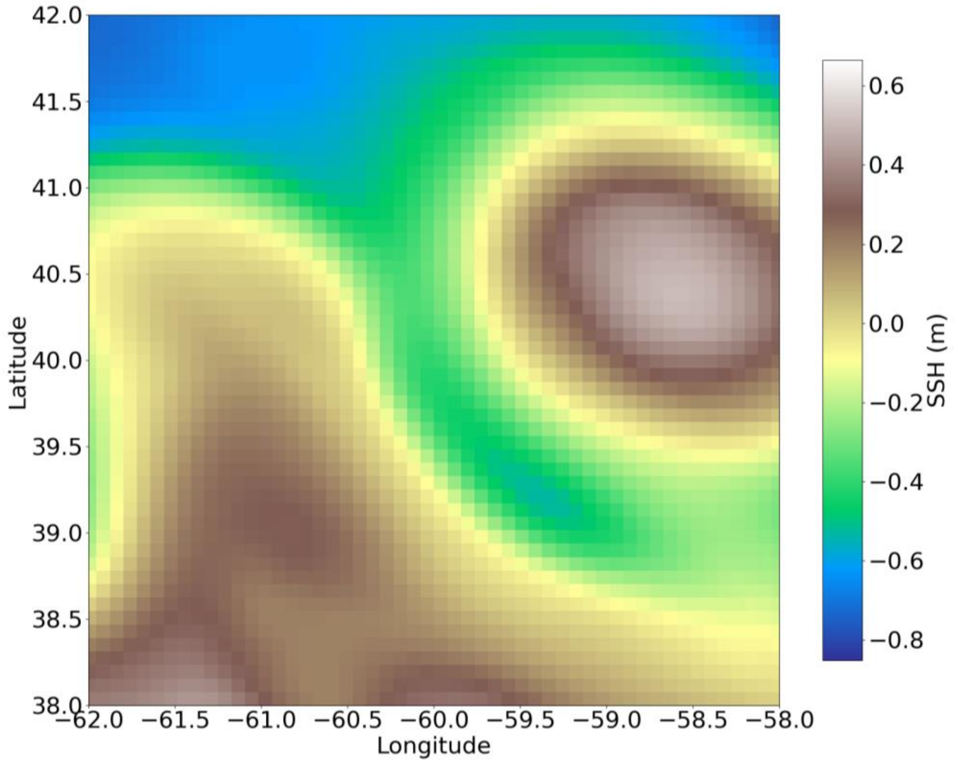


**L4 SST**



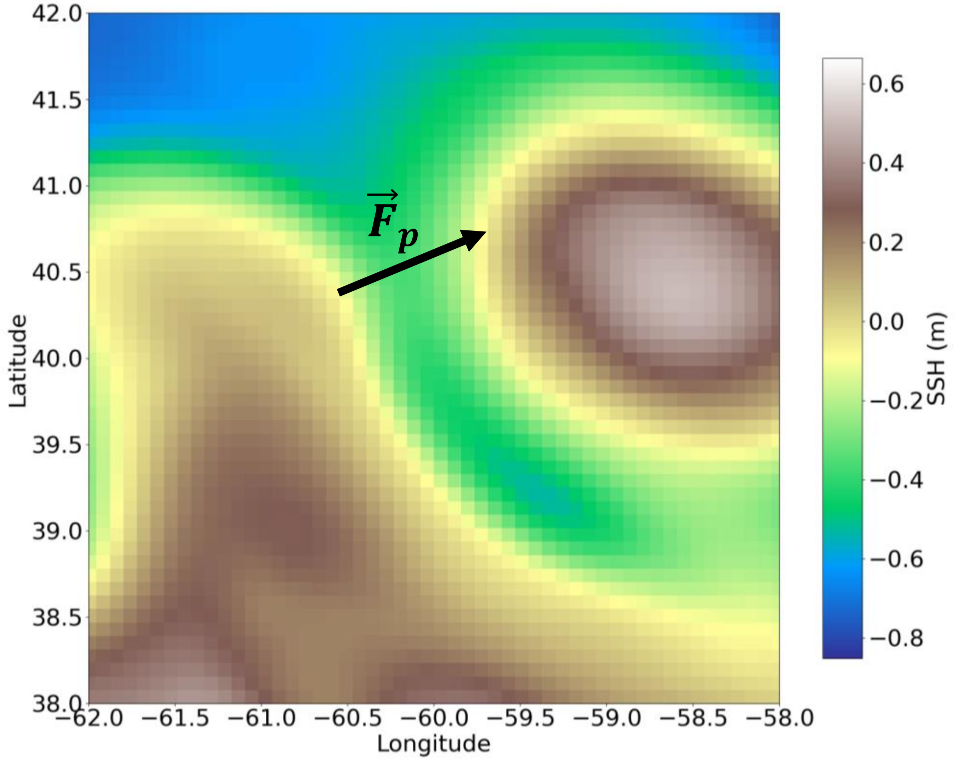
# Physical relationship between SSH and SST

- SSH and SST are linked to circulation
- Geostrophic approximation: static equilibrium between pressure and Coriolis forces



# Physical relationship between SSH and SST

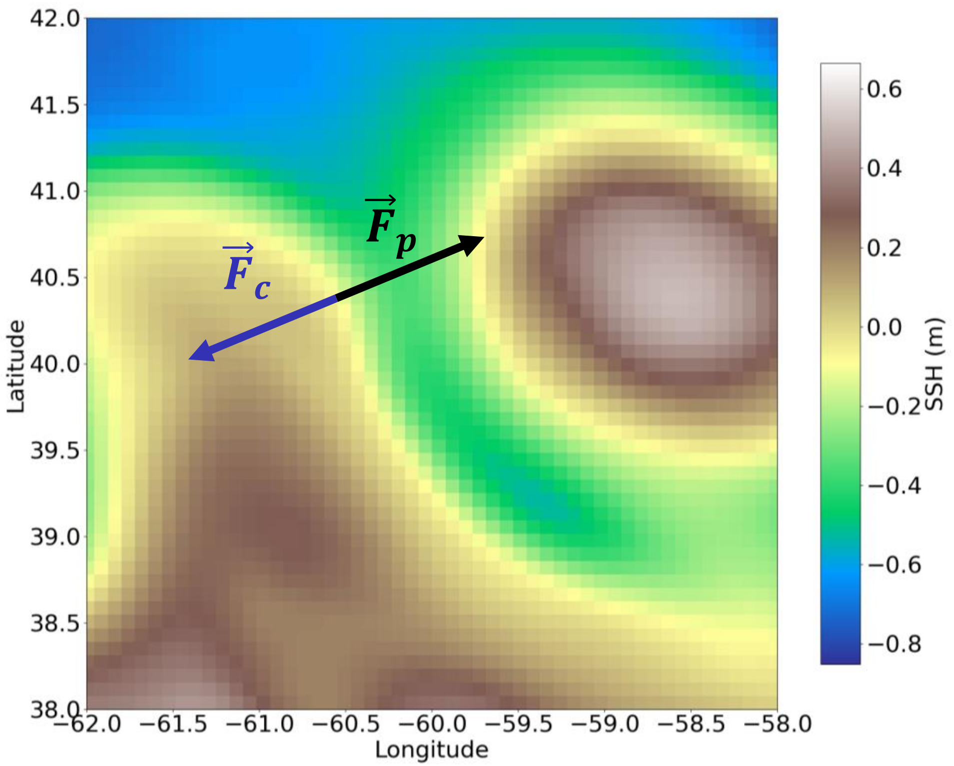
- SSH and SST are linked to circulation
- Geostrophic approximation: static equilibrium between pressure and Coriolis forces




← Pressure force

# Physical relationship between SSH and SST

- SSH and SST are linked to circulation
- Geostrophic approximation: static equilibrium between pressure and Coriolis forces

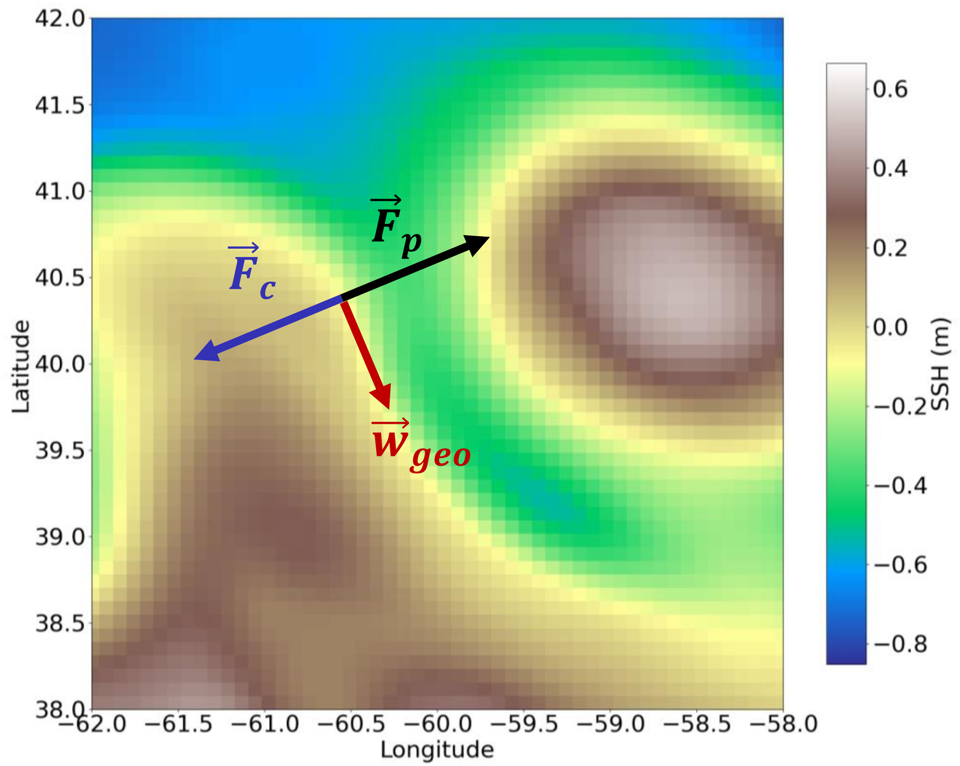


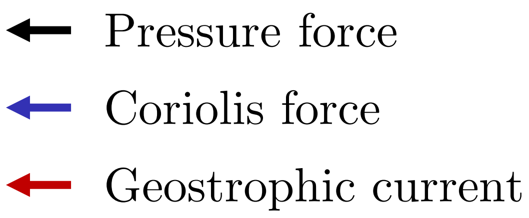

Pressure force

Coriolis force

# Physical relationship between SSH and SST

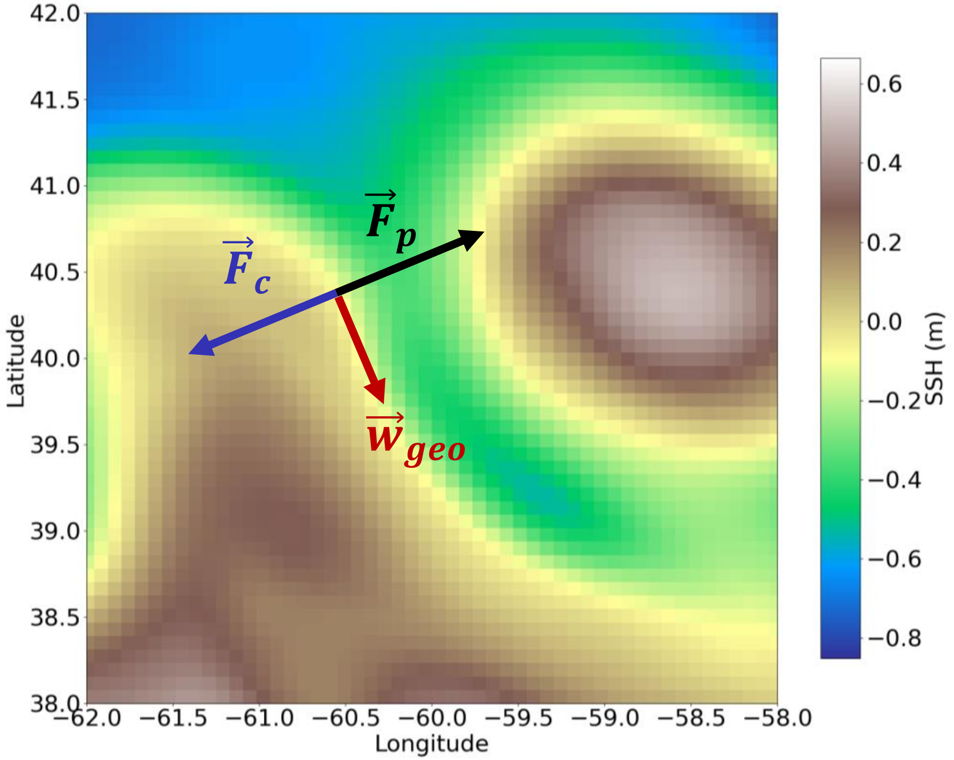
- SSH and SST are linked to circulation
- Geostrophic approximation: static equilibrium between pressure and Coriolis forces



# Physical relationship between SSH and SST

- SSH and SST are linked to circulation
- Geostrophic approximation: static equilibrium between pressure and Coriolis forces




Pressure force

Coriolis force

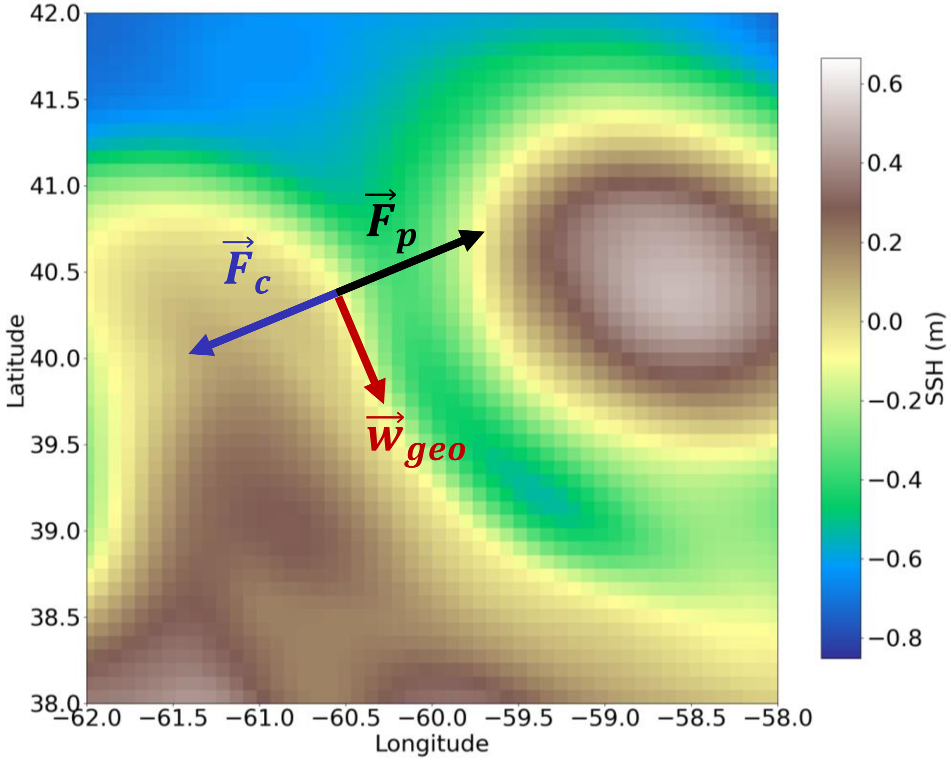
Geostrophic current

$$\vec{F}_c = \vec{F}_p$$

$$\Leftrightarrow \vec{w}_{geo} = \begin{pmatrix} u_g \\ v_g \end{pmatrix} = \frac{g}{f_0} \begin{pmatrix} -\frac{\partial h}{\partial y} \\ \frac{\partial h}{\partial x} \end{pmatrix}$$

# Physical relationship between SSH and SST

- SSH and SST are linked to circulation
- Geostrophic approximation: static equilibrium between pressure and Coriolis forces




Pressure force

Coriolis force

Geostrophic current

$$\vec{F}_c = \vec{F}_p$$

$$\Leftrightarrow \vec{w}_{geo} = \begin{pmatrix} u_g \\ v_g \end{pmatrix} = \frac{g}{f_0} \begin{pmatrix} -\frac{\partial h}{\partial y} \\ \frac{\partial h}{\partial x} \end{pmatrix}$$

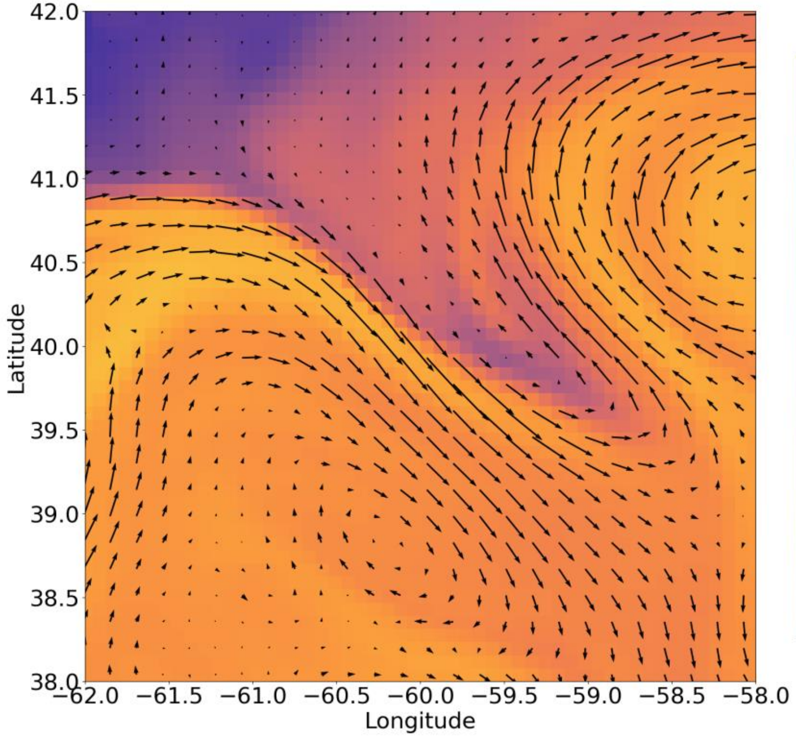
- Valid approximation of the surface currents far from the equator

# Physical relationship between SSH and SST

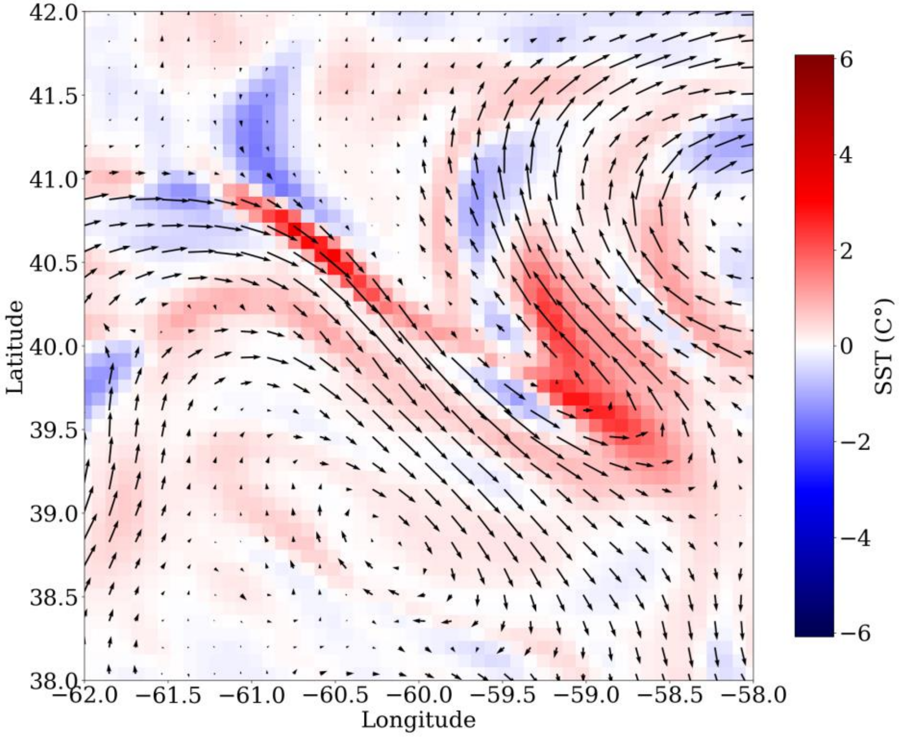
- Temperature is a tracer of the currents
  - Heat is transported by the flow in an advection dynamic

$$\frac{\partial T}{\partial t} + \vec{w} \cdot \nabla T = 0$$

SST and currents




Temporal evolution of SST



# Summary

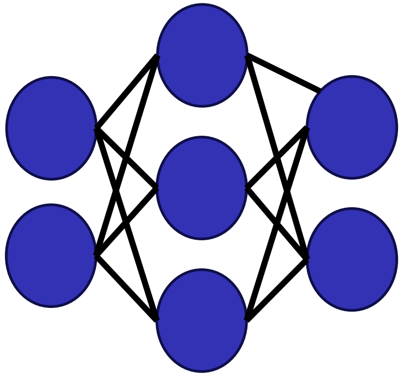
- SSH and SST are physically related to ocean surface currents
- SST images present more high frequency information than SSH

# Summary

- SSH and SST are physically related to ocean surface currents
- SST images present more high frequency information than SSH

➢ How to use SST data to improve the SSH?

➢ How to use Neural Networks to reconstruct SSH?



# Outline

# Image inverse problems

Reconstructing SSH can be modeled as an image inverse problem. Let $\mathbf{X}^{\text{ssh}}$ be the state and $\mathbf{Y}^{\text{ssh}}$ the observations:

$$\mathbf{Y}^{\text{ssh}} = \mathcal{F}\left(\mathbf{X}^{\text{ssh}}\right) + \varepsilon$$

forward problem

$$\widehat{\mathbf{X}}^{\text{ssh}} = f\left(\mathbf{Y}^{\text{ssh}}\right)$$

inverse problem

# Image inverse problems

Reconstructing SSH can be modeled as an image inverse problem. Let $\mathbf{X}^{\text{ssh}}$ be the state and $\mathbf{Y}^{\text{ssh}}$ the observations:

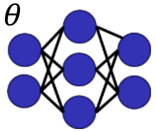
$$\mathbf{Y}^{\text{ssh}} = \mathcal{F}\left(\mathbf{X}^{\text{ssh}}\right) + \varepsilon$$

forward problem

$$\widehat{\mathbf{X}}^{\text{ssh}} = f\left(\mathbf{Y}^{\text{ssh}}\right)$$

inverse problem

In our case the inverse operator is a neural network $f_\theta$



# Image inverse problems

Reconstructing SSH can be modeled as an image inverse problem. Let $\mathbf{X}^{\text{ssh}}$ be the state and $\mathbf{Y}^{\text{ssh}}$ the observations:

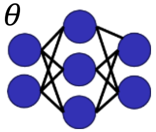
$$\mathbf{Y}^{\text{ssh}} = \mathcal{F}\left(\mathbf{X}^{\text{ssh}}\right) + \varepsilon$$

forward problem

$$\widehat{\mathbf{X}}^{\text{ssh}} = f\left(\mathbf{Y}^{\text{ssh}}\right)$$

inverse problem

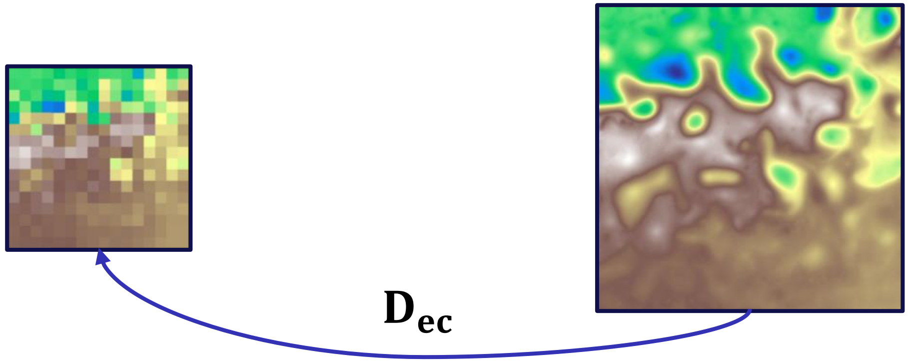
In our case the inverse operator is a neural network $f_\theta$



Two inverse problems:

**Downscaling (or Super-Resolution)**

- **Goal**: Improve the resolution of an LR image
- $\boldsymbol{\mathcal{F}}$ : a decimation operator $\mathbf{D_{ec}}$ reducing the spatial resolution. An average pool for example.



# Image inverse problems

Reconstructing SSH can be modeled as an image inverse problem. Let $\mathbf{X}^{\mathrm{ssh}}$ be the state and $\mathbf{Y}^{\mathrm{ssh}}$ the observations:

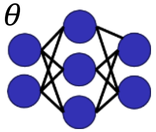
$$\mathbf{Y}^{\mathrm{ssh}} = \mathcal{F}\left(\mathbf{X}^{\mathrm{ssh}}\right) + \varepsilon$$

forward problem

$$\widehat{\mathbf{X}}^{\mathrm{ssh}} = f\left(\mathbf{Y}^{\mathrm{ssh}}\right)$$

inverse problem

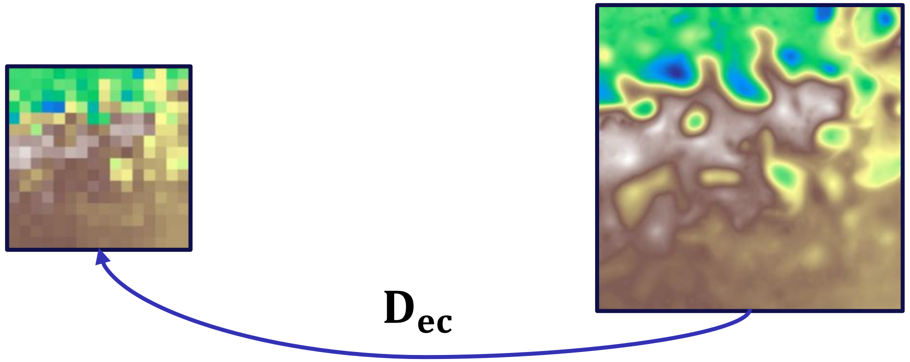
In our case the inverse operator is a neural network $f_\theta$



Two inverse problems:

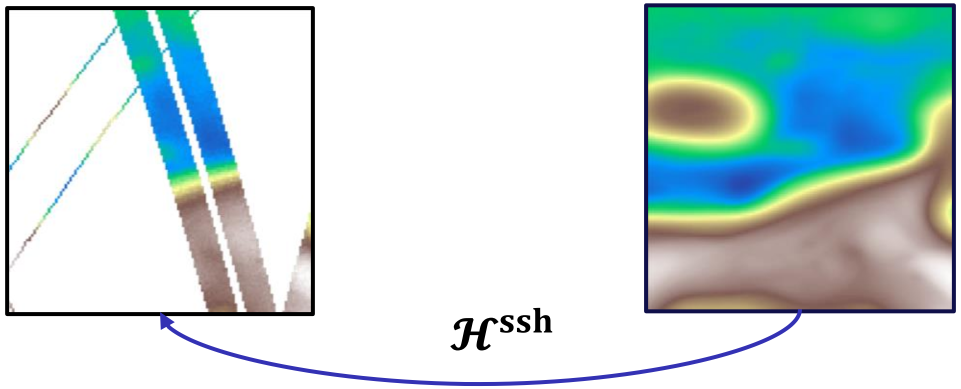
## Downscaling (or Super-Resolution)

- **Goal**: Improve the resolution of an LR image
- $\boldsymbol{\mathcal{F}}$ : a decimation operator $\mathbf{D_{ec}}$ reducing the spatial resolution. An average pool for example.




## Interpolation (or inpainting)

- **Goal**: Estimate the value of unobserved pixels
- $\boldsymbol{\mathcal{F}}$ : an observation operator $\boldsymbol{\mathcal{H}}^{\mathrm{ssh}}$ masking unobserved areas.



# Image inverse problems

Reconstructing SSH can be modeled as an image inverse problem. Let $\mathbf{X}^{\text{ssh}}$ be the state and $\mathbf{Y}^{\text{ssh}}$ the observations:

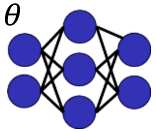
$$\mathbf{Y}^{\text{ssh}} = \mathcal{F}\left(\mathbf{X}^{\text{ssh}}\right) + \varepsilon$$

forward problem

$$\widehat{\mathbf{X}}^{\text{ssh}} = f\left(\mathbf{Y}^{\text{ssh}}\right)$$

inverse problem

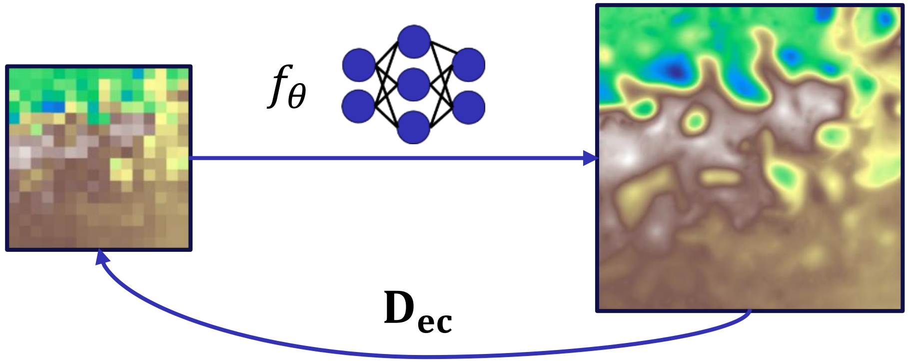
In our case the inverse operator is a neural network $f_\theta$



Two inverse problems:

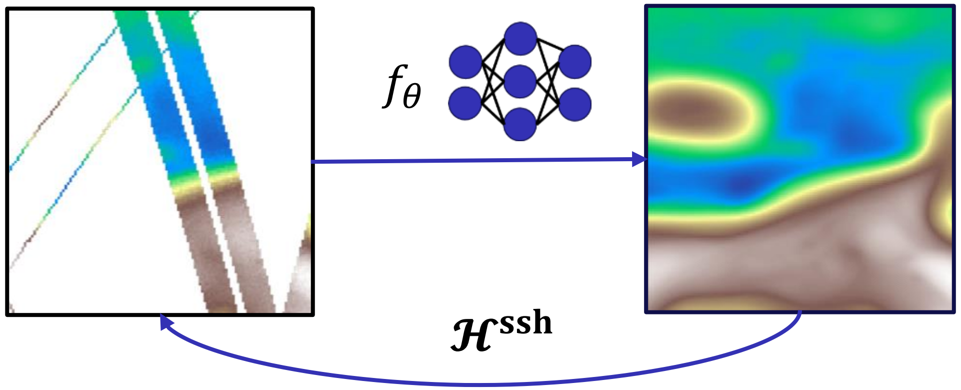
## Downscaling (or Super-Resolution)

- **Goal**: Improve the resolution of an LR image
- $\boldsymbol{\mathcal{F}}$ : a decimation operator $\mathbf{D_{ec}}$ reducing the spatial resolution. An average pool for example.




