

CIP: TurboSWAN to assess confidence intervals for SWAN forecasts

Surrogate North Sea wave model to apply wind ensembles in wave forecasts



CIP: TurboSWAN to assess confidence intervals for SWAN forecasts
Surrogate North Sea wave model to apply wind ensembles in wave forecasts

Author(s)

Caroline Gautier
Elias de Korte
Joost den Bieman
Joana van Nieuwkoop
Guus van Hemert

CIP: TurboSWAN to assess confidence intervals for SWAN forecasts

Surrogate North Sea wave model to apply wind ensembles in wave forecasts

Client	Rijkswaterstaat Verkeer- en Watermanagement, locatie Westraven
Contact	Joost Driebergen
Reference	SITO-PS CIP project plan
Keywords	Surrogate model, AI, Machine Learning, SWAN, waves, wind ensembles, uncertainty, confidence interval

Document control

Version	1.0
Date	18-12-2025
Project nr.	11211520-009
Document ID	11211520-009-BGS-0002
Pages	36
Classification	
Status	Final

Author(s)

	Caroline Gautier Elias de Korte Joost den Bieman Joana van Nieuwkoop Guus van Hemert	

Summary

Within the Rijkswaterstaat Operational Systems (RWsOS), SWAN wave models are used to provide wave forecasts for both the Dutch coast and the larger lakes. On average these forecasts are quite good but the scatter in wave height can be rather large, especially for the low frequency wave height H_{E10} . There is a need for confidence intervals to the wave forecasts.

Therefore we have tried to set up TurboSWAN, a fast surrogate model trained on SWAN in- and output, which should be able to make wave simulations on the entire SWAN-North Sea domain for 50 ECMWF wind ensemble members within minutes. The result, being the spread in wave parameters H_{m0} , (and ultimately $T_{m-1,0}$ and H_{E10}) will be applied on the outcome of the one SWAN run which used the control wind field as input.

Initially, the dataset for training and testing (i.e. validation) came from SWAN-North Sea runs with input from the TIGGE project. Various experiments were done to improve TurboSWAN but the error statistics did not reduce significantly. Aspects considered in these experiments were time lag of input fields, focussing on part of the domain, definition of the loss function to be minimised, data augmentation, architecture (drop out layers, up- and downsampling of data, etc). The hypothesis that the training set was too uniform seemed to be false: changing to a more varied dataset (18 selected years out of the 45 year ERA5 dataset) did not result in the desired improvement.

TurboSWAN represents large-scale wave height features quite well. However, with typical RMSE values of approximately 0.6 m in significant wave height relative to the SWAN results for eight selected locations, it is not yet good enough for operational implementation. Apart from this rather large RMSE, there is no full confidence in the present set up of TurboSWAN as some experiments with a physically sound basis, did not lead to large improvements in the performance. SWAN has a typical RMSE of 0.3 m for significant wave height relative to observations (Deltares, 2023). There are various criteria to judge the experiments, i.e. RMSE, bias, for specific locations or for the entire domain and it is hard to define an overall best score. When using different training sets, the statistical scores like RMSE are not necessarily directly comparable.

TurboSWAN is very fast: 25ms to compute one time step. For eight 6-hr-time steps covering 48 hours, and fifty ensembles, it would take about 10 seconds on GPU (and approximately a factor 40 more on CPU) to compute the wave field. These time-steps can even be executed in parallel, contrary to SWAN.

The amount of publications on surrogate wave modelling is scarce, especially for spatial applications in a regional shelf sea. Given the convergence of improvements in the current experiments, we advise to decompose the experiments into smaller components and tests and use a modular step-by-step approach. For example, this could be combined with a timeseries-CNN (convolutional neural network) for the core stations allowing for fast experiments and transferring lessons learnt to the spatial CNN.

Contents

	Summary	4
1	Introduction	7
1.1	General	7
1.2	Objective	7
1.3	Main message	7
1.4	Set up of the report	7
2	Approach and data	8
2.1	Brief overview of previous findings (Deltares, 2024c)	8
2.2	User stories from Hydrometeo Centre (HMC)	8
2.3	Approach	8
2.4	Data	9
2.4.1	TIGGE dataset	9
2.4.2	ERA5 dataset	10
2.4.3	Observations	10
3	Literature study	11
3.1	Surrogate Modelling	11
3.2	Data-driven prediction of wave fields	11
3.3	The temporal dimension	11
3.4	Conclusion	12
4	Setting up TurboSWAN	13
4.1	Introduction TurboSWAN	13
4.1.1	Summary of experiments	14
4.1.2	Introduction experiments	16
4.2	Results	18
4.2.1	Loss curves	18
4.2.2	Validation results	19
4.2.3	Best TIGGE -trained model (V16)	21
4.2.4	V18 ERA5-trained	23
4.2.5	Effect of dataset	24
4.2.6	Effect of time lag	24
4.2.7	Effect of applying North Sea mask	25
4.2.8	Effect of data augmentation	26
4.2.9	Effect of architecture	26
4.2.10	Effect of definition loss function	27
4.2.11	Non traceable effects	27
4.3	Discussion	27
5	Conclusions and recommendations	29

5.1	Conclusions	29
5.2	Recommendations	29
	References	31
A	18 year ERA5 dataset	33

1 Introduction

1.1 General

Within the Rijkswaterstaat Operational Systems (RWsOS), SWAN wave models are used to provide wave forecasts for both the Dutch coast and the larger lakes. These forecasts are essential for safe navigation and safety during highwater conditions.

Based on several hindcast studies (Deltares, 2023 and Deltares, 2024a,b) the overall model performance of these wave forecasts is well known. The bias in significant wave height H_{m0} is in general good with just a few percent deviation. However, the scatter in wave height can be rather large, especially for the low frequency wave height H_{E10} . As the wave forecast can deviate significantly from the actual wave conditions, there is a need for confidence intervals to the wave forecasts.

This project - part of the Rijkswaterstaat Corporate Innovation Program (CIP) – is set up to find a suitable way to add confidence intervals to wave forecasts. Although various uncertainty sources affect the forecasts, the present project focusses for now on the wind uncertainty only, by applying ECMWF wind ensemble members.

The project was initiated last year by setting up the fast TurboSWAN surrogate model (Deltares, 2024c). TurboSWAN makes it possible to simulate the waves with the fifty wind ensemble members on the North Sea domain within just a few minutes. The idea is to apply the spread in results as confidence intervals to the operational deterministic SWAN forecast results.

For this year, the focus is on improving TurboSWAN and defining suitable postprocessing to judge its results. A project follow-up will be needed to bring the concept eventually to the end goal: confidence intervals added to the operational wave forecasts of RWsOS.

1.2 Objective

The objective of this project is to improve the fast surrogate wave model TurboSWAN so that it can run fifty wind ensemble members within minutes, providing realistic prototype wind related confidence intervals to the operational SWAN-North Sea wave forecasts.

1.3 Main message

The surrogate model TurboSWAN was trained on large datasets covering years of input and output of the SWAN-North Sea model. Model input consists of wind fields, water depth and wave boundary conditions. The output is fields of significant wave height. TurboSWAN represents large-scale wave height features quite well. However, with typical RMSE values of approximately 0.6 m in significant wave height relative to the SWAN results for eight selected locations, it is not good enough yet for operational use. The initial approach for a complete surrogate model on a large domain did not lead to the desired results. However, based on literature and experience so far it must be possible to further improve the results. We recommend to switch to a stepwise approach, reducing the complexity.

1.4 Set up of the report

After this introducing chapter, Chapter 2 describes the approach and data. Chapter 3 is a literature study. Details about setting up TurboSWAN and the results can be found in Chapter 4. Conclusions and recommendations are given in Chapter 5.

2 Approach and data

2.1 Brief overview of previous findings (Deltares, 2024c)

The SWAN-North Sea model swan-noordzee-j22_6-v1a was used to compose a large dataset to train a surrogate model: TurboSWAN. The dataset includes SWAN runs of fifty ensemble members covering a year of three-day forecasts (2022) and several additional non-sequential months including storms (in 2013 and 2017-2022). These fifty members of ECWMF wind fields come from the TIGGE project and are freely available.

To train TurboSWAN, a convolutional neural network with a U-net based architecture was used, mainly because these types of networks excel in spatial pattern recognition. As input we used spatial maps of wind velocity (both east/west and north/south components), sea surface height, current (both east/west and north/south components), as well as three arrays with boundary conditions (significant wave height, swell wave height and wave period $T_{m-1,0}$). The five output parameters cover the entire domain and are significant wave height, swell wave height, wave period $T_{m-1,0}$, peak wave period and mean wave direction.

TurboSWAN is very fast (25 ms to compute one time step; for eight time steps and fifty ensembles, it would take about ten seconds on GPU) and already able to model the dominant dynamics. However, during storms and some other cases, errors at the validation stations are still too large for an application. Over the entire domain, the root-mean-square-error of TurboSWAN – compared to the SWAN results – is roughly 0.5 m for wave height and swell wave height. The wave period $T_{m-1,0}$ has an error of approximately 1.5 s. All these output variables clearly have higher errors along the coastal areas as well as near the northern boundaries of the North Sea.

2.2 User stories from Hydrometeo Centre (HMC)

Rijkswaterstaats Hydrometeo Centre provides North Sea wave forecasts, typically for – but not limited to - nautical traffic. These forecasts are based on corrected SWAN model results, in combination with observations and experience.

Their wishes regarding the forecasts are defined in the following five user stories. These user stories may help in putting the right focus within TurboSWAN.

- 1) I want to be able to say something about high swell (>1.0 m) occurring during the coming 24 hours, anywhere in the approximately 100 km wide coastal strip along the Dutch coast.
- 2) I want to be able to say something about chances on swell exceeding the local criteria in the entrance channels of the main Dutch harbours (IJgeul, Eurogeul, ...), during the coming 12 hours.
- 3) I want to be able to say something about the chances on critical sea states just north of Ameland during the coming 12 hours.
- 4) I want to be able to say something about the chances on high waves and swell waves during the coming 48 hours, anywhere in the approximately 100 km wide coastal strip along the Dutch coast.

2.3 Approach

The work described in the present report is a follow up of Deltares, 2024c and the focus is on improving TurboSWAN. We start with a literature study, looking for AI implementations for wave forecasts.

Next, we select promising adjustments to improve TurboSWAN. Sensitivity tests will be performed for various variations. We start with the same datasets as previously in Deltares, 2024c but make a different division of the dataset into training and validation data, preventing ensemble members of the same moments in both datasets. Also, we set up another training dataset with a wider spread in conditions, based on more than 40 years of ERA5. Furthermore, we define suitable metrics for a selection of locations to be able to judge the model variations. All training calculations and validation have been carried out at the Snellius super computer at Surf.

2.4 Data

2.4.1 TIGGE dataset

The trainings and validation set consists of 6-hourly input and output of the SWAN-North Sea model (swan-noordzee-j22_6-v1a) for the entire year 2022 plus seven additional months with storm periods, see Table 2.1 (Deltares, 2024c). 30 wind ensemble members of the ECMWF model have been used as input for SWAN runs. These runs cover a three-day forecast with start dates at 00:00 of each day. The wind fields, having a spatial resolution of 0.5° x 0.5°, have been downloaded from the TIGGE website.

