DEEP VISION-BASED FRAMEWORK FOR COASTAL FLOOD PREDICTION UNDER CLIMATE CHANGE IMPACTS AND SHORELINE ADAPTATIONS

Areg Karapetyan **Aaron C.H. Chow** **Samer Madanat**
Division of Engineering
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
{areg.karapetyan,cc6307,samer.madanat}@nyu.edu

ABSTRACT

Climate change and its byproducts, such as sea-level rise (SLR), pose escalating threats to coastal regions, intensifying the need for efficient and accurate flood prediction models. While traditional physics-based hydrodynamic simulators offer high fidelity, they are computationally prohibitive for integration into planning or optimization workflows. On the other hand, Deep Learning (DL) techniques can produce substantially faster models, yet, their development typically requires large number of training samples. To remove this barrier, we present a systematic framework for training *high-fidelity* and *high-resolution* coastal flood prediction models in *low-data settings*. We test the proposed framework on multiple DL architectures, including a fully Transformer-based model and a Convolutional Neural Network (CNN) with additive attention gates. Additionally, we introduce a deep CNN architecture tailored specifically to the studied climate adaptation-aware coastal flood prediction problem. The model was designed with a particular focus on its compactness so as to cater to resource-constrained scenarios and accessibility aspects. The performance of the DL models developed through the proposed framework is validated against state-of-the-practice methods as well as traditional Machine Learning approaches. The results demonstrate substantial improvement in prediction quality ranging from 100% to 400% across key metrics. Lastly, we round up the contributions by providing a meticulously curated dataset of simulated inundation maps for the coast of Abu Dhabi, which can serve as a benchmark for evaluating future coastal flood prediction models. The complete source code of the proposed framework, including the trained models, data and the evaluation scripts, can be accessed at https://github.com/Arnukk/CASPIAN.

1 INTRODUCTION

The looming global warming and its byproducts, such as rising sea levels and more frequent and severe storm surges (Garner et al., 2017; Sweet et al., 2022), render coastal regions increasingly susceptible to floods, which can inflict massive humanitarian, ecological, and economic devastation. To safeguard the lives and livelihoods of their residents, coastal cities have been actively armoring their shorelines with engineered structures such as seawalls, levees, and storm barriers. Although beneficial for *local flood protection*, these coastal defense structures may alter the hydrodynamics along the coast, amplifying water levels outside of their perimeter and spreading inundations to other (otherwise unaffected) regions (Wang et al., 2018; Haigh et al., 2020; Hummel et al., 2021).

To this end, physics-based high-fidelity simulators, such as Delft3D (Deltares), can be employed to resolve the detailed hydrodynamics around seawalls under SLR. While these tools can simulate complex coastal processes with detailed accuracy, they are notoriously expensive in terms of computational time and resources. As a result, direct adoption of such simulators in iterative routines requiring extensive number of model realizations (e.g., optimal shoreline protection planning, sensitivity analysis) remains impractical. Data-driven approaches, commonly referred to as *surrogate models* or *metamodels* in the coastal engineering literature, have been proposed as promising

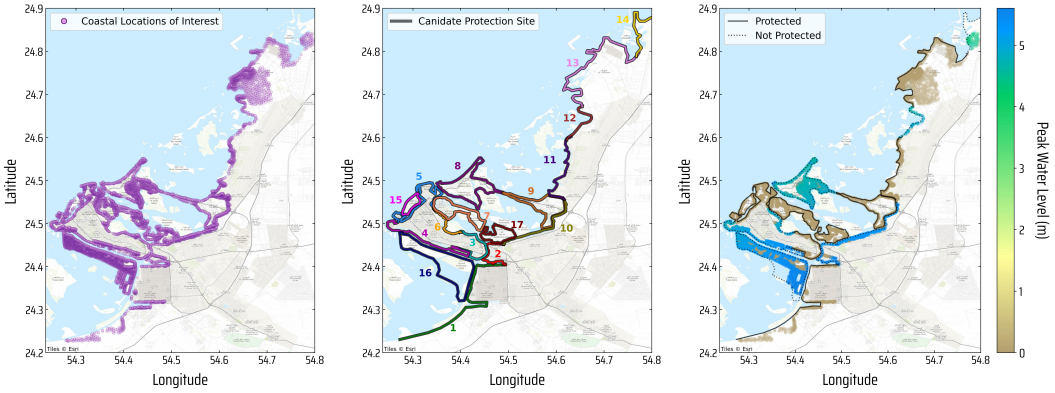
alternatives that can unlock substantial speed-up gains (Mosavi et al., 2018; Bentivoglio et al., 2022; Jones et al., 2023). A prominent issue in the design of coastal flooding metamodels concerns the *output dimensionality*. To allow for high-resolution predictions, most prior studies, such as (Jia et al., 2016; 2019; Kyprioti et al., 2022; El Garroussi et al., 2022; Rohmer et al., 2023; Kyprioti et al., 2021), resort to dimensionality reduction techniques, breaking down the problem into two consecutive steps. First, the original high-dimensional output is transformed into a low-dimensional latent representation, for example through PCA or an Autoeconder, then the surrogate model is trained to map the initial input space to the reduced latent space. An additional major challenge, specific to high-resolution surrogate models that account for future climate change-induced impacts (e.g., SLR), lies in the *scarcity of training data* since generation of annotated samples via physics-based high-fidelity simulators is both computationally expensive and time-intensive.

Despite the extensive development of coastal flood prediction models in the literature, limited attention has been paid to the incorporation of shoreline adaptations. To the best of our knowledge, with the exception of the work by (Jia et al., 2019) flood prediction has not been studied *under the joint consideration* of future climate change impacts and shoreline armoring scenarios. In (Jia et al., 2019), the authors employed the aforementioned common two-stage approach. The model, based on Kriging (a.k.a, Gaussian process regression) with PCA, was applied to the coast of the San Francisco Bay area and evaluated on a dataset of 40 scenarios simulated with Delft3D. Departing from this methodology, we approach the problem through the lens of computer vision, then develop end-to-end DL-based surrogate models that are scalable (w.r.t. coastal locations) and highly accurate. More concretely, our contributions are as follows:

- First, we propose a systematic approach for training efficient Deep Vision-based end-to-end coastal flood prediction models in low-data settings. We test the proposed methodology on different neural networks, including two existing vision models: a fully Transformer-based architecture (SWIN-Unet (Cao et al., 2022)) and a CNN with additive attention gates (Attention U-net (Oktay et al., 2018)).
- Next, we introduce a deep CNN architecture, dubbed Cascaded Pooling and Aggregation Network (CASPIAN), stylized explicitly for the climate adaptation-aware coastal flood prediction problem. On the current dataset, the performance of CASPIAN approached remarkably close to the results produced by the physics-based hydrodynamic simulator (on average, with 97% of predicted floodwater levels having less than 10 cm. error), effectively reducing the computational cost of producing a flood inundation map from *days to milliseconds*.
- Lastly, we round up the contributions by providing a database of synthetic flood depth maps of Abu Dhabi's coast under 174 different shoreline protection scenarios. The provided dataset, available at https://doi.org/10.7910/DVN/M9625R, to the best of our knowledge, is the first of its kind, and thus can serve as a benchmark for evaluating future coastal flooding metamodels.

2 Study Area and Hydrodynamic Model

The coastal area selected for the application of the proposed surrogate modeling framework stretches along the city of Abu Dhabi, which is the capital of the United Arab Emirates (UAE) situated inside the Persian Gulf. Given the complex structure of Abu Dhabi's shoreline, it was necessary to consider the protection of different sections. The considered partitioning scheme, illustrated in Fig.1B, was informed by the precincts defined in the 2030 Urban Structure Framework Plan of Abu Dhabi (Abu Dhabi Urban Planning Council, 2007). For simulating the tidal dynamics of the considered region, we utilized Delft3D, which is a hydrodynamic model that solves the time-dependent Reynolds Averaged Navier Stokes differential equations. To account for wind-induced wave activity in the vicinity of Abu Dhabi's coast, the validated Delft3D model was rerun with wind forcing from the ERA5 database, and the results were fed to an additional spectral wave model, Simulating Waves Nearshore (SWAN) (Delft University of Technology, 2022). Fig. 1C visualizes a flood depth map produced with the above coupled hydrodynamic model under a sample shoreline protection scenario and an SLR of 0.5 meters. Further details concerning the employed model and its validation can be found in (Chow and Sun, 2022) and Sec. A.



