Super-resolution of satellite observations of sea ice thickness using diffusion models and physical modeling

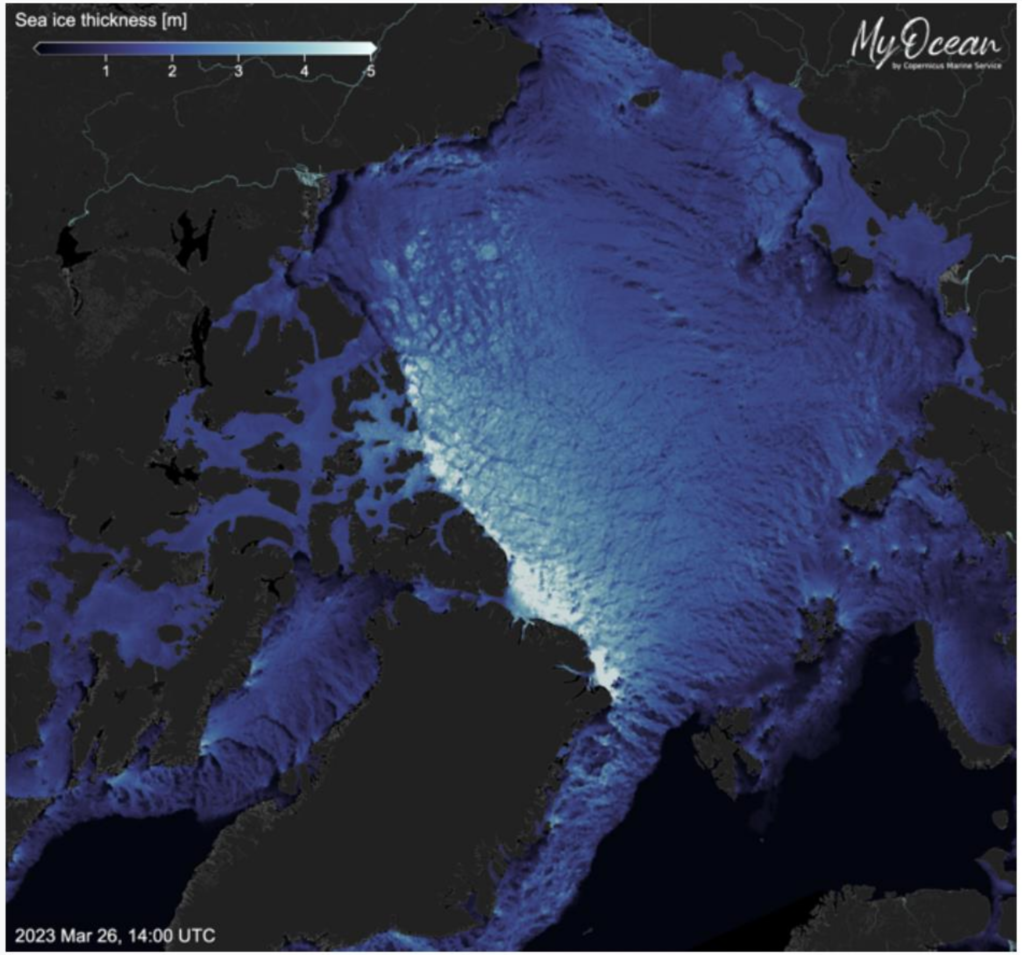
Julien Brajard, Anton Korosov, Fabio Mangini, Adrien Perrin, Richard Davy, and Yiguo Wang

NERSC
STIFTELSEN NANSEN SENTER FOR MILJØ OG FJERNMÅLING
NANSENSENTERET · BERGEN · NORGE

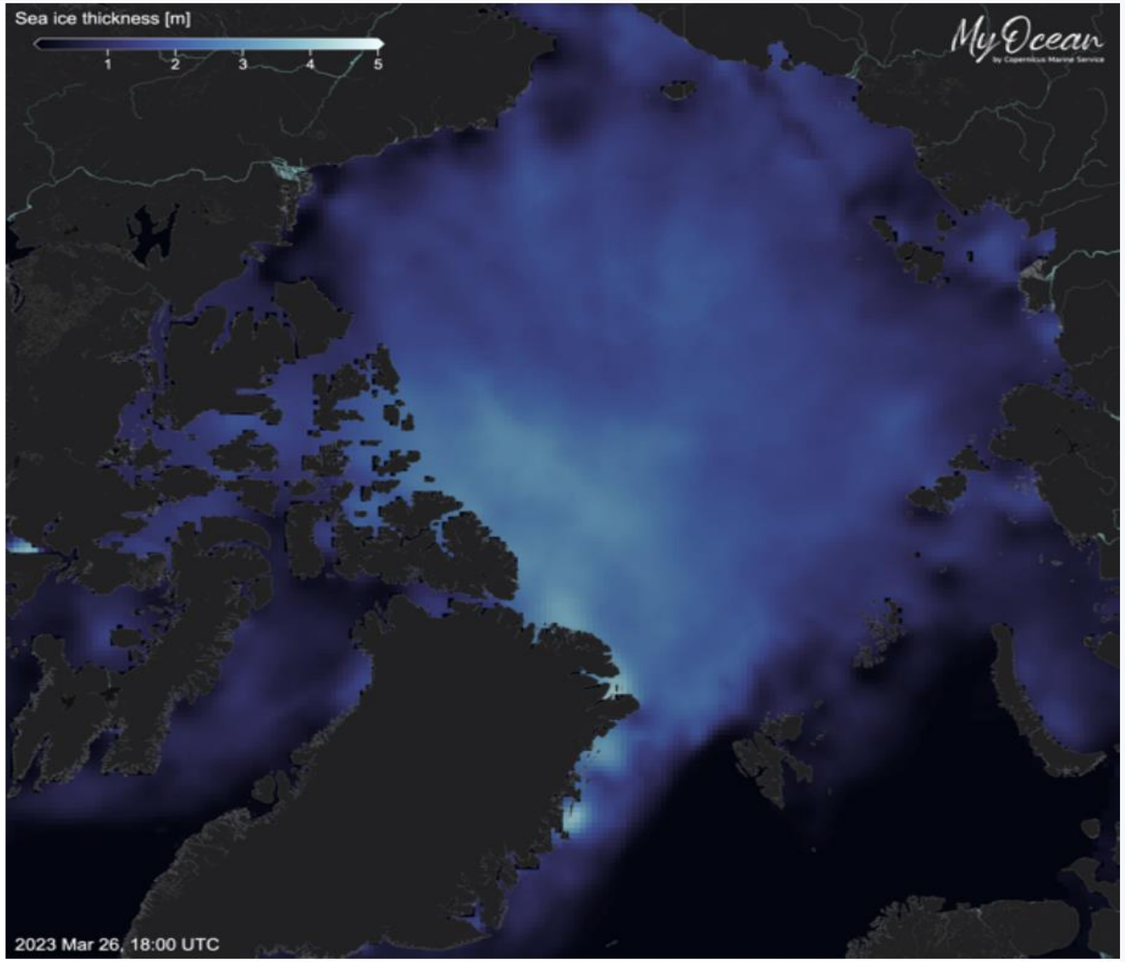
esa

Context

High-resolution model simulation of Sea ice thickness in the Arctic
(3 km resolution)

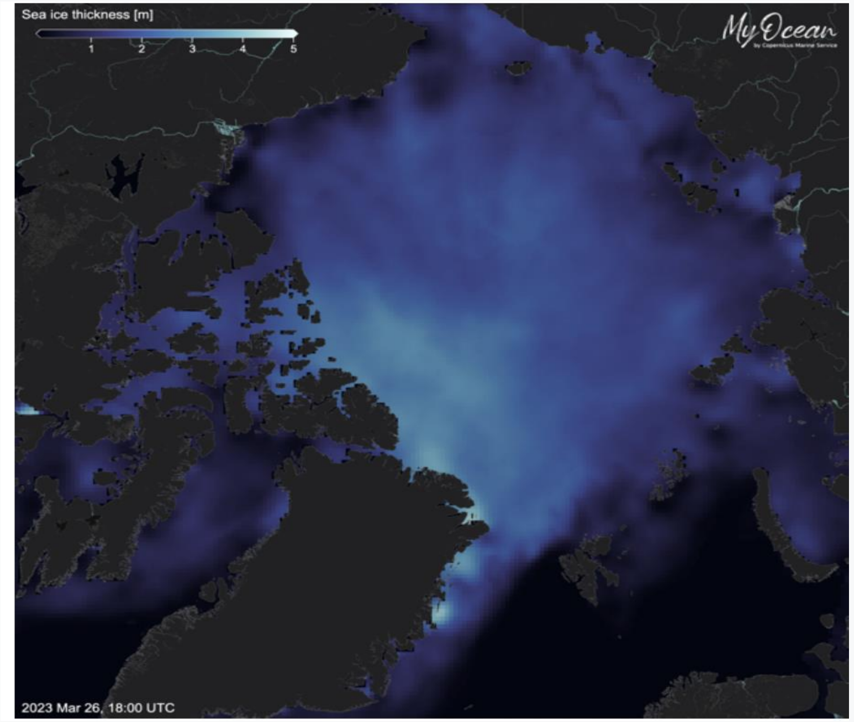


The resolution we can observe
(~90 km resolution)



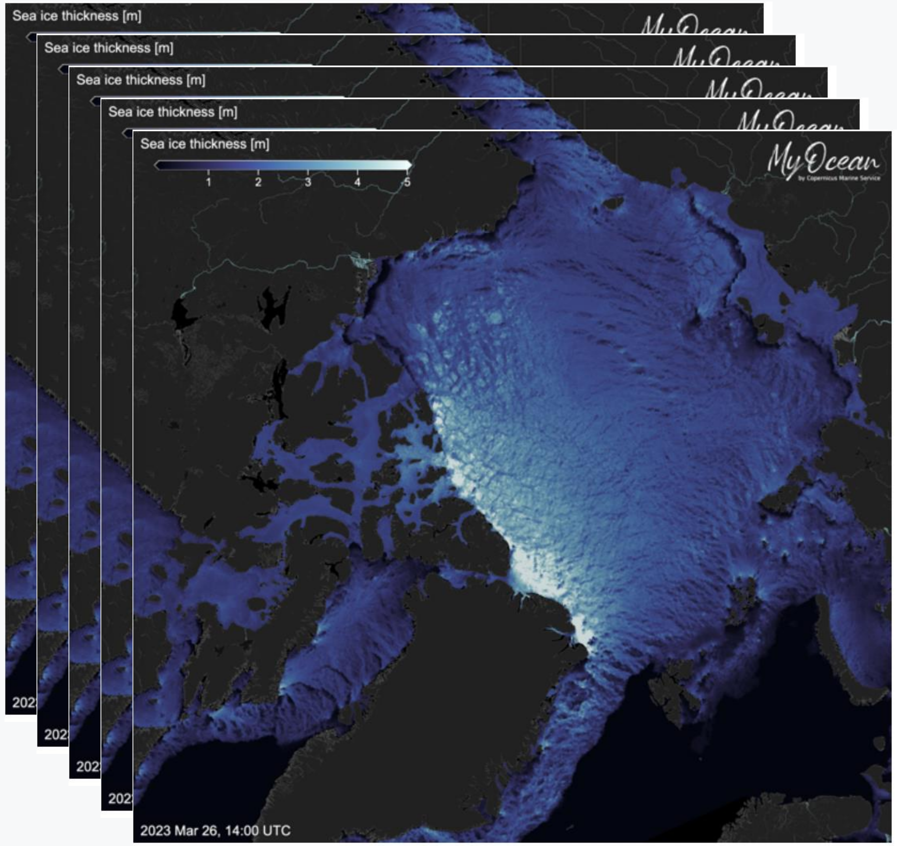
Objective of SuperIce

Low resolution image of sea ice thickness



Neural Networks

High-resolution images



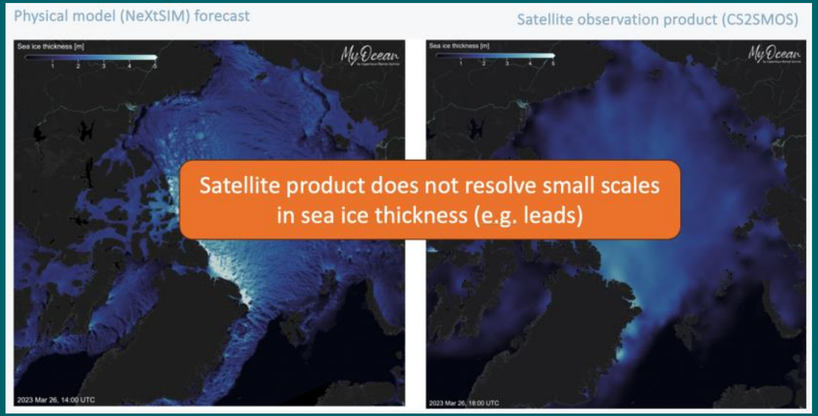
We use diffusion models to generate an ensemble of high-resolution sea ice thickness.

Good accuracy and realism of the generated fields.

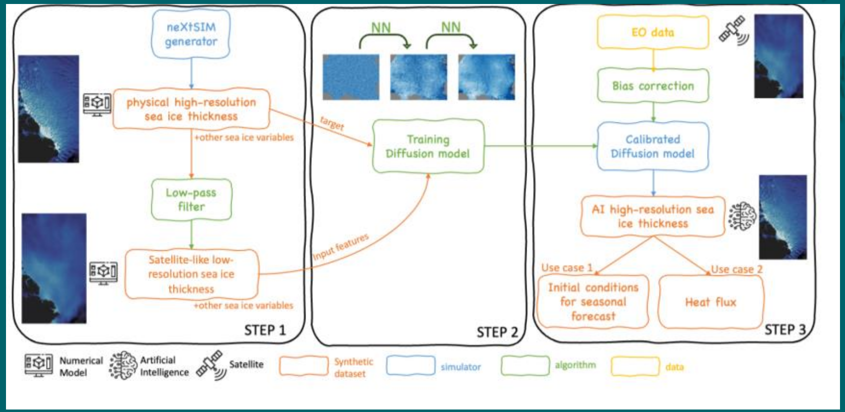
Please visit my poster for more details #13

Super-resolution

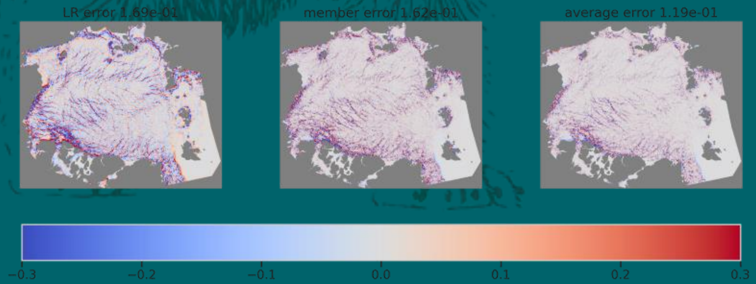
Motivation



Sketch of the project



Results



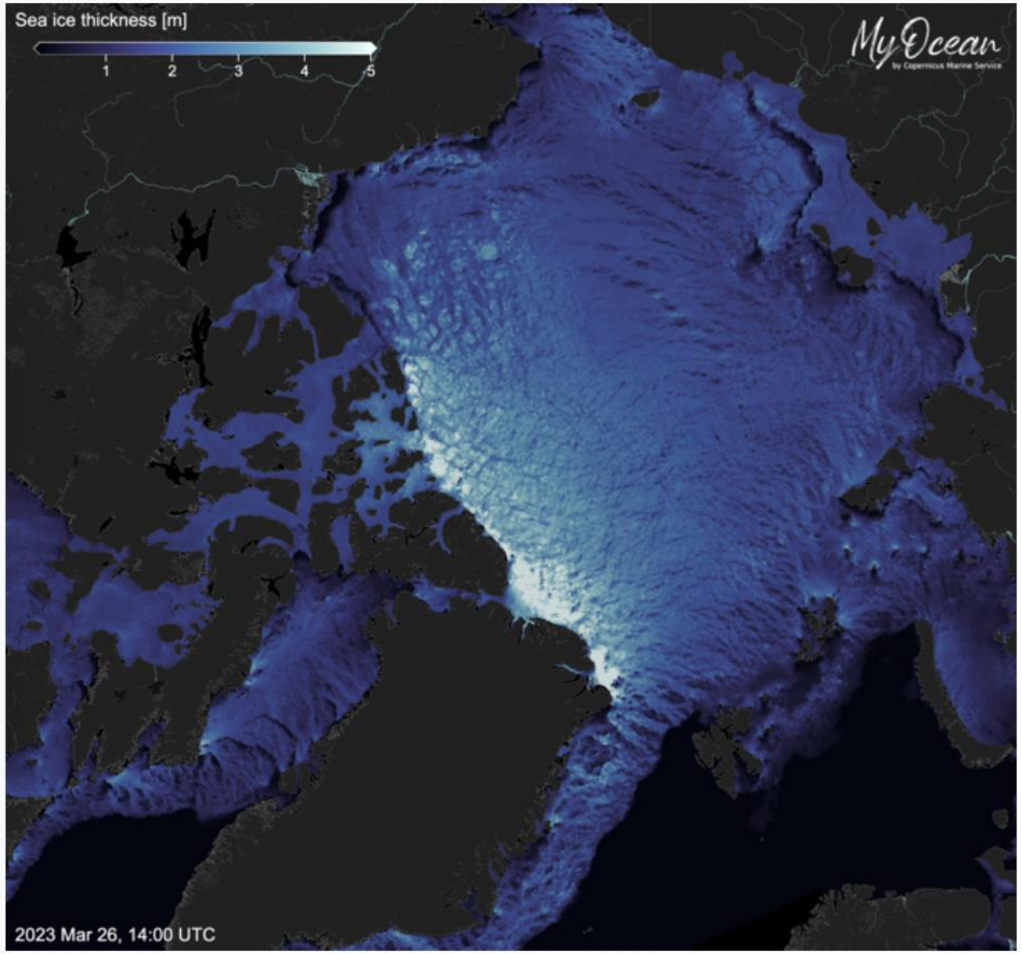
NERSC
STIFTELSEN NANSEN SENTER FOR MILJØ OG FJERNMÅLING
NANSENSENTERET · BERGEN · NORGE