Combined, this leads to a dataset of about 175.000 samples, of which about one third is storm data.

Table 2.1: Overview of model input to SWAN-North Sea for production of the initial dataset. The wave boundary conditions and water level and flow fields have been taken from MATROOS.rws.nl

Period	Wave boundary conditions	Boundary type	Water level & Flow	Wind forcing	Storms	Missing data
Year 2022	knmi_ecmwf	2D spectra	DCSM_v6_harmonie	TIGGE	Corrie, Malik Eunice	
Dec 2023	Knmi_ecmwf	2D spectra	DCSM_v6_harmonie	TIGGE	Pia	
Feb 2023	Knmi_ecmwf	2D spectra	DCSM_v6_harmonie	TIGGE	Swell event	
Feb 2020	Knmi_ecmwf	2D spectra	DCSM_v6_hirlam	TIGGE	Ciara	
Jan 2019	Knmi_ecmwf	2D spectra	DCSM_v6_hirlam	TIGGE		
Jan 2018	Knmi_ecmwf	2D spectra	DCSM_v6_hirlam	TIGGE		
Oct 2017	Knmi_ecmwf	2D spectra	DCSM_v6_hirlam	TIGGE		24-31 Oct
Dec 2013	Knmi_ecmwf	2D spectra	DCSM_v6_hirlam	TIGGE		

For the initial TurboSWAN (Deltares, 2024c) this dataset has been split into training data (ensemble member 0-24) and validation data (ensemble 25-29). Later we found that the ensemble members differ only slightly from each other so that the training set doesn't offer sufficient variation for the model to learn generic patterns.

In the present study, we applied a more thoughtful split into training and validation set, namely in time blocks. We split the year 2022 into time blocks of two weeks, with a period of 12 hours, or two time steps, between two consecutive blocks. These blocks were then randomly split between training data and validation data, using a 70%-30% split. For the storms, we randomly picked 4 storms for training and the other 2 storms for validation. We did this for the first 30 ensembles, meaning that the same time blocks still appear 30 times in the data sets, however the similarity between the training set and the validation set is lower.

2.4.2 ERA5 dataset

Deltares has performed a 43 year hindcast, resulting in an hourly time step of various wave parameters covering the SWAN-North Sea domain for the period 1979 – 2022. This contrasts with a 6-hour time step for the TIGGE-trained model. The model domain equals that of swan-noordzee-j22_6-v1a – and hence that of the TIGGE based dataset - but there are some differences in model settings and sources for wave boundary conditions and wind fields. The wind fields as well as the wave boundary conditions in this dataset come from ERA5. We made a selection of 18 full years out of this dataset, using the criterion that such year must contain at least one moment that the swell wave height HE10 at a specific North Sea location (3.45°E, 52.9667°N) is larger than 4 m or – for wave directions that are not in the north western quadrant – larger than 3 m. This leads to a dataset of over 120.000 samples. The selected years are:

1979	<u>1980</u>	1985	1987	1988	1990	1993	<u>1994</u>	<u>1995</u>
1996	1998	1999	2000	2006	2007	<u>2012</u>	2017	2022

The underlined years are available as validation test data (only 1994 presented in this study now), the others as trainings data.

2.4.3 Observations

The initial goal for TurboSWAN is to reproduce SWAN within acceptable bounds. However, where available we benchmarked results against wave buoy measurements (downloaded from MATROOS) to have an additional source for validation. This was only done for the TIGGE-based test data. For the ERA5 dataset, only year 1994 was processed for validation, where observations were not available. This allowed us to run the last experiments, but withheld us to process the other validation years

3 Literature study

3.1 Surrogate Modelling

An active branch of Artificial Intelligence (AI) research is dedicated to surrogate modelling: creating a very fast data-driven surrogate of an existing numerical model. The most common aim of surrogate modelling is the reduction of both time and computational resources needed to achieve a calculation result. Since prediction with AI models (commonly called inference) is very fast this provides a welcome speed-up, especially with respect to the more complex, resource heavy numerical models.

In order to reach a fast-predicting AI model, however, both a computational demanding training process and an appropriately large amount of training data is needed for proper generalization of the AI model (*i.e.* the AI model performs well on unseen data). The amount of necessary training data depends on the complexity of the numerical model (including variations in its input) to be replicated and the complexity of the AI technique used. Since surrogate modelling is often used for very complex numerical models, typically very large amounts of varied training data are required. Since this training data consists of output from the (complex) numerical model to be replicated, the generation of the training data too creates a large computational demand.

3.2 Data-driven prediction of wave fields

In literature, data-driven wave prediction has been an active field for quite a while already. Part of these existing studies focus on one or several locations – often buoy locations – to predict waves inside a harbour (Tsai et al., 2002), swell and low frequency waves (Lopez et al., 2015), spectral wave parameters (Callens et al., 2020) or entire wave spectra (Den Bieman et al., 2023).

When it comes to predicting entire (wave) fields, the spatial dimension needs to be explicitly addressed. To do this seemingly most common approach in literature is based on Convolutional Neural Networks (CNNs). Example applications include a next-frame-prediction of the SWASH wave model (Wei & Davison, 2022), improving wind forecasts in the Mediterranean (Yevnin & Toledo, 2022), or identifying nearshore wave breaking (Sáez et al., 2021). Convolutional Neural Networks have their origin in the field of computer vision, which aims to automate human vision tasks such as classification, segmentation, and object detection. In this field, Deltares has prior experience with using CNNs in computer vision tasks either in our physical modelling facilities (Den Bieman et al., 2020) or in the field (Moreno-Rodenas et al., 2021). Evolving from computer vision tasks to surrogate modelling applications, CNNs are adapted to enable image-to-image prediction (also known as image translation) since in the envisaged surrogate model the output needs to be an image (wave field) as well. Examples of successfully modelling very large domains – such as the North Sea – are rare though, with Yevnin & Toledo (2022) modelling the Mediterranean as one of the few exceptions.

3.3 The temporal dimension

When moving past the image-to-image prediction mentioned in the previous paragraph and adding the temporal dimension, things get a bit more complex. From our physical understanding of wave generation and propagation, we know that what happens *e.g.* off the coast of Norway now can affect the Dutch coast hours later. There are several ways of including this temporal dimension in an AI model. One of the most straightforward ways is simply supplying more than one input field to your deep learning model – say both the current wind

field and multiple wind fields of the preceding hours – so it can learn these temporal patterns in addition to the spatial patterns.

To account for the diminishing importance of ‘older’ temporal input, a Long Short-Term Memory (LSTM; Hochreiter & Schmidhuber, 1997) models introduce an architecture which allows the model to ‘forget’ information that is not (or no longer) conducive to learning the patterns exhibited by the target variable (wave height in our case). Hence, this type of model is more common in time series analysis problems, but the basic premise could also be applied to images.

A recent, alternative model architecture is the Transformer, which has largely overtaken LSTM models in natural language processing tasks. Dosovitskiy et al. (2020) recently adapted this concept to enable its use on images: the Vision Transformer (ViT). For images, the ViT’s attention mechanism allows it to learn patterns between the different patches (smaller parts of an image) within an image, thus focussing on spatial patterns. Adding to the ViT’s ability to learn spatial patterns, Schrama et al. (2026) combines the ViT and LSTM architecture to form a hybrid model that should be able to excel at learning patterns in both space and time. Note that, apart from a successful proof-of-concept, this model architecture is still very much on the cutting (or even bleeding) edge and not yet (robustly) proven technology.

3.4 Conclusion

For TurboSWAN the most logical choice is a CNN as it is a proven technology, suitable for spatial fields. Although LSTM is an interesting and relatively simple option, it is mainly used for time series on a single location, not for entire wave fields. The rather new Vision Transformer might be a good option for future applications but not yet sufficiently robust to use now in the present study.

4 Setting up TurboSWAN

4.1 Introduction TurboSWAN

TurboSWAN is a CNN deep learning model which takes multiple input fields along with a 1D vector of boundary conditions to predict one or more output fields. During training, the model learns to recognize patterns in the input data and how these patterns translate to the output fields.

We built upon the TurboSWAN model architecture (Deltares, 2024c) and made some small adjustments to the network in different experiments. We will now give a short summary of the architecture of TurboSWAN. For a more detailed explanation of the network we refer to (Deltares, 2024c).

The neural network for the TurboSWAN model consists of an encoder part and a decoder part. Note that here the encoder and the decoder refer to the first half and second half of the network respectively. The encoder learns the patterns from the different input parameters and maps them to a latent space. The decoder will then learn to construct the output parameter maps from this latent space. In addition to the encoder-decoder structure, the network also has skip connections between the encoder and the decoder. When a neural network is very deep and has a lot of layers, as is the case with the network for TurboSWAN, information learnt in the earlier layers of the network can get lost due to vanishing gradients. To try to retain some of this information, we included these so-called skip connections between the different layers in the encoder and in the decoder. This type of network structure with skip connections is typically called a U-net, due to its shape, and has proven to be successful in retaining the information of low- and high-level features present in the data (Ronneberger et al. 2015).

To further prevent information loss due to the depth of the network, we also used residual blocks in the network. Residual blocks were introduced in the ResNet model to address the issue of vanishing gradients in deep neural networks (He et al. 2015).

The network, which we present in Figure 4.1, consists of a combination of residual blocks, downsampling blocks, and upsampling blocks. In addition, we use multiple dense layers to combine the image input with the input from the boundary conditions. The downsampling blocks, shown in Figure 4.2, use three residual blocks followed by a single max-pooling layer, to halve the field in both height and width. After each residual block, we have a skip connection to an upsampling block at the same level in the network (Figure 4.1). The upsampling block itself, which we show in Figure 4.3, has a structure similar to that of the downsampling blocks. It again uses three residual blocks, though these residual blocks are made of two deconvolutional layers instead of two normal convolutional layers and then followed by an upsampling layer. The skip connections from the downsampling block are connected and concatenated to the block output and the layer output from the upsampling layer. We used a dropout layer after every residual block in the network to reduce overfitting during training. A dropout layer randomly 'removes' a percentage, in our case 20%, of the neurons during training.

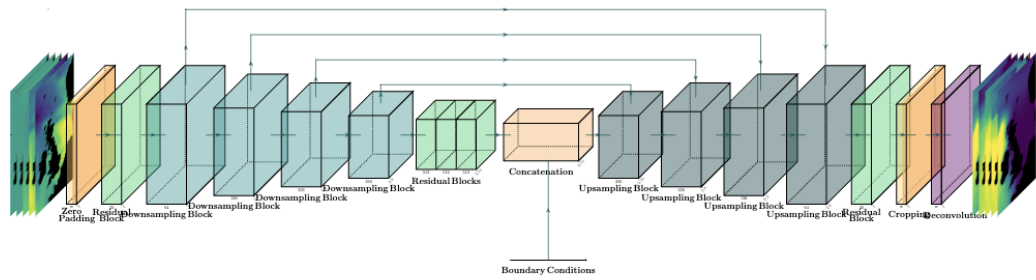


Figure 4.1: The architecture of the TurboSWAN network. The network uses multiple downsampling and upsampling blocks, combined with residual blocks. In the middle of the network the boundary conditions are added to the flattened image input.