(A) (B) (C)

Figure 1: (A) Nearshore locations susceptible to flooding under a 0.5 m SLR scenario without any shoreline protection. (B) Candidate coastal segments considered for installation of seawalls. (C) Flood depth map of Abu Dhabi's coast produced with a physics-based high-fidelity hydrodynamic simulator under a sample shoreline fortification scenario and an SLR projection of 0.5 meters.

3 Problem Statement and Proposed Framework

In the studied climate adaptation-aware costal flood prediction problem, we seek to predict the maximum floodwater levels along the coast based on a given input *protection scenario*. To formalize, denote by $d_{\boldsymbol{x}}$ the number of candidate shoreline segments considered for fortification and let $x_i \in \{0, 1\}$ be the corresponding decision made for the segment i, with 1 indicating the placement of containments and 0 otherwise. Then, a protection scenario would be represented by a $d_{\boldsymbol{x}}$-dimensional binary vector $\boldsymbol{x}$ and the set of all possible protection scenarios ($2^{d_{\boldsymbol{x}}}$ in total) can be defined as $\mathcal{X} \triangleq \{\boldsymbol{x} \mid \boldsymbol{x} \in \{0, 1\}^{d_{\boldsymbol{x}}}\}$. Let $\boldsymbol{y}$ be a (non-negative) real-valued vector quantifying the peak water levels at $d_{\boldsymbol{y}}$ nearshore locations of interest. With this notation, the problem can be formulated as a regression task of learning a mapping function $f : \boldsymbol{x} \in \mathcal{X} \rightarrow \boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$ provided with a set $\{(\boldsymbol{x}^k, \boldsymbol{y}^k) \mid k \in [n], \boldsymbol{x}^k \in \mathcal{X}, \boldsymbol{y}^k \in \mathbb{R}^{d_{\boldsymbol{y}}}\}$ of n available training examples. Note that, for double-digit values of $d_{\boldsymbol{x}}$, the cardinality of the training set can turn disproportionately small compared to that of the input space (i.e., $n \ll 2^{d_{\boldsymbol{x}}}$), enforcing an *extremely low-resource learning setting*. The inference of f is further complicated by its output size $d_{\boldsymbol{y}}$, which is typically in the order of tens of thousands (here, $d_{\boldsymbol{y}} = 12066$).

The workflow of the proposed vision-based surrogate modeling framework, graphically summarized in Fig. 2, can be dissected into four parts: (1) Sampling of protection scenarios for training data generation with the adopted physics-based high-fidelity simulator; (2) Transformation of 1D input protection scenarios into image-like representations; (3) Extensive data augmentation by masking out randomly selected patches in the input images; (4) Neural network selection and training of the surrogate costal flooding model. Observe that by remodeling inputs and outputs, the proposed framework effectively transforms the initial regression problem into one of learning a mapping between image-like representations. From a computer vision viewpoint, this problem generally falls under the umbrella of image-to-image translation tasks (Isola et al., 2017), however, it can also be deemed as a variant of monocular depth estimation from a single image (Yang et al., 2024; Fu

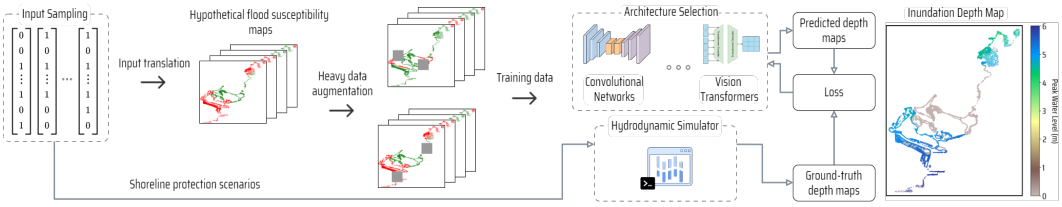


Figure 2: Schematic diagram of the proposed low-resource Deep Vision-based learning framework.

et al., 2024) since the predicted output is a depth map (of floodwaters). A more detailed and formal description of the framework is deferred to Sec. B.

3.1 CASPIAN

As previously noted, we evaluate the proposed framework on two existing vision models, SWIN-Unet and Attention U-net (which were originally designed for medical imaging tasks), thereby demonstrating the general applicability of the framework. Additionally, to cater to resource-constrained scenarios and accessibility aspects, we introduce CASPIAN, a minimalistic CNN model (illustrated in Fig. 3) which features only about 0.36 Mil. trainable parameters and is tailored explicitly for climate adaptation-aware coastal flood prediction. Due to space scarcity, the description of the model is presented in Sec. C.

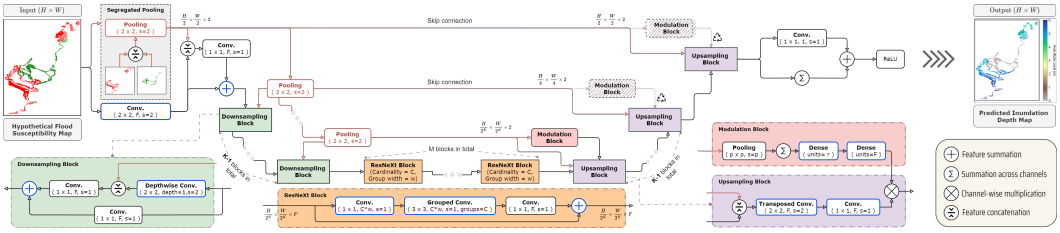


Figure 3: Detailed architecture of the proposed minimalistic CNN model, CASPIAN, for high-resolution coastal flood prediction under SLR and shoreline fortifications. The input image passes through two concurrent paths: a pooling path (colored in red) and a fully convolutional path. The modulation blocks drawn as sketches in dotted outlines are optional and could be substituted by the output from the initial block. The operations followed by non-linear activation functions are marked with a blue border.

4 RESULTS

Dataset: Following the proposed workflow, a total of 142 input protection scenarios were sampled, and the corresponding inundation maps were produced with the employed physics-based hydrodynamic model to construct the main dataset. To test the generalizability of the developed models, a Holdout dataset consisting of 32 handcrafted protection was additionally generated (see Sec. G).

Quantitative Results: To validate the performance of the DL-based coastal flood prediction models trained with the proposed framework, we benchmark them against two approaches proposed by most recent related works (Jia et al., 2019; Rohmer et al., 2023), namely Kriging with PCA and Support Vector Regression (SVR), as well as with two traditional ML techniques, namely Linear Regression and Lasso Regression with polynomial features (referred to as Lasso with Poly.). The models were assessed considering 6 different metrics, including both error and accuracy measures. The comparison results reveal significant gains in predictive performance, with improvements from the proposed DL-based models ranging from 100% to 400% across key metrics. Due to page limitations, details of the quantitative evaluation are reported in Sec. D.

Qualitative Results: For further scrutiny, we analyze the flood inundation maps produced by the candidate methods by categorizing and visualizing the percentage errors of predicted peak water levels as compared to ground truth values. Figure 4 presents the resulting error maps for three protection scenarios from the Holdout dataset with varying degrees of shoreline armoring: a highly protected case (the topmost row), a moderately protected case (the middle row) and mostly unprotected scenario (the bottom row). For clarity of illustration, the maps produced by the two best-performing DL models, CASPIAN and SWIN-Unet, and the top three benchmark methods, Kriging with PCA, Lasso with Poly., and SVR, are examined.

As can be deduced from Fig. 4, the flood depth maps produced by the proposed DL models accurately and closely aligned with those generated by the physics-based hydrodynamic simulator. This is reflected in the predominance of green areas in the error maps, indicating low relative errors ($\approx 0\%$ or $< 5\%$), and is consistent with the results of the quantitative evaluation. Importantly, the errors

