Super-resolution on unstructured coastal wave computations with graph neural networks and polynomial regressions

Jannik Kuehn, Stéphane Abadie, Matthias Delpey, Volker Roeber

To cite this version:

Jannik Kuehn, Stéphane Abadie, Matthias Delpey, Volker Roeber. Super-resolution on unstructured coastal wave computations with graph neural networks and polynomial regressions. Coastal Engineering, 2024, 194, pp.104619. 10.1016/j.coastaleng.2024.104619 . hal-04704696

HAL Id: hal-04704696
https://hal.science/hal-04704696v1

Submitted on 25 Sep 2024

Highlights

Super-resolution on unstructured coastal wave computations with graph neural networks and polynomial regressions

Jannik Kuehn,Stéphane Abadie,Matthias Delpey,Volker Roeber

- Super-resolution can speed up spectral wave forecasts by up to 80 times
- Prediction errors are negligible for most applications
- Polynomial regressions outperform graph neural networks in most cases

Super-resolution on unstructured coastal wave computations with graph neural networks and polynomial regressions

Jannik Kuehn[a,*], Stéphane Abadie[a], Matthias Delpey[b] and Volker Roeber[a,c]

[a] *Université de Pau et des Pays de l'Adour, E2S-UPPA, SIAME, Anglet, 64600, France*
[b] *Center Rivages Pro Tech, SUEZ Eau France, Bidart 64210, France*
[c] *University of Hawai'i at Manoa, Department of Oceanography, Honolulu, 96822, HI, USA*

ARTICLE INFO

Keywords:
Super-Resolution
Coastal Waves
Neural Networks
SWAN Model
Deep Learning

ABSTRACT

Accurate high-resolution wave forecasts are essential for coastal communities, but local and even coastal coverage is often still missing due to the heavy computational load of modern state-of-the-art wave models. This study presents a machine learning super-resolution approach that drastically reduces the computational effort, while keeping errors negligible for the majority of forecasting applications. The method consists of first computing a wave forecast on a coarse mesh which is then converted to a forecast of finer resolution with the help of machine learning. To demonstrate the feasibility and the potential for practical applications of this approach, we present a case study of a 44-year hindcast along the French Basque coast over an unstructured mesh. We introduce two machine learning approaches, a graph neural network and a polynomial ridge regression and compare their performances in different sea states and spatial environments. Both models exhibit very small prediction errors for the significant wave heights, with Root Mean Square Errors (RMSEs) ranging from 0.3 cm to 2 cm, depending on the study region, while being up to 80 times faster than a direct computation of a numerical wave model at the corresponding spatial resolution. To the best of our knowledge, this is the first time that a super-resolution approach is extended to unstructured meshes in the field of coastal sciences.

1. Introduction

From ship navigation safety to coastal risk management, accurate ocean wave forecasts play a vital role for coastal communities (Gopinath and Dwarakish, 2015). Thanks to a vast number of buoy networks and state-of-the-art numerical wave models such as SWAN — Simulating WAves Nearshore (Booij et al., 1999) and WAVEWATCH III (Tolman, 2009), forecast coverage spans the entire globe. Nevertheless, local forecasts are often still missing (Camus et al., 2011). Indeed, particularly in shallow water, an accurate wave description becomes more complex as a consequence of depth-limited processes such as shoaling, wave refraction, and depth-induced breaking. These wave-bottom interactions have a significant effect on the space-time variability of the wave field, especially in highly heterogeneous, shallow bathymetries that can introduce distinctive small-scale features (Gorrell et al., 2011; Ardhuin et al., 2012). Owing to this complexity, high-resolution forecasts of larger parts of the coast demand a lot of resources and often necessitate access to high-performance computing facilities (James et al., 2018).

With the recent rise of machine learning and artificial intelligence, data-driven approaches have become more popular and their adoption by the coastal engineering community has increased considerably. Typical applications include short-term wave forecasting (Londhe and Panchang, 2006; Zhang and Dai, 2019), improving predictions of numerical models (Londhe et al., 2016; Callens et al., 2020; Lucero et al., 2023), or efficiently computing surf-zone hydrodynamics (Ricondo et al., 2024). Despite various successful applications of numerical models over decades, deep learning approaches are becoming a popular alternative since coastal wave models are usually resource-intensive. Surrogate models bypass numerical wave models completely and try to learn patterns directly from a training dataset. Generally, the input is similar to what is used for coastal wave models and might include, for example, wind (Huang et al., 2022; Michel et al., 2022), wave buoy (Chen et al., 2021b), and bathymetry (Jörges et al., 2023) data. These surrogate models are extremely fast and once trained they can compute the forecast more than a thousand times faster than a direct calculation. However, this speed-up comes at the price of some reduction in accuracy. The super-resolution approach proposed in this study offers a reasonable balance between the accuracy of direct numerical computations and the speed of surrogate models.

The concept is to convert results from a numerical model run over a coarse mesh to a higher resolution with the help of a machine learning model. Due to the low-resolution computation of the numerical wave model that is used as an input, the data-driven prediction is based, at least to some extent, on a physical basis. The model only has to adjust the prediction to the correct value, whereas a surrogate

---

[*] Corresponding author

✉ `jannik.kuhn@univ-pau.fr` (J. Kuehn); `stephane.abadie@univ-pau.fr` (S. Abadie); `matthias.delpey@suez.com` (M. Delpey); `volker.roeber@univ-pau.fr` (V. Roeber)

ORCID(s): `0000-0001-7678-5455` (J. Kuehn); `0000-0002-9852-3517` (S. Abadie); `0000-0001-7596-8524` (M. Delpey); `0000-0002-3768-9863` (V. Roeber)

model will need to discover the physical relationships by itself. This generally results in a substantially lower overall accuracy of the latter (Obiols-Sales et al., 2021; Kuehn et al., 2023). On the other hand, a super-resolution approach still speeds-up the calculations considerably since the numerical solvers only have to run on a coarse resolution and the conversion to a higher resolution with a statistical model is very fast. However, both surrogate and super-resolution model methods need an initial high-resolution computation to be trained on — a non-negligible point that we will revisit in the discussion.

There are many examples of successful applications of deep-learning super-resolution in fluid mechanics on structured (Dong et al., 2016; Fukami et al., 2019; Gao et al., 2021) and unstructured meshes (Belbute-Peres et al., 2020-07-13/2020-07-18; Xu et al., 2023). In the ocean sciences, super-resolution based on deep neural networks was used to increase the resolution of bathymetry data (Sonogashira et al., 2020; Yutani et al., 2022), sea surface temperature (Ducournau and Fablet, 2016; Su et al., 2021; Lloyd et al., 2022), and sea surface height (Lopez-Radcenco et al., 2017). Furthermore, Zhu et al. (2023) and Kuehn et al. (2023) applied super-resolution to decrease the computation time of wave height forecasts of numerical models on a structured grid. Although surrogate models for wave forecasts exist on unstructured meshes (Shi et al., 2022), this is not the case for super-resolution.

The goal of this study is to first, extend the super-resolution approach for coastal wave forecasts to unstructured grids, second, underline the value of choosing an appropriate model, and lastly, explore the performances of different models in different wave regimes and bathymetries. To this end, we present and compare two approaches: Graph Neural Networks (GNNs) — a relatively natural choice for unstructured data, often seen as a generalization of the successful Convolutional Neural Networks (CNNs) (Bronstein et al., 2021); and Polynomial ridge Regressions (PRs) — an extension of linear regressions that are used in certain non-linear machine learning tasks, at times even outperforming neural networks (Choon et al., 2008; Cheng et al., 2019).

The article is structured as follows: Section 2 describes the dataset used in our analyses and introduce three distinct study regions with varying wave characteristics. Section 3 is dedicated to the description and explanation of the super-resolution approach and details both data-driven models, the GNN and PR. Following this, section 4 presents our results of super-resolution on unstructured meshes and compares both models with each other. Lastly, we discuss our results and future research directions in section 5 and finish on some concluding remarks in section 6.

2. Dataset

As for any machine learning task, the underlying dataset is of great importance. We chose the 44-year coastal wave hindcast of Lastiri et al. (2020) as a study dataset, as it allows to extend the super-resolution approach to unstructured grids, while at the same time to examine the performance of the GNN and PR in different wave regimes and bathymetries.

The hindcast was created to assess the local wave energy resource at the southwestern coast of France. Its study area includes multiple regions with distinct characteristics, ranging from rocky beaches with strongly varying bathymetric features in the south, over a dominating submarine canyon around the Capbreton area, to sandy beaches with gentle slopes in the north (see Figure 1). Sheltered by mainland France in the east and the Iberian Peninsula in the south, the study area is mostly exposed to energetic swells from the north-western Atlantic. Over the study period, the significant wave height can reach 9 m, with a mean on the order of 1 m. Similarly, the peak wave period exhibits a range of 4 s to 20 s, with a mean of 10.5 s. The peak wave direction ranges from 280° to 310°, but especially in the southern part of the domain nearshore refraction processes greatly influence the shape of the local wave field. Considerable seasonal variations of the wave field can also be observed with differences of almost 1 m for the mean significant wave height between winter and summer months and 2 s for the mean peak period.

The hindcast was computed with the state-of-the-art third-generation wave propagation model SWAN, developed for studying coastal and nearshore domains. The offshore boundary condition is composed of spatially distributed spectra of the BOBWA (Bay of Biscay Wave Atlas) dataset (Charles et al., 2012) that covers the same 44 years (1958 to 2001) as the chosen hindcast. The computation is done on a triangle-based unstructured mesh of 45 156 nodes, where grid steps range from 100 m around the canyon and at the coast to 2000 m further offshore. Local wind generation, tidal oscillations, and triad interaction effects were neglected in this computation. The water level is taken to be constant and spatially uniform, its value being set to the mean-tide level of 2.25 m. Even though the resolution is comparatively coarse near the coastline, depth-induced wave breaking is considered. It is modeled with the bore-based model of Battjes and Stive (1985) with constant values of $\alpha = 1$ and $\gamma = 0.73$. The spectral space was discretized with 20 frequencies ranging from 0.0373 Hz to 0.25 Hz with a logarithmic step and a 3° directional step. Various bulk parameters of the sea state were saved every 3 h over the entire computational domain, including significant wave height H_s, mean absolute wave period T_{m01}, and mean wave direction θ_m as well as the relative peak period T_p and peak wave direction Θ_p. The computational step was set to 15 min and the maximum number of iterations per SWAN sweep was set to 50. For more detailed information about the setup of the hindcast, we refer to the authors' original article. Additionally, the GitHub repository of the Data Availability section 7.1 contains example steering SWAN files.

2.1. Creation of the low-resolution dataset

To create a corresponding low-resolution dataset, we use the exact same setup for SWAN as in the high-resolution hindcast, except for the computation grid, which is much coarser in the spatial domain. To coarsen the original grid, we follow closely the geographical mesh creation process of the original article, but increase the edge size of the cells by a factor of 8. The mesh is generated with a constrained Delaunay triangulation with minimum angle sizes of 25° to ensure that it conforms to SWAN prerequisites. The number of nodes of the low-resolution mesh is 1059, which results in an overall downsampling factor of 42. Figure 9 in the appendix shows the low- and high-resolution mesh employed in this study. In contrast to recent super-resolution approaches (Fukami et al., 2019; Chen et al., 2023; Zhu et al., 2023) that downsample the high-resolution dataset to obtain a low-resolution version, our study creates a low-resolution dataset by running a distinct low-resolution numerical SWAN computation. This makes it more applicable since a machine learning model trained on downsampled high-resolution computations as an input is not necessarily going to perform well in practice with actual low-resolution inputs. In summary, using the coarsened computational grid, the entire hindcast computation was run again to provide a low-resolution dataset. Thus two datasets are available for our study, one at high and the other at low spatial resolution. To keep the amount of data to a practical size, we use for both datasets a subset of 5 years of the 44-year hindcast for training, ranging from 1959 to 1963, and use the entire year of 1964 for testing purposes. The training set itself is split up further, with 20 % that are set aside for a validation dataset and 80 % that are reserved for the actual training of the models. The validation dataset is used for early stopping and tuning of hyperparameters. Note that once the model is trained on the subset of 5 years, it could be used to reconstitute the entire hindcast of 44 years.

2.2. Study regions

While it is possible to train a machine learning model on the entire hindcast domain, it is computation-wise significantly more onerous. Moreover, in many practical applications the areas of interest are usually delimited to a few specific regions nearshore. That is why for our analyses, we divide the hindcast dataset in three geographical subregions with different characteristic wave properties and bathymetries. Figure 1 shows the locations of the regions with respect to the entire study area. An inlet indicates the location of the study area with respect to mainland France. The subfigures on the right-hand side provide a more detailed insight of the bathymetric variations in the different regions. Region 1 is located south of the submarine canyon and covers the beaches around Biarritz city. The region is characterized by rocky beaches with a complex local bathymetry that changes rapidly cross- and long-shore. This results in a high spatiotemporal variability of the local wave field in the area, mostly controlled by wave refraction over shallow bathymetric gradients (Varing et al., 2021; Delpey et al., 2021). Further up north, Region 2 encompasses the coastline around Capbreton town, at the outlet of the submarine canyon that has a dominant impact on the wave field in the area (Abadie et al., 2006). Lastly, Region 3 is situated further north in the *Landes* region, where the sandy coastline exhibits a much more regular bathymetry. Consequently the wave field is much less variable in space in this subregion. Table 1 indicates the number of nodes for each regional low-resolution (n_{LR}) and high-resolution (n_{HR}) mesh.

Study Area	n_{LR}	n_{HR}
Full	1059	45156
Region1	36	2448
Region2	69	3773
Region3	96	2730

Table 1
Number of nodes in the low-resolution (n_{LR}) and high-resolution (n_{HR}) meshes for each region. Full denotes the entire study area.

3. Methods

Originally, super-resolution is a computer vision problem tackling the question whether it is possible to reconstruct high-resolution (HR) details from a low-resolution (LR) image, and if so, to which degree (Capel and Zisserman, 2003). Certain forms of super-resolution are widely in use, such as bilinear or bicubic interpolation (Keys, 1981). A generally more performant alternative is the use of neural networks, particularly fully convolutional (Dong et al., 2016; Fukami et al., 2021) or generative adversarial neural networks (Xie et al., 2018; Stengel et al., 2020). In Kuehn et al. (2023) we show that super-resolution is able to improve the resolution of low-resolution numerical wave computation with mean errors of under 2 cm. However, this and the majority of other studies assume an image-like input; that is, a regular, rectangular grid in the case of numerical wave models. This is a major constraint since a large part of wave fore- and hindcasts nowadays work with unstructured grids (Piggott et al., 2008). One attempt to revert back to the successful CNN-based super-resolution is to interpolate the input to a structured grid, compute the prediction with the CNN and interpolate the prediction back to the unstructured grid, but this leads to inherent interpolation errors or heavy computational loads due to small cell sizes (Pfaff et al., 2021).