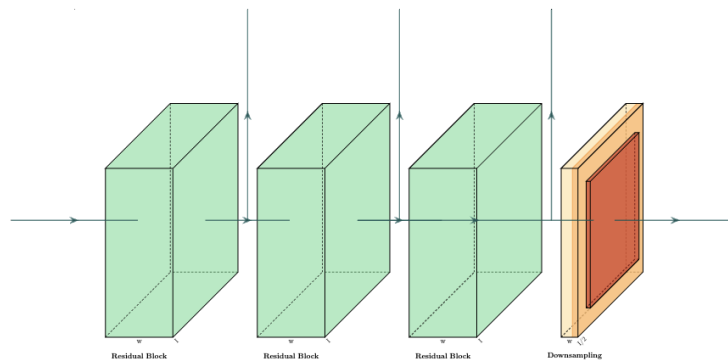


Figure 4.2: The structure of the downsampling blocks used in the TurboSWAN network. The block consists of three residual blocks and has a convolutional layer with stride 2 at the end. After each residual block, there is a skip connection to an upsampling block further down the network.

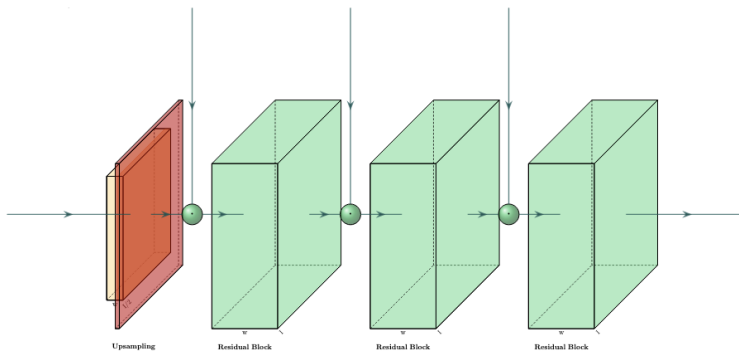


Figure 4.3: The structure of the upsampling blocks used in the TurboSWAN network. The block consists of three residual blocks and has a deconvolutional layer with stride 2 at the start. Before each residual block, the output of a residual block in a downsampling block earlier up the network. Experiments

4.1.1 Summary of experiments

A summary of the conducted experiments and the main results is given below in Table 4.1 and Figure 4.4. The explanation on the experiments and how the results are quantified is given in the following sections.

Table 4.1: Overview of runs. TIGGE* is the dataset used in 2024, TIGGE is the same dataset but with a better split in training and validation. Lag is the time lag for input. Wave B.C. is wave boundary conditions. Lev is level, either just water level ('WL') or the total water depth including water level ('dep'). Augm is augmentation of data. Architecture is explained in the text. The loss calculation is based on scaled or unscaled MSE ('sMSE' or 'uMSE') or relative MAE ('rMAE'). Mask refers to applying a North Sea mask or not. The output is either 5 parameters (Hs, HE10, Tmm10, Tps, Θ) or just Hs.

runid	input										Output
	dataset	Lag [hr]	Wave B.C.	Wind	Lev	Flow	Augm	Architecture	Loss	Mask	
2024	TIGGE*	0, -6	yes	U,V	WL	yes	-	0	sMSE	-	5 pars
Base	TIGGE	0, -6	yes	U,V	WL	yes	-	0	sMSE	-	5 pars
V02	TIGGE	0, -6	yes	U,V	dep	yes	-	1	sMSE	-	5 pars
V03	TIGGE	0, -6	yes	U,V	dep	yes	-	2	sMSE	-	5 pars
V04	TIGGE	0, -6	yes	U,V	dep	yes	-	3	sMSE	-	5 pars
V05	TIGGE	0, -6	yes	U,V	dep	yes	-	4	sMSE	-	5 pars
V06	TIGGE	0	yes	U,V	dep	yes	-	4	sMSE	-	5 pars
V07	TIGGE	0, -12	yes	U,V	dep	yes	-	4	sMSE	-	5 pars
V09	TIGGE	0, -6	yes	U,V	dep	yes	-	3	sMSE	yes	5 pars
V10	TIGGE	0	yes	U,V	dep	yes	-	3	sMSE	-	5 pars
V11	TIGGE	0, -6	-	T _x , T _y	dep	-	-	3	sMSE	-	Hs
V12	TIGGE	0, -6	-	T _x , T _y	dep	-	-	3	sMSE	yes	Hs
V14	TIGGE	0, -6	-	T _x , T _y	dep	-	yes	3	sMSE	yes	Hs
V16	TIGGE	0	-	U,V	dep	-	-	3	sMSE	-	Hs
V18	18yrERA5	0	yes	U,V	dep	-	-	3	sMSE	-	Hs
V19	18yrERA5	0	yes	U,V	dep	-	-	3	rMAE	-	Hs
V20	18yrERA5	0	yes	U,V	dep	-	-	3	uMSE	-	Hs
V22	18yrERA5	0, -3	yes	U,V	dep	-	-	3	uMSE	-	Hs
V23	18yrERA5	0, -6	yes	U,V	dep	-	-	3	uMSE	-	Hs
V24	18yrERA5	0, -3 -6 -9	yes	U,V	dep	-	-	3	uMSE	-	Hs

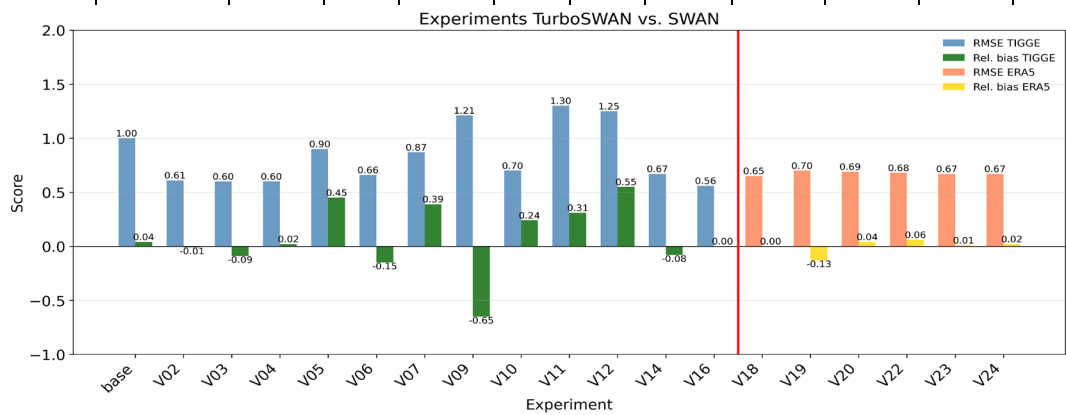


Figure 4.4 RMSE and relative bias (relative to SWAN) based on significant wave height (Hs) at 8 locations of various TurboSWAN experiments. Datasets TIGGE (<V16) and 18yrERA5 (>V16). For Vbase to V16 the validation is based on the independent 3 test blocks described in 4.1.2 and for V18-V24 the validation is based on an independent test year, 1994.

4.1.2 Introduction experiments

Test-train split

Nineteen experiments were conducted for this report with variations in architecture, in the input, time lag, loss function and the output. Note the first version (version 2024 in Table 4.1) did not have a fully independent test-train split, because of correlations between ensemble members for the same window. Version 'base' and v02 up to v16 were done with a new independent train-test split for the TIGGE-driven ensemble model results. We observed that the ensemble spread in the TIGGE-driven forecasts was small. In order to exclude that TurboSWAN was trained on a too small dataset, we continued to train on a 14 year ERA5 dataset with 4 years of test data (v18-v24). For these results we used one year (1994) to validate on the test dataset.

The TIGGE-trained model is tested on four random time blocks in the year 2022. These blocks are the same for experiments v02-v16 and include:

- 2022-02-15 to 2022-03-01
- 2022-04-27 to 2022-05-10
- 2022-10-18 to 2022-11-01

The ERA5-trained model is trained on 14 years and 4 years were kept apart to independently validate. The results shown here are only validated for the year 1994.

Architecture variations

We made several small adjustments to the network for the different experiments. In total we made use of five different architectures, including the architecture used in 2024. The first architecture, architecture 0 is identical to the architecture described in (Deltares, 2024c), which means that dropout was not included yet. Architecture 1 uses the same architecture as architecture 0 but includes the dropout layers as described in Section 4.1, while also only using two residual blocks in each down sampling and up sampling block instead of 3. Architecture 2 again has three residual blocks per down- and up sampling block while still including dropout layers. Additionally, the number of features used in the final down sampling block and the first up sampling block are changed. Where architecture 1 has 256 features in those blocks, architecture 2 uses 512 features instead. Architecture 3 again uses 256 features in the last block of both the encoder and the decoder. Moreover, the network uses a kernel size of five instead of three in the convolutional layers whilst keeping the other components the same as architecture 2. Lastly, architecture 4 uses 512 features with the other components the same as architecture 3.

Table 4.2: architecture variations

Architecture	Dropout	# Residual blocks	# Features in block	Kernel size
0	No	3	256	3
1	yes	2	256	3
2	yes	3	512	3
3	yes	3	256	5
4	yes	3	512	5

Time lag

The wave system is a wind-forced system that typically has correlation scales up to 15 hours within the North Sea domain. This means that the wind and wave fields from earlier time steps can influence the wave field at the current time step. In order to account for this, a time lag input layer can be added to the CNN U-net. Several experiments were run with time lags of (0,) -3, -6, -9 and -12 hours for all input fields (spatial fields of wind, waves and water levels).

Variations in input

In order to find the optimal TurboSWAN, a range of input data was tested. This includes experiments with and without wave boundary conditions. and Wind speed (U_{10}) is applied as zonal and meridional u- and v-components, but also as wind stress x- and -y components. This is motivated by the fact that momentum transfer and stress are more directly related than wind speed, so the relationship is easier to learn for the model. Initially water level was included, without explicitly taking into account bathymetry. This was changed into water depth (as bathymetry and water level combined). Experiments also include runs with and without currents (Flow in table 4.1).

Data Augmentation

Data augmentation is a technique to augment the training dataset, by applying rotations on the dataset to artificially increase the training dataset. This can help in generalization of the network, leading to better results (Wang et al. 2025). Experiment v14 included data augmentation. We chose to use data augmentation on the TIGGE dataset to increase variety in the dataset, which was limited due to the nature of ensembles being similar to each other. We augmented the data by randomly applying flips both horizontally and vertically. Note that directional fields will also need to change sign depending on the flip to preserve the correct direction. However, we did not take this into account when applying our data augmentation.

Loss function

The loss function penalizes the network training based on a certain metric. Here we mainly used the scaled Mean Square Error (sMSE), being scaled with the maximum wave height in the dataset. Other experiments were conducted with the unscaled MSE (uMSE) or with the relative Mean Absolute Error (rMAE).

