



SuperIce project

Deliverable 3.1

AI-based generation report

Version 2.0

20.01.2025

ESA Contract No. 4000142335/231/I-DT · Physically-based synthetic data for Earth observation (SD4EO)



History of Changes		
Version	Publication date	Changes
1.0		Initial Version
1.1	13.08.24	Adding SSIM metric
2.0		Add model tuning, observational data, update results

Contributors	
Authors	Julien Brajard
Contributors	
Reviewers	Anton Korosov

1. Summary of main results

This document introduces an AI-based simulator designed to enhance low-resolution sea ice data. The simulator takes observable low-resolution fields—such as sea ice concentration, thickness, and deformation—as input and generates an ensemble of high-resolution sea ice thickness fields. The report begins with a description of the dataset used for training, validation, and testing the AI model. This is followed by an explanation of the training and inference processes. Finally, the AI algorithm is applied to both physical model-generated data and observational data.

Results demonstrate that the AI-based simulator produces sea ice thickness fields that are significantly more accurate and realistic than the original low-resolution fields, particularly at smaller spatial scales. This improvement is also evident when applied to observational data, although limitations, such as missing regions due to observational masking and reduced small-scale reconstruction compared to physical model data, are noted.

All data generated in this project can be accessed following the instructions available at:
https://github.com/brajard/superice_data.

2. Preparation of the dataset

The dataset used to calibrate and validate the AI-based generator is described in the deliverable 2.1 “Report on neXtSIM simulation and validation”[1]. This dataset contains fields of sea ice concentration (SIC), thickness (SIT), divergence (Div), and shear (Shear) for the Arctic Ocean between 2013 and 2023, during the freezing season (October 18 to April 15). The fields are defined on a 3km spatial grid (total size *1086x1308 points*) but the practical resolution has been degraded to mimic those of the satellite observation (see [1]). The high-resolution SIT (SIT_HR) is also used as a ground



truth to train the neural network. Table 1 summarizes the fields in the dataset together with their practical resolution. The divergence and shear are combined into a total deformation (Def) field:

$$Def = \sqrt{Div^2 + Shear^2} \quad (1)$$

Table 1 summarizes the variables used as input features and target

Acronym	Long name	Practical resolution
Input features		
SIC	Sea Ice concentration	15km
SIT	Sea ice thickness	90km
Def	Deformation	120 km
mask	Land mask	3 km
Target		
SIT_HR	High-resolution sea ice thickness	3km

The mask is equal to 0 on the land and ocean and 1 over the sea ice.

The dataset is split between training, validation, and test set. The training set includes samples between 2013 and April 2020 (1157 samples), the validation set between October 2020 and April 2022 (360 samples), and the test set between October 2022 to April 2023 (180 samples)

Before feeding the input features in the AI model, we standardize them. First, the deformation is log-transformed: $lDef = \log(Def)$. Then, each input feature except the mask is transformed using the following formula:

$$x_{\text{norm}} = \frac{x - m(x)}{\sigma(x)}, \quad (2)$$

Where x stands for a variable among (SIC, SIT, Def), x_{norm} is the corresponding normalized variable, $m(x)$ and $\sigma(x)$ are the mean and standard deviation of the variable computed over the training dataset.

The target variable is also processed, instead of predicting the raw SIT_HR, we predict the increment ΔSIT defined as

$$\Delta SIT = SIT_{\text{HR}} - SIT \quad (3)$$

So, when the AI-based generator predicts the value $y = \Delta SIT$, we reconstruct the final SIT_HR by adding back the low-resolution SIT specified as input: $\Delta SIT + SIT$.

The target value is also normalized following Eq.(2).

3. Training of the diffusion model



For training the diffusion we adapted a continuous version of the Denoising Diffusion Implicit Models (DDIMs) [2] by adding a conditional part. We train a UNET, denoted f_{θ} , taking different types of input features:

- A context \mathbf{x} , corresponding to the input features, containing the normalized input features described in section 2 of this report.
- A noisy image \mathbf{z} that is produced following a procedure described hereafter
- A scalar diffusion time $0 < t < 1$

The neural net is a standard UNET composed of 4 residual blocks of convolution layers for the encoding and decoding part. The size of the parameters vector θ is approximately 3.9 millions.

Computation of the loss function

We describe here the procedure to compute the loss function for one sample of the training set:

1. Randomly (uniform) draw a pair of normalized input features and target (\mathbf{x}, \mathbf{y}) from the training dataset described in section 2.
2. Randomly draw a scalar diffusion step t between 0 (full signal) and 1 (full noise)
3. Draw a Gaussian 2D noise ϵ of mean 0 and variance 1 as the same dimension as the input features.
4. Compute a diffusion angle: $\gamma = \gamma_{min} + t \cdot (\gamma_{max} - \gamma_{min})$
5. Compute the signal and noise rate: $r_s = \cos \gamma, r_e = \sin \gamma$
6. Compute a noisy image: $\mathbf{z} = r_s \cdot \mathbf{x} + r_e \cdot \epsilon$
7. Predict the output of the neural network $\epsilon' = f_{\theta}(\mathbf{z}, t, \mathbf{x})$ (note: in the UNET, t is embedded in a space of the same dimensions as the other inputs \mathbf{z}_t and \mathbf{x} using sinusoidal embedding [3])
8. Compute the loss: $l(\theta) = \|\epsilon - \epsilon'\|^2$

Training epoch

For the training, a traditional mini-batch gradient descent is used. For each iteration, the procedure described here below is applied for a batch of 4 samples, and then the parameters θ of the UNET are updated. Iterations are repeated until every sample of the training is drawn, which constitutes an epoch. The relatively small size of the batch can be explained by the large size of the input features ($1086 \times 1308 \times 5 \approx 8.5 \cdot 10^6$)

Training characteristics

The model is trained over 250 epochs using the Adam optimizer. The training was done on the LUMI supercomputer (<https://lumi-supercomputer.eu/>) using 2 AMD MI250x GPUs.