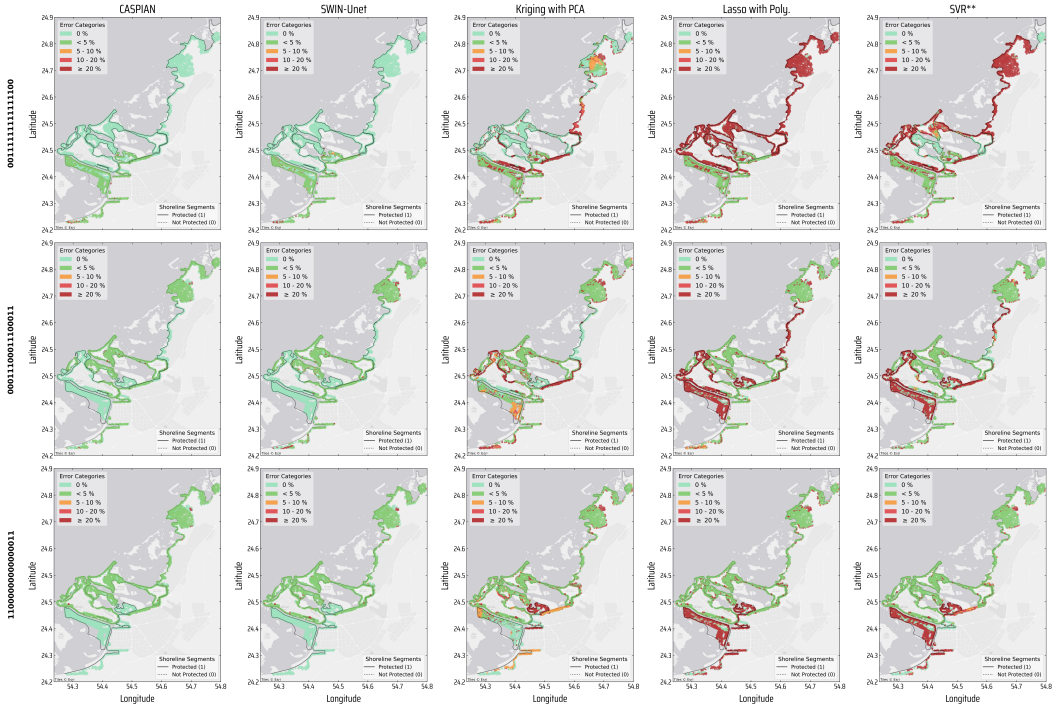


Figure 4: Qualitative performance comparison of the developed DL-based models (CASPIAN and SWIN-Unet) vs. state-of-the-practice approaches. Rows correspond to different protection scenarios, whereas columns to models. Green colors indicate $\leq 5\%$ percentage error from ground-truth values, while orange and red colors represent higher errors. The superscript ** denotes an ensemble of individual models, each trained for one specific coastal location.

in CASPIAN and SWIN-Unet predictions are sparse, spatially scattered, and appear to be isolated outliers rather than systematic deviations. The flood maps produced with these two DL models exhibited strong spatial consistency across all three protection scenarios. In contrast, the three benchmark models produced a large number of errors that are spatially clustered in concentrated regions, with such clusters appearing across multiple parts of the shoreline. This indicates poor generalization across protection scenarios and limited robustness in capturing spatial flood dynamics.

5 Concluding Remarks

In this paper, we presented a data-driven surrogate modeling framework for developing accurate and reliable coastal flooding metamodels powered by vision-based DL techniques. The proposed framework was tested on three different DL architectures, including a lightweight CNN model CASPIAN introduced in this work. The developed models were shown to significantly outperform existing geostatistical methods and standard regression techniques commonly employed in prior studies. The best-performing model, CASPIAN, closely and consistently emulated the results of the high-fidelity hydrodynamic simulator, on average achieving an AMAE error of 0.06 and $\delta > 0.5$ error of only around 1% on both Test and Holdout datasets.

One limitation of the developed coastal metamodels is that they are currently domain-specific since the training was performed considering one specific shoreline, SLR scenario, and fixed set of wind parameters. Without major modifications, one can extend the proposed framework to account for different coastal areas, for example, by including other geographical data such as local slopes, and hydraulic connectivities in the input maps. Another possible extension would be to expand the predictive scope and, in addition to peak water levels, also estimate the maximum velocities of floodwaters, a key prediction needed for coastal damage assessment.

REFERENCES

Abu Dhabi Urban Planning Council. Plan Abu Dhabi 2030: Urban Structure Framework Plan, 2007.

Azin Al Kajbaf and Michelle Bensi. Application of surrogate models in estimation of storm surge: A comparative assessment. *Applied Soft Computing*, 91:106184, 2020.

Authors Anonymous. Anonymous title. *Hydrology*, 2022.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

Patrick L Barnard, Maarten van Ormondt, Li H Erikson, Jodi Eshleman, Cheryl Hapke, Peter Ruggiero, Peter N Adams, and Amy C Foxgrover. Development of the coastal storm modeling system (cosmos) for predicting the impact of storms on high-energy, active-margin coasts. *Natural hazards*, 74:1095–1125, 2014.

Roberto Bentivoglio, Elvin Isufi, Sebastian Nicolaas Jonkman, and Riccardo Taormina. Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrology and Earth System Sciences*, 26(16):4345–4378, 8 2022. ISSN 16077938. doi: 10.5194/HESS-26-4345-2022.

Mohamed Amine Bouhlel, John T. Hwang, Nathalie Bartoli, Rëmi Lafage, Joseph Morlier, and Joaquim R. R. A. Martins. A python surrogate modeling framework with derivatives. *Advances in Engineering Software*, 2019.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022.

F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

Aaron CH Chow and Jiayun Sun. Combining sea level rise inundation impacts, tidal flooding and extreme wind events along the Abu Dhabi coastline. *Hydrology*, 9(8):143, 2022.

Delft University of Technology. Simulating waves nearshore (swan). https://swanmodel.sourceforge.io, 2022.

Deltares. Delft3d. https://oss.deltares.nl/web/delft3d.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Gary D Egbert and Svetlana Y Erofeeva. Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic technology*, 19(2):183–204, 2002.

Siham El Garroussi, Sophie Ricci, Matthias De Lozzo, Nicole Goutal, and Didier Lucor. Tackling random fields non-linearities with unsupervised clustering of polynomial chaos expansion in latent space: application to global sensitivity analysis of river flooding. *Stochastic Environmental Research and Risk Assessment*, 36(3):693–718, 3 2022. ISSN 14363259. doi: 10.1007/S00477-021-02060-7/FIGURES/11.

Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024.

Andra J. Garner, Michael E. Mann, Kerry A. Emanuel, Robert E. Kopp, Ning Lin, Richard B. Alley, Benjamin P. Horton, Robert M. DeConto, Jeffrey P. Donnelly, and David Pollard. Impact of climate change on New York City's coastal flood hazard: Increasing flood heights from the preindustrial to 2300 CE. *Proceedings of the National Academy of Sciences*, 114(45):11861–11866, 2017. doi: 10.1073/pnas.1703568114.

Ivan D. Haigh, Mark D. Pickering, J. A.Mattias Green, Brian K. Arbic, Arne Arns, Sönke Dangendorf, David F. Hill, Kevin Horsburgh, Tom Howard, Déborah Idier, David A. Jay, Leon Jänicke, Serena B. Lee, Malte Müller, Michael Schindelegger, Stefan A. Talke, Sophie Berenice Wilmes, and Philip L. Woodworth. The Tides They Are A-Changin': A Comprehensive Review of Past and Future Nonastronomical Changes in Tides, Their Driving Mechanisms, and Future Implications. *Reviews of Geophysics*, 58(1):e2018RG000636, 3 2020. ISSN 1944-9208. doi: 10.1029/2018RG000636.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. doi: 10.1109/CVPR.2018.00745.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

Michelle A. Hummel, Robert Griffin, Katie Arkema, and Anne D. Guerry. Economic evaluation of sea-level rise adaptation strongly influenced by hydrodynamic feedbacks. *Proceedings of the National Academy of Sciences of the United States of America*, 118(29):e2025961118, 7 2021. ISSN 10916490. doi: 10.1073/PNAS.2025961118/SUPPL{_}FILE/PNAS.2025961118.SAPP.PDF.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Gaofeng Jia, Alexandros A. Taflanidis, Norberto C. Nadal-Caraballo, Jeffrey A. Melby, Andrew B. Kennedy, and Jane M. Smith. Surrogate modeling for peak or time-dependent storm surge prediction over an extended coastal region using an existing database of synthetic storms. *Natural Hazards*, 81(2):909–938, 3 2016. ISSN 15730840. doi: 10.1007/S11069-015-2111-1/TABLES/7.

Gaofeng Jia, Ruo Qian Wang, and Mark T. Stacey. Investigation of impact of shoreline alteration on coastal hydrodynamics using Dimension REduced Surrogate based Sensitivity Analysis. *Advances in Water Resources*, 126:168–175, 4 2019. ISSN 0309-1708. doi: 10.1016/J.ADVWATRES.2019.03.001.

Anne Jones, Julian Kuehnert, Paolo Fraccaro, Ophélie Meuriot, Tatsuya Ishikawa, Blair Edwards, Nikola Stoyanov, Sekou L. Remy, Kommy Weldemariam, and Solomon Assefa. AI for climate impacts: applications in flood risk. *npj Climate and Atmospheric Science*, 6(1):63, Jun 2023.

Aikaterini P. Kyprioti, Alexandros A. Taflanidis, Norberto C. Nadal-Caraballo, and Madison Campbell. Storm hazard analysis over extended geospatial grids utilizing surrogate models. *Coastal Engineering*, 168:103855, 9 2021. ISSN 0378-3839. doi: 10.1016/J.COASTALENG.2021.103855.