For these reasons, we propose to use altogether different neural network architectures. While a simple fully-connected neural network is technically suitable for super-resolution on unstructured grids, we decided, after an initial performance analysis, to concentrate on graph neural networks.

3.1. Graph neural networks and MeshGraphNets

Graph neural networks have seen a large spike of interest over the last few years due to their state-of-the-art

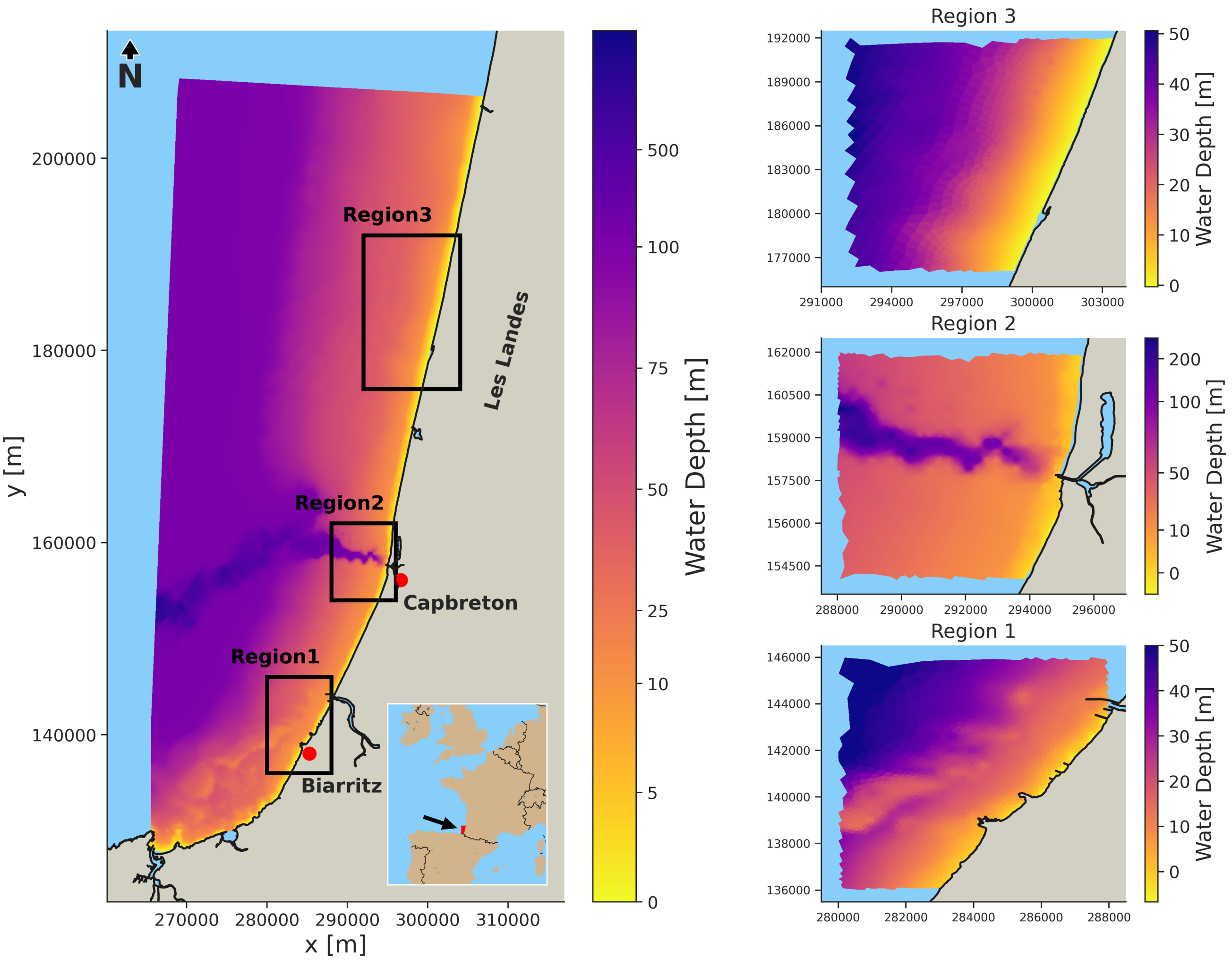

Figure 1: Bathymetry map of the study area and the three regions chosen for an in-depth analysis. The coordinate system is Lambert Zone III, EPSG: 27573. Note the nonlinear colormaps for the study area and Region 2 that were chosen to show the spatial variability of the bathymetry also off the canyon.

performance on a large variety of tasks ranging from traffic forecasts (Jiang and Luo, 2022) to laminar flow prediction around two-dimensional shapes (Chen et al., 2021a). Their arguably biggest advantage is the capacity of working with non-euclidean data, allowing them to generalize over a vast range of datasets (Zhou et al., 2020). In fluid mechanics and lately also in ocean sciences, GNNs are adapted for surrogate, super-resolution, and forecast models with good success. One of the most successful frameworks in working with mesh-based data of physical numerical equations is the Encoder-Processor-Decoder architecture with GraphNet blocks. Battaglia et al. (2018) and Sanchez-Gonzalez et al. (2018) were among the first to introduce the concept of a GraphNet block which was later extended to meshes (Pfaff et al., 2021), and further improved in Fortunato et al. (2022). In Lam et al. (2023) the authors use the same framework in a landmark study, where their GraphCast neural network produces more accurate forecasts than the European Centre for Medium-Range Weather Forecasts at a fraction of the computation time.

Due to its great success, we adapt this framework to our super-resolution approach. In the following, we will use a similar notation as in Pfaff et al. (2021) and Fortunato et al. (2022) for easier comparison.

An undirected graph $G = (V, E)$ is defined by a set of nodes or vertices V together with a set of edges E connecting them. We convert a mesh by setting mesh nodes to graph nodes and using the sides of mesh cells as edges. While using cell edges as graph edges seems intuitive, it is not the only and possibly not the most optimal way to obtain graph edges. Other approaches range from learning edges with the help of static node features like bathymetry, coordinates, or node type (Cachay et al., 2021), to manually creating several short- and long-range interaction edges (Lino et al., 2022; Lam et al., 2023). In a graph each node can have a feature associated with it, denoted by $\mathbf{v}_i \in \mathbb{R}^{f_n}, \forall i \in V$, where f_n is the number of node features. Similarly, edges can have edge features $\mathbf{e}_{i,j} \in \mathbb{R}^{f_e}, \forall (i, j) \in E$, where f_e is the number of edge features. Node features can represent physical parameters like significant wave height, mean wave period or direction at the location of the node, but also static node features such as depth, the coordinates of the node, or the type of node (ocean boundary, coastal boundary, interior, etc.). Common edge features are the distance between two nodes or their relative position.

Our adapted framework of the MultiScale MeshGraphNets (Fortunato et al., 2022) equally consists of three distinct parts: an encoder, a processor, and a decoder. As in the original article, we work with two distinct graphs, a low-resolution $G_l = (V^l, E^l)$ and a high-resolution graph $G_h = (V^h, E^h)$, that are obtained from the corresponding low- and high-resolution meshes. In the original paper, the authors use the low-resolution graph only to pass information more efficiently and further in the graph, allowing them to accurately predict long-range interactions. In a super-resolution approach the low-resolution graph constitutes an intrinsic part of the method. Also, in contrast to the original paper, dynamic node features (like significant wave height) are defined on G_l, thus there is no need for downsampling graphs.

In our study, the node features of the low-resolution graph are the significant wave height H_s and the mean wave direction θ_m at the location of the node. To avoid numerical inconsistencies at 360°, the direction is transformed to a 2D unit vector. A sensitivity analysis of the input features is given in the appendix A.1. The node feature of the high-resolution graph is the significant wave height since it is the variable that we want to predict. As in Kuehn et al. (2023) it is possible to predict other variables, but this study only focuses on H_s. The edge features are the relative coordinates $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ and their norm $||\mathbf{u}_{ij}||$ for each edge $\mathbf{e}_{ij} \in E^l, E^h$. All node features, both low- and high-resolution, are z-normalized over the training data, the edge features are normalized by the largest edge distance.

Encoder

The low-resolution graph G^l and the high-resolution edge features E^h are encoded the same way as in Pfaff et al. (2021) with a multi-layer perceptron (MLP) of 2 hidden layers with a latent size of 128 and a Sigmoid Linear Unit (SiLU) activation. In contrast to that study, we do not use LayerNorm since it showed a decrease in performance in our case.

Processor

The core part of the framework is the processor, which consists of N_l low-resolution blocks, an upsampling block, and N_h high-resolution blocks. In both the low- and high-resolution blocks, the nodes on the graph are updated by message passing — each node aggregates the information of itself and its neighbors to update its current value. We follow the implementation of the original authors and use GraphNet blocks, defined by an initial update of the edge features that are then aggregated to update the node features. More specifically the updates are (see also Fortunato et al. (2022)):

Low-resolution

$$\mathbf{e}'^l_{ij} \leftarrow \mathrm{MLP}^{E,l}(\mathbf{e}^l_{ij}, \mathbf{v}^l_i, \mathbf{v}^l_j) + \mathbf{e}^l_{ij}\,,$$
$$\mathbf{v}'^l_j \leftarrow \mathrm{MLP}^{V,l}(\mathbf{v}^l_j, \sum_{i\in\mathcal{N}(j)} \mathbf{e}'^l_{ij}) + \mathbf{v}^l_j\,,$$

High-resolution

$$\mathbf{e}'^h_{ij} \leftarrow \mathrm{MLP}^{E,h}(\mathbf{e}^h_{ij}, \mathbf{v}^h_i, \mathbf{v}^h_j) + \mathbf{e}^h_{ij}\,,$$
$$\mathbf{v}'^h_j \leftarrow \mathrm{MLP}^{V,h}(\mathbf{v}^h_j, \sum_{i\in\mathcal{N}(j)} \mathbf{e}'^h_{ij}) + \mathbf{v}^h_j\,,$$

where the MLPs have the same structure as for the encoder, and the sums are over the neighbors of j. The updates include residual connections, which is known to be helpful for convergence (Szegedy et al., 2017). To upsample the data from the low-resolution mesh, we employ the fast k-nearest-neighbor interpolation algorithm introduced in Qi et al. (2017):

$$\mathbf{v}^h_i = \frac{\sum_{j\in\mathcal{N}_k(i)} w_j \mathbf{v}^l_j}{\sum_{j\in\mathcal{N}_k(i)} w_j}\,,\ \text{with}\ w_j = \frac{1}{||\mathbf{u}^h_j - \mathbf{u}^l_i||}\,, \tag{1}$$

where $\mathcal{N}_k(i)$ denotes the k-nearest neighbors of the node i. This is not a trainable operation and thus does not add further parameters to the network. In all of our computations we use $k = 3$ since early tests on the validation dataset showed that increasing k to larger numbers does not add much benefit.

Decoder

The decoder is situated at the end of the processor and transforms the updated embedding into a prediction. We use the same MLP structure as for the encoder, but with only one output feature — the predicted variable.

We implement the neural network in Python with the PyTorch (Paszke et al., 2019), PyTorch Geometric (Fey and Lenssen, 2019), and PyTorch Lightning (Falcon et al., 2020) frameworks. A link to the GitHub repository with our source code can be found in the Data Availability section 7.1. As an optimizer, we use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 0.0005, a weight decay of 0.001, and $\beta_1 = 0.9$, $\beta_2 = 0.95$. The network is trained to convergence with early stopping (Prechelt, 1998) of a patience of 30 on the validation dataset. The loss function employed in this study is the mean absolute error since it tends to improve performance and convergence in super-resolution applications (Wang et al., 2021). Lastly, after a sensitivity analysis (see appendix A.1), we found that setting the number of low-resolution layers to $N_l = 15$ and of high-resolution layers to $N_h = 5$ presented the best efficiency/accuracy ratio of the tested setups.

3.2. Polynomial ridge regression

Neural networks are an excellent tool to model complex nonlinear functions and can approximate a wide range of functions with arbitrary precision (Cybenko, 1989). Nevertheless, it is still of utmost importance to understand the underlying task that one wants to model so as to find the most appropriate method.

In our super-resolution approach, the resolved wave processes at the two resolutions are in theory identical, and at the least, strongly correlated for most of the domain. It is expected that only close to the shoreline the difference in resolution becomes manifest, due to bathymetry-driven processes like wave refraction and depth-induced breaking that are not properly resolved on the low-resolution mesh. Figure 2 visualizes this spatial variation of the correlation between both resolutions for Region 1. Similar plots for the other two regions are found in the appendix in Figure 10 and Figure 11. The linear relationship is displayed by calculating the squared Pearson correlation coefficient r^2 of one low-resolution node (panel (a), loc. 1) with all high-resolution nodes. The correlation is computed over the whole training dataset and the resulting coefficients plotted spatially are shown in panel (c). As a help for interpretation, bathymetry contour lines are shown too. Note that triangles that contained nodes with invalid values are not plotted, which at times results in the omission of valid nodes too. However, in the scatter plots and all performance measures *all* valid values are included.

For the chosen offshore LR node, we observe that the correlation is very high ($r^2 > 0.95$) with almost all high-resolution nodes. A clear drop in the correlation is observed only at a water depth of around 5 m at the beginning of the surf zone. This drop is also observed in the other two regions, and is a indicator that the low-resolution computation is not able to resolve nonlinear phenomena such as shoaling, refraction, and wave breaking correctly. Further insight is gained by the panels (d)-(f), that give concrete examples of the correlation when both low- and high-resolution nodes are offshore, both nodes lie nearshore, or one node lies near- and the other offshore, respectively. While both offshore nodes are strongly correlated, the two last panels indicate that at least some nonlinearity is needed to accurately perform super-resolution. Nevertheless, the strong linear correlations over large parts of the domain indicate that a super-resolution approach based on a multivariate linear or polynomial ridge regression might be pertinent.

In a multivariate linear regression, a linear combination of all input nodes is used to predict one output node (Timm, 2004). Assuming a high-resolution dataset $\mathbf{Y}$ of dimension $(n_{\text{steps}}, n_{\text{HR}})$ and a corresponding low-resolution dataset $\mathbf{X}$ of size $(n_{\text{steps}}, n_{\text{LR}} + 1)$, the prediction $\hat{\mathbf{Y}}$ of the multivariate linear regression can be written as

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W}\,. \tag{2}$$

Here, n_{steps} is the number of time steps in the training dataset (in our study $n_{\text{steps}} = 8\times365\times5+8 = 14\,608$, where the additional "8" is due to a lap year), $n_{\text{HR/LR}}$ the amount of high-/low-resolution nodes of the output/input (see Table 1), and W a $(n_{\text{LR}} + 1, n_{\text{HR}})$ matrix with the weights that have to be determined. Note that n_{LR} is usually increased by one to take into account a constant bias feature. Furthermore, note that while equation (2) computes all high-resolution predictions at once, the predictions of each node are independent of each other. For each high-resolution node, we are trying to find a hyperplane of dimension n_{LR} that minimizes the distance to the instances of the given node over time. Under certain assumptions this equation can be solved with ordinary least-squares (Dempster et al., 1977).