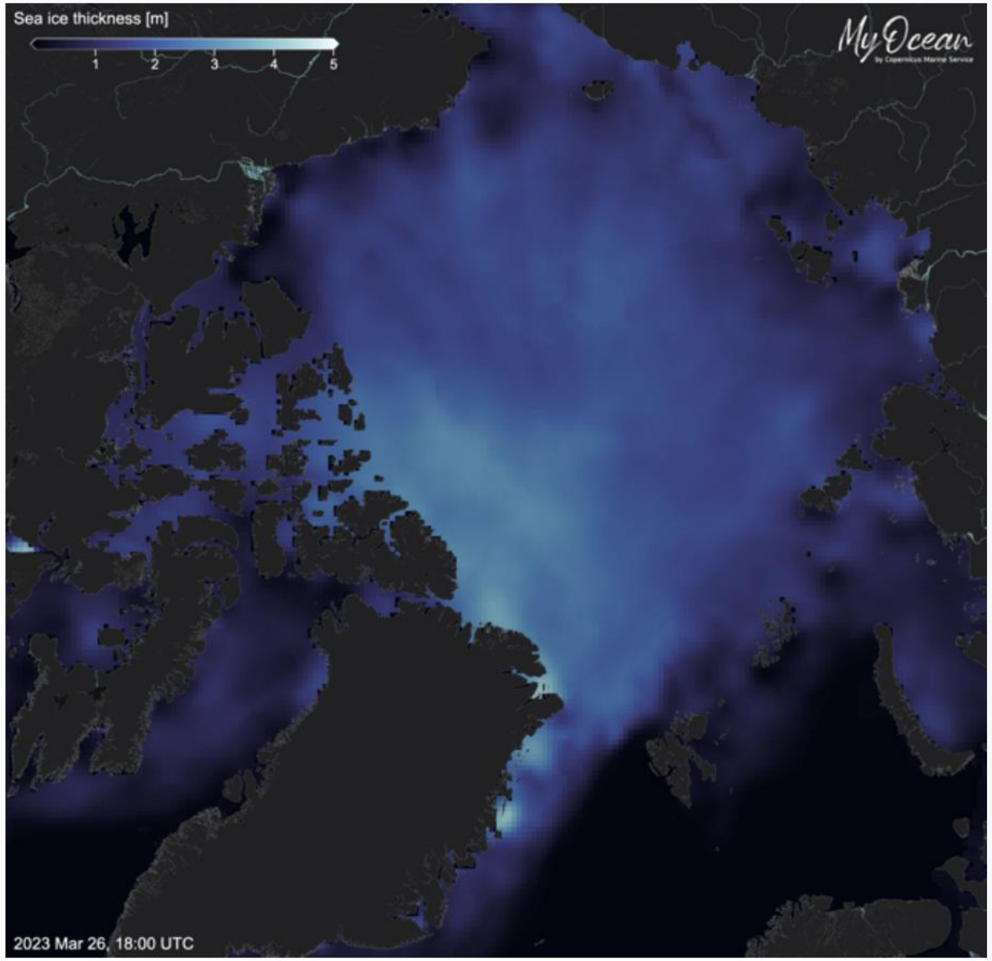
esa

Motivation

Physical model (NeXtSIM) forecast

Satellite observation product (CS2SMOS)





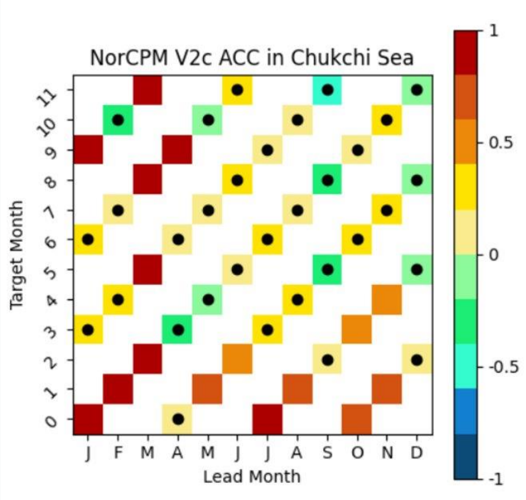
Satellite product does not resolve small scales in sea ice thickness (e.g. leads)

Why is it important?

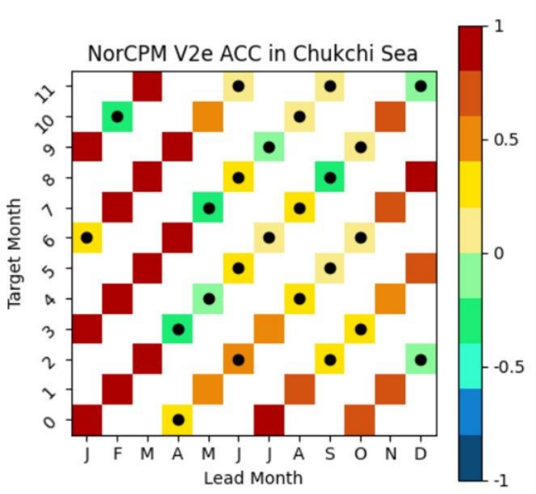
Case 1: Predictability

Forecast skill

Detrended correlation coefficient of Sea ice extent in Chukchi Sea



Initialization using Sea ice concentration observations only

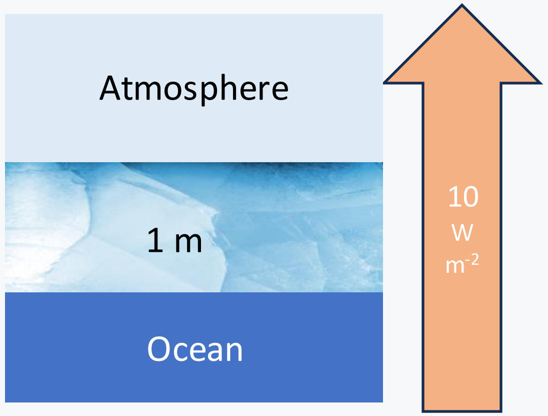


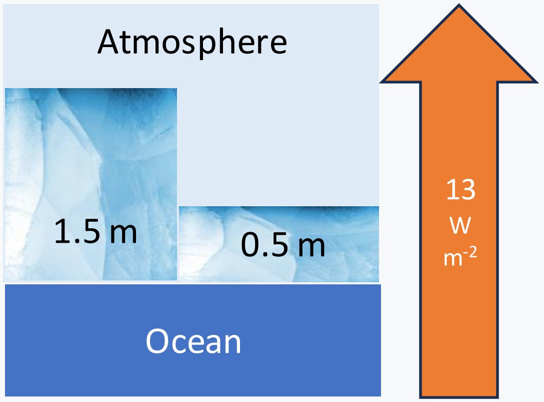
Initialization using Sea ice concentration observations + Sea ice thickness only

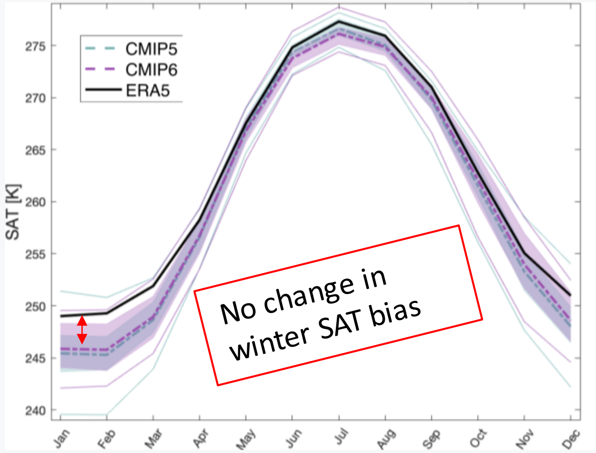
Black dot means not significant

Courtesy of N. Williams

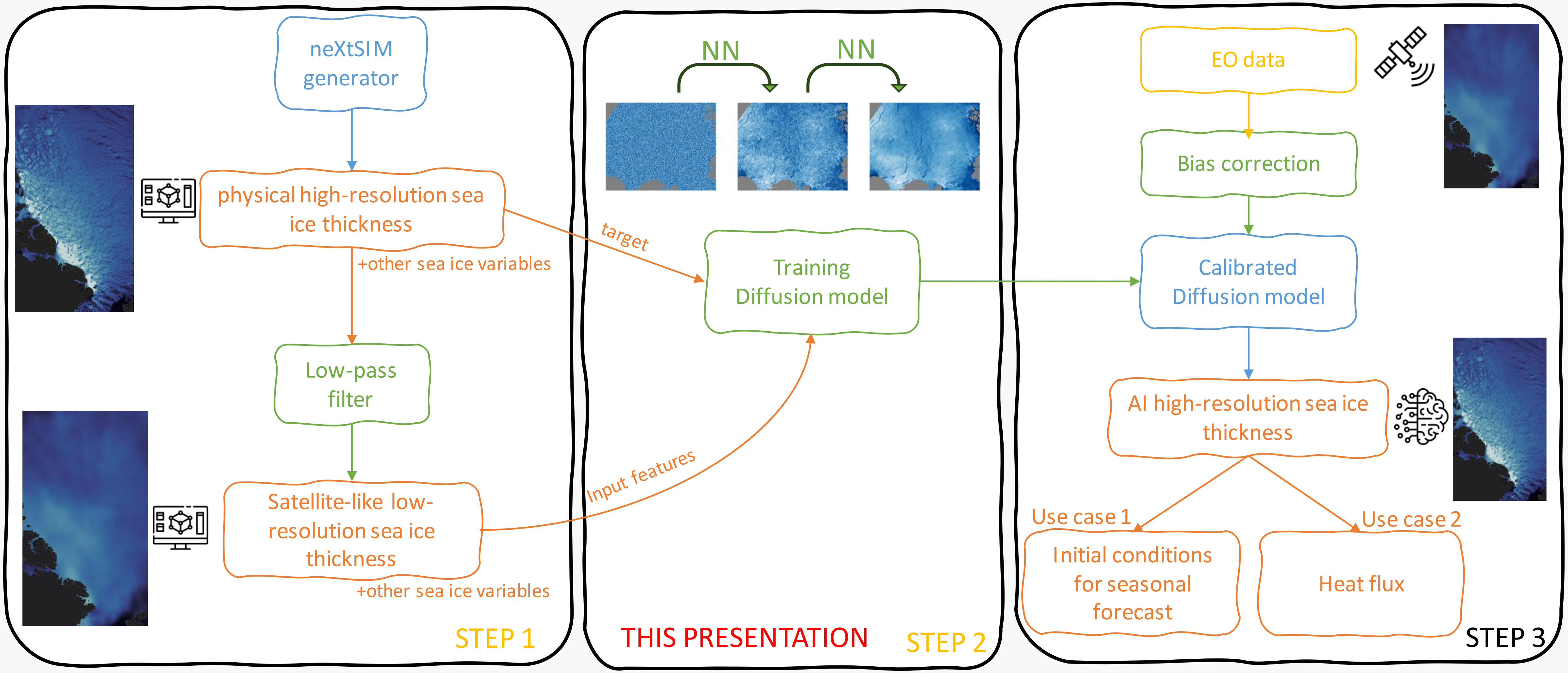
Case 2: Surface fluxes







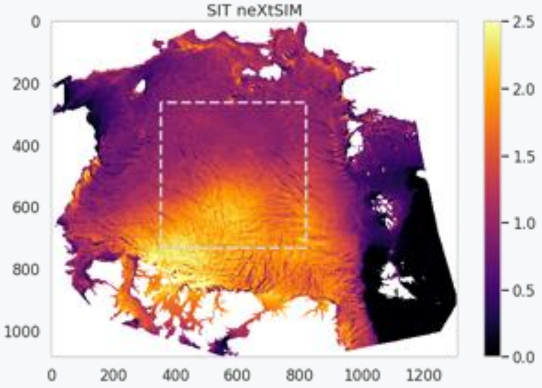
Overview of the project



Step 1: Dataset constitution

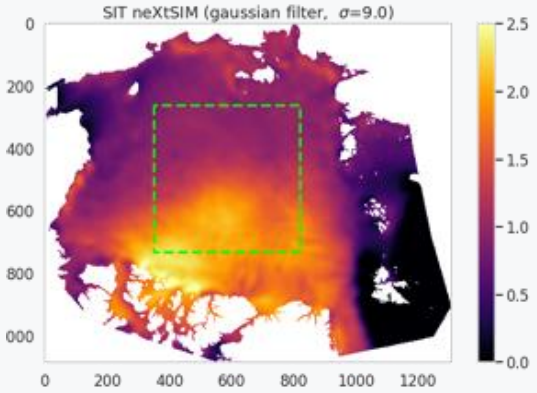
Principle: Filtering of NeXtSIM simulations

NeXtSIM sea ice thickness
01-01-2021 (res 3km)

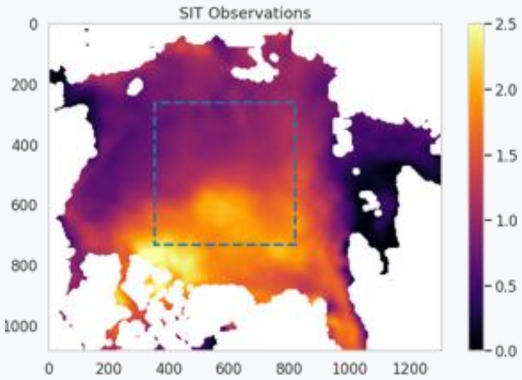


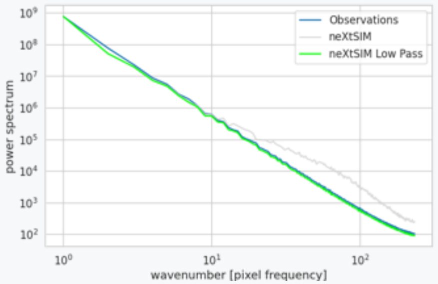
Convolution constant kernel (size 120 km)

Filtered sea ice thickness
(res ~ 90 km)

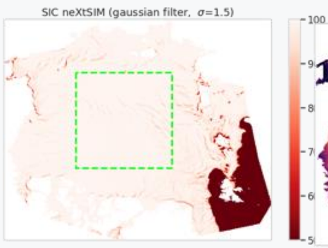


Observation product (CS2SMOS)
01-01-2021

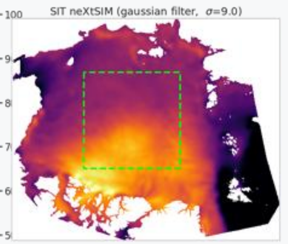




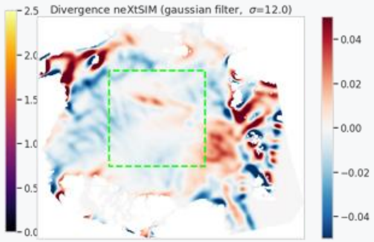
Concentration
(res ~15km)