Field Code Changed

Figure 1 shows both the loss computed on the noise and on the reconstructed image for the training and the validation dataset. We can see that we reach the convergence without detected overfitting

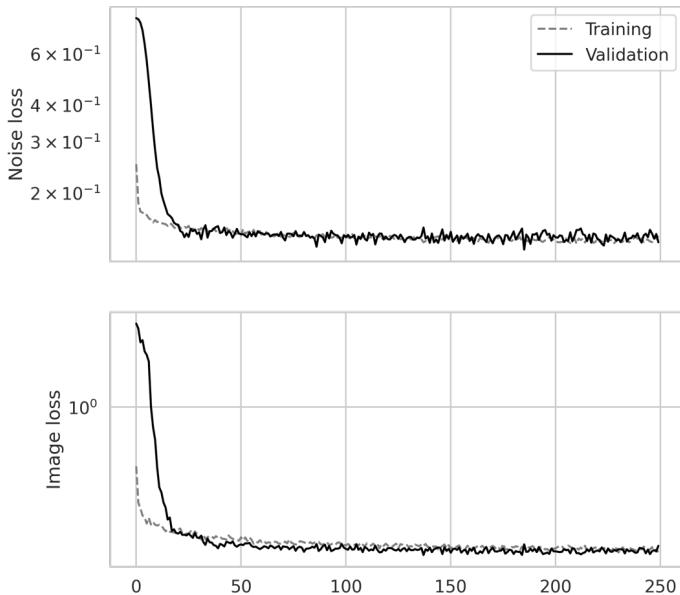


Fig. 1 Training and Validation losses over 250 epochs for the noise (upper panel) and the reconstructed image (lower panel)

4. Inference

Once the UNET is trained, it can be used to generate an output from a given input feature \mathbf{x} containing the fields SIC, SIT, Def, and mask (described in section 2). Since the inference relies on an input noise ϵ , each inference produces a different output, so the procedure can be used to generate an ensemble.

We describe here the inference procedure to generate one member of the ensemble from the UNET f_θ . The procedure being iterative we need to set the number of diffusion steps N_d . The input feature is denoted \mathbf{x}

1. Draw a Gaussian 2D noise ϵ of mean 0 and variance 1 as the same dimension as the input features
2. Initialize $\mathbf{z} = \epsilon$
3. For s from 0 to $N_d - 1$
 - 3.1. Compute $t = 1 - s/N_d$
 - 3.2. Compute the rates r_s and r_ϵ from t following steps 3 and 4 of the training algorithm (section 3)
 - 3.3. Predict the noise with the trained U-net $\epsilon = f_\theta(\mathbf{z}, t, \mathbf{x})$



- 3.4. Predict the image $\mathbf{y} = \frac{\mathbf{z} - r_e \epsilon}{r_s}$
- 3.5. Predict the next diffusion time $t = 1 - (s + 1)/N_d$ and the corresponding rates r_s and r_e
- 3.6. Predict the next noisy image $\mathbf{z} = r_s \cdot \mathbf{y} + r_e \cdot \epsilon$
4. Return the image \mathbf{y}

Figure 2 shows an example of SIT_HR generated on the whole Arctic. Figure 3 is a zoom of the same image which allows to better distinguish the details of the SIT_HR produced by the AI-based simulator. The images were produced with a number of diffusion steps $N_d = 40$. Sensitivity studies show low sensitivity to the number of diffusion steps.

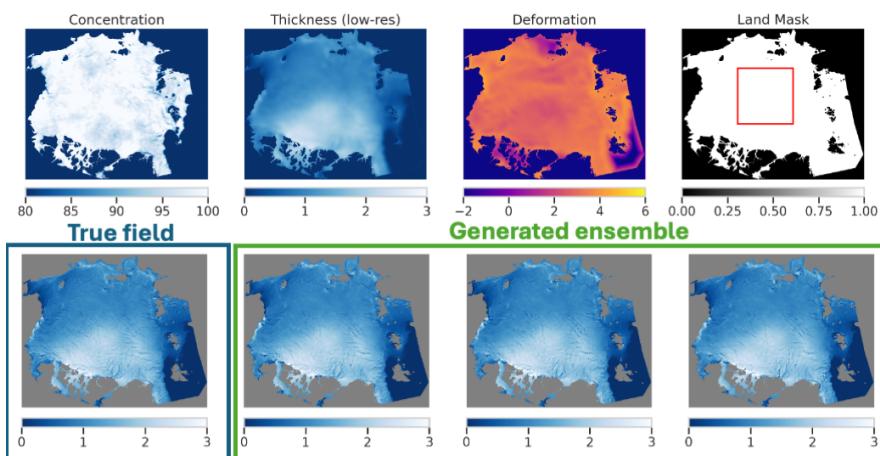


Figure 2. Example of 3 members generated by the AI-based generator for 1 January 2021. The upper row represents the input feature at low-resolution, the red box in the Land Mask figure corresponds to the zoom presented in Figure 3. In the lower row we present the true image from the physical simulator (blue box) together with 3 members generated by the AI-based simulator.