In addition to variations in the loss function metric, three experiments were done with a mask for the North Sea to add more importance to the North Sea domain compared to the North Atlantic parts. The mask is displayed in Figure 4.5.



Figure 4.5: Mask of North Sea

Output

Initially five output integral wave parameters were chosen for vBase to v10 (wave heights H_s and H_{swell} , wave direction, peak water period T_{ps} and spectral wave period $T_{m-1,0}$). In later experiments H_s was chosen as the only output parameter to be learned, in order to reduce the complexity.

Validation Metrics

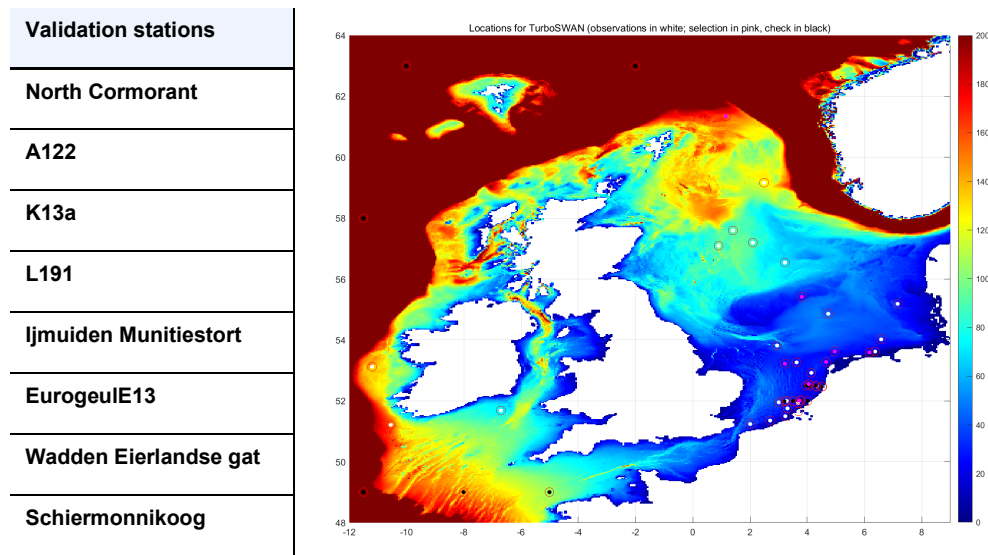
Several metrics were used to validate the TurboSWAN performance against SWAN and observations. The Root Mean Square Error (RMSE), Scatter Index (SI), the relative bias (Rel. Bias). These metrics are evaluated on 8 core validation locations mentioned in Table 4.3. Aggregate statistics were calculated for four ensemble members and all 8 stations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

$$SI = \frac{\sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}}{\bar{y}}$$

$$Relative\ Bias = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)}{\bar{y}}$$

Table 4.3: Eight core locations to assess combined statistical scores (magenta filled markers in the bathymetry [m] plot)



Apart from the 8 core stations, we calculated RMSE across the entire domain to validate spatial patterns. Furthermore, we processed scatter plots, timeseries and snapshots of spatial maps to evaluate TurboSWAN reliability. For the sake of brevity we only display these figures for the best model trained on the TIGGE-dataset (vBase to v16) and trained on the ERA-5 dataset (v17 to v24). Note that the metrics of the two models strictly cannot be compared to each other, since the train-test splits, the structure of the training data, inputs and outputs and time steps are different.

4.2 Results

4.2.1 Loss curves

Figure 4.6 and Figure 4.7 show the training loss and validation loss during training of TurboSWAN V16 and TurboSWAN V18 respectively. We can see that in both figures, the model converges quite fast, especially V16 where it already converges after 10 epochs. When we compare the two figures, we can see a clear difference in overfitting. Model V16 is not able to generalize well onto the validation set, whereas V18 has very similar losses for training and

validation. This implies that the ERA5 dataset provides a better spread of the input and output variables compared to the TIGGE dataset, which is rather uniform due to the ensemble members that are quite alike. The similarity in the TIGGE training data causes the model to overfit. The model doesn't sufficiently recognize the general pattern, but just a pattern that exists in the training data and not in the validation data. Also the loss value itself is lower in V18, but this cannot be directly compared to the loss value of V16, due to differences in the datasets.

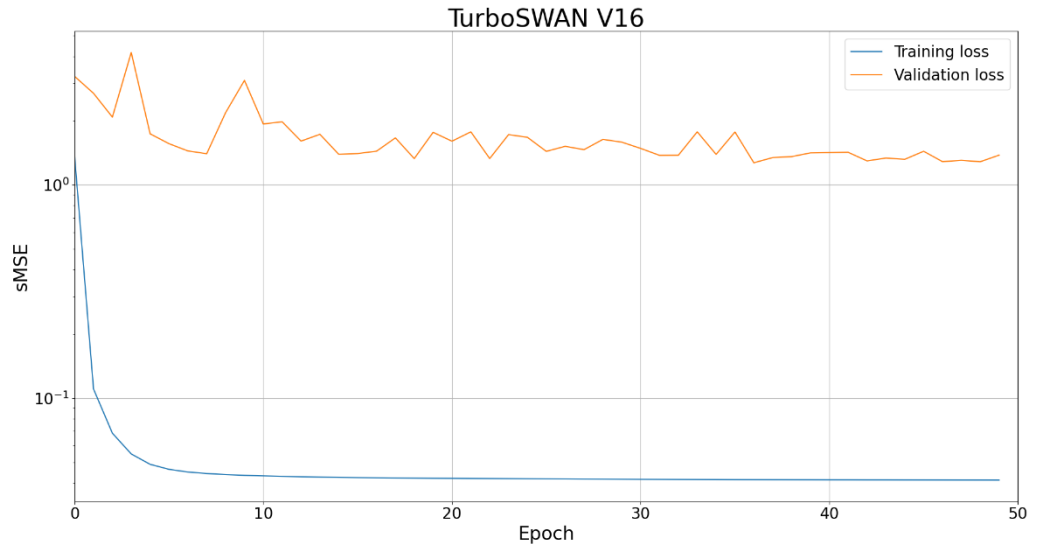


Figure 4.6: The training and validation loss of TurboSWAN V16 (TIGGE dataset) during training (sMSE is scaled mean squared error).

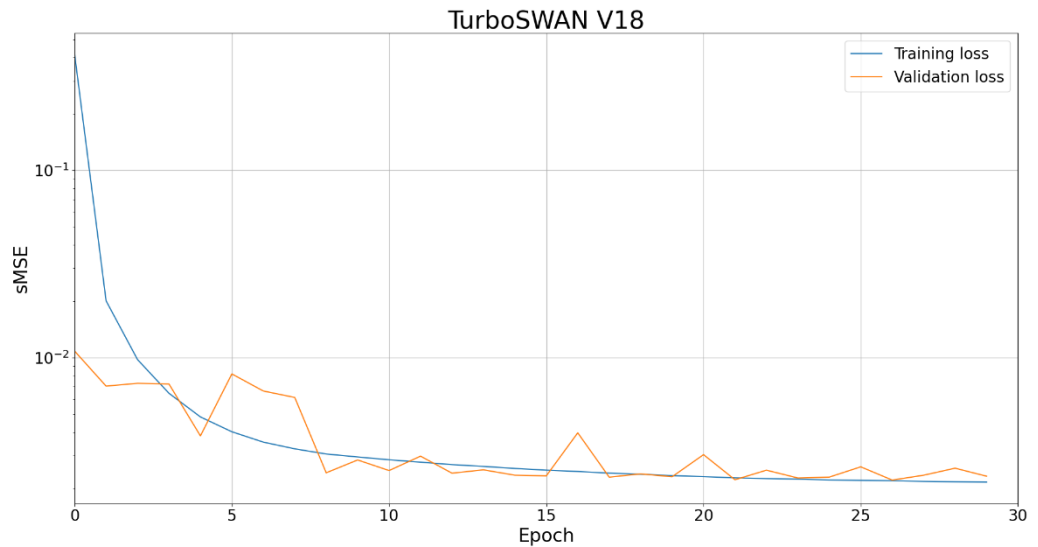


Figure 4.7: The training and validation loss of TurboSWAN V18 (18yrERA5 dataset) during training (sMSE is scaled mean squared error).

4.2.2 Validation results

The metrics RMSE and relative bias from all experiments (see Table 4.1) are displayed in Figure 4.8. The upper panel shows the TurboSWAN scores of significant wave height H_s relative to the SWAN results, the lower panel relative to the observations. Observations were not available for 1994, so V18-V24 are not benchmarked against observations. The blue (RMSE) and green (relative bias) bars refer to the experiments that were trained on the TIGGE

dataset ('base' to V16). The yellow and orange bars represent the TurboSWAN runs trained on the ERA5 dataset (V18-V24).

The best scoring experiment for the TIGGE-trained model is V16 (0.56 m RMSE and 0% bias), while the best model for the ERA5-trained model is V18 (0.65 RMSE m and 0% bias), both based on the comparison to SWAN. Merely to give an impression of the statistical scores of SWAN relative to observations, the RMSE for significant wave height is 0.33 m, the relative bias -10%, based on four stormy periods in 2021-2023 (Deltares, 2023).

Unfortunately, the experiments with the different datasets cannot directly be compared to each other as they cover different periods in time. The TIGGE-trained experiments were tested on 3 random blocks of two weeks in 2022. The ERA5-trained experiments were tested on a full year (1994). Furthermore, the timestep of ERA5 is 1 hour whilst it is 6 hours in the TIGGE set. Since the input of both datasets is slightly different, it is not possible to apply the trained versions V2-V16 to a test set of V18-V24. Just considering the RMSE overcomes this problem to some extent, but the comparison remains strictly unfair. More temporal variability of the outputs may make it harder to train, but more variability of the inputs can also help to find the correct physical relationships. In theory a performance gain is expected with the higher resolution dataset that is better spread, but that mainly depends on the ability of the CNN-Unet to identify the physical relationships.

There is more variation between the ensemble trained TIGGE dataset, because we tested fundamental changes to the architecture and inputs that are reflected in these results. For the ERA5-trained dataset we started with the best architecture and inputs and varied settings less fundamentally.

Figure 4.8 shows that the relative bias is clearly reduced in the ERA5-trained experiments (V18-V24) compared to the TIGGE-trained experiments (V2-V16). The time lag experiments (V20-V24) lead to a small improvement, but not significantly. From theory and in the physical SWAN model, the impact of past wind fields is evident, but this is not shown in the results of our experiments. This means in the current experimental setting that TurboSWAN is not able to learn the impact of past winds on the current wave field yet.

The lower panel in which the TurboSWAN statistics relative to observations are presented should be considered with care. Since TurboSWAN is trained on SWAN results, a perfectly trained model would equal SWAN and not necessarily the observations. If TurboSWAN would come closer to the observations than SWAN, that would merely be coincidence. Given the scarcity of observations and that the research is experimental, the logical approach is to first build a TurboSWAN version that closely approximates SWAN. In a later stage, the training dataset could be augmented with observations. We still considered observations in the validation for reference.