To increase performance it is possible to perform a polynomial regression by artificially adding nonlinear features to the input and computing a multivariate linear regression with those new features (Ostertagová, 2012). The new features are usually polynomial combinations of the previous ones. For example, in the case of only 2 input features x_1, x_2, we could add x_1^2, x_1x_2, x_2^2, x_1^3, $x_1^2x_2$, ... In this article, only transformations of degree $d = 2$ are added to avoid overfitting and an excessive number of input features. Moreover, the scatter plots in panel (e) and (f) of Figure 2 indicate that at least some of the nonlinearities are approximately parabola-shaped.

Even with a relatively low degree, however, a standard multivariate polynomial regression tends to overfit. We found that in some cases the maximum error was on the range of a few centimeters on the training set, but grew absurdly large to more than 60 m on the validation set (even though the mean and median error stayed relatively low). We found that this is mostly due to very large coefficients that balance each other, which works on the majority of cases, but fails catastrophically in others.

To tackle this, we use a variant of polynomial regression called polynomial ridge regression, where instead of only optimizing the squared distances, the value of the coefficients are minimized as well, which can help the regression to generalize better. More precisely, this is implemented by the objective

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = \min_{\mathbf{W}}(||\mathbf{Y} - \mathbf{X}\mathbf{W}||_2^2 + \alpha||\mathbf{W}||_2^2)\,, \tag{3}$$

where α is a hyperparameter to control the strength of the regularization and $\mathcal{L}$ the loss function to minimize. Since the loss function is convex, we can find the minimum by differentiating with respect to $\mathbf{W}$ and setting it equal to zero, which gives the explicit solution

$$\hat{\mathbf{W}} = (\mathbf{X}^{\intercal}\mathbf{X} + \alpha)^{-1}\mathbf{X}^{\intercal}\mathbf{Y}\,. \tag{4}$$

Typical implementations of ridge regression compute the pseudo-inverse of a matrix via a singular value decomposition that scales as $\mathcal{O}(n_{\text{LR}}^2)$. A large number of features, in our case low-resolution input nodes, quickly grows computationally inefficient, especially if ridge regression is paired with polynomial features. Nevertheless, in a super-resolution approach the low-resolution input is usually relatively low dimensional, so that Eq. (4) can be computed directly in most cases. As to our implementation, with a coarse grid search we found that a value of $\alpha = 0.005$ gives the best performance on the validation dataset. In the following, all regression results refer specifically to a polynomial ridge

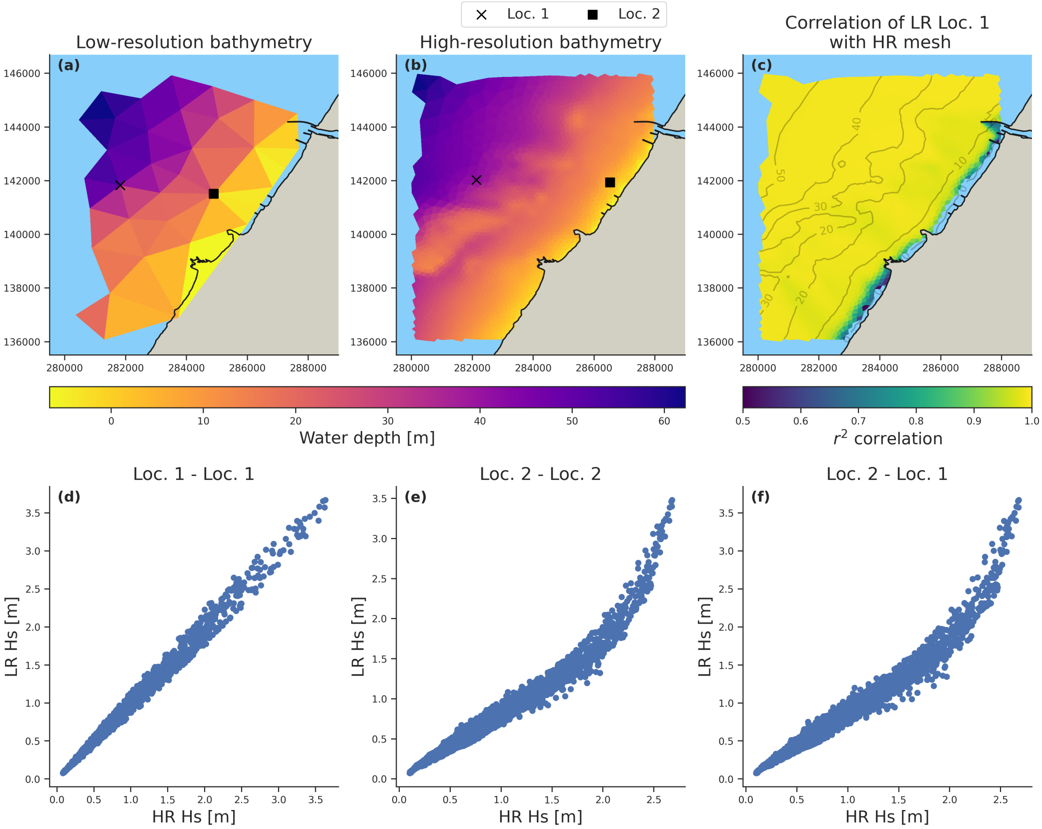

Figure 2: Panel (a) and (b): Low- and high-resolution mesh bathymetries of Region 1 with one off- and one nearshore location highlighted, that are used to calculate the correlations of significant wave height. Panel (c): A map of all the correlation coefficients r^2 of the low-resolution node loc. 1 with all high-resolution nodes. Bathymetry contour lines were added to aid interpretation. Triangles with invalid node values are not plotted, see main text. Panel (d)-(f): Examples of the correlations between different off- and nearshore locations. Note also the differences in the meshes between panel (b) and (c), which is due to the exclusion of NaN values when calculating the correlation map.

regression of degree 2 and $\alpha = 0.005$. As for the GraphNet, we use a combination of significant wave height H_s and mean wave direction θ_m as an input (see the sensitivity analysis in the appendix A.1). To keep the amount of input parameters reasonably low, only H_s is transformed polynomially and θ_m is kept linear. As before, both the low- and high-resolution data are z-normalized over the training data. Table 2 summarizes the hyperparameters used for the GraphNet and the polynomial regression.

3.3. Measuring performance

To compare the models between each other and to the usual errors of spectral wave models, we will focus on two indicators commonly reported in literature, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), as well as the coefficient of determination R^2 in certain cases. Furthermore, to give an idea of the kind of maximal error to expect from the models, we also report the Maximum absolute Error (MaxE). These indicators are computed as follows:

$$
\begin{aligned}
\mathrm{MAE} &= \frac{1}{n_{\text{steps}} n_{\text{HR}}} \sum_{i}^{n_{\text{steps}}} \sum_{j}^{n_{\text{HR}}} |\hat{y}_{i,j} - y_{i,j}| \,, \\
\mathrm{RMSE} &= \sqrt{\frac{1}{n_{\text{steps}} n_{\text{HR}}} \sum_{i}^{n_{\text{steps}}} \sum_{j}^{n_{\text{HR}}} (\hat{y}_{i,j} - y_{i,j})^2} \,, \\
R^2 &= 1 - \frac{\sum_{i}^{n_{\text{steps}}} \sum_{j}^{n_{\text{HR}}} (\hat{y}_{i,j} - y_{i,j})^2}{\sum_{i}^{n_{\text{steps}}} \sum_{j}^{n_{\text{HR}}} (y_{i,j} - \bar{y})^2} \,, \\
\mathrm{MaxE} &= \max_{i \le n_{\text{steps}}, j \le n_{\text{HR}}} |\hat{y}_{i,j} - y_{i,j}| \,.
\end{aligned}
\tag{5}
$$

Here, $\hat{y}_{i,j}$ and $y_{i,j}$ are the prediction of a model and the high-resolution SWAN reference, respectively, at a time step i and at the node of the high-resolution mesh j, while $\bar{y}$ denotes the mean of the reference. Note that these indicators perform averaging or max operations over both each node and each time step, which are helpful to obtain a quick overview of the model in question, but may conceal many important nuances that we will analyze further throughout the next section.

GraphNet

Hyperparameter	Value
Optimizer	AdamW
Learning rate	5×10^{-4}
Weight decay	1×10^{-3}
AdamW β_1	0.9
AdamW β_2	0.95
Early stopping	30 epochs
Loss function	MAE

Polynomial Regression

Hyperparameter	Value
α	5×10^{-3}
Loss function	See eq. (3)

Table 2
Hyperparameters of the GraphNet implementation and the polynomial ridge regression.

3.4. Baseline: Interpolation of low-resolution data

To evaluate the two proposed super-resolution approaches it is important to define a baseline. A widely spread method to convert low-resolution data to a higher resolution is bilinear or bicubic interpolation. It has the advantage that it can be performed on the data directly, without the need for training or additional training data. Furthermore, it is comparatively fast and works reasonably well for slowly-varying data, especially if the initial resolution is fine enough (Xia et al., 2013). Typical implementations, such as in SciPy (Virtanen et al., 2020), compute the convex hull of the area to be interpolated and tessellate it into simplices (typically triangles). The function is then interpolated by computing piece-wise cubic interpolations on those simplices (Alfeld, 1984). As our regions are not entirely convex, we compute the bicubic interpolation only on the convex part and use a k-nearest-neighbors approach with $k = 3$ (see Eq. (1)) on the results of the interpolation to compute the missing values.

Figure 3 gives an example of such an interpolation for a fairly typical sea state in Region 1. The average significant wave height of the test year is 0.83 m, the mean wave period 10.1 s, the direction 301°. Here, the chosen example has values of 0.83 m, 10.9 s, and 308°, respectively.

The interpolated version does capture an averaged value of the wave height, but it completely lacks any details of the spatial variability of the refraction patterns and depth-induced breaking. For a more quantitative estimate of the performance, we compute the MAE, RMSE, and MaxE (see equations (5)) for the three regions averaged over all time steps and over all nodes. The results are displayed in Table 3.

We emphasize again that these are values averaged over all time steps and over all nodes. Depending on the location of the node, the results might differ considerably. In the case of Region 1, for example, the RMSE of an *offshore* node is only 5.7 cm, compared to the much higher, general 20.89 cm. On the other hand, if a *nearshore* node is chosen, the errors shoot up to 41.5 cm, and the maximum error of 3.6 m, as well as the largest errors in general, stem all from nearshore nodes. Large errors are to be expected given the lack of resolution in the original low-resolution input. A maximum discrepancy of 3.6 m is somewhat of an artifact of not well-resolved shorelines and the fact that interpolation does not take into account phenomena like wave breaking close to shore. Nevertheless, a RMSE between 10 cm to 20 cm is not to be neglected, especially when these are only average values and might increase under certain conditions. In the following, we show how a super-resolution approach

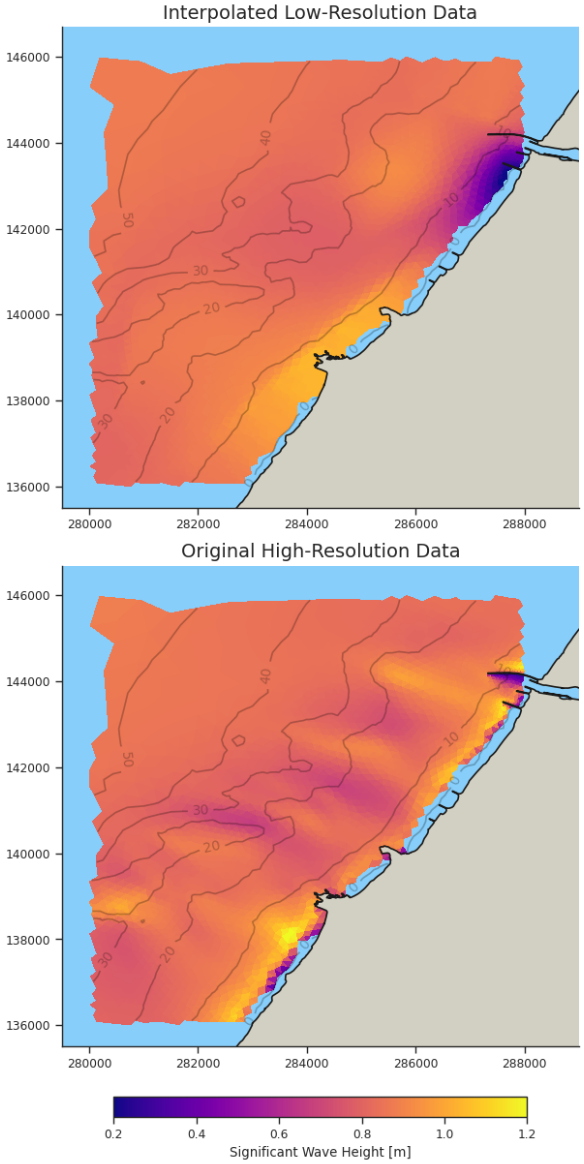
Interpolated Low-Resolution Data
Original High-Resolution Data

Figure 3: Comparison of the interpolated low-resolution data to the high-resolution reference computation for a typical wave field. The interpolation is a combination of bicubic and k-nearest-neighbors interpolation.

Domain	MAE [cm]	RMSE [cm]	MaxE [cm]
Region 1	10.35	20.89	361.63
Region 2	5.17	11.14	165.25
Region 3	7.43	16.41	149.81

Table 3
Performance indicators MAE, RMSE, and MaxE computed on the three regions for a simple interpolation scheme.

with both polynomial ridge regressions and graph neural networks outperform this baseline significantly.

4. Results

Whenever possible, the results are shown for the three study regions in the main text. However, in certain situations, this would lead to very large figures that might hinder comprehensibility. Wherever this is the case, we present the results of Region 1 — the study area with most variability — in the main text and refer to the figures of the other two regions in the appendix.

4.1. Density scatter plots

As an initial comparison between the predictions of the polynomial ridge regression and the GraphNet (GN) model, Figure 4 shows a scatter density plot for Region 1, where the x-coordinates are the high-resolution SWAN results, and the y-coordinates are the respective predictions of the models. Each point corresponds to one node at a given time, so the total number of points is $n_{\text{steps}} \times n_{\text{HR}}$ (e.g., $2928 \times 2338 \approx$ 7 million points for Region 1). To facilitate the distinction of denser regions, the number of points per pixel are binned and density is indicated by color. As a supplementary tool for interpretation, histograms on both axes show the distribution of the data. Similar figures for the other two regions are found in the appendix.