## Interpolation (or inpainting)

- **Goal**: Estimate the value of unobserved pixels
- $\boldsymbol{\mathcal{F}}$ : an observation operator $\boldsymbol{\mathcal{H}}^{\text{ssh}}$ masking unobserved areas.



# Deep Neural Network training

The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

In geosciences the **ground truth $\mathbf{X}^{\text{ssh}}$ is not available**.

Two options:

- Simulation-based training
- Observations-based training

# Deep Neural Network training

The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

In geosciences the **ground truth** $\mathbf{X}^{\text{ssh}}$ **is not available**.

**Simulation-based training**

1. The ground truth $\mathbf{X}^{\text{ssh}}$ is taken from a numerical model

# Deep Neural Network training

The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

In geosciences the **ground truth** $\mathbf{X}^{\text{ssh}}$ **is not available**.

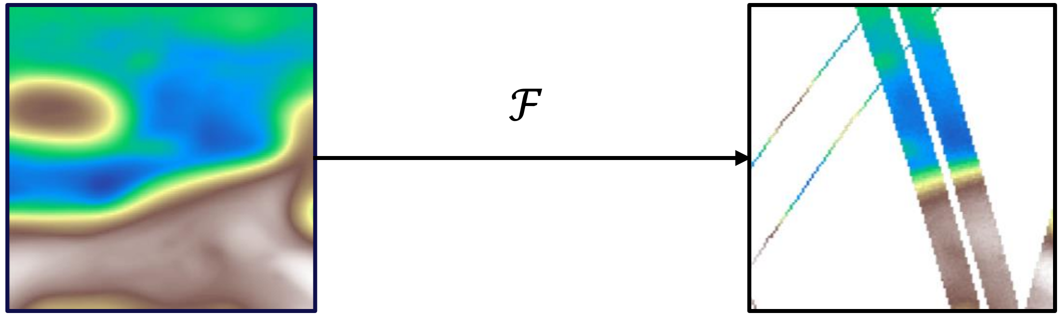
**Simulation-based training**

1. The ground truth $\mathbf{X}^{\text{ssh}}$ is taken from a numerical model
2. Emulate the forward operator $\mathcal{F}$ to generate observations : $\mathbf{Y}^{\text{ssh}} = \mathcal{F}\left(\mathbf{X}^{\text{ssh}}\right) + \varepsilon$

**OSSE**

**1.** $\mathbf{X}^{\text{ssh}}$ from a simulation

**2.** Model $\mathcal{F}$ and retrieve $\mathbf{Y}^{\text{ssh}}$




$\mathbf{D}$

# Deep Neural Network training

The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

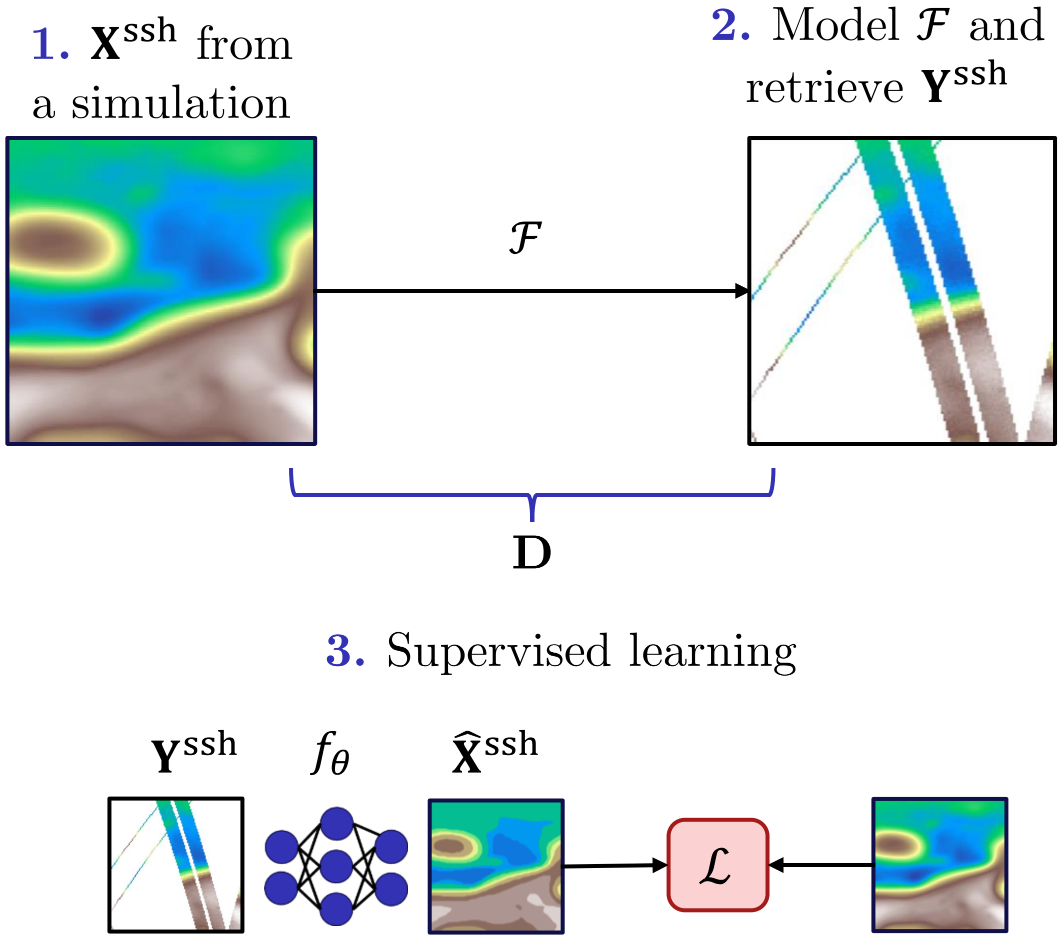
In geosciences the **ground truth $\mathbf{X}^{\mathrm{ssh}}$ is not available**.

**Simulation-based training**

1. The ground truth $\mathbf{X}^{\mathrm{ssh}}$ is taken from a numerical model
2. Emulate the forward operator $\mathcal{F}$ to generate observations : $\mathbf{Y}^{\mathrm{ssh}} = \mathcal{F}\left(\mathbf{X}^{\mathrm{ssh}}\right) + \varepsilon$

**OSSE**

3. Supervised learning: $\mathbf{D} = \left\{\mathbf{X}^{\mathrm{ssh}}, \mathbf{Y}^{\mathrm{ssh}}\right\}$ pairs
   - State estimation: $\widehat{\mathbf{X}}^{\mathrm{ssh}} = f_\theta\left(\mathbf{Y}^{\mathrm{ssh}}\right)$
   - Compute a loss function $\mathcal{L}(\widehat{\mathbf{X}}^{\mathrm{ssh}}, \mathbf{X}^{\mathrm{ssh}})$
   - Iterative parameter optimization by gradient descent: $\theta_{i+1} = \theta_i - \gamma \dfrac{\partial \mathcal{L}\left(\widehat{\mathbf{X}}^{\mathrm{ssh}}, \mathbf{X}^{\mathrm{ssh}}\right)}{\partial \theta_i}$



# Deep Neural Network training

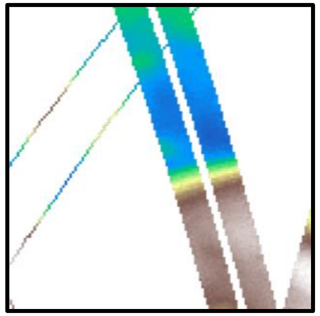
The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

In geosciences the **ground truth** $\mathbf{X}^{\text{ssh}}$ **is not available**.

**Observations-based training**

1. We only have access to real observations $\mathbf{Y}^{\text{ssh}}$

1. $\mathbf{Y}^{\text{ssh}}$ from real data



$\mathbf{D}$

# Deep Neural Network training

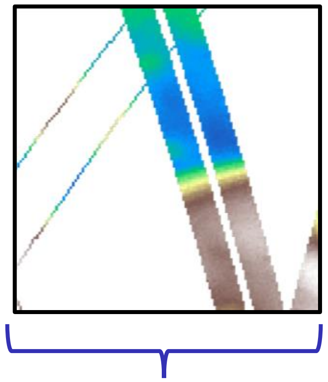
The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

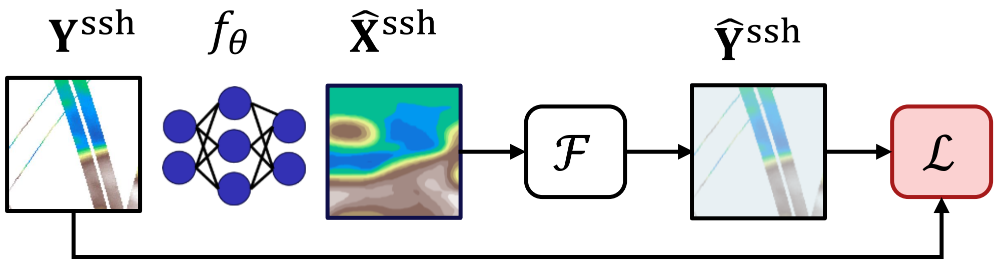
In geosciences the **ground truth** $\mathbf{X}^{\text{ssh}}$ **is not available**.

**Observations-based training**

1. We only have access to real observations $\mathbf{Y}^{\text{ssh}}$
2. Unsupervised learning: $\mathbf{D} = \{\mathbf{Y}^{\text{ssh}}\}$
   - **Self-supervised** learning: use observations as a proxy for the ground truth
   - State estimation: $\widehat{\mathbf{X}}^{\text{ssh}} = f_\theta(\mathbf{Y}^{\text{ssh}})$
   - Apply the forward operator: $\widehat{\mathbf{Y}}^{\text{ssh}} = \mathcal{F}(\widehat{\mathbf{X}}^{\text{ssh}})$
   - Compute $\mathcal{L}(\widehat{\mathbf{Y}}^{\text{ssh}}, \mathbf{Y}^{\text{ssh}})$
   - Adjust the parameters

- Works only if we find a way that prevent the network from copying its input on its output



# Deep Neural Network training

The neural network $f_\theta$ is trained using a dataset $\mathbf{D}$

In geosciences the **ground truth** $\mathbf{X}^{\text{ssh}}$ **is not available**.

Two options:

**Simulation-based** training

**Learning**

Allows supervised learning
Learns physical relationship

**Application to real data**

Domain gap problem: difference of distribution between simulation and observations

**Observations-based** training

**Learning**

Unsupervised learning is harder

**Application to real data**

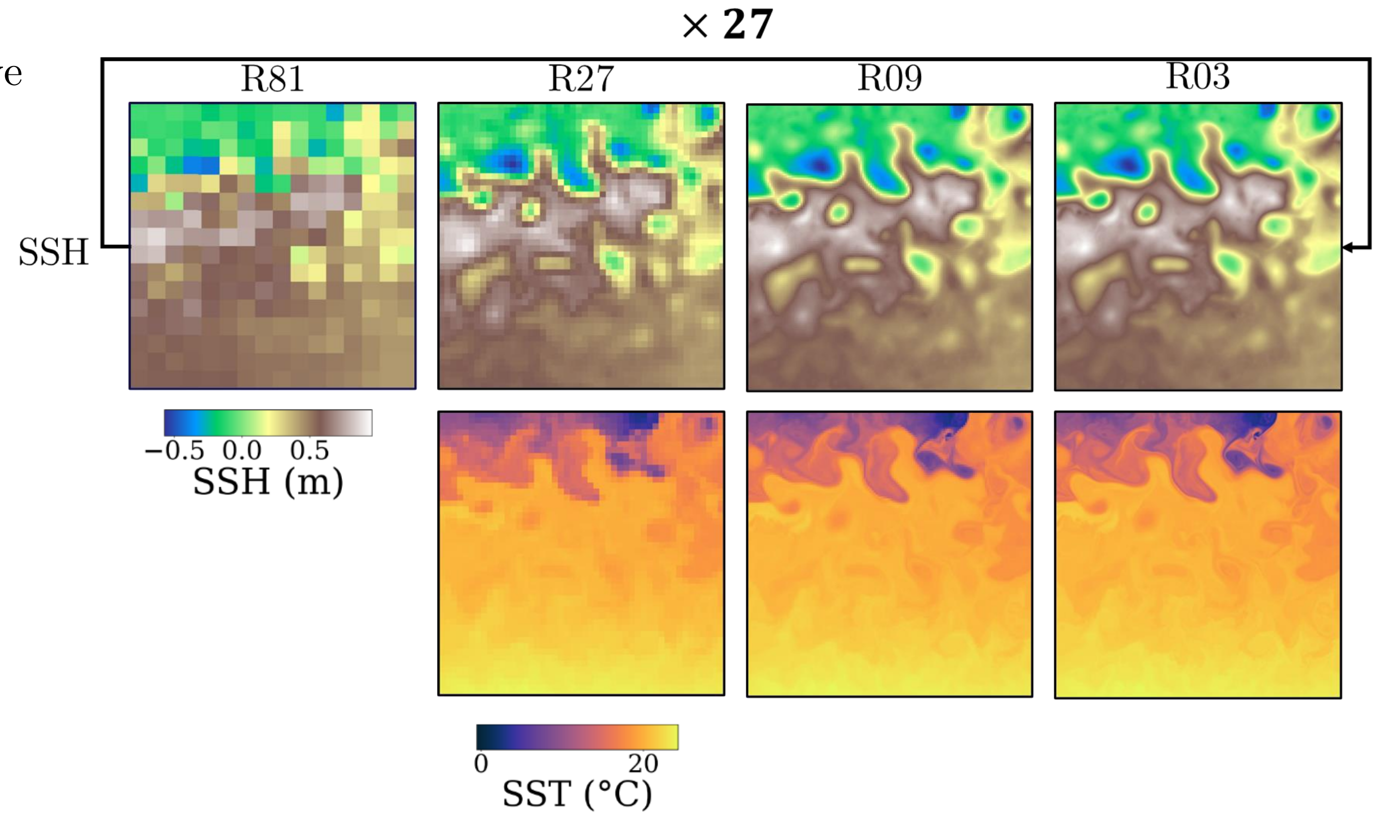
No domain gap problem

# Outline

1. Introduction
2. Satellite observations of height and temperature
3. Reconstruction using deep neural network
4. **An example of downscaling**
5. An example of interpolation
6. Conclusions and perspectives

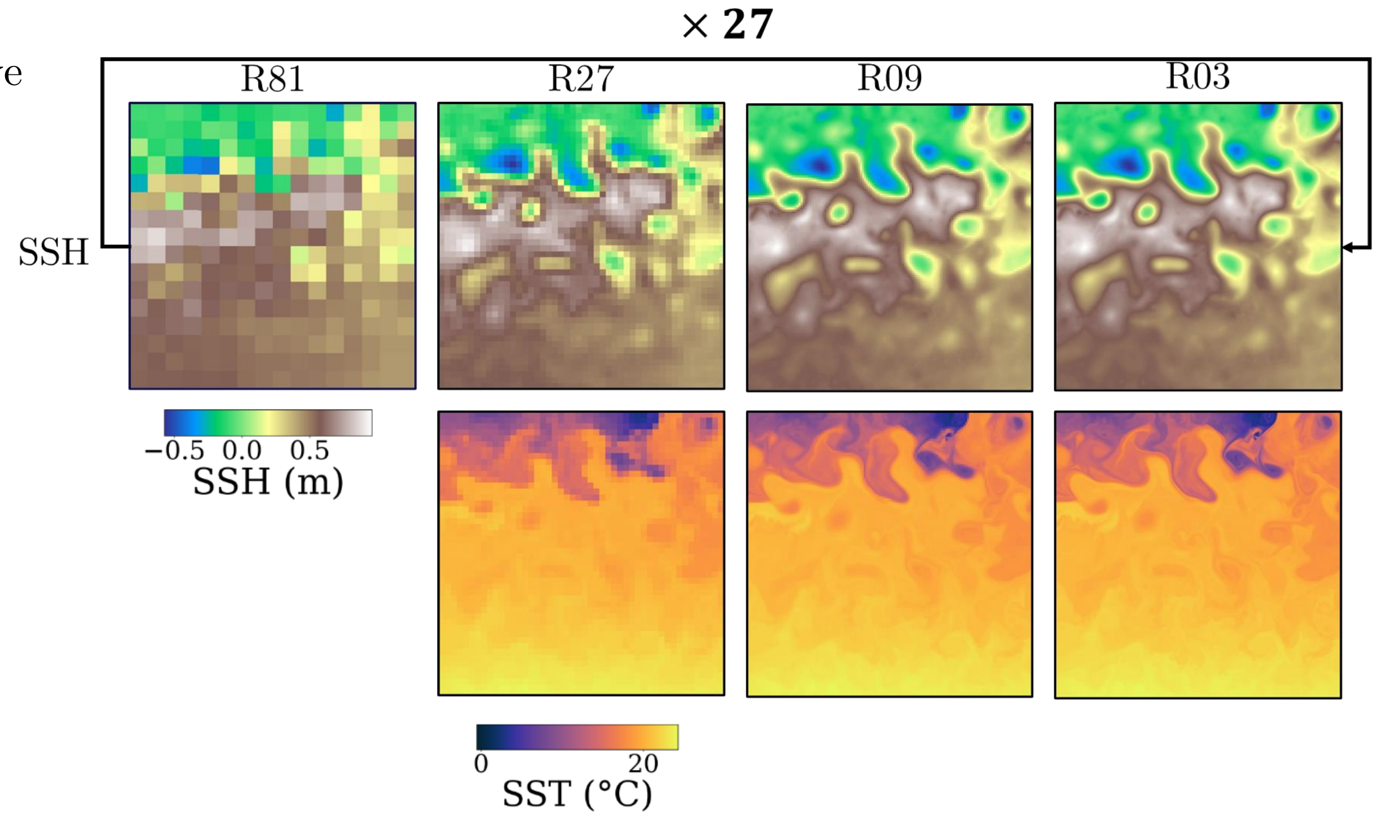
# Simulation data: NATL60

- NATL60:
  - No assimilation, free run of NEMO 3.6 model
  - Very high resolution 1/60 degrees ($\approx$ 1km) which we call R01
  - 365 days + 4 separated month (122 days)



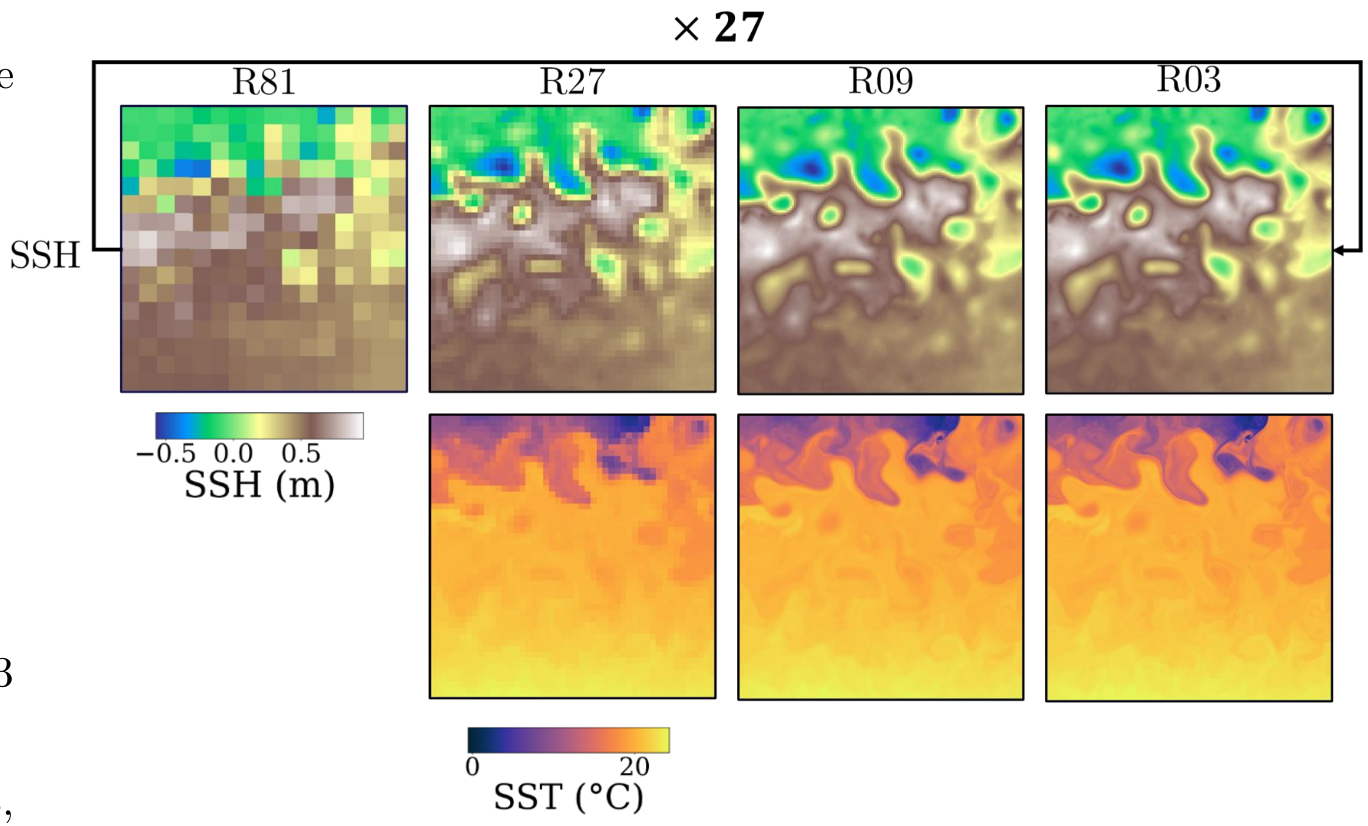
# Simulation data: NATL60

- NATL60:
  - No assimilation, free run of NEMO 3.6 model
  - Very high resolution 1/60 degrees ($\approx$ 1km) which we call R01
  - 365 days + 4 separated month (122 days)
- Working area: Gulf Stream
  - Latitudes 27° to 44° and longitude -64° to -42°
  - Geostrophic approximation valid
  - Important currents and temperature contrast



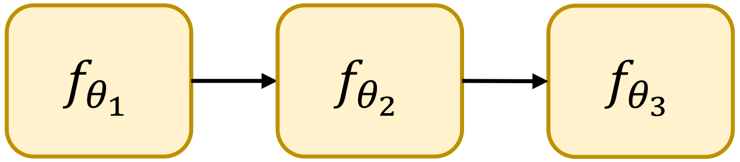
# Simulation data: NATL60

- NATL60:
  - No assimilation, free run of NEMO 3.6 model
  - Very high resolution 1/60 degrees ($\approx$ 1km) which we call R01
  - 365 days + 4 separated month (122 days)
- Working area: Gulf Stream
  - Latitudes 27° to 44° and longitude -64° to -42°
  - Geostrophic approximation valid
  - Important currents and temperature contrast
- Downscaling
  - Decimation operator $\mathbf{D_{ec}}$ is an average pool on 3 $\times$ 3 square
  - We downgrade the resolution from are R01 to R03, R09, R27, R81
  - **x27** SSH downscaling
  - Estimation of surface currents



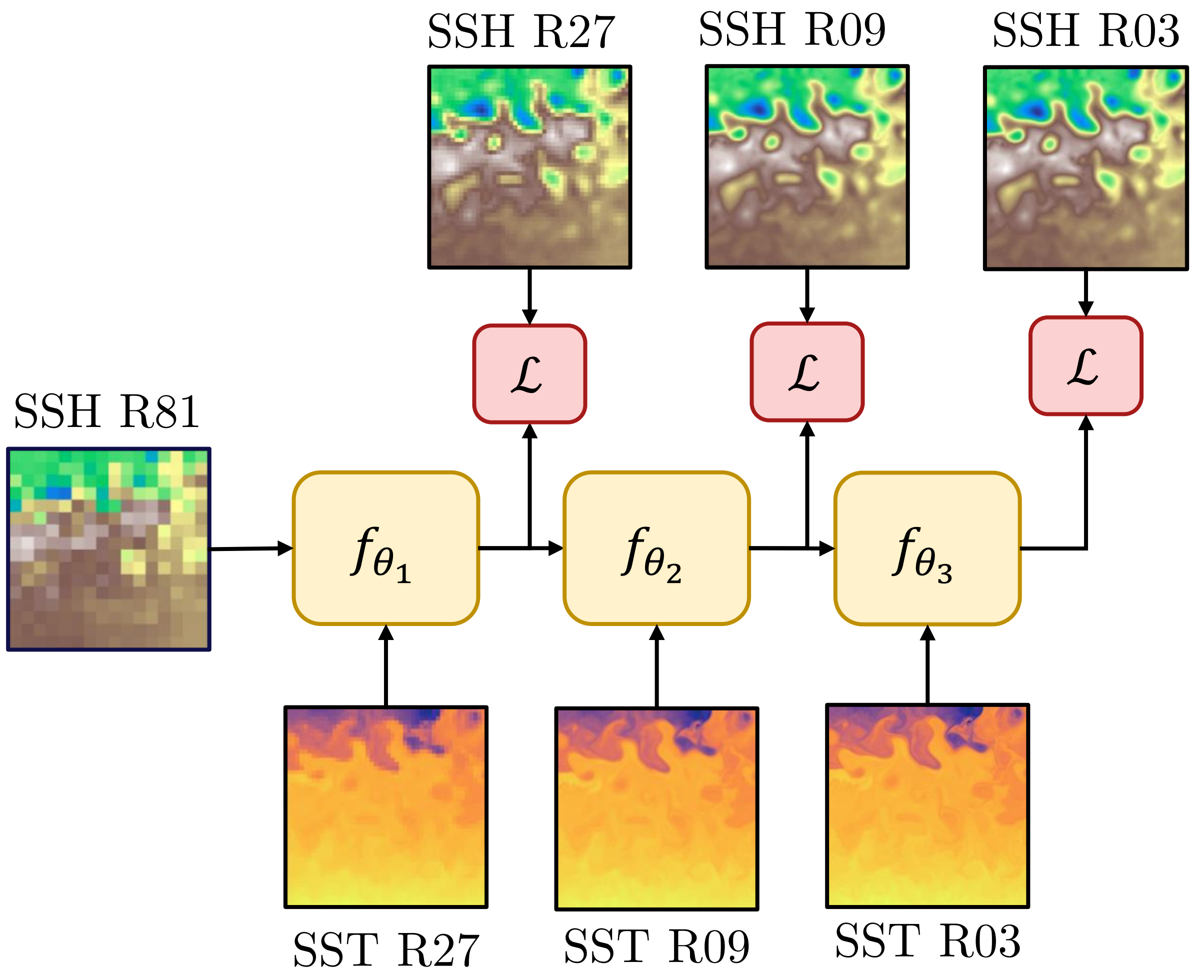
# Resolution by Stage of Altimetry and Currents

- RESAC: 3 neural networks, each performing a $\times 3$ upsampling



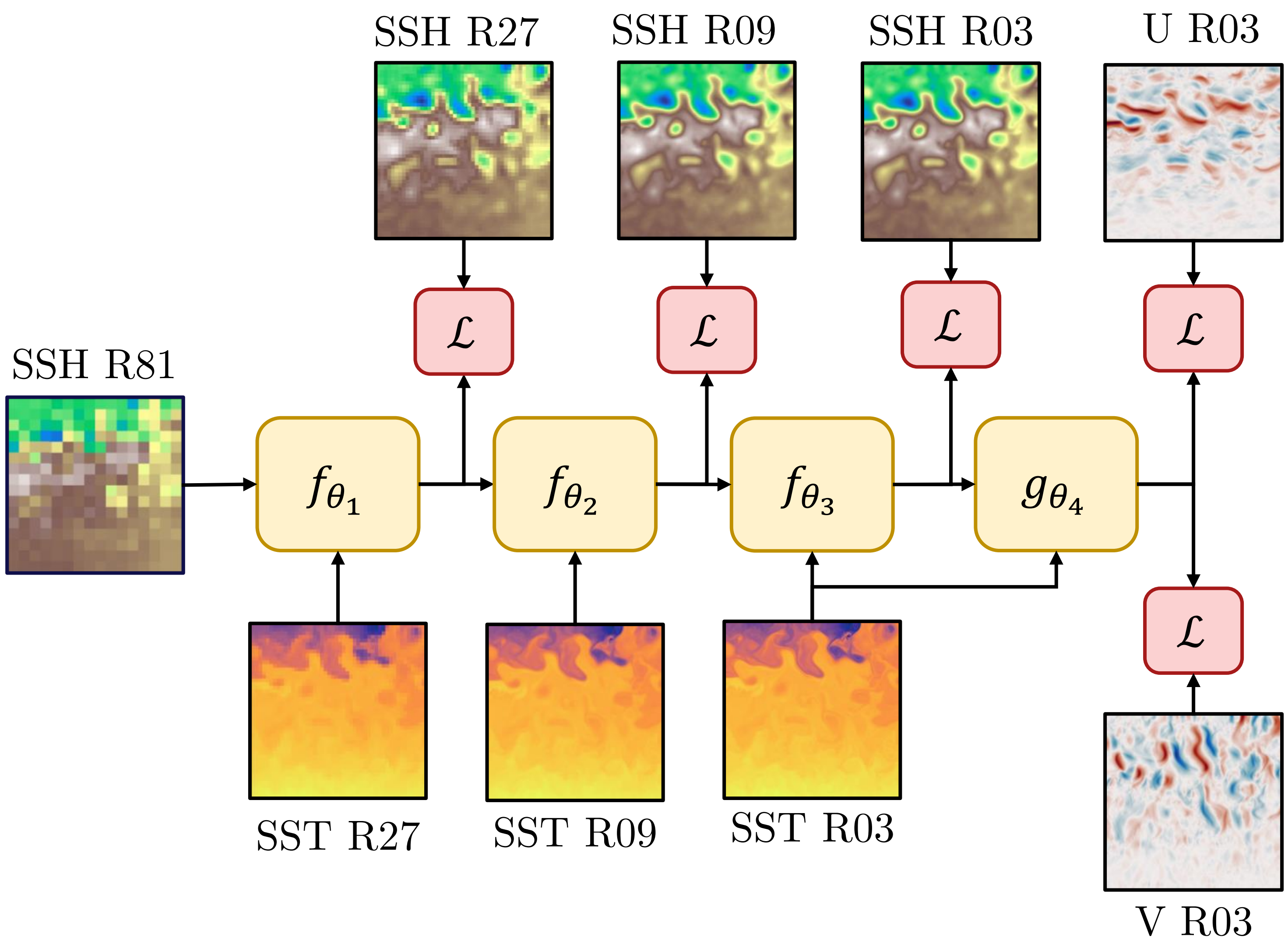
# Resolution by Stage of Altimetry and Currents

- RESAC: 3 neural networks, each performing a $\times 3$ upsampling
- Each block takes a LR SSH and a HR SST image



