Dear editor, dear reviewer.

Before we proceed with point-by-point addressing of the issues raised by the reviewer, we would like to make a very short synopsis of what was done during this revision.

1. Based on the recommendation of Referee #2, we conducted experiments to test the failure of regional tide gauges. In light of this, we propose changing the title of the manuscript by removing the word “robust” and replacing it with a more descriptive phrase “in the presence of tide gauge sensor failures,” so the new title reads: “HIDRA3: deep-learning model for multi-point ensemble sea level forecasting in the presence of tide gauge sensor failures.”
2. We have added new sections to the manuscript: 2.4 “*Summary of differences to HIDRA2,*” 3.1 “*NEMO model description*," 3.3 “*Uncertainty estimation analysis*,” 3.4.2 “*Extreme scenario of a regional tide gauge failure”* and 3.7 “*HIDRA3 limitations”*.
3. We have provided additional explanations and figures in the architecture description sections, rewriting Section 2.2.3, "*Feature fusion module*," and Section 2.2.4, "*SSH regression module*."
4. We have included explanations for the data quality check procedures and computation of the tidal signal, expanded the analysis of sea surface temperature and waves' impact on the performance of the model, and corrected some typos.

We proceed to a detailed point-by-point response below.

COMMENT 1:

Review of HIDRA3: a robust deep-learning model for multi-point ensemble sea level forecasting.

The paper presents a new version of the HIDRA sea level forecasting model. HIDRA3 is a machine learning model with a deep convolutional architecture. The most important update from version 2 is that the current version uses data not just from the local tide gauge it predicts, but also from neighbouring tide gauges, which allows prediction also when the local tide gauge is not operational.

The paper is well written, figures are nice and the model architecture and modelling choices are well described. I recommend publications after some minor revisions.

RESPONSE 1:

We thank the reviewer for their encouraging comments.

COMMENT 2:

General points:

1) The manuscript makes the point that their machine learning model outperforms the numerical ocean model NEMO on SSH prediction. But that is not really what is tested. They do get better results on most metrics than what is seen in the specific NEMO run they compare with. However, the performance in that specific NEMO run, says almost nothing about the capabilities of numerical ocean models in general or even of the NEMO modelling systems capabilities. The SSH performance in the NEMO run they compare with depends on modelling choices (resolution, parametrisations, coordinate systems used. etc.) of which NEMO has very many and of course also on the forcing used to run the NEMO model.

Especially on the forcing side the HIDRA model has a great advantage in this comparison as it is allowed to use tide gauge data, whereas as I understand it the NEMO run they compare with does not assimilate sea level data. I would expect, although I don't know, that HIDRA3 without tide gauge data as inputs would perform worse that the specific NEMO run. Anyway, it should be made more clear in the text that although they outperform this specific Copernicus product it does not really imply much about the capabilities of numerical ocean models in general.

RESPONSE 2:

Thank you for pointing this out. We now provide a more detailed description of the specific CMEMS Copernicus NEMO model setup in Section 3.1 and include a reference. We would like to clarify that the version of NEMO used in this paper incorporates sea-level data assimilation for satellite altimetry (SLA), but not for tide gauges. However, tide gauge measurements are used to correct the bias. This is discussed further in Response 6. Below is the new Section 3.1, which describes the version of NEMO used in this study:

3.1 NEMO model description

We compare HIDRA3 with the state-of-the-art deep model HIDRA2 (Rus et al., 2023) and with the standard Copernicus Marine Environment Monitoring Service (CMEMS) product MEDSEA_ANALYSISFORECAST_PHY_006_013 (Clementi et al., 2021) numerical model Nucleus for European Modelling of the Ocean (NEMO) v4.2 (Madec, 2016). The Mediterranean Sea Physical Analysis and Forecasting model (MEDSEA_ANALYSISFORECAST_PHY_006_013) operates on a regular grid with a horizontal resolution of 1/24° (approximately 4 km) and 141 vertical levels. It uses a staggered Arakawa C-grid with masking for land areas, and a z* vertical coordinate system with partial cells to accurately represent the model topography. The model incorporates tidal forcing using eight components (M2, S2, N2, K2, K1, O1, P1, Q1) and is forced at its boundaries by the Global analysis and forecast product (GLOBAL_ANALYSISFORECAST_PHY_001_024) on the Atlantic side, while in the Dardanelles Strait, it uses a combination of daily climatological fields from a Marmara Sea model and the global analysis product. Atmospheric forcing comes from ECMWF forecasting product. Initial conditions are taken from the World Ocean Atlas (WOA) 2013 V2 winter climatology as of January 1, 2015. Data assimilation is performed using the OceanVar (3DVAR) scheme, which integrates in-situ vertical profiles of temperature and salinity from ARGO, Glider, and XBT, as well as Sea Level Anomaly (SLA) data from multiple satellites (including Jason 2 & 3, Saral-Altika, Cryosat, Sentinel-3A/3B, Sentinel6A, and HY-2A/B). The hydrodynamic part of the model is coupled to the wave model WaveWatch-III. Further information about the model is available in Clementi et al. (2021).