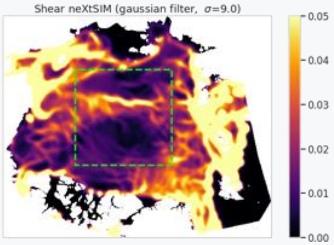
Thickness
(res ~90km)



Divergence
(res ~120km)



Shear
(res ~90km)



+ land mask
(res ~3km, no smoothing)

Dataset:

- 5 input features (low-pass filtered) 1086x1308
- 529 samples, from 18-10-2013 to 15-04-2023 (Only Oct-Apr)
- Training: 2013-> Apr. 2020 (1157 samples)
- Validation/test Oct. Oct. 2020-> 2023 (540 samples)

Apply diffusion model to sea ice thickness super-resolution

Used for AI image generator (Ex: Midjourney)

A prompt

“man walking dog at dusk --ar 4:3”

Generative diffusion model

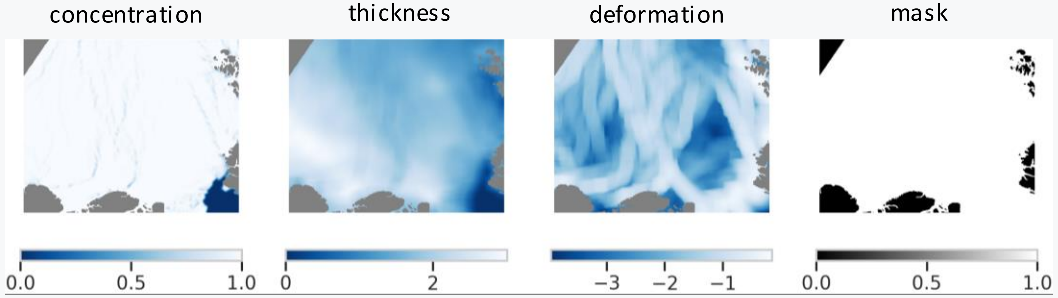
A generated image

Source: https://aituts.com/midjourney-camera-prompts/

Apply diffusion model to sea ice thickness super-resolution

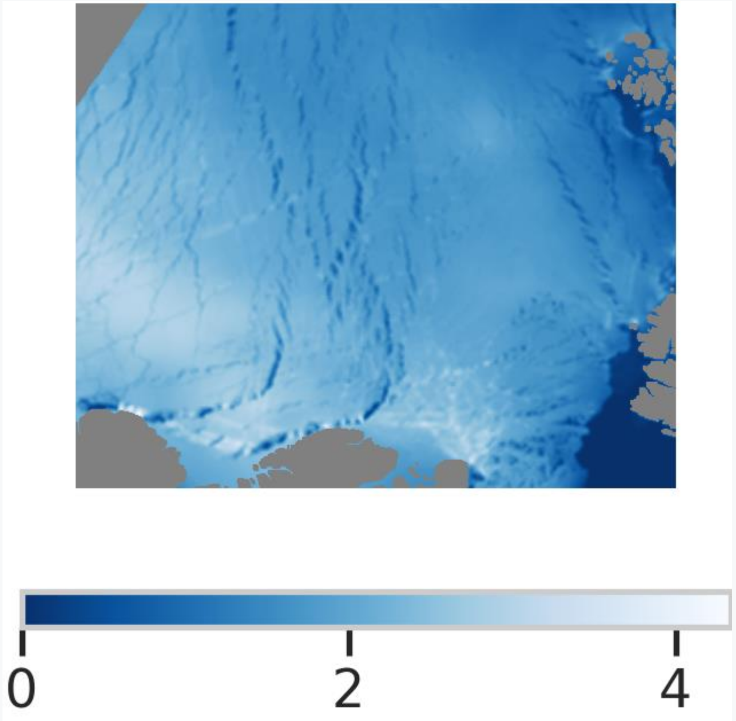
Used for AI image generator (Ex: Midjourney)

Observable low-resolution images



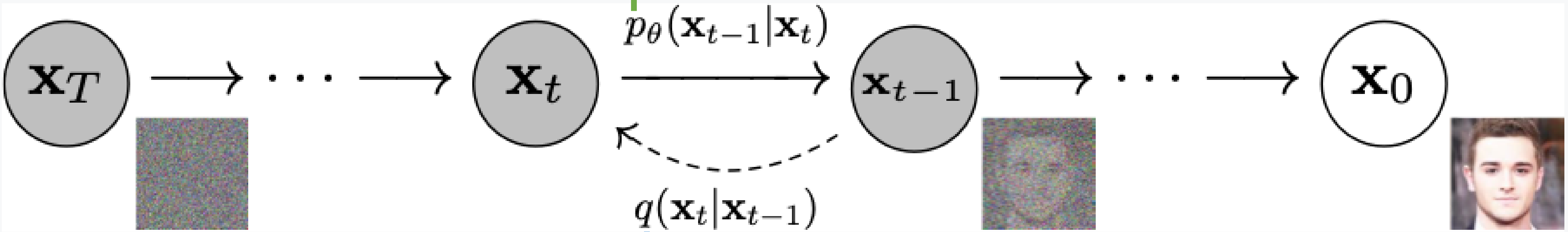
Generative diffusion model

A generated high-resolution image



Principle of the diffusion model

To be learn
using a neural network

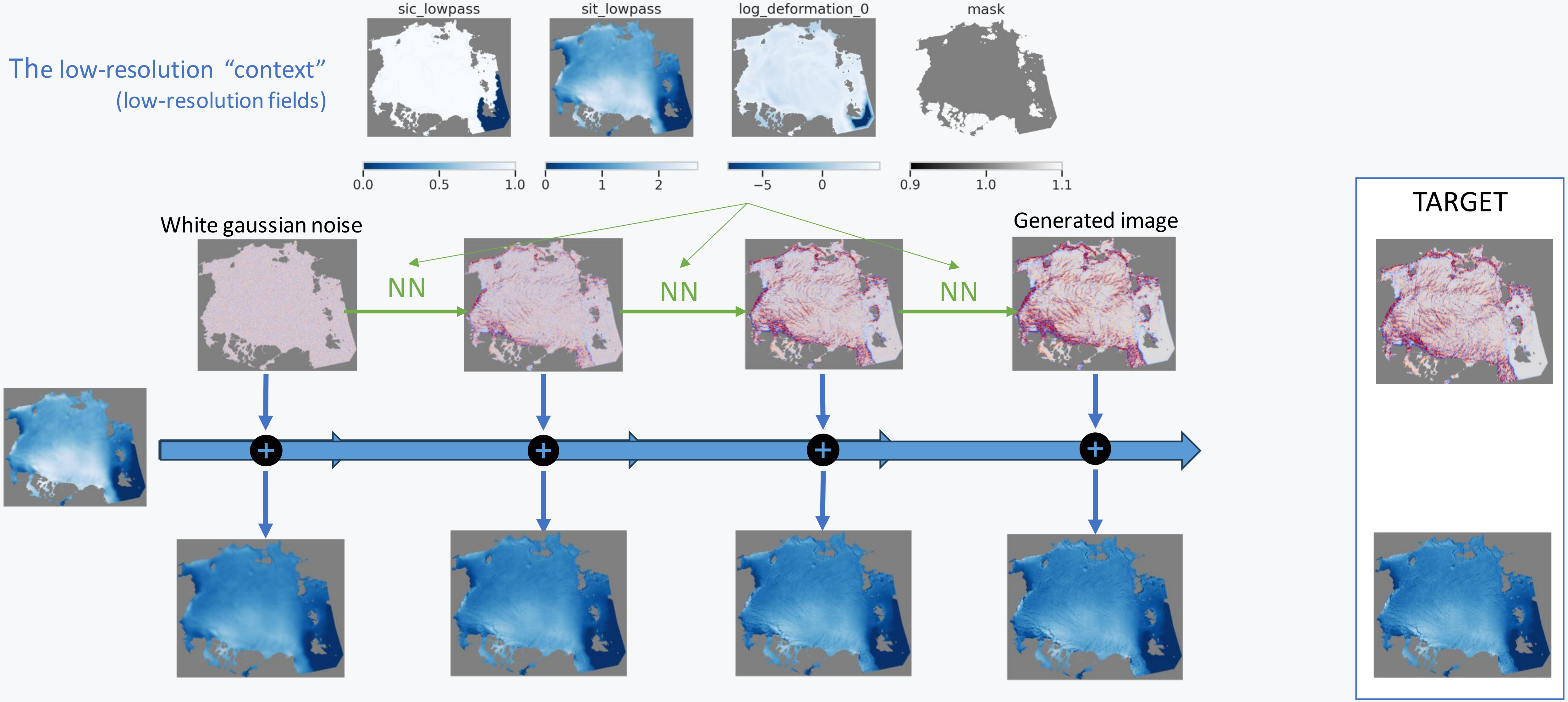


Predefined:
Adding a white Gaussian noise

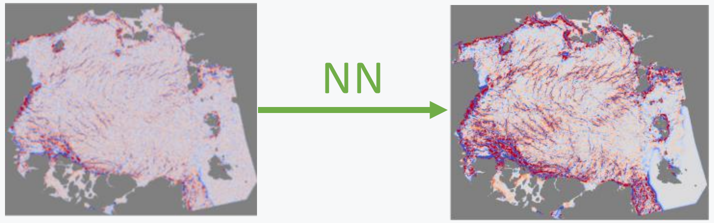
Source Ho et al. 2020 https://arxiv.org/abs/2006.11239

Principle of the diffusion model in SuperIce

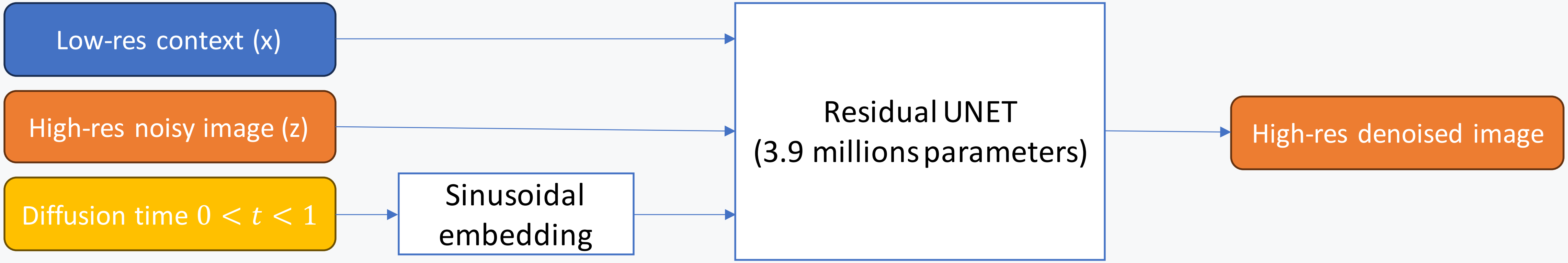
The low-resolution “context”
(low-resolution fields)



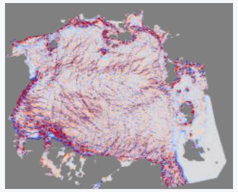
Implementation details



Model Residual Neural network f(x,z,t)



Training algorithm

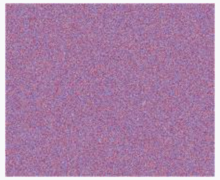
1 y  x

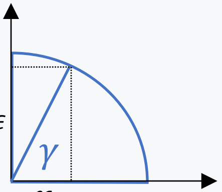
r_ϵ

For one sample

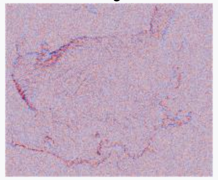
1. Draw a HR image y and a LR context x in the training set
2. Draw a diffusion time t between 0 (full signal) and 1 (full noise)
3. Draw a white Gaussian noise ϵ
4. Compute diffusion angle: $\gamma = \gamma_{min} + t.(\gamma_{max} - \gamma_{min})$
5. Compute the signal and noise rate: $r_s = \cos\gamma$, $r_\epsilon = \sin\gamma$
6. Compute the noisy image: $z_t = r_s . y + r_\epsilon . \epsilon$
7. Predict the noise by the NN: $\hat{\epsilon} = f_\theta(z_t, t, x)$
8. Predict the image: $\hat{z}_{t-1} = (z_t - r_\epsilon . \hat{\epsilon}) / r_s$
9. Compute the loss on the noise: = $L(\theta) = \|\epsilon - \hat{\epsilon}\|^2$
10. Minimize L

ϵ

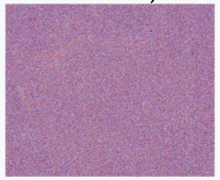
3 

5 

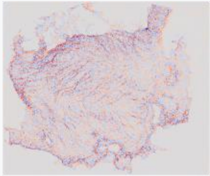
z_t

6 

$\hat{\epsilon} = f_\theta(z_t, t, x)$

7 

$\epsilon - \hat{\epsilon}$

9 

Learning curve

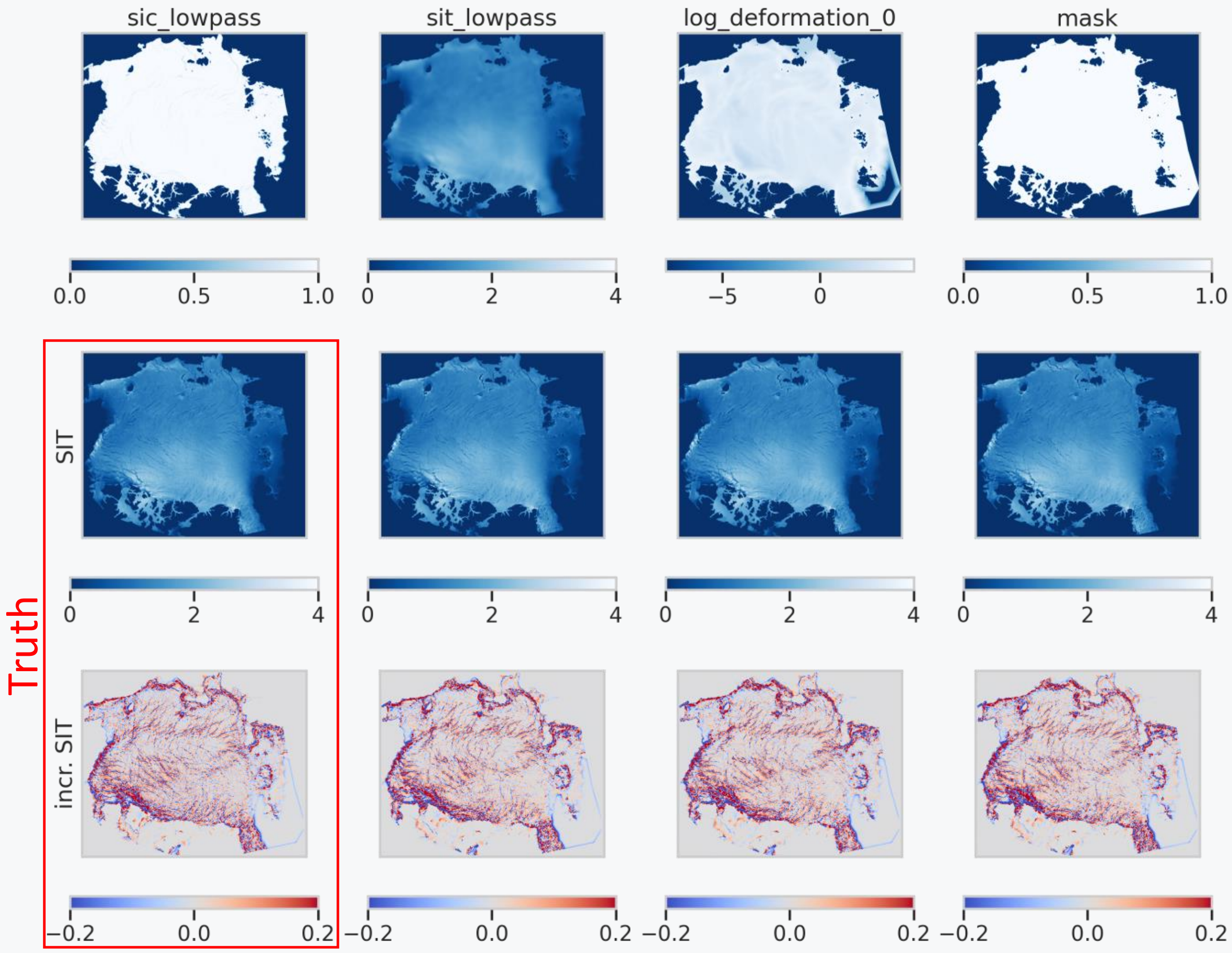
Results

- ✓ Results of the 10 January 2021
- ✓ Results of the 23 October 2020
- ✓ Global results