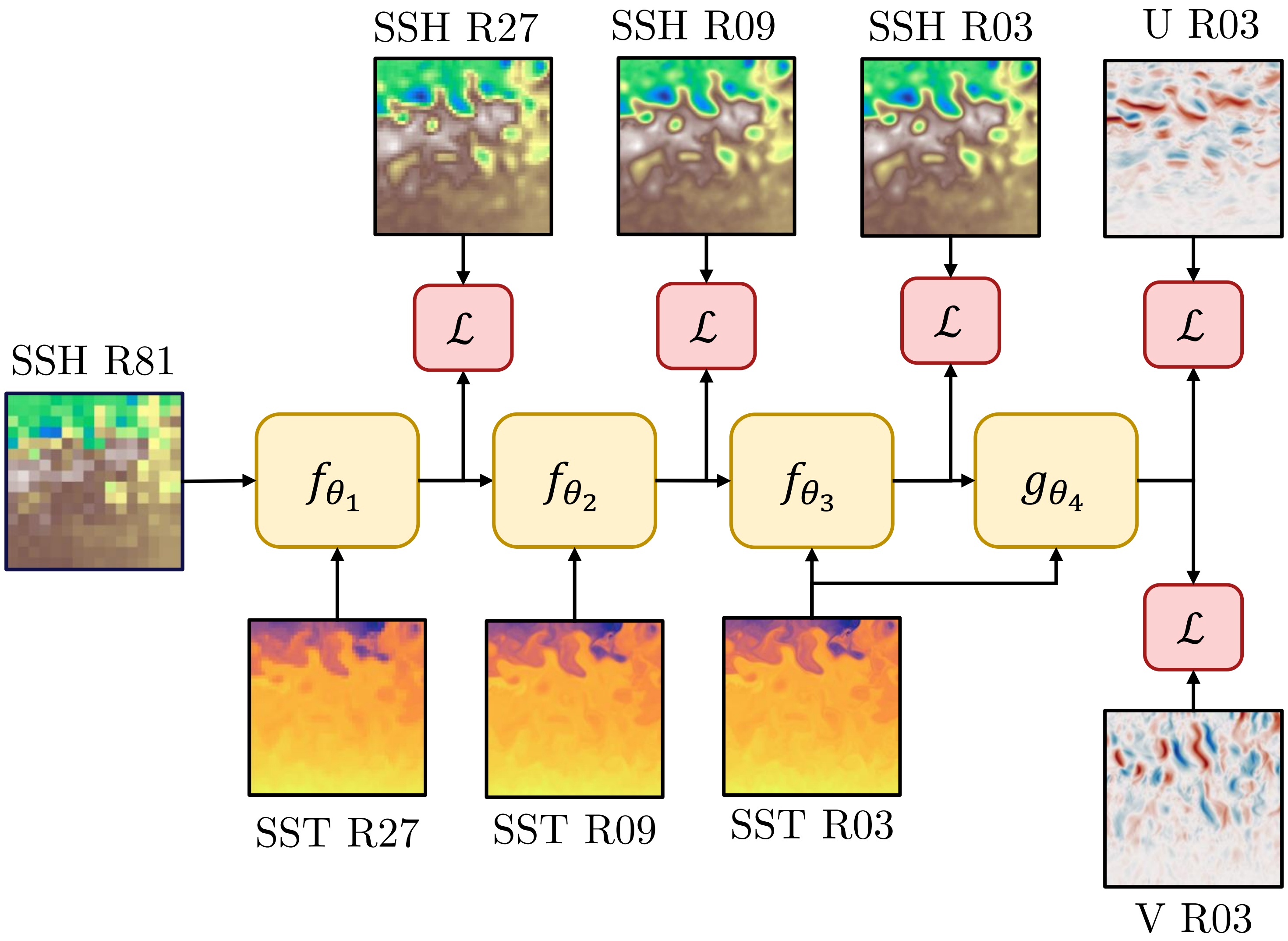
# Resolution by Stage of Altimetry and Currents

- RESAC: 3 neural networks, each performing a $\times 3$ upsampling
- Each block takes a LR SSH and a HR SST image
- Each block is controlled at its output resolution
- Optionally, a last neural network estimates the surface currents, from the SR SSH and the SST



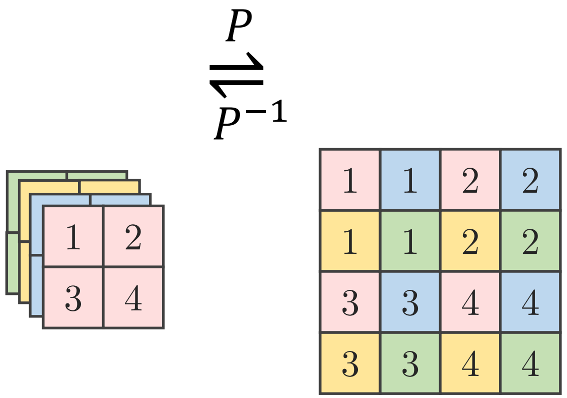
# Resolution by Stage of Altimetry and Currents

- RESAC: 3 neural networks, each performing a $\times 3$ upsampling
- Each block takes a LR SSH and a HR SST image
- Each block is controlled at its output resolution
- Optionally, a last neural network estimates the surface currents, from the SR SSH and the SST
- The final loss function is the sum of the all losses

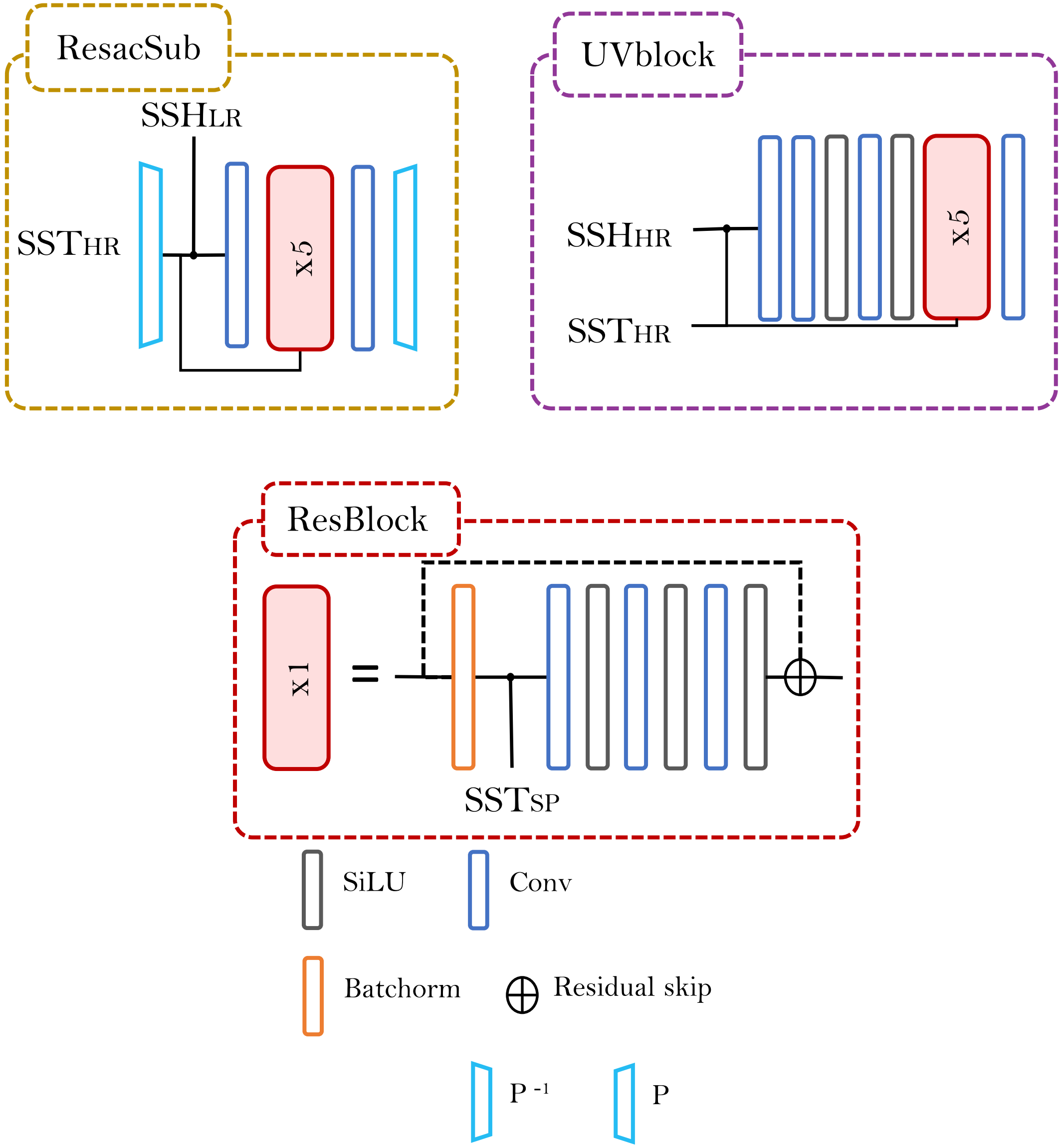


# Super Resolution Convolutional Neural Network

- Subpixel convolution

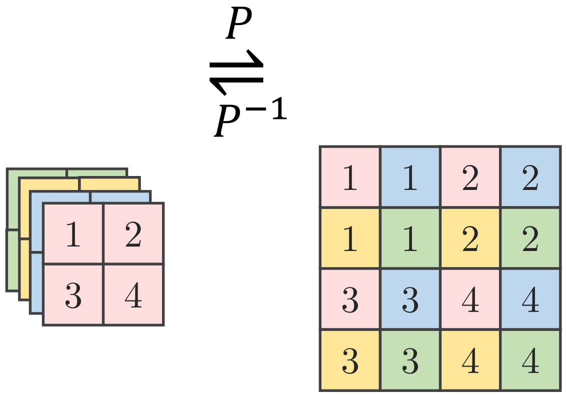



- Residual learning

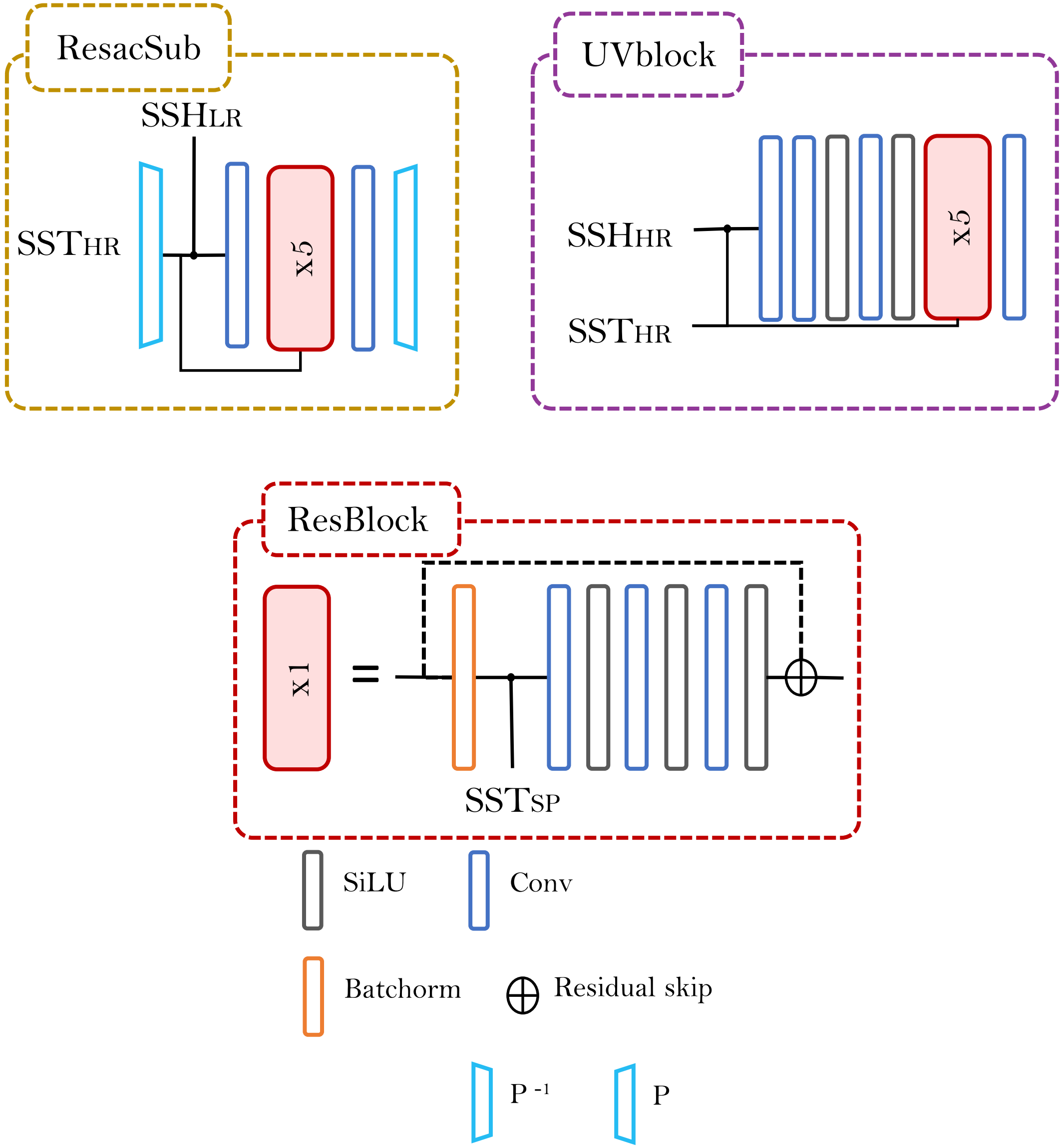


# Super Resolution Convolutional Neural Network

- Subpixel convolution




- Residual learning
- Training details:
  - 10-member ensemble
  - Normalization by (center reduce the distribution of each data)
  - Dataset split: 1 year for training, March and June for validation and September and December to test



# Results on the test set

- Experiments: Bicubic upsampling, RESAC using SSH only or SSH + SST as input
- Loss: Multiscale and current MSE

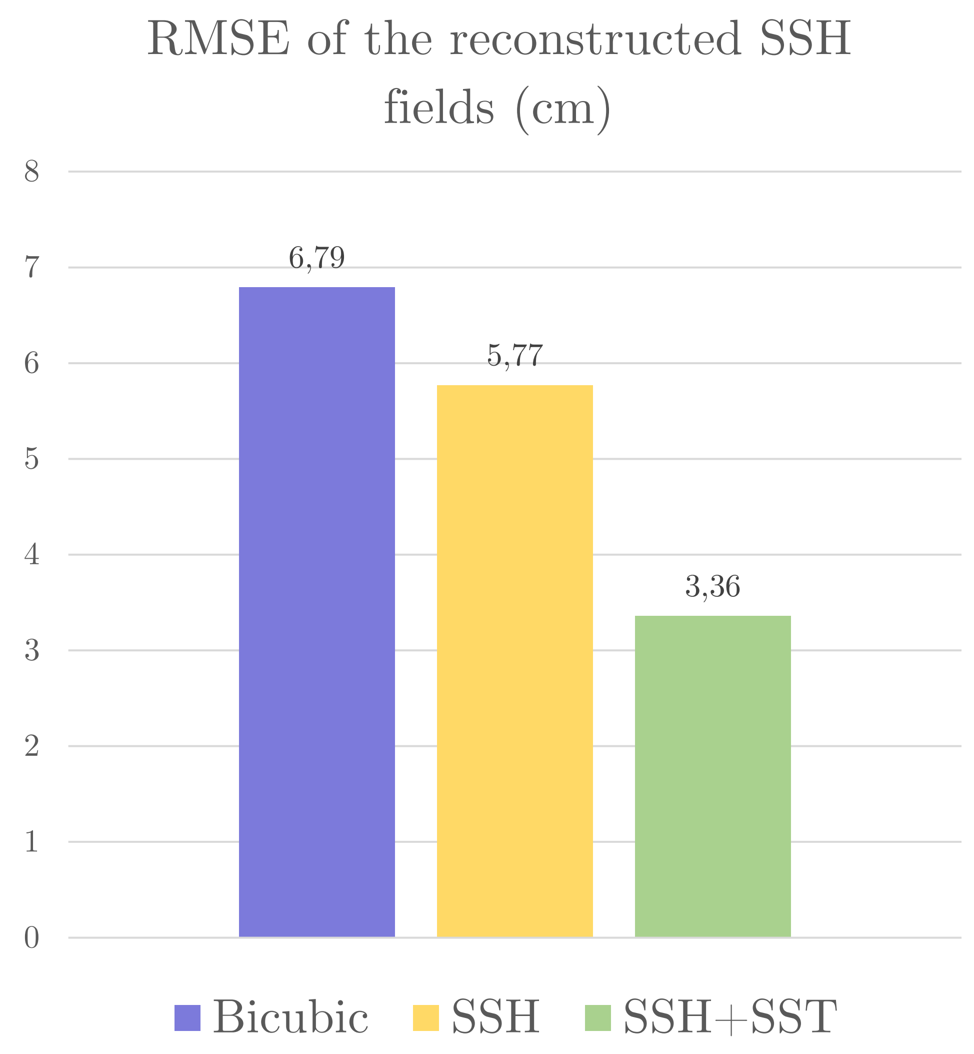
# Results on the test set

- Experiments: Bicubic upsampling, RESAC using SSH only or SSH + SST as input
- Loss: Multiscale and current MSE

Conclusions

- SST decreases the errors:

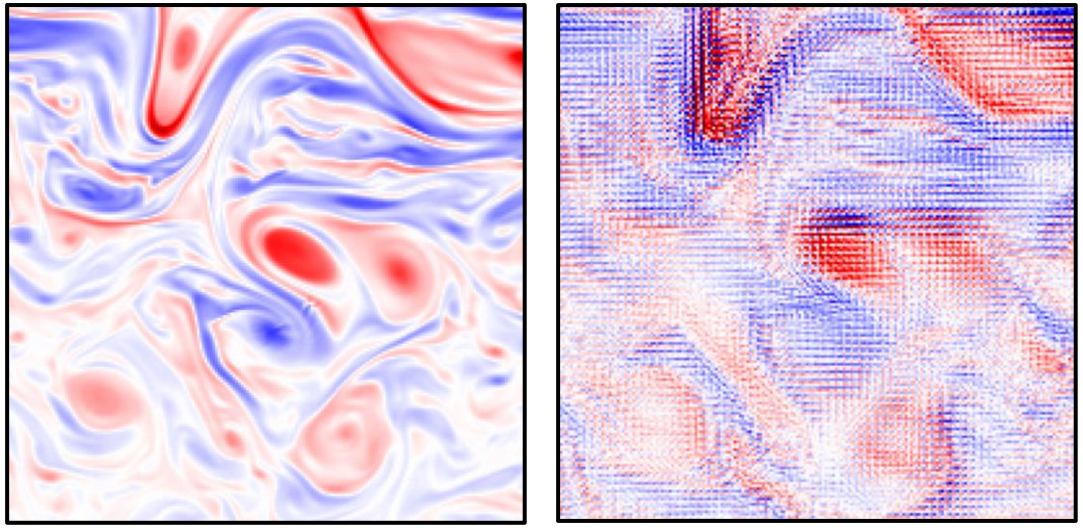
  -2,11 cm and -47% of the RMSE
- Subpixel convolution introduces artifacts



# Visual comparison

| | | |
|---|---|---|
| **Inputs**: | ∅ | SSH + SST |
| **Currents**: | Ground truth | Geostrophy |

Relative Vorticity

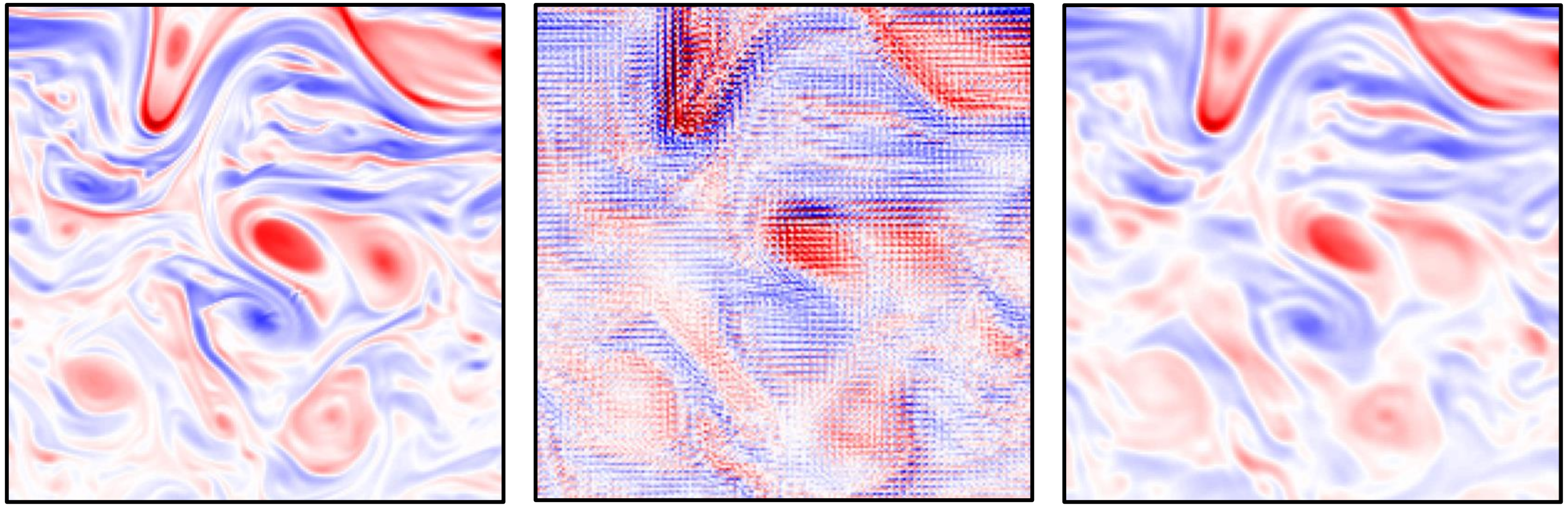


- The currents computed from the subpixel output using geostrophy present strong artifacts

# Visual comparison

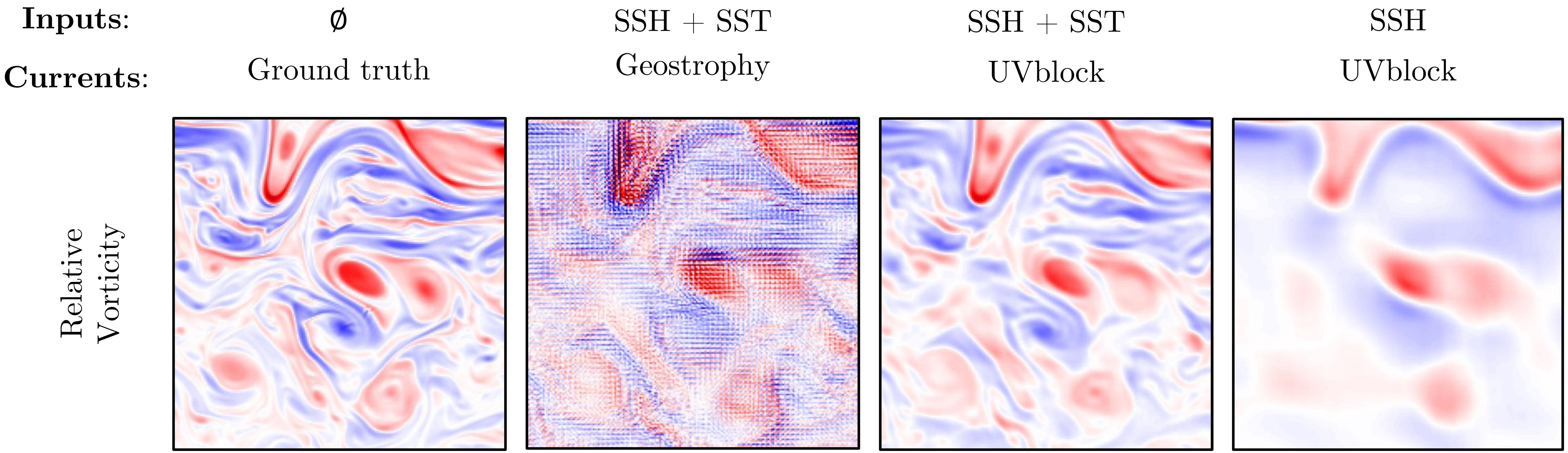
| | | | |
|---|---|---|---|
| **Inputs**: | ∅ | SSH + SST | SSH + SST |
| **Currents**: | Ground truth | Geostrophy | UVblock |

Relative Vorticity



- The currents computed from the subpixel output using geostrophy present strong artifacts
- These artifacts disappear after a the current block

# Visual comparison




- The currents computed from the subpixel output using geostrophy present strong artifacts
- These artifacts disappear after a the current block
- Using SST image, more precise structures are retrieved

# Limitations

Difficulties toward applicability to real-world observations:

# Limitations

Difficulties toward applicability to real-world observations:

- The chosen decimation operator $\mathbf{D_{ec}}$ is naive:
  - The errors of the DUACS product are not easy to model
  - Adapting $\mathbf{D_{ec}}$ to emulate DUACS product is difficult

# Limitations

Difficulties toward applicability to real-world observations:

- The chosen decimation operator $\mathbf{D_{ec}}$ is naive:
  - The errors of the DUACS product are not easy to model
  - Adapting $\mathbf{D_{ec}}$ to emulate DUACS product is difficult
- SST observation are noisy

# Limitations

Difficulties toward applicability to real-world observations:

- The chosen decimation operator $\mathbf{D_{ec}}$ is naive:
  - The errors of the DUACS product are not easy to model
  - Adapting $\mathbf{D_{ec}}$ to emulate DUACS product is difficult
- SST observation are noisy
- NATL60 simulation is extremely computationally intensive to run, which limits the amount of data available

# Limitations

Difficulties toward applicability to real-world observations:

- The chosen decimation operator $\mathbf{D_{ec}}$ is naive:
  - The errors of the DUACS product are not easy to model
  - Adapting $\mathbf{D_{ec}}$ to emulate DUACS product is difficult
- SST observation are noisy
- NATL60 simulation is extremely computationally intensive to run, which limits the amount of data available

➢ The application of this method to real data is difficult

# Outline

1. Introduction
2. Satellite observations of height and temperature
3. Reconstruction using deep neural network
4. An example of downscaling
5. **An example of interpolation**
6. Conclusions and perspectives

# From downscaling to interpolation

1. Produce a new simulation dataset
    - Representing the interpolation task
    - As realistic as possible

# From downscaling to interpolation

1. Produce a new simulation dataset
   - Representing the interpolation task
   - As realistic as possible
2. Develop new training methods on this dataset
   - Able to learn without ground truth

# From downscaling to interpolation

1. Produce a new simulation dataset
   - Representing the interpolation task
   - As realistic as possible
2. Develop new training methods on this dataset
   - Able to learn without ground truth
3. Apply the method to real data

# Observing System Simulation Experiment

Existing OSSE: **Ocean Data challenge 2020**[1] **Ours**

[1] *CLS and MEOM Team from IGE (CNRS-UGA-IRD-G-INP)*

# Observing System Simulation Experiment

| | | | |
|---|---|---|---|
| Existing OSSE: | **Ocean Data challenge 2020**[1] | **Ours** | |
| Simulation: | NATL60 | Glorys12 | $\rightarrow$ lower resolution/more data |

[1] *CLS and MEOM Team from IGE (CNRS-UGA-IRD-G-INP)*

# Observing System Simulation Experiment

| | | | |
|---|---|---|---|
| Existing OSSE: | **Ocean Data challenge 2020**[1] | **Ours** | |
| Simulation: | NATL60 | Glorys12 | → lower resolution/more data |
| Temporal extent: | 1 year | 20 years | → more suited to DL learning |

[1] *CLS and MEOM Team from IGE (CNRS-UGA-IRD-G-INP)*

# Observing System Simulation Experiment

| Existing OSSE: | **Ocean Data challenge 2020**[1] | **Ours** | |
|---|---|---|---|
| Simulation: | NATL60 | Glorys12 | → lower resolution/more data |
| Temporal extent: | 1 year | 20 years | → more suited to DL learning |
| Data: | SSH only (nadir and SWOT) | SSH (nadir) and SST → multivariate | |

[1] *CLS and MEOM Team from IGE (CNRS-UGA-IRD-G-INP)*

# Observing System Simulation Experiment

| | | |
|---|---|---|
| Existing OSSE: | **Ocean Data challenge 2020**[1] | **Ours** |
| Simulation: | NATL60 | Glorys12 → lower resolution/more data |
| Temporal extent: | 1 year | 20 years → more suited to DL learning |
| Data: | SSH only (nadir and SWOT) | SSH (nadir) and SST → multivariate |
| Area: | Gulf Stream between latitude 33° to 43° and longitude -65° to -55° | Identical |

[1] *CLS and MEOM Team from IGE (CNRS-UGA-IRD-G-INP)*

# Observing System Simulation Experiment

| Existing OSSE: | **Ocean Data challenge 2020**[1] | **Ours** | |
|---|---|---|---|
| Simulation: | NATL60 | Glorys12 | → lower resolution/more data |
| Temporal extent: | 1 year | 20 years | → more suited to DL learning |
| Data: | SSH only (nadir and SWOT) | SSH (nadir) and SST → multivariate | |
| Area: | Gulf Stream between latitude 33° to 43° and longitude -65° to -55° | Identical | |

State-of-the-art reconstruction methods:

- **Optimal Interpolation**
- **Nudging**
- **Neural Network**

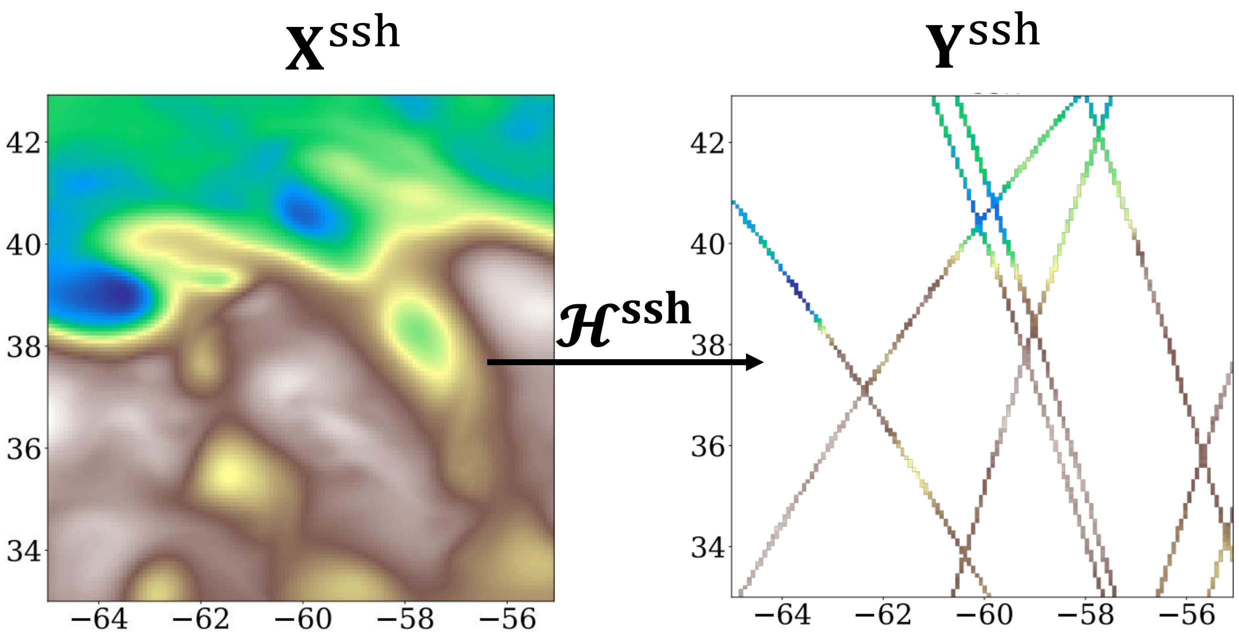
| Method | Source | Type | SST | learning |
|---|---|---|---|---|
| **DUACS** | Taburet et al., 2019 | OI | no | no |
| **DYMOST** | Ubelmann et al., 2016<br>Ballarotta et al., 2020 | OI | no | no |
| **MIOST** | Ardhuin et al., 2020 | OI | no | no |
| **BFN-QG** | Le Guillou et al., 2020 | DA | no | no |
| **4DVarNet** | Fablet et al., 2021 | NN | no | simulation |
| **ConvLSTM** | Martin et al., 2023 | NN | yes | observations |

