Climate Adaptation-Aware Flood Prediction for Coastal Cities Using Deep Learning

Bilal Hassan* **Areg Karapetyan** **Aaron Chung Hin Chow** **Samer Madanat**
Division of Engineering
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates

Abstract

Climate change and sea-level rise (SLR) pose escalating threats to coastal cities, intensifying the need for efficient and accurate methods to predict potential flood hazards. Traditional physics-based hydrodynamic simulators, although precise, are computationally expensive and impractical for city-scale coastal planning applications. Deep Learning (DL) techniques offer promising alternatives, however, they are often constrained by challenges such as data scarcity and high-dimensional output requirements. Leveraging a recently proposed vision-based, low-resource DL framework, we develop a novel, lightweight Convolutional Neural Network (CNN)-based model designed to predict coastal flooding under variable SLR projections and shoreline adaptation scenarios. Furthermore, we demonstrate the ability of the model to generalize across diverse geographical contexts by utilizing datasets from two distinct regions: Abu Dhabi (AD) and San Francisco (SF). Our findings demonstrate that the proposed model significantly outperforms state-of-the-art methods, reducing the mean absolute error (MAE) in predicted flood depth maps on average by nearly 20%. These results highlight the potential of our approach to serve as a scalable and practical tool for coastal flood management, empowering decision-makers to develop effective mitigation strategies in response to the growing impacts of climate change. **Project Page: https://caspiannet.github.io/**

1 Introduction

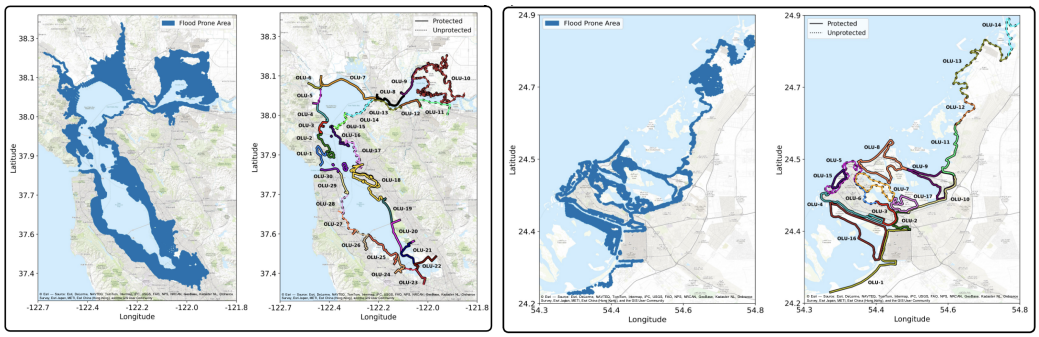
Low-elevation coastal zones [1] are hotspots for climate change-induced risks, with flood threats to cities projected to rise nine-fold by 2050 [2]. To safeguard residents, communities are armoring their shorelines with *engineered structures* like seawalls [3, 4]. While beneficial for *local flood protection*, these defenses can alter coastal hydrodynamics, unintentionally amplifying flooding in other, otherwise unaffected, regions [5, 6, 7].

To understand these complex dynamics, physics-based simulators like Delft3D [8] are employed. While these tools provide detailed accuracy, they are *computationally prohibitive*, often requiring days to simulate a single shoreline scenario [9, 10, 11]. This burden limits their use in coastal planning, where numerous adaptation strategies must be evaluated [12, 13]. In response, data-driven methods (commonly referred to as *surrogate models* or *metamodels*) have emerged as promising alternatives for rapid flood prediction [14, 15], learning complex input-output relationships without explicitly modeling the underlying physics.

Despite these advancements, limited attention has been paid to the *joint* incorporation of *SLR* and *shoreline protection* scenarios. Developing accurate DL models for this *climate adaptation-aware* setting poses challenges like *data scarcity* and the *high dimensionality of outputs* [9, 10, 11]. A prior

*Corresponding author: bilal.hassan@nyu.edu

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2025.



(a) (b)

Figure 1: Study areas: (a) San Francisco Bay (30 OLUs) and (b) Abu Dhabi (17 OLUs). Left panels show baseline flooding without protection, while right panels show the defined OLUs.

vision-based framework demonstrated the viability of this approach, but was *limited* to a *single location* and a *particular* SLR scenario [11]. Taking a step further, this work introduces a novel DL model that generalizes across distinct coastal regions and multiple SLR levels. Our key contributions are:

1. We develop a novel DL model, CASPIAN-v2, that accurately predicts high-resolution coastal flooding under various SLR and shoreline protection strategies. Its lightweight design enables fast, scalable prediction, slashing the compute time of hydrodynamic models.
2. We provide a new dataset of flood maps for two vulnerable locations, Abu Dhabi and the San Francisco Bay Area, covering diverse SLR scenarios and shoreline adaptations.
3. We conduct a rigorous benchmark against state-of-the-art (SOTA) models to validate the performance and generalization capabilities of our proposed framework.

Put together, this research provides engineers and policymakers with a practical flood prediction tool, readily integrated into large-scale planning workflows to enhance the resilience of coastal cities against accelerating sea level rise.

2 Study Areas and Dataset Generation

In this research, we examine two vulnerable metropolitan coastal areas (AD and the SF Bay Area) to predict coastal flooding under various SLR values and shoreline protection scenarios. Both locations feature low-lying topographies and significant urbanization, making them particularly susceptible to environmental effects. To model diverse shoreline protection strategies under various SLR scenarios, the complex coastlines were discretized into Operational Landscape Units (OLUs): 17 for AD, as outlined in [16], and 30 for the SF Bay Area, based on prior studies [3, 5, 17]. Figure 1a depicts the OLU delineations and flood-prone zones in both regions.

The ground truth flood data for training our DL model was generated using the Delft3D model [8], a high-fidelity hydrodynamic simulator that integrates key physical processes like SLR and tidal dynamics. Further details concerning the employed hydrodynamic model, its validation, and the specifics of the resulting dataset, including training, validation, and test splits, can be found in Appendix Sec. A.

3 Problem Statement

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* and SLR value. 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

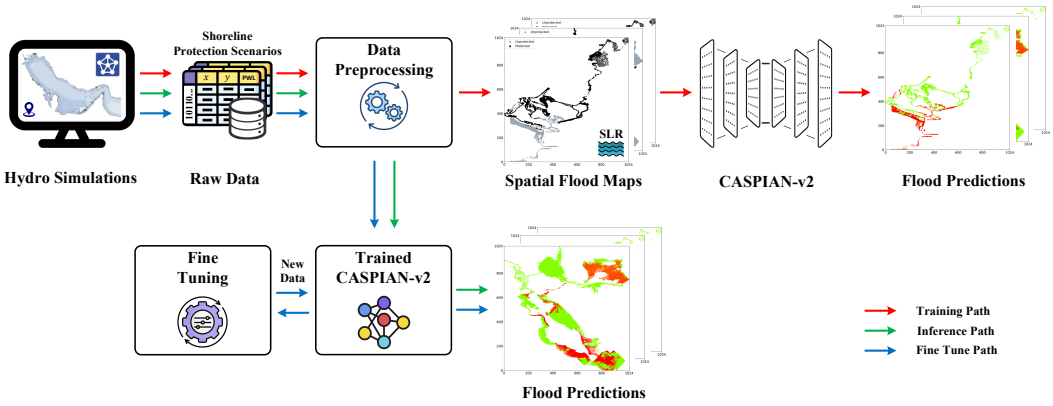


Figure 2: An overview of the proposed framework for coastal flood prediction.

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 $f : \boldsymbol{x} \in \mathcal{X}, l \in \mathbb{R} \rightarrow \boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$, where l denotes the SLR level, provided with a set $\{(\boldsymbol{x}^k, l^k, \boldsymbol{y}^k) \mid k \in [n], l^k \in \mathbb{R}, \boldsymbol{x}^k \in \mathcal{X}, \boldsymbol{y}^k \in \mathbb{R}^{d_{\boldsymbol{y}}}\}$ 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 even when considering a single value for SLR (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 to hundreds of thousands.

4 Proposed Methodology

Our end-to-end framework, illustrated in Figure 2, enables rapid and accurate coastal flood prediction. The pipeline converts data from physics-based hydrodynamic simulations into 2D spatial maps, which are used to train our core predictive model, CASPIAN-v2. Once trained, the model can generate high-resolution flood maps for new scenarios in seconds (inference) and can be efficiently adapted to new geographical regions or climate conditions via a fine-tuning path.

The core of the framework is CASPIAN-v2, an advanced encoder-decoder architecture designed for robust flood prediction. It incorporates several key innovations: a novel Multi-Attention ResNeXt (MARX) block in the bottleneck stage enhances the model's focus on the most critical spatial features of a flood event. Furthermore, a specialized SLR-Enhanced Encoding (SEE) block in the decoder integrates SLR data, guiding the reconstruction process to produce predictions conditioned on different climate scenarios.