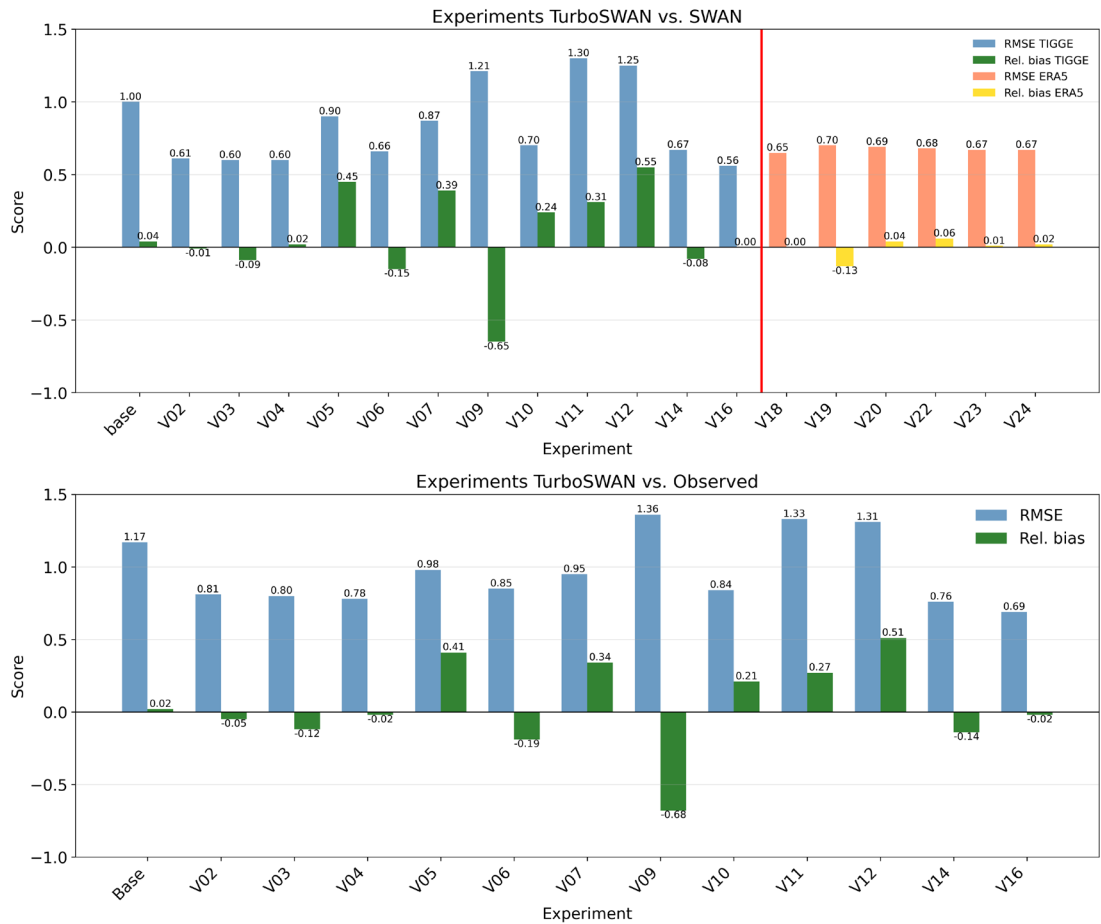


Figure 4.8 Aggregate RMSE and relative bias scores for H_s averaged over the 8 core validation stations for TurboSWAN trained on TIGGE (blue and green) and TurboSWAN trained on ERA5 (orange and yellow). Upper panel TurboSWAN vs SWAN, lower panel TurboSWAN vs observations

4.2.3 Best TIGGE -trained model (V16)

All experiments were evaluated based on the metrics, timeseries, snapshots and RMSE maps. For brevity these are only shown for the best performing experiments for the TIGGE-trained model and the ERA5-trained model

V16 Snapshots

Figure 4.9 shows snapshots of 2022-02-18 00:00 for SWAN (left) and TurboSWAN (right) for significant wave height. The coarse spatial patterns are reasonably captured by TurboSWAN, but the local details and magnitudes are different. The snapshots illustrate the limited skill of TurboSWAN to evaluate timeseries on coastal stations, which is different from capturing large scale features. Note that H_s was the only output for V16, hence wave direction is missing.

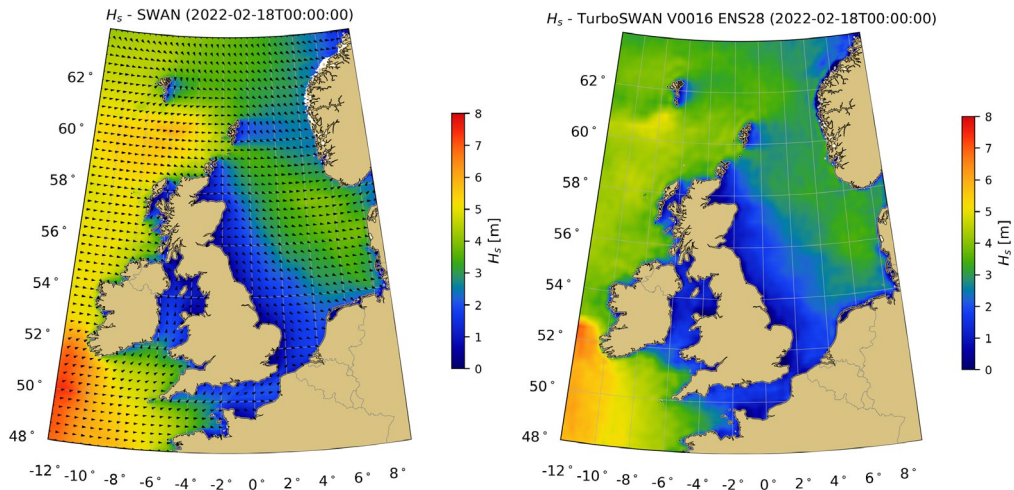


Figure 4.9 SWAN and TurboSWAN (TIGGE-trained) H_s snapshots for 2022-02-18 00:00

V16 RMSE map

The RMSE map in Figure 4.10 displays the RMSE for the entire domain for the test dataset. Some significant differences throughout the domain can be observed. The coastal areas perform best and the Atlantic northwest part performs worse. Although this likely includes the natural variability of larger significant wave heights in the Atlantic part and smaller in the coastal areas, some experiments also show higher RMSE values in the coastal areas (e.g. V10). Based on a comparison between mean H_s and the H_s RMSE, we conclude that the bathymetric features are not only caused by natural variations in wave height. This suggests that during training the model learns the coastal small-scale features more easily than the deep water wave growth. Moreover, errors close to the boundary are still relatively large, indicating that the boundary conditions are not included effectively.

Scatter density

Figure 4.10 displays the scatter density plot for V16 with an RMSE of 0.56 m, a relative bias of 0.0% and a scatter index (SI) of 56%. The underestimation of the extremes becomes apparent, whereas for the range 0.5-3.5 m, the data is closer to the 1:1 line. Matches are aggregate of 4 ensemble members and 8 validation stations.

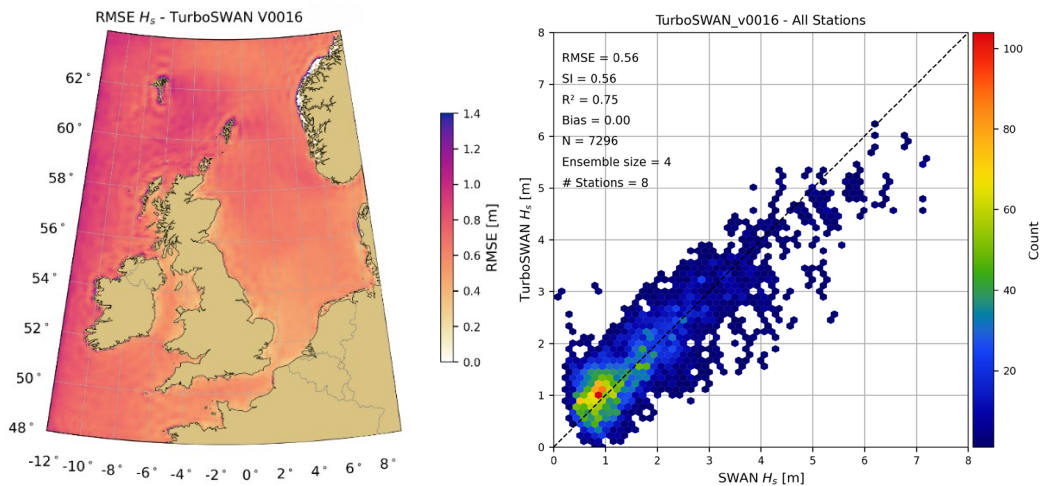


Figure 4.10 left: Spatial RMSE for H_s TurboSWAN v16; right: scatter density plot for TurboSWAN V16 H_s compared to SWAN

4.2.4 V18 ERA5-trained

Map snapshots

Figure 4.11 displays snapshots of 1994-01-22 12:00 for SWAN (left) and TurboSWAN (right). Similarly as for V16, the large-scale features are well represented. However, the magnitude of the significant wave height in the southwest and northeast corners of the domain are too small. Additionally there are differences in the small-scale features of the waves that are entering the North Sea. Even though the metrics are not better than V16, it is encouraging that the large-scale features and smoothness reproduced by V18 is closer to SWAN compared to V16.

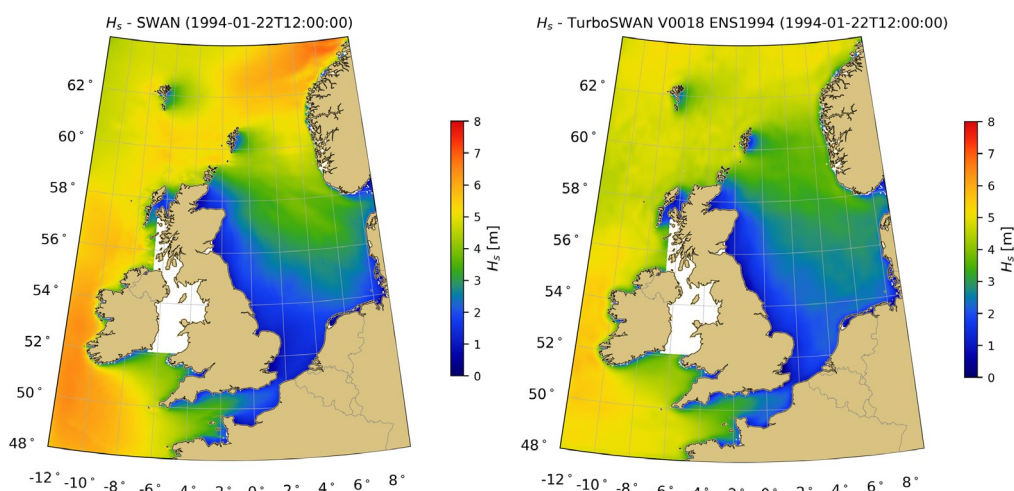


Figure 4.11 SWAN and TurboSWAN (ERA5-trained) H_s snapshots for 1994-01-22 12:00

RMSE Map

Figure 4.12 displays the RMSE for significant wave height for the test year 1994. RMSE scores are larger in the Atlantic ocean parts, similar to V16. However, they are also larger in the centre of the North Sea. The signature of the bathymetric features shows in the spatial RMSE pattern, with smaller RMSE scores for the coastal areas. Similarly to V16 RMSE map in Figure 4.10, this could point to the model having difficulty to learn the wind-wave growth relationship in the central North Sea. This pattern is comparable with V16.