[1] *CLS and MEOM Team from IGE (CNRS-UGA-IRD-G-INP)*

# Observing System Simulation Experiment

SSH observing operator $\mathcal{H}^{\mathbf{ssh}}$:

- Goal: emulate nadir-pointing along track observations

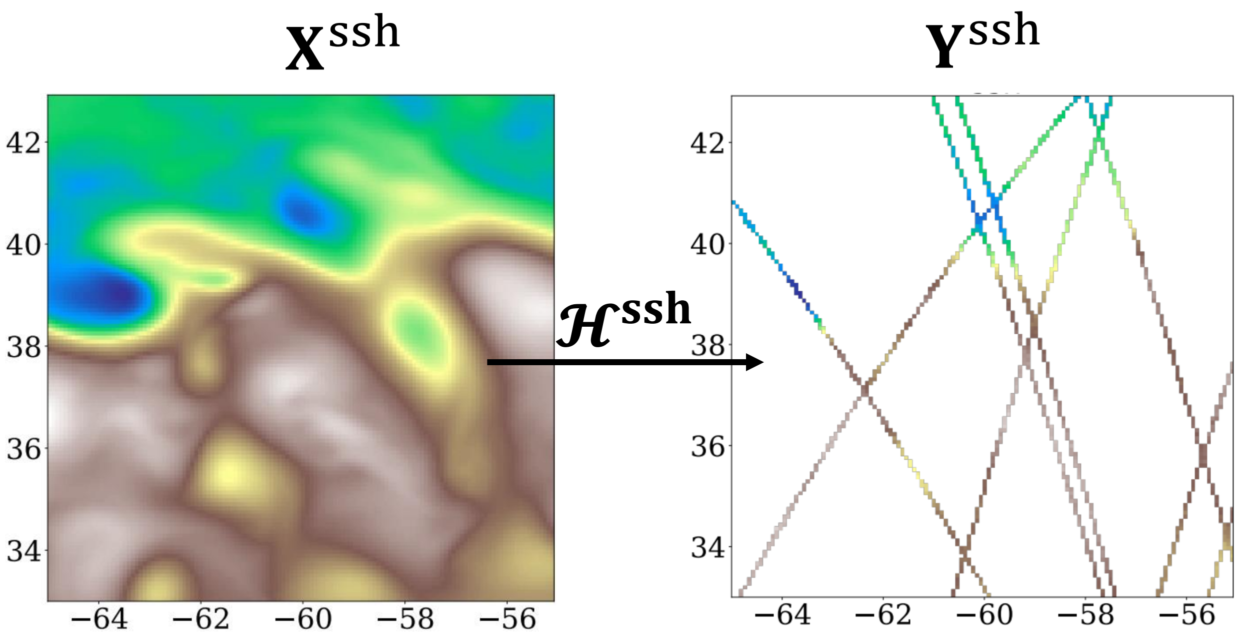


# Observing System Simulation Experiment

SSH observing operator $\boldsymbol{\mathcal{H}}^{\mathbf{ssh}}$:

- Goal: emulate nadir-pointing along track observations

1. Retrieve the support of real nadir-pointing observations from satellite products

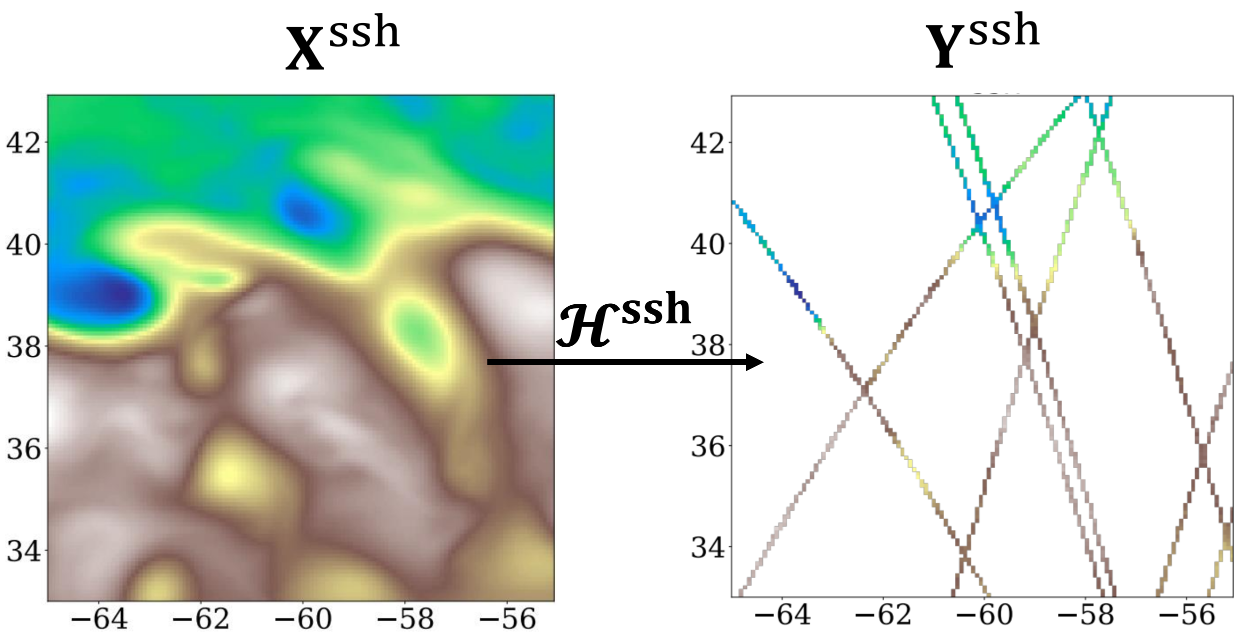


# Observing System Simulation Experiment

SSH observing operator $\boldsymbol{\mathcal{H}}^{\mathbf{ssh}}$:

- Goal: emulate nadir-pointing along track observations

1. Retrieve the support of real nadir-pointing observations from satellite products
2. Select the value in $\mathbf{X}^{\mathbf{ssh}}$ corresponding to the spatio-temporal coordinate of each observation

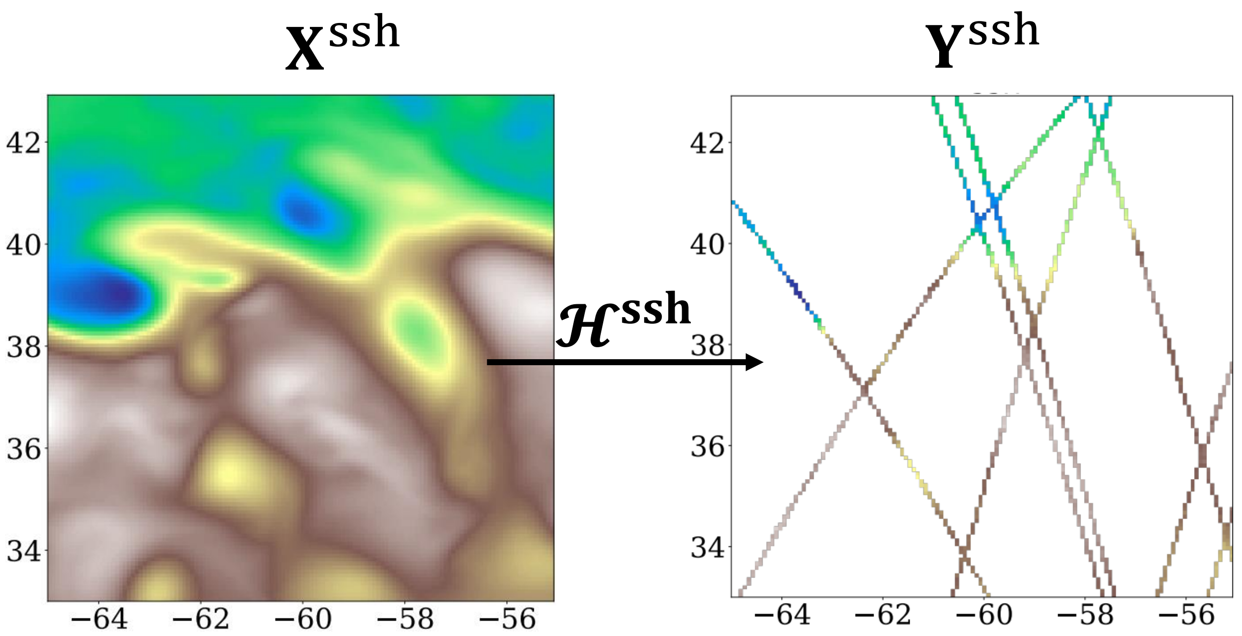


# Observing System Simulation Experiment

SSH observing operator $\boldsymbol{\mathcal{H}}^{\mathbf{ssh}}$:

- Goal: emulate nadir-pointing along track observations

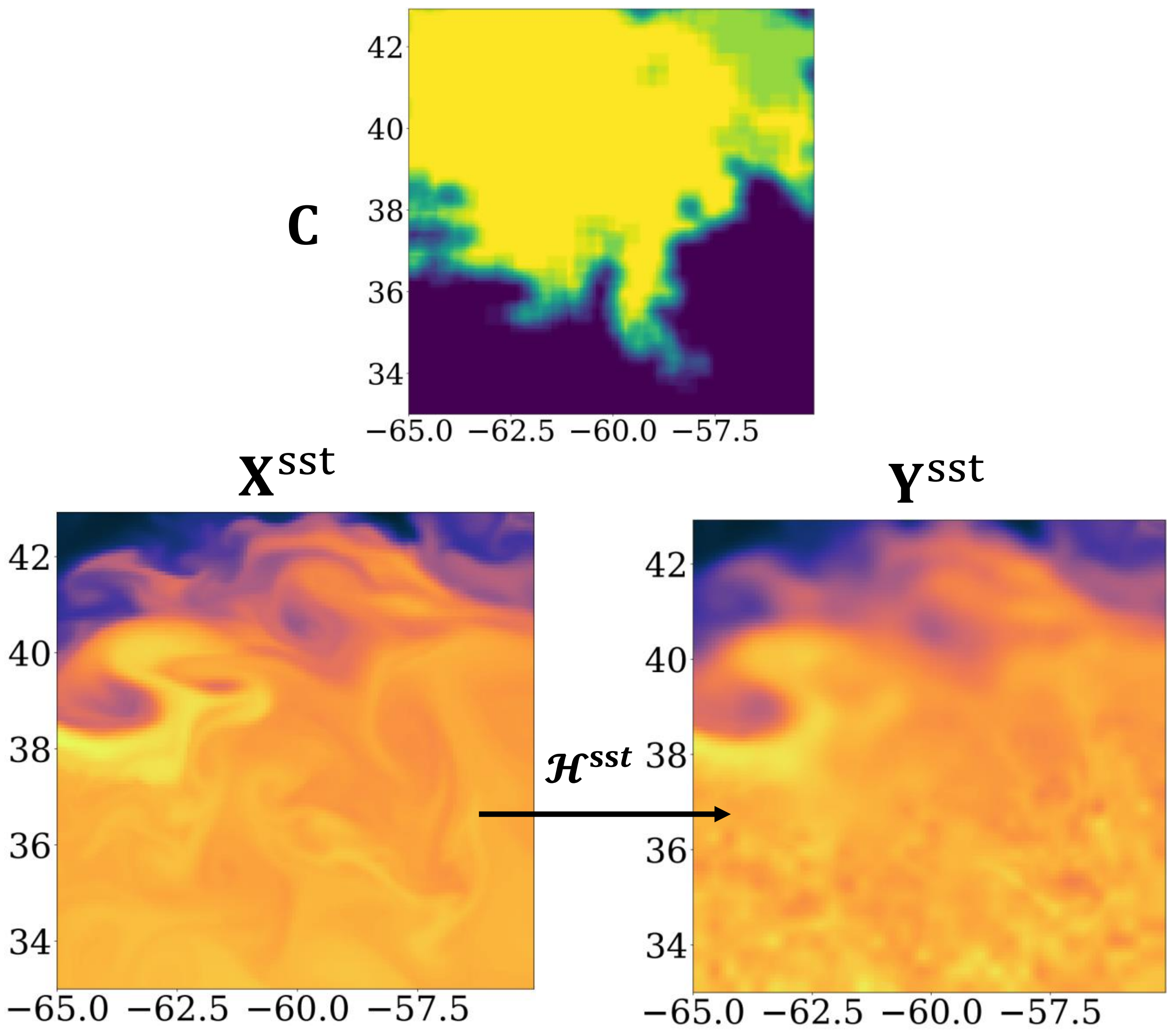
1. Retrieve the support of real nadir-pointing observations from satellite products
2. Select the value in $\mathbf{X}^{\mathrm{ssh}}$ corresponding to the spatio-temporal coordinate of each observation
3. Add Gaussian noise with 2 cm of std



# Observing System Simulation Experiment

SST observing operator $\mathcal{H}^{sst}$ :

- Goal: emulate the errors of complete SST product
  - Errors with inequal resolution
  - Blurring effect from OI process

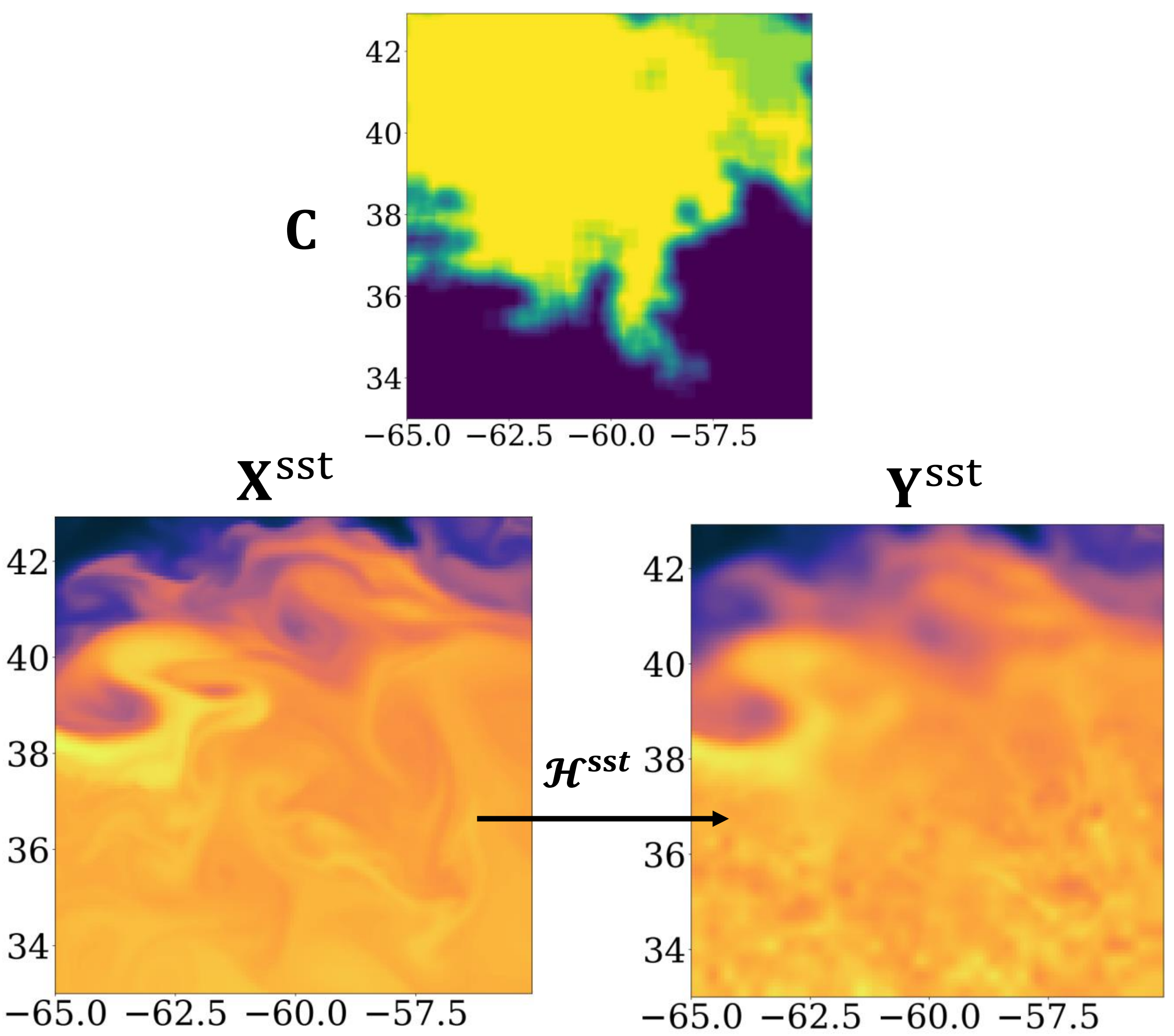


# Observing System Simulation Experiment

SST observing operator $\mathcal{H}^{sst}$ :

- Goal: emulate the errors of complete SST product
  - Errors with inequal resolution
  - Blurring effect from OI process

1. Retrieve a cloud cover $\mathbf{C}$, from real SST data

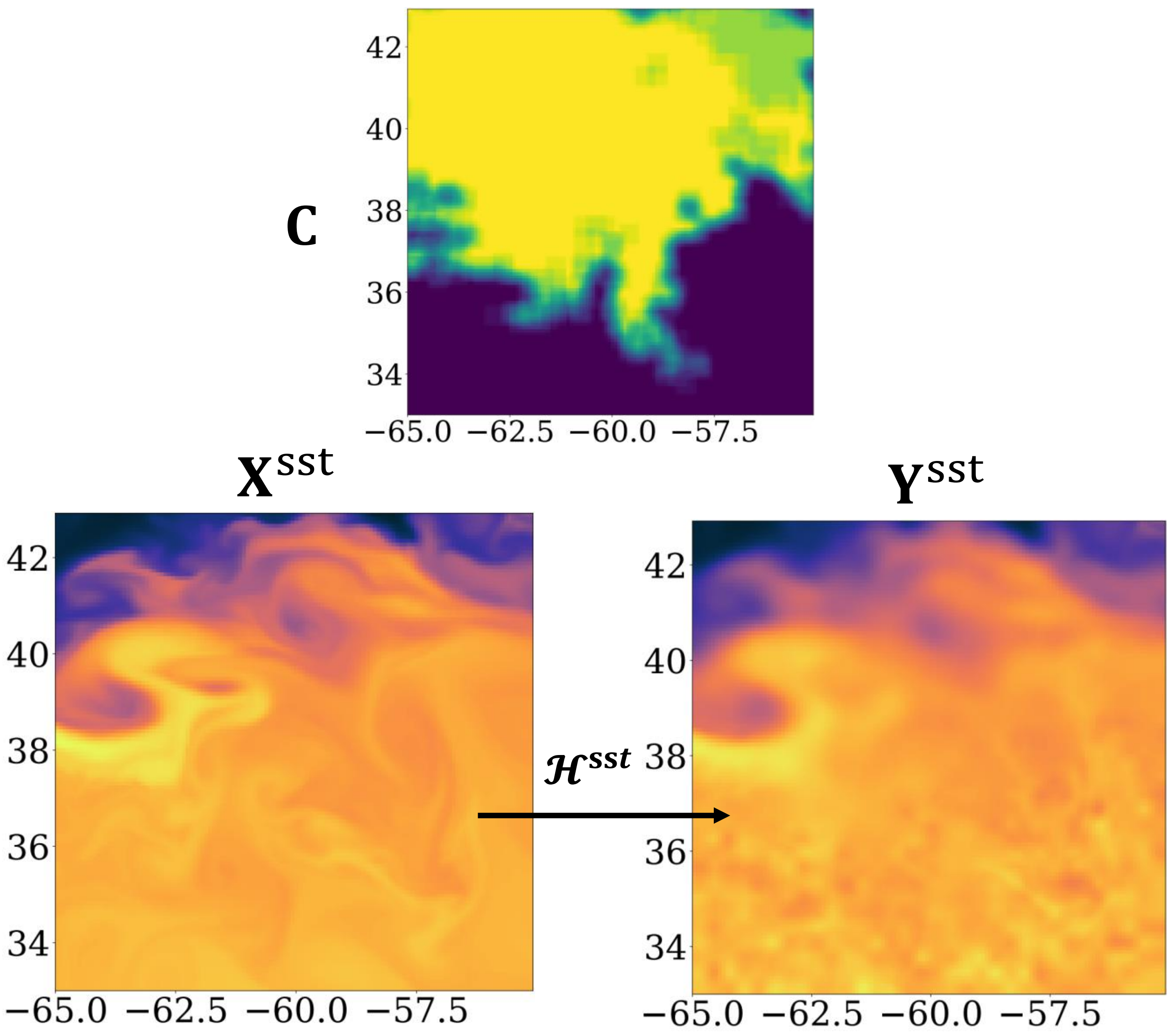


# Observing System Simulation Experiment

SST observing operator $\boldsymbol{\mathcal{H}}^{sst}$ :

- Goal: emulate the errors of complete SST product
  - Errors with inequal resolution
  - Blurring effect from OI process

1. Retrieve a cloud cover $\mathbf{C}$, from real SST data
2. We add noise $\varepsilon$, to emulate instrumental errors

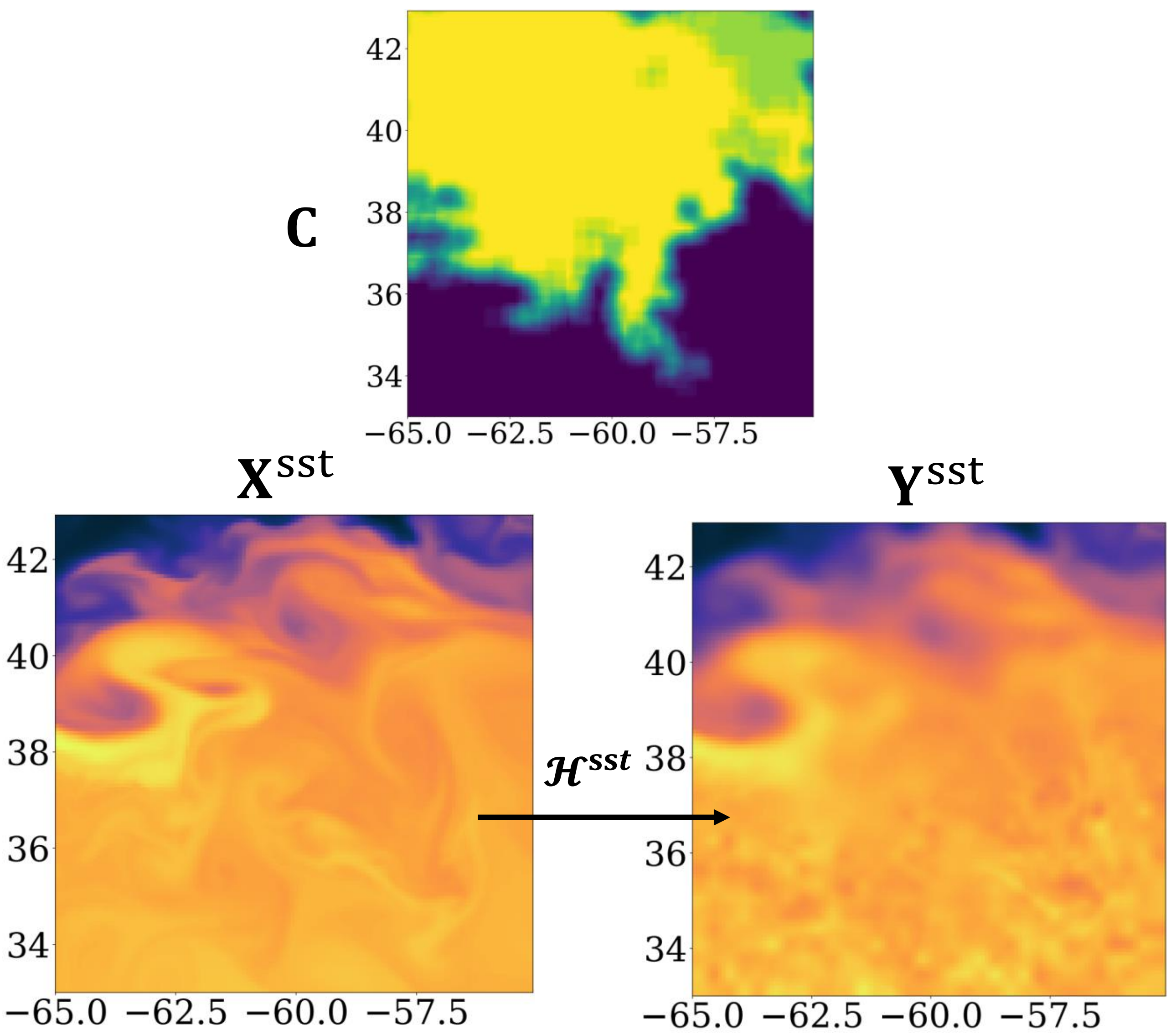


# Observing System Simulation Experiment

SST observing operator $\boldsymbol{\mathcal{H}}^{sst}$ :

- Goal: emulate the errors of complete SST product
    - Errors with inequal resolution
    - Blurring effect from OI process

1. Retrieve a cloud cover $\mathbf{C}$, from real SST data
2. We add noise $\varepsilon$, to emulate instrumental errors
3. Perform a spatio-temporal Gaussian blur $\mathrm{G}_{\sigma_t,\sigma_x}$ in cloudy area to emulate OI's blurring effect



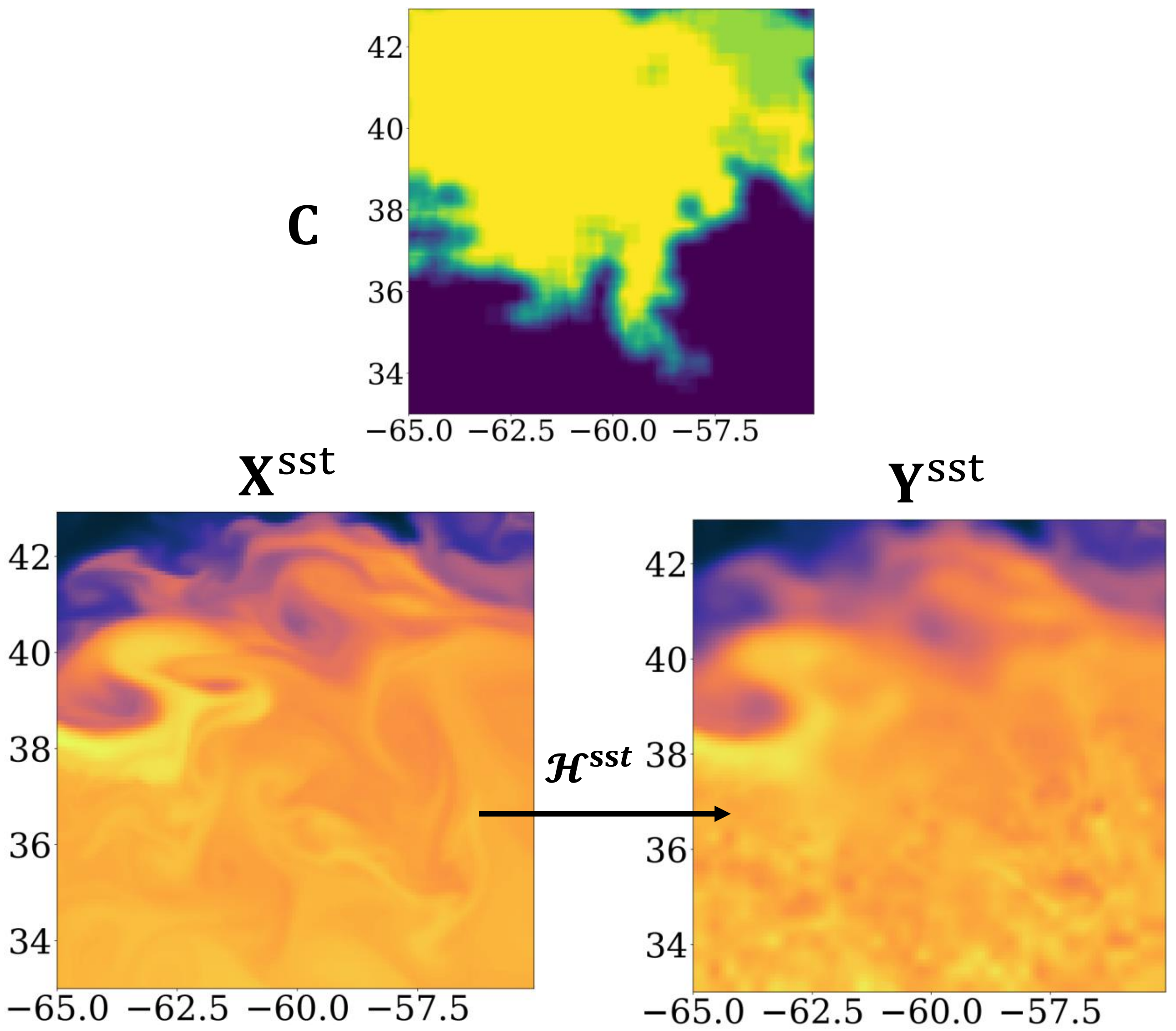
# Observing System Simulation Experiment

SST observing operator $\boldsymbol{\mathcal{H}}^{\mathbf{sst}}$ :

- Goal: emulate the errors of complete SST product
  - Errors with inequal resolution
  - Blurring effect from OI process

1. Retrieve a cloud cover $\mathbf{C}$, from real SST data
2. We add noise $\varepsilon$, to emulate instrumental errors
3. Perform a spatio-temporal Gaussian blur $\mathrm{G}_{\sigma_t,\sigma_x}$ in cloudy area to emulate OI's blurring effect
4. Combine the two noises