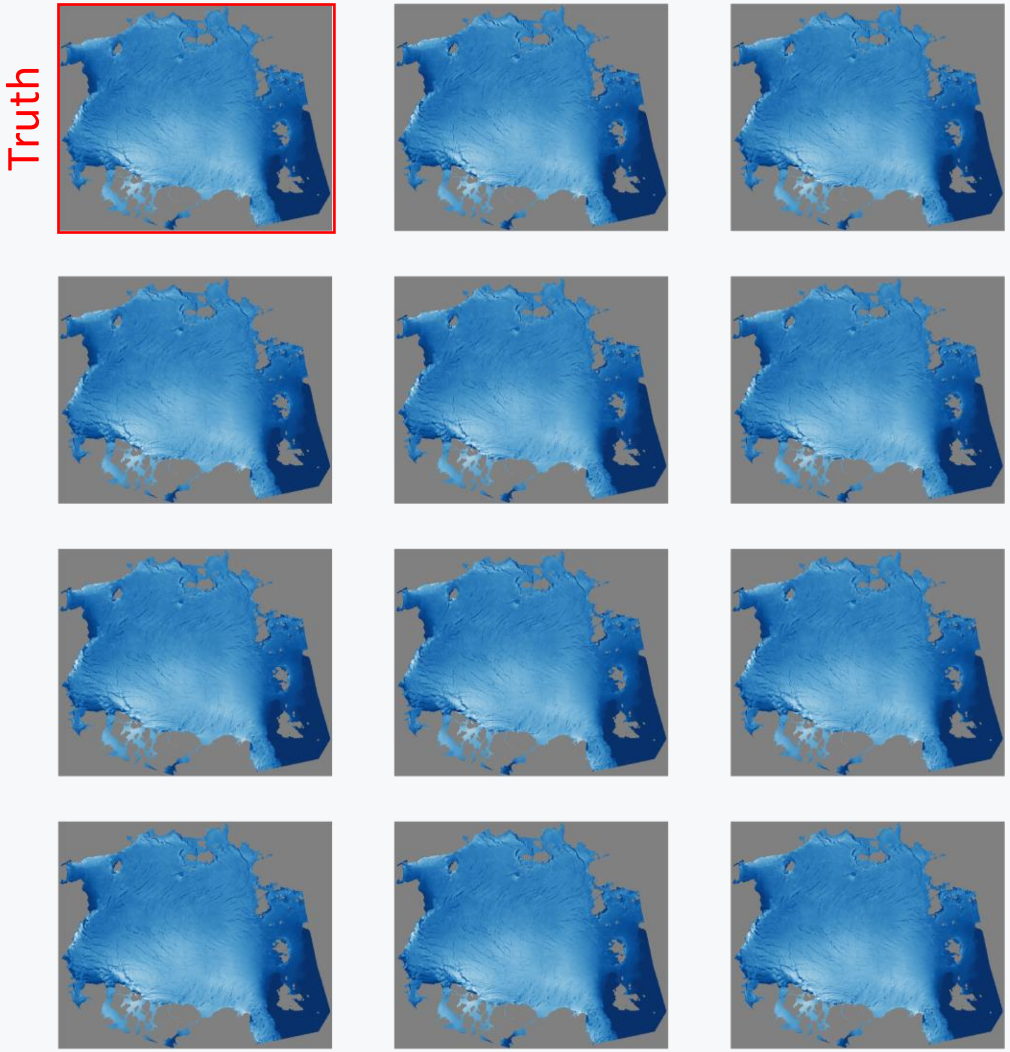
Link to 23 October 2020

Results of 26 January 2021

Generation of high-res SIT and residuals



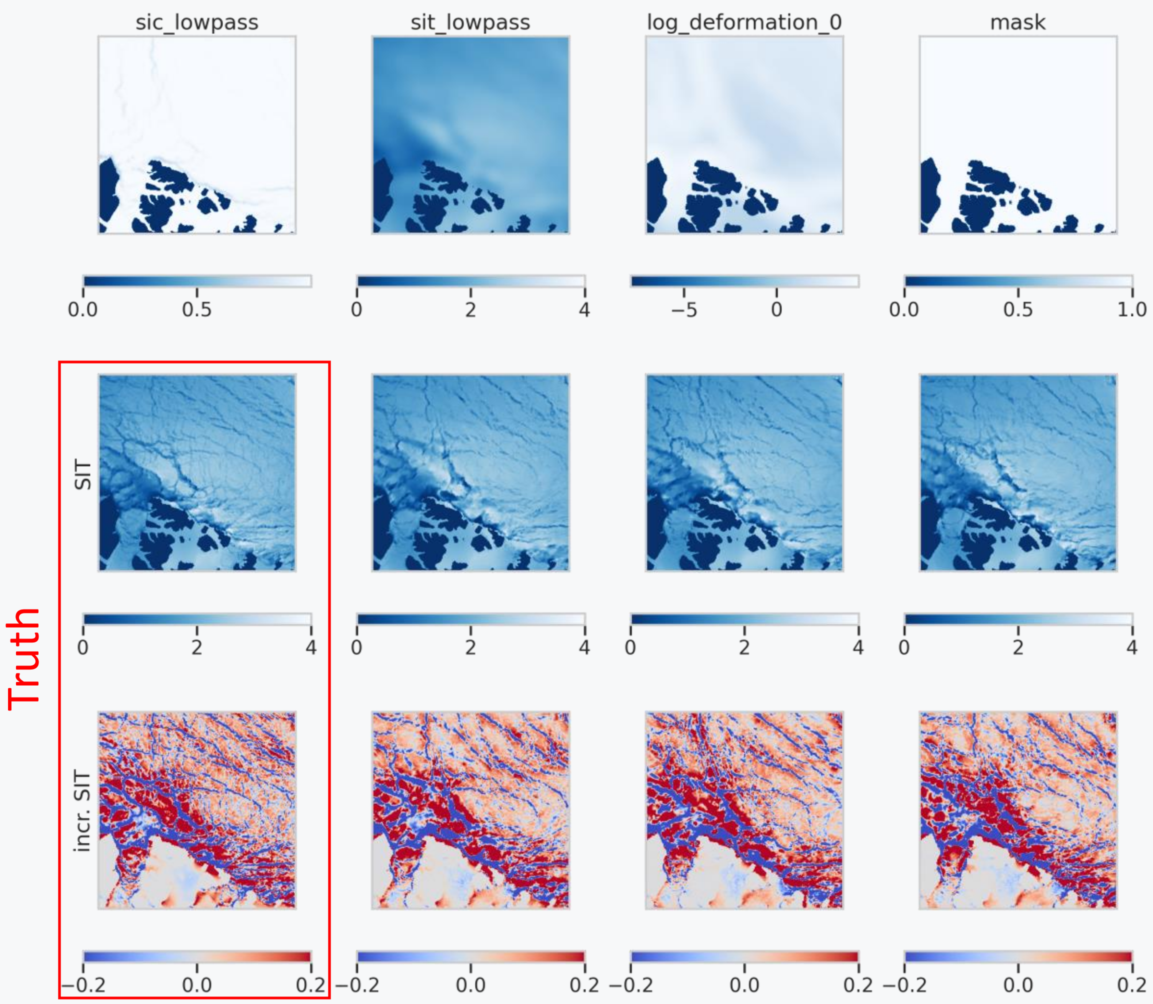
Generation of an ensemble



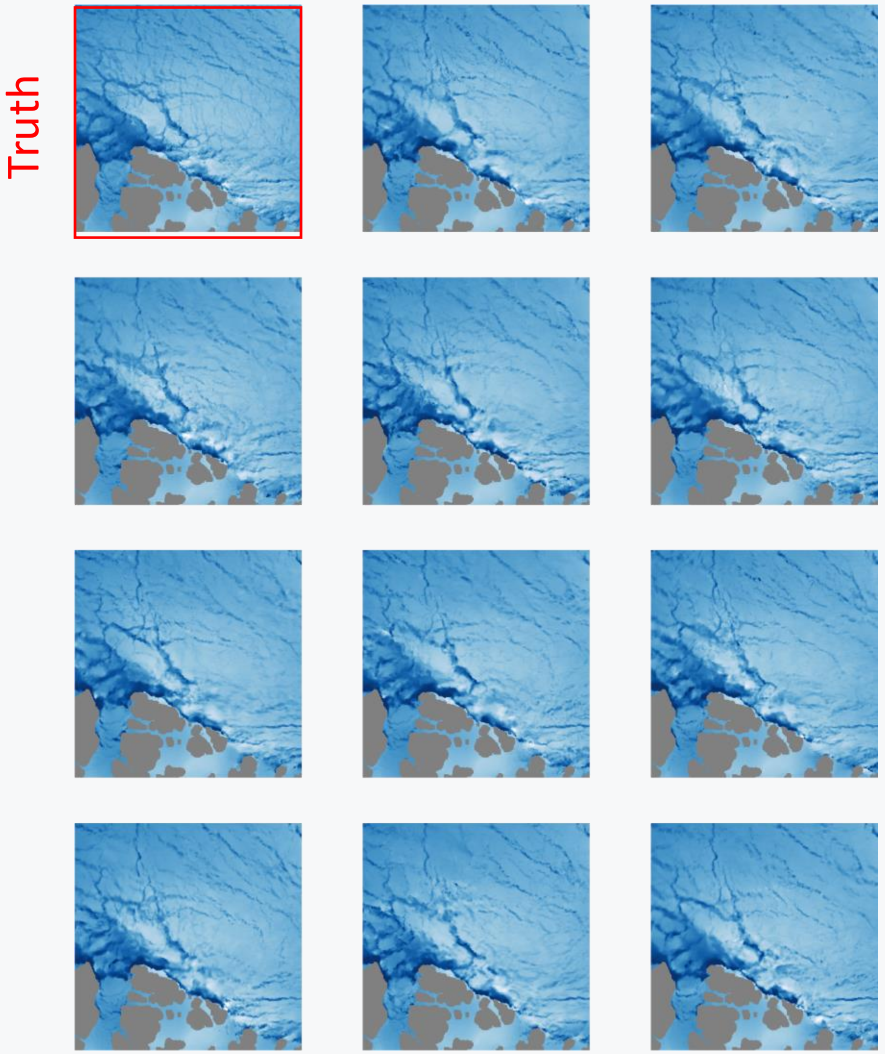
[Link to 23 October 2020](#)

Results of 26 January 2021 (zoom)