Scatter density

The right panel of Figure 4.12 displays the scatter density plot for the test year 1994. The aggregate RMSE and SI scores for V18 (0.70 RMSE and 69% SI) are higher than V16. Note that for the ensemble dataset test blocks were smaller and 4 ensemble members were used for validation. Moreover the time step for V16 is 6 hours, whereas this is 1 hour for V18. This means the test dataset used for V18 is considerably larger.

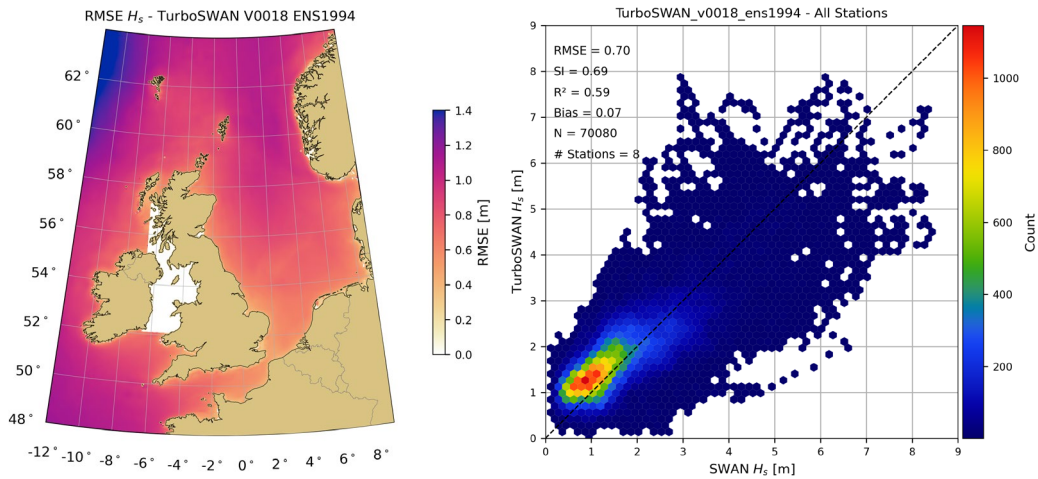


Figure 4.12 Spatial RMSE for H_s TurboSWAN V18 (ERA5-trained) compared to SWAN; b) scatter density plot for TurboSWAN V18 H_s compared to SWAN

4.2.5 Effect of dataset

The ERA5-trained experiments did not lead to RMSE reductions compared to the TIGGE-trained experiments, although biases are consistently low for the ERA5-trained experiments and the spatial patterns are more smooth and realistic (Table 4.1). The absence of further RMSE reductions could be related to the fact that the ERA5 dataset contains a small time step (1 hour instead of 6 hour time step), hence more detail which is harder to learn. The loss curves for V18 show similar curves for training and validation loss. This means that the model generalizes better on the validation data for experiment V18, evidencing the improvement of the training dataset. However, this hypothesis is not confirmed in the aggregate station RMSE scores. Therefore it remains unclear why the RMSE scores are higher than expected. The differences in train-test splits could also play a role, as the V18 is validated on 1 random year, while the V16 validation data are 3 random blocks of two weeks and 4 ensemble members (due to the nature of the ensemble dataset). Thus, the ERA5-trained V18 validation is more robust than the TIGGE-trained V16.

Table 4.4: Experiments with the same set-up where only the training data sets differ (V16 and V18). V19 and V20 only differ in the loss function metric.

	Dataset	loss	Hm0			HE10		
			RMSE [m]	Relative Bias	Scatter Index	RMSE [m]	Relative Bias	Scatter Index
V16	TIGGE	sMSE	0,56	0	0,56	-	-	-
V18	18yrERA5	sMSE	0,65	0	0,64	-	-	-
V19	18yrERA5	rMAE	0,70	-0,13	0,67	-	-	-
V20	18yrERA5	uMSE	0,69	0,04	0,69	-	-	-

4.2.6 Effect of time lag

One would expect that past information up to 15 hours would inform the current wave field for the North Sea. Hence, adding a time lag component (or more) was therefore expected to lead to better performance. Table 4.2 presents experiments with time lag components. For experiments V05 and V07, a time lag of 6 hours and 12 hours lead to larger RMSE and relative bias scores than using no time lag (V6). Experiments with time lags V23 and V24 lead to a small reduction in RMSE and relative bias.

Table 4.5: Experiments where only the time lag components were changed

	dataset	Time lag [hr]	Hm0			HE10		
			RMSE [m]	Relative Bias	Scatter Index	RMSE [m]	Relative Bias	Scatter Index
V05	TIGGE	0, -6	0,90	0,45	0,57	0,89	2,39	0,42
V06	TIGGE	0	0,66	-0,15	0,59	0,40	0,01	0,39
V07	TIGGE	0, -12	0,87	0,39	0,62	0,85	2,06	0,45
V20	18yrERA5	0	0,69	0,04	0,69	-	-	-
V22	18yrERA5	0, -3	0,68	0,06	0,67	-	-	-
V23	18yrERA5	0, -6	0,67	0,01	0,67	-	-	-
V24	18yrERA5	0 -3, -6, -9	0,67	0,02	0,67	-	-	-

It is worth describing the spatial differences in RMSE between experiments V05,V06 and V07 with different time lags. Figure 4.14 clearly shows the bathymetric features in all three experiments. For V6 (no time lag), the Atlantic parts perform worse and the coastal parts perform well, while for V5 (6 hour time lag) and V7 (12 hour time lag) it is the opposite. Although V5 and V7 show different patterns. This shows that for different regions, different time correlation scales are important. However, for the 8 coastal stations that are evaluated, the time lag components are not beneficial.

For V22,V23 and V24 the time lag components are added and lead to small reductions in the error metrics. This is in line with the results of V5-V7, although more time lag components might be needed to further improve the performance.

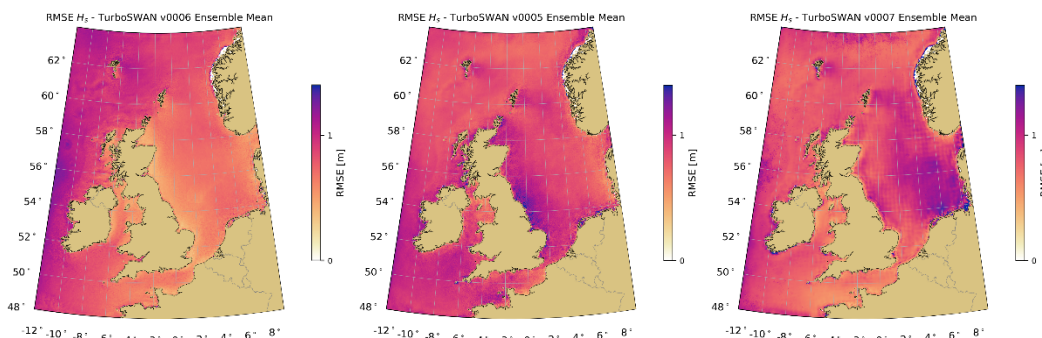


Figure 4.13: RMSE Hs for TurboSWAN V6,V5,V7. V6 has no time lag component, V5 includes a 6 hour time lag in the input and V7 includes a 12 hour time lag in the input

4.2.7 Effect of applying North Sea mask

The effect of applying a North Sea mask in the loss function reduces the RMSE score with 0,05 m, but increases the relative bias from 0,31 to 0,55. Strictly only V11 and V12 exactly have the same settings and can be equally compared, but it is worth comparing the spatial RMSE of V14. V14 also uses data augmentation apart from the North Sea mask in the loss function. Figure 4.14 shows that the addition of the mask leads to a small improvement in the coastal domain, but including data augmentation leads to a significant RMSE reduction in the North Sea domain, as can be observed in Figure 4.15.

Table 4.6: Metrics for experiments related to a North Sea mask in the loss function

	dataset	North Sea Mask	Augmentation	Hm0		
				RMSE [m]	Relative Bias	Scatter Index
V11	TIGGE	-	-	1,30	0,31	0,82
V12	TIGGE	yes	-	1,25	0,55	0,81
V14	TIGGE	yes	yes	0,67	-0,08	0,63

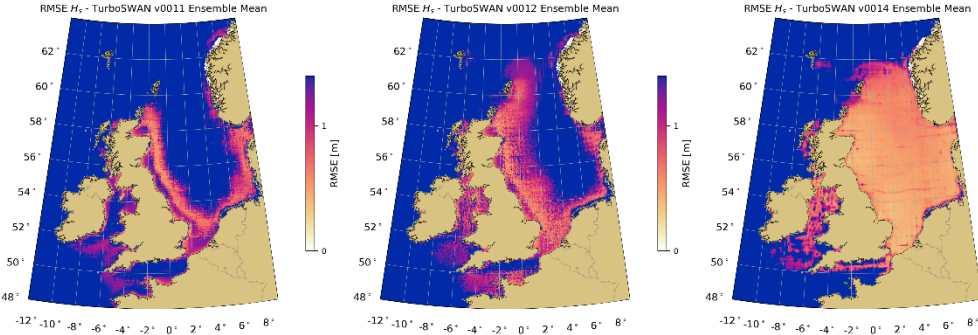


Figure 4.14: Spatial RMSE for V11, V12 and V14.

4.2.8 Effect of data augmentation

As shown in Section 4.2.7 (Effect of applying the North Sea Mask), the data augmentation has a significant effect in the context of the North Sea mask in the loss function. While the data augmentation was only used once, parallel with using the North Sea mask for the loss function, we expect it would also give an improvement on the entire domain. However, there are some risks that may cause negative impact: Data flipping on a sphere grid instead of a rectangle effects the deformation. Also, the change of vector values due to rotation of the augmentation was not properly taken into account. No tests with data augmentation were performed on the ERA5 dataset. Data augmentation is used to increase the variety within the dataset, which can lead to better generalization of the model. Since ERA5 already has more variety than TIGGE and the models generalize well on this dataset, we do not expect a significant increase in performance.

Table 4.7: Metrics for experiments related to data augmentation

	dataset	Data Augmentation	Hm0			HE10		
			RMSE [m]	Relative Bias	Scatter Index	RMSE [m]	Relative Bias	Scatter Index
V12	TIGGE	-	1,25	0,55	0,81	-	-	-
V14	TIGGE	yes	0,67	-0,08	0,63	-	-	-

4.2.9 Effect of architecture

The extra features in the final layer of the encoder and the first layer of the decoder seem to have a fairly big impact on performance for the swell, while only having a small increase in performance for the wave height. This could indicate that it is more difficult to learn the pattern of HE10 for which the extra features and complexity of the network can help.