To optimize this complex regression task, we developed a custom hybrid loss function. By combining Huber, Log-Cosh, and Quantile losses, this function makes training more robust to outliers, stabilizes gradients, and provides a balanced handling of over- and under-prediction errors. This tailored approach is critical to achieving the model's high accuracy. A detailed exposition of the model architecture and the formal loss function are provided in the Appendix Sec. B.

5 Results

Quantitative Analysis: To validate the performance of CASPIAN-v2, we benchmarked it against a comprehensive suite of SOTA ML and DL models. The full experimental protocol, including dataset splits, baseline model specifications, and evaluation metrics, is detailed in the Appendix Sec. C. The evaluation demonstrates the superior predictive power of our proposed model. Quantitatively, CASPIAN-v2 significantly outperforms the best traditional ML model (Lasso with Polynomial features), reducing the AMAE by 51.65%. It also shows a 19.96% AMAE reduction compared to the best-performing SOTA DL model. A key advantage of CASPIAN-v2 is its computational efficiency; it can predict 72 flood scenarios in approximately 16 seconds, a task that would require the physics-based Delft3D simulator around 115 days to complete. This dramatic speed-up underscores its potential as a practical tool for rapid, real-world coastal planning. The full quantitative evaluation, including

performance on test and holdout sets, a complete SOTA comparison table, and generalizability assessments is provided in the Appendix Sec. D.1.

Qualitative Analysis: For further scrutiny, we visually analyzed the model's ability to accurately capture the spatial extent of flooding. Figure 3 presents a visual comparison of the spatial accuracy between CASPIAN-v2 and the top-performing baseline model on a representative test case. The map breaks down the prediction into correctly matched inundated areas (green), over-predicted areas where the model incorrectly flagged flooding (orange), and under-predicted areas where it missed flooding (purple). The visualization clearly reveals that while the baseline model produces a more fragmented prediction with significant patches of both over- and under-prediction, the output from CASPIAN-v2 aligns much more closely with the ground truth. Its predicted flood extent is more coherent and captures the true inundation boundaries with far fewer spatial errors. This visual evidence aligns with the quantitative metrics, confirming the model's superior ability to learn and reproduce complex flood dynamics. Additional qualitative visualizations are available in the Appendix Sec. D.2.

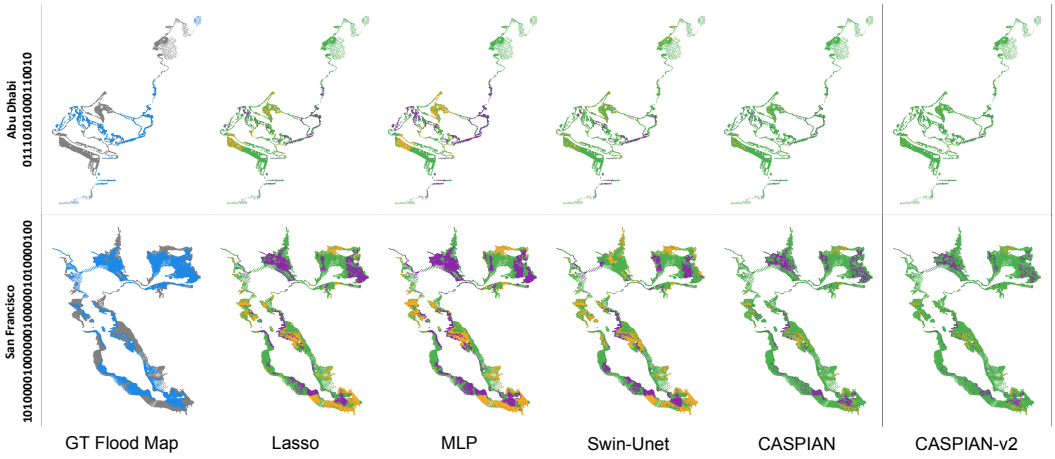


Figure 3: Visual comparison of spatial prediction accuracy. Green indicates correctly predicted inundated areas (true positives), orange indicates over-prediction (false positives), and purple indicates under-prediction (false negatives). CASPIAN-v2 demonstrates a larger matched area and more coherent flood boundaries than the top-performing baseline.

6 Concluding Remarks

In this work, we introduced CASPIAN-v2, a state-of-the-art deep learning framework for coastal flood prediction that is significantly more accurate and computationally efficient than existing methods, achieved through a novel architecture with custom loss and attention mechanisms. Crucially, the model is designed for practical use, offering both interpretability to build trust and predictive uncertainty quantification to guide decision-making (see Appendix, Sec. G). This makes CASPIAN-v2 a powerful tool for stakeholders, enabling the rapid assessment of thousands of coastal protection scenarios to address critical climate adaptation challenges.

References

[1] Roderik van de Wal et al. "Sea Level Rise in Europe: Impacts and consequences". In: *State of the Planet* 3 (2024), pp. 1–33.

[2] Stephane Hallegatte et al. "Future flood losses in major coastal cities". In: *Nature Climate Change* 3.9 (Sept. 2013), pp. 802–806. ISSN: 1758-6798. DOI: 10.1038/nclimate1979.

[3] Julie Beagle et al. *San Francisco Bay shoreline adaptation atlas: Working with nature to plan for sea level rise using operational landscape units*. SFEI publication# 915. 2019.

[4] Andrew Lewis. *After a Decade of Planning, New York City Is Raising Its Shoreline*. Yale School of the Environment. https://e360.yale.edu/features/new-york-city-climate-plan-sea-level-rise. Dec. 2023.

[5] Michelle A Hummel et al. “Economic evaluation of sea-level rise adaptation strongly influenced by hydrodynamic feedbacks”. In: *Proceedings of the National Academy of Sciences* 118.29 (2021), e2025961118.

[6] Ruo-Qian Wang et al. “The influence of sea level rise on the regional interdependence of coastal infrastructure”. In: *Earth’s Future* 6.5 (2018), pp. 677–688.

[7] Ivan D Haigh et al. “The tides they are a-Changin’: A comprehensive review of past and future nonastronomical changes in tides, their driving mechanisms, and future implications”. In: *Reviews of Geophysics* 58.1 (2020), e2018RG000636.

[8] Deltares. *Delft3d*. `https://oss.deltares.nl/web/delft3d`.

[9] Aikaterini P Kyprioti et al. “Storm hazard analysis over extended geospatial grids utilizing surrogate models”. In: *Coastal Engineering* 168 (2021), p. 103855.

[10] Jeremy Rohmer et al. “Improved metamodels for predicting high-dimensional outputs by accounting for the dependence structure of the latent variables: application to marine flooding”. In: *Stochastic Environmental Research and Risk Assessment* 37.8 (2023), pp. 2919–2941.

[11] Anonymous Authors. “Anonymous Title”. In: *Anonymous Journal* (2024).

[12] Guangzhao Chen et al. “Urban inundation rapid prediction method based on multi-machine learning algorithm and rain pattern analysis”. In: *Journal of Hydrology* 633 (2024), p. 131059.

[13] Bingkun Du et al. “Urban flood prediction based on PCSWMM and stacking integrated learning model”. In: *Natural Hazards* (2024), pp. 1–25.

[14] Amir Mosavi, Pinar Ozturk, and Kwok-wing Chau. “Flood prediction using machine learning models: Literature review”. In: *Water* 10.11 (2018), p. 1536.

[15] Ainaa Hanis Zuhairi et al. “Review of flood prediction hybrid machine learning models using datasets”. In: *IOP Conference Series: Earth and Environmental Science*. Vol. 1091. IOP Publishing. 2022, p. 012040.

[16] Aaron CH Chow and Jiayun Sun. “Combining Sea level rise inundation impacts, tidal flooding and extreme wind events along the Abu Dhabi coastline”. In: *Hydrology* 9.8 (2022), p. 143.

[17] Jiayun Sun, Aaron CH Chow, and Samer Michel Madanat. “Multimodal transportation system protection against sea level rise”. In: *Transportation Research Part D: Transport and Environment* 88 (2020), p. 102568.

[18] Gustavo AM De Almeida and Paul Bates. “Applicability of the local inertial approximation of the shallow water equations to flood modeling”. In: *Water Resources Research* 49.8 (2013), pp. 4833–4844.

[19] Jeffrey Neal, Guy Schumann, and Paul Bates. “A subgrid channel model for simulating river hydraulics and floodplain inundation over large and data sparse areas”. In: *Water Resources Research* 48.11 (2012).