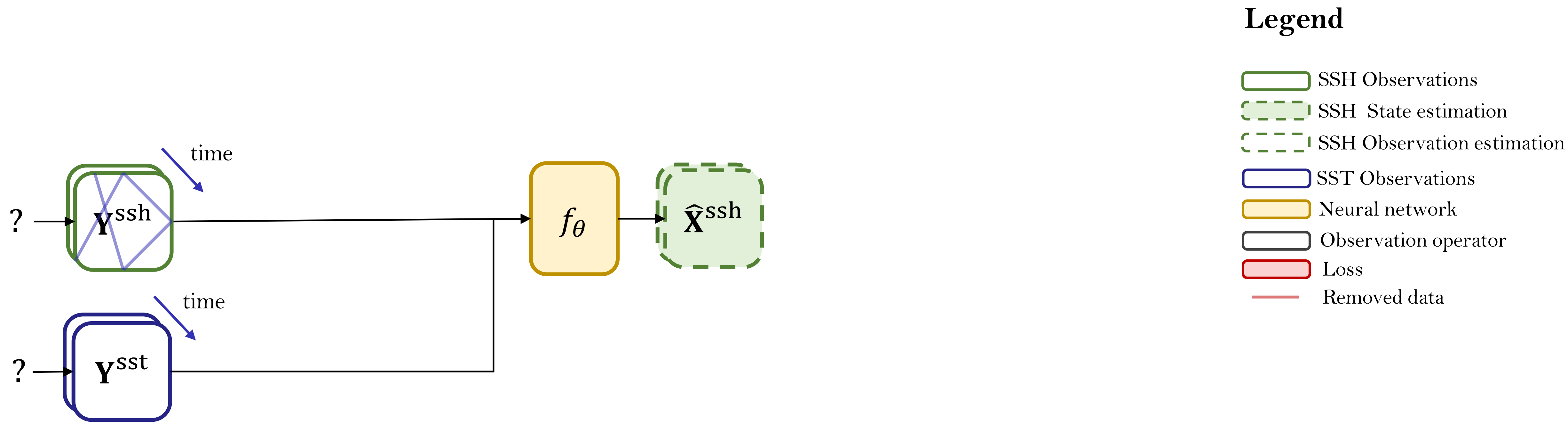
$$\mathbf{Y}^{\mathrm{sst}} = (1 - \mathbf{C})\odot(\mathbf{X}^{\mathrm{sst}}+\varepsilon) + \mathbf{C}\odot\mathrm{G}_{\sigma_t,\sigma_x} \star (\mathbf{X}^{\mathrm{sst}} + \varepsilon)$$



# Unsupervised learning

Given this dataset we can test unsupervised learning methods:

- $\hat{\mathbf{X}}^{\mathrm{ssh}} = f_\theta(\mathbf{Y}^{\mathrm{ssh}}, \mathbf{Y}^{\mathrm{sst}})$

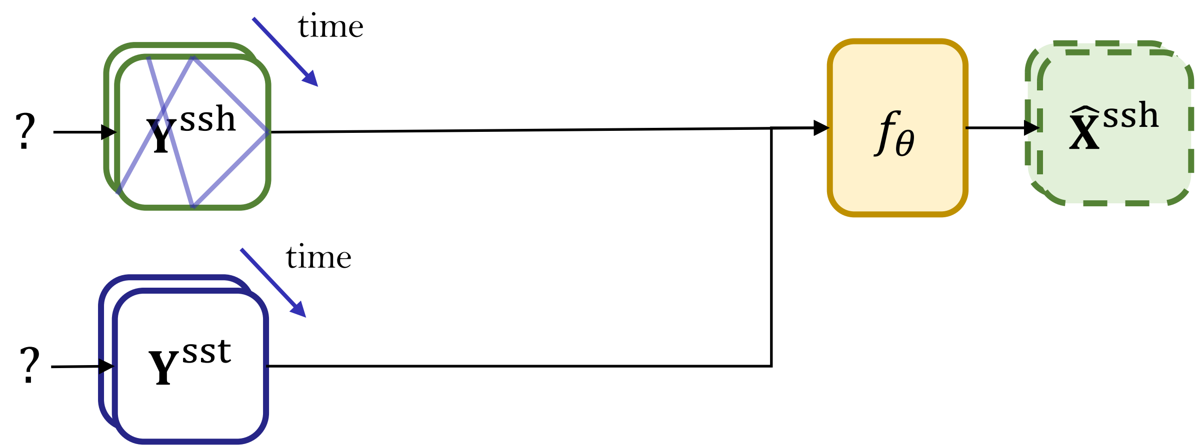


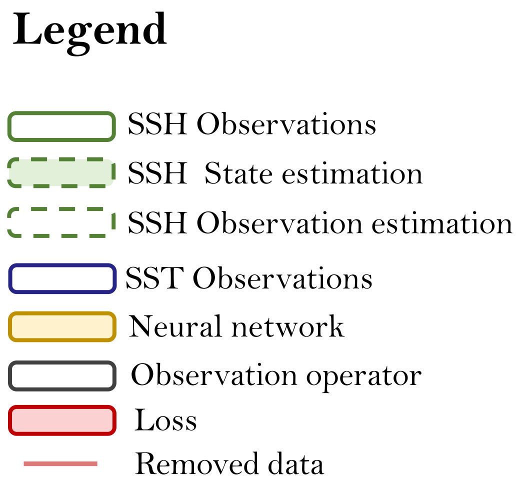
# Unsupervised learning

Given this dataset we can test unsupervised learning methods:

- $\hat{\mathbf{X}}^{\text{ssh}} = f_\theta(\mathbf{Y}^{\text{ssh}}, \mathbf{Y}^{\text{sst}})$

In this setting the all the tensors are **spatio-temporal data** of dimension TxHxW



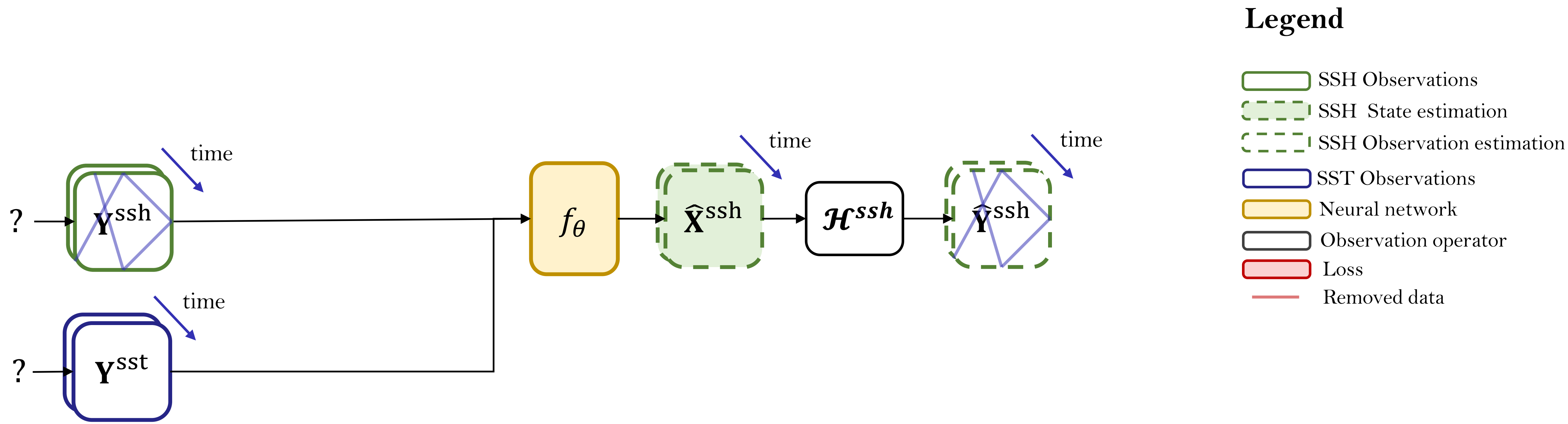



# Unsupervised learning

Given this dataset we can test unsupervised learning methods:

- $\widehat{\mathbf{X}}^{\mathrm{ssh}} = f_\theta\left(\mathbf{Y}^{\mathrm{ssh}}, \mathbf{Y}^{\mathrm{sst}}\right)$
- $\widehat{\mathbf{Y}}^{\mathrm{ssh}} = \mathcal{H}^{\mathrm{ssh}}\left(\widehat{\mathbf{X}}^{\mathrm{ssh}}\right)$

In this setting the all the tensors are **spatio-temporal data** of dimension TxHxW

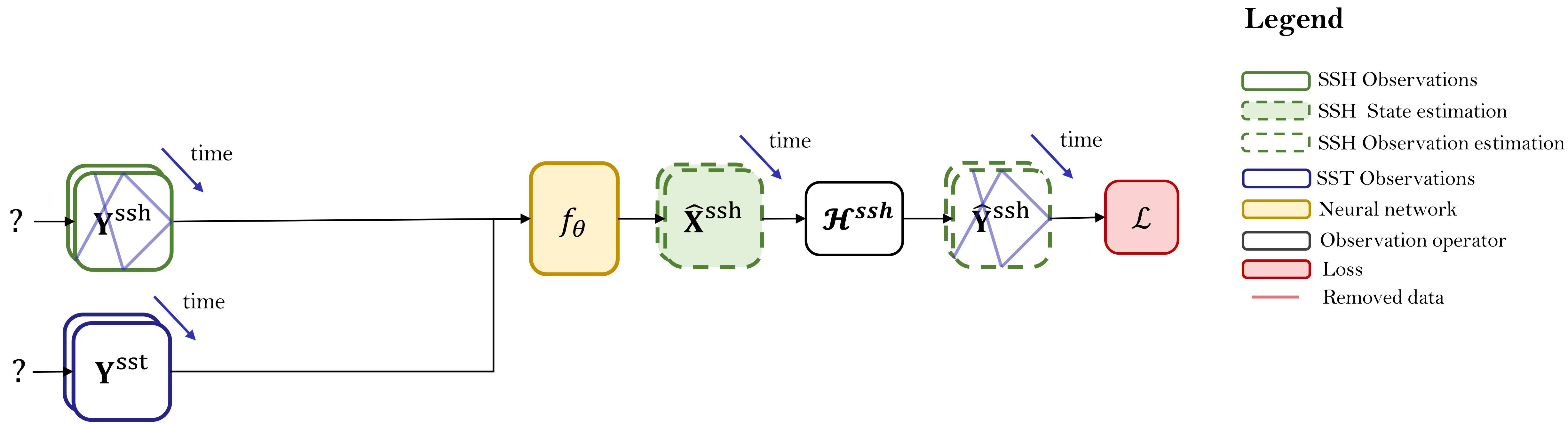


# Unsupervised learning

Given this dataset we can test unsupervised learning methods:

- $\hat{\mathbf{X}}^{\text{ssh}} = f_\theta\left(\mathbf{Y}^{\text{ssh}}, \mathbf{Y}^{\text{sst}}\right)$
- $\hat{\mathbf{Y}}^{\text{ssh}} = \mathcal{H}^{\text{ssh}}\left(\hat{\mathbf{X}}^{\text{ssh}}\right)$
- Compute $\mathcal{L}(\hat{\mathbf{Y}}^{\text{ssh}}, \mathbf{Y}^{\text{ssh}})$

In this setting the all the tensors are **spatio-temporal data** of dimension TxHxW

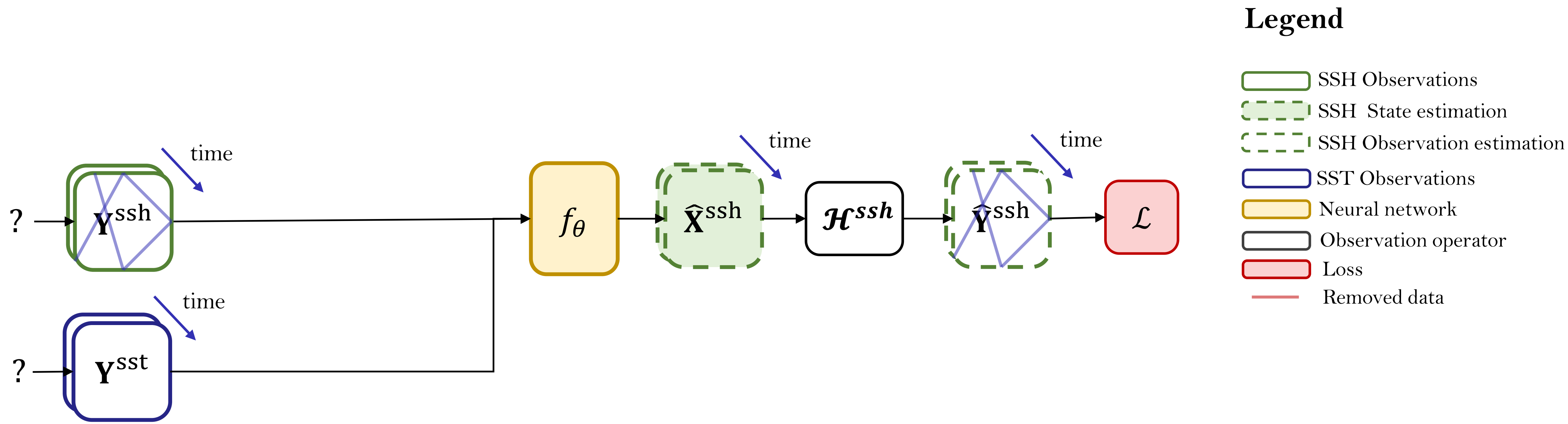


# Unsupervised learning

Given this dataset we can test unsupervised learning methods:

- $\widehat{\mathbf{X}}^{\text{ssh}} = f_\theta\left(\mathbf{Y}^{\text{ssh}}, \mathbf{Y}^{\text{sst}}\right)$
- $\widehat{\mathbf{Y}}^{\text{ssh}} = \mathcal{H}^{\text{ssh}}\left(\widehat{\mathbf{X}}^{\text{ssh}}\right)$
- Compute $\mathcal{L}(\widehat{\mathbf{Y}}^{\text{ssh}}, \mathbf{Y}^{\text{ssh}})$
- Issue: the network is able to copy its input on its output

In this setting the all the tensors are **spatio-temporal data** of dimension TxHxW

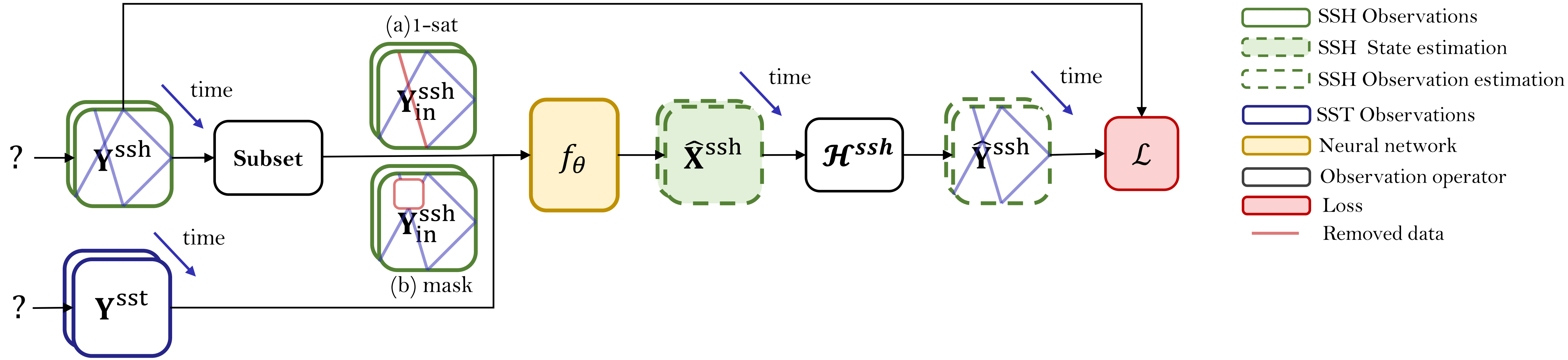


# Unsupervised learning

Given this dataset we can test unsupervised learning methods:

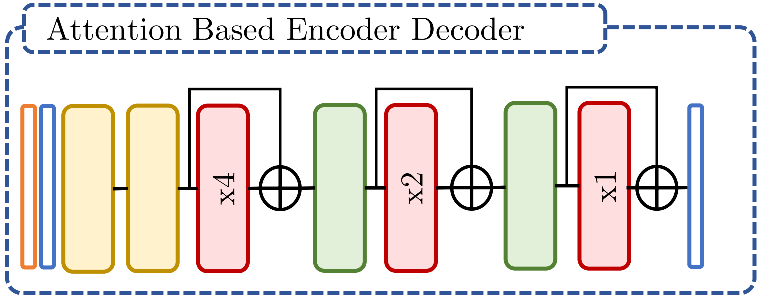
- $\hat{\mathbf{X}}^{\mathrm{ssh}} = f_\theta\left(\mathbf{Y}^{\mathrm{ssh}}, \mathbf{Y}^{\mathrm{sst}}\right)$
- $\hat{\mathbf{Y}}^{\mathrm{ssh}} = \mathcal{H}^{\mathrm{ssh}}\left(\hat{\mathbf{X}}^{\mathrm{ssh}}\right)$
- Compute $\mathcal{L}(\hat{\mathbf{Y}}^{\mathrm{ssh}}, \mathbf{Y}^{\mathrm{ssh}})$
- Issue: the network is able to copy its input on its output
- Solution: remove some observations from the inputs

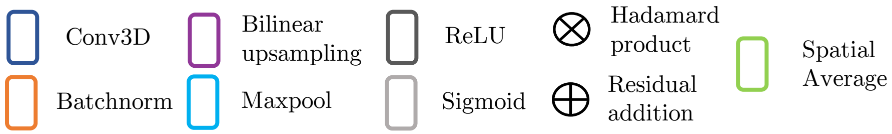
In this setting the all the tensors are **spatio-temporal data** of dimension TxHxW



Attention Based Encoder Decoder (ABED):

Input: 21 daily images of $\mathbf{Y}^{\mathrm{ssh}}$ and $\mathbf{Y}^{\mathrm{sst}}$



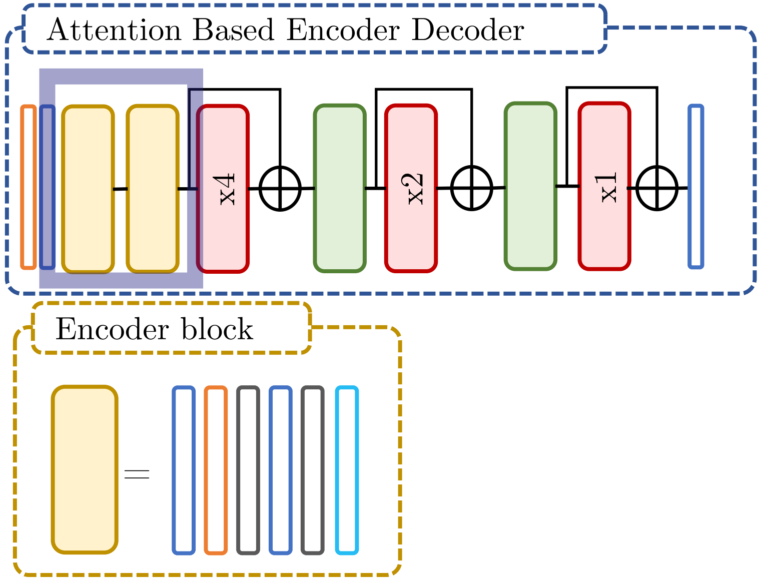



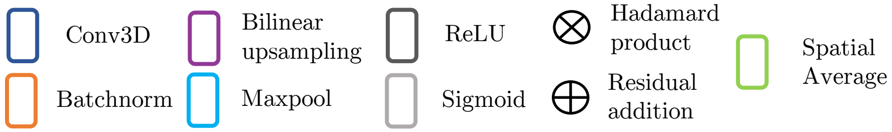
# Architecture

Attention Based Encoder Decoder (ABED):

Input: 21 daily images of $\mathbf{Y}^{\text{ssh}}$ and $\mathbf{Y}^{\text{sst}}$

- Two encoder blocks (dividing spatial dimension by 2)
  - 2 convolutions and a max polling



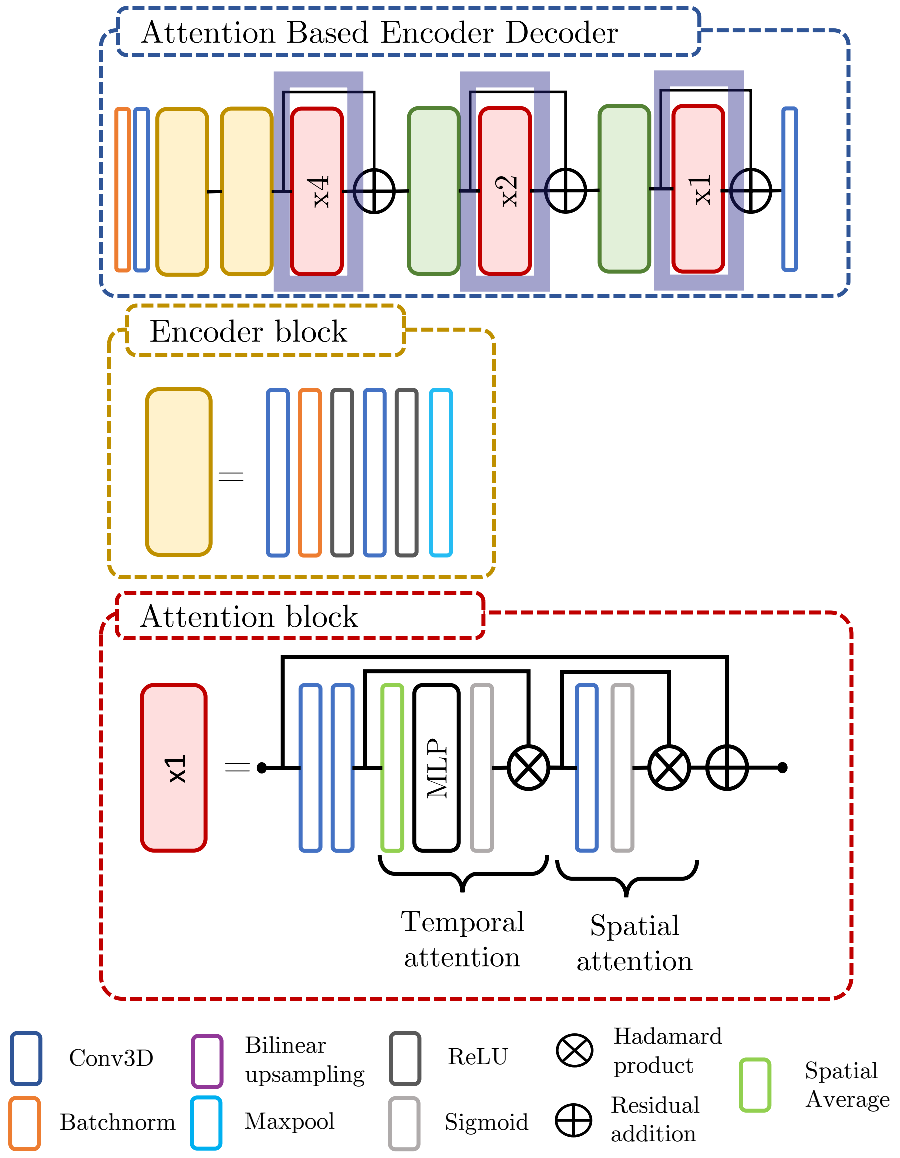



# Architecture

Attention Based Encoder Decoder (ABED):

Input: 21 daily images of $\mathbf{Y}^{ssh}$ and $\mathbf{Y}^{sst}$

- Two encoder blocks (dividing spatial dimension by 2)
  - 2 convolutions and a max polling
- Attention blocks:
  - Principle: multiply the inputs by a factor between 0 and 1
  - Interest: allow to select relevant information and erase irrelevant one

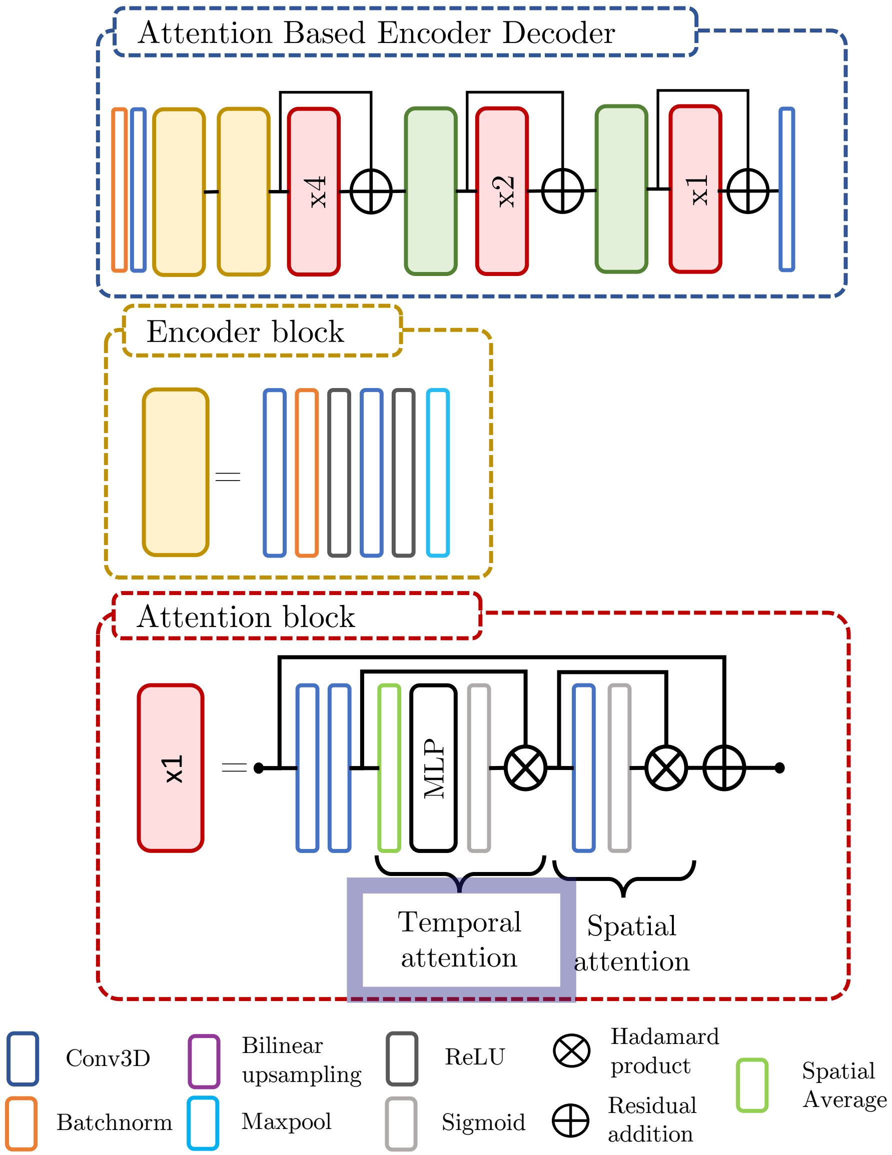


# Architecture

Attention Based Encoder Decoder (ABED):

Input: 21 daily images of $\mathbf{Y}^{ssh}$ and $\mathbf{Y}^{sst}$

- Two encoder blocks (dividing spatial dimension by 2)
  - 2 convolutions and a max polling
- Attention blocks:
  - Principle: multiply the inputs by a factor between 0 and 1
  - Interest: allow to select relevant information and erase irrelevant one
  - Temporal attention: spatial average and then compute channel and temporal attention

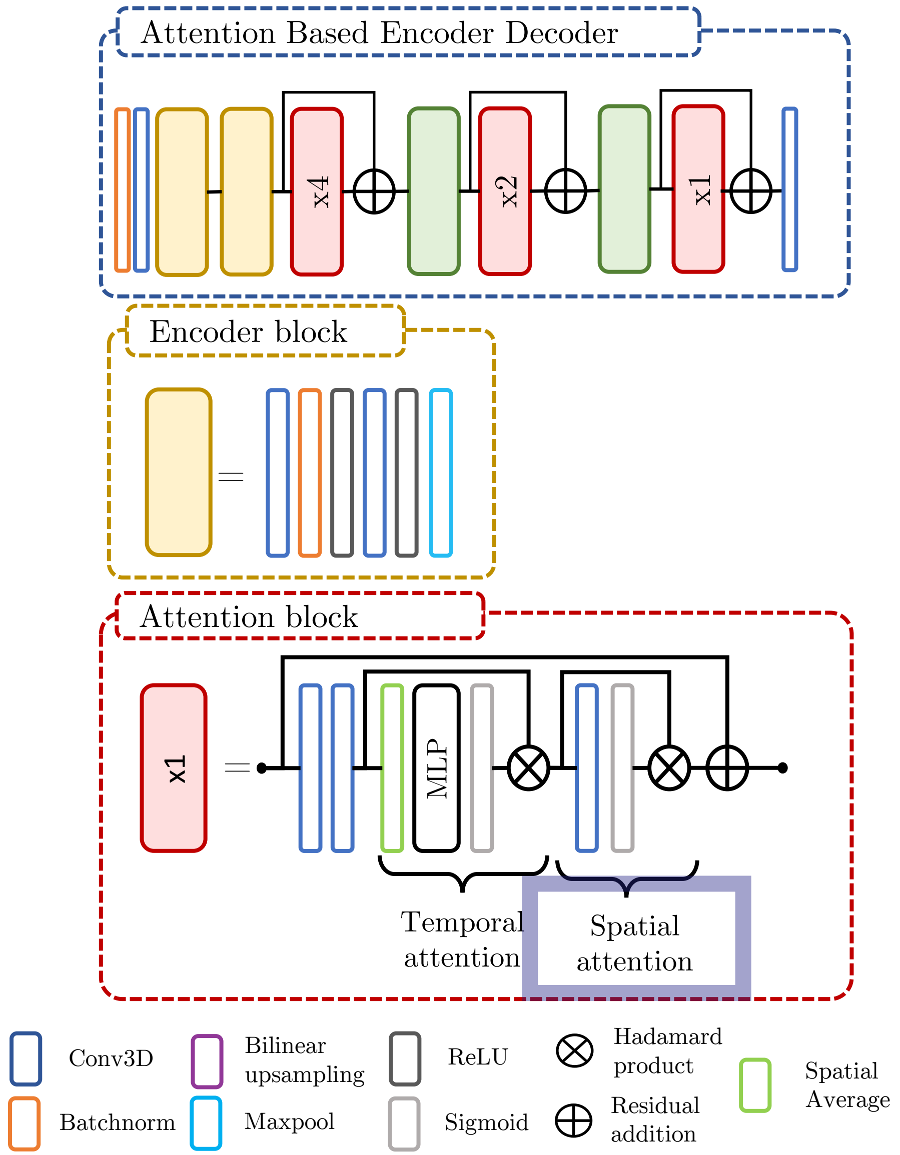


# Architecture

Attention Based Encoder Decoder (ABED):

Input: 21 daily images of $\mathbf{Y}^{ssh}$ and $\mathbf{Y}^{sst}$

- Two encoder blocks (dividing spatial dimension by 2)
    - 2 convolutions and a max polling
- Attention blocks:
    - Principle: multiply the inputs by a factor between 0 and 1
    - Interest: allow to select relevant information and erase irrelevant one
    - Temporal attention: spatial average and then compute channel and temporal attention
    - Spatial attention: 2D convolution in space