Aikaterini P Kyprioti, Alexandros A Taflanidis, Norberto C Nadal-Caraballo, Madison C Yawn, and Luke A Aucoin. Integration of node classification in storm surge surrogate modeling. *Journal of Marine Science and Engineering*, 10(4):551, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

Amir Mosavi, Pinar Ozturk, and Kwok-wing Chau. Flood Prediction Using Machine Learning Models: Literature Review. *Water*, 10(11), 2018.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. In *Medical Imaging with Deep Learning*, 2018.

Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. https://github.com/keras-team/keras-tuner, 2019.

Art B Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7): 59–72, 2007.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeremy Rohmer, Charlie Sire, Sophie Lecacheux, Deborah Idier, and Rodrigo Pedreros. Improved metamodels for predicting high-dimensional outputs by accounting for the dependence structure of the latent variables: application to marine flooding. *Stochastic Environmental Research and Risk Assessment*, pages 1–23, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

Yingkai Sha. Keras-unet-collection. https://github.com/yingkaisha/keras-unet-collection, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

William V Sweet, Benjamin D Hamlington, Robert E Kopp, Christopher P Weaver, Patrick L Barnard, David Bekaert, William Brooks, Michael Craghan, Gregory Dusek, Thomas Frederikse, et al. *Global and regional sea level rise scenarios for the United States: Updated mean projections and extreme water level probabilities along US coastlines*. National Oceanic and Atmospheric Administration, 2022.

Ruo Qian Wang, Mark T. Stacey, Liv Muir Meltzner Herdman, Patrick L. Barnard, and Li Erikson. The Influence of Sea Level Rise on the Regional Interdependence of Coastal Infrastructure. *Earth's Future*, 6(5):677–688, 5 2018. ISSN 2328-4277. doi: 10.1002/2017EF000742.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017. doi: 10.1109/CVPR.2017.634.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data, 2024.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021. doi: 10.1109/JPROC.2020.3004555.

A Details of the Adopted Coupled Hydrodynamic Model

To allow for detailed and accurate modeling of coastal hydrodynamics of the selected area under SLR, storm events and shoreline fortifications, we adopt a coupled model that combines a Gulf-wide tidal model, a spectral wave model, and a wave run-up model. Fig. 5 provides a schematic of the inputs and outputs of the models' components.

For simulating the tidal dynamics of the considered region, we utilized Delft3D, which is a hydrodynamic model that solves the time-dependent Reynolds Averaged Navier Stokes differential equations. That is, Delft3D is a physics-based numerical model that considers the time-varying forces exerted on a water body (such as the entire Persian Gulf) due to hydrostatic pressures (such as SLR), tidal forcing, wind and storm stresses, bottom (seabed) friction, and river inflows over a finite-element computational grid (up to 30 m in horizontal resolution) spread over variable bathymetry. For any point in this grid, it can provide time series outputs, with 30-minute intervals, of water levels and local water circulation velocities throughout the specified simulation period. Importantly, Delft3D can handle computational grid cells that alternate between dry and wet states (Barnard et al., 2014).

The tidal model was validated by running the Delft3D over a simulated 3-month period between 1 January and 31 March 2017 (without wind forcing) and computing the root mean squared error between the model outputs of hourly water levels at 194 locations throughout the gulf and hourly tidal gauge water level data obtained from the TPXO8 Ocean Atlas for the same period (https://www.tpxo.net/global/tpxo8-atlas. Also see (Egbert and Erofeeva, 2002)). The model was calibrated by adjusting the bottom Manning's roughness coefficient for the entire gulf domain from 0.015 to 0.030. The lowest overall error was attained under the coefficient of 0.02, which was taken as the calibrated roughness value for the gulf model going forward. Fig. 6 demonstrates a typical fit between water level values (relative to the mean sea level) outputted by the model and the tidal gauge data at two representative locations near the UAE shore. Further details concerning the employed hydrodynamic model and its validation can be found in (Anonymous, 2022).

To account for wind-induced wave activity in the vicinity of Abu Dhabi's coast, the validated Delft3D model was rerun with wind forcing from the ERA5 database, and the results were fed to an additional spectral wave model, SWAN (Delft University of Technology, 2022), which allows

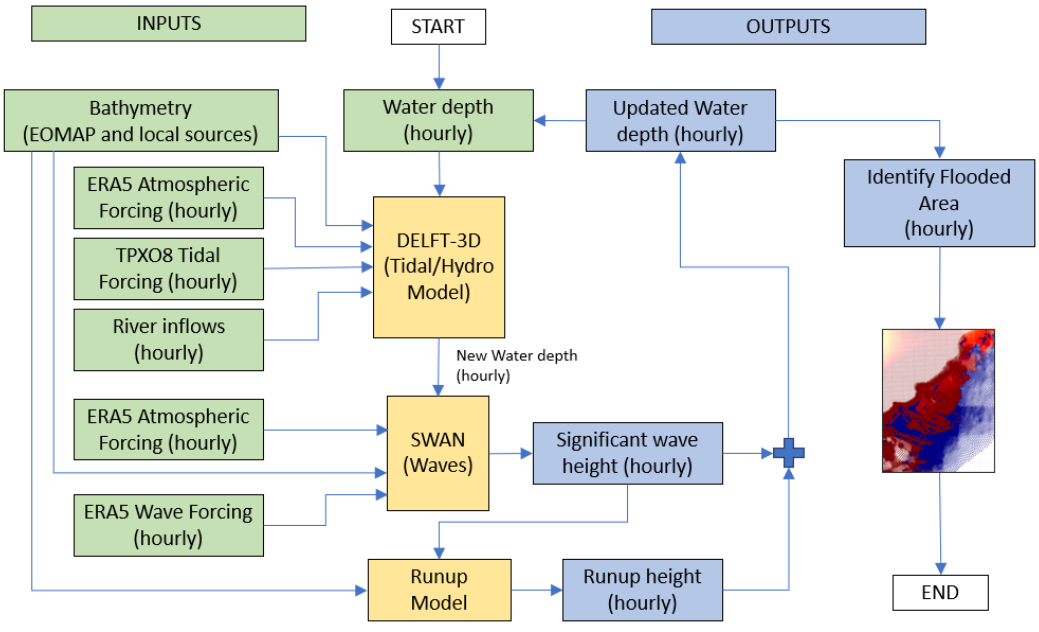


Figure 5: Schematic of hydrodynamic model elements. Green elements denote input parameters; Yellow are the constituent sub-models; Blue are model outputs. Running one cycle of the above model will generate an hourly update of the water depth throughout the UAE coastline.

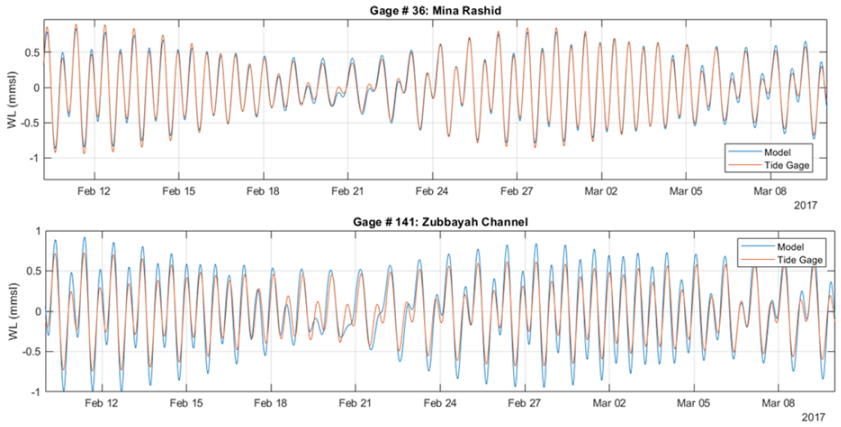


Figure 6: Comparison of the simulated tidal model's output and TPXO8 Atlas' time-series data on water levels from 10 Feb to 10 March 2017 at Mina Rashid (on top) and Zubbayah Channel (on bottom). The units of vertical axes are in meters relative to the mean sea level.

capturing wind-wave generation, wave diffraction, amplification and refraction of water surface waves as they approach the shoreline. The SWAN model was applied at a scale of about 100 km along the shoreline to about 50 km offshore under the same forcing from the ERA5 database. Finally, along the interface of the waves with the coastline, the SWAN-computed significant wave heights and the local shoreline slope were used to compute the run-up elevations along the coastline where the waves hit the shore.