Since NEMO predicts the full ocean state, including SSH, on a regular grid, the respective tide gauge locations are approximated by the nearest-neighbor locations in the grid. One important thing to note is that ocean models like NEMO calculate sea surface height as a local departure from the geoid in the computational cell. A typical cell dimension is of the order of hundreds of meters. This means that the model's SSH represents a departure from the geoid, averaged over hundreds of squared meters, and is not directly relatable to the in-situ observations from local tide gauges, which measure local water depth. Therefore, to align NEMO forecasts with local tide gauge water depth, an offset adjustment of the initial 12-hour forecast is necessary to ensure zero bias compared to the respective tide gauge, as explained in Rus et al. (2023).

COMMENT 3:

2) The uncertainty quantifications and it's capabilities should be elaborated on more in the manuscript.

RESPONSE 3:

We agree with the reviewer. To address this concern, we add a new Section 3.3, *"Uncertainty Estimation Analysis"*, to discuss and analyze HIDRA3's uncertainty prediction capabilities:

3.3 Uncertainty estimation analysis

Besides the forecasted SSH values, HIDRA3 also estimates uncertainties by predicting the standard deviation for each forecasted value. To assess the reliability of the estimated standard deviations, we employ the scaled error metric

$$\epsilon_{i,t} = \frac{\boldsymbol{\mu}_{i,t} - \hat{\boldsymbol{\mu}}_{i,t}}{\hat{\boldsymbol{\sigma}}_{i,t}}, \tag{5}$$

as proposed by Barth et al. (2020). This metric quantifies the difference between the ground truth SSH value $\boldsymbol{\mu}_{i,t}$ and the predicted SSH value $\hat{\boldsymbol{\mu}}_{i,t}$, scaled by the estimated standard deviation $\hat{\boldsymbol{\sigma}}_{i,t}$, where i is the tide gauge index and t is the prediction lead time. To assess the accuracy of our predicted uncertainty, we calculate the mean μ_ϵ and the standard deviation σ_ϵ of the scaled error. If the predicted and the observed error distributions align, the standard deviation of the scaled error should be $\sigma_\epsilon = 1.0$.

For the second half of 2019 data, which was not used in training, the standard deviation of the scaled error for HIDRA3 is $\sigma_\epsilon = 0.98$. This indicates that HIDRA3 has good uncertainty prediction capabilities, with a slight overestimation of standard deviations of the prediction errors. To further analyze the distribution of the scaled error metric $\epsilon_{i,t}$, we plot its histogram in Fig. 13, along with the ideal model (a zero-centered unit-sigma Gaussian), and with the Gaussian distribution characterized by the estimated mean μ_ϵ and standard deviation σ_ϵ of HIDRA3. Note that the estimated Gaussian distribution aligns well

with the distribution of the ideal uncertainty prediction model, suggesting that HIDRA3 has excellent uncertainty prediction capabilities.

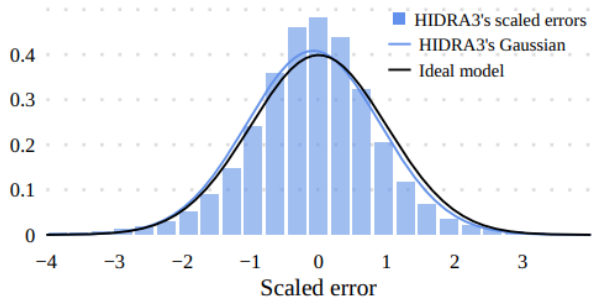

Figure 13. Histogram of the scaled error $\epsilon_{i,t}$, overlaid with ideal Gaussian distribution with mean 0 and variance 1, and a Gaussian distribution characterized by the estimated mean μ_ϵ and standard deviation σ_ϵ from HIDRA3. The estimated Gaussian distribution aligns well with the ideal one, suggesting excellent uncertainty prediction capabilities of HIDRA3.