[20] Zhi Li and Ben R Hodges. “Modeling subgrid-scale topographic effects on shallow marsh hydrodynamics and salinity transport”. In: *Advances in Water Resources* 129 (2019), pp. 1–15.

[21] Brett F Sanders and Jochen E Schubert. “PRIMo: Parallel raster inundation model”. In: *Advances in Water Resources* 126 (2019), pp. 79–95.

[22] N Nithila Devi and Soumendra Nath Kuiry. “A novel local-inertial formulation representing subgrid scale topographic effects for urban flood simulation”. In: *Water Resources Research* 60.5 (2024), e2023WR035334.

[23] Patrick L Barnard et al. “Development of the Coastal Storm Modeling System (CoSMoS) for predicting the impact of storms on high-energy, active-margin coasts”. In: *Natural hazards* 74 (2014), pp. 1095–1125.

[24] Ruo-Qian Wang et al. “Interactions of estuarine shoreline infrastructure with multiscale sea level variability”. In: *Journal of Geophysical Research: Oceans* 122.12 (2017), pp. 9962–9979.

[25] Fahad Al Senafi and Ayal Anis. “Shamals and climate variability in the Northern Arabian/Persian Gulf from 1973 to 2012”. In: *International Journal of Climatology* 35.15 (2015), pp. 4509–4528.

[26] Dapeng Li, Ayal Anis, and Fahad Al Senafi. “Physical response of the Northern Arabian Gulf to winter Shamals”. In: *Journal of Marine Systems* 203 (2020), p. 103280.

[27] IPCC. “Climate change 2021: the physical science basis”. In: *Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change* 2.1 (2021), p. 2391.

[28] Saining Xie et al. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.

[29] Sanghyun Woo et al. “Cbam: Convolutional block attention module”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[30] Peter J Huber. “Robust estimation of a location parameter”. In: *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 492–518.

[31] Resve A Saleh and AK Saleh. “Statistical properties of the log-cosh loss function used in machine learning”. In: *arXiv preprint arXiv:2208.04564* (2022).

[32] Roger Koenker and Gilbert Bassett Jr. “Regression quantiles”. In: *Econometrica: journal of the Econometric Society* (1978), pp. 33–50.

[33] Ali Hassani et al. “Escaping the big data paradigm with compact transformers”. In: *arXiv preprint arXiv:2104.05704* (2021).

[34] Ozan Oktay et al. “Attention u-net: Learning where to look for the pancreas”. In: *arXiv preprint arXiv:1804.03999* (2018).

[35] Hu Cao et al. “Swin-unet: Unet-like pure transformer for medical image segmentation”. In: *European conference on computer vision*. Springer. 2022, pp. 205–218.

[36] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

A Details of the Study Areas and Dataset Generation

A.1 Data Sources and Hydrodynamic Simulations

The ground truth flood data used for training and evaluating our surrogate model was generated through a series of physics-based hydrodynamic simulations using the Delft3D model. This model integrates key physical processes including SLR and tidal dynamics. High-resolution bathymetry and digital elevation models (DEM) (with data sources such as TanDEM-X, Landsat-8, and Nautical Charts) were used for both regions (AD and SF Bay Area) to ensure accurate modeling of coastal topography that transitions smoothly between sea and the land. While some authors [18, 19, 20, 21, 22] address subgrid details by using separate subgrid nesting methods, we have retained the same governing equations but used a 30 m model grid in the areas of interest, and Delft3D is capable of automatically modeling wetting and drying of grid cells from one time step to the next.

The accuracy and reliability of these physics-based models were established through rigorous validation against real-world observations. For San Francisco Bay, the Delft3D model was adapted from the CoSMoS model originally developed by [23] and adapted to San Francisco Bay by [24], and validated in the past using tidal gages at 9 tidal gage locations in and around San Francisco Bay. Pearson correlation coefficients ranged from 0.9862 to 0.9996, while the root mean square (RMS) ratios (the ratio of modeled versus measured RMS amplitudes) ranged from 0.973 to 1.027 (please refer to [24])

For Abu Dhabi, the Delft3D model was validated using water level data from 196 tidal gage locations throughout the Gulf (as the hydrodynamic model encompassed the entire Gulf in addition to the western portions of the Gulf of Oman). The water levels at these locations were compared with one month's worth of hydrodynamic simulation, and the resulting absolute root mean square error (RMSE) values ranged from 0.0013 to 0.0043 m in the vicinity of Abu Dhabi. More validation details for Abu Dhabi can be found in [16]. Given this strong validation, the outputs of the hydrodynamic simulations were considered a reliable proxy for ground truth for the purposes of training and evaluating our deep learning framework.

While the Gulf does not typically experience tropical cyclones, it is known for its northwesterly winds generally occurring with winds at about 20 m/s with sudden onset and sustained over a period of up to 3-5 days. These are called the Shamal winds (meaning "North" in Arabic) and occur at least 10 times annually, mainly during the winter months [25, 26]. Accordingly, for Abu Dhabi, we applied a nested SWAN model to simulate wind and wave effects, particularly the impact of these Shamal winds, which can significantly intensify tidal flooding risks. Both the SWAN model and Delft3D models were forced using ERA5 meteorological data in the Gulf.

In both geographic locations, our aim was to generate data that correspond to a hypothetical future extreme flooding scenario, where there was little to no flooding observed without SLR. For AD, simulations were based on a 0.5 m SLR scenario, consistent with regional projections for mid-century SLR (as described above) [27]. The 0.5 m SLR scenario was then coupled with storm surges resulting from a sample 3-month long Shamal event. In contrast, flood simulations for the SF Bay Area were conducted under three SLR scenarios: 0.5 m, 1.0 m, and 1.5 m, which reflects a possible future scenario for San Francisco Bay in the year (somewhere between 2050-2100 depending on the climate change scenario pathway (between SSP2-4.5 and SSP5-8.5) from IPCC AR6 report [27]. Table 1 provides a comprehensive overview of the datasets generated for this study, which are partitioned into three categories based on their purpose. The *Main Set*, comprising the largest datasets from AD (0.5 m SLR) and SF (1.0 m SLR), was used for the primary training, validation, and testing of the CASPIAN-v2 model. The *Holdout Set* consists of scenarios intentionally curated to be challenging (such as protecting one entire side of the SF Bay while leaving the other exposed) and was used for blind testing of the primarily trained model's performance on complex spatial schemes not seen during training. Finally, the *Generalizability Set* includes SF scenarios at different SLR levels (0.5 m and 1.5 m) and was used exclusively to evaluate the ability of the model to adapt to new environmental conditions via fine-tuning.

To balance the need to model a larger number of modeled tidal cycles per simulation, with the computational time and storage space used for the simulations, a 3-month simulation period was also applied for San Francisco Bay. Although our San Francisco model includes riverine input from the Sacramento and San Joaquin Rivers, the inflow rates into the Bay were baseline values rather than for extreme fluvial flood events. While we acknowledge that incorporating more hydrodynamic

forcing conditions to include pluvial and riverine floods, as well as extreme storm events, can refine the hydrodynamic model to reflect more extreme flooding, our overall scope in this paper is in the use of machine learning to be able to act as a surrogate for a hydrodynamic model running under different SLR scenarios. The detailed protocols for how these datasets were split and used are described in Section A.3.

Table 1: Dataset details for AD and SF regions, including OLUs, SLR depths, and the number of unique shoreline protection scenarios. The Main Set was used for primary model training and testing. The Holdout Set was used for blind testing on challenging scenarios. The Generalizability Set was used to evaluate model adaptability to new SLR conditions via fine-tuning.