Generation of high-res SIT and residuals



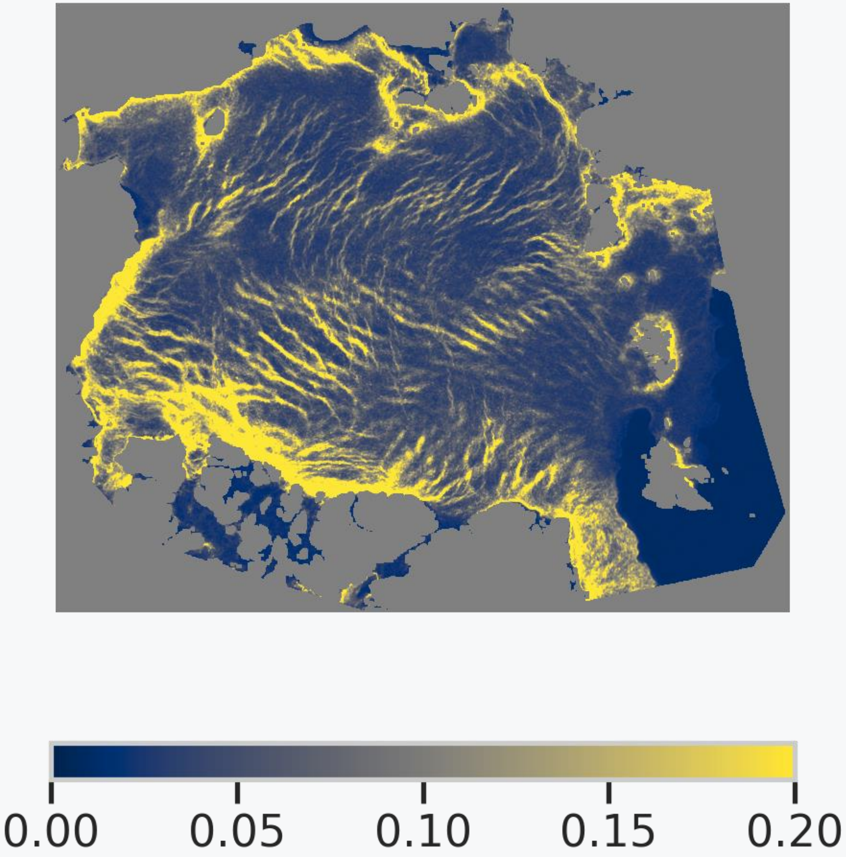
Generation of an ensemble

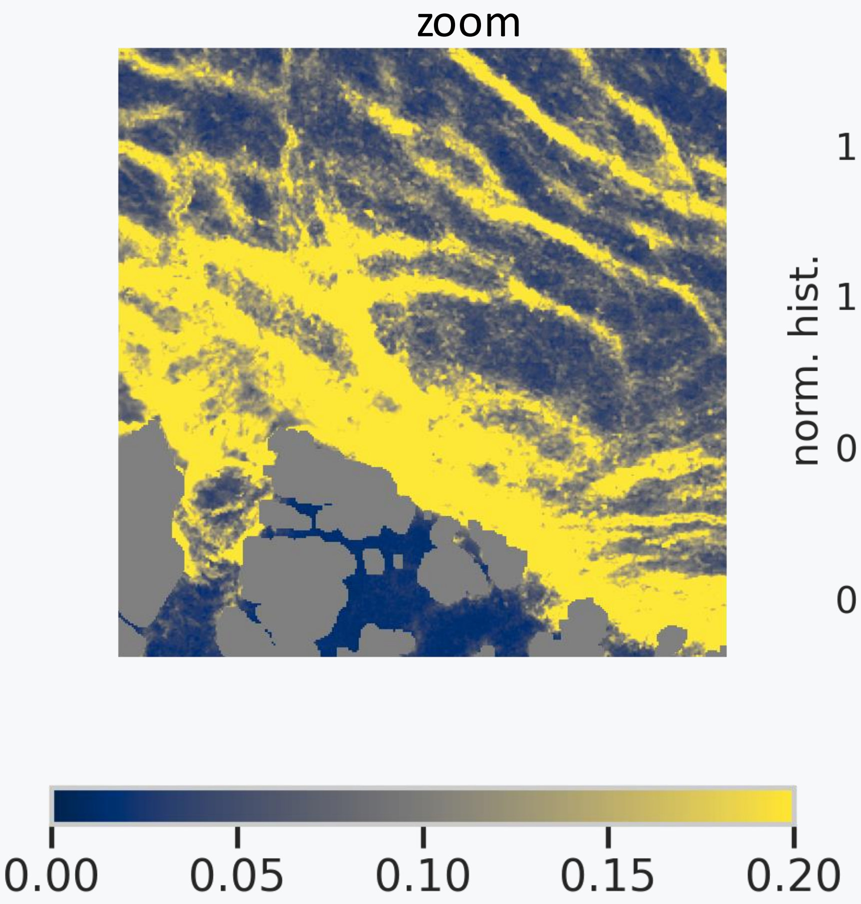


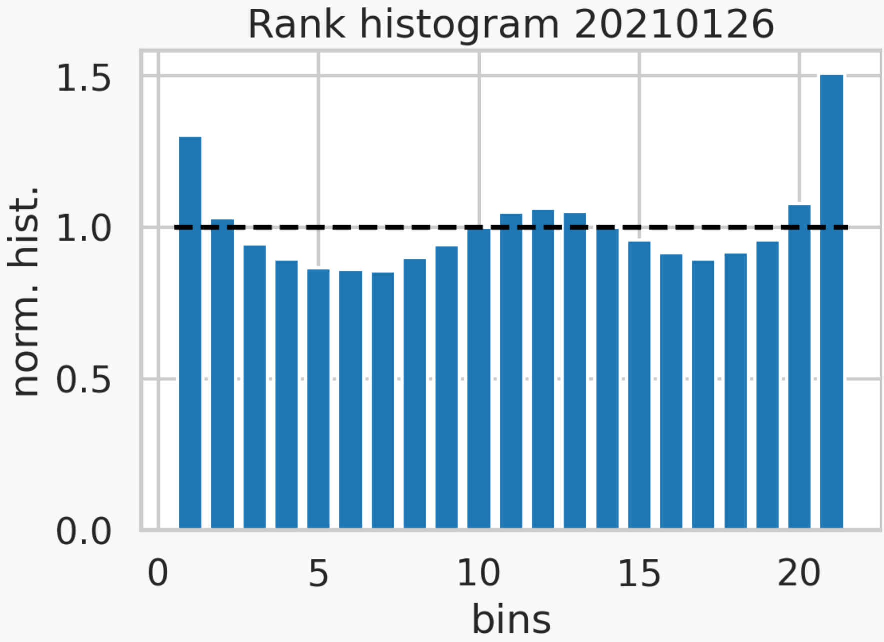
Link to 23 October 2020

Results of 26 January 2021

Spread and reliability



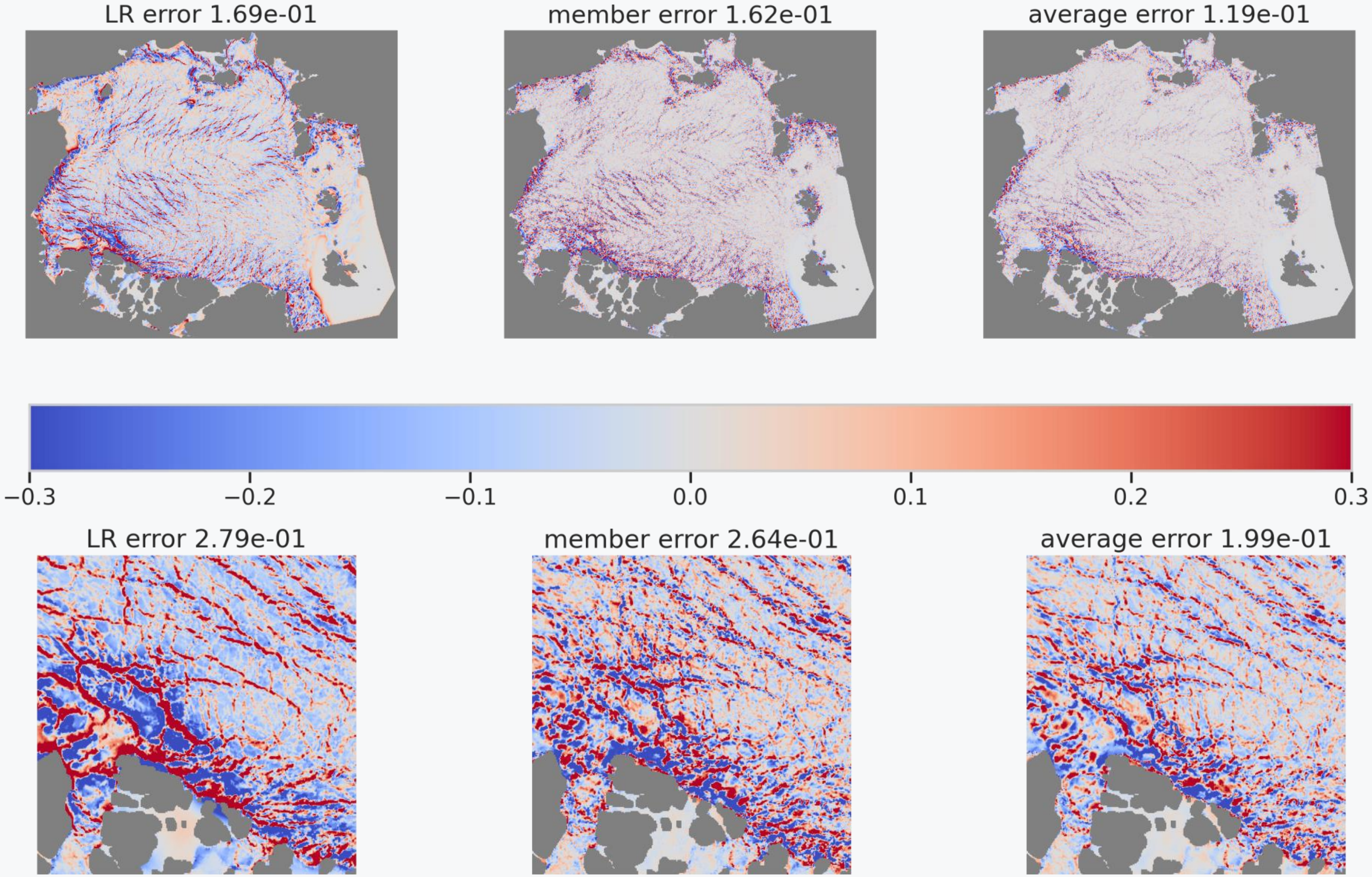




[Link to 23 October 2020](#)

Results of 26 January 2021

Accuracy

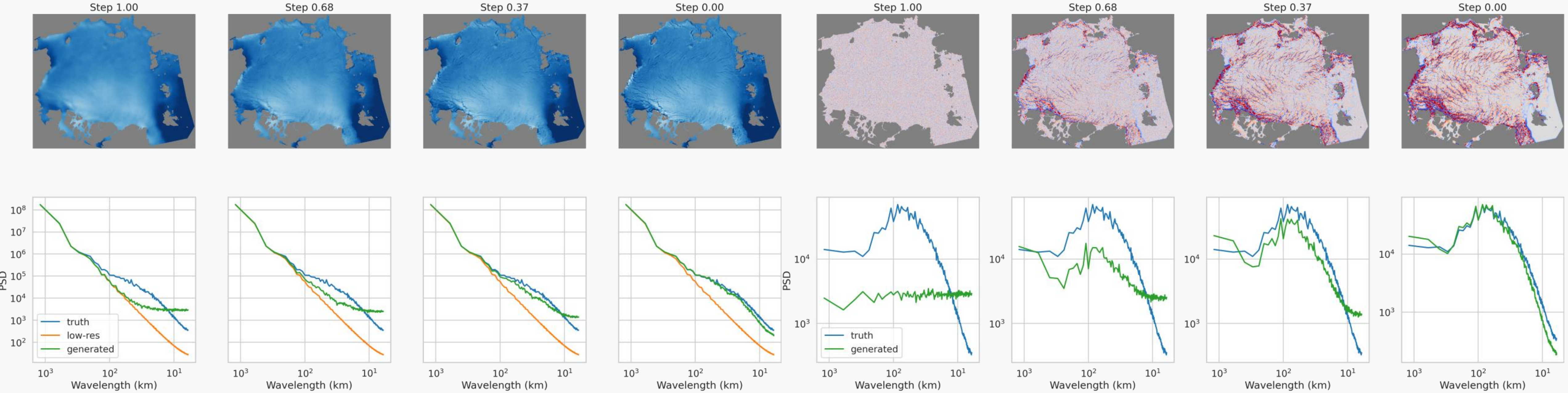


zoom

Link to 23 October 2020

Spectrum 26 January 2021

Power Spectrum Density as a function of the diffusion time (Step)

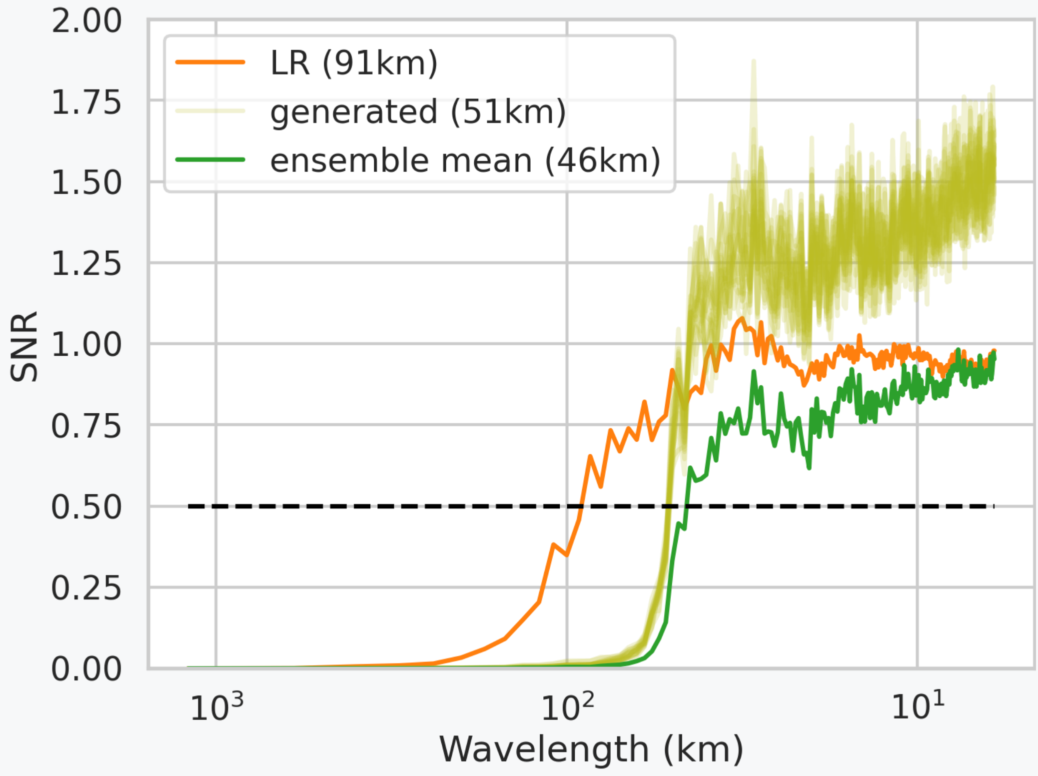


Link to 23 October 2020

Spectrum

Signal to noise

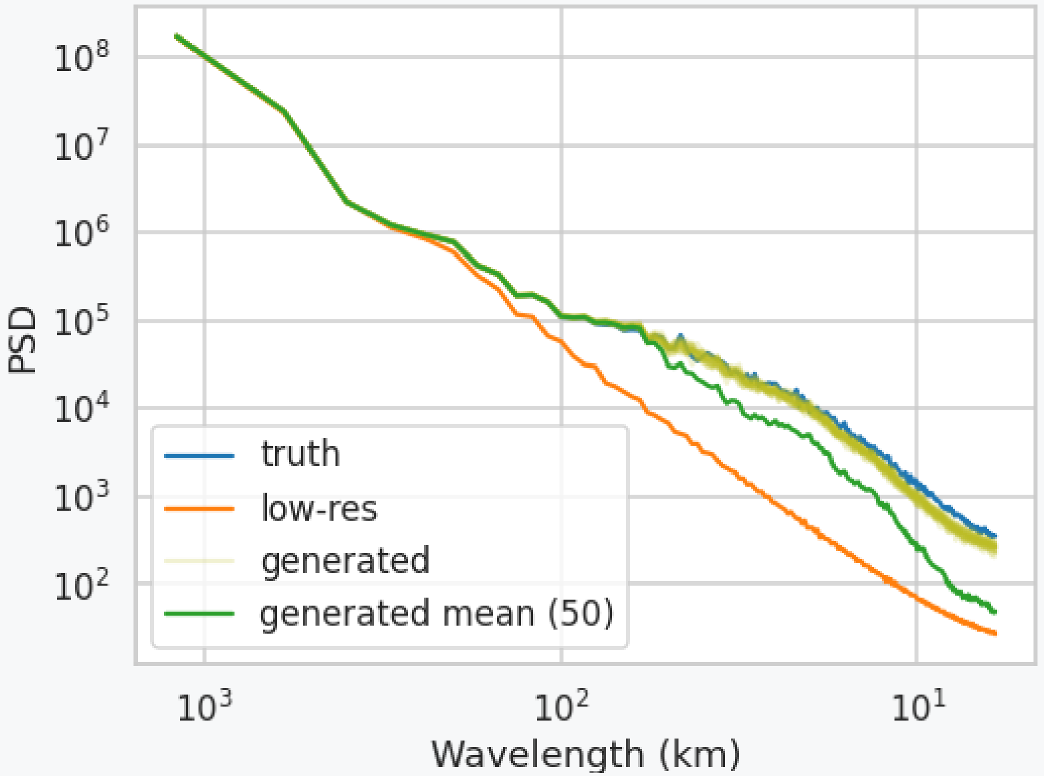
Spectral ration = PSD (reconstruction-truth) / PSD(truth)