Table 4.8: Metrics for experiments related to the architecture

	dataset	Architecture	Hm0			HE10		
			RMSE [m]	Relative Bias	Scatter Index	RMSE [m]	Relative Bias	Scatter Index
V06	TIGGE	4	0,66	-0,15	0,59	0,40	0,01	0,39
V10	TIGGE	3	0,70	0,24	0,58	0,65	1,46	0,44

Runs V06 and V10 are the only runs that differ only in architecture.

4.2.10 Effect of definition loss function

Changing the loss function from a scaled MSE (here called: sMSE) to an unscaled MSE (uMSE) to increase the sensitivity for significant wave height does not have the intended effect. The convergence behavior of the model remains the same, while the actual RMSE of the wave height increased slightly compared to the scaled loss. The difference between using sMSE and uMSE is small, so this difference could also be caused by the random nature of the neural network during training. Using the relative MAE increased the RMSE with 0,05 m compared to the sMSE and resulted in a larger relative bias (-13%).

Table 4.9: Metrics for experiments related to the loss function

	dataset	loss	Hm0			HE10		
			RMSE [m]	Relative Bias	Scatter Index	RMSE [m]	Relative Bias	Scatter Index
V18	18yrERA5	sMSE	0,65	0	0,64	-	-	-
V19	18yrERA5	rMAE	0,70	-0,13	0,67	-	-	-
V20	18yrERA5	uMSE	0,69	0,04	0,69	-	-	-

4.2.11 Non traceable effects

The effects of the following aspects – although varied in the experiments - cannot be assessed since they have not been varied exclusively, see Table 4.1.

- Total water depth or water level
- Wind velocity or wind stress
- Wind components or magnitude and direction
- Current
- Wave boundary conditions
- Loss function on one or several wave parameters

4.3 Discussion

The scope of the initial study was to provide uncertainty values with the wave forecasts. Later, the scope changed to improve computational time as well with a surrogate model to run a large ensemble fast. Therefore, the first part of the study includes a version of the surrogate model that was trained on one year of SWAN model forecasts forced with the TIGGE wind ensemble. The TIGGE ensemble is under-dispersed in the first day of the forecast and ensemble members are similar. This challenges the assumption of the dataset being sufficiently diverse. To test this hypothesis, the second version of TurboSWAN was trained on a longer deterministic wave model dataset derived from ERA5 of 14 years and 4 years to validate.

Although we do compare metrics across the experiments, note that an equal comparison is difficult due to the differences in the nature of the training dataset and train-test splits. The ERA5-trained experiments generalize better to the validation data as evidenced by the loss

curve. Bias is generally reduced for the ERA-5 trained experiments. However, the RMSE scores do not follow this trend and the best experiment (V18) still has a higher RMSE than the best TIGGE-trained experiment (V16) for the 8 core stations. The fact that both training and validation loss curves converge, does not necessarily mean that TurboSWAN is skilful. Rather it assures that TurboSWAN is not overfitting.

For the best experiments, large-scale wave features are to a large extent learned by TurboSWAN, but the magnitude and small-scale wave features of the wave field are still incorrect. This poses challenges in accurately modelling timeseries on fixed point locations, evidenced by RMSE scores that are not within a standard deviation of the wave physics SWAN model.

Many experiments with a physically sound basis, did not lead to large improvements in the performance. With these experiments we tested variations in the input, output, the time lag components, variations in architecture and variations of the loss function. This suggests that the CNN-Unet architecture needs to be revisited. The relatively long training time of the CNN-Unet experiments poses a challenge in running a large amount of systematic experiments.

The amount of publications on surrogate wave modelling are scarce, especially in the context of regional shelf seas. Given the convergence of improvements in the current experiments, we advise to decompose the experiments into smaller components and tests and use a step-by-step approach that is modular. For example, this could be combined with experimenting with a CNN for the core stations timeseries. This allows to run many experiments fast and transfer lessons to the spatial CNN.

5 Conclusions and recommendations

5.1 Conclusions

To create a fast method for wave forecasts, a surrogate model TurboSWAN was trained on large datasets covering years of input and output of the SWAN-North Sea model. Model input consists of wind fields, water depth and wave boundary conditions. The output is fields of significant wave height.

TurboSWAN represents large-scale wave height features quite well. However, with typical RMSE values of approximately 0.6 m in significant wave height relative to the SWAN results for eight selected locations, it is not good enough. SWAN has a typical RMSE of 0.3 m for significant wave height relative to observations (Deltares, 2023).

TurboSWAN is very fast: 25ms to compute one time step. For eight 6-hour-time steps covering 48 hours, and fifty ensembles, it would take about 10 seconds on GPU (and approximately a factor 40 more on CPU) to compute the wave field. These time-steps can even be executed in parallel, contrary to SWAN.

Making use of a large training set based on TIGGE data, various experiments were done to improve TurboSWAN but the error statistics did not reduce significantly. The 18 selected years out of the 45 year ERA5 dataset gave more variation in input but this did not result in the desired improvement.

There are various criteria to judge the experiments, i.e. RMSE, bias, for specific locations or for the entire domain and it is hard to define an overall best score. When using different training sets, the statistical scores like RMSE are not necessarily directly comparable.

Many experiments with a physically sound basis, did not lead to large improvements in the performance. With these experiments we tested variations in the input, output, the time lag components, variations in architecture and variations of the loss function. This suggests that the CNN-Unet architecture needs to be revisited. The relatively long training time of the CNN-Unet experiments poses a challenge in running a large amount of systematic experiments.

5.2 Recommendations

The initial approach in which we did various experiments aiming for a complete surrogate model on a large domain with input from wind, water level, currents and boundary conditions did not lead to the desired results. We recommend to switch to a stepwise approach, reducing the complexity.

For the short term we suggest a timeseries approach for a few locations: Set up a timeseries-CNN for a few locations along the coast. Observations could be used for training. The training of such model is very fast so that many tests can be done and experience can be gained, for instance in the dependency of input parameters, definition of loss function, timing of input, selection of output parameters, etc.

For the longer term we recommend the following steps, contributing to the development of a spatial 2D TurboSWAN model

- Make sure that the method is well implemented by extremely simplifying the problem. For instance, a straight domain with uniform deep water and uniform wind (one direction only). Stepwise, more complexity can be added.

- Focus on a small domain, easier to optimise.
- Check whether the trainings dataset is sufficient in size and variation. Is there need for more extremes?
- Check why the convergence in the loss is good whilst the errors in wave height are still large. Should another loss function be applied?
- Explore possibilities for including observed data in the model.

References

- Deltares, 2023. *Actualization and validation of SWAN-North Sea and SWAN-Kuststrook models*. Deltares report 11209278-005-ZKS-0005, final version, 18-08-2023.
- Deltares, 2024a. *Hindcast SWAN-Markermeer*. Ref 11210333-009-ZWS-0003 dd 4 Okt 2024.
- Deltares, 2024b. *Hindcast SWAN-IJsselmeer inclusief IJssel-Vechtdelta*. Ref 11210333-009-ZWS-0003 dd 21 Nov 2024.
- Deltares, 2024c. *Confidence intervals for SWAN wave forecasts*. Deltares report 11210320-018-BGS-0001_v2.0-CIP dd 20 December 2024.
- den Bieman, J.P., de Ridder, M.P. & van Gent, M.R.A., 2020. *Deep learning video analysis as measurement technique in physical models*. Coastal Engineering, Volume 158, 103689.
<https://doi.org/10.1016/j.coastaleng.2020.103689>
- Callens, A., Morichon, D., Abadie, S., Delpy, M., Liquet, B., 2020. Using random forest and gradient boosting trees to improve wave forecast at a specific location. Applied Ocean Research, 104, 102339.
<http://dx.doi.org/10.1016/j.apor.2020.102339>
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houshy, N., 2020. *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv:2010.11929 [cs.CV].
<https://doi.org/10.48550/arXiv.2010.11929>
- Hochreiter, S. & Schmidhuber, J., 1997. *Long Short-Term Memory*. Neural Computation, 9 (8), 1735-1780.
<https://doi.org/10.1162/neco.1997.9.8.1735>
- Lopez, M., Lopez, I., Iglesias, G., 2015. *Hindcasting long waves in a port: An ANN approach*. Coastal Engineering Journal, 57 (04), 1550019. <https://doi.org/10.1142/S0578563415500199>
- Moreno-Rodenas, A.M., Duinmeijer, A. & Clemens, F.H.L.R., 2021. *Deep-learning based monitoring of FOG layer dynamics in wastewater pumping stations*. Water Research, Volume 202, 117482,
<https://doi.org/10.1016/j.watres.2021.117482>
- Sáez, F.J., Catalán, P.A. & Valle, C., 2021. *Wave-by-wave nearshore wave breaking identification using U-Net*. Coastal Engineering, Volume 170, 104021. <https://doi.org/10.1016/j.coastaleng.2021.104021>
- Schrama, W.P., van Bergeijk, V.M., Mares-Nasarre, P., den Bieman, J.P., van Gent, M.R.A. & Aguilar-Lopez, J.P., 2026. *Surrogate modelling of wave overtopping hydrodynamics using an adapted deep learning Vision Transformer*. Coastal Engineering, Volume 203, 104874.
<https://doi.org/10.1016/j.coastaleng.2025.104874>
- Tsai, C.-P., Lin, C., Shen, J.-N., 2002. Neural network for wave forecasting among multi-stations. Ocean Engineering, 29 (13), 1683–1695. [http://dx.doi.org/10.1016/S0029-8018\(01\)00112-3](http://dx.doi.org/10.1016/S0029-8018(01)00112-3)
- Wei, Z. & Davison, A., 2022. *A convolutional neural network based model to predict nearshore waves and hydrodynamics*. Coastal Engineering, Volume 171, 104044.
<https://doi.org/10.1016/j.coastaleng.2021.104044>

- Yevnin, Y. & Toledo, Y., 2022. *A Deep Learning Model for Improved Wind and Consequent Wave Forecasts*. Journal of Physical Oceanography, Volume 52, Issue 10. <https://doi.org/10.1175/JPO-D-21-0280.1>
- Wang, Z., Wang, P., Liu, K., Wang, P., Fu, Y., Lu, C.-T., Aggarwal, C. C., Pei, J., & Zhou, Y. (2025). *A Comprehensive Survey on Data Augmentation* (No. arXiv:2405.09591). arXiv. <https://doi.org/10.48550/arXiv.2405.09591>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (No. arXiv:1512.03385). arXiv. <https://doi.org/10.48550/arXiv.1512.03385>

A 18 year ERA5 dataset

We have 43 years of hourly maps on the entire SWAN-DCSM domain, including wind fields, waterlevel fields and wave boundary conditions.

We made a selection of 18 full years out of this dataset, using the criterion that such year must contain at least one moment that the swell wave height HE10 at a specific North Sea location (3.45°E, 52.9667°N) is larger than 4 m or – for wave directions that are not in the north western quadrant – larger than 3 m.