Region	OLUs	SLR	Protection Scenarios
AD	**17 OLUs:** 1 *(Mussafah)*, 2 *(Bain Al Jesrain)*, 3 *(Grand Mosque District)*, 4 *(AD Island West)*, 5 *(Marina, CBD, Al Mina)*, 6 *(AD Island East)*, 7 *(Al Reem Island)*, 8 *(Saadiyat Island)*, 9 *(Yas Island)*, 10 *(Al Raha Island)*, 11 *(Al Shahama)*, 12 *(Al Rahba)*, 13 *(New Port City)*, 14 *(Ghantoot)*, 15 *(Lulu Island)*, 16 *(Hudayriat Island)*, 17 *(Inner Islands)*	0.5 m 0.5 m	142 (Main Set) 32 (Holdout Set)
SF	**30 OLUs:** 1 *(Richardson)*, 2 *(Corte Madera)*, 3 *(San Rafael)*, 4 *(Gallinas)*, 5 *(Novato)*, 6 *(Petaluma)*, 7 *(Napa - Sonoma)*, 8 *(Carquinez North)*, 9 *(Suisun Slough)*, 10 *(Montezuma Slough)*, 11 *(Bay Point)*, 12 *(Walnut)*, 13 *(Carquinez South)*, 14 *(Pinole)*, 15 *(Wildcat)*, 16 *(Point Richmond)*, 17 *(East Bay Crescent)*, 18 *(San Leandro)*, 19 *(San Lorenzo)*, 20 *(Alameda Creek)*, 21 *(Mowry)*, 22 *(Santa Clara Valley)*, 23 *(Stevens)*, 24 *(San Francisquito)*, 25 *(Belmont - Redwood)*, 26 *(San Mateo)*, 27 *(Colma - San Bruno)*, 28 *(Yosemite - Visitacion)*, 29 *(Mission - Islais)*, 30 *(Golden Gate)*	1.0 m 1.0 m 0.5 m 1.5 m	285 (Main Set) 46 (Holdout Set) 32 (Generalizability Set) 32 (Generalizability Set)

We ran individual Delft3D scenarios (each with a 3-month simulation time as described above) to collect hourly inland inundation data under different coastal protection scenarios to create a dataset for training and validating our DL model. The detailed process for transforming this raw simulation output into the 2D spatial flood maps required by our model is presented in the Section A.2. Our findings highlight the importance of holistic regional flood control measures, especially given the intricate interplay between protected and unprotected zones. Further, the datasets from two regions allowed us to assess the applicability and reliability of the DL model in different vulnerable coastal settings.

The computational cost of generating a peak flood depth map using the coupled hydrodynamic model, which underscores the need for an efficient surrogate, varies significantly between the two study regions. For the coast of Abu Dhabi, the process to generate a map such as the one shown in Fig. 1a(b) takes approximately 71 to 73 hours of elapsed runtime, equating to 1500 to 1660 CPU-hours, depending on the specific protection scenario. This comprehensive simulation includes Delft3D runs, which require 6 to 7 hours on 28 CPU cores (Intel Xeon E5-2680 @ 2.40 GHz; $\approx$ 168–196 CPU-hours), and SWAN simulations, which take about 10 to 11 hours on 128 CPU cores (AMD EPYC 7742 @ 2.25GHz; $\approx$ 1280–1408 CPU-hours). Subsequent post-processing and run-up calculations using Matlab scripts add approximately 55 hours on a single core. In contrast, generating a similar map for San Francisco Bay (see Fig. 1a(a)) is computationally less demanding, requiring approximately 3.5 to 6.0 hours of elapsed time, or 84.5 to 141 CPU-hours. The Delft3D runs for this region take about 3 to 5 hours on 28 CPU cores, and the post-processing of these outputs takes between 0.5 and 1.0 hours on a single core. It is important to note that SWAN and run-up calculations were not performed for the San Francisco Bay shoreline, as its relatively sheltered inland location makes these components unnecessary, accounting for the substantial difference in computational cost.

A.2 Data Preprocessing

The raw, tabular data generated by the Delft3D simulator, which consists of inundation coordinates and corresponding peak water level (PWL) values, is not directly compatible with our 2D DL model. Therefore, a multi-step preprocessing pipeline was developed to transform this data into a structured grid format suitable for a computer vision task.

The first key step was to map the inundation coordinates onto a standardized 1024×1024 spatial grid. This was achieved by defining the grid boundaries based on the maximum spatial extent of all simulation data and then assigning each inundation point to its nearest grid cell. In cases where multiple inundation points mapped to the same cell due to the high density of the data, a conflict resolution strategy was employed that reassigned the conflicting points to the nearest available empty cell, ensuring a unique one-to-one mapping.

Subsequently, we incorporated the shoreline protection information. For each inundation point, we calculated its proximity to the nearest protected and unprotected OLUs and assigned it a class based on which was closer. This classification, along with the PWL values, was then used to construct the final input and output matrices for training. The shoreline protection scenarios were encoded as binary strings, where '0' indicates unprotected OLUs and '1' denotes protected OLUs. This entire process ensures that the model receives spatially coherent input that encodes not just water levels, but also the crucial context of shoreline defense configurations.

A.3 Dataset Splits

The data from both regions is divided into sets for primary model training and for subsequent fine-tuning to assess generalization. The composition of these datasets is detailed in Table 2.

Table 2: Dataset details for primary training and fine-tuning.

Type	Region	SLR	Total	Train	Validation	Test
Primary	AD	0.5 m	142	96	10	36
	SF	1.0 m	285	225	24	36
Fine-tuning	SF	0.5 m	30	20	4	6
	SF	1.5 m	30	20	4	6

To enhance the model's generalization ability and robustness for primary training, we employed a systematic data augmentation strategy on the AD (0.5 m) and SF (1.0 m) training and validation subsets. The augmentation process primarily involves a random remove function, which applies random spatial cutouts and scaling factors to the original samples. Specifically, this technique first identifies the spatial coordinates of the shoreline protection segments and then occludes small, square regions around a random subset of them in the input maps. This process simulates scenarios with imperfect or missing data, forcing the model to learn more robust contextual features rather than memorizing the impact of any single protection segment. We create distinct yet related variants of the original dataset by systematically applying these transformations multiple times ($24\times$ for AD and $10\times$ for SF). Compared to the original sparse dataset, this strategy produces a richer dataset for primary training, comprising 2,304 training samples and 240 validation samples for AD, along with 2,250 training samples and 240 validation samples for SF.

The fine-tuning datasets for SF (0.5 m and 1.5 m SLR) consist of 30 protection scenarios where one OLU was protected at a time. For evaluation, 20% of the data (6 samples) was reserved, while the remaining 80% (24 samples) was used for fine-tuning and validation.

B Details of the Proposed Methodology

B.1 CASPIAN-v2 Architecture

This section provides a detailed technical implementation of the *CASPIAN-v2* architecture, expanding on the conceptual overview presented in Section 4 of the main text. The block diagram of the

CASPIAN-v2 model, as illustrated in Figure 4, comprises three primary stages: encoder, bottleneck, and decoder.

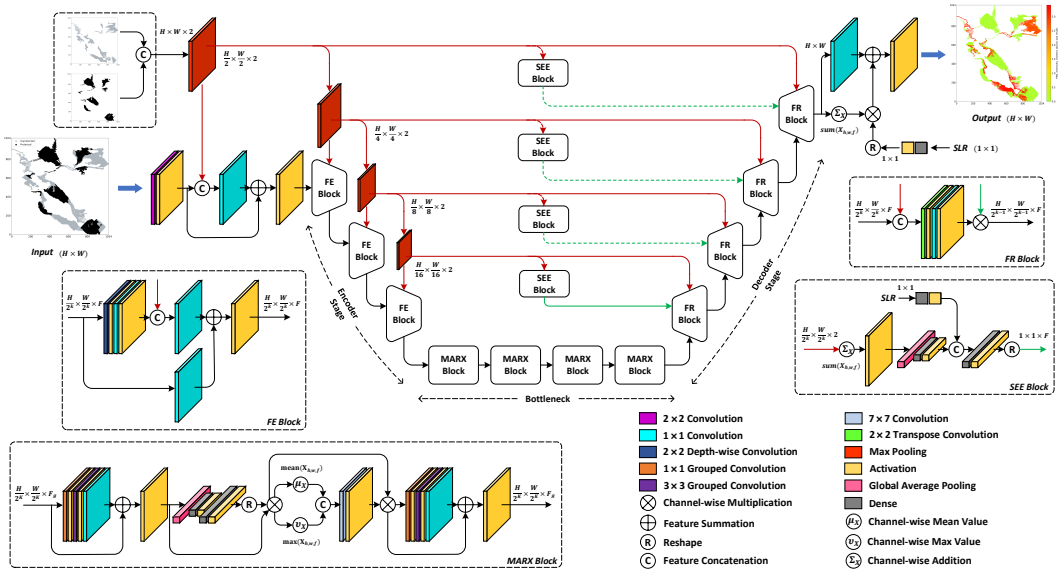


Figure 4: The CASPIAN-v2 model architecture. The encoder extracts hierarchical features using FE blocks; the bottleneck employs MARX blocks to capture high-level representations; and the decoder reconstructs outputs using FR and SEE blocks. Different colors show separate layer operations.

B.1.1 Encoder Stage

The encoder stage of the CASPIAN-v2 model is designed to extract hierarchical features by progressively reducing the spatial dimensions of the input grid while increasing the depth of the feature maps. This process enables the network to capture multi-scale patterns essential for accurate flood prediction.

The model accepts two inputs: an input grid $\mathbf{I} \in \mathbb{R}^{H \times W \times 1}$, where H and W are the spatial dimensions, and a scalar SLR value denoted as $\mathcal{S}$. Since the SLR input $\mathcal{S}$ contains global information affecting the entire spatial domain, it is integrated directly into the decoder part of the network. The encoder stage contains a series of *feature extraction* (FE) blocks, allowing the model to capture both local features, such as specific inundation points and their immediate surroundings, and global patterns, including the overall spatial distribution of protected and unprotected areas.