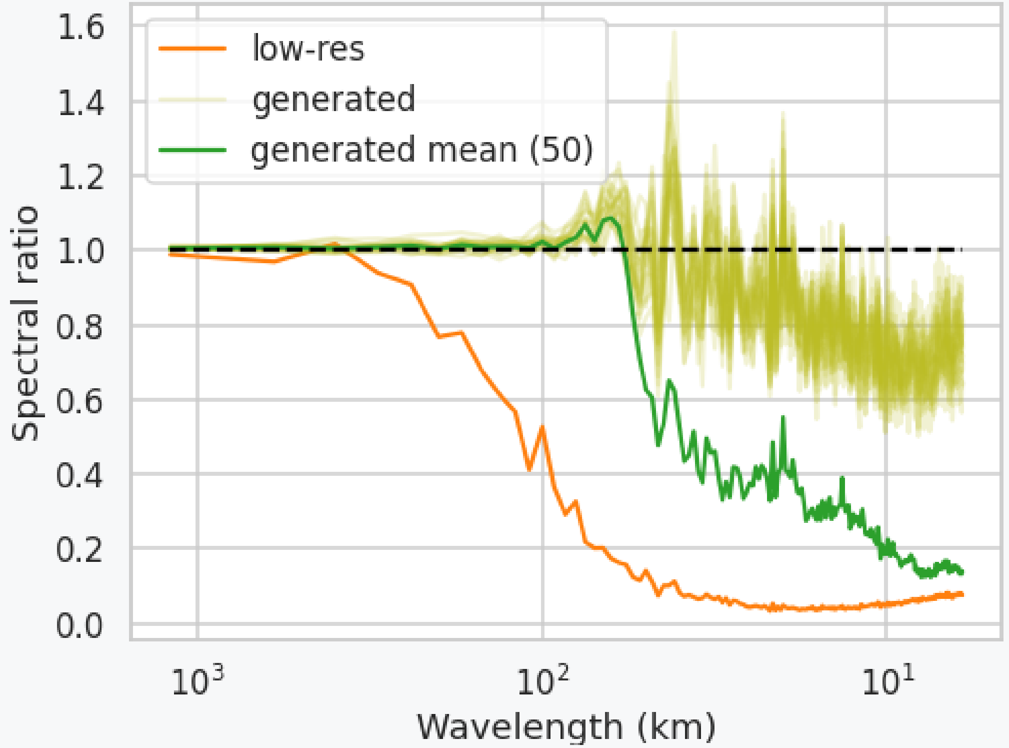
Link to 23 October 2020

Spectrum

Power Spectrum Density

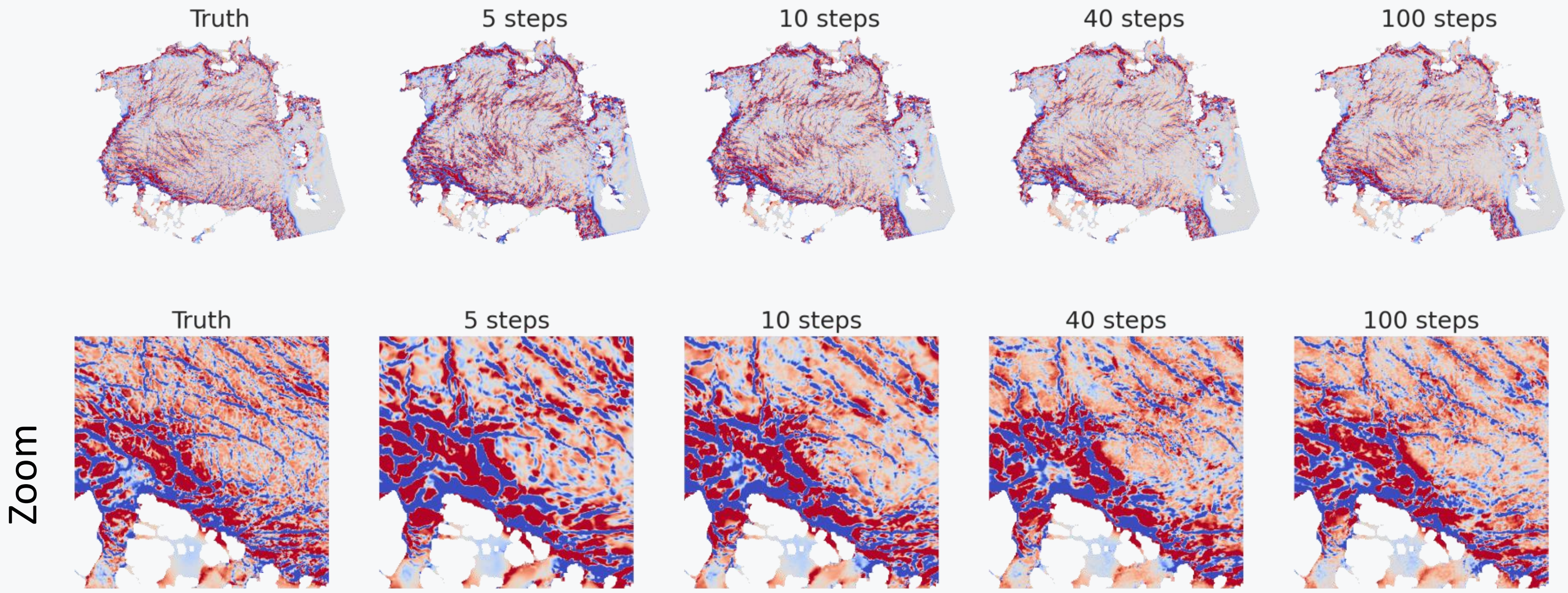


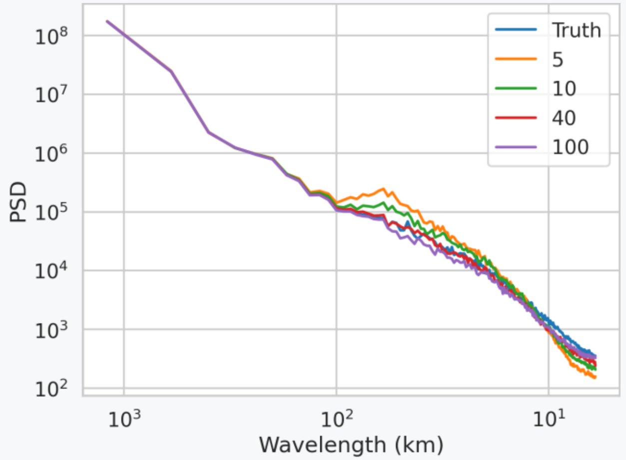
Spectral ratio = PSD-reconstruction / PSD-truth



Link to 23 October 2020

Sensitivity to the number of diffusion steps



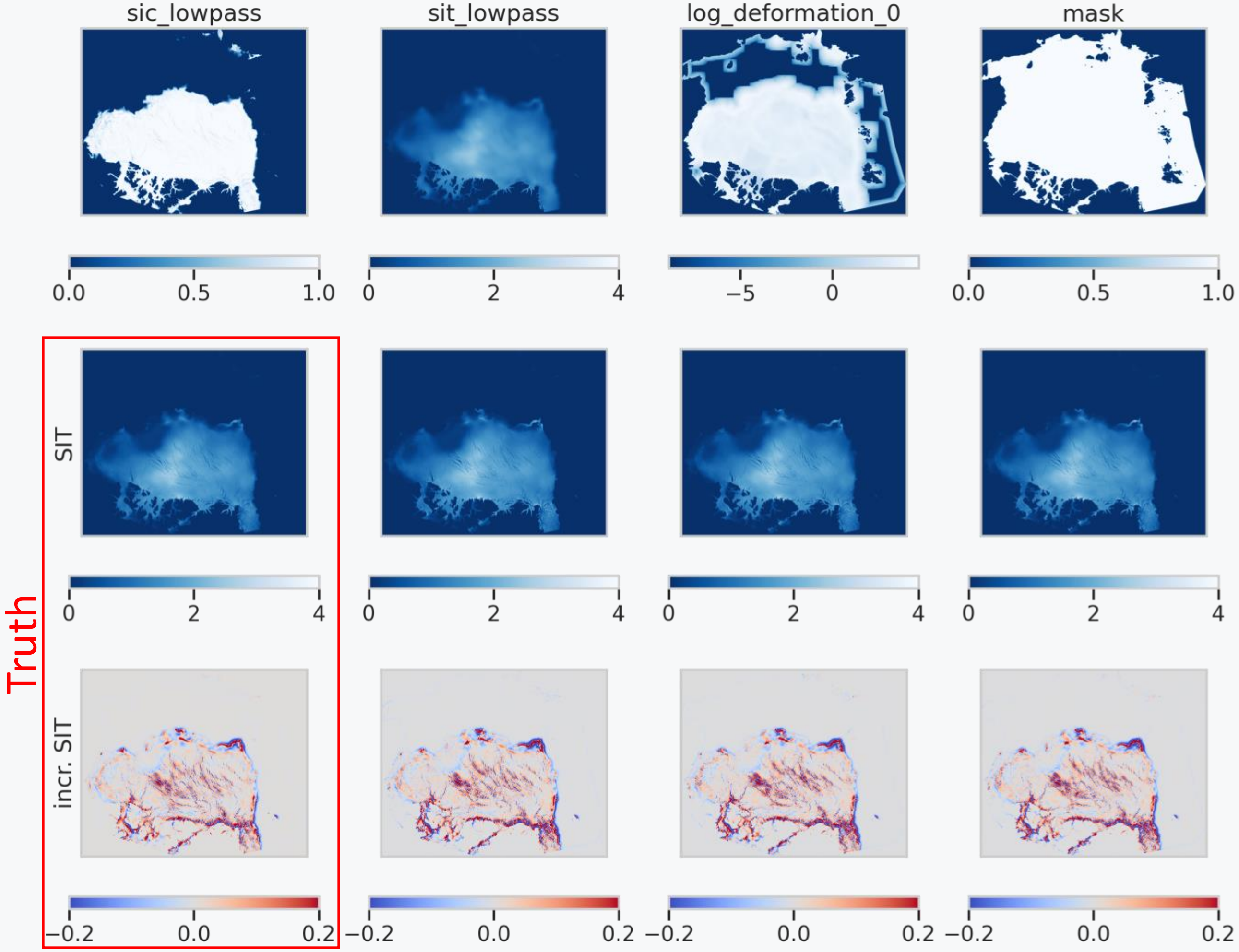


In the other sections, 40 steps are used

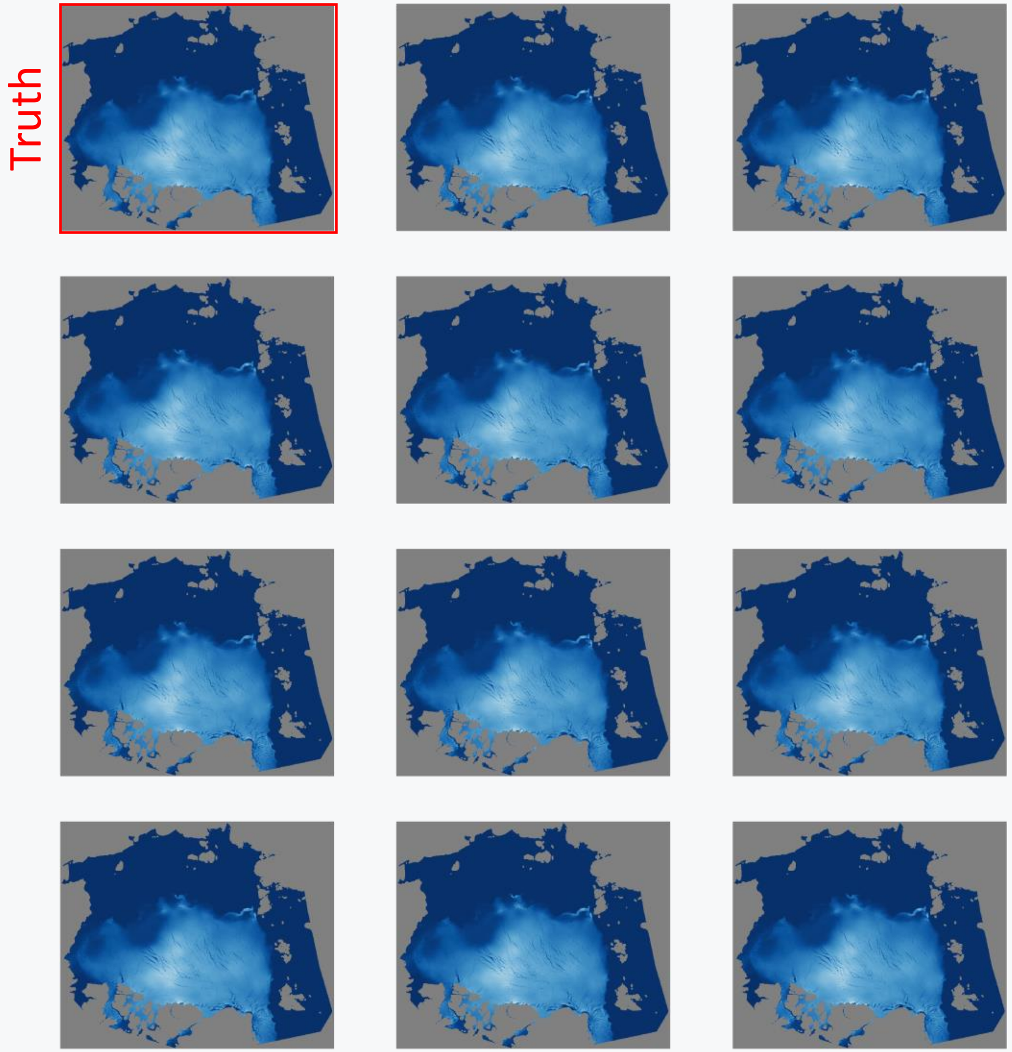
Link to 26 January 2021

Results of 23 October 2020

Generation of high-res SIT and residuals



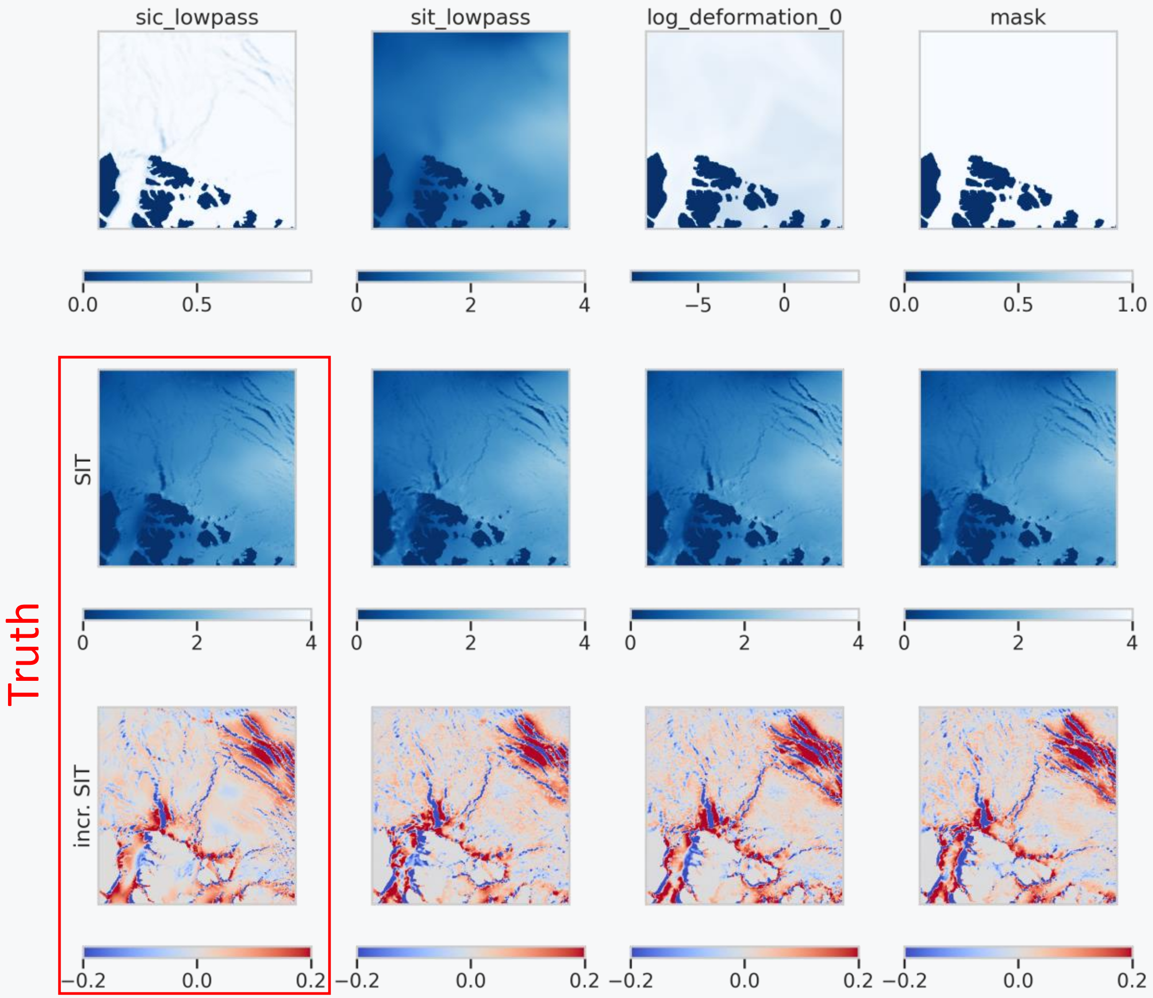
Generation of an ensemble



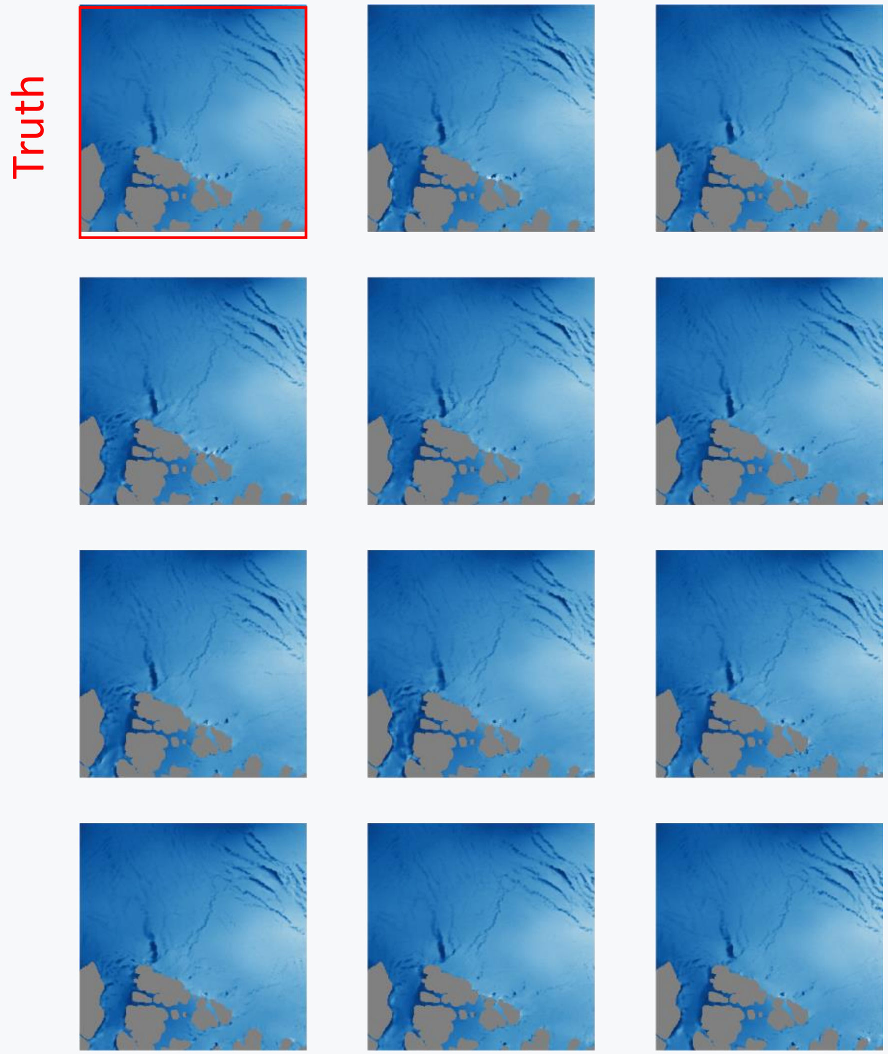
Link to 26 January 2021

Results of 23 October 2020 (zoom)