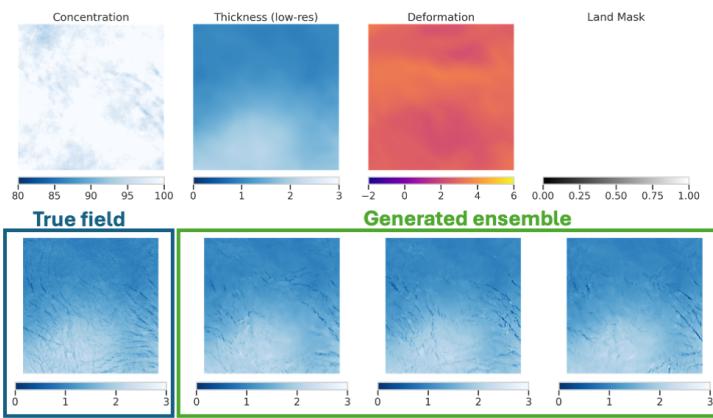


Figure 3. Same as Figure 2 but a zoom corresponding to the red box presented in Fig.2.

5. Metrics

The metrics used to validate the high-resolution Sea Ice Thickness (SIT) produced by SuperIce are described below. Two categories of metrics are applied:

1. **Accuracy metrics (A):** These evaluate the accuracy of the reconstructed product and are computed based on the average of 30 high-resolution members generated by the AI-simulator.
 2. **Realism metrics (R):** These assess the realism of the product and are computed for each individual member.
- **Root-mean-square error (RMSE) (A):** Measures the accuracy of the reconstruction. Lower values indicate better performance.
 - **Ratio of spatial variability (R):** Evaluates whether the reconstructed field captures spatial variability. It is calculated as the ratio of the standard deviations of the reconstructed and reference fields. A ratio > 1 indicates overestimation, while a ratio < 1 indicates underestimation. The optimal value is 1.
 - **Fine-scale bias (R):** Assesses biases in fine-scale variability by comparing the power density spectra of both the reconstructed and the true fields for scales below 20 km. The average ratio of the two spectra indicates overestimation (ratio > 1) or underestimation (ratio < 1). The optimal value is 1.
 - **Structural similarity index measure (SSIM) (A, R):** was included in a previous version of the report. However, it has been excluded here as it was found to be insufficiently sensitive across the different models considered..



6. Model selection

In this section, we assess the sensitivity of the results to some hyper-parameters of the model training and inference. To simplify the reading, only RMSE and Finae-scale bias are reported in this section. The ratio of spatial variability (not shown) leads to the same conclusions. The table 2 lists the considered hyper-parameters:

Parameters name	Description	Value tested	Reference value
Training uncertainty	Several training with exactly the same hyper-parameters can differ due to randomness in the processs.	3 trainings	N/A
Size of the model	The number of unit in each layers of the UNET	[32,64,96,128], [64,64,96,128]	[32,...]
Input features	Combinations of input features including the sea ice thickess	All four combinations	All features
Min_rate	Minimum signal rate: $\cos \gamma_{min}$	0.01, 0.012, 0.015, 0,02	0.02
Max_rate	Maximum signal rate: $\cos \gamma_{max}$	0.95, 0.97,0.99	0.95
Diffusion steps	Number of diffusion steps during inference N_d		40

Table 2. hyper-parameters and list of respective values considered to select the final model.

6.1 Sensitivity to different initialization and the size of the model.

We conducted limited experiments with models of varying sizes, focusing exclusively on four-layer architectures. The size of each layer in the UNET was adjusted, and we present results for two configurations: [32, 64, 96, 128] and [64, 64, 96, 128]. The metrics, displayed in Fig. 4, indicate that the results are generally stable, with the exception of one training instance (the first in the figure), which converged to a suboptimal local minimum. Overall, the impact of model size is minor and falls within the uncertainty associated with training randomness.