Both models reproduce the high-resolution results well with no major outliers throughout the whole test year. The spread around the red reference line is slightly cone-shaped with the tendency of smaller errors at small wave heights, that increase up to a certain extent with increasing wave heights, a trend that we will discuss further below. Both models slightly overestimate large wave heights. While the predictions are relatively similar at first glance, the GraphNet exhibits a larger spread at around 3 m to 4 m than the polynomial regression, the inverse holds true at around 1 m to 2 m for the latter.

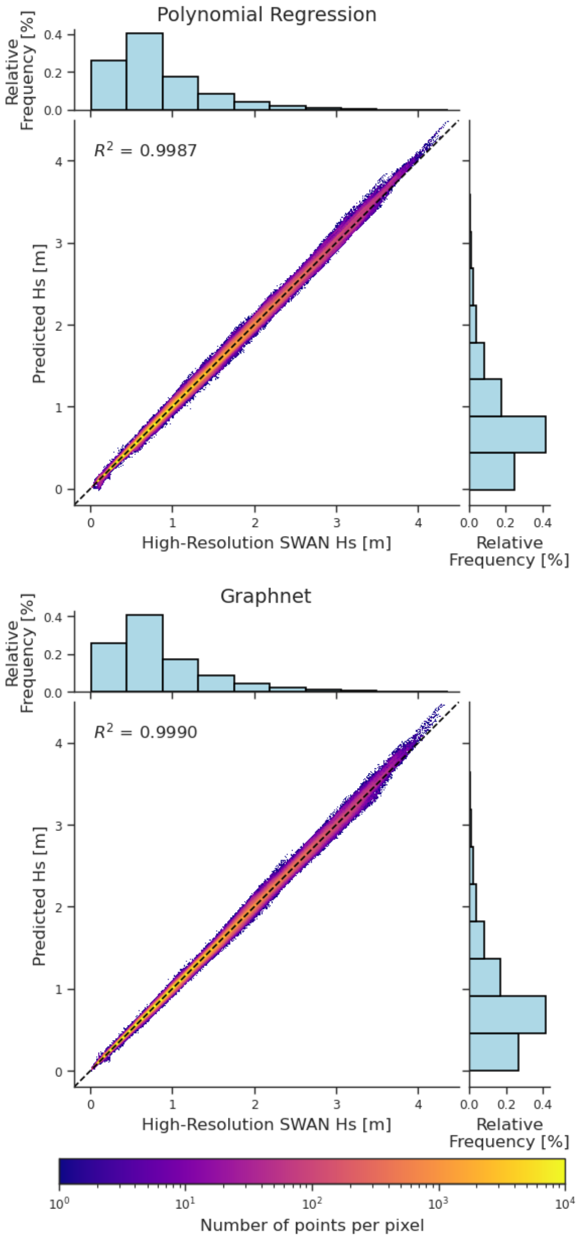

Figure 4: A scatter density plot of the model predictions for Region 1, where color indicates the number of points per pixel on a logarithmic scale. Additionally, histograms on both axes show the distribution of the significant wave heights. The upper figure refers to the predictions of the polynomial ridge regression, the lower to the GraphNet.

For Region 2 and Region 3 (see Figure 12 and Figure 13 in the appendix) the polynomial regressions perform clearly better than the GraphNet, even though the latter still shows excellent results. As before, the variation of the prediction is the highest at around 3 m to 4 m for both models and both regions, even though this trend is more pronounced for the GraphNet. In the case of Region 3, apart from some minor discrepancies, the prediction is very accurate and becomes difficult to distinguish from the reference diagonal. Note also that for all three regions, more than 60 % of the data points have a significant wave height smaller than 1 m, a range where the deviations are particularly small.

The variability of the performances in the different regions is to be expected to a certain extent. Region 1, for example, has a wave field that is governed by its strongly varying bathymetry that gives rise to a variety of different, localized wave focusing and shadowing zones, only resolved in the higher resolution. On the other hand, the bathymetry profile of Region 3 is almost perfectly uniform longshore

	Region 1		
	MAE [cm]	RMSE [cm]	MaxE [cm]
Poly. Regr.	1.32	2.05	24.80
GraphNet	**1.13**	**1.79**	**23.76**

	Region 2		
	MAE [cm]	RMSE [cm]	MaxE [cm]
Poly. Regr.	**0.41**	**0.64**	**14.52**
Graphnet	0.55	0.85	**14.52**

	Region 3		
	MAE [cm]	RMSE [cm]	MaxE [cm]
Poly. Regr.	**0.16**	**0.27**	**4.85**
Graphnet	0.30	0.48	**7.89**

Table 4
Mean absolute, root mean square, and maximum prediction errors of the polynomial regression and GraphNet for the three study regions, averaged over all nodes and the entire test year.

at both spatial resolutions. As a consequence, the low-resolution SWAN computation is already a highly valuable and correlated input and the predictive models only have to learn some minor variations, mostly in the surf zone.

4.2. Wave height-dependent error distribution

As the scatter plots only give a global overview of the performance, it is crucial to study the errors on a finer scale. In Table 4, we compute and present the MAE, RMSE, and MaxE for the polynomial regression and GraphNet models and highlight in bold the best performances for the given region per indicator. For Region 1, the GraphNet outperforms the polynomial regression slightly on each of the indicators, but performs less well than the latter on the other two regions. This trend is confirmed in panel (a) of Figure 5 that adds further nuances with boxplots of the absolute differences between the model predictions and the SWAN reference computations. The lower and upper whiskers extend to the 5th and the 95th percentile, the edges of the boxes to the 25th and 75th, respectively. The black line inside the box denotes the median, whereas the white dot indicates the mean.

Region 1 is the most difficult study area to predict for both models, with the third quartile being larger than almost all of the other 95th percentiles. However, the errors are still less than 2 cm for 75 % of the data. For Region 2, this increases to 95 % and in Region 3 almost all of the errors are smaller than 1 cm.

In the panels (b)-(d) the error distributions are further split into bins of the incident wave height at that node. More precisely, for each prediction the absolute difference to the reference computation $|\hat{y}_{i,j} - y_{i,j}|$ is put into a certain bin depending on the value of $\hat{y}_{i,j}$. This helps to estimate the error at a certain location given its significant wave height. Mostly, for each region and for both model types, the absolute error tends to increase with increasing wave height. While this trend is quite distinct in Region 1, it is less so in the other two regions, where at the largest wave heights the error decreases after an initial rise. This is in accordance with the previous scatter density plots in Figure 4, where the slight cone shape hints at a multiplicative error of the prediction. This means that the high-resolution reference can be approximately described as a function of the prediction by $y_{i,j} \approx \hat{y}_{i,j} e_{i,j}$, with $e_{i,j} \in \mathbf{E}$ for a constant error matrix $\mathbf{E}$. With this (simplified) assumption, the absolute error simplifies to $|y_{i,j} e_{i,j} - y_{i,j}| = |y_{i,j}(e_{i,j} - 1)|$ with a linear dependence on the wave height. A possible source of this multiplicative error might be explained by the individual correlation plots of panel (d) in Figure 2 and to a lesser extent in Figure 10. The two wave heights form again a relatively cone-shaped linear relationship, with little variance at low wave heights, that increases with larger values. We suspect that this trend is a direct consequence of how SWAN computes results at different resolutions, but could not find any studies on wave-height dependent analyses of the differences between SWAN computations at different resolutions.

A multiplicative error as described above implicates that the percentage of the errors does not vary with wave height. The error might be 5 % of 50 cm, but also 5 % of 5 m, which seems plausible. A more extensive analysis of the error trend is out of the scope of this study and the explanations provided here are more a simplified interpretation of the results rather than a rigorous proof. As seen in Region 2 and 3, the drop of the wave heights after an initial rise defies this first-order approximation. Furthermore, for both the PR and GN the dependence on the low-resolution wave height is not linear and consequently the way that the cone-shape in Figure 2 contributes to the final predictions is decidedly more complex.

4.3. Spatial error comparison

While Figure 5 gives a good overview of the performances of the different models in the three regions in various wave regimes, it contains no information about where the errors occur. A linear regression might, for example, work exceptionally well in the linearly correlated areas of the low- and high-resolution computations, but perform poorly close to the shoreline and in wave refraction areas. To compare the error of the models spatially, we calculate the MAE over the test year for each node.

Figure 6 presents the results for each region and both models in the form of error maps. From the figure it is clear that models with a lower error perform better not only locally but rather throughout the whole region. This is particularly evident in Region 1 where the error of the ridge regression model has a distinct spatial dependence, but compared to the GraphNet the error is still higher over almost the entire domain. The remark is also true for the other two regions, but with a larger error for the GraphNet method. In general, the errors are larger closer to the shoreline or on wave refraction patterns for both models. In the case of the polynomial

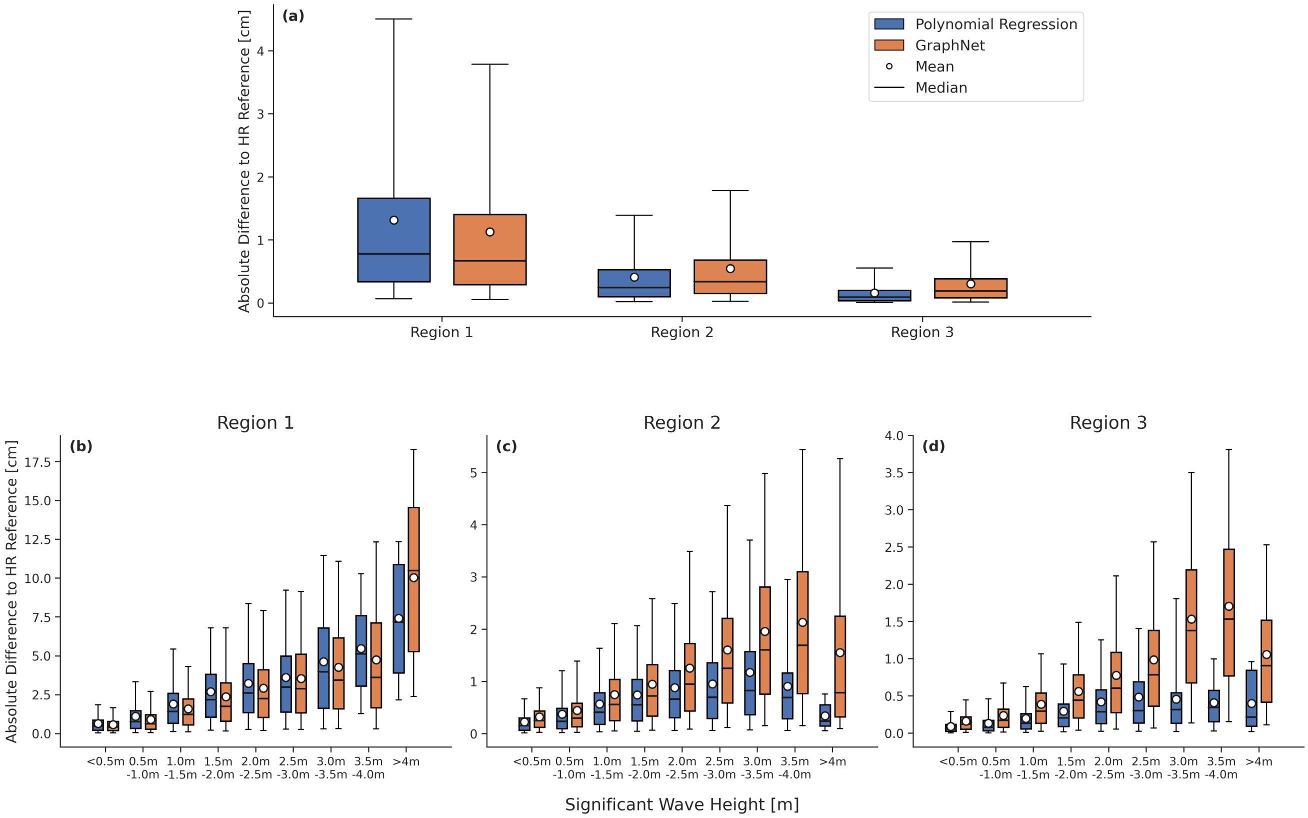

Figure 5: Absolute differences of the polynomial ridge regression and GraphNet models in comparison to the high-resolution SWAN computation. The panels characterize the distribution of the error with boxplots. Panel (a) details the errors globally for each region, panel (b)-(d) focus on the absolute error at different wave heights.

regression, this is especially clear and the shape of the wave field is markedly visible for all three regions. This is not too surprising given that offshore nodes are likely well predicted due to the strong linear relationship between the LR and HR datasets in that area. As for the shoreline and the refraction zones, they are not at all, or only barely, resolved in the low-resolution input and are thus much more prone to errors, also due to the fact that nonlinearities are only modeled up to a second degree. The same arguments apply to the graph neural network, with the difference that the method is inherently nonlinear, so that the distinction of the errors in the linear and the nonlinear regime are less clear. This more homogeneous distribution can be observed especially for the first and third region, even though some wave refraction zones can still be easily discerned.

While for Region 2 and 3, the polynomial regression is outperforming GraphNet considerably, the latter still exhibits impressive results, with mean errors on the scale of 1 cm for Region 2 and only around 0.5 cm for Region 3 — a scale of errors that is negligible in almost all use cases.

4.4. Computation time

The advantage of a super-resolution approach over a classical numerical wave model computation is that it can reduce the computation time considerably without sacrificing much accuracy. Once the data-driven models are trained, the conversion from low to high resolution by the polynomial regression and GraphNet is done very quickly, the limiting factor is usually how fast the low-resolution input can be obtained in comparison to the finer resolution. To fairly compare the computation times, the original SWAN computation times have to be divided up according to the regions since the numerical wave models ran on the full study area, but the data-driven models only on a specific region. While the true, underlying computation times are surely different, we approximate it by attributing to each region a time proportional to its number of nodes. More specifically, for Region 1 the times are computed as follows: $T_1 = \frac{n_1}{n_{\text{tot}}} T_{\text{tot}} \approx 5.4\,\% T_{\text{tot}}$ for the high-resolution and $3.4\,\% T_{\text{tot}}$ for the low-resolution. Similarly, the percentages for Region 2 are 8.3 % and 6.5 %, and for Region 3 they are 6.0 % and 9.1 %, respectively. Table 5 shows the original computation times of the high- and low-resolution SWAN computations, along with the fractions corresponding to the regions.