First, the preprocessed input grid $\mathbf{I}$ is processed through a series of depthwise separable convolutional layers to reduce the spatial dimensions and extract complex features. At each depth level $k = 1$ to K, where K is the total depth of the encoder, the feature map $\mathbf{X}_k$ undergoes several transformations. First, a 2×2 depthwise convolution with stride 2 is applied to the input feature map $\mathbf{X}_k$, capturing spatial correlations within each channel independently while significantly reducing computational cost compared to standard convolutions. Following that, a single stride 1×1 pointwise convolution is used to combine the outputs across channels, allowing for feature interaction and increasing the depth of the feature map. In addition, 2×2 pooling operations with stride 2 are applied at each depth level k. The pooled features from the previous layer are also concatenated with the output of the pointwise convolution, enhancing the feature representation by merging hierarchical features from different scales. This enables the network to capture intricate patterns by combining information in different resolutions, which is essential to interpret how local features contribute to the overall risk of flooding.

We used residual connections to maintain key spatial data and increase network depth. Incorporating a modified $\mathbf{X}_k$ into the concatenated output mitigates gradient vanishing and improves identity mapping learning. These connections preserve crucial features of the early layer and streamline network training.

This process is repeated for each FE block in the encoder, leading to a gradual reduction in the spatial dimensions of the feature maps. At each FE block k, the spatial dimensions are reduced by factor 2,

so the resulting feature maps have dimensions $\frac{H}{2^k} \times \frac{W}{2^k} \times F$, where F is the number of channels after concatenation. Such a progressive decrease in the spatial dimensions allows the network to capture more significant receptive fields, collecting information from more extensive regions of the input grid, which is critical for the simulated spread of inundation under various SLR situations. The input grid $\mathbf{I}$, by the end of the encoder stage, transforms into dense feature maps $\mathbf{X}_{\text{enc}} \in \mathbb{R}^{H' \times W' \times F}$ that capture both local and global data patterns, serving as input to the next stage.

B.1.2 Bottleneck Stage

The bottleneck stage of the CASPIAN-v2 comprises a novel block called the *multi-attention ResNeXt* (MARX) block to enhance the ability of the model to focus on the most informative parts of the data. It integrates ResNeXt blocks [28] with the *convolutional block attention module* (CBAM) [29]. The output feature maps from the encoder ($\mathbf{X}_{\text{enc}} \in \mathbb{R}^{H' \times W' \times F}$) serve as the input to the bottleneck stage. The MARX Blocks process feature maps through a sequence of operations, starting with a ResNeXt block, followed by a CBAM module, and concluding with an additional ResNeXt block. In the first ResNeXt block, the input feature map is divided into G groups, and group-specific convolutions are applied using 1×1 and 3×3 kernels, as expressed in Eq. (1):

$$\mathbf{X}_{\text{R1}} = \sigma\left(\mathbf{X}_{\text{enc}} + \sigma\left(\sum_{g=1}^{G} \mathbf{W}_g * \mathbf{X}_{\text{enc},g} + \mathbf{b}_g\right)\right), \tag{1}$$

where $\mathbf{X}_{\text{enc}}$ is the input feature map, $\mathbf{W}_g$ and $\mathbf{b}_g$ are the weights and biases for the g-th group, $\circledast$ denotes the grouped convolution operation, and σ is the activation function. $\mathbf{X}_{\text{R1}} \in \frac{H}{2^K} \times \frac{W}{2^K} \times F_g$ is the output feature map of the first ResNeXt block, where K is the total depth of the encoder stage, and F_g is the number of channels determined by multiplying the cardinality $\mathcal{C}$ with the bottleneck width $\mathcal{B}$ ($F_g = \mathcal{C} \times \mathcal{B}$).

Next, to refine the feature maps and enable the model to focus on the most informative aspects of the data, we integrated the CBAM principle within the MARX block. The CBAM enhances the representational power of the model by sequentially applying attention mechanisms along both the channel and spatial dimensions. In the channel attention module, inter-channel relationships are captured by computing a channel attention map $\mathbf{M}_{\text{c}} \in \mathbb{R}^{1 \times 1 \times F}$, which reweights the channels of the feature map, as expressed in Eq. (2):

$$\mathbf{M}_{\text{c}} = \delta\left(\mathbf{W}_{\text{c2}} \cdot \sigma\left(\mathbf{W}_{\text{c1}} \cdot \mathbf{z}\right)\right), \tag{2}$$

where $\mathbf{W}_{\text{c1}} \in \mathbb{R}^{F^r \times F}$ and $\mathbf{W}_{\text{c2}} \in \mathbb{R}^{F \times F^r}$ are the weights of the fully connected layers, F^r is a reduction ratio parameter, σ denotes the activation function, and δ is the sigmoid function. The aggregated channel descriptor $\mathbf{z} \in \mathbb{R}^F$ is obtained by applying global average pooling over the spatial dimensions.

$$\mathbf{z}_f = \frac{1}{H'W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} \mathbf{X}_{\text{R1},f}(i,j), \tag{3}$$

where $\mathbf{X}_{\text{R1},f}$ is the f-th channel of the feature map $\mathbf{X}_{\text{R1}}$. The channel attention map $\mathbf{M}_{\text{c}}$ is then applied to the feature map via element-wise multiplication ($\mathbf{X}'_{\text{R1}} = \mathbf{M}_c \odot \mathbf{X}_{\text{R1}}$). This operation emphasizes channels that are more informative for predicting inundation patterns influenced by SLR and protection measures. Following that, a spatial attention map $\mathbf{M}_{\text{s}} \in \mathbb{R}^{H' \times W' \times 1}$ is computed by initially aggregating the feature map across the channel dimension using the average pooling, as expressed in Eq. (4):

$$\mathbf{q}(i,j) = \frac{1}{F} \sum_{f=1}^{F} \mathbf{X}'_{\text{R1}}(i,j,f), \tag{4}$$

where $\mathbf{q} \in \mathbb{R}^{H' \times W'}$ are the aggregated feature maps. Afterward, a 7×7 convolution is performed to extract intricate inundation patterns. Next, the spatial attention map $\mathbf{M}_{\text{s}}$ is applied to the refined channel feature map ($\mathbf{X}_{\text{cbam}} = \mathbf{M}_s \odot \mathbf{X}'_{\text{R1}}$), allowing the model to concentrate on the spatial regions that are most pertinent for predicting flood inundation, such as areas with high vulnerability due to

low elevation or insufficient protection. Finally, the output of the CBAM module, $\mathbf{X}_{\text{cbam}}$, is processed through a second ResNeXt block to better capture the representations of complex features.

$$\mathbf{X}_{\text{R2}} = \sigma\left(\mathbf{X}_{\text{cbam}} + \sigma\left(\sum_{g=1}^{G} \mathbf{W}'_g * \mathbf{X}_{\text{cbam},g} + \mathbf{b}'_g\right)\right), \tag{5}$$

where $\mathbf{X}_{\text{R2}}$ is the output feature map of the MARX Block, and $\mathbf{W}'_g$, $\mathbf{b}'_g$ are the weights and biases for the g-th group in the second ResNeXt block. The MARX blocks allow the CASPIAN-v2 model to generalize across complex datasets by adaptively concentrating on the most informative features in both channel and spatial dimensions. By the end of the bottleneck stage, the feature maps ($\mathbf{X}_{\text{bn}} \in \mathbb{R}^{H' \times W' \times F}$) are transformed into rich, high-level representations that capture key information about the candidate input scenario. These refined features serve as a strong foundation for the decoder stage, where they are progressively upsampled and combined with the SLR scalar $\mathcal{S}$ to reconstruct the spatial resolution of the input grid.

B.1.3 Decoder Stage

The decoder stage in the proposed CASPIAN-v2 model employs a series of *feature reconstruction* (FR) blocks to progressively upsample the feature maps. After the bottleneck, the refined feature maps $\mathbf{X}_{\text{bn}} \in \mathbb{R}^{H' \times W' \times F}$ serve as input to the decoder. The main goal is to restore the near-original spatial dimensions $H \times W$. At each depth level k, the decoder up-samples feature maps by a factor of 2 through transpose convolution, producing $\mathbf{X}_{\text{dec},k} \in \mathbb{R}^{\frac{H'}{2^{k-1}} \times \frac{W'}{2^{k-1}} \times F}$. Upsampled maps are then concatenated to corresponding encoder outputs through skip connections, ensuring critical spatial details lost during downsampling are maintained.