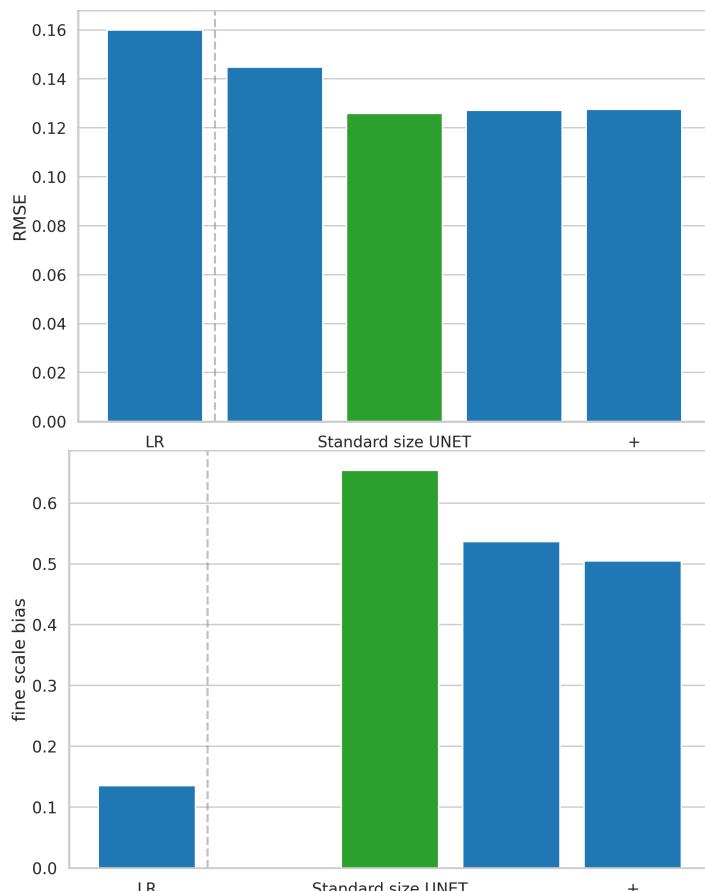


Figure 4. RMSE (upper panel) and fine-scale bias (lower panel) for different reconstructed field: the low-resolution field (LR), several trainings with standard sizes (i.e., [32, 64, 96, 128]), and one training (indicated by a "+" on the x-axis) with a larger model (i.e., [64, 64, 96, 128]). The first training produced unrealistic values, leading to NaN (Not-a-Number) in the computation of the fine-scale bias, which explains the missing metric. The green bar represents the final model used in this study.

6.2 Sensitivity to the input features

Several training experiments were conducted using different input features, including sea ice thickness and all possible combinations of concentration and deformation. Figure 5 indicates that



incorporating all features results in only a minor improvement in RMSE. Surprisingly, adding deformation does not appear to enhance either accuracy or realism. A model based solely on thickness and concentration could be considered. However, we chose to retain deformation, as we believe this variable holds significant potential for improvement if high-resolution deformation fields, such as those derived from SAR, are utilized in the future.

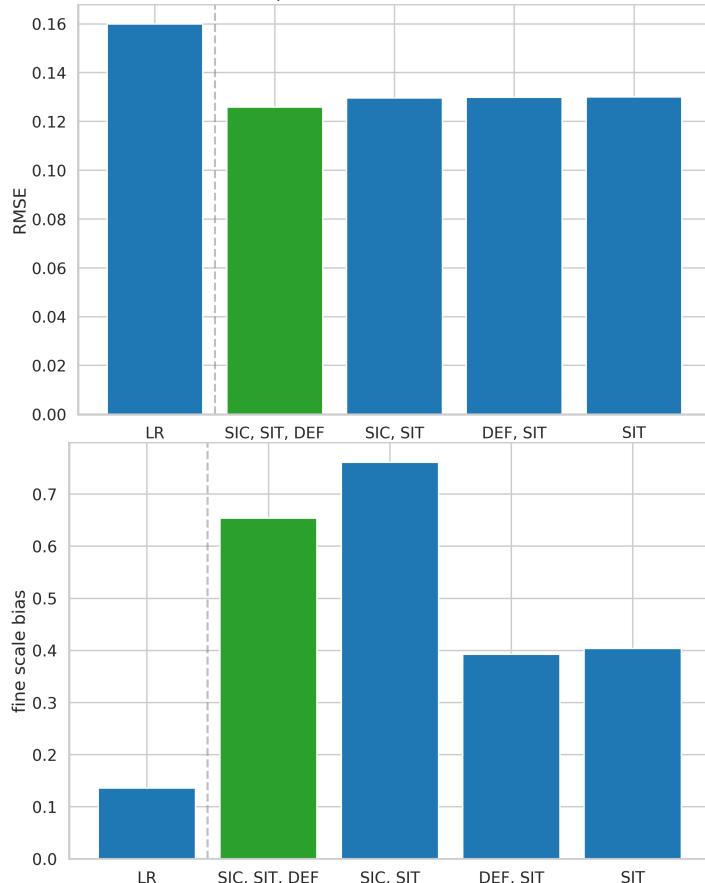


Figure 5. RMSE (upper panel) and fine-scale bias (lower panel) for various combinations of input features, including SIC (sea ice concentration), SIT (sea ice thickness), and DEF (sea ice deformation). The green bar represents the final model used in this study.

6.3 Maximum or minimum signal rate



During training, we defined two values that set the limits of noise that can be added to the signal, controlled by the hyperparameters `min_rate` and `max_rate`. For example, a high `max_rate` allows for the inclusion of samples with low noise. We explored these hyperparameters because they were identified as sensitive in the reference model we used [2]. The results, presented in Fig. 6, confirm that these hyperparameters are indeed sensitive, but their specific effects are challenging to interpret. For certain values, the AI simulator generates either too many or too few small-scale features. Our chosen reference values represent a compromise, balancing low error with an accurate representation of small-scale features.