Furthermore, the training and inference times of the data-driven models for all three regions are provided in Table 6. The last column indicates the gained speed-up of the super-resolution approach for this particular model in the given region. The speed-up is computed as

$$\text{Speed-up} = \frac{T_{\text{HR}}}{T_{\text{LR}} + T_{\text{Pred}}}, \tag{6}$$

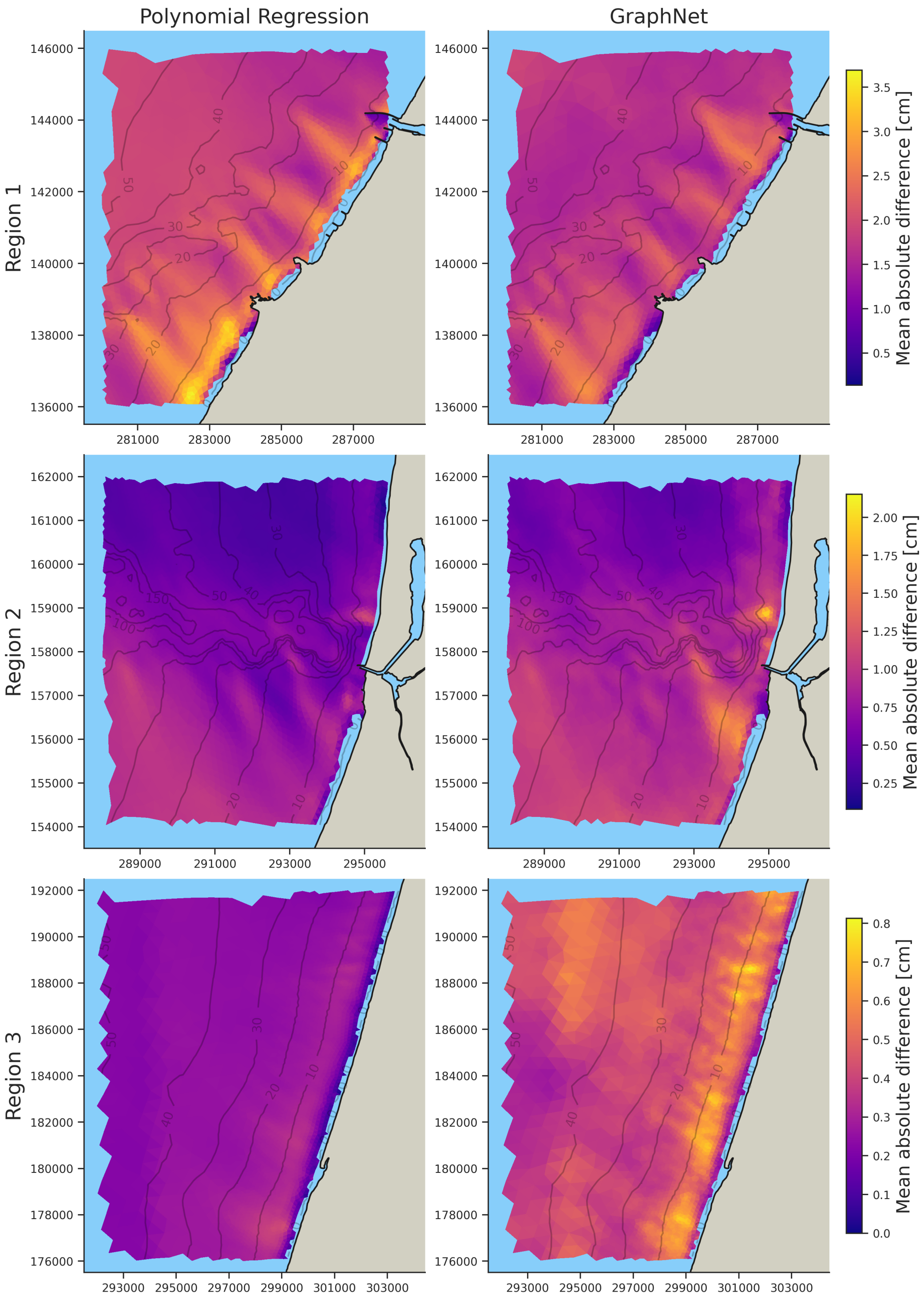

Figure 6: A node-to-node mean absolute error map for polynomial ridge regression and GraphNet models for the three different regions.

where T_{HR} is the computation time of the high-resolution SWAN model (column "High-Resolution" in Table 5), T_{LR} the corresponding computation time of the low-resolution SWAN input (column "Low-Resolution in Table 5), and T_{Pred} is the inference time of an already trained data-driven model (column "Prediction" in Table 6).

All time measures, apart from the training times, refer to the whole test year. The training times refer to training the model on a total of 5 years, with significant wave height and wave direction as an input. Both SWAN computations and the training and testing of the polynomial regression were performed on 6 parallel threads of an Intel Core i7 9750H processor with a maximum clock rate of 4.5 GHz. The graph neural network was trained and tested on a Nvidia GeForce RTX 2070 Mobile Graphic Card with 8 GB VRAM. Note also that while the computation times of the SWAN and polynomial ridge regression models are relatively constant, the *training* time of the GNN can vary considerably due to the stochastic nature of the optimization process. During training, the model might get stuck for some time in a local minimum, or the randomly chosen batches might align in an opportune way for a fast convergence. Here, we report the average of three training runs, rounded to the next thousand seconds. We point out that the *prediction* time on the other hand, does not vary since only non-stochastic calculations are performed.

Tables 5 and 6 indicate that, while the speed-up, inference and training times differ for different regions and models, a super-resolution approach is always distinctively faster than a traditional, high-resolution numerical wave computation with gains in computation time that reach a factor of 80. However, an important caveat is that, as for most data-driven approaches, this only holds true for already trained models. The generation of the training sets, which are not included in this calculation, is a non-negligible initial computational effort. Furthermore, in the case of the GNN, not only obtaining the training set, but also actually training the model is rather time-intensive and might warrant a more extensive future usage to actually reduce computation time with a super-resolution approach.

Region	High-Resolution [s]	Low-Resolution [s]
Full	181 000	4020
1	9800	120
2	15 000	240
3	10 800	360

Table 5
SWAN computation times for the low- and high-resolution mesh. For the three regions, the times correspond to fractional times proportional to their number of nodes. The region "Full" indicates the entire study area.

	Region	Training [s]	Prediction [s]	Speed-up
PR	1	1	0.1	81
	2	8	0.7	62
	3	17	0.7	30
GN	1	40 000	22	70
	2	54 000	31	55
	3	38 000	23	28

Table 6
Prediction and training times of the polynomial ridge regression (PR) and the GraphNet (GN) model as well as the speed-up (see Eq. (6)) in comparison to directly computing the high-resolution SWAN result for the three different regions. The training times of the GraphNet models is an average over three runs and rounded to the nearest thousand seconds.

5. Discussion

The previous sections showed that a super-resolution approach can decrease the computation time of significant wave height by spectral wave forecasts on an unstructured mesh considerably. While the graph neural networks were able to convert the low-resolution inputs to a higher resolution with remarkable accuracy, we found that a polynomial ridge regression performed in many cases equally well, if not better. The advantage of the latter is that the coefficients of its weight matrices can be calculated analytically, provided that the number of input features times the number of output features does not grow too large and fits into memory. Indeed, this analytical solution can be computed very quickly. Taking Region 1 as an example, for a conversion of roughly 500 (polynomial) input features to 2400 output nodes for 5 years of training data, the weight matrix is determined in less than a second on our (commodity) hardware. What is more, the prediction of an entire year of data, with the computation of the low-resolution included, is only slightly more than $120\,\mathrm{s} = 2\,\mathrm{min}$, in contrast to a direct high-resolution approach that takes roughly $9800\,\mathrm{s} = 163\,\mathrm{min}$ (compare with the fractioned times of Table 5). Certainly, a super-resolution approach can never be 100 % accurate since at least some information is missing in the low-resolution input, due to the underdetermined nature of the problem. Even so, the results presented in the previous section suggest that for the vast majority of the time, the errors of the predicted significant wave height are remarkably low in our study area. Even for Region 1, a study area with a highly heterogeneous wave field, the absolute error is lower than 5 cm for 95 % of the data. In less complex regions, like Region 3, this errors drops as low as 1 cm.

This error can be considered negligible compared to the errors usually reported when confronting spectral wave models with coastal and nearshore measurements. For example, in the same study region, Delpey et al. (2021) performed a detailed comparison of SWAN model results with several field measurement datasets. They used a refined modeling strategy targeting specifically the nearshore area around Biarritz (our Region 1), with a maximum spatial resolution of up to 10 m, while including tidal water level oscillations and local wind generation into the computation. With this remarkably high resolution model they report RMSEs of 19 cm to 33 cm and a Normalized RMSE of 11 % to 19 %, which can be considered very accurate in the complex nearshore study area. However, in comparison, the RMSEs of our super-resolution approach range from approximately 0.3 cm to 2.0 cm for the polynomial ridge regression. This is equivalent to adding only 1 % to 2 % to the Normalized RMSE, for a 30-80 times faster calculation.

In certain cases, very accurate predictions are important and further inaccuracies are not acceptable, but other applications could benefit from a slightly less accurate but much faster computation. This could be the case for emergency forecasts, when wave predictions have to be updated quickly to support real-time crisis management. Another case might be ensemble forecasting, where the speed-up would allow to increase the number of ensemble members, compensating the small accuracy loss due to super-resolution.

5.1. Comparison of graph neural networks and polynomial regressions

In our study, we found that the predictions of a polynomial ridge regression worked as well as or better than the predictions made by a much more complex graph neural network. At first glance, this seems surprising. The chosen architecture is based on a neural network able to predict accurately complex fluid mechanical phenomena such as turbulence (Pfaff et al., 2021) and a variant of it even outperforms the weather forecasts of the highly reputable European Centre for Medium-Range Weather Forecasts. Moreover, the amount of parameters of the neural network is orders of magnitude larger than for the polynomial ridge regression. Nevertheless, the very strong linear correlation between the two resolutions reduces the super-resolution problem to a practically linear task with nonlinear perturbations at given locations. Addressing these perturbations with polynomial features reduces the errors, as seen in the results, even in the refraction and breaking zones (although the polynomial approximation is clearly not perfect). Neural networks are by design nonlinear, and may produce sub-optimal results in strongly linear tasks. In theory, a neural network can approximate a wide variety of different functions, including linear ones, but they might need a large amount of parameters to do so (Brüel Gabrielsson, 2020). This problem can be likened to approximating a straight line with a Fourier series. Given enough terms it is possible to get arbitrarily close, but estimating a straight line directly is easier and more accurate.

Still, a polynomial regression has its disadvantages and a graph neural network its advantages. If the number of features grows too large, the solution to the polynomial regression might not be calculated directly and has to be computed with iterative methods such as gradient descent, similar to a neural network. This slows down training substantially and convergence problems might arise, especially for strongly correlated features (Tran et al., 2015). Furthermore, the regressions are always only calculated for one study area and it is not possible to re-use the results on another domain. Graph neural networks, on the other hand, use weights based on the interactions of nodes with their nearest neighbors and do not have rigid weights linked to individual nodes. This makes it possible to train a graph neural network for multiple regions at the same time, resulting in a more versatile tool. However, in preliminary studies we found that specialized networks focusing on only one specific region had better performances.

Another advantage of GNNs is that adding different input features, such as mean wave period or direction is very natural; at each node the features are concatenated to the existing ones and computations are performed separately on all the features. In the case of a polynomial regression, on the other hand, each additional feature has to be connected to every single output node, thus adding n_{HR} additional weights.

Lastly, in our case the problem was strongly linear, especially for Region 2 and 3, but in the more nonlinear Region 1, the GraphNet did perform slightly better. This difference could grow larger in cases where linear correspondence between the low- and the high-resolution is less strong, such as in highly variable wave fields or situations where the low-resolution stems from a different computation or wave model than the high-resolution. For example, one might want to use a low-resolution input that is taken from an external database based on a coarse WAVEWATCH III run to convert it to a higher resolution, training on an internal SWAN hindcast. Now a super-resolution approach has to take not only the differences in resolution into account, but also the model differences (WAVEWATCH III to SWAN).

5.2. Computation time analysis

We mentioned in section 4.4 that, when we refer to the speed-up of a model, we do not include neither the training time of the data-driven model, nor the actual creation of the training dataset. This process can potentially be quite time consuming, but we argue that there are multiple ways to justify this initial effort. First, in Kuehn et al. (2023), we compute an estimate of how long a hindcast has to be in order for super-resolution to be beneficial. The limiting factor, in that example, was the training time of the neural network since it was 2-3 times longer than computing one year of high-resolution data. In the given study, this problem is mitigated for the neural network since the training time for one year (column "Training" in Table 6 divided by 5) is already less than computing one year of high-resolution results.

Assuming that we want to reconstitute the 44-year hindcast of this study for Region 1, then, using the high-resolution SWAN model would take roughly 5 days. With the graph neural network trained on 5 years, this is reduced to 1.1 d, creation of the training set, training, low-resolution computations and their conversions included. Note that the 5 years of high-resolution data to set up the training dataset can (and should be) used to constitute this new hindcast. Finally, with a polynomial regression the computation time further drops down to only 0.6 d, where a bit more than 0.5 d are due to the calculation of the high-resolution training data. Thus, everything included, the super-resolution approach is 5-10 times faster in those two cases. Note that for predictions over a long period of time (almost 40 years in this case), the wave climate might evolve — due to climate change or morphological changes of the domain — which could negatively affect prediction performance.

It is often the case that one of the datasets is already provided. For example, in our case, the high-resolution hindcast was already available, thus reconstituting the 44-years with the polynomial regression (the low-resolution calculation included) would only take 1.5 h. Theoretically, we could go even further, assuming that we want to increase the resolution of some low-resolution results that are readily available in a database (and for example might get updated every month). If a short high-resolution hindcast is available in the same region, a super-resolution model could be trained on these two datasets and with every future update, the high-resolution results are obtained almost immediately. Lastly, in the case of ensemble forecasting mentioned earlier, the data-driven models can be used over and over again for multiple computations, compensating quickly the initial effort to set these models up. It is due to these various possible setups that we chose to exclude the training time and the creation of the dataset in our speed-up calculations. These are only one-time costs and might be vastly different depending on various factors, such as the length of the training dataset. For an extensive usage of the super-resolution model, the overall speed-up with all steps included (creation of the datasets, training of the model, etc.) converges to the speed-up factor of an already trained model.