To further strengthen the focus of the decoder on critical spatial regions and reflect SLR effects, we propose a novel *SLR-enhanced encoding* (SEE) block. It learns dynamic weighting from the SLR input to adjust decoder features. In the SEE block, the pooled feature maps of each encoder level are aggregated and passed through dense layers to generate spatial weighting coefficients, as expressed in Eq. (6).

$$\mathbf{w}_{\text{sp},k} = \sigma\left(\mathbf{W}_{\text{sp},k}\, \mathcal{F}\left(\mathcal{AP}\left(\mathbf{X}_{\text{enc},k}\right)\right) + \mathbf{b}_{\text{sp},k}\right), \tag{6}$$

where $\mathbf{X}_{\text{enc},k}$ is the k-th channel of the encoder feature map, $\mathcal{AP}$ denotes average pooling, $\mathcal{F}$ denotes the flattening operation, $\mathbf{W}_{\text{sp},k}$ is weight matrix, and $\mathbf{b}_{\text{sp},k}$ is bias term. Simultaneously, the SLR scalar input $\mathcal{S}$ is processed through a dense layer to produce $\mathbf{w}_{\text{SLR}}$. The spatial and SLR features are then concatenated to form $\mathbf{w}_{\text{comb}} = [\mathbf{w}_{\text{sp},k}; \mathbf{w}_{\text{slr}}]$, which is passed through another dense layer with sigmoid activation to produce the final weighting coefficients $\mathbf{w}_{\text{see},k}$. These weighting coefficients are then reshaped and applied to the decoder feature map via element-wise multiplication $\mathbf{X}'_{\text{dec},k} = \mathbf{X}_{\text{dec},k} \odot \mathbf{w}_{\text{see},k}$, where $\mathbf{X}_{\text{dec},k}$ is the decoder feature map at the corresponding depth level, and $\odot$ denotes element-wise multiplication. This configuration of the SEE Block allows the model to adaptively weigh the decoder features based on both spatial information from the encoder and the global influence of SLR, enhancing the model's ability to predict flood inundation patterns under varying SLR scenarios.

At the output of the final FR block, a convolutional operation is applied to produce a preliminary output grid $\mathbf{O}_{\text{conv}} \in \mathbb{R}^{H \times W \times 1}$. To further enhance this output, the model computes the sum of the features across the channels of the last decoder feature map $\mathbf{X}'_{\text{dec},K} \in \mathbb{R}^{H \times W \times F}$. Moreover, the SLR input $\mathcal{S}$ is again incorporated at this stage by processing through a dense layer and then applying to the summed features via element-wise multiplication, as expressed in Eq. (7):

$$\mathbf{X}_{\text{sum}} = \left(\sum_{f=1}^{F} \mathbf{X}'^{(f)}_{\text{dec},K} \odot \mathbf{w}_{\text{slr}}\right), \tag{7}$$

where $\mathbf{X}'^{(f)}_{\text{dec},K}$ is the f-th channel of the feature map. Finally, the enhanced summed features are added to the preliminary output grid before applying the activation function.

$$\mathbf{O} = \sigma\left(\mathbf{O}_{\text{conv}} + \mathbf{X}_{\text{sum}}\right), \tag{8}$$

where $\mathbf{O} \in \mathbb{R}^{H \times W \times 1}$ is the final output grid representing the predicted flood inundation map, and σ is the activation function. This allows extra information from the decoder feature maps by summing

across the channels, thereby enriching the final output with more comprehensive spatial features. The grid **O** reflects the likelihood or extent of flooding at each spatial point, considering both the local features learned by the encoder and the broader SLR effects used in the decoder. This integrated design helps the CASPIAN-v2 model generate accurate and robust flood inundation maps, which are crucial for planning and mitigating coastal regions impacted by SLR.

B.2 Loss Function

Predicting PWL under different SLR scenarios is challenging due to outliers and the need to balance error sensitivity across multiple regions. To tackle these issues, we introduce a hybrid loss function that combines Huber [30], Log-Cosh [31], and Quantile [32] losses in a weighted setup. The Huber loss L_h aims to robustly minimize small prediction errors while limiting the impact of outliers, and it uses a threshold δ to manage the sensitivity of the error. The L_h for each sample i is computed as expressed in Eq. (9):

$$L_{h,i} = \begin{cases} \frac{1}{2}(y_{\text{p},i} - y_{\text{t},i})^2 & \text{if } |y_{\text{p},i} - y_{\text{t},i}| \leq \delta, \\ \delta \cdot |y_{\text{p},i} - y_{\text{t},i}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{9}$$

where $y_{\text{t},i}$ and $y_{\text{p},i}$ represent the actual and estimated PWL values. We set δ within the range of 0.3 and 0.7, which is dynamically determined to balance sensitivity and robustness. Moreover, we integrate Log-Cosh loss (L_{cosh}) to smooth gradients in regions with large variations, helping to maintain prediction stability in different areas affected by SLR. The L_{cosh} is expressed as in Eq. (10):

$$L_{\text{cosh},i} = \log\left(\cosh(y_{\text{p},i} - y_{\text{t},i})\right), \tag{10}$$

In addition, the quantile loss L_q differentiates errors by assigning distinct penalties to underestimation and overestimation, dictated by a quantile parameter $\tau = 0.75$. This loss dynamically adjusts to minimize quantile-specific errors, calculated as in Eq. (11):

$$L_{q,i} = \begin{cases} \tau \cdot (y_{\text{p},i} - y_{\text{t},i}) & \text{if } y_{\text{p},i} \geq y_{\text{t},i}, \\ (1-\tau) \cdot (y_{\text{t},i} - y_{\text{p},i}) & \text{otherwise} \end{cases} \tag{11}$$

To achieve an optimal balance, we linearly combine the three loss components into a comprehensive hybrid loss function L_{total}, weighted by empirically tuned coefficients $\alpha_h, \alpha_c, \alpha_q$. The final loss is expressed as in Eq. (12):

$$L_{\text{custom}} = \alpha_h \cdot L_h + \alpha_c \cdot L_{\text{cosh}} + \alpha_q \cdot L_q, \tag{12}$$

where $\alpha_h, \alpha_c, \alpha_q \geq 0$ and $\alpha_h + \alpha_c + \alpha_q = 1$. These weights are empirically determined to optimize predictive performance. By integrating these components, our custom hybrid loss function balances error sensitivity, maintains robustness to outliers, and addresses asymmetric error distributions, enhancing the model's predictive accuracy for PWL under varying SLR scenarios.

C Details of Experimental Setup

This section outlines the parameters employed to train, validate, and evaluate the proposed DL model. We detail the CASPIAN-v2 optimization and training protocol, baseline models, and evaluation metrics used to validate and compare the performance of the CASPIAN-v2 model.

C.1 Model Optimization and Training Protocol

The CASPIAN-v2 model was implemented in Python 3.10 using TensorFlow 2.10.1 and was trained on a 64-bit Windows operating system. We utilized an Intel Core i9-14900K (3.20 GHz) machine with 64 GB of RAM and an NVIDIA GeForce RTX 4090 GPU. The CASPIAN-v2 architecture

(shown in Figure 4) was refined through extensive ablation studies, which evaluated the impact of loss functions, MARX blocks, the SEE block, and SLR integration. The model was trained using the proposed hybrid loss function. The training process lasted for 200 epochs with a batch size of 2. The remaining hyperparameters were fine-tuned using Bayesian Optimization and Random Search to ensure optimal performance.

C.2 Baseline Models

We assessed the performance of CASPIAN-v2 model for coastal flood prediction against several SOTA ML and DL techniques. We considered conventional ML methods, including the Naïve model, which utilizes a dummy regressor to forecast the mean value of the target variable to serve as a basic reference for assessing more advanced models. Additionally, we trained random forest, linear regression, extreme gradient boosting, support vector regression, lasso regression with polynomial features, and kriging with principal component analysis to establish an ML benchmark. The hyperparameters for training these models were optimized through a combination of Bayesian optimization and random search methods, allowing for efficient exploration of the parameter space while preventing overfitting on the validation set.

In addition to traditional ML baselines, we tested several DL models adapted to the flood prediction task. These include a simple feed-forward neural network architecture, specifically a multi-layer perceptron (MLP), and compact convolutional transformers (CCT) [33], which serve as baseline 1D DL models. Furthermore, we evaluated several 2D DL models, including Attention-Unet [34], and Swin-Unet [35]. To adapt these models for flood prediction, we replaced their segmentation heads with a 1×1 convolution layer followed by activation to output real-valued flood depth predictions. We evaluated two versions of Attention-Unet: one with randomly initialized weights and another (denoted as Atten-Unet*) with an encoder pre-trained on ImageNet [36], leveraging transfer learning to improve performance in low-data scenarios. The final DL baseline was CASPIAN, which we previously proposed in [11]. All DL models were trained using the Adam optimizer and the proposed hybrid loss function (L_{custom}). Additionally, each model was trained for 200 epochs with a batch size of 2, and early stopping based on validation loss. The remaining training hyperparameters for each model were tuned using Bayesian Optimization and Random Search with the Keras Tuner to ensure a fair comparison.