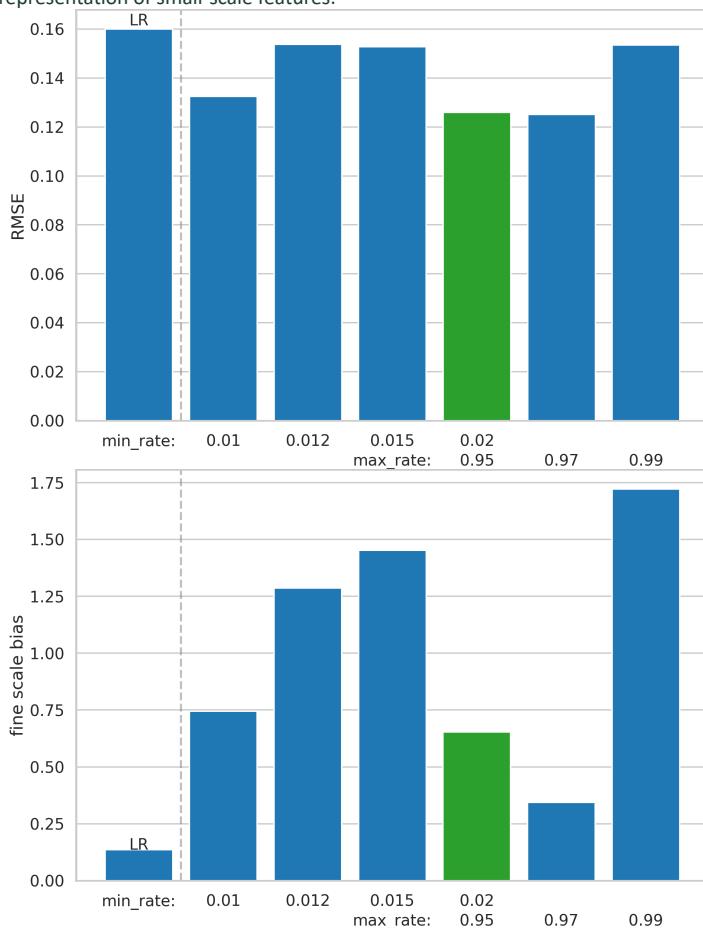


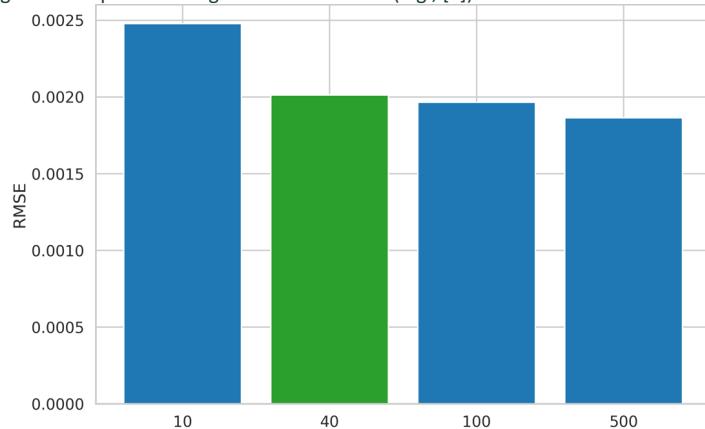
Figure 6. RMSE (upper panel) and fine-scale bias (lower panel) for different values of `min_rate` and `max_rate` during training. The specific values are shown on the x-axis labels. If no values are



specified, the reference values (0.02 for min_rate and 0.95 for max_rate) are used. The low-resolution field (LR) serves as a baseline.. The green bar represents the final model used in this study.

6.4. Number of diffusion steps

The hyperparameter "diffusion steps" is applied only during inference. Due to computational constraints, the presented scores were calculated for January 1, 2021; however, additional tests on other dates confirm that the conclusions hold across all dates. Figure 7 demonstrates that increasing the number of diffusion steps improves both accuracy (RMSE) and realism (fine-scale bias), although the RMSE improvement becomes marginal beyond 40 steps. Considering the computational cost of 500 steps compared to 40 steps (an overhead factor of 12.5 per member), we used 40 steps to generate the dataset in this study. As a follow-up, exploring additional steps or optimizing the generation process using alternative models (e.g., [4]) could be worthwhile.



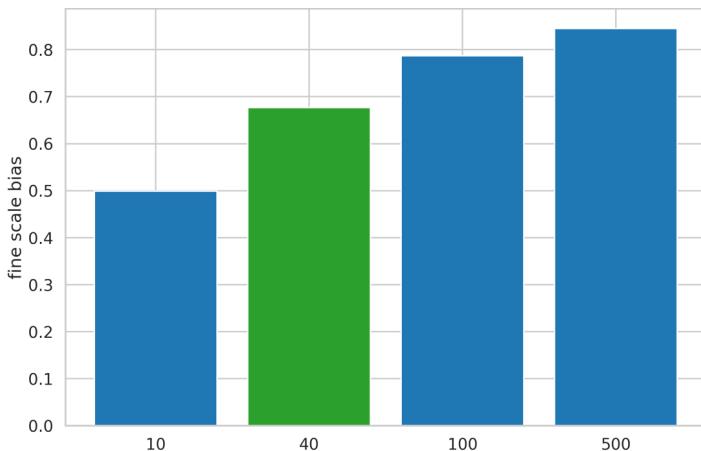


Figure 7. RMSE (upper panel) and fine-scale bias (lower panel) for different number of diffusion steps ranging from 10 to 500. The green bar represents the final model used in this study.