With the above coupled hydrodynamic model in place, one can run a reference case with no shoreline armoring (except those already existing in Abu Dhabi) to evaluate the maximum extent of flooding due to SLR and storms. The placement of protective structures (e.g., seawalls) at candidate coastal sites was be realized by inserting "fixed weirs", which enact flow barriers at the corresponding locations of the domain. For every such configuration of containments, the raw output of the model will include 3 months worth of hourly water levels for more than $400,000$ grid point locations throughout the Persian Gulf. To filter the nearshore inland locations of interest that are susceptible to flooding, the following two steps were taken: (i) the points lying outside the urban region of Abu Dhabi were excluded; (ii) the inland cells that never experienced flooding even in the case of no coastal protection (i.e., are not hydraulically connected to the Gulf or bear no correlation with the input) were removed. This resulted in the final set of 12066 locations along the coastline, which appear in Fig. 1A. For each location, the peak water levels (i.e., the maximum value of water depth over the simulated timeframe of 3 months) under different protection scenarios were then extracted to construct the dataset for training coastal flood prediction models.

B Details of the Proposed Deep Visual Learning Framework

The workflow of the proposed vision-based surrogate modeling framework, graphically summarized in Fig. 2, can be dissected into four parts, of which first is the generation of training tuples $(\boldsymbol{x}^k, \boldsymbol{y}^k)$. It is crucial to ensure a sufficiently representative selection of points $(\boldsymbol{x}^k)_{k\in[n]}$ for which f will be evaluated, especially under the imposed low-data regime. The scheme adopted herein relies on a combination of judicious manual selection and random sampling. From the former category, we include the following base scenarios: full protection (i.e., $\boldsymbol{x} = \mathbf{1}$), protection of the first and second halves, no protection (i.e., $\boldsymbol{x} = \mathbf{0}$), protection of single precincts (i.e., all binary unit vectors in $\mathcal{X}$) and the inverses thereof, resulting in a total of $4 + 2d_{\boldsymbol{x}}$ input instances. The remaining (out of n) random cases are constructed by drawing uniformly distributed random points from a $d_{\boldsymbol{x}}$-dimensional unit cube via Latin Hypercube Sampling (McKay et al., 1979), then rounding their entries to the nearest integer value. For each selected input $\boldsymbol{x}^k$, the respective output $\boldsymbol{y}^k$ was computed by carrying out a numerical simulation with the coupled hydrodynamic model described in Sec. A.

Recall that every element of $\boldsymbol{y}$ corresponds to a specific geographical location parameterized by its latitude and longitude. In vectorial representation, however, this information is abstracted away,

leaving the potential of exploiting the spatial correlations and interdependencies between these locations untapped. To enrich the data representation, the proposed pipeline remodels the input and output vectors into matrices as follows. From each $\boldsymbol{y}^k, k \in [n]$, we construct a corresponding *flood inundation map* $\boldsymbol{Y}^k \in \mathbb{R}^{H \times W}$ through a mapping $\Phi : \mathbb{R}^2 \to (i, j), i \in [H], j \in [W]$ that converts the geographic coordinates associated with the components of $\boldsymbol{y}$ into grid indices (i, j). This transformation Φ and the grid size $H \times W$ should be selected such that the existing spatial relationships among the output locations are minimally distorted. For the current application site, the coordinate conversion was performed by discretizing the axes of the geographical domain. The dimensions of the formed regular mesh grid, which underlies $\boldsymbol{Y}^k$-s, were equated for ease of processing, and the grid size was set to $H \times W = 1024 \times 1024$ to sustain the desired fine geographic granularity of predictions at a reasonable computational cost while maintaining the overall spatial structure of output locations. The mapping conflicts due to discretization were resolved according to the nearest neighbor principle. Subsequently, the established indexing is leveraged to translate the binary protection scenarios $(\boldsymbol{x}^k)_{k \in [n]}$ into hypothetical *flood susceptibility maps* $(\boldsymbol{X}^k)_{k \in [n]}$, where each $\boldsymbol{X}^k \in \mathcal{C}^{H \times W}$ and $\mathcal{C}$ stands for some discrete set of three predefined values that represent categories. Here, the latter was defined as $\mathcal{C} \triangleq \{-1, 0, 1\}$ and for $\forall\, k \in [n]$, $X^k_{i,j}$ was assigned -1 if the shoreline segment in $\boldsymbol{x}^k$ closest to the location tied to the (i, j)-th index was marked as unprotected, 1 if protected and the rest of the cells were filled with zeros[1]. In a sense, $\boldsymbol{X}^k$-s are segmented matrices in which the $d_{\boldsymbol{y}}$ nearshore locations are classified by their distance to unprotected parts of the coast, and the proximity is perceived as a proxy indication of flood risk. It should, nevertheless, be noted that these input matrices may not necessarily reflect the actual risk or susceptibility of flooding but are, instead, conceptual constructs devised for modeling input protection scenarios, hence the terming "hypothetical".

Observe that with the remodeled input-output format, the initial regression model is effectively transformed into a problem of learning a mapping of the form $\boldsymbol{X} \in \mathcal{C}^{H \times W} \to \boldsymbol{Y} \in \mathbb{R}^{H \times W}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be visualized graphically as grayscale (i.e., single channel) images. From a computer vision viewpoint, this problem generally falls under the umbrella of image-to-image translation tasks (Isola et al., 2017), however, it can also be deemed as a variant of monocular depth estimation from a single image (Yang et al., 2024; Fu et al., 2024) since the predicted output is a depth map (of floodwaters). While both of these directions have been extensively researched, to the best of our knowledge, the present problem of inferring depth information from a grayscale, segmented image has not been explored.

Capitalizing on the new image-like representation of inputs and outputs, as a third step of the proposed framework, we artificially increase the volume of training data through image augmentation. Let $\mathcal{D} \triangleq \{(\boldsymbol{X}^k, \boldsymbol{Y}^k) \mid k \in [n]\}$ be the dataset constructed as prescribed above. From each existing pair $(\boldsymbol{X}^k, \boldsymbol{Y}^k)$ in $\mathcal{D}$, m new training examples $(\boldsymbol{X}^{k(1)}, \boldsymbol{Y}^k), \ldots, (\boldsymbol{X}^{k(m)}, \boldsymbol{Y}^k)$ are generated via the Cutout technique of (DeVries and Taylor, 2017), which applies a fixed-size zero-mask to a random location(s) within the input. This method can be applied in conjunction with other image augmentation techniques, such as rotation, flipping, or shifting, although for the current case study, the application of the former method alone (yet in an excessive manner) was found to be sufficient.

As the final part of the proposed pipeline, it remains to select the type of neural network that will power the surrogate model and the loss function it will learn to minimize. Now that the problem has been transformed into an image processing task, one has a powerful arsenal of DL techniques at disposal, including both generative models, such as GANs (e.g., pix2pix (Isola et al., 2017)) and Diffusion models (e.g., GeoWizard (Fu et al., 2024)), as well as discriminative models, such as Vision Transformers (e.g., SWIN Transformer (Liu et al., 2021)) and CNNs. Drawing on the success of the U-Net architecture (Ronneberger et al., 2015), we design (in Sec. 3.1) a minimalistic U-Net-like CNN model aligned to the priorities set forth in this work and the characteristics of the high-resolution flood prediction problem at hand. Additionally, we test the proposed surrogate modeling strategy on two existing architectures: a purely transformer-based model and a CNN with additive attention gates. The comparison results are reported in Sec. D.

Turning to the selection of the loss function, a number of alternatives can be considered, including mean squared Error (MSE), mean absolute error (MAE), Huber loss (Huber, 1992), and its reversed variant Berhu (Owen, 2007). The choice can be informed by analyzing the distribution of water depth values in the dataset and through experimentation. For the current data, the best results were attained

[1]The choice of values in $\mathcal{C}$ is intended for centering the input data around 0.

with the Huber loss function L_{Huber}, which sets the loss for each point in the output to

$$L_{\text{Huber}}(\delta) = \begin{cases} \frac{1}{2}\delta^2, & \text{If } |\delta| \leq \theta \\ \theta|\delta| - \frac{1}{2}\theta^2 & \text{otherwise} \end{cases}, \tag{1}$$

where δ denotes the error between the predicted and ground truth water depth values and $\theta \geq 0$ is a parameter. Recall that by construction, the predicted inundation maps will contain artificially added (background) points for which depth estimation is irrelevant. Therefore, the latter were masked out, and the loss was evaluated only on the valid points that correspond to the $d_{\boldsymbol{y}}$ locations of interest.