C.3 Evaluation Metrics

To evaluate the performance of our model in predicting PWL values, we employ several metrics that capture various aspects of predictive accuracy and robustness.

- Average relative total absolute error (ARTAE): It quantifies the relative error between the predicted y_{p}^k and true values y_{t}^k by measuring the normalized L_1 difference:

$$\text{ARTAE} \triangleq \frac{1}{N} \sum_{k=1}^{N} \frac{\|y_{\text{t}}^k - y_{\text{p}}^k\|_1}{\|y_{\text{t}}^k\|_1} \tag{13}$$

  where N denotes the total data samples.

- Average root mean square error (ARMSE): It captures the root mean square error for each sample, as expressed:

$$\text{ARMSE} \triangleq \frac{1}{N} \sum_{k=1}^{N} \sqrt{\frac{1}{d_y} \sum_{i=1}^{d_y} \frac{(y_{\text{t},i}^k - y_{\text{p},i}^k)^2}{d_y}} \tag{14}$$

  where d_y indicates the dimensionality of each sample.

- Average mean absolute error (AMAE): It provides an average absolute error measure over samples. Unlike ARMSE, it does not severely penalize larger deviations. The AMAE is calculated as:

$$\text{AMAE} \triangleq \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{d_y} \frac{|y_{\text{t},i}^k - y_{\text{p},i}^k|}{d_y} \tag{15}$$

- Coefficient of determination (R^2): The R^2 measures the proportion of variance explained by the model, indicating how well the predictions approximate the true values. It is computed as:

$$R^2 \triangleq \frac{1}{N}\sum_{k=1}^{N}\left(1-\frac{\sum_{i=1}^{d_y}(y_{\mathrm{t},i}^k-y_{\mathrm{p},i}^k)^2}{\sum_{i=1}^{d_y}(y_{\mathrm{t},i}^k-\bar{y}_{\mathrm{t}}^k)^2}\right) \tag{16}$$

  where $\bar{y}_{\mathrm{t}}^k$ is the mean of the true values for the k-th sample.

- Threshold exceedance metric ($\delta > \Delta$): This metric represents the fraction of cases where the prediction error exceeds a specified threshold Δ, which is crucial for applications sensitive to large errors. It is defined as:

$$\delta > \Delta \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{\left|\{i : |y_{\mathrm{t},i}^k - y_{\mathrm{p},i}^k| > \Delta, i \in [d_y]\}\right|}{d_y} \tag{17}$$

- Non-inundated prediction accuracy (Acc[0]): The Acc[0] measures the fraction of instances where the true values are zero (non-inundated points), and the predictions accurately reflect this. This is particularly relevant for sparse targets. It is computed as:

$$\mathrm{Acc}[0] \triangleq \frac{1}{N}\sum_{k=1}^{N}\frac{\left|\{i : y_{\mathrm{t},i}^k = 0, i \in [d_y]\}\right|}{d_y} \tag{18}$$

D Details of the Results and Comparison

In this section, we evaluate the performance of CASPIAN-v2 model through quantitative and qualitative analyses.

D.1 Quantitative Results

D.1.1 Performance Metrics on Test Set

We first report the performance of CASPIAN-v2 on the test set, as shown in Table 3. For AD data, the model achieves an AMAE of 0.0586, ARMSE of 0.4079, and a high average R^2 score of 0.9556, indicating excellent explanatory power. The ARTAE of 4.2793% and low error percentages ($\delta > 0.5\%$: 1.02% and $\delta > 0.1\%$: 4.37%) highlight higher precision in accurately predicting flood inundation levels. Similarly for SF, the model achieves an AMAE of 0.0320, ARMSE of 0.2094, and an average R^2 score of 0.9214. While the ARTAE is higher at 8.8129%, the model maintains high accuracy metrics with an Acc[0] of 99.76% compared to 99.04% in AD.

On the combined dataset, CASPIAN-v2 performs consistently well with an AMAE of 0.0453, ARMSE of 0.3087, and an average R^2 score of 0.9385. The combined ARTAE of 6.5461% and low error percentages ($\delta > 0.5$: 0.89% and $\delta > 0.1$: 3.55%) demonstrate balanced performance across regions. The high Acc[0] of 99.39% further underscores the reliability of the model in accurately predicting coastal inundation.

Table 3: Evaluation of CASPIAN-v2 on test set. ↓ indicates that lower values are better, and ↑ indicates that higher values are better.

Dataset	MAE ↓	RMSE ↓	RTAE ↓	$\delta > 0.5$ ↓	$\delta > 0.1$ ↓	R^2 Score ↑	Acc[0] ↑
AD	0.0586	0.4079	4.2793	1.02%	4.37%	0.9556	99.04%
SF	0.0320	0.2094	8.8129	0.75%	2.72%	0.9214	99.76%
Combined	0.0453	0.3087	6.5461	0.89%	3.55%	0.9385	99.39%

D.1.2 Performance Metrics on Holdout Set

In this section, we present CASPIAN-v2 performance on the holdout set. The results are reported in Table 4, where it can be observed that the model achieves an AMAE of 0.0792, an ARMSE of 0.4871, and an average R^2 score of 0.9525 for AD. Furthermore, the small percentages of errors ($\delta > 0.5$: 1.29% and $\delta > 0.1$: 5.48%) underscore its accuracy in predicting flood inundation levels.

Similarly, for SF, CASPIAN-v2 achieves an AMAE of 0.0317, an ARMSE of 0.2259, and an average R^2 score of 0.9694. Compared to AD, the ARTAE of 4.0009% indicates slightly more predictions that have larger relative errors. However, with Acc[0] of 99.64%, the model achieves better non-inundated prediction accuracy compared to 99.07% in AD-Holdout.

Overall, CASPIAN-v2 achieves an AMAE of 0.0512, an ARMSE of 0.3331, and an average R^2 score of 0.9625 on the aggregated holdout dataset. The ARTAE of 3.7167% and small error percentages ($\delta > 0.5$: 1.04% and $\delta > 0.1$: 4.17%) signify consistent performance in both regions. The higher Acc[0] of 99.41% further confirms its reliability in predicting flood inundation across diverse and challenging shoreline scenarios.

Table 4: Evaluation of CASPIAN-v2 on holdout set.

Dataset	**MAE** $\downarrow$	**RMSE** $\downarrow$	**RTAE** $\downarrow$	$\delta > 0.5 \downarrow$	$\delta > 0.1 \downarrow$	$\mathbf{R^2}$ **Score** $\uparrow$	**Acc[0]** $\uparrow$
AD - Holdout	0.0792	0.4871	3.3081	1.29%	5.48%	0.9525	99.07%
SF - Holdout	0.0317	0.2259	4.0009	0.86%	3.26%	0.9694	99.64%
Combined	0.0512	0.3331	3.7167	1.04%	4.17%	0.9625	99.41%

D.1.3 Performance Benchmarking against SOTA Methods

To comprehensively evaluate the performance of CASPIAN-v2, we benchmarked it against a suite of SOTA traditional ML and DL models. The selection and implementation details for these baseline models are described in Section C.2. This section presents a detailed comparison the prediction performance across all models, with the full results presented in Table 5. The analysis is broken down by model class, first comparing against traditional ML methods, and then against other DL architectures.

Comparison with Machine Learning Models:

In this section, we compare the performance of CASPIAN-v2 against various traditional ML models for flood prediction, as shown in Table 5. The Naïve model shows high errors with an AMAE of 1.5343, ARMSE of 3.5444, and an average R^2 score of 0.5450. Among traditional approaches, linear regression reduces errors significantly, achieving an AMAE of 0.1272, ARMSE of 0.1946, and an average R^2 score of 0.9464. The lasso with polynomial model further improves performance, giving an AMAE of 0.0937, ARMSE of 0.1202, and the highest average R^2 score of 0.9618 among traditional ML models.

Compared to the best traditional model (lasso with polynomial), CASPIAN-v2 reduces the AMAE by 51.65% (from 0.0937 to 0.0453). However, CASPIAN-v2 has a higher ARMSE of 0.3087 compared to 0.1202, indicating it minimizes mean errors effectively but may experience larger individual prediction errors. Despite this, CASPIAN-v2 outperforms traditional models across multiple metrics, leveraging DL and multi-dimensional data integration to achieve superior accuracy in flood prediction.