Selected model

The final selected model is a UNET with the following hyper-parameters:

- 4 layers of size [32,64,96,128]
- Minimum signal rate during training: 0.02
- Maximum signal rate during training: 0.95
- Number of epochs: 250
- Number of diffusion steps at inference: 40
- Code of the model: C_DiffusionModel in <https://github.com/nansencenter/SID-NN/blob/stoch/diffusion.py>

7. Performance on model data

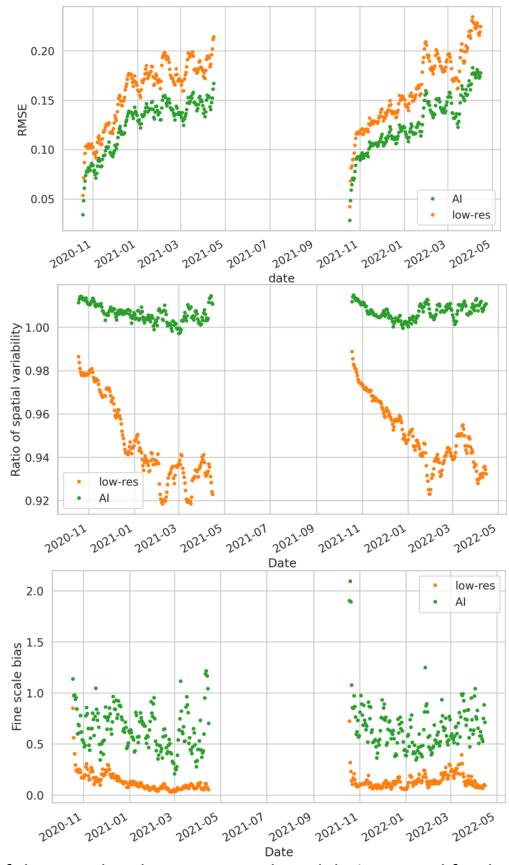


Figure 8. Evolution of the considered metrics over the validation period for the AI-based simulation (green) and the low-resolution field (orange). The upper panel shows the RMSE (lower values indicate better performance), the middle panel the ratio of spatial variability (values closer to 1 are better), and the lower panel the spectral bias (values closer to 1 are better).

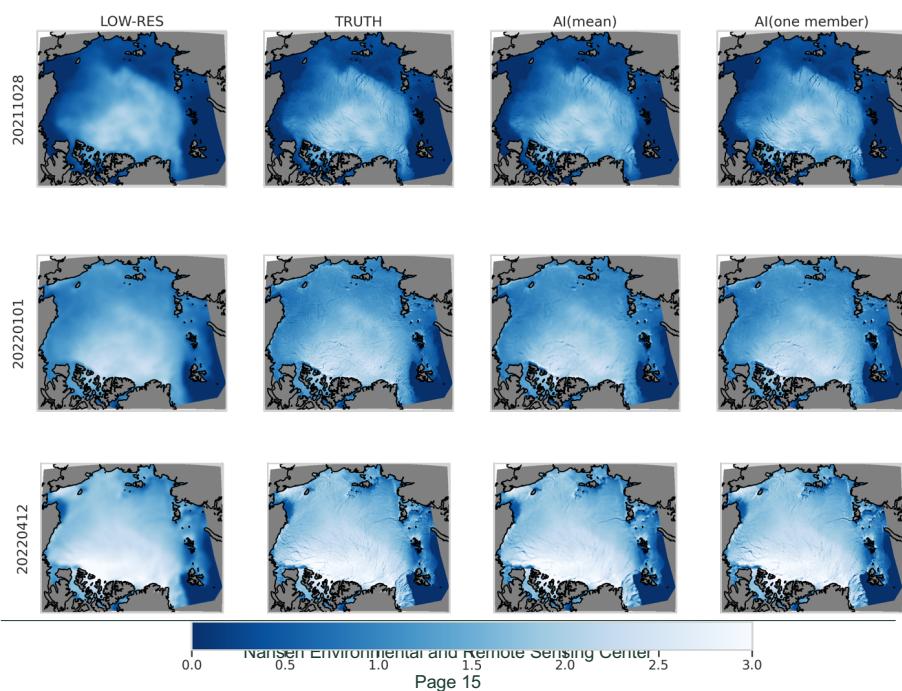




Figure 9. Example of SIT of the low-resolution field (1st column), the reference high-resolution neXtSIM simulation (2nd column), the average over 30 AI-based simulations (3rd column) and an arbitrary AI-based simulation (last columns). Three dates have been selected at the beginning, the middle, and the end of the freezing period.

We present the results of the new AI simulator applied to the validation dataset, evaluated using the metrics defined in Section 5. In addition to these global metrics, we also provide results for three individual days selected from the beginning, middle, and end of the testing period. For each metric, the AI-based reconstruction is compared against a baseline: a simple interpolation of the SIT low-resolution field.