Generation of high-res SIT and residuals



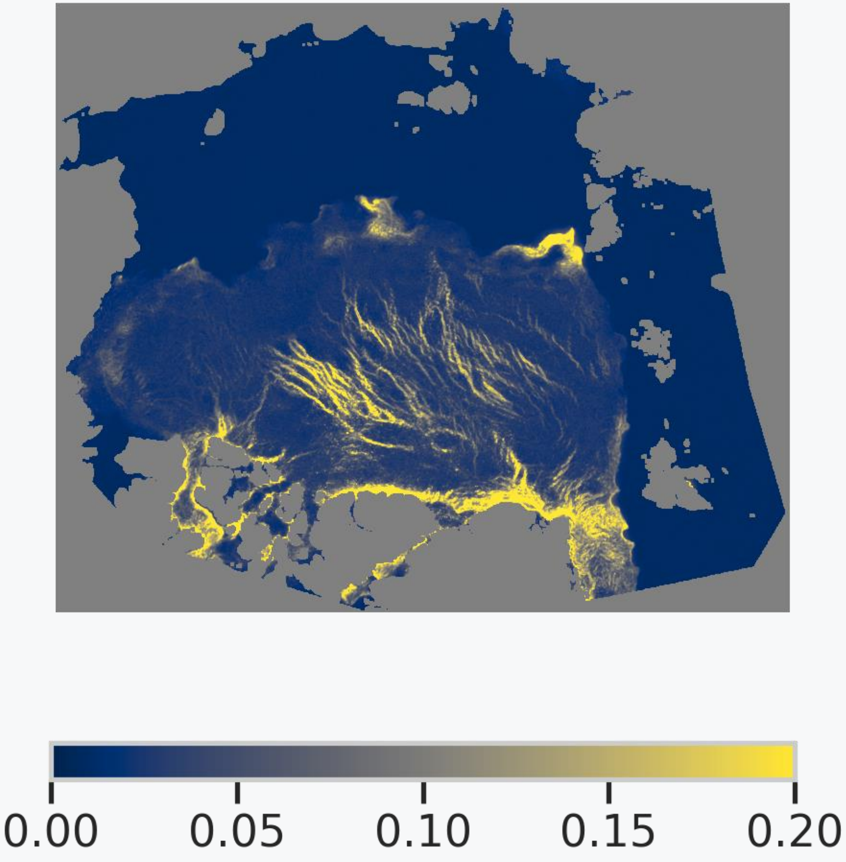
Generation of an ensemble

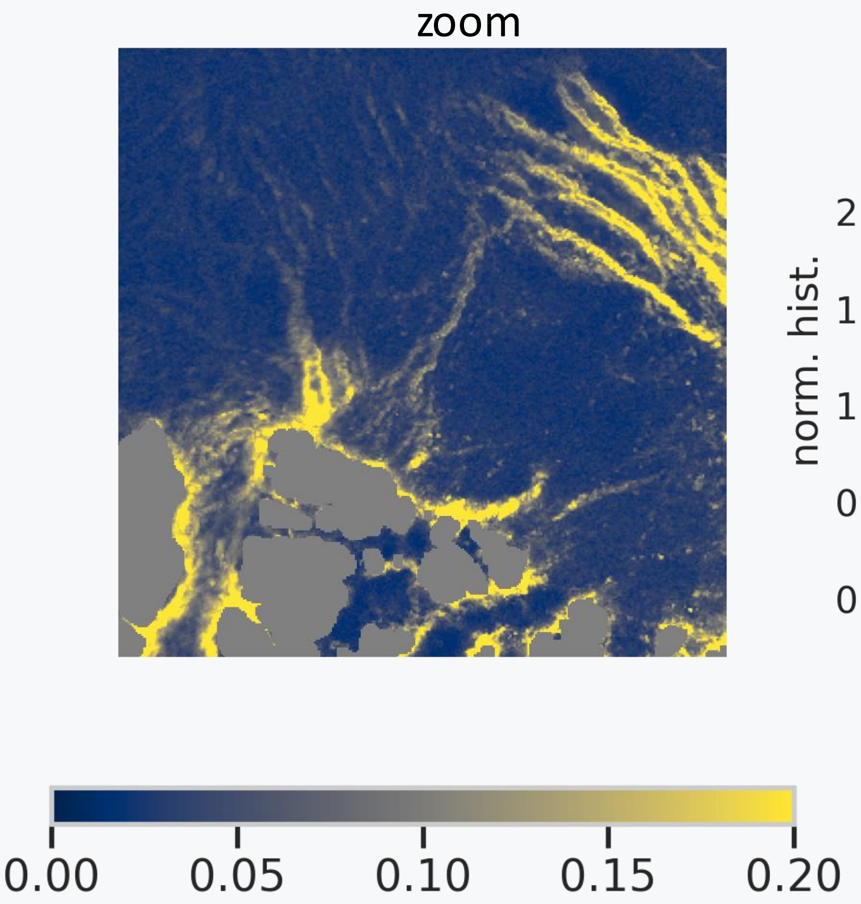


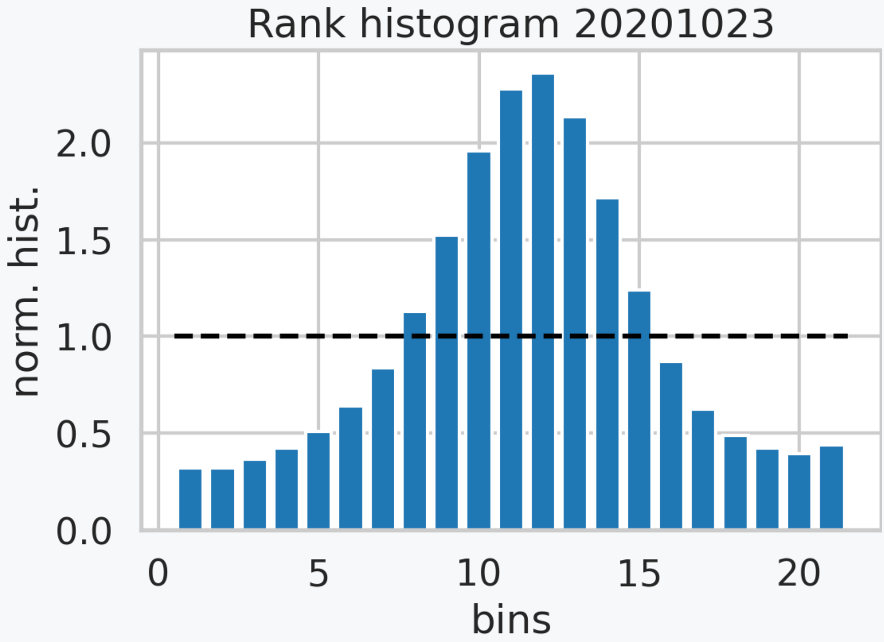
Link to 26 January 2021

Results of 23 October 2020

Spread and reliability





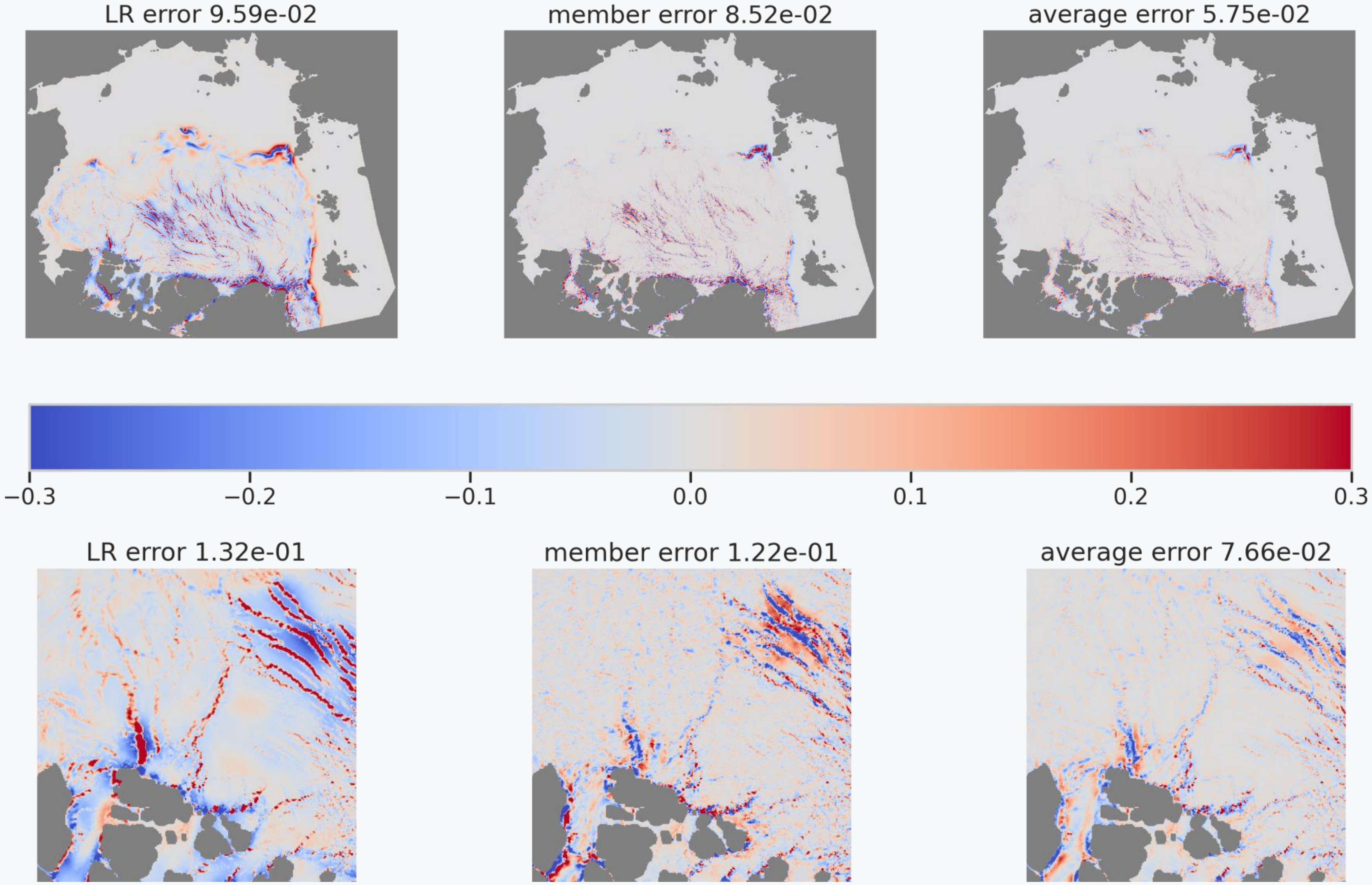


Ensemble over-dispersive

Link to 26 January 2021

Results of 23 October 2020

Accuracy

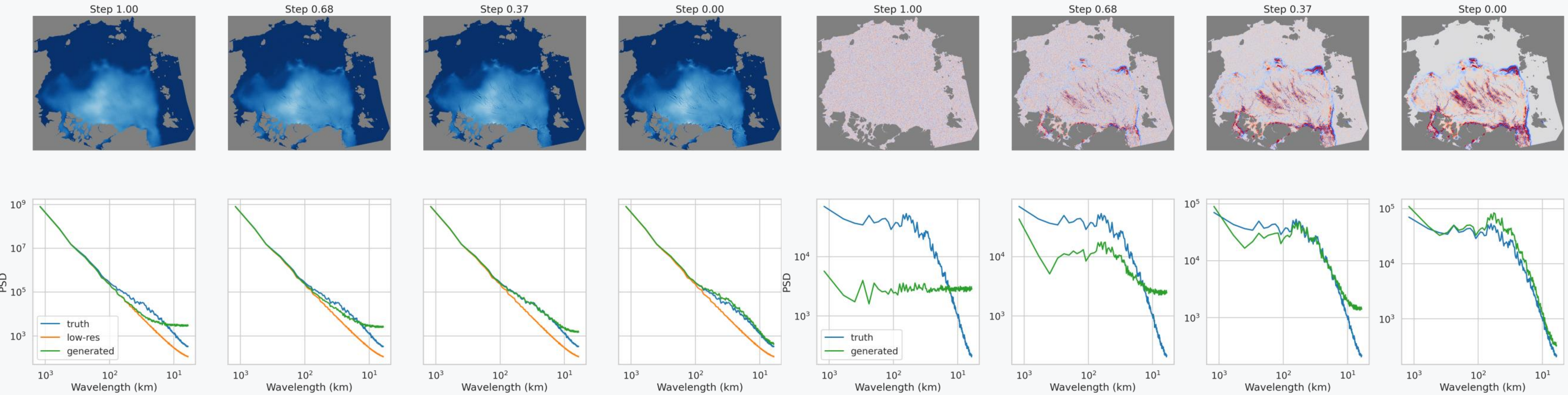


zoom

Link to 26 January 2021

Spectrum 23 October 2020

Power Spectrum Density as a function of the diffusion time (Step)

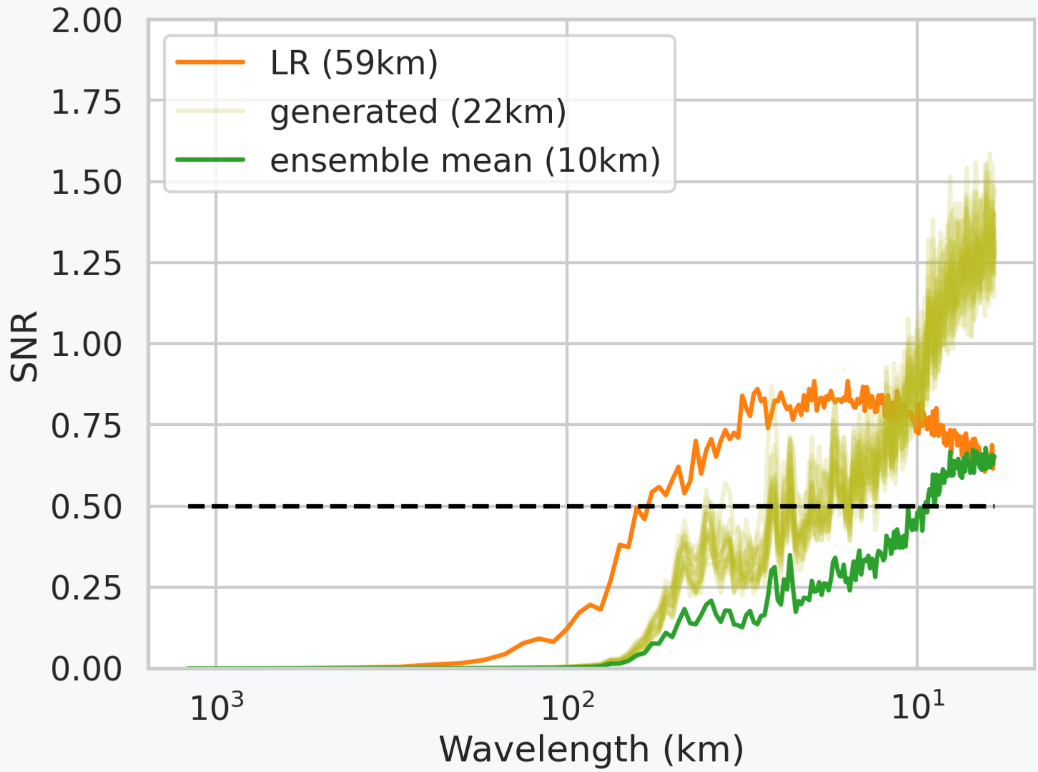


Link to 26 January 2021

Spectrum

Signal to noise

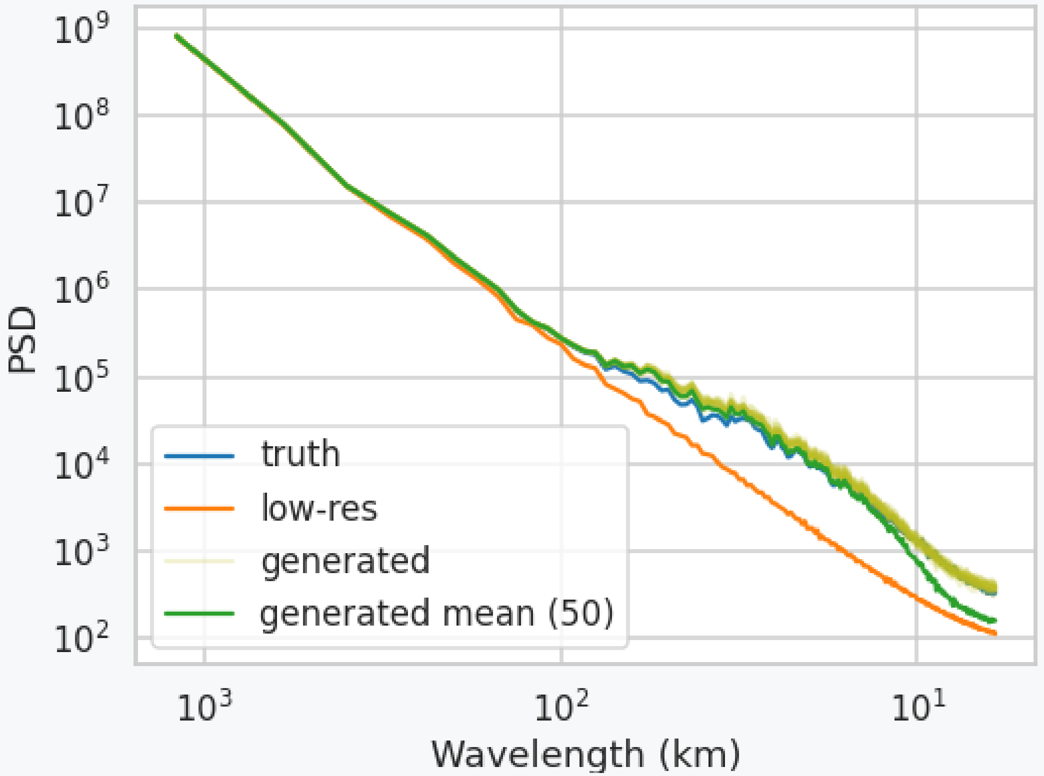
Spectral ration = PSD (reconstruction-truth) / PSD(truth)