The discussion about the speed-up still needs to be further nuanced. In this study, the machine learning models are trained to reproduce only the significant wave height. A direct computation with a numerical wave model like SWAN, on the other hand, is able to compute as many bulk parameters as needed (or output directly the entire spectrum) with a negligible increase in run time. To have the equivalent amount of information with the approach proposed in this study, one model would have to be trained for each bulk parameter, which can quickly result in a significant upfront computational investment. In the case of a GNN, it is technically possible to train a model that predicts multiple bulk parameters simultaneously, but this commonly leads to inferior results (Schultz et al., 2021). Additionally, as it is, the data-driven model can not work with spectral data directly. Consequently, if many different bulk parameters are required for an application, a super-resolution approach might be ultimately slower than a direct computation. An exciting future research direction to mitigate this problem is the extension of the super-resolution approach to wave spectra, which then can be used for a direct calculation of the required bulk parameters.

5.3. Outlook

While super-resolution is already well established in computer vision and is recently receiving a lot of attention in fluid mechanics, its application to numerical coastal wave models remains limited for the moment. Apart from our current and recent work (Kuehn et al., 2023), there were only two other similar publications (Chen et al., 2023; Zhu et al., 2023). This highlights the need for further research of super-resolution in coastal wave modeling. One of the factors that influence the results considerably is the cell size and node number of the low-resolution input. Given a finer input, the results are likely more accurate, but the speed-up is diminished, whereas the opposite holds true for a coarser input. However, this notion has to be studied more deeply. It is not clear if the relationship is linear or if there might be diminishing returns with possibly an optimal mesh size for a given problem. Analyzing this problem on structured grids is facilitated by a clear definition of what a reduction in resolution means since the sides of the rectangular cells can be divided by a given factor. On unstructured meshes, this approach is not as clear-cut, given that no one correct downsampling method exists, particularly one that guarantees the convergence of the numerical solver. A few initial tests with our downsampling method hint at a nonlinear dependence on resolution, but more research is needed for conclusive results.

Furthermore, in this study, super-resolution was tested specifically on the spectral coastal wave model SWAN, but applications on other phase-averaged models (e.g., WAVEWATCH III) or also phase-resolved models such as XBeach (Roelvink et al., 2010) or BOSZ (Roeber and Cheung, 2012) could be of great interest. Additionally, super-resolution could be potentially also extended to the prediction of surf-zone hydrodynamics, where it could be compared to other hybrid approaches such as HySwash (Ricondo et al., 2024).

Another potential research direction is the inclusion of physical equations or wave measurements to potentially improve upon the high-resolution SWAN results. For example, Chen et al. (2021b) and Lam et al. (2023) use buoy wave measurements and reanalysis hindcasts, respectively, to create surrogate models that outperform their respective reference numerical model used in their studies. A similar approach is certainly applicable to super-resolution forecasts too and was already employed in the field of fluid mechanics by Fortunato et al. (2022) for wake flow simulations. Finally, some physics-informed neural networks integrate prior knowledge about the governing physical equations into the neural network, that makes them more robust to out-of-distribution samples and helps them generalize to new domains (Raissi et al., 2019; Gupta and Brandstetter, 2022). While these approaches are relatively difficult to implement for complex numerical wave models such as SWAN or WAVEWATCH III, recent advances on integrating advection-diffusion equations (de Wolff et al., 2021), shallow water equations (Bihlo and Popovych, 2022; Giladi et al., 2021), or energy balance equations (Wang et al., 2022) show the potential and the great interest of this emerging domain.

6. Conclusion

Our study shows that data-driven super-resolution can be an efficient tool to quickly and accurately compute wave fore- and hindcasts of specific bulk parameters of the sea state. Once the data-driven model is trained, this approach is able to convert a low-resolution wave computation to much higher resolution of equivalent quality, thus bypassing lengthy direct computations over fine grids. We applied this approach on the hindcast presented in Lastiri et al. (2020) and were able to reduce the computation time by up to 80 times, with RMSEs of only around 2 cm or less — errors that are minor for most wave forecast applications.

In particular:

- A polynomial ridge regression performed either almost equally well or better than a graph neural network for the study domains considered in our region. The strong linearity between low- and high-resolution results favors linear approaches, even though the graph neural network still performs very well.
- Performances are dependent on wave height, with a slight linear correlation between the median absolute error and the wave height regime. Furthermore, the errors also depend on the overall spatial variability of the wave field, which is mostly controlled by bathymetric features in our study case. More homogeneous bathymetries lead to predictions with considerably smaller errors.
- Over an entire test year, the maximum absolute error, all nodes and all time steps included, stays below 25 cm, but for certain regions and models is as low as 5 cm. The RMSE for 2 out of 3 study areas is smaller than 1 cm.

Data-driven models are increasingly gaining in popularity in natural sciences, and while they will not and should not replace numerical models, they are a useful complementary tool in scientific computing. We believe that super-resolution approaches make an important contribution to coastal wave forecasts, offering a balance between slow, but accurate direct numerical calculations and the faster, but more error-prone surrogate models. Future research could extend the approach to spectral data, making it more versatile for different applications.

7. Open research

7.1. Data availability

The pre-processing and machine learning scripts along with Jupyter notebooks for the figures are available through GitHub under `https://github.com/janfer95/coastal-super-resolut`

CRediT authorship contribution statement

Jannik Kuehn: Conceptualization, Methodology, Software, Writing - Original Draft. **Stéphane Abadie:** Conceptualization, Writing - review & editing, Supervision, Funding acquisition. **Matthias Delpey:** Conceptualization, Writing - review & editing, Supervision. **Volker Roeber:** Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Acknowledgments

This research was carried out in the framework of the E2S-UPPA chair HPC-Waves. The authors acknowledge financial support from E2S-UPPA, the Communauté d'Agglomération Pays Basque (CAPB), and the Communauté Région Nouvelle Aquitaine (CRNA) for the chair HPC-Waves.

A. Appendix

A.1. Sensitivity analysis

This section analyzes the influence of certain architectural choices in section 3 on the mean absolute and root mean square error of the predictions of the GraphNet and polynomial regression. More particularly, the focus is on the choice of the different input variables and the number of low- and high-resolution layers in the GraphNet.

Dependence on input variables

The goal of super-resolution in this study is to convert low-resolution results of a variable, such as significant wave height, to a higher resolution. Nevertheless, other wave parameters might contain complementary information and might help to improve performance. Figure 7 compares the MAE and RMSE of four different combinations of inputs: only significant wave height (H_s), significant wave height and mean wave period ($H_s + T_{m01}$), significant wave height and mean wave period ($H_s+\theta_m$), and all three input variables together (All). The errors are calculated on the validation dataset. All models exhibit the same trend for both the MAE and RMSE that shows that additional variables do improve the performance in comparison to a H_s-only approach. While the addition of the mean wave period T_{m01} has only a minor impact on the performance, adding the mean wave direction θ_m reduces the error considerably. A further, much smaller, reduction of the error is obtained by

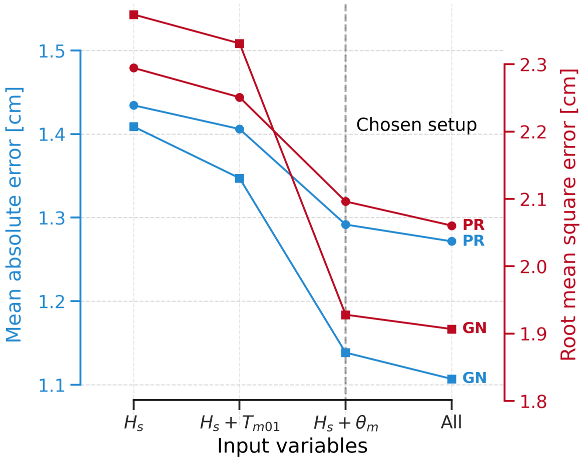

Figure 7: Mean absolute and root mean square error of the GraphNet architecture and the polynomial regression for different combinations of input variables. Errors computed on the validation dataset of Region 1.

using a combination of all variables. To keep the number of parameters as low as possible (particularly in the case of the polynomial regression), while still retaining a good performance, we opt for the $H_s + \theta_m$ setup, since the benefit of adding the mean wave period is minor.

Dependence on number of message passing layers

The GraphNet architecture contains various hyperparameters, such as latent vector width, activation functions, number of MLP layers and the size of their hidden layers. However, a sensitivity analysis by Pfaff et al. (2021) showed that the GraphNet is mostly insensitive to those hyperparameters, except for the number of message passing layers / GraphNet blocks, where more layers led to a better performance, but also to longer computations. They found that a total of 15 message passing layers presented a good efficiency/accuracy ratio. Fortunato et al. (2022) obtained satisfying results with 15 to 25 layers, with varying amounts of low- and high-resolution layers. Here, we set the total number of message passing layers to 20 (a middle ground between the two articles) and look at three different combinations of message passing layers at low and high resolution: 5 low-resolution layers followed by 15 high-resolution layers (5-15), 10 low-resolution layers followed by 10 high-resolution layers (10-10), 15 low-resolution layers followed by 5 high-resolution layers (15-5). Figure 8 shows the MAE and RMSE computed over the validation dataset for these three combinations. In the case of the RMSE, the error decreases with an increasing number of low-resolution layers; in the case of the mean absolute error, the minimum is at 10-10, although 15-5 presents a similar error. Note also that the variations of the error are much smaller compared to the influence of the input variables (e.g., a difference of 0.01 cm to 0.03 cm between the MAEs for the number of layers in contrast to 0.05 cm to 0.2 cm between variables). Another aspect that has to be taken into account is the training time, that is the highest for the 5-15 setup (20.5 h), followed by 10-10 (14.5 h), and the lowest for 15-5 (8.5 h). Given the better computational efficiency and similar or better accuracy, we opt for an architecture with 15 low-resolution and 5 high-resolution layers.

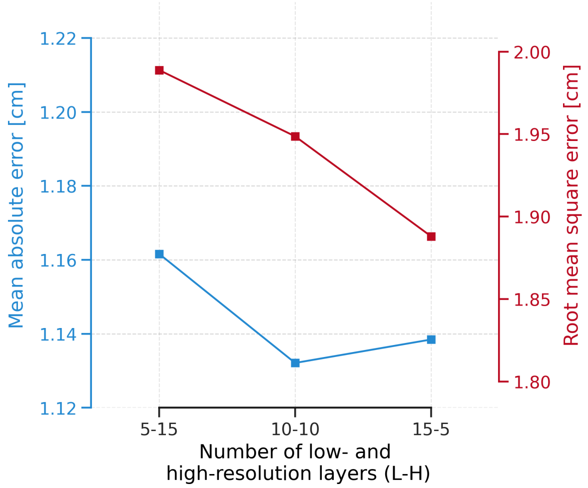

Figure 8: Mean absolute and root mean square error of the GraphNet architecture for different numbers of low- and high-resolution layers. Errors computed on the validation dataset of Region 1.

A.2. Additional figures

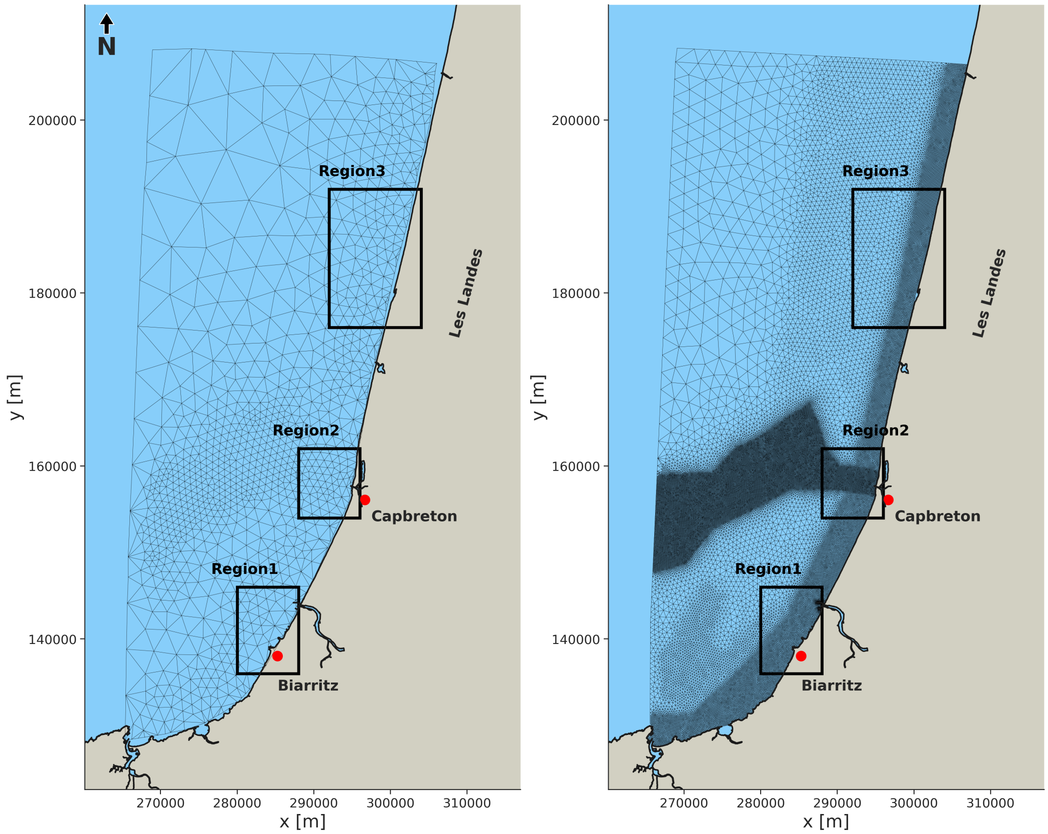

Figure 9: Low- and high-resolution mesh employed for the SWAN computations, as well as the locations of the study regions. The low-resolution mesh contains 1059 nodes and the high-resolution mesh 45 156.

References

Abadie, S., Butel, R., Mauriet, S., Morichon, D., Dupuis, H., 2006. Wave climate and longshore drift on the South Aquitaine coast. Continental Shelf Research 26, 1924–1939. doi:10.1016/j.csr.2006.06.005.

Alfeld, P., 1984. A trivariate clough—tocher scheme for tetrahedral data. Computer Aided Geometric Design 1, 169–181. doi:10.1016/0167-8396(84)90029-3.

Ardhuin, F., Roland, A., Dumas, F., Bennis, A.C., Sentchev, A., Forget, P., Wolf, J., Girard, F., Osuna, P., Benoit, M., 2012. Numerical Wave Modeling in Conditions with Strong Currents: Dissipation, Refraction, and Relative Wind. Journal of Physical Oceanography 42, 2101–2120. doi:10.1175/JPO-D-11-0220.1.

Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., Pascanu, R., 2018. Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261.

Battjes, J.A., Stive, M.J.F., 1985. Calibration and verification of a dissipation model for random breaking waves. Journal of Geophysical Research 90, 9159. doi:10.1029/JC090iC05p09159.