Figure 8 displays the RMSE, spatial standard deviation, and fine-scale bias. The RMSE of the ensemble mean of the AI simulation (green line) is consistently lower than that of the low-resolution field, indicating that the AI simulation is more accurate. The seasonal increase in RMSE over time can be attributed to a size effect, as the total extent of sea ice grows during the freezing period.

Regarding spatial variability (computed for each simulation rather than the average), the AI simulations are closer to the true reference field than the low-resolution field, which strongly underestimates spatial variability due to its smoothing effect. However, the AI simulations exhibit a slight and systematic overestimation of spatial variability, likely caused by residual noise from the diffusion process. Additionally, there remains a negative bias for small scales, indicating that not all fine-scale signals are fully recovered. As discussed earlier, alternative models could improve small-scale reconstruction but would result in higher overall reconstruction errors.

Figure 9 provides visual examples of AI reconstructions for three selected days. These examples confirm the realism of the small-scale features, although slightly fewer small-scale structures are observed in the AI simulations (last column) compared to the reference field (second column), consistent with the

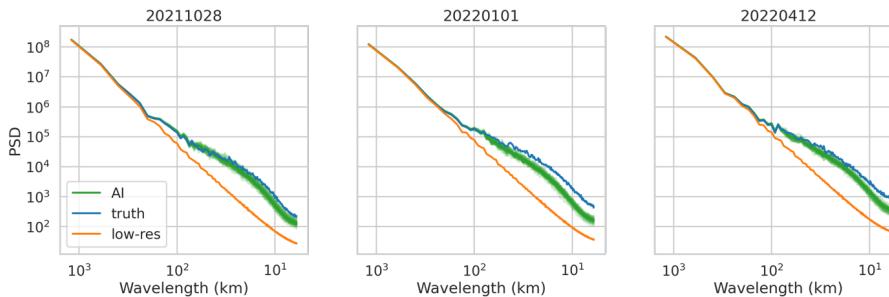


Figure 10. Power spectrum of the fields presented in Fig. 9 for the 3 selected dates 28 Oct. 2021, 1st January 2022 and 12 April 2022. The low-resolution spectrum is presented in orange, the true field in blue and the 30 AI-generated fields are in blue.

metrics in Figure 8.

This conclusion is further supported by **Figure 10**, which shows the power spectrum. The 30 AI-simulated members closely align with the true spectrum and outperform the low-resolution field,



although there is still a slight underestimation of small-scale features. Moreover, the spectra are highly consistent across different dates and among all generated members, demonstrating the robustness of the algorithm.

The data are available here: https://github.com/brajard/superice_data

8. Application to observational product

8.1. Application to observational product

We use the model defined in the box (end of Section 6) and apply it directly to the observations described in Deliverable 2.1 of the project. The input features consist of sea-ice thickness from CS2SMOS, as well as sea-ice concentration and drift from the OSISAF product. These products include pixels that remain unprocessed due to limitations in their processing algorithms, which are subsequently masked. In the input feature corresponding to the mask, a pixel is considered masked if it is masked in at least one input feature.

8.2. Correction for the interpolation noise

We corrected for error in the observational product due to the interpolation into our reference grid. For that we apply a light smoothing with a gaussian kernal of size 5x5 pixels. Figure 11 (upper row) shows the Laplacian of the sea ice thickness product for a small region for January 1st 2021-. The noise is visible, and it can be seen in the second row that it affects the AI-generated thickness (second row) field that display unrealistic features, which translate in a bump in the power spectrum at a scale of ~75km (3rd row of Fig. 11). It can be seen that after the little smoothing, this effect is corrected at the cost of less small scales. In the data delivered in th project, we apply this correction to generate our observational dataset