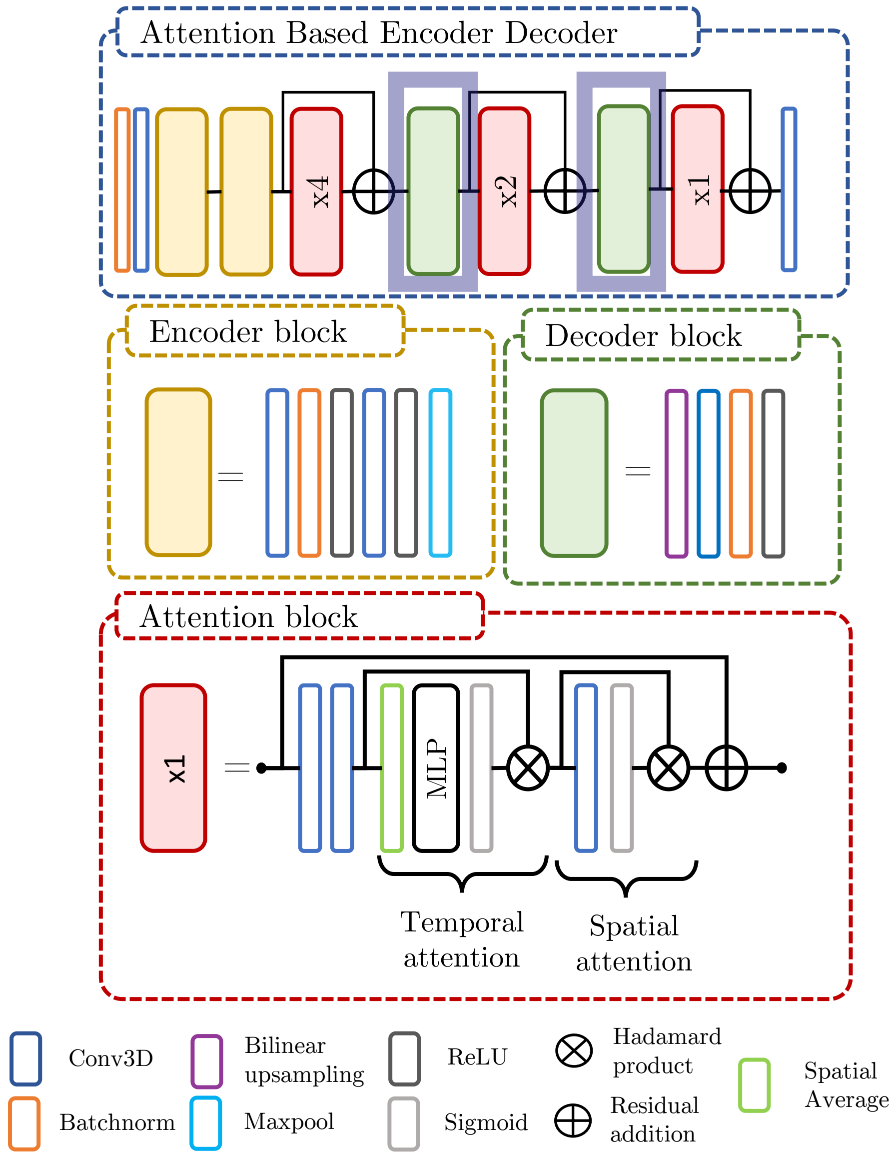


# Architecture

Attention Based Encoder Decoder (ABED):

Input: 21 daily images of $\mathbf{Y}^{ssh}$ and $\mathbf{Y}^{sst}$

- Two encoder blocks (dividing spatial dimension by 2)
  - 2 convolutions and a max polling
- Attention blocks:
  - Principle: multiply the inputs by a factor between 0 and 1
  - Interest: allow to select relevant information and erase irrelevant one
  - Temporal attention: spatial average and then compute channel and temporal attention
  - Spatial attention: 2D convolution in space
- Two decoder blocks (multiplying spatial dimensions by 2)
  - Bilinear up-sampling and convolution



# Experiment on simulation

Delayed-time interpolation

Training methods:

- Supervised
- Unsupervised

# Experiment on simulation

Delayed-time interpolation

Training methods:

- Supervised
- Unsupervised

Input data settings:

- SSH only
- SSH + noisy SST
- SSH + SST

# Experiment on simulation

Delayed-time interpolation

Training methods:

- Supervised
- Unsupervised

Input data settings:

- SSH only
- SSH + noisy SST
- SSH + SST

Training details:

- 3-member ensemble
- Normalization center-reduce

# Experiment on simulation

Delayed-time interpolation

## Training methods:

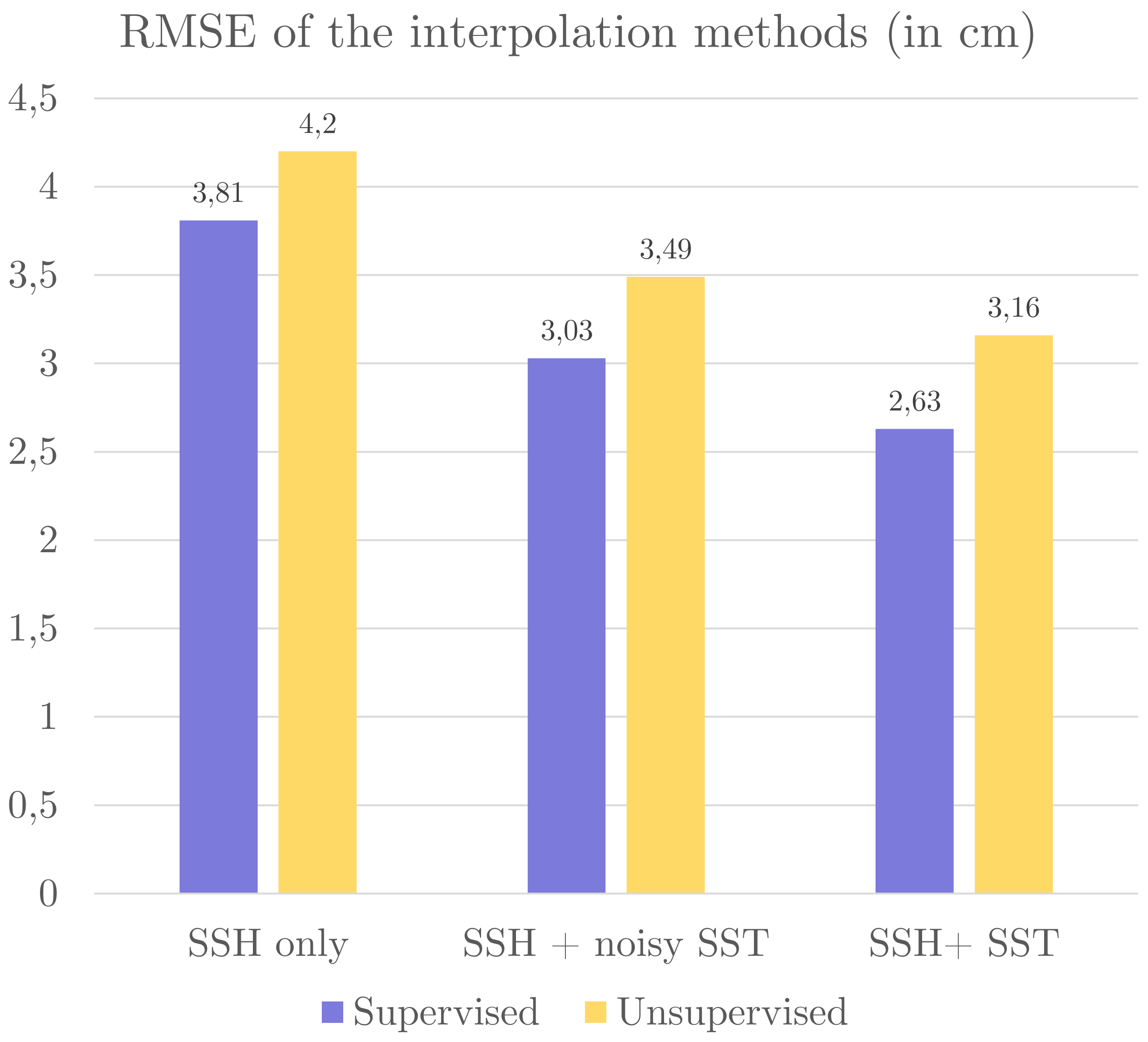
- Supervised
- Unsupervised

## Input data settings:

- SSH only
- SSH + noisy SST
- SSH + SST

## Training details:

- 3-member ensemble
- Normalization center-reduce



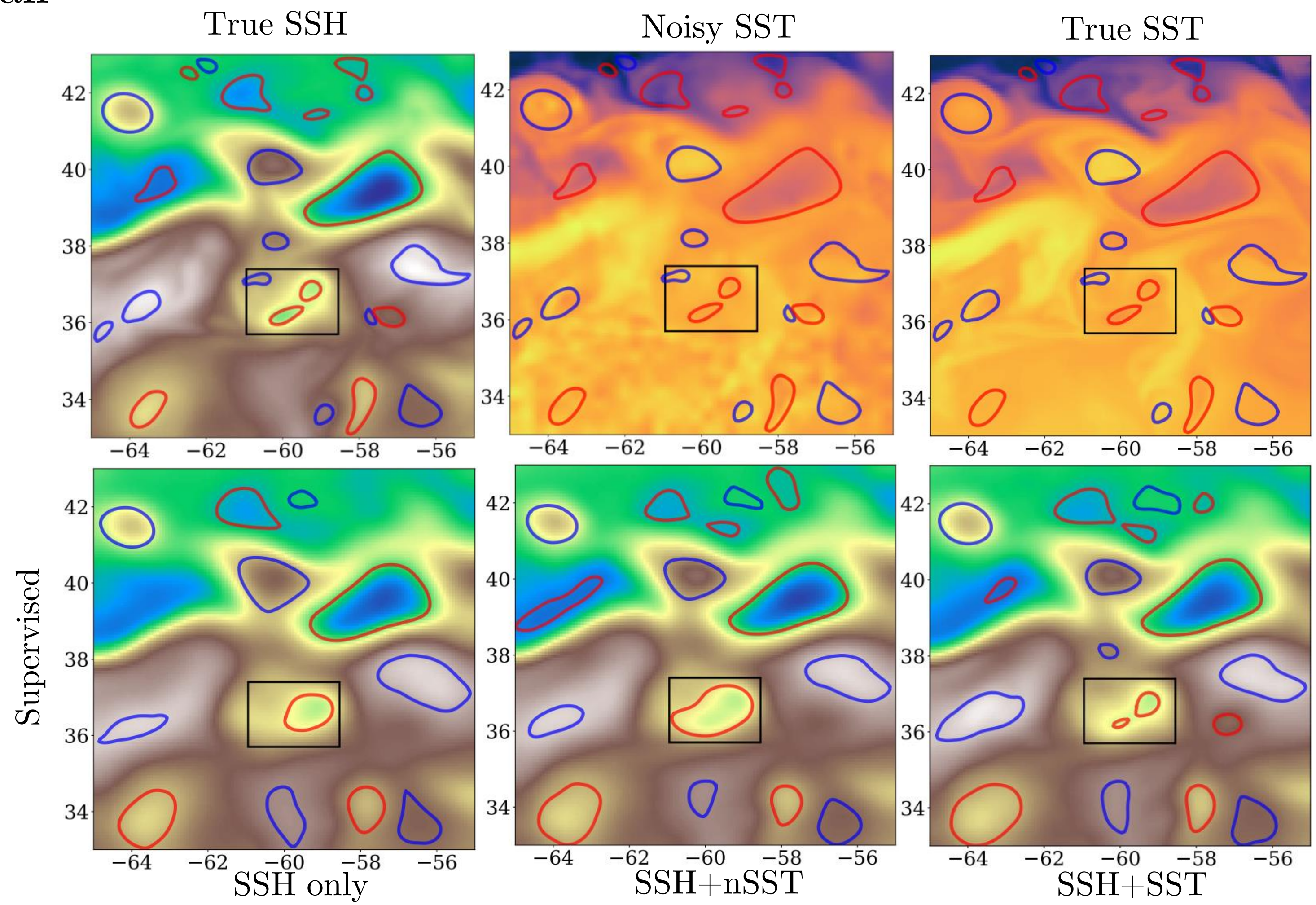

## Conclusions:

- SST still decreases RMSE
- Unsupervised learning is possible but with a drop in performances

# Structure analysis

Does SST helps in reconstructing small ocean structures?

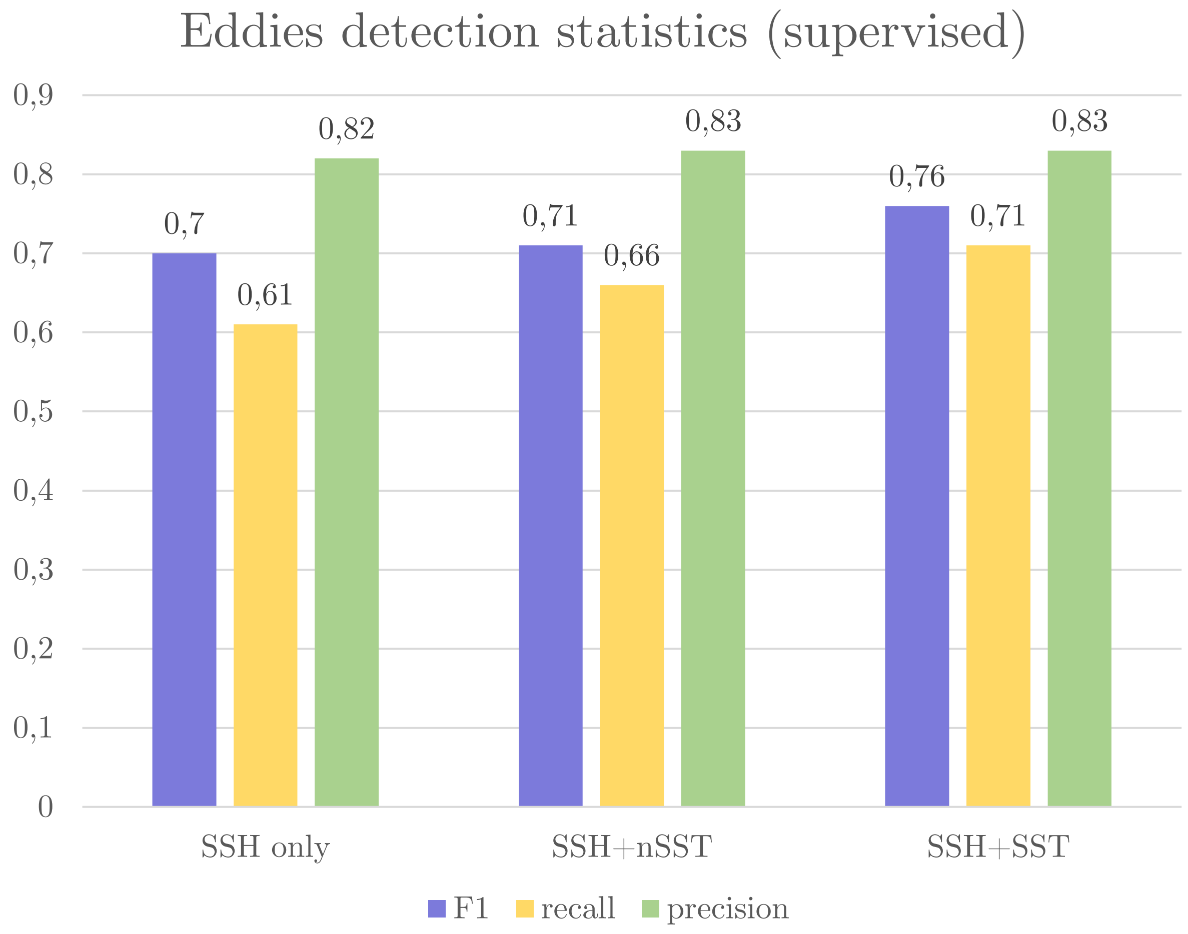
- Automatic detection of eddies with AMEDA on SSH geostrophic currents



# Structure analysis

Does SST helps in reconstructing small ocean structures?

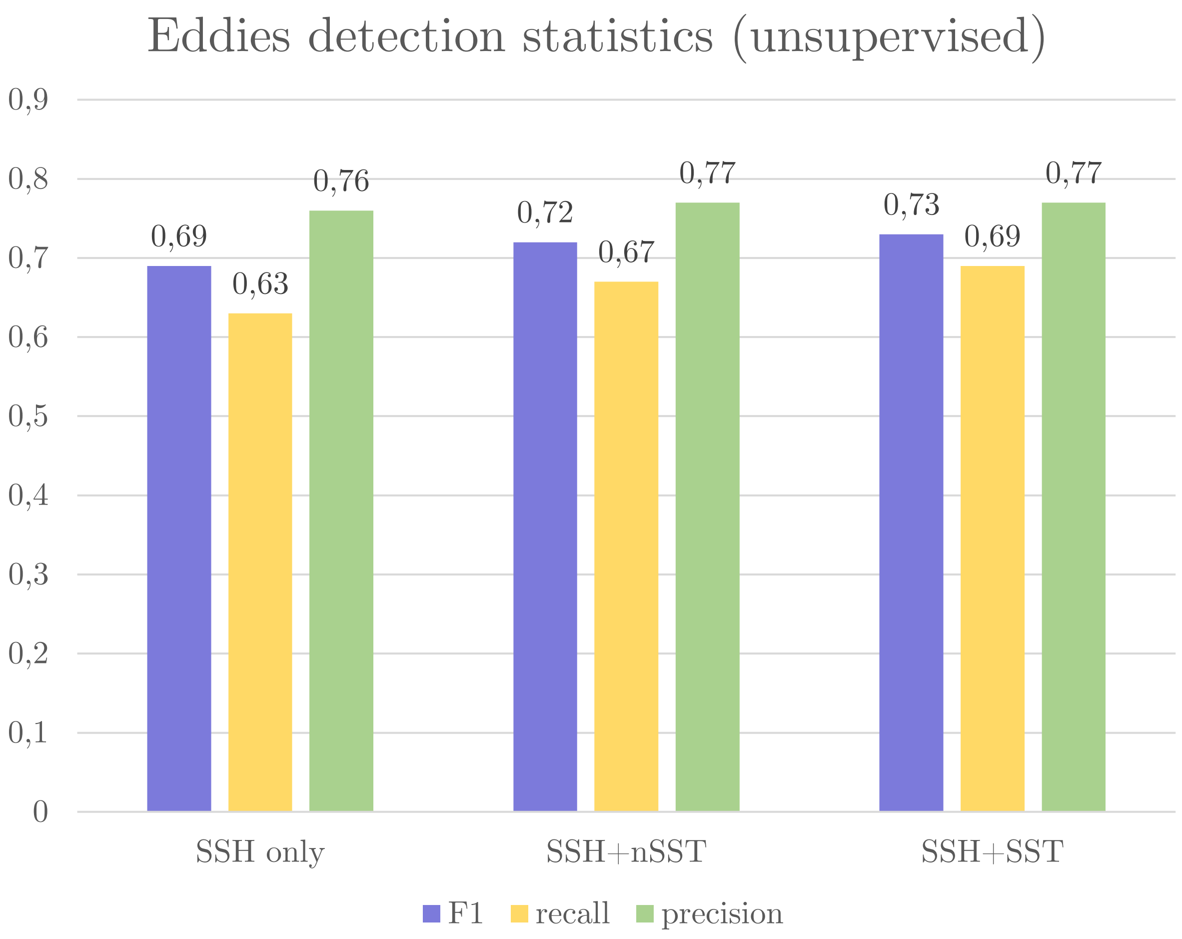
- Automatic detection of eddies with AMEDA on SSH geostrophic currents
- From a statistical point of view:
  - Precision: rate of true positive among the detected eddies
  - Recall: rate of true positive among true eddies
  - F1: harmonic mean of the two scores
- Results:
  - SST helps to recover more eddies



# Structure analysis

Does SST helps in reconstructing small ocean structures?

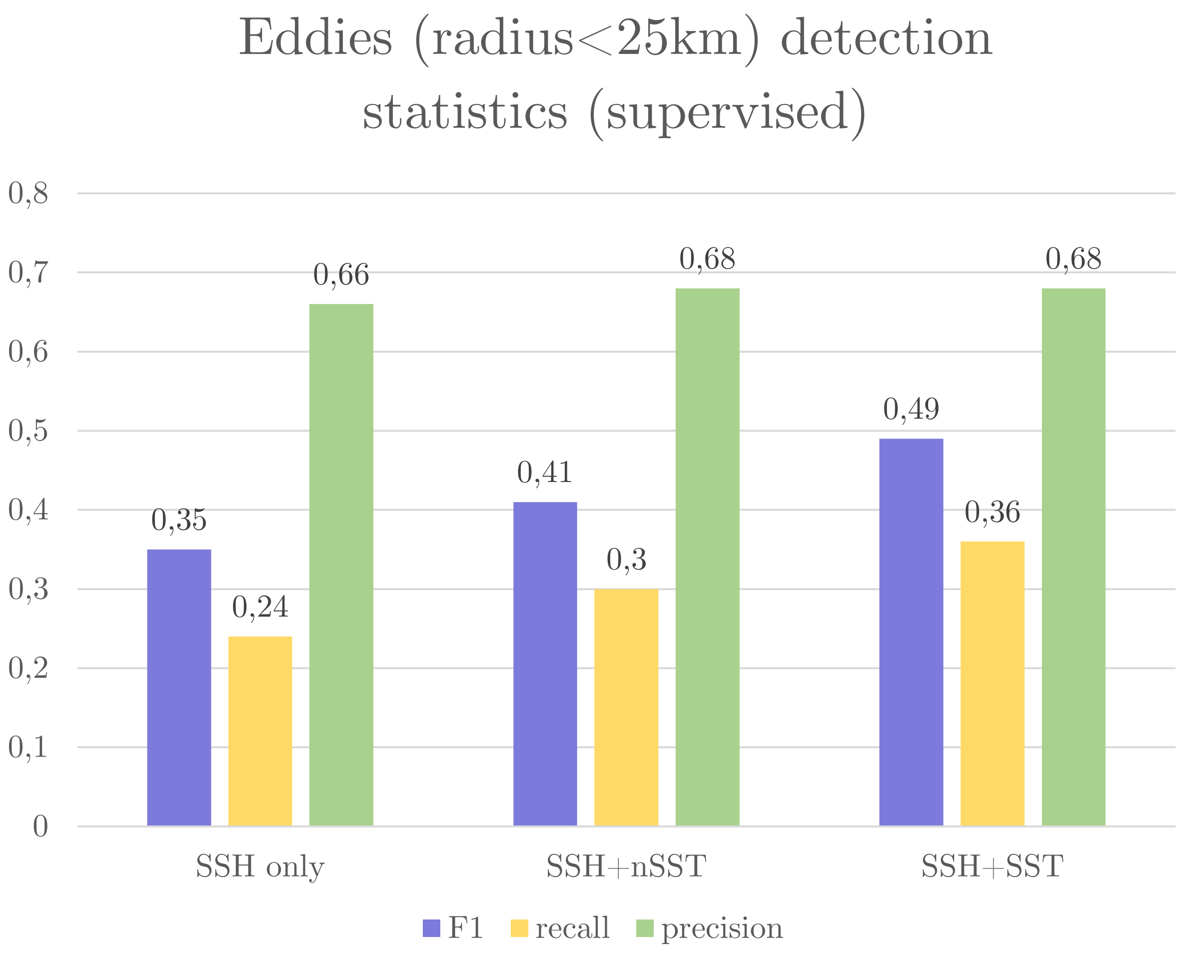
- Automatic detection of eddies with AMEDA on SSH geostrophic currents
- From a statistical point of view:
  - Precision: rate of true positive among the detected eddies
  - Recall: rate of true positive among true eddies
  - F1: harmonic mean of the two scores
- Results:
  - SST helps to recover more eddies
  - It is also the case in the unsupervised case even the detection is worse in every setting



# Structure analysis

Does SST helps in reconstructing small ocean structures?

- Automatic detection of eddies with AMEDA on SSH geostrophic currents
- From a statistical point of view:
  - Precision: rate of true positive among the detected eddies
  - Recall: rate of true positive among true eddies
  - F1: harmonic mean of the two scores
- Results:
  - SST helps to recover more eddies
  - It is also the case in the unsupervised case even the detection is worse in every setting
  - The impact of SST is even more visible when considering the detection of eddies with radius<25km



# Transfer to real observations

How to use and evaluate these methodologies on real observations?

- SSH nadir L3 [**CMEMS, 2021**] and L4 MUR SST [**NASA/JPL, 2019**]

- Combining the OSSE and real observations we can study 3 main strategies:

# Transfer to real observations

How to use and evaluate these methodologies on real observations?

- SSH nadir L3 [**CMEMS, 2021**] and L4 MUR SST [**NASA/JPL, 2019**]

- Combining the OSSE and real observations we can study 3 main strategies:
  - **Simulation-only**: supervised learning and direct application to real data

# Transfer to real observations

How to use and evaluate these methodologies on real observations?

- SSH nadir L3 [**CMEMS, 2021**] and L4 MUR SST [**NASA/JPL, 2019**]

- Combining the OSSE and real observations we can study 3 main strategies:
  - **Simulation-only**: supervised learning and direct application to real data
  - **Observation-only**: unsupervised learning on observations

# Transfer to real observations

How to use and evaluate these methodologies on real observations?

- SSH nadir L3 [**CMEMS, 2021**] and L4 MUR SST [**NASA/JPL, 2019**]

- Combining the OSSE and real observations we can study 3 main strategies:
  - **Simulation-only**: supervised learning and direct application to real data
  - **Observation-only**: unsupervised learning on observations
  - **Simulation and observation**: supervised on simulation and then fine-tuned on observations

  For each we train 3 networks and provide their ensemble estimation

# Transfer to real observations

How to use and evaluate these methodologies on real observations?

- SSH nadir L3 [**CMEMS, 2021**] and L4 MUR SST [**NASA/JPL, 2019**]

- Combining the OSSE and real observations we can study 3 main strategies:
  - **Simulation-only**: supervised learning and direct application to real data
  - **Observation-only**: unsupervised learning on observations
  - **Simulation and observation**: supervised on simulation and then fine-tuned on observations

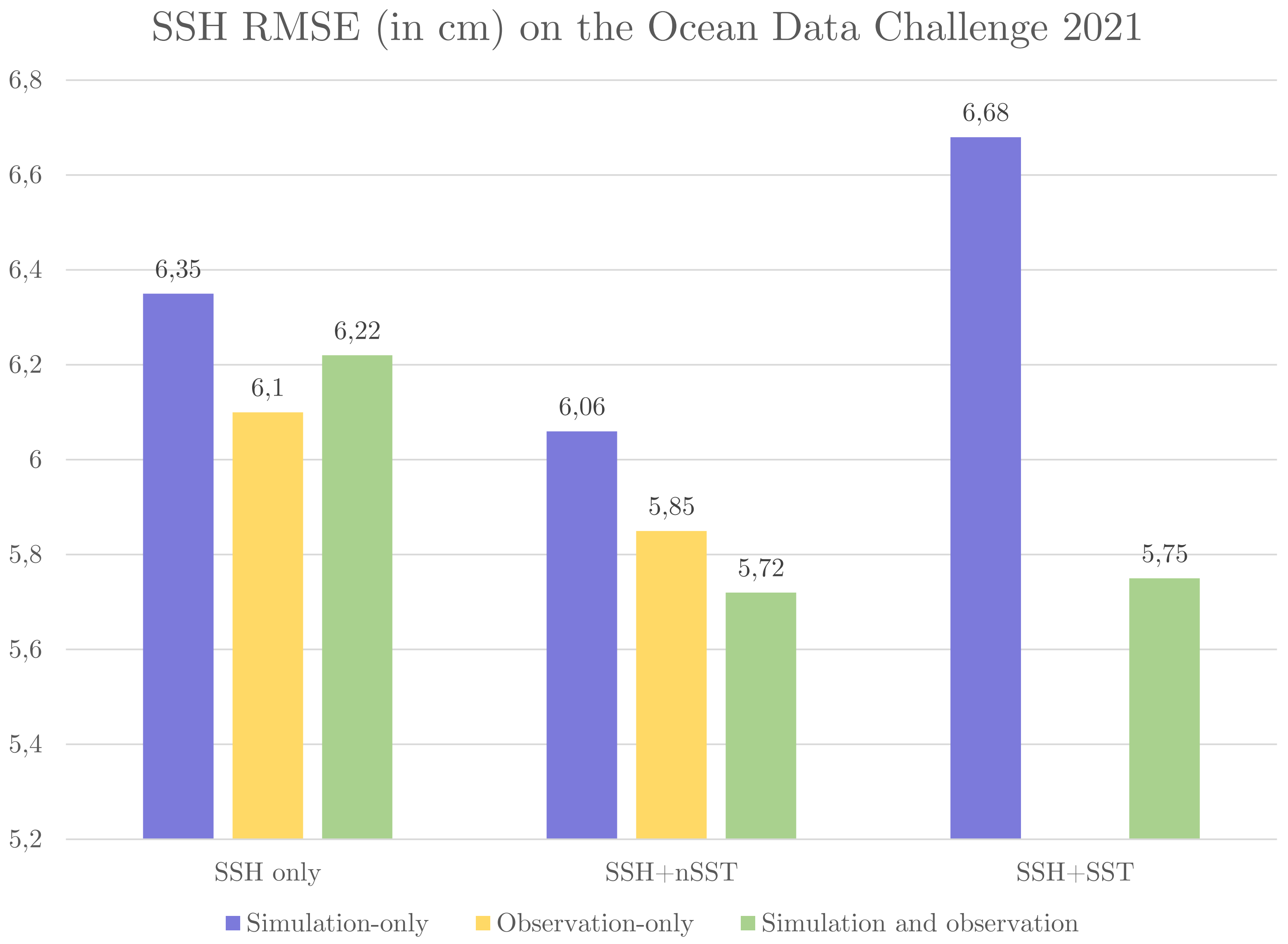
  For each we train 3 networks and provide their ensemble estimation

- Evaluation dataset: Ocean Data Challenge 2021
  - One year of SSH observations in the same Gulf Stream area
  - State-of-the-art reconstruction methods
- Metrics: computed along the satellite tracks

# Transfer to real observations

Results

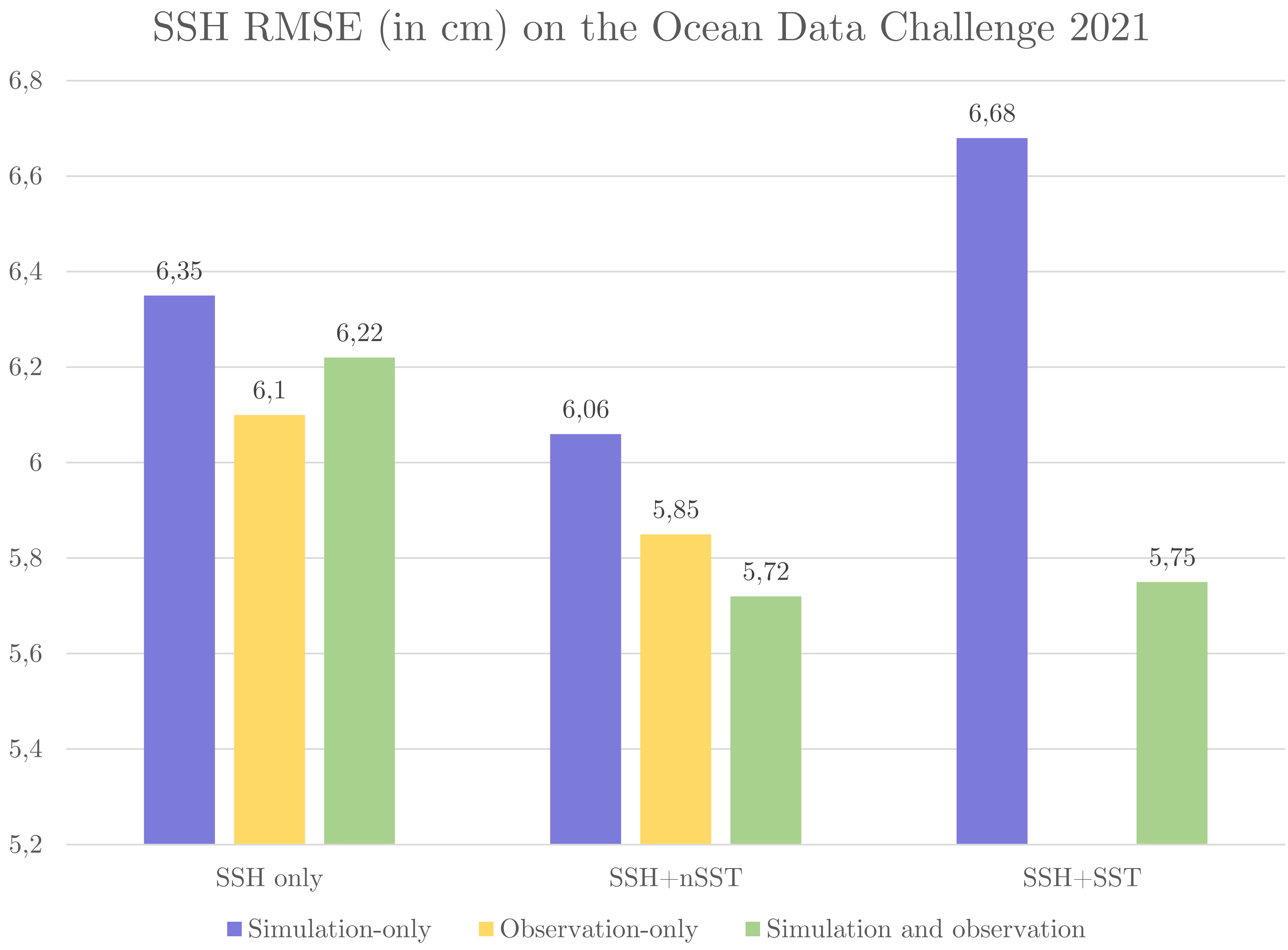
- SST improves the reconstruction
  - Especially after fine-tuning (in green)
  - When training on simulation-only, developing a realistic noise is essential