Belbute-Peres, F.d.A., Economon, T., Kolter, Z., 2020-07-13/2020-07-18. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction, in: III, H.D., Singh, A. (Eds.), Proceedings of the 37th International Conference on Machine Learning, PMLR. pp. 2402–2411.

Bihlo, A., Popovych, R.O., 2022. Physics-informed neural networks for the shallow-water equations on the sphere. Journal of Computational Physics 456, 111024. doi:10.1016/j.jcp.2022.111024, arXiv:2104.00615.

Booij, N., Ris, R.C., Holthuijsen, L.H., 1999. A third-generation wave model for coastal regions: 1. Model description and validation. Journal of Geophysical Research: Oceans 104, 7649–7666. doi:10.1029/98JC02622.

Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P., 2021. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. arXiv:2104.13478.

Brüel Gabrielsson, R., 2020. Universal function approximation on graphs, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 19762–19772. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/e4acb4c86de9d2d9a41364f93951028d-Paper.pdf.

Cachay, S.R., Erickson, E., Bucker, A.F.C., Pokropek, E., Potosnak, W., Bire, S., Osei, S., Lütjens, B., 2021. The World as a Graph: Improving El Niño Forecasts with Graph Neural Networks. arXiv:2104.05089.

Callens, A., Morichon, D., Abadie, S., Delpey, M., Liquet, B., 2020. Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. Applied Ocean Research 104, 102339. doi:10.1016/j.apor.2020.102339.

Camus, P., Mendez, F.J., Medina, R., 2011. A hybrid efficient method to downscale wave climate to coastal areas. Coastal Engineering 58, 851–862. doi:10.1016/j.coastaleng.2011.05.007.

Capel, D., Zisserman, A., 2003. Computer vision applied to super resolution. IEEE Signal Processing Magazine 20, 75–86. doi:10.1109/MSP.2003.1203211.

Charles, E., Idier, D., Thiébot, J., Le Cozannet, G., Pedreros, R., Ardhuin, F., Planton, S., 2012. Present Wave Climate in the Bay of Biscay: Spatiotemporal Variability and Trends from 1958 to 2001. Journal of Climate 25, 2020–2039. doi:10.1175/JCLI-D-11-00086.1.

Chen, J., Hachem, E., Viquerat, J., 2021a. Graph neural networks for laminar flow prediction around random two-dimensional shapes. Physics of

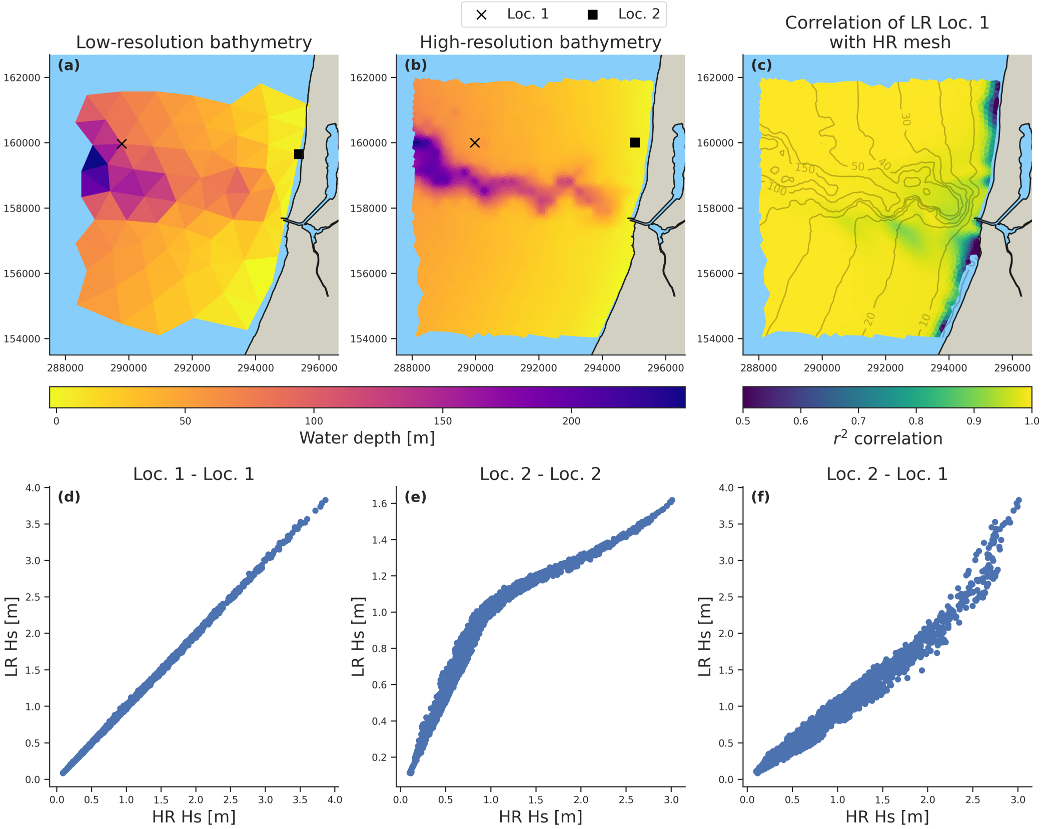

Figure 10: Panel (a) and (b): Low- and high-resolution mesh bathymetries of Region 2 with one off- and one nearshore location highlighted, that are used to calculate the correlations of significant wave height. Panel (c): A map of all the correlation coefficients r^2 of the low-resolution node loc. 1 with all high-resolution nodes. Bathymetry contour lines were added to aid interpretation. Triangles with invalid node values are not plotted, see main text in section 3.2. Panel (d)-(f): Examples of the correlations between different off- and nearshore locations. Note also the differences in the meshes between panel (b) and (c), which is due to the exclusion of NaN values when calculating the correlation map.

Fluids 33, 123607. doi:10.1063/5.0064108.

Chen, J., Pillai, A.C., Johanning, L., Ashton, I., 2021b. Using machine learning to derive spatial wave data: A case study for a marine energy site. Environmental Modelling and Software 142, 105066. doi:10.1016/j.envsoft.2021.105066.

Chen, M., Wang, L., Luo, Z., Xu, J., Zhang, B., Li, Y., Tan, A.C.C., 2023. Super-resolution reconstruction framework of wind turbine wake: Design and application. Ocean Engineering 288, 116099. doi:10.1016/j.oceaneng.2023.116099.

Cheng, X., Khomtchouk, B., Matloff, N., Mohanty, P., 2019. Polynomial Regression As an Alternative to Neural Nets. arXiv:1806.06850.

Choon, O.H., Hoong, L.C., Huey, T.S., 2008. A functional approximation comparison between neural networks and polynomial regression. WSEAS Transactions on Mathematics 7, 353–363.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems 2, 303–314. doi:10.1007/BF02551274.

de Wolff, T., Carrillo, H., Martí, L., Sanchez-Pi, N., 2021. Towards Optimally Weighted Physics-Informed Neural Networks in Ocean Modelling. arXiv:2106.08747.

Delpey, M., Lastiri, X., Abadie, S., Roeber, V., Maron, P., Liria, P., Mader, J., 2021. Characterization of the wave resource variability in the French Basque coastal area based on a high-resolution hindcast. Renewable Energy 178, 79–95. doi:10.1016/j.renene.2021.05.167.

Dempster, A.P., Schatzoff, M., Wermuth, N., 1977. A Simulation Study of Alternatives to Ordinary Least Squares. Journal of the American Statistical Association 72, 77–91. doi:10.1080/01621459.1977.10479910.

Dong, C., Loy, C.C., Tang, X., 2016. Accelerating the Super-Resolution Convolutional Neural Network, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham. volume 9906, pp. 391–407. doi:10.1007/978-3-319-46475-6_25.

Ducournau, A., Fablet, R., 2016. Deep learning for ocean remote sensing: An application of convolutional neural networks for super-resolution on satellite-derived SST data, in: 2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS), IEEE, Cancun, Mexico. pp. 1–6. doi:10.1109/PRRS.2016.7867019.

Falcon, W., Borovec, J., Wälchli, A., Eggert, N., Schock, J., Jordan, J., Skafte, N., Ir1dXD, Bereznyuk, V., Harris, E., Tullie Murrell, Yu, P., Præsius, S., Addair, T., Zhong, J., Lipin, D., Uchida, S., Shreyas Bapat, Schröter, H., Dayma, B., Karnachev, A., Akshay Kulkarni, Shunta Komatsu, Martin.B, Jean-Baptiste SCHIRATTI, Mary, H., Byrne, D., Cristobal Eyzaguirre, Cinjon, Bakhtin, A., 2020. PyTorchLightning/pytorch-lightning: 0.7.6 release. Zenodo. doi:10.5281/ZENODO.3828935.

Fey, M., Lenssen, J.E., 2019. Fast Graph Representation Learning with PyTorch Geometric. doi:10.48550/ARXIV.1903.02428, arXiv:1903.02428.

Fortunato, M., Pfaff, T., Wirnsberger, P., Pritzel, A., Battaglia, P., 2022. MultiScale MeshGraphNets. arXiv:2210.00612.

Fukami, K., Fukagata, K., Taira, K., 2019. Super-resolution reconstruction of turbulent flows with machine learning. Journal of Fluid Mechanics 870, 106–120. doi:10.1017/jfm.2019.238, arXiv:1811.11328.

Fukami, K., Fukagata, K., Taira, K., 2021. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. Journal of

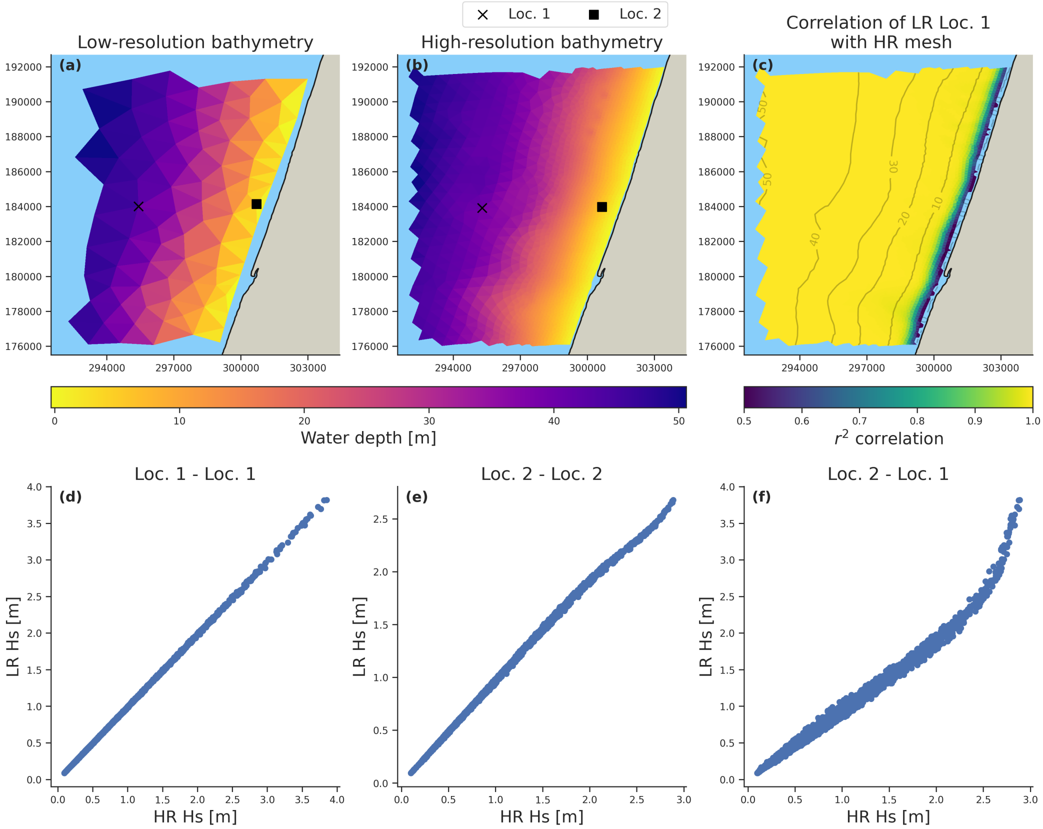

Figure 11: Panel (a) and (b): Low- and high-resolution mesh bathymetries of Region 3 with one off- and one nearshore location highlighted, that are used to calculate the correlations of significant wave height. Panel (c): A map of all the correlation coefficients r^2 of the low-resolution node loc. 1 with all high-resolution nodes. Bathymetry contour lines were added to aid interpretation. Triangles with invalid node values are not plotted, see main text in section 3.2. Panel (d)-(f): Examples of the correlations between different off- and nearshore locations. Note also the differences in the meshes between panel (b) and (c), which is due to the exclusion of NaN values when calculating the correlation map.

Fluid Mechanics 909, A9. doi:`10.1017/jfm.2020.948`.

Gao, H., Sun, L., Wang, J.X., 2021. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. Physics of Fluids 33, 073603. doi:`10.1063/5.0054312`.

Giladi, N., Ben-Haim, Z., Nevo, S., Matias, Y., Soudry, D., 2021. Physics-aware downsampling with deep learning for scalable flood modeling, in: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 1378–1389.

Gopinath, Deepthi.I., Dwarakish, G., 2015. Wave Prediction Using Neural Networks at New Mangalore Port along West Coast of India. Aquatic Procedia 4, 143–150. doi:`10.1016/j.aqpro.2015.02.020`.

Gorrell, L., Raubenheimer, B., Elgar, S., Guza, R., 2011. SWAN predictions of waves observed in shallow water onshore of complex bathymetry. Coastal Engineering 58, 510–516. doi:`10.1016/j.coastaleng.2011.01.013`.

Gupta, J.K., Brandstetter, J., 2022. Towards Multi-spatiotemporal-scale Generalized PDE Modeling. `arXiv:2209.15616`.

Huang, L., Jing, Y., Chen, H., Zhang, L., Liu, Y., 2022. A regional wind wave prediction surrogate model based on CNN deep learning network. Applied Ocean Research 126, 103287. doi:`10.1016/j.apor.2022.103287`.

James, S.C., Zhang, Y., O'Donncha, F., 2018. A machine learning framework to forecast wave conditions. Coastal Engineering 137, 1–10. doi:`10.1016/j.coastaleng.2018.03.004`, `arXiv:1709.08725`.

Jiang, W., Luo, J., 2022. Graph neural network for traffic forecasting: A survey. Expert Systems with Applications 207, 117921. doi:`10.1016/j.eswa.2022.117921`.

Jörges, C., Berkenbrink, C., Gottschalk, H., Stumpe, B., 2023. Spatial ocean wave height prediction with CNN mixed-data deep neural networks using random field simulated bathymetry. Ocean Engineering 271, 113699. doi:`10.1016/j.oceaneng.2023.113699`.