COMMENT 4:

3) The models architecture is well described, but the reasons for the modelling choices made is not. Perhaps, much of this information is available in earlier HIDRA papers, but I would like to see more motivations for the different modelling choices.

RESPONSE 4:

We thank the reviewer for their suggestion to provide more explanations for our modelling choices. We include additional explanations and insights in Section 2.2.1, "*Geophysical encoder module*," and Section 2.2.2, "*Feature extraction module*," and revised Section 2.2.3, "*Feature fusion module*," and Section 2.2.4, "*SSH regression module*." Below, we include a latexdiff of Section 2.2.1 and 2.2.2, and the new Section 2.2.3 and Section 2.2.4.

2.2.1 Geophysical encoder module

The Geophysical encoder takes past and future, i.e. -72:72 h in hourly steps, of geophysical variables as the input: wind ($\mathbf{v} = (u_{10}, v_{10}) \in \mathbb{R}^{2\times W\times H}$), pressure ($\mathbf{p} \in \mathbb{R}^{W\times H}$), sea surface temperature ($\mathbf{T} \in \mathbb{R}^{W\times H}$), and waves ($\mathbf{w} \in \mathbb{R}^{4\times W\times H}$), where $W = 9$ and $H = 12$ are the spatial dimensions of the resampled input fields. The waves tensor $\mathbf{w}$ is composed of the following four components (hence the dimension $4 \times W \times H$): the first two are sine and cosine encodings of the mean wave direction, while the remaining two components are the mean wave period and significant height of combined wind waves and swell.

The input variables are processed in two steps. In the first step, variables (wind, pressure, SST and waves) are encoded separately, and then in the second step, encodings from different variables are fused together. We model each variable independently in the first step to avoid modeling low-level interactions between different variables, which would increase the number of parameters. The first step is shown in Fig. 5. The encoder processes variable of size $c \times 144 \times 9 \times 12$ by a 3D convolution block with 64 $2 \times 3 \times 3$ kernels with stride $2 \times 2 \times 2$ (the first dimension is temporal, while the second two are spatial), followed by

a ReLU, a 3D dropout (Tompson et al., 2014) and another 3D convolution with 512 $2 \times 4 \times 5$ kernels with stride $2 \times 1 \times 1$. Note that the strides and the kernel sizes are adjusted to reduce the dimensions to $512 \times 36 \times 1 \times 1$. This adjustment aims to compress the information into a low number of features, which helps reduce the risk of overfitting in the network and limits the number of output features of the Geophysical encoder. For the sea surface temperature and wave variables, the number of kernels in the last convolutional layer is 64 instead of 512, forming the output of size $64 \times 36 \times 1 \times 1$. This modification has led to a marginal enhancement in performance, presumably due to the larger number of features representing wind and pressure compared to SST and waves, thereby assigning greater significance to the former variables, which hold more relevance in the context of SSH forecasting.

~~In~~ Encodings from all variables are concatenated, resulting in a total of $2 \cdot 512 + 2 \cdot 64 = 1152$ channels. By removing spatial dimensions of size 1, concatenation is of size 1152×36, which is then in the second step of the Geophysical encoder (see Fig. 6) ~~, encodings from all variables are concatenated and fused~~ processed by a 1D convolution with 256 kernels of temporal size 5. We have chosen the kernel size of 5 ~~and~~ to increase the receptive field of the layer, which, with this setup, covers approximately 20 hours. We have found this to be effective in preliminary studies (not shown here). Next, two 1D convolutions with 256 kernels of temporal size ~~1. This is~~ 1 are applied and followed by a SELU activation (Klambauer et al., 2017), 1D dropout and a residual connection (see Fig. 6). These layers enable the extraction of more complex features from the geophysical variables. After flattening, the output is a single vector of 8192 geophysical features.