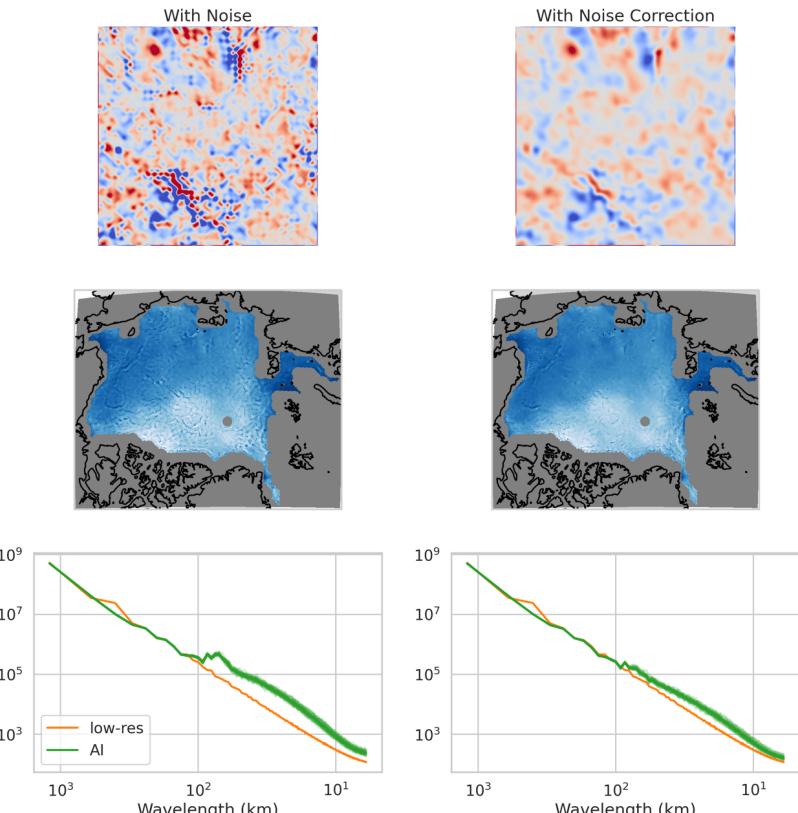


Figure 11 – The first row shows the Laplacian (spatial derivative) of the low-resolution sea-ice thickness, used as an input feature for a zoomed-in region in the middle of the domain. The second row shows an AI-generated field. The last row shows the power spectrum of 30 AI-generated members (in green) compared to that of the low-resolution sea-ice thickness field (in orange).

8.3. Results on the observationsIn

In figure, 12., wWe display the ensemble average and one member of AI-generated sea ice thickness from observations for 3 dates. First it can be seen that the AI algorithm successfully generates high-resolution feature that are not present in the low-resolution field, albeit with less variability than the model data product. This can be explained by the fact that there is a slight distributional shift between NeXtSIM-based feature and observational features. It can also be noticed, compared with the model product in Fig. 9 that there a large part of the domain that is masked. This is due to a large part that



unprocessed regions near the coast for the drift product and also to the polar hole. Additionally it can be seen that the mask appears to create a few unrealistic patterns following the coastline. This is due to the fact that the AI model was trained without the masks. We see two options to overcome this problem that could be developed as a follow-up of the project: (i) a preprocessing step could be added to the observational feature to interpolate in the masked regions; (ii) the AI model could be re-trained with overlapping masks.

Nevertheless, we have an observational-based sea ice thickness with an enhanced resolution. The benefits of such a dataset are described in the deliverable 1.1

The observational dataset is available at the following link: https://github.com/brajard/superice_data

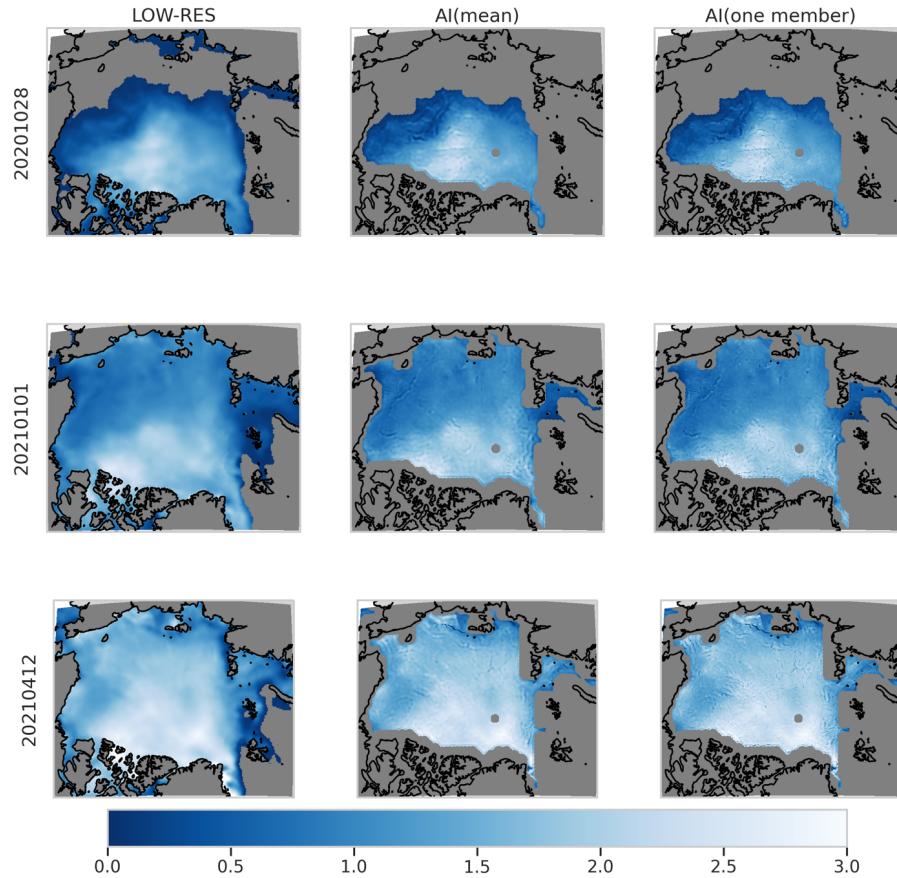




Figure 12 – Example of SIT of the CS2SMOS low-resolution field (1st column) the average over 30 AI-based simulations (2nd column) and an arbitrary AI-based simulation (last column). Three dates have been selected at the beginning, the middle, and the end of the freezing period.

9. References

- [1] Korosov, A., Mangini F., 2024, Deliverable 2.1 version 1.0, Report on NextSIM simulation and validation, Report of the ESA project SuperIce
- [2] Song, J., Meng, C. and Ermon, S., 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.
- [3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- [4] Song, Y., Dhariwal, P., Chen, M. and Sutskever, I., 2023. Consistency models. *arXiv preprint arXiv:2303.01469*