Keys, R., 1981. Cubic convolution interpolation for digital image processing. IEEE Transactions on Acoustics, Speech, and Signal Processing 29, 1153–1160. doi:`10.1109/TASSP.1981.1163711`.

Kuehn, J., Abadie, S., Liquet, B., Roeber, V., 2023. A deep learning super-resolution model to speed up computations of coastal sea states. Applied Ocean Research 141, 103776. doi:`10.1016/j.apor.2023.103776`.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., Battaglia, P., 2023. Learning skillful medium-range global weather forecasting. Science 0, eadi2336. doi:`10.1126/science.adi2336`.

Lastiri, X., Abadie, S., P., M., Delpey, M., Liria, P., Mader, J., Roeber, V., 2020. Wave Energy Assessment in the South Aquitaine Nearshore Zone from a 44-Year Hindcast. Journal of Marine Science and Engineering 8, 199. doi:`10.3390/jmse8030199`.

Lino, M., Fotiadis, S., Bharath, A.A., Cantwell, C.D., 2022. Multi-scale rotation-equivariant graph neural networks for unsteady Eulerian fluid dynamics. Physics of Fluids 34, 087110. doi:`10.1063/5.0097679`.

Lloyd, D.T., Abela, A., Farrugia, R.A., Galea, A., Valentino, G., 2022. Optically Enhanced Super-Resolution of Sea Surface Temperature Using Deep Learning. IEEE Transactions on Geoscience and Remote Sensing 60, 1–14. doi:`10.1109/TGRS.2021.3094117`.

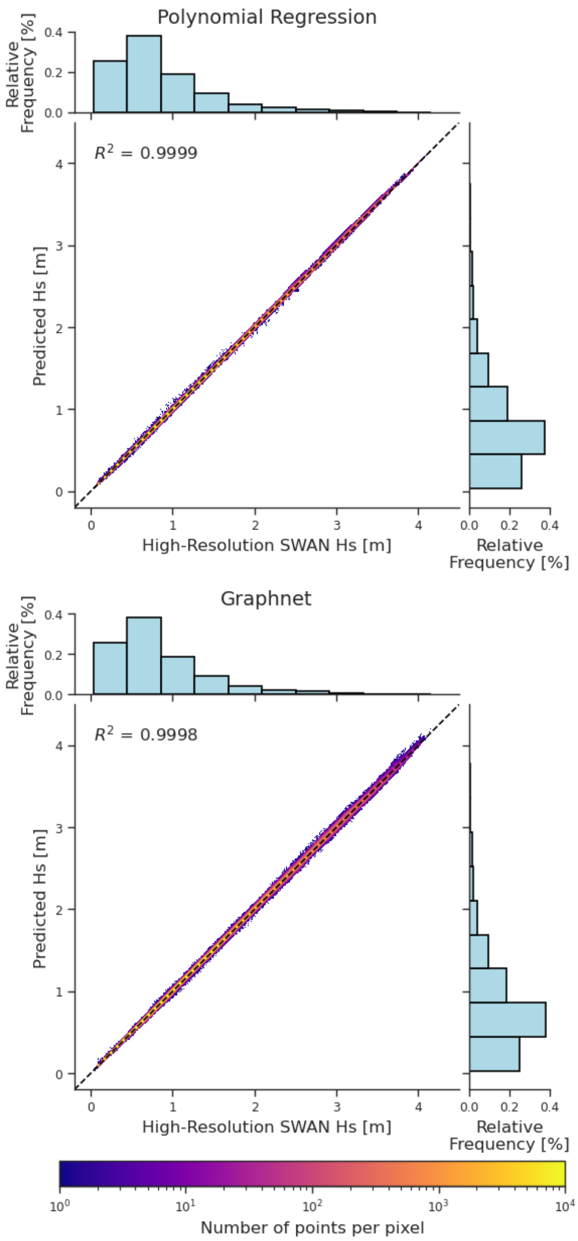

Figure 12: A scatter density plot of the model predictions for Region 2, where color indicates the number of points per pixel on a logarithmic scale. Additionally, histograms on both axes show the distribution of the significant wave heights. The upper figure refers to the predictions of the polynomial ridge regression, the lower to the GraphNet.

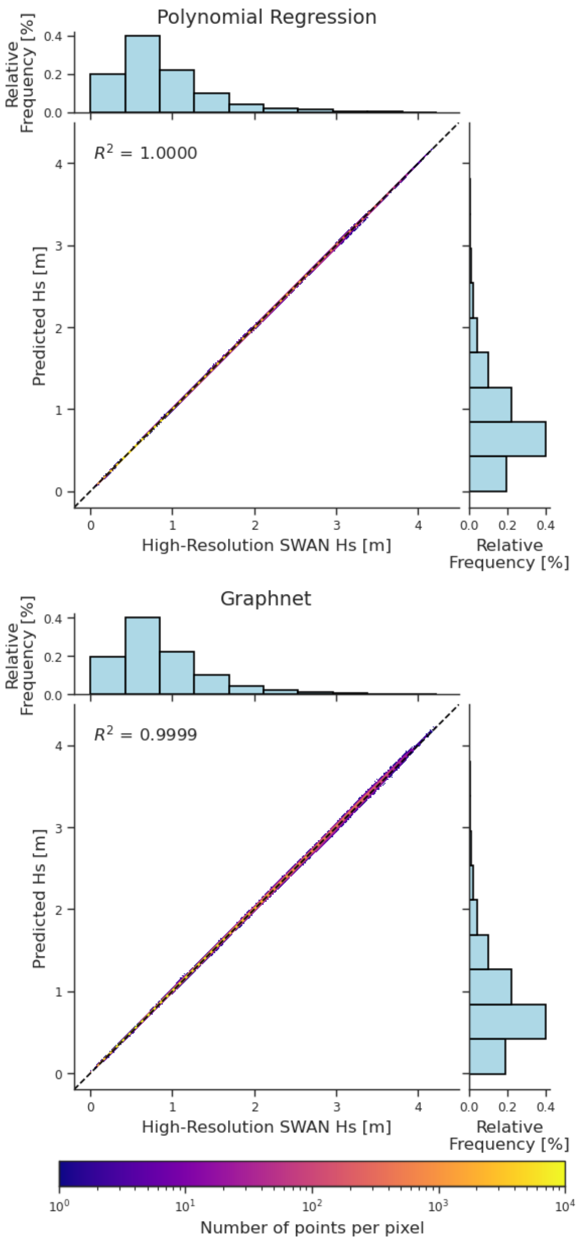

Figure 13: A scatter density plot of the model predictions for Region 3, where color indicates the number of points per pixel on a logarithmic scale. Additionally, histograms on both axes show the distribution of the significant wave heights. The upper figure refers to the predictions of the polynomial ridge regression, the lower to the GraphNet.

Londhe, S.N., Panchang, V., 2006. One-day wave forecasts based on artificial neural networks. Journal of Atmospheric and Oceanic Technology 23, 1593–1603. doi:10.1175/JTECH1932.1.

Londhe, S.N., Shah, S., Dixit, P.R., Nair, T.M., Sirisha, P., Jain, R., 2016. A coupled numerical and artificial neural network model for improving location specific wave forecast. Applied Ocean Research 59, 483–491. doi:10.1016/j.apor.2016.07.004.

Lopez-Radcenco, M., Fablet, R., Aissa-El-Bey, A., Ailliot, P., 2017. Locally-adapted convolution-based super-resolution of irregularly sampled ocean remote sensing data, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, Beijing. pp. 4307–4311. doi:10.1109/ICIP.2017.8297095.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. doi:10.48550/arXiv.1711.05101, arXiv:1711.05101.

Lucero, F., Stringari, C.E., Filipot, J.F., 2023. Improving WAVEWATCH III hindcasts with machine learning. Coastal Engineering 185, 104381. doi:10.1016/j.coastaleng.2023.104381.

Michel, M., Obakrim, S., Raillard, N., Ailliot, P., Monbet, V., 2022. Deep learning for statistical downscaling of sea states. Advances in Statistical Climatology, Meteorology and Oceanography 8, 83–95. doi:10.5194/ascmo-8-83-2022.

Obiols-Sales, O., Vishnu, A., Malaya, N.P., Chandramowlishwaran, A., 2021. SURFNet: Super-Resolution of Turbulent Flows with Transfer Learning using Small Datasets, in: 2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), IEEE, Atlanta, GA, USA. pp. 331–344. doi:10.1109/PACT52795.2021.00031.

Ostertagová, E., 2012. Modelling using Polynomial Regression. Procedia Engineering 48, 500–506. doi:10.1016/j.proeng.2012.09.545.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library, in: Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P.W., 2021. Learning Mesh-Based Simulation with Graph Networks, in: 9th International Conference on Learning Representations, ICLR 2021.

Piggott, M.D., Pain, C.C., Gorman, G.J., Marshall, D.P., Killworth, P.D., 2008. Unstructured adaptive meshes for ocean modeling, in: Hecht, M.W., Hasumi, H. (Eds.), Geophysical Monograph Series. American Geophysical Union, Washington, D. C.. volume 177, pp. 383–408. doi:10.1029/177GM22.

Prechelt, L., 1998. Early Stopping - But When?, in: Goos, G., Hartmanis, J., Van Leeuwen, J., Orr, G.B., Müller, K.R. (Eds.), Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, Berlin, Heidelberg. volume 1524, pp. 55–69. doi:10.1007/3-540-49430-8_3.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686–707. doi:10.1016/j.jcp.2018.10.045.

Ricondo, A., Cagigal, L., Pérez-Díaz, B., Méndez, F.J., 2024. HySwash: A hybrid model for nearshore wave processes. Ocean Engineering 291, 116419. doi:10.1016/j.oceaneng.2023.116419.

Roeber, V., Cheung, K.F., 2012. Boussinesq-type model for energetic breaking waves in fringing reef environments. Coastal Engineering 70, 1–20. doi:10.1016/j.coastaleng.2012.06.001.

Roelvink, D.J., Reniers, A., van Dongeren, A., Thiel de Vries, J., Lescinski, J., McCall, R., 2010. Xbeach model – description and manual .

Sanchez-Gonzalez, A., Heess, N., Springenberg, J.T., Merel, J., Riedmiller, M., Hadsell, R., Battaglia, P., 2018. Graph networks as learnable physics engines for inference and control, in: Dy, J., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, PMLR. pp. 4470–4479.

Schultz, M.G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L.H., Mozaffari, A., Stadtler, S., 2021. Can deep learning beat numerical weather prediction? Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 379, 20200097. doi:10.1098/rsta.2020.0097.

Shi, N., Xu, J., Wurster, S.W., Guo, H., Woodring, J., Van Roekel, L.P., Shen, H.W., 2022. GNN-Surrogate: A Hierarchical and Adaptive Graph Neural Network for Parameter Space Exploration of Unstructured-Mesh Ocean Simulations. arXiv:2202.08956.

Sonogashira, M., Shonai, M., Iiyama, M., 2020. High-resolution bathymetry by deep-learning-based image superresolution. PLOS ONE 15. doi:10.1371/journal.pone.0235487.

Stengel, K., Glaws, A., Hettinger, D., King, R.N., 2020. Adversarial super-resolution of climatological wind and solar data. Proceedings of the National Academy of Sciences 117, 16805–16815. doi:10.1073/pnas.1918964117.

Su, H., Wang, A., Zhang, T., Qin, T., Du, X., Yan, X.H., 2021. Super-resolution of subsurface temperature field from remote sensing observations based on machine learning. International Journal of Applied Earth Observation and Geoinformation 102, 102440. doi:10.1016/j.jag.2021.102440.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the AAAI Conference on Artificial Intelligence 31. doi:10.1609/aaai.v31i1.11231.

Timm, N.H., 2004. Multivariate Regression Models, in: Applied Multivariate Analysis. Springer New York, New York, NY, pp. 185–309. doi:10.1007/978-0-387-22771-9_4.

Tolman, H.L., 2009. User manual and system documentation of WAVEWATCH III TM version 3.14. Technical note, MMAB Contribution 276, 220.

Tran, D., Toulis, P., Airoldi, E.M., 2015. Stochastic gradient descent methods for estimation with large data sets. doi:10.48550/arXiv.1509.06459, arXiv:1509.06459.

Varing, A., Filipot, J.F., Delpey, M., Guitton, G., Collard, F., Platzer, P., Roeber, V., Morichon, D., 2021. Spatial distribution of wave energy over complex coastal bathymetries: Development of methodologies for comparing modeled wave fields with satellite observations. Coastal Engineering 169, 103793. doi:10.1016/j.coastaleng.2020.103793.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17, 261–272. doi:10.1038/s41592-019-0686-2.

Wang, N., Chen, Q., Chen, Z., 2022. Reconstruction of nearshore wave fields based on physics-informed neural networks. Coastal Engineering 176, 104167. doi:10.1016/j.coastaleng.2022.104167.

Wang, Z., Chen, J., Hoi, S.C.H., 2021. Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 3365–3387. doi:10.1109/TPAMI.2020.2982166.

Xia, P., Tahara, T., Kakue, T., Awatsuji, Y., Nishio, K., Ura, S., Kubota, T., Matoba, O., 2013. Performance comparison of bilinear interpolation, bicubic interpolation, and B-spline interpolation in parallel phase-shifting digital holography. Optical Review 20, 193–197. doi:10.1007/s10043-013-0033-2.

Xie, Y., Franz, E., Chu, M., Thuerey, N., 2018. tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. ACM Transactions on Graphics 37, 1–15. doi:10.1145/3197517.3201304.

Xu, W., Grande Gutierrez, N., McComb, C., 2023. MegaFlow2D: A Parametric Dataset for Machine Learning Super-resolution in Computational Fluid Dynamics Simulations, in: Proceedings of Cyber-Physical Systems and Internet of Things Week 2023, ACM, San Antonio TX USA. pp. 100–104. doi:10.1145/3576914.3587552.

Yutani, T., Yono, O., Kuwatani, T., Matsuoka, D., Kaneko, J., Hidaka, M., Kasaya, T., Kido, Y., Ishikawa, Y., Ueki, T., Kikawa, E., 2022. Super-Resolution and Feature Extraction for Ocean Bathymetric Maps Using Sparse Coding. Sensors 22, 3198. doi:10.3390/s22093198.

Zhang, X., Dai, H., 2019. Significant wave height prediction with the CRBM-DBN model. Journal of Atmospheric and Oceanic Technology 36, 333–351. doi:10.1175/JTECH-D-18-0141.1.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open 1, 57–81. doi:10.1016/j.aiopen.2021.01.001.

Zhu, X., Wu, K., Huang, W., 2023. Deep learning approach for downscaling of significant wave height data from wave models. Ocean Modelling 185, 102257. doi:10.1016/j.ocemod.2023.102257.