2.2.2 Feature extraction module

~~In~~ At prediction time, some tide gauges may not have measurements available, e.g., due to temporary sensor failures. Therefore, in the Feature extraction module~~(see Fig. 7),~~ , the location-specific features ~~$\mathbf{x}_i$~~ $\mathbf{x}_i$ are extracted independently for each location. Then, in the Feature fusion module, the features from all stations are merged, taking into account that the data at some locations is unavailable.

As is visualized in Fig. 7, The SSH observations (72 values) and the astronomic tide (144 values) are concatenated and encoded with a dense layer (i.e. fully-connected layer) with 512 output channels. Geophysical features are also passed through a dense layer with 512 output channels, whose task is to extract the features relevant to the specific tide gauge. Outputs of both layers are concatenated and processed with four dense layers with 1024 output channels. A SELU, a dropout and residual connections are applied as shown in Fig. 7~~, yielding the enriched SSH latent representation of size 1024.~~ . The skip-connection-based approach is chosen for its excellent feature construction capability, which proceeds by adding non-linearly transformed input features as residuals to the original features. This, in practice, leads to better training behavior due to a stronger gradient in the backpropagation phase (He et al., 2016). We chose SELU (Klambauer et al., 2017) for its smoother gradient propagation properties compared to its RELU counterpart, while the dropout is employed as a classical technique to curb overfitting.

Below are rewritten sections (not latexdiff):

2.2.3 Feature fusion module

As indicated in Fig. 4, the feature fusion module combines the location-specific features $\mathbf{x}_i \in \mathbb{R}^{1024}$, into a joint state vector $\mathbf{s} \in \mathbb{R}^{8192}$. A critical design requirement for the module is robustness to missing data; specifically, the number of location-specific features $\mathbf{x}_i$ may vary, as they are only available for the tide gauges with available measurements.

A partial reconstruction $\mathbf{s}_i$ of the state is computed from each location-specific feature vector $\mathbf{x}_i$ by applying a location-specific dense layer (see Fig. 8). In addition, a weight vector $\mathbf{w}_i$ is computed by applying another location-specific dense layer to $\mathbf{x}_i$. Each coordinate in $\mathbf{w}_i$ reflects the extent by which the particular location contributes to the respective coordinate in the joint state vector. If some tide gauges are non-descriptive, their weights would be reduced during training, lowering their influence on the final state vector $\mathbf{s}$, which is thus computed as

$$\mathbf{s} = \sum_{i \in V} \hat{\mathbf{w}}_i \odot \mathbf{s}_i, \tag{1}$$

where $\odot$ denotes element-wise array multiplication, $\hat{\mathbf{w}}_i$ are the coordinate-normalized weight vectors and V contains indices of tide gauges with available SSH measurements, so that $\mathbf{s}$ is approximated from tide gauges with available measurements. The components $\hat{w}_{i,j}$ are computed by a softmax, i.e.,

$$\hat{w}_{i,j} = \frac{e^{w_{i,j}}}{\sum_{k \in V} e^{w_{k,j}}}, \tag{2}$$

where the softmax ensures that the coordinate-wise weights sum to one across all locations with available measurements.

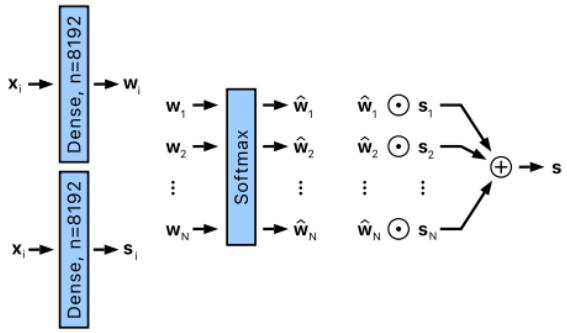

Figure 8. The Feature Fusion Module takes features $\mathbf{x}_i$ from location i and uses them to generate weights $\mathbf{w}_i$ and a partial reconstruction of state $\mathbf{s}_i$. These weights and partial reconstructions are combined into a joint state vector $\mathbf{s}$ using weighting and aggregation mechanisms. Locations without available measurements are excluded from the softmax calculation and aggregation. The parameters of the dense layers are specific to each location. The element-wise multiplication is denoted by the symbol $\odot$.