C Details of CASPIAN

The architecture of the introduced lightweight CNN model CASPIAN, a detailed breakdown of which is presented in Fig. 3, can be interpreted as a two-layered structure consisting of (i) a fully convolutional encoder-decoder network with a central bottleneck comprised of a series of ResNeXt (Xie et al., 2017) blocks, and (ii) a cascade of consecutive pooling operations and corresponding supervision blocks linked by skip connections and stacked on top of the encoder and decoder, respectively. In what follows, we discuss the constituents of the proposed model separately, elaborating on their structure, role, and key parameters.

The encoder part of CASPIAN consists of K successive downsampling blocks, which progressively filter and downscale (by a factor of 2 each) the input image (of size $H \times W$) to generate low-resolution hierarchical feature representations. To allow for efficient utilization of model parameters, we construct these blocks in a style similar to Xception Chollet (2017). Specifically, each block, except the first, is built from depthwise convolutions (with stride 2) followed by concatenation (with the feature maps from the pooling path), then pointwise (i.e., 1×1) convolutions and a residual connection around them. The initial downsampling block, which for clarity is illustrated in a disassembled form in the topmost left corner of Fig. 3, instead employs a regular convolution with F filters. To save the number of parameters at higher resolutions, we keep the number of filters F constant across all downsampling blocks. The first block is additionally supplied with the output of a stack of operations from the pooling path, which collectively we refer to as *segregated pooling*. This unit filters the non-background points in the segmented input maps based on their class values into separate channels, which are then concatenated and fed into a pooling layer.

The central segment of CASPIAN, which serves as a bridge between the encoder and decoder, is formed by M repeated ResNeXt blocks with identical configurations and fixed output size of $\frac{H}{2^K} \times \frac{W}{2^K} \times F$. Each block aggregates identity mapping with a set of transformations realized through grouped and pointwise convolutions, as illustrated in Fig. 3. As the low-resolution feature maps produced by the encoder undergo these transformations, the proposed network learns more complex and increasingly global (due to enlarging receptive field) feature representations. In addition to the depth M, this bottleneck path is parameterized by cardinality C and group width w, which control the size and extent of the transformations.

The decoding module in CASPIAN, structurally mirroring the encoder, is assembled from K blocks, which, relying on transposed convolutions (a.k.a., deconvolutions) and pointwise operations, learn to gradually upsample the feature maps distilled by the bottleneck back to the original input resolution $H \times W$. Similar to SegNet Badrinarayanan et al. (2017), instead of channeling the entire feature maps from the encoder to the decoder through skip connections as in U-Net, the proposed network transfers only the output of corresponding pooling layers as depicted in Fig. 3. Additionally, these feature maps are reused for modeling the hydrodynamic interactions among protected and unprotected parts of the coast and guiding the decoding process accordingly. In particular, we complement the first upsampling block with a Modulation block constructed similarly to Squeeze-and-Excitation (SE) unit Hu et al. (2018). This block takes the propagated pooling maps as input and produces a set of F weights, one for each channel in the upsampled feature maps. Scaling the latter with these weights allows the network to recalibrate and rectify the decoding process, selectively emphasizing some channels and suppressing others. As illustrated in Fig. 3, in subsequent upsampling steps, the corresponding modulation blocks can be substituted by the output of the first block.

The output from the decoder is fed to a 1×1 convolution and simultaneously summed over channels. The resulting two $H \times W \times 1$ feature maps are summed, and a ReLU activation is applied to it to

produce the predicted inundation depth map. The effects of the summation operator (which incurs no additional trainable parameters) are examined in Sec. D.3.

D Details of the Quantitative Performance Comparison

D.1 Evaluation Setup and Settings

Dataset Splitting: As mentioned in Sec. 4, a total of 142 input protection scenarios were generated, and the corresponding inundation maps were produced with the employed physics-based coupled hydrodynamic model to construct the main dataset, denoted by $\mathcal{D}$. To ensure the robustness of the results and the reliability of the evaluation, splitting of $\mathcal{D}$ into training, validation and testing sets was repeated multiple times. Specifically, $\mathcal{D}$ was randomly split thrice according to 112-12-18 partitioning, resulting in three different training, validation, and testing sets. For each split, it was ensured there were no overlaps among the three sets. On the training and validation sets, 19-fold data augmentation was applied through the Cutout technique with two patches, each of size 60×60. To test the generalizability of the developed models, a Holdout dataset consisting of 32 handcrafted protection scenarios was additionally constructed (see Sec. G).

Candidate Approaches: The pool of coastal flood prediction methods selected for evaluation involves two main groups. The first (benchmark) group includes two standard regression techniques, namely Linear Regression and Lasso Regression with polynomial features (referred to as Lasso with Poly.), and two commonly employed coastal flooding metamodels, namely Kriging with PCA and Support Vector Regression (SVR). The second group contains four end-to-end DL-based models developed under the proposed framework. Among these, three are based on two existing networks, Attention U-net Oktay et al. (2018) and SWIN-Unet Cao et al. (2022), originally designed for medical image segmentation. To adapt to the present settings, the segmentation heads in both networks were replaced by a 1×1 convolution with a ReLU activation. Additionally, to experiment with the transfer learning technique, we substitute the encoder stack in Attention U-net with the first 16 convolutional layers from the VGG19 network Simonyan and Zisserman (2014) and consider two versions, one with the encoder weights initialized randomly while the other with those pre-trained on the popular ImageNet dataset, which contains more than a million images. The latter model is denoted as Attention U-net‡‡ to discern between these two. Accordingly, to conform to the three-channel input format of VGG19, for both models, the depth of input matrices was expanded by replacing the class values with RGB codes, resulting in an input size of $H \times W \times 3$. The final fourth model in this cohort is based on the introduced lightweight CNN network CASPIAN. To allow for an impartial inter-group comparison, predictions produced by the models in the first group were *post-processed* to replace the *negative values* with zeros. As an additional reference, we employ a naive regressor, termed Baseline Predictor, which outputs 0 if the corresponding coastal location in the input vector was (hypothetically) classified as inundation-safe or otherwise the average peak water level across the entire dataset.

Model Configurations and Implementation Details: Settings of the classical and generalized regression models comprising the first group were determined under experimentation and are listed in Table 2. Linear Regression, SVR, Lasso with Poly., and Kriging with PCA were implemented via Scikit-learn Pedregosa et al. (2011) and SMT Bouhlel et al. (2019) Python packages. The implementations (in Tensorflow Keras v2.1) of Attention U-net and SWIN-Unet were borrowed and adapted from Sha (2021). The proposed model CASPIAN was built with Tensorflow Keras v2.1. The hyperparameters of Attention U-net were tuned manually and then transferred identically (except the weight initialization in the encoder) to Attention U-net‡‡. For SWIN-Unet and CASPIAN, the selection of hyperparameters was optimized through the Random Search algorithm provided as part of the Keras Tuner library O'Malley et al. (2019) (see Sec. F for details). For brevity, Table 2 reports only the total number of trainable parameters of these models, whereas the hyperparameter values are relegated to Sec. E.

Training Specifications: All the four DL models were trained with Adam optimizer under the Huber Loss (as defined in Eq. 1) function with $\theta = 0.5$ and batch size of 2. The adopted learning schedule, determined through trials with several alternatives, was set to start with a gradual warm-up that increases the learning rate from 0 to LR linearly for 20 epochs, followed by 200 epochs of the main training session wherein the learning rate was reduced ($\times 0.85$) whenever the validation loss plateaued (patience = 10). During the main training, early stopping was applied if no improvement in

the validation loss was recorded for 40 consecutive epochs. LR was set to $1.5 \cdot 10^{-4}$ for Attention U-Net and Attention U-net[‡‡], to $1.8 \cdot 10^{-4}$ for SWIN-Unet, and to $8 \cdot 10^{-4}$ for CASPIAN. All the models were trained and evaluated on a desktop machine with an Intel Core i9 3.00 GHz CPU, 64 GB of RAM and a single NVIDIA RTX 4090 GPU.