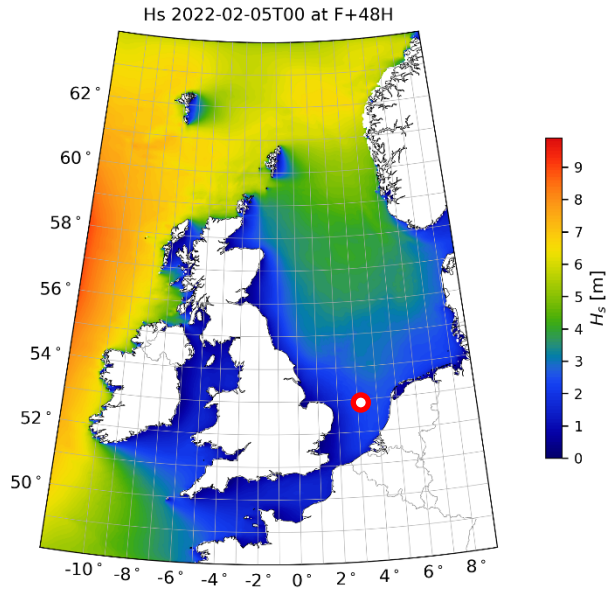


Figure-Appendix A.1: Considered location

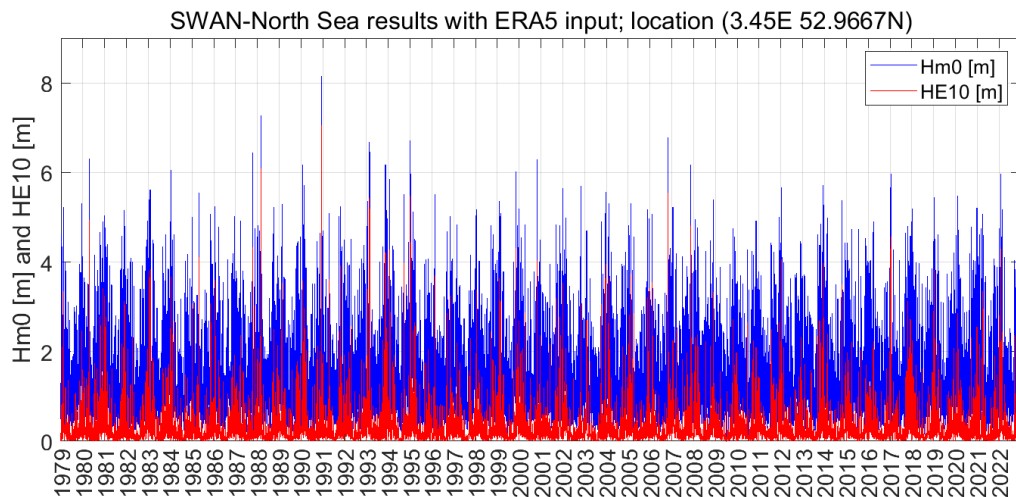


Figure-Appendix A.2: SWAN North-Sea timeseries Hm0 and HE10, 1979-2022

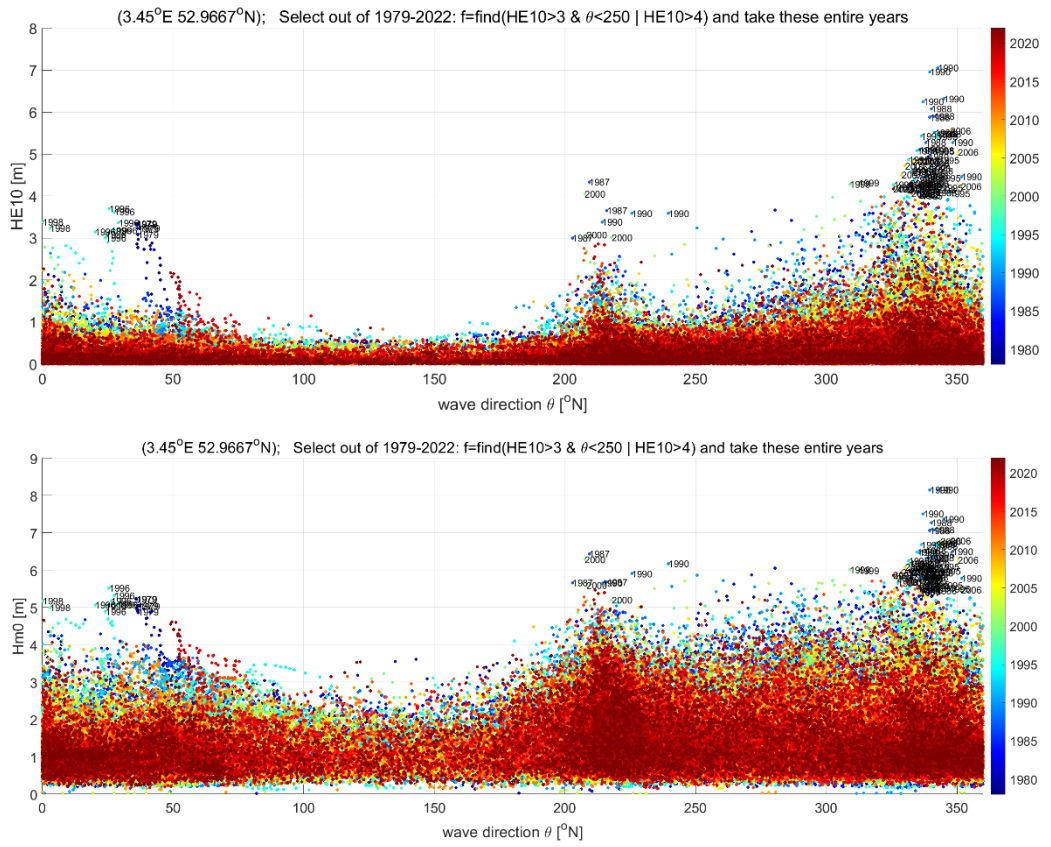


Figure-Appendix A.3: Selection of relevant years; HE10 (upper) and Hm0 (lower) vs direction

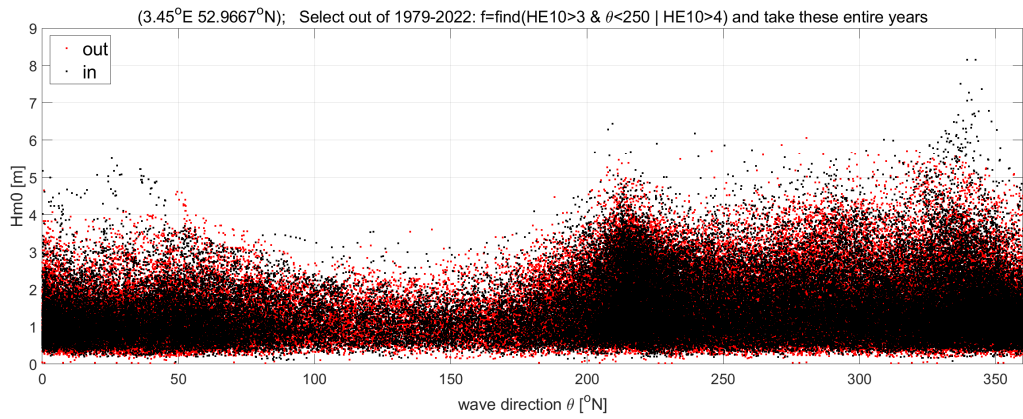
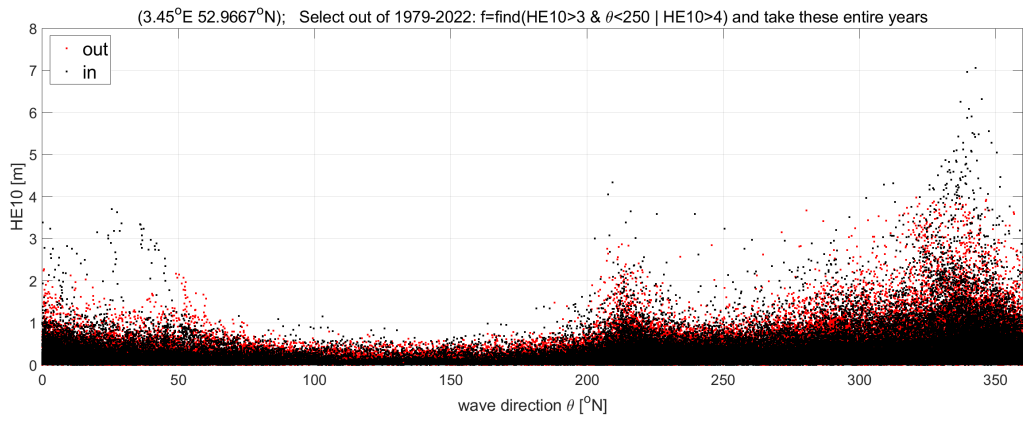


Figure-Appendix A.4: Selection of relevant years; HE10 (upper) and Hm0 (lower) vs direction

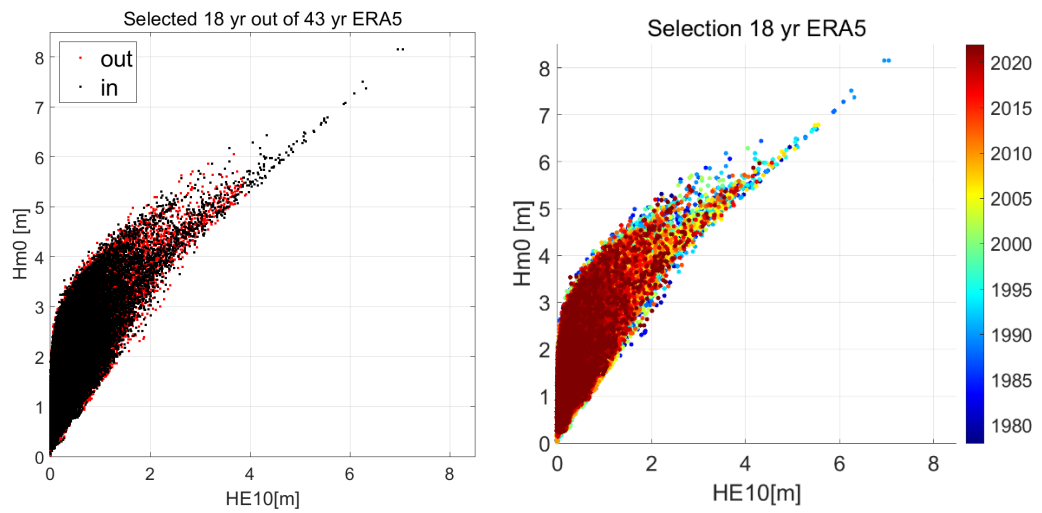


Figure-Appendix A.5: Selection of relevant years

Deltares is an independent institute for applied research in the field of water and subsurface. Throughout the world, we work on smart solutions for people, environment and society.

Deltares

www.deltares.nl