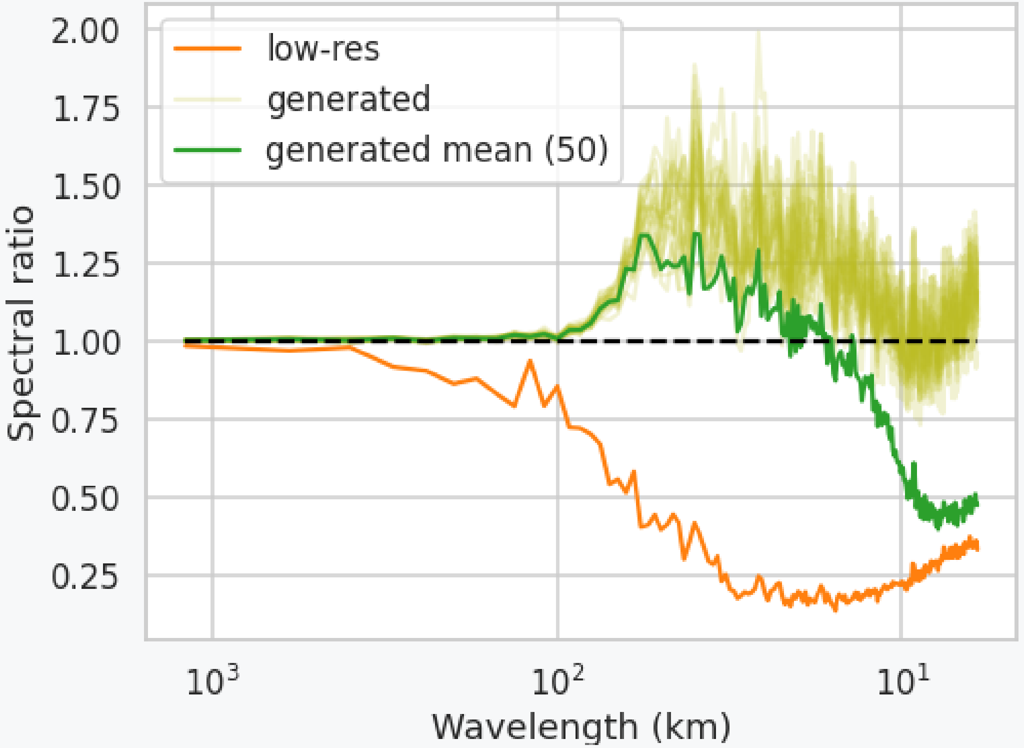
Link to 26 January 2021

Spectrum

Power Spectrum Density

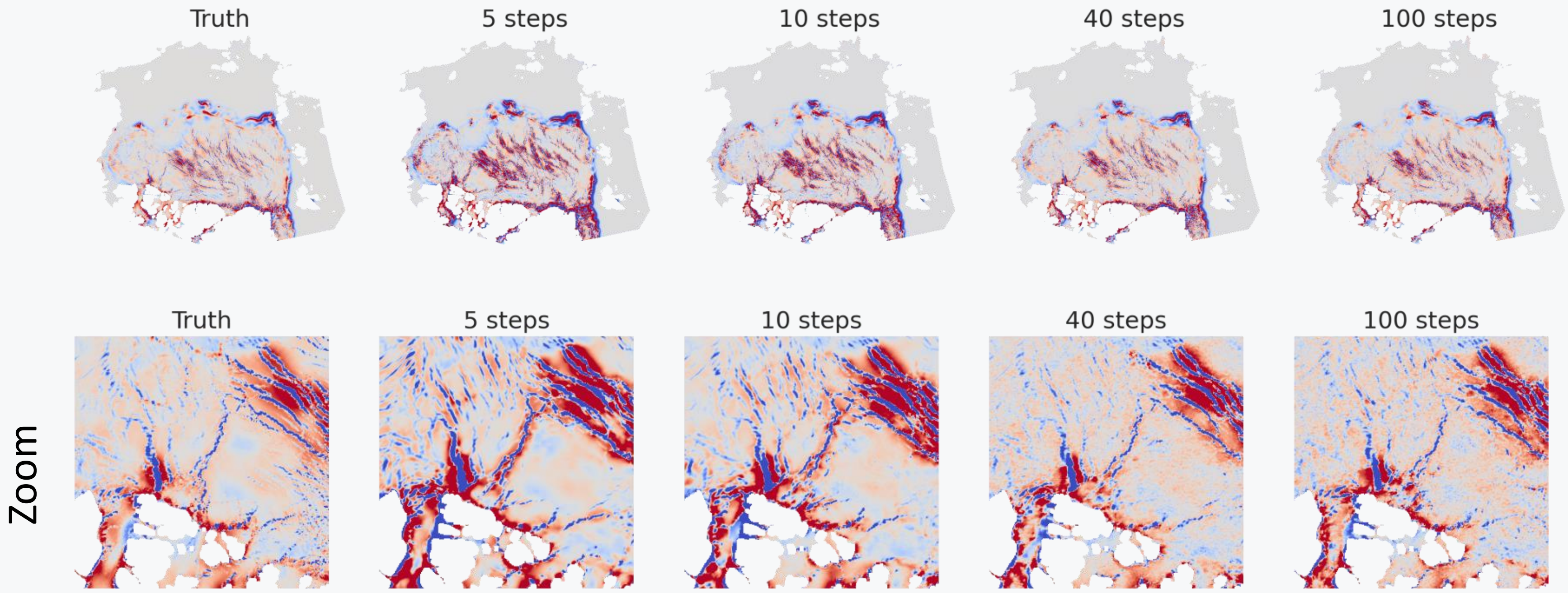


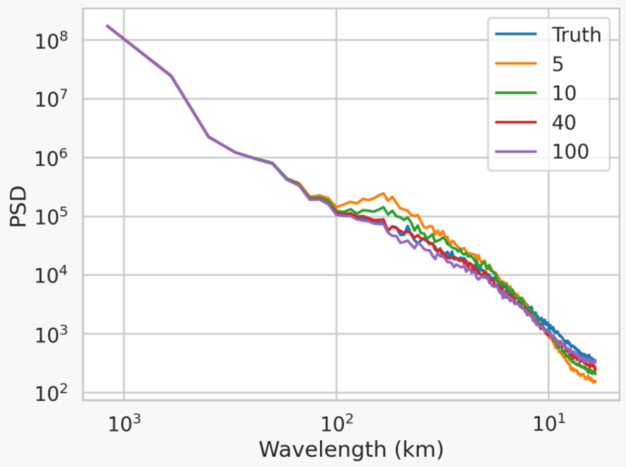
Spectral ratio = PSD-reconstruction / PSD-truth



Link to 26 January 2021

Sensitivity to the number of diffusion steps

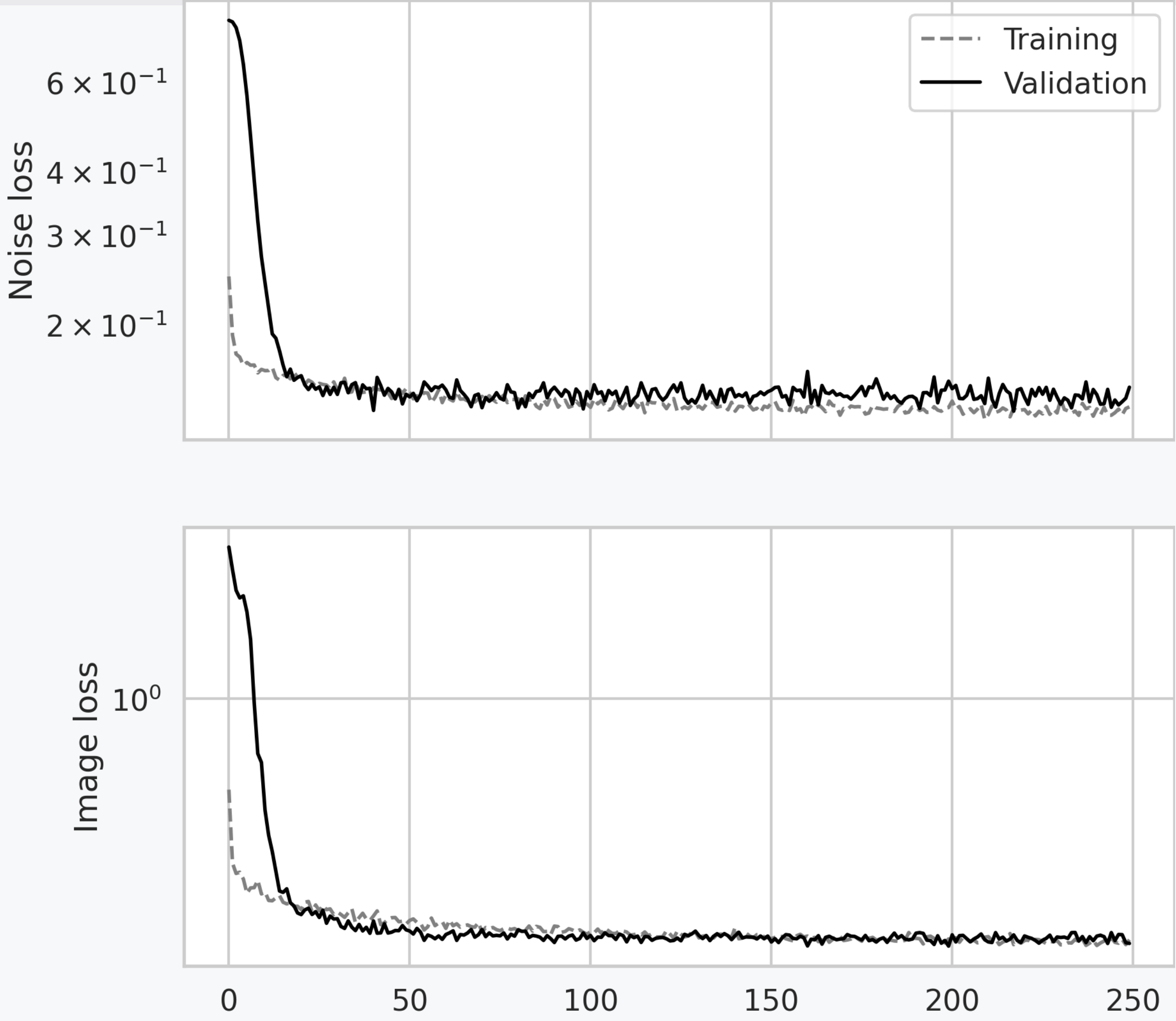




In the other sections, 40 steps are used

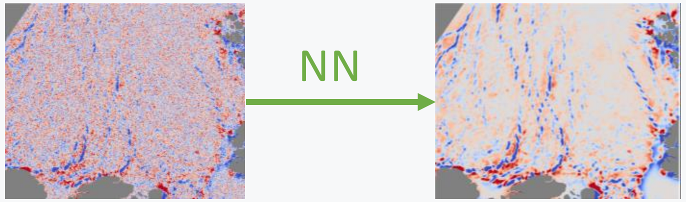
Globals results

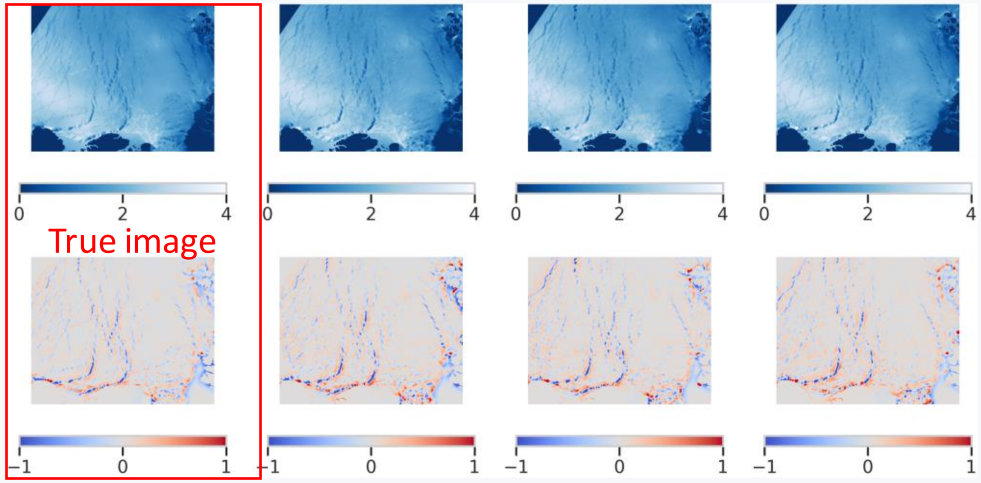
Learning curve



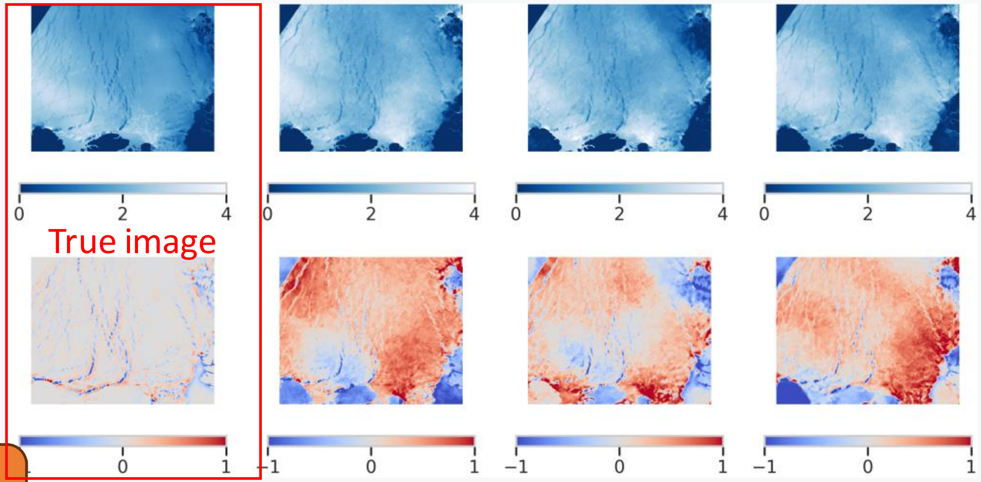
Residual Vs full field generation

Residual generation





Full-field generation



Full-field induces large-scale biases