# Transfer to real observations

Results

- SST improves the reconstruction
  - Especially after fine-tuning (in green)
  - When training on simulation-only, developing a realistic noise is essential
- Pre-training on simulation and fine-tuning on observations seems to be the best training method
  - Interpretation: the supervised training allows to learn important physical relationship on the simulation
  - And the finetuning reduces the domain gap when using SST data



# State of the art comparison (Ocean Data Challenge 2021)

Results

| Method | Source | Type | SST | learning | $\mu$ (cm) | $\sigma_t$ (cm) | $\lambda_x$ (km) |
|---|---|---|---|---|---|---|---|
| **DUACS** | Taburet et al., 2019 | OI | no | no | 7,7 | 2,7 | 149 |
| **DYMOST** | Ubelmann et al., 2016<br>Ballarotta et al., 2020 | OI | no | no | 6,8 | 2,0 | 131 |
| **MIOST** | Ardhuin et al., 2020 | OI | no | no | 6,8 | 2,4 | 139 |
| **DIP** | ours | NN & OI | no | no | 6,5 | 2,3 | 131 |
| **BFN-QG** | Le Guillou et al., 2020 | nudging | no | no | 7,5 | 2,6 | 119 |
| **4DVarNet** | Fablet et al., 2021 | NN | no | simulation | 6,5 | 1,9 | 107 |
| **MUSTI** | ours | NN | yes | observations | 6,3 | 2,0 | 114 |
| **ConvLSTM** | Martin et al., 2023 | NN | no | observations | 6,8 | 1,9 | 114 |
| | | | yes | | 6,3 | **1,6** | 108 |
| **ABED** | ours | NN | no | observations | 6,1 | 1,7 | 111 |
| | | | yes | both | **5,7** | **1,6** | **105** |

**OI: Optimal Interpolation**

**NN: Neural Network**

**Nudging**

$\mu$ (cm): along track RMSE

$\sigma_t$ (cm): temporal standard deviation of $\mu$

$\lambda_x$ (km): half resolved spatial wavelength

# State of the art comparison (Ocean Data Challenge 2021)

Results

- SST-using methods have a clear advantage in all the considered metrics

| Method | Source | Type | SST | learning | $\mu$ (cm) | $\sigma_t$ (cm) | $\lambda_x$ (km) |
|---|---|---|---|---|---|---|---|
| **DUACS** | Taburet et al., 2019 | OI | no | no | 7,7 | 2,7 | 149 |
| **DYMOST** | Ubelmann et al., 2016<br>Ballarotta et al., 2020 | OI | no | no | 6,8 | 2,0 | 131 |
| **MIOST** | Ardhuin et al., 2020 | OI | no | no | 6,8 | 2,4 | 139 |
| **DIP** | ours | NN & OI | no | no | 6,5 | 2,3 | 131 |
| **BFN-QG** | Le Guillou et al., 2020 | nudging | no | no | 7,5 | 2,6 | 119 |
| **4DVarNet** | Fablet et al., 2021 | NN | no | simulation | 6,5 | 1,9 | 107 |
| **MUSTI** | ours | NN | yes | observations | 6,3 | 2,0 | 114 |
| **ConvLSTM** | Martin et al., 2023 | NN | no | observations | 6,8 | 1,9 | 114 |
| | | | yes | | 6,3 | **1,6** | 108 |
| **ABED** | ours | NN | no | observations | 6,1 | 1,7 | 111 |
| | | | yes | both | **5,7** | **1,6** | **105** |

**OI: Optimal Interpolation**

**NN: Neural Network**

**Nudging**

$\mu$ (cm): along track RMSE

$\sigma_t$ (cm): temporal standard deviation of $\mu$

$\lambda_x$ (km): half resolved spatial wavelength

# State of the art comparison (Ocean Data Challenge 2021)

Results

- SST-using methods have a clear advantage in all the considered metrics
- Our method is competitive with SOTA

| Method | Source | Type | SST | learning | $\mu$ (cm) | $\sigma_t$ (cm) | $\lambda_x$ (km) |
|---|---|---|---|---|---|---|---|
| **DUACS** | Taburet et al., 2019 | OI | no | no | 7,7 | 2,7 | 149 |
| **DYMOST** | Ubelmann et al., 2016 Ballarotta et al., 2020 | OI | no | no | 6,8 | 2,0 | 131 |
| **MIOST** | Ardhuin et al., 2020 | OI | no | no | 6,8 | 2,4 | 139 |
| **DIP** | ours | NN & OI | no | no | 6,5 | 2,3 | 131 |
| **BFN-QG** | Le Guillou et al., 2020 | nudging | no | no | 7,5 | 2,6 | 119 |
| **4DVarNet** | Fablet et al., 2021 | NN | no | simulation | 6,5 | 1,9 | 107 |
| **MUSTI** | ours | NN | yes | observations | 6,3 | 2,0 | 114 |
| **ConvLSTM** | Martin et al., 2023 | NN | no | observations | 6,8 | 1,9 | 114 |
| | | | yes | | 6,3 | **1,6** | 108 |
| **ABED** | ours | NN | no | observations | 6,1 | 1,7 | 111 |
| | | | yes | both | **5,7** | **1,6** | **105** |

**OI: Optimal Interpolation**

**NN: Neural Network**

**Nudging**

$\mu$ (cm): along track RMSE

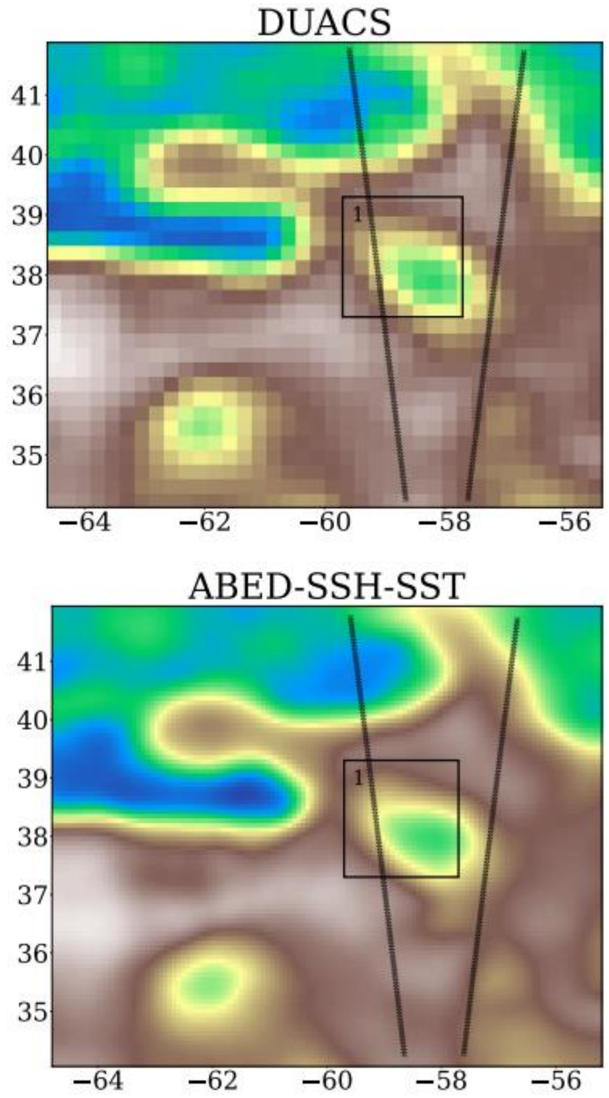
$\sigma_t$ (cm): temporal standard deviation of $\mu$

$\lambda_x$ (km): half resolved spatial wavelength

# Outline

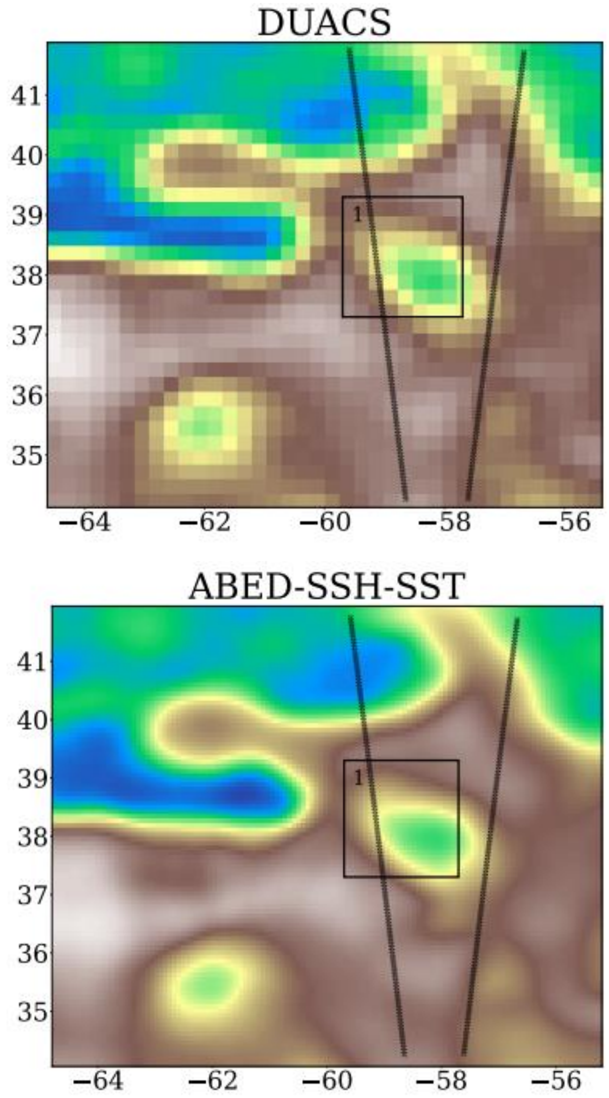
# Conclusions

- SST improves SSH reconstruction in downscaling or interpolation
  - Demonstrated through multiple methodologies
    - SR : RESAC [**Thiria et al., 2023**] and RESACsub [**Archambault et al., 2022**]
    - Interpolation : DIP [**Filoche et al., 2022**], MUSTI [**Archambault et al., 2023**], ABED [**Archambault et al., 2024a, Archambault et al., 2024b**]
  - Evaluated on errors maps and on a physical analysis of eddies



# Conclusions

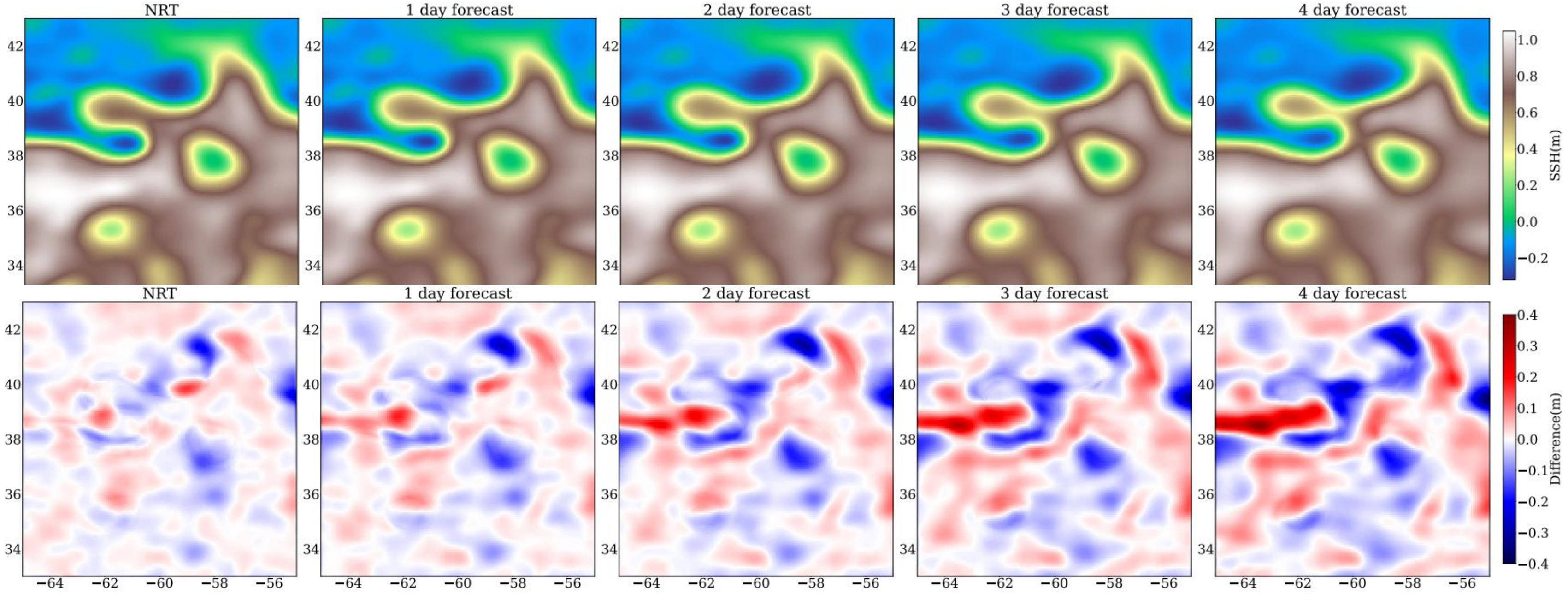
- SST improves SSH reconstruction in downscaling or interpolation
  - Demonstrated through multiple methodologies
    - SR : RESAC [**Thiria et al., 2023**] and RESACsub [**Archambault et al., 2022**]
    - Interpolation : DIP [**Filoche et al., 2022**], MUSTI [**Archambault et al., 2023**], ABED [**Archambault et al., 2024a, Archambault et al., 2024b**]
  - Evaluated on errors maps and on a physical analysis of eddies
- New training approaches
  - Supervised on simulation
  - Unsupervised on observations
  - Hybrid approach: supervised pre-training and unsupervised fine-tuning
    - ABED leads to a RMSE decrease of 26 % compared to DUACS



# Perspectives

**Pushing further toward operational products:**

- Delayed time interpolation → Near real time and forecast
- Single area → global product
  - Global model or several local
  - Include position information




**Other targets or inputs:**

- Including SWOT data
- Including other physical data (Chl-A)

**Beyond regression:**

- Sampling methods
- Uncertainty quantification

# References

[**Amores et al., 2018**] Amores, A., Jordà, G., Arsouze, T., and Le Sommer, J. (2018). Up to what extent can we characterize ocean eddies using present-day gridded altimetric products? Journal of Geophysical Research: Oceans, 123:7220–7236

[**Archambault et al., 2022**] Archambault, T., Charantonis, A., Béréziat, D., and Thiria, S. (2022). Sea surface height super-resolution using high-resolution sea surface temperature with a subpixel convolutional residual network. In Environmental Data Science.

[**Archambault et al., 2023**] Archambault, T., Filoche, A., Charantonis, A., and Béréziat, D. (2023). Multimodal unsupervised spatio-temporal interpolation of satellite ocean altimetry maps. In International Conference on Computer Vision Theory and Applications (VISAPP), Lisboa, Portugal.

[**Archambault et al., 2024a**] Archambault, T., Filoche, A., Charantonis, A., and Béréziat, D. (2024a). Pre-training and fine-tuning attention based encoder decoder improves sea surface height multi-variate inpainting. In VISAPP, Rome, Italy.

[**Archambault et al., 2024b**] Archambault, T., Filoche, A., Charantonis, A., Béréziat, D., and Thiria, S. (2024b). Learning sea surface height interpolation from multi-variate simulated satellite observations. Journal of Advances of Modelling Earth Systems, 16(6).

[**Ardhuin et al., 2020**] Ardhuin, F., Ubelmann, C., Dibarboure, G., Gaultier, L.,Ponte, A., Ballarotta, M., and Faugère, Y. (2020). Reconstructing ocean sur-face current combining altimetry and future spaceborne doppler data. Earth and Space Science Open Archive.

[**Ballarotta et al., 2019**] Ballarotta, M., Ubelmann, C., Pujol, M.-I., Taburet, G.,Fournier, F., Legeais, J.-F., Faugère, Y., Delepoulle, A., Chelton, D., Dibarboure, G., and Picot, N. (2019). On the resolutions of ocean altimetry maps. Ocean Science, 15:1091–1109

[**Ballarotta et al., 2020**] Ballarotta, M., Ubelmann, C., Rogé, M., Fournier, F.,Faugère, Y., Dibarboure, G., Morrow, R., and Picot, N. (2020). Dynamic mapping of along-track ocean altimetry: Performance from real observations. Journal of Atmospheric and Oceanic Technology, 37:1593–1601.

# References

[**Fablet et al., 2021**] Fablet, R., Amar, M., Febvre, Q., Beauchamp, M., and Chapron,B. (2021). End-to-end physics-informed representation learning for satellite ocean remote sensing data: Applications to satellite altimetry and sea surface currents. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 5:295–302

[**Fablet et al., 2023**] Fablet, R., Febvre, Q., and Chapron, B. (2023). Multimodal4DVarNets for the reconstruction of sea surface dynamics from SST-SSH synergies. IEEE Transactions on Geoscience and Remote Sensing, 61.

[**Filoche et al., 2022**] Filoche, A., Archambault, T., Charantonis, A., and Béréziat,D. (2022). Statistics-free interpolation of ocean observations with deep spatio-temporal prior. In ECML/PKDD Workshop on Machine Learning for Earth Observation and Prediction (MACLEAN).

[**Le Guillou et al., 2020**] Le Guillou, F., Metref, S., Cosme, E., Ubelmann, C., Bal-larotta, M., Verron, J., and Le Sommer, J. (2020). Mapping altimetry in the forthcoming SWOT era by back-and-forth nudging a one-layer quasi-geostrophic model. Earth and Space Science Open Archive.

[**Martin et al., 2023**] Martin, S. A., Manucharyan, G. E., and Klein, P. (2023). Synthesizing sea surface temperature and satellite altimetry observations using learning improves the accuracy and resolution of gridded sea surface height anomalies. Journal of Advances in Modeling Earth Systems, 15(5)

[**Ubelmann et al., 2016**] Ubelmann, C., Cornuelle, B., and Fu, L. (2016). Dynamic mapping of along-track ocean altimetry: Method and performance from observing system simulation experiments. Journal of Atmospheric and Oceanic Technology,33:1691–1699.

[**Stegner et al., 2021**] Stegner, A., Le Vu, B., Dumas, F., Ghannami, M., Nicolle, A.,Durand, C., and Faugere, Y. (2021). Cyclone-anticyclone asymmetry of eddy detection on gridded altimetry product in the mediterranean sea. Journal of Geophysical Research: Oceans, 126

[**Taburet et al., 2019]** Taburet, G., Sanchez-Roman, A., Ballarotta, M., Pujol, M.-I., Legeais, J.-F., Fournier, F., Faugere, Y., and Dibarboure, G. (2019). DUACSDT2018: 25 years of reprocessed sea level altimetry products. Ocean Sci, 15:1207–1224

[**Thiria et al., 2023**] Thiria, S., Sorror, C., Archambault, T., Charantonis, A.,Béréziat, D., Mejia, C., Molines, J.-M., and Crepon, M. (2023). Downscaling of ocean fields by fusion of heterogeneous observations using deep learning algorithms. Ocean Modeling, 182.

# Data


[**CMEMS, 2021**] CMEMS (2021). Global ocean along-track L3 sea surface heights reprocessed (1993-ongoing) tailored for data assimilation [dataset]. https://doi.org/10.48670/MOI-00146.

[**CMEMS, 2023a**] CMEMS (2023a). Global ocean gridded L4 sea surface heights and derived variables reprocessed 1993 ongoing. https://doi.org/10.48670/moi-00148.

[**CMEMS, 2023b**] CMEMS (2023b). Global ocean ostia sea surface temperature and sea ice analysis [dataset]. https://doi.org/10.48670/moi-00168.

[**CMEMS, 2023c**] CMEMS (2023c). Global oceans sea surface temperature multi-sensor L3 observations [dataset]. https://doi.org/10.48670/MOI-00164.

[**NASA/JPL, 2019**] NASA/JPL (2019). GHRSST level 4 MUR 0.25 deg global foundation sea surface temperature analysis (v4.2) [dataset]. https://doi.org/10.5067/GHGMR-4FJ04.

[**NASA/JPL and CNES, 2024**] NASA/JPL and CNES (2024). The SWOT_L3_LR_SSH product, derived from the L2 SWOT KaRIn low rate ocean data products [dataset]. https://doi.org/10.24400/527896/A01-2023.018.

# Supplementary materials

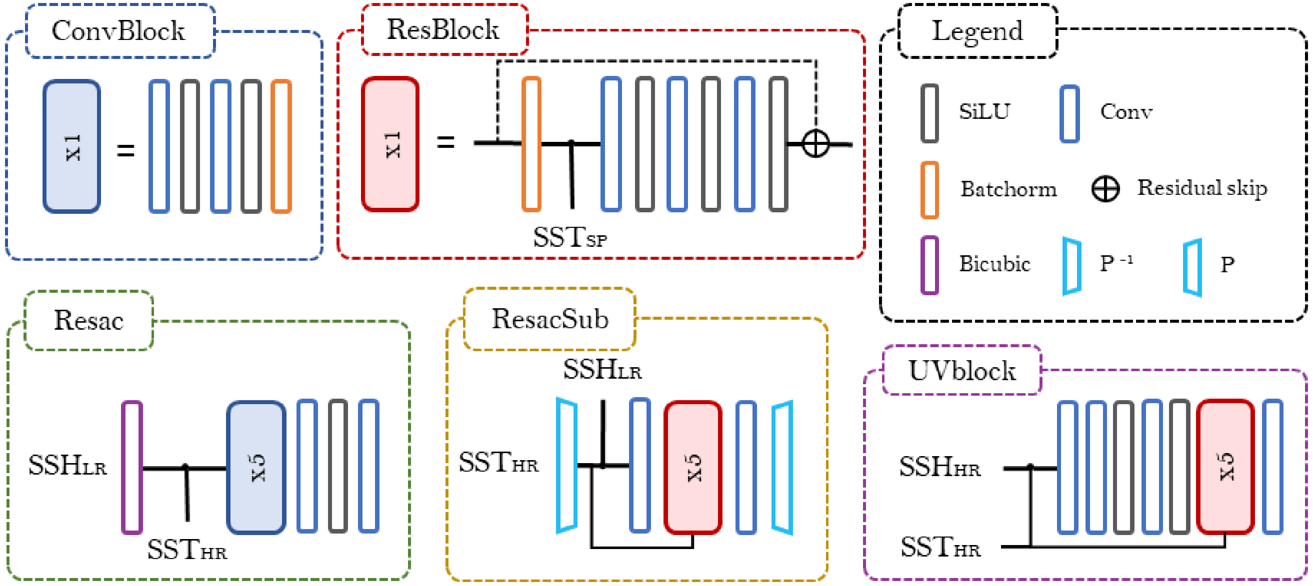
1. RESAC and RESAC subpixel comparison
2. Deep Image Prior principle and application to SSH interpolation
3. Adapting DIP to use SST: MUSTI
4. Details on State-Of-The-Art methods

# RESAC and RESACsub

- Two RESAC up-sampling architecture:
- RESACsub:
- Subpixel convolution
  - Up-sampling at the end of the network
  - Computationally efficient
  - Trainable
  - Introduce checkerboard artifacts
- Residual learning
  - Add the input to the output
  - Helps solving the vanishing gradient problem

RESAC:

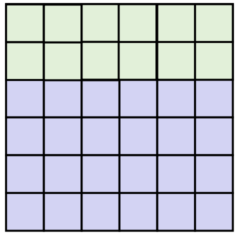
- Standard convolution
  - Bicubic up-sampling at the beginning of the network
  - Computationally efficient
- No residual learning
  - Hard to train very deep networks, which explains the multiscale loss function



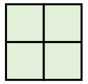
# RESAC and RESACsub

- Computational advantage of subpixel convolution:
  - For more parameters in the filter, we have the same number of multiplications

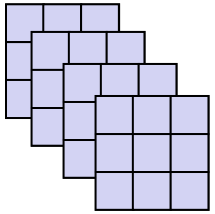
HR image



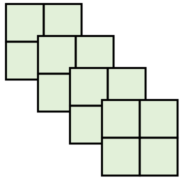
2x2 filter



LR image
4×H/2×W/2



4x2x2 filter



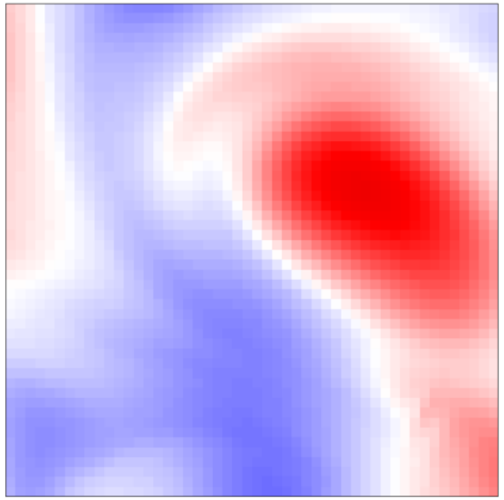
Number of multiplications:
4+4+4+4+4

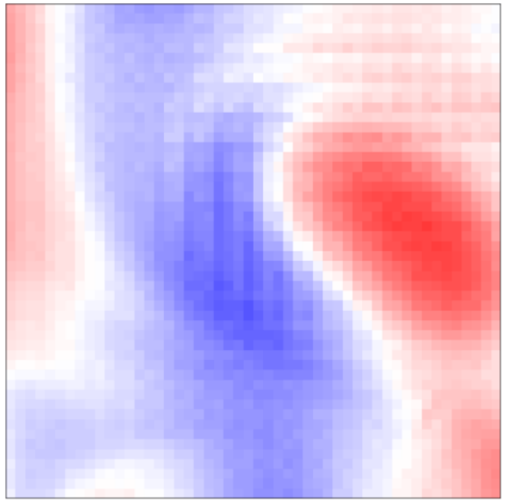
In total: 4×(W)×(H)

Number of multiplications:

In total:
4×4×(W/2)×(H/2)

- Checkerboard artifacts:
  - Each small image is generated with a different kernel
  - When we shuffle the pixels, spatial neighbor are generated with independent kernel
  - To reduce these artifacts, another method, not tested in the thesis is the ICnR initialization [**Aitken, et al., 2017**]



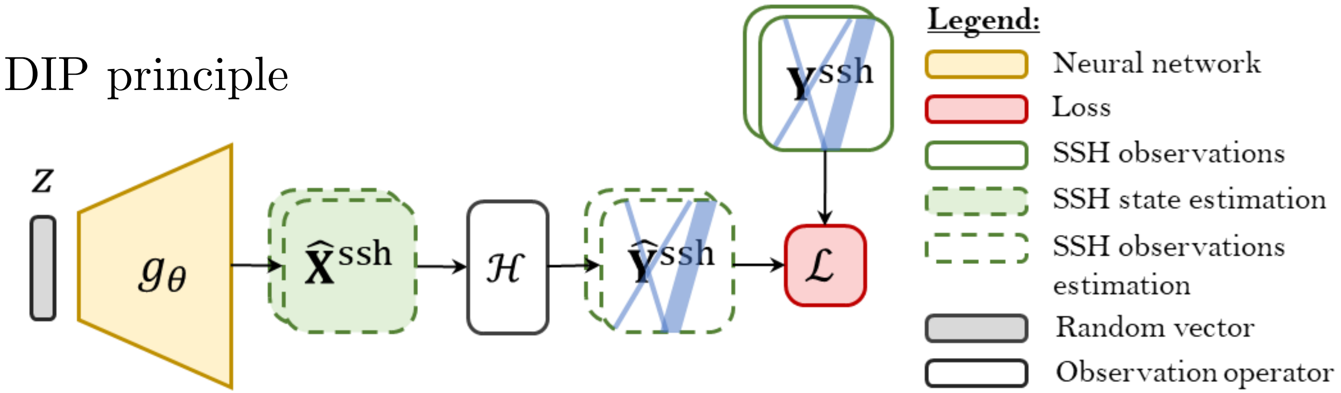


# DIP

Deep Image Prior (DIP) [**Ulyanov et al., 2017**] :

- Use neural architecture as a prior on the output distribution
- Optimization of a neural network on a single example of observations

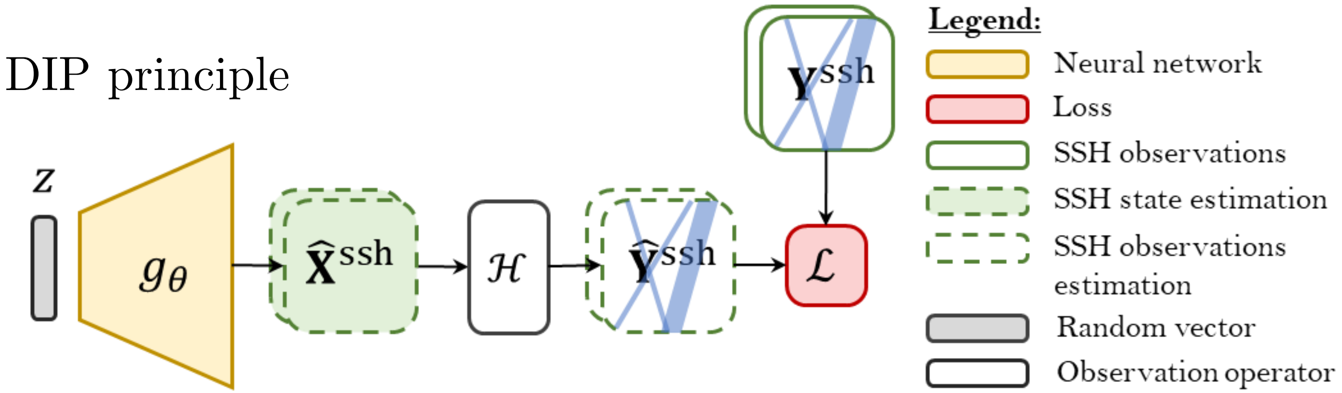
DIP principle



# DIP

Deep Image Prior (DIP) [**Ulyanov et al., 2017**] :