2.2.4 SSH regression module

The i-th SSH regression module receives the joint state vector $\mathbf{s}$. But since some stations may be deprioritized in the state vector by the Feature fusion module, i.e., due to a smaller amount of training data available for that station, the state vector is concatenated with a station-specific feature vector $\hat{\mathbf{x}}_i$. The concatenated feature vector is then processed by a dense layer with 72 output features to regress the 72 hourly SSH predictions $\hat{\boldsymbol{\mu}}_i$.

Note that two situations emerge in the construction of the station-specific feature vector $\hat{\mathbf{x}}_i$, depending on the sensor data availability. If the measurements are available for the i-th tide-gauge, the station-specific feature vector is simply obtained by

passing the features computed for the respective tide-gauge $\mathbf{x}_i$ (Sect. 2.2.2) through a dense layer with 1024 output features. This nonlinear transformation is applied to enable the network to emphasize the part of the feature vector possibly under-represented in the joint state vector. However, if the measurements are unavailable, e.g. due to sensor failure, then station-specific feature vector is approximated by transforming the joint state vector, i.e. passing it through a dense layer with 1024 output features (see Fig. 9).

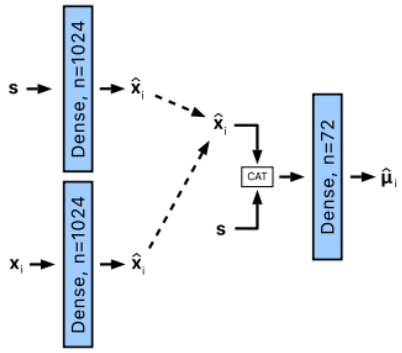

Figure 9. The SSH regression module for a location i receives joint state vector $\mathbf{s}$ and location-specific features $\mathbf{x}_i$. The features $\mathbf{x}_i$ are processed by a dense layer to produce the features $\hat{\mathbf{x}}_i$. In cases where measurements from the tide gauge at location i are unavailable, and therefore $\mathbf{x}_i$ is undefined, $\hat{\mathbf{x}}_i$ is approximated from $\mathbf{s}$ using a separate dense layer. The features $\hat{\mathbf{x}}_i$ and $\mathbf{s}$ are then concatenated and passed through a final dense layer to obtain SSH predictions, denoted as $\hat{\boldsymbol{\mu}}_i$.

COMMENT 5:

Specific comments:

L13 I think standard numerical model NEMO is the wrong label.

RESPONSE 5:

We agree. We changed the label to be more specific into “the Mediterranean basin NEMO setup of the Copernicus CMEMS service.”

COMMENT 6:

L44 Goes back to general point 1), these comparisons are very much of apples and oranges

RESPONSE 6:

The reviewer is raising an interesting point. We agree that it is often difficult to compare models between themselves or at least to attribute which level of performance corresponds to which algorithmic aspect of the model. In this sense, comparing ROMS to NEMO using different lateral boundary conditions, different atmospheric forcing and different parameterization schemes is no less difficult than comparing NEMO to HIDRA. We are unfortunately not in a position to feed the exact same atmospheric input and the exact same tide gauge input into NEMO (which does not ingest tide gauges as HIDRA3 but does assimilate satellite SLA) and into HIDRA3 (which does not ingest SLA as NEMO but receives tide gauges), therefore any comparison needs to keep in mind that the errors of both numerical approaches are accumulated across all input sources and the models themselves. Both NEMO and HIDRA3 are at the end of the modeling chain so their performance also reflects the accuracy of their input data.

We nevertheless feel that these comparisons can serve a purpose to establish the bottom line - *which of the models at our disposal gives the best forecast for civil rescue* and other emergency responses. In this optics, the comparisons presented in the paper are simply comparisons between the best sea level prediction setups at our disposal. We agree that these setups may be structurally quite heterogeneous but they all answer the same key question: what is the evolution of sea level in the next 72 hours? For the civil rescue response, coastal safety and the economy, this is a key issue.

We hope the reviewer agrees that the comparisons between admittedly heterogeneous modeling setups nevertheless hold some valuable information for the downstream services.

COMMENT 7:

L193 The NEMO setup has to be better described. What NEMO version is used? What forcing is used (also temporal and spatial resolution). What is the vertical and horizontal resolution of the model. What vertical coordinate system is used? Does it have a wave model? Does it have data assimilation? Does it have a minimum depth? Information of that kind is needed to give more context to the different comparisons.

RESPONSE 7:

We agree and have addressed this issue in our Response 2.