Evaluation Metrics: As emphasized in Al Kajbaf and Bensi (2020), in most prior works studying coastal flood prediction, the performance of developed coastal metamodels was assessed considering only a few basic aggregate metrics, such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) or Coefficient of Determination (R^2), which may not adequately reflect the actual quality of predictions. To provide a more comprehensive evaluation, we consider 6 different metrics, including both error and accuracy measures, formally defined as follows:

$$\text{ARTAE} \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{\|\boldsymbol{y}^k-\hat{\boldsymbol{y}}^k\|_1}{\|\boldsymbol{y}^k\|_1}, \quad R^2 \triangleq \frac{1}{N}\sum_{k=1}^{N}(1-\Psi)$$

$$\text{ARMSE} \triangleq \frac{1}{N}\sum_{k=1}^{N}\sqrt{\sum_{i=1}^{d_{\boldsymbol{y}}}\frac{(y_i^k-\hat{y}_i^k)^2}{d_{\boldsymbol{y}}}}, \quad \delta > \Delta \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{\left|\mathcal{S}_\Delta\right|}{d_{\boldsymbol{y}}} \ ,$$

$$\text{AMAE} \triangleq \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{d_{\boldsymbol{y}}}\frac{|y_i^k-\hat{y}_i^k|}{d_{\boldsymbol{y}}}, \quad \text{ACC}[0] \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{\left|\mathcal{O}_k \cup \hat{\mathcal{O}}_k\right|}{\left|\mathcal{O}_k\right|}$$

where N is the number of evaluated samples; $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ correspond to the ground truth and predicted peak water levels of the $d_{\boldsymbol{y}}$ locations of interest, respectively; $\bar{\boldsymbol{y}}^k \triangleq \mathbf{1} \cdot \frac{1}{d_{\boldsymbol{y}}}\sum_{i=1}^{d_{\boldsymbol{y}}} y_i^k$ denotes the mean vector of actual peak water level values for the k-th sample; Δ is an error threshold (in meters); $\mathcal{S}_\Delta \triangleq \{i \in [d_{\boldsymbol{y}}] \ : \ |y_i^k - \hat{y}_i^k| > \Delta\}$; $\Psi \triangleq \dfrac{\|\boldsymbol{y}^k-\hat{\boldsymbol{y}}^k\|_2^2}{\|\boldsymbol{y}^k-\bar{\boldsymbol{y}}^k\|_2^2}$; $\mathcal{O}_k \triangleq \{i \in [d_{\boldsymbol{y}}] \ : \ y_i^k = 0\}$; $\hat{\mathcal{O}}_k \triangleq \{i \in [d_{\boldsymbol{y}}] \ : \ \hat{y}_i^k = 0\}$. Here, ARTAE, ARMSE, and AMAE stand for average relative total absolute error, average RMSE, and average MAE, respectively; $\delta > \Delta$ quantifies the fraction of cases where the error in predicted floodwater levels exceeds the specified threshold Δ – an important metric for assessing models' reliability and gaining a more nuanced understanding of their performance; and ACC[0] measures the zero detection rate (that is, models' accuracy of detecting non-inundated locations).

D.2 Results and Comparison

Table 1 summarizes the candidate models' performance on the Test and Holdout datasets averaged over the three data splits. Notable observations from the first group of benchmark models are as follows. The two-stage Kriging with PCA approach, commonly employed in prior works, slightly improves upon Linear Regression by achieving 22.1% ARTAE, but δ errors are comparable, indicating a similar prediction quality. On the other hand, we observe significant improvement with Lasso with Poly, which reached the highest ARMSE of 0.42 and $R^2 = 0.95$ across all the models while achieving a $\delta > 0.5$ error of 5-7%, about half of the error produced by Kriging with PCA. This signifies that the incorporation of engineered input features that model the interactions between the protected and unprotected precincts can improve the prediction quality. Among the first group of models, SVR achieved the lowest δ errors, which indicates a higher quality of predictions, yet ACC[0] is extremely low at around 20%. However, it should be noted that in the case of SVR, a separate model has to be trained for every output coastal location independently, which raises potential scalability issues.

The DL models trained with the proposed framework, comprising the second group, significantly outperformed the above benchmark models in terms of AMAE (by a factor of 2 on average), ARTAE (by a factor of 2-5) and δ errors (by a factor of 2-5), and especially for ACC[0] (more than two-fold). The version of Attention U-net with the pre-trained weights achieved only modest improvements over the version trained from scratch, specifically a marginal improvement of 0.1% of ARTAE and 0.01% of $\delta > 0.5$ error. This result could possibly be attributed to the stark difference in image modalities and sizes between the current dataset and ImageNet, aligning with the findings and conclusions drawn in Zhuang et al. (2021). The two best-performing candidate models were SWIN-Unet and CASPIAN, the former a close runner-up to the latter for the majority of the metrics. Notably,

CASPIAN attained the highest scores for all metrics in the Holdout dataset while requiring only a fraction of SWIN-Unet's model's size. As reported in Table 1, CASPIAN achieved $\delta > 0.5$ error of only 1% on both datasets, and the performance with respect to the other metrics is consistent on both datasets, demonstrating the generalizability of the introduced CNN model.

Model	Test Dataset (18 samples)							Holdout Dataset (32 samples)						
	(Lower is better)					*(Higher is better)*		*(Lower is better)*					*(Higher is better)*	
	AMAE	ARMSE	ARTAE (%)	$\delta > 0.5$ (%)	$\delta > 0.1$ (%)	R^2	ACC[0] (%)	AMAE	ARMSE	ARTAE (%)	$\delta > 0.5$ (%)	$\delta > 0.1$ (%)	R^2	ACC[0] (%)
Baseline Predictor	1.14 ± 0.46	2.19 ± 0.63	58.5 ± 32.3	27 ± 11.2	55.9 ± 25.7	-0.11 ± 0.25	57.9 ± 29.9	1.33 ± 0.28	2.49 ± 0.28	62.2 ± 19	29.9 ± 7	52.6 ± 12.3	-0.05 ± 0.23	70.9 ± 12
Linear Regression	0.20 ± 0.06	0.57 ± 0.15	34 ± 88.2	12.5 ± 2.85	19.3 ± 4.01	0.88 ± 0.17	41.4 ± 17.1	0.2 ± 0.09	0.58 ± 0.21	9.18 ± 2.69	12.3 ± 3.61	18.6 ± 4.12	0.93 ± 0.05	41.8 ± 15.6
Kriging with PCA	0.19 ± 0.07	0.55 ± 0.16	22.1 ± 49.5	11.3 ± 4.2	18.4 ± 5.26	0.91 ± 0.09	42.1 ± 16.4	0.21 ± 0.09	0.59 ± 0.22	9.31 ± 3.01	12 ± 3.83	19.3 ± 4.51	0.93 ± 0.05	42.8 ± 14.6
SVR**	0.17 ± 0.08	0.69 ± 0.26	19.9 ± 42.1	4.47 ± 1.68	6.27 ± 2.76	0.87 ± 0.08	21.3 ± 8.1	0.17 ± 0.12	0.7 ± 0.33	7.65 ± 3.94	4.44 ± 2.61	6.13 ± 3.01	0.9 ± 0.1	19.2 ± 6.84
Lasso with Poly.	0.14 ± 0.05	**0.42 ± 0.15**	22.5 ± 55.6	5.45 ± 2.65	16.3 ± 5.16	**0.95 ± 0.03**	10.8 ± 3.82	0.16 ± 0.08	0.46 ± 0.22	7.09 ± 2.42	6.72 ± 3.07	14.3 ± 4.58	0.95 ± 0.05	10.8 ± 4.6
Attention U-net	0.11 ± 0.07	0.64 ± 0.26	5.56 ± 5.73	1.9 ± 1.35	4.04 ± 2.4	0.89 ± 0.08	97.3 ± 3.08	0.13 ± 0.1	0.69 ± 0.32	5.38 ± 3.59	2.29 ± 2.08	4.71 ± 2.79	0.9 ± 0.08	**98.8 ± 1.49**
Attention U-net‡‡	0.10 ± 0.07	0.64 ± 0.26	5.41 ± 5.73	1.92 ± 1.37	**3.95 ± 2.38**	0.89 ± 0.08	97.2 ± 3.12	0.12 ± 0.1	0.69 ± 0.32	5.3 ± 3.57	2.28 ± 2.08	**4.52 ± 2.84**	0.9 ± 0.08	98.8 ± 1.51
SWIN-Unet	**0.07 ± 0.04**	**0.42 ± 0.18**	**3.21 ± 2.15**	**1.37 ± 0.89**	7.19 ± 3.69	**0.95 ± 0.03**	**97.3 ± 2.44**	**0.08 ± 0.06**	**0.45 ± 0.21**	**3.32 ± 2.05**	**1.70 ± 1.61**	7.58 ± 3.84	**0.95 ± 0.04**	98.1 ± 3.57
CASPIAN	**0.06 ± 0.03**	0.46 ± 0.18	**3.12 ± 1.89**	**1.06 ± 0.67**	**3.14 ± 1.65**	**0.94 ± 0.03**	**98.5 ± 1.84**	**0.06 ± 0.04**	**0.45 ± 0.19**	**2.80 ± 1.46**	**1.01 ± 0.87**	**3.99 ± 2.67**	**0.96 ± 0.03**	**99.1 ± 1.10**

Table 1: Quantitative comparison of the candidate coastal flood prediction methods. The models developed with the proposed Deep Visual Learning framework are highlighted in green. The results for each metric are reported as the mean and standard deviation across the samples over the three data splits. The top scores are highlighted in blue, and the runner-ups are in orange. The superscript ** denotes an ensemble of individual models, each trained for one specific coastal location.