- Use neural architecture as a prior on the output distribution
- Optimization of a neural network on a single example of observations



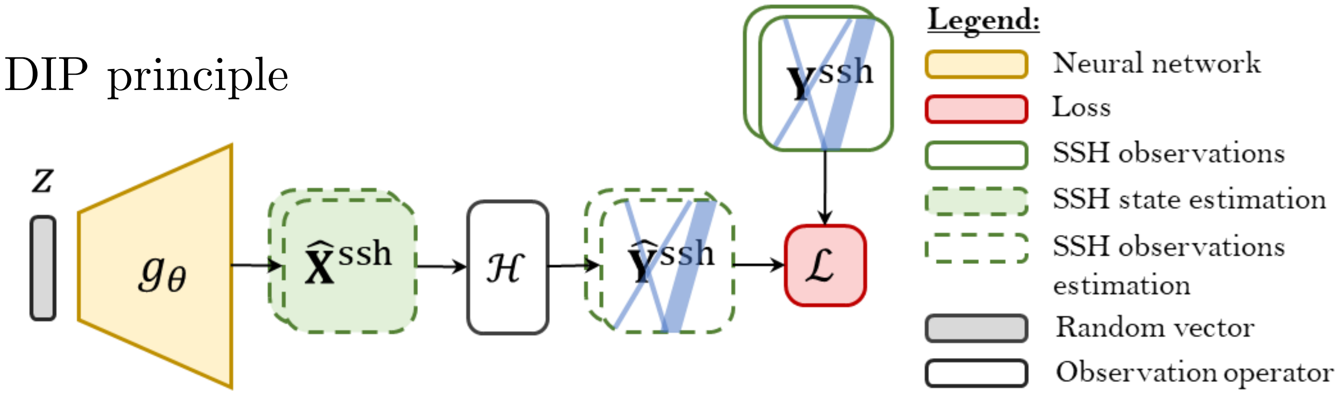

Example taken from [**Ulyanov et al., 2017**] Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Deep image prior. International Journal of Computer Vision, 128:1867–1888. Paper: https://arxiv.org/abs/1711.10925

# DIP

Deep Image Prior (DIP) [**Ulyanov et al., 2017**] :

- Use neural architecture as a prior on the output distribution
- Optimization of a neural network on a single example of observations

DIP principle




Algorithm:

1. Starts from random vector: Z
2. State estimation: $\widehat{\mathbf{X}}^{ssh} = g_\theta(Z)$
3. Apply Observing operator: $\widehat{\mathbf{Y}}^{ssh} = \mathcal{H}(\widehat{\mathbf{X}}^{ssh})$
4. Compute loss: $\mathcal{L}(\widehat{\mathbf{Y}}^{ssh}, \mathbf{Y}^{ssh})$

# DIP

Deep Image Prior (DIP) [**Ulyanov et al., 2017**] :

- Use neural architecture as a prior on the output distribution
- Optimization of a neural network on a single example of observations

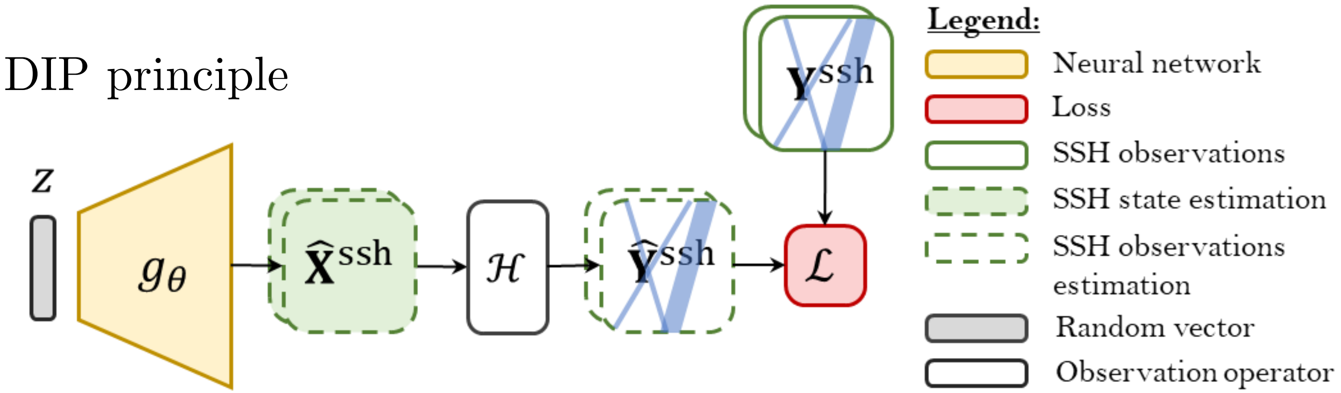
Algorithm:

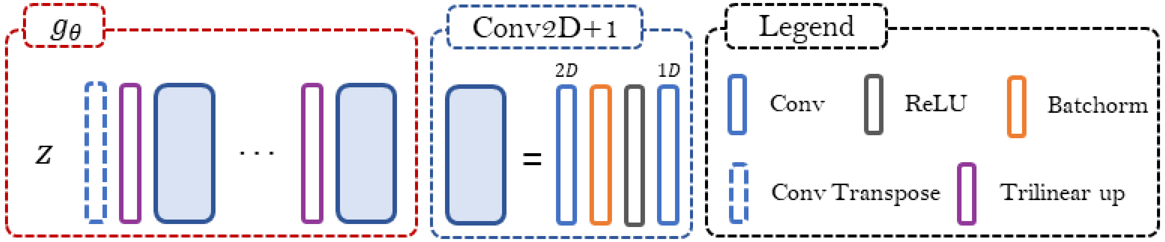
1. Starts from random vector: Z
2. State estimation: $\widehat{\mathbf{X}}^{ssh} = g_\theta(Z)$
3. Apply Observing operator: $\widehat{\mathbf{Y}}^{ssh} = \mathcal{H}(\widehat{\mathbf{X}}^{ssh})$
4. Compute loss: $\mathcal{L}(\widehat{\mathbf{Y}}^{ssh}, \mathbf{Y}^{ssh})$

Why does it works?

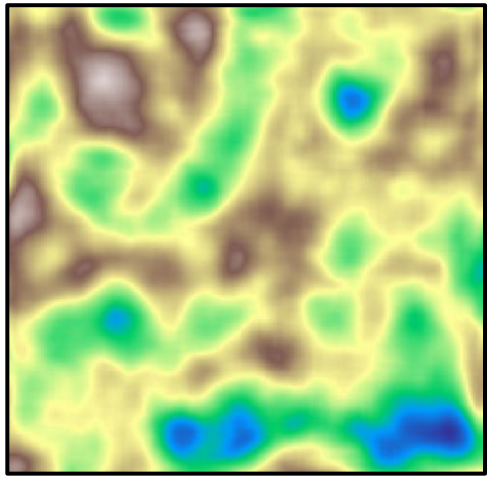
- The operations performed by the network (convolutions) have invariance properties
- The output of a neural network is spatially and temporally correlated

DIP principle




Output of an untrained neural network



# MUSTI

How to Adapt DIP idea to use the SST?

- Instead of starting from Z we can start from a SST image
- Optimize on a single example or a small number of examples

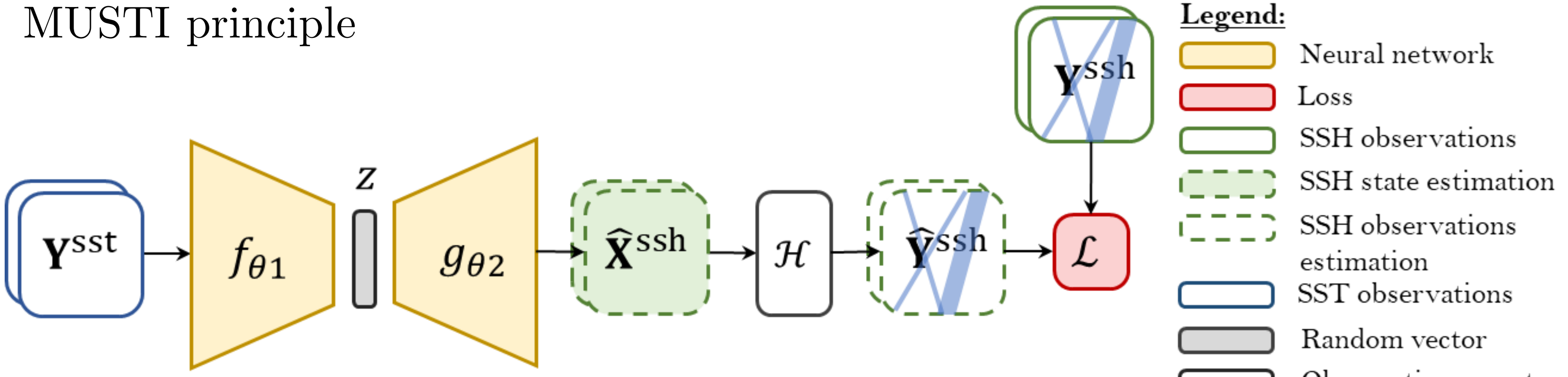
Algorithm:

1. Starts from SST obs: $\mathbf{Y}^{sst}$
2. Encode SST image: $Z = f_{\theta_1}(\mathbf{Y}^{sst})$
3. State estimation: $\widehat{\mathbf{X}}^{ssh} = g_{\theta_2}(Z)$
4. Apply Observing operator: $\widehat{\mathbf{Y}}^{ssh} = \mathcal{H}(\widehat{\mathbf{X}}^{ssh})$
5. Compute loss: $\mathcal{L}(\widehat{\mathbf{Y}}^{ssh}, \mathbf{Y}^{ssh})$

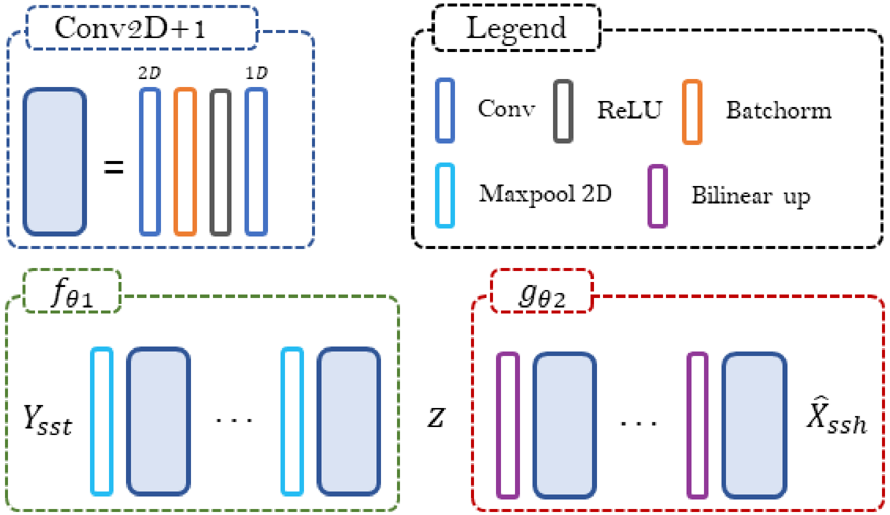
Why does it works?

- Invariances properties
  - The network transforms a SST image into a SSH image, being supervised only on observations
  - This transformation is the same on the entire image

MUSTI principle




MUSTI architecture

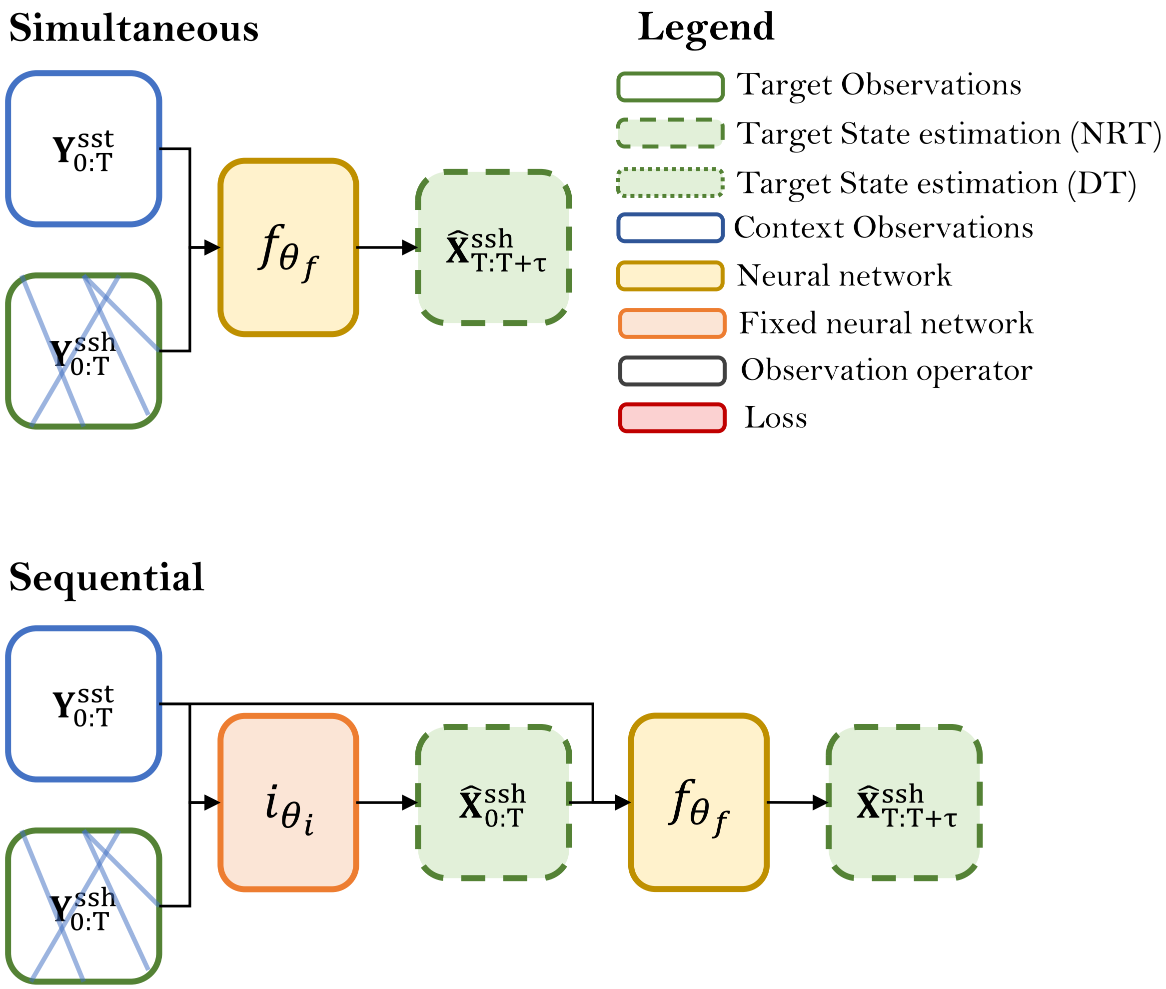


# Forecast

Goal: Estimate present and future SSH from past observations


Two ways to perform the forecast and interpolation:

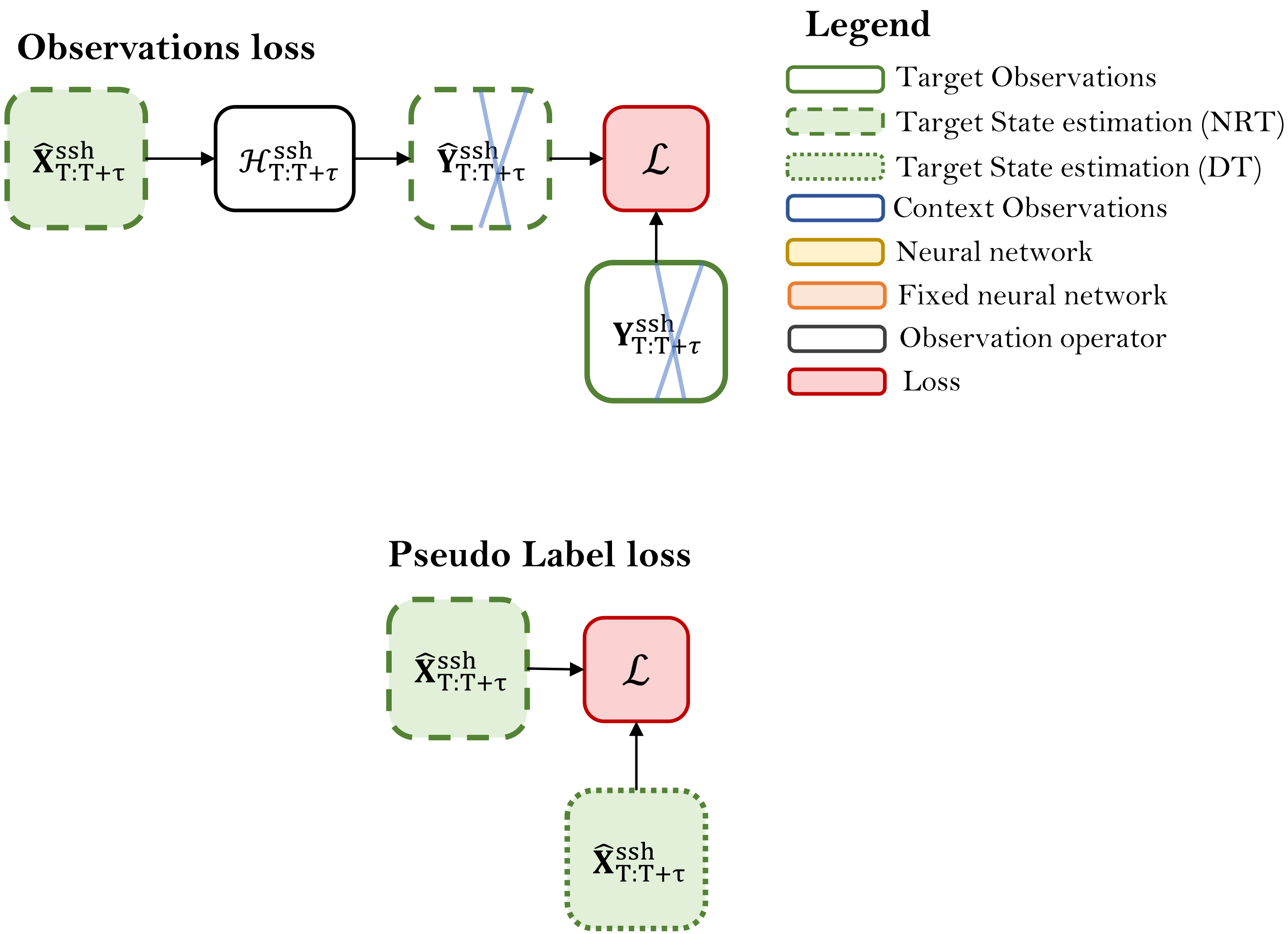
- Simultaneously: a single neural network performs the two tasks
  - Inputs: incomplete SSH and SST between 0 to T
  - Outputs: gridded and forecasted SSH between T and T+$\tau$
- Sequential:
  - A first network is used to interpolate SSH between 0 and T
  - A second network performs the forecast from SST and SSH



# Forecast

How to control the forecast? Two methods:

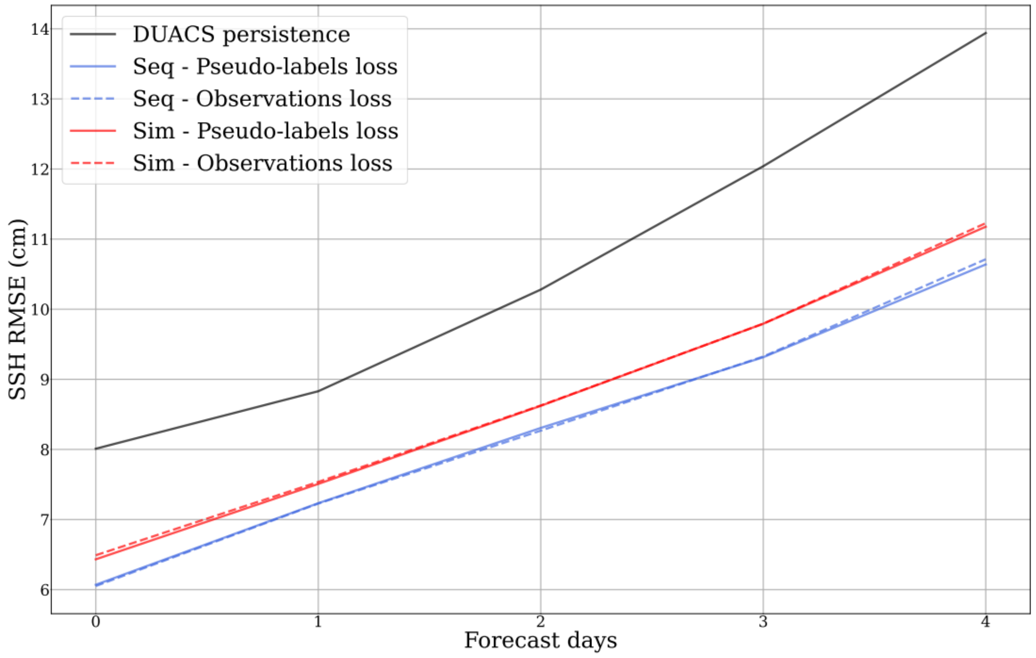
- Observations loss
  - Has in Delayed Time interpolation we compute MSE on satellite data
  - No need to remove any observations from the days that are not given in input
- Pseudo-Label loss:
  - Pseudo labels are generated using a delayed time neural network
  - We supervised the forecast network with it



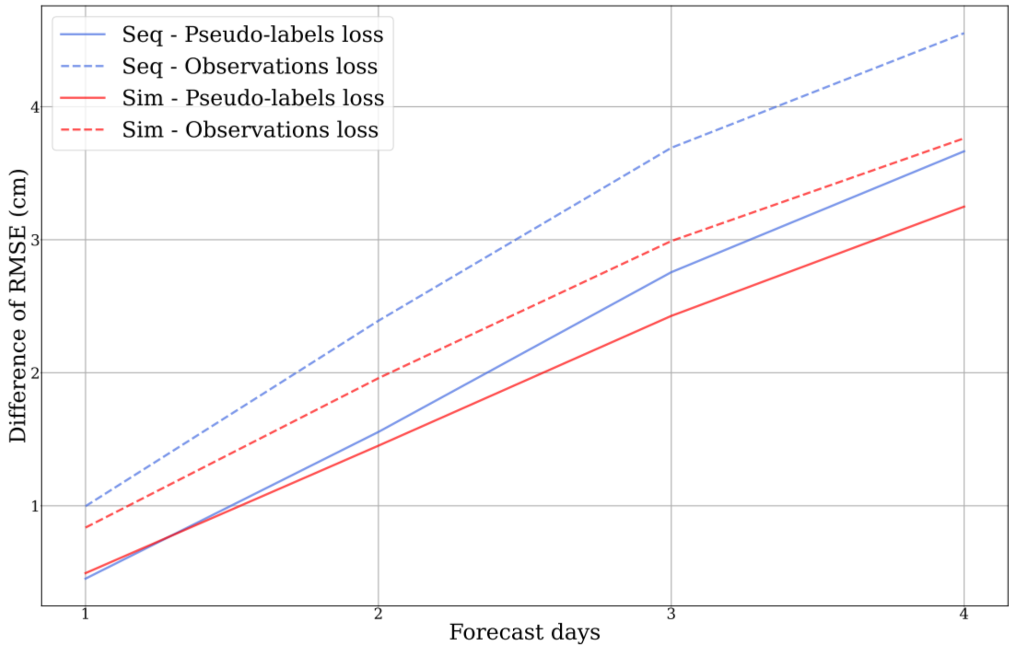
# Forecast

Results:

Forecast errors

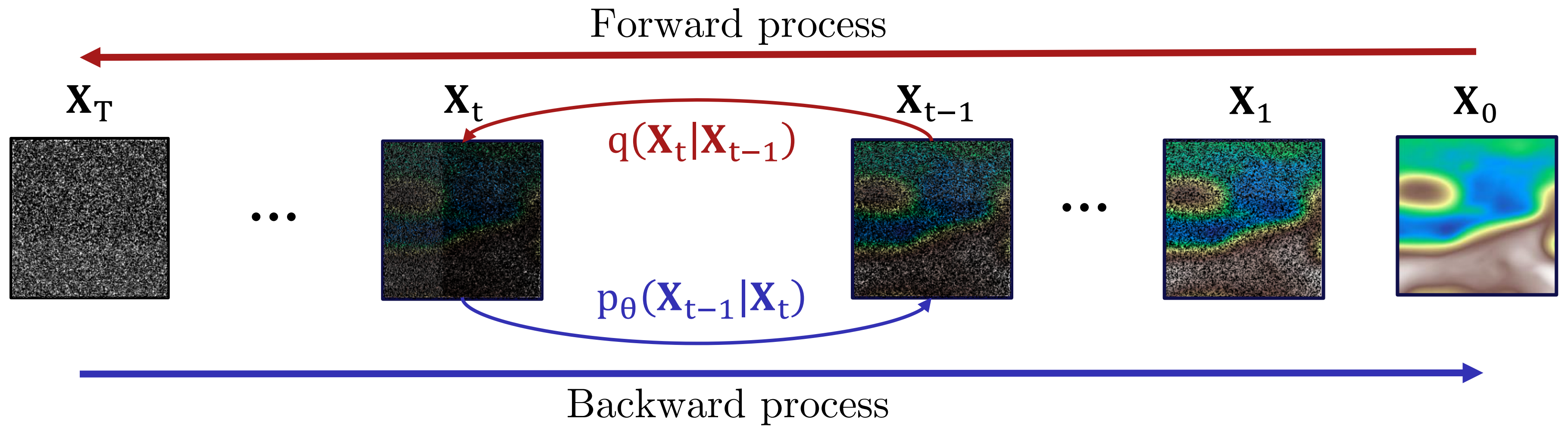



Difference between NRT persistence and forecast




- Sequential forecast outperforms simultaneous
- No differences between observations and pseudo-labels losses
- The forecast is useful as the persistence performances are systematically lower

# DDPM




**Forward process:**

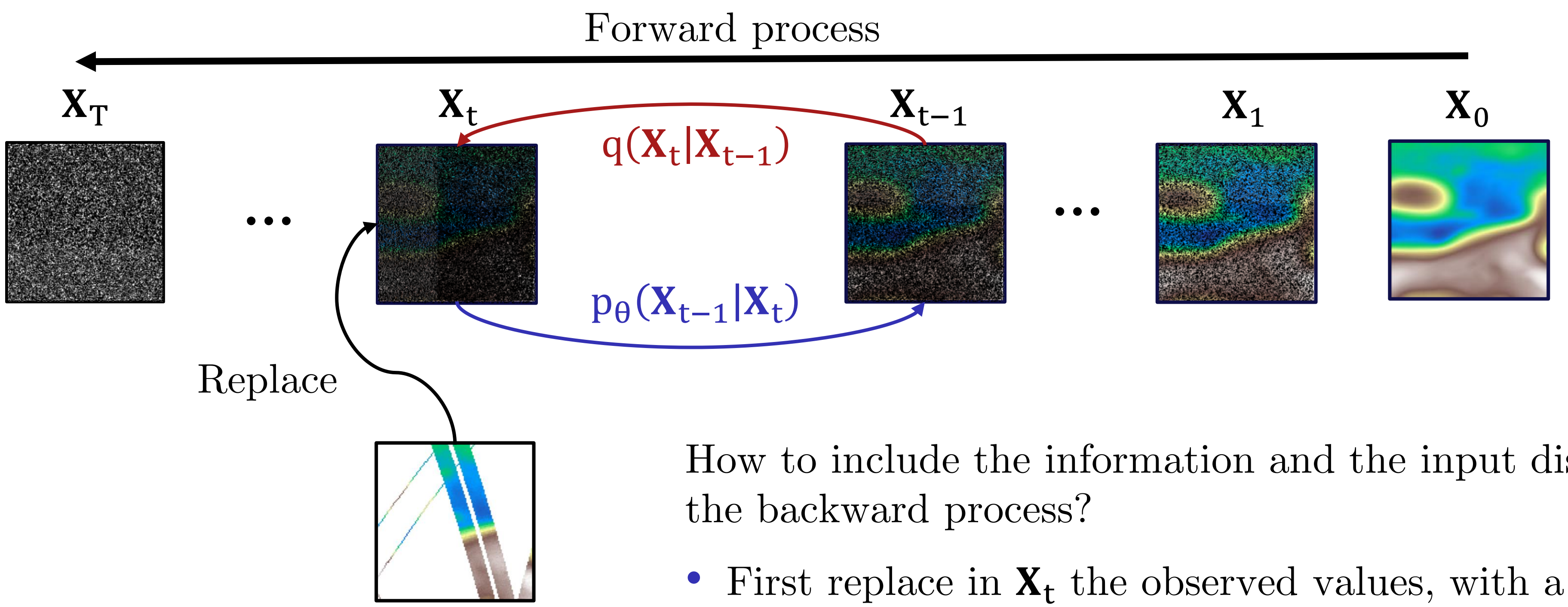
- Progressive noising process
  $q(\mathbf{X}_t|\mathbf{X}_{t-1}) = N(\mathbf{X}_t; \sqrt{1-\beta_t}\mathbf{X}_{t-1}, \beta_t \mathbf{I})$

**Backward process:**

- A neural network inverses one step of the noising process
  $p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t) = N(\mathbf{X}_{t-1}; \mu_\theta(\mathbf{X}_t, t), \beta_t \mathbf{I})$

- Samples a tractable distribution
- Progressively transform the sample into the target distribution
- By doing this inference several times, we can have several example of the taget distribution

# DDPM




How to include the information and the input distribution in the backward process?

- First replace in $\mathbf{X}_t$ the observed values, with a noise equivalent to the step t
- Then perform the denoising step

Doing so the network is guided to the specific example and conditioned to observations

# Optimal Interpolation

Optimal interpolation : $\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^b)$ with $\mathbf{K} = \mathbf{B}\mathbf{H}^\intercal(\mathbf{H}\mathbf{B}\mathbf{H}^\intercal + \mathbf{R})^{-1}$

- **H**: Observation operator supposed linear
- **B**: Covariance matrix of the background
- **R**: Covariance matrix of the observations errors

In DUACS:

- p observations : $\mathbf{y} \in \mathbb{R}^p = (x_i + \varepsilon_i), i \in [1,\ p]$
- State: $\mathbf{x} \in \mathbb{R}^{p+1} = (v,\ x_1, \dots x_p)^\intercal$
- Estimation of the unobserved value v:

$$\hat{v} = \sum_{i,j=1}^{p} A_{ij}^{-1} C_i y_i ,$$

$$\text{where } A_{ij} = \mathrm{cov}(x_i , x_j) + \mathrm{cov}(\varepsilon_i , \varepsilon_j)$$
$$\text{and } C_i = \mathrm{cov}(v , x_i)$$

Covariance tuning:

DUACS

- Covariances obtained by a gaussian correlation function (for the state), and estimation of the error for the observations

DYMOST

- A Quasi-Geostrophic model is used to propagate contributions of the observations.

MIOST

- The covariance model is a multiscale wavelet decomposition