Model	Input size	Output size	Training time (hr.)	Inference time (ms.)	Parameters (# or settings)
Baseline Predictor	12066	12066	$\ll 0.001$	23	1
Linear Regression	17	12066	$\ll 0.001$	≈ 1	17
Kriging with PCA	17	12066	≈ 0.001	≈ 8.5	Reg. f. : linear Cor. f. : sq. exp.
SVR**	17	$1 \cdot (12066)$	≈ 0.001	≈ 3770	Kernel : linear $C = 5, \epsilon = 0.05$
Lasso with Poly.	154	12066	≈ 0.029	≈ 0.49	Int._only: True Degree : 2
Attention U-net	$H \times W \times 3$	$H \times W \times 1$	≈ 7.5	≈ 25	≈ 12 Mil.
Attention U-net ‡‡	$H \times W \times 3$	$H \times W \times 1$	≈ 8.6	≈ 25	≈ 12 Mil.
SWIN-Unet	$H \times W \times 1$	$H \times W \times 1$	≈ 2	≈ 50	≈ 8.3 Mil.
CASPIAN	$H \times W \times 1$	$H \times W \times 1$	≈ 3.5	≈ 25	≈ 0.36 Mil.

Table 2: Addendum to Table 1.

D.3 Ablation Experiments

To supplement the quantitative results reported in Sec. D.2, this section ablates the key architectural components introduced in CASPIAN. In particular, we remove/truncate the blocks/modules individually in four steps, resulting in the following versions: (i) CASPIAN without the final channel-wise summation, abbreviated as CASPIAN$_{\mathrm{B}}$, (ii) CASPIAN with the depth of the central bottleneck reduced to 2 (i.e., $M = 2$), denoted as CASPIAN$_{\Gamma}$, (iii) CASPIAN with the modulation block removed, denoted as CASPIAN$_{\mathrm{Z}}$, (iiii) CASPIAN with the pooling path completely eliminated, referred to as CASPIAN$_{\Omega}$.

The results of ablation experiments are tabulated in Table 3. CASPIAN$_{\mathrm{B}}$ and CASPIAN$_{\Gamma}$ achieved similar results on the test dataset compared to CASPIAN, yet $\delta > 0.1$ error of the produced predictions on the holdout dataset nearly doubled, indicating poorer generalizability. This observation corroborates the importance of the proposed deep central bottleneck and the final summation operator. In the case of CASPIAN$_{\mathrm{Z}}$ and CASPIAN$_{\Omega}$, a significant drop in performance was observed on both datasets approaching that of Attention U-Net. This outcome can be expected since, after the elimination of the pooling path, the architecture of the latter two more closely resembles that of Attention U-Net.

Model	Test Dataset (18 samples)			Holdout Dataset (32 samples)			# of parameters
	AMAE	ARMSE	$\delta > 0.1$ (%)	AMAE	ARMSE	$\delta > 0.1$ (%)	
CASPIAN$_{\text{B}}$	0.06 $\pm$ 0.03	0.44 $\pm$ 0.19	3.61 $\pm$ 2.12	0.07 $\pm$ 0.06	0.45 $\pm$ 0.20	6.27 $\pm$ 10.5	= CASPIAN
CASPIAN$_{\Gamma}$	0.06 $\pm$ 0.04	0.45 $\pm$ 0.19	5.62 $\pm$ 4.91	0.07 $\pm$ 0.05	0.47 $\pm$ 0.20	5.59 $\pm$ 4.41	$\approx$ 0.215 Mil.
CASPIAN$_{\text{Z}}$	0.10 $\pm$ 0.07	0.58 $\pm$ 0.30	7.04 $\pm$ 3.40	0.11 $\pm$ 0.09	0.62 $\pm$ 0.29	7.46 $\pm$ 2.81	$\approx$ 0.344 Mil.
CASPIAN$_{\Omega}$	0.10 $\pm$ 0.07	0.55 $\pm$ 0.31	6.21 $\pm$ 2.97	0.11 $\pm$ 0.09	0.62 $\pm$ 0.29	6.93 $\pm$ 2.81	$\approx$ 0.341 Mil.

Table 3: Results of the ablation studies.

E HYPERPARAMETERS OF THE TRAINED DL MODELS

Attention U-net and **Attention U-net**‡‡**:**

- `input_size` = (1024, 1024, 3)
- `filter_num` = [32, 64, 128, 256] (number of filters for each down- and up-scaling level)
- `stack_num_down` = 2 (number of layers per downsampling level/block)
- `stack_num_up` = 2 (number of layers (after concatenation) per upsampling level/block
- `activation` = 'ReLU'
- `atten_activation` ='ReLU'
- `output_activation` ='ReLU'
- `batch_norm` = False
- `backbone` = 'VGG19' (the backbone model name)
- `encoder_weights` = 'random' (**for Attention U-net**) and 'imagenet' (**for Attention U-net**‡‡)
- `freeze_backbone` = False

SWIN-Unet:

- `filter_num_begin` = 64 (number of channels in the first downsampling block)
- `depth` = 4 (the depth of Swin U-net, 4 means 3 down/upsampling levels and a bottom level)
- `stack_num_down` = 2 (number of Swin Transformers per downsampling level)
- `stack_num_up` = 2 (number of Swin Transformers per upsampling level)
- `patch_size` = 8
- `att_heads` = 4 (number of attention heads per down/upsampling level)
- `w_size` = 4 (the size of attention window per down/upsampling level)
- `mlp_ratio` = 4 (ratio of MLP hidden dimension to embedding dimension)

CASPIAN:

- `input_shape` = (1024, 1024, 1)
- `F` = 72
- `K` = 4
- `C` = 34
- `M` = 8
- `modulation_level` = 1

- `p` = 4
- `r` = 0.85 × F
- `activation` = `'tanh'`
- `init` = `"glorot_normal"`

F Details of the Hyperparameter Tuning

The hyerparameters of the trained DL models reported in the previous section were selected as follows. Since Attention U-net and Attention U-net[‡‡] were implemented with the VGG19 backbone, we tuned the available remaining hyperparameters (learning rate and activations) manually. For SWIN-Unet and CASPIAN, the selection of hyperparameters was optimized through the Random Search algorithm provided as part of the Keras Tuner library. The considered search space of hyperparameter values for both models is listed hereunder.

Hyperparameter Search Space for CASPIAN:

- `F` : $\{min = 64, max = 128, step = 4\}$
- `K` : $\{min = 3, max = 6, step = 1\}$
- `C` : $\{min = 12, max = 64, step = 4\}$
- `M` : $\{min = 4, max = 12, step = 2\}$
- `modulation_level` : $\{min = 1, max = 3, step = 1\}$
- `p` : $\{min = 2, max = 8, step = 2\}$
- `r` : $\{min = 0.5, max = 0.95, sampling = linear\}\cdot$ `F`
- `activation` : ['relu', 'tanh', 'swish']
- `learning_rate` : $\{min = 5e-5, max = 5e-3, sampling = log\}$

Hyperparameter Search Space for SWIN-Unet:

- `filter_num_begin` : $\{min = 64, max = 160, step = 16\}$
- `depth` : $\{min = 3, max = 4, step = 1\}$
- `stack_num_(up down)` : $\{min = 1, max = 2, step = 1\}$
- `patch_size` : [4, 8, 16]
- `att_heads` : $\{min = 2, max = 4, step = 2\}$
- `dropout` : [0.0, 0.05, 0.1, 0.2]
- `learning_rate` : $\{min = 1e-5, max = 5e-4, sampling = log\}$

G Holdout Dataset

Table 4: Protection scenarios considered in the Holdout dataset.

Holdout protection scenarios (32 in total)				
00110011001100110	11100000000000111	00000111100000111	00011000110001100	11110000111100001
00000011111100000	11110000000001111	00000111111100000	11111100000111111	00001111111110000
11111000001111100	00001110000111000	10101010101010101	11111110000001111	00000001111110000
11111000000011111	11111110000000111	00000111110000011	00011100011100011	00000001111111000
11000000000000011	00111111111111100	01010101010101010	11111100000011111	11111000011111000
00000011111000000	11110001111000111	11100011100011100	00001111000011110	11001100110011001
11100111001110011	00011111111111000			