This trend is even more pronounced in the spatial accuracy results. While the lasso model achieved a DSC of 0.6438, CASPIAN-v2 scored 0.8437, representing a 31.05% improvement. This significant gap underscores the inherent limitations of traditional ML models in capturing the complex geometric shape of flood events, a task for which our deep learning architecture is better suited.

Comparison with Deep Learning Models:

Existing 1D and 2D DL models show varied performance, as reported in Table 5. The CCT model achieves an AMAE of 0.9064, an ARMSE of 2.3292, and an average R^2 score of 0.6649, indicating moderate predictive capabilities. Atten-Unet and its variant Atten-Unet* improve performance with AMAE values of 0.1061 and 0.1032 and average R^2 scores of 0.9195 and 0.9210, respectively. Swin-Unet achieves further improvements, reducing the AMAE to 0.0629 and attaining an average R^2 score of 0.9514, reflecting its effectiveness in capturing spatial dependencies.

Compared to the second-best DL model, CASPIAN-v2 reduces the AMAE by 19.96% (from 0.0566 to 0.0453) and achieves an exceptional average Acc[0] of 99.39%, surpassing CASPIAN's 98.84%. These results highlight superior accuracy and robust generalization capabilities of CASPIAN-v2.

In terms of spatial fitness, CASPIAN-v2 (with DSC of 0.8437) also demonstrates a clear advantage over the best-performing DL baseline, CASPIAN (0.8261), representing a 2.13% improvement in

Table 5: A comprehensive performance comparison between our proposed CASPIAN-v2 and state-of-the-art models, grouped into a baseline physics-based simulator (Delft3D), traditional ML, and DL approaches. Prediction accuracy is evaluated using eight standard metrics, where arrows indicate the desired direction (↑ for higher is better, ↓ for lower is better). Computational efficiency is assessed by three key indicators: the total number of trainable parameters (M = millions), the total training time (TT), and the average inference time (IT) per sample. The physics-based simulations, which provide the ground truth data, are included for reference. The top-performing result for each metric is highlighted in red, and the second-best is highlighted in blue.

Type	Model	Prediction Accuracy								Computational Efficiency		
		MAE ↓	RMSE ↓	RTAE ↓	$\delta > 0.5$ ↓	$\delta > 0.1$ ↓	R^2 ↑	Acc[0] ↑	DSC ↑	Param↓	TT ↓	IT ↓
Simulator	AD Pipeline†	*Served as the ground truth*								-	-	71–73h
	SF Pipeline°									-	-	3.5–6.0h
ML (1-D)	Naïve	1.53	3.54	1746.06	74.92%	80.11%	0.54	31.01%	0.38	-	62s	0.15s
	RF	0.54	0.73	264.95	36.77%	72.20%	0.79	34.19%	0.41	-	75s	0.18s
	Linear	0.12	0.19	64.98	7.87%	14.03%	0.94	59.28%	0.62	-	65s	0.16s
	XGBoost	0.25	0.24	164.16	16.27%	49.88%	0.93	44.10%	0.47	-	198s	0.21s
	SVR	0.20	0.24	72.31	9.24%	41.17%	0.92	45.46	0.48	-	79s	0.19s
	Lasso Poly	0.09	0.12	28.15	4.47%	15.04%	0.96	55.78%	0.64	-	72s	0.17s
	Kriging	0.10	0.24	39.90	5.22%	11.59%	0.94	62.88%	0.63	-	76s	0.18s
DL (1-D)	MLP	0.64	2.72	524.17	32.82%	41.94%	0.65	36.91%	0.43	0.01M	14h	5.03s
	CCT	0.90	2.32	843.54	48.08%	64.63%	0.66	34.01%	0.42	11.05M	18h	0.26s
DL (2-D)	Atten-Unet	0.10	0.37	11.82	3.14%	16.70%	0.91	95.26%	0.73	12.07M	46h	0.24s
	Atten-Unet*	0.10	0.36	11.65	3.31%	15.62%	0.92	94.99%	0.74	12.07M	47h	0.27s
	Swin-Unet	0.06	0.27	6.72	1.47%	12.94%	0.95	98.10%	0.80	8.29M	26h	0.24s
	CASPIAN	0.05	0.36	5.85	1.01%	4.79%	0.92	98.84%	0.82	0.36M	22h	0.22s
	Ours	0.04	0.30	6.54	0.89%	3.55%	0.93	99.39%	0.84	0.38M	22h	0.22s

* with pre-trained encoder on ImageNet [36].
† AD pipeline includes Delft3D + SWAN + post processing.
° AD pipeline includes Delft3D + post processing.

spatial accuracy. Taken together, these results highlight the superior accuracy and robust generalization capabilities of CASPIAN-v2. The integration of advanced components such as the MARX and SEE blocks, combined with an optimized Hybrid loss function, enables the effective modeling of complex flood dynamics.

D.1.4 Computational Efficiency Analysis

A primary motivation for this research is to overcome the significant computational burden of physics-based hydrodynamic simulators. The final three columns of Table 5 provide a comprehensive comparison of the computational efficiency of all evaluated models.

As expected, the traditional ML models are the fastest to train, typically requiring only a few minutes. However, this speed comes at the cost of significantly lower prediction accuracy. Among the more accurate DL models, CASPIAN-v2 demonstrates a highly favorable balance of performance and efficiency. With only 0.38 million parameters, it is one of the most lightweight 2D models, comparable in size to the original CASPIAN (0.36M) and substantially smaller than transformer-based models like Swin-Unet (8.29M) or other U-Net variants (12.07M). Its training time (22 hours) and inference time (0.22s per scenario) are also highly competitive within this high-performing group.

The most critical comparison, however, is against the physics-based simulator. Generating a single flood scenario is an exceptionally demanding task. For Abu Dhabi, a full simulation requires 71 to 73 hours of elapsed runtime on high-performance computing infrastructure due to the coupling of Delft3D and SWAN models and extensive post-processing. For San Francisco Bay, where the simulation was less complex, the process still required a substantial 3.5 to 6.0 hours (as detailed in Section A.1). Extrapolating these figures, simulating our full test set of 72 scenarios (36 for each region) would demand approximately 2,763 hours (nearly 115 days) of continuous computation. In stark contrast, CASPIAN-v2 can predict the outcomes for all 72 scenarios in just under 16 seconds. This represents a monumental reduction in computational time, transforming a months-long endeavor

into a near-instantaneous task and positioning CASPIAN-v2 as a practical and scalable tool for real-world coastal planning.

Table 6: CASPIAN-v2 generalizability evaluation using different SLR data.

Dataset *(SLR)*	**MAE** $\downarrow$	**RMSE** $\downarrow$	**RTAE** $\downarrow$	$\delta > 0.5 \downarrow$	$\delta > 0.1 \downarrow$	**R^2 Score** $\uparrow$	**Acc[0]** $\uparrow$
SF - Generalizability *(0.5 m)*	0.0626	0.2996	6.4240	1.89%	7.79%	0.9336	97.99%
SF- Generalizability *(1.5 m)*	0.1005	0.4565	4.3961	1.97%	14.51%	0.9196	98.23%
AD - Holdout *(0.5 m)*	0.0567	0.2274	2.5225	0.53%	17.87%	0.9901	99.18%
SF - Holdout *(1.0 m)*	0.0433	0.2318	4.6277	0.79%	9.61%	0.9685	99.34%
Overall	0.0652	0.3040	4.5871	1.31%	12.07%	0.9520	98.69%

D.1.5 Numerical Assessment of Generalizability

This section reports the generalization performance of CASPIAN-v2 on unseen data. The model was fine-tuned using new SF data corresponding to 0.5 m and 1.5 m SLR depths, encompassing 30 protection scenarios where one OLU was protected at a time (more details in Supplementary Material Section S5). For evaluation, 20% of the data (6 samples) was reserved, while the remaining 80% (24 samples) was used for fine-tuning and validation. Fine-tuning spanned 100 epochs with a progressive gradual recall approach, mixing the new data with the AD and SF holdout data in a 20:80 test/train ratio. The training set began with 70% of the AD and SF holdout set combined with 30% of the new data, gradually increasing to 70% by the end of training.

The results in Table 6 demonstrate strong generalization by CASPIAN-v2 across SLR scenarios. For SF 0.5 m data, the model achieved an AMAE of 0.0626, ARMSE of 0.2996, and average R^2 score of 0.9336. An ARTAE of 6.4240% and low error percentages ($\delta > 0.5$: 1.89% and $\delta > 0.1$: 7.79%) highlight its precision. For SF 1.5 m data, the model showed slightly suboptimal performance with an AMAE of 0.1005, ARMSE of 0.4565, and average R^2 score of 0.9196. The ARTAE of 4.3961% indicates balanced performance, with an average Acc[0] of 98.23% compared to 97.99% for 0.5 m data.

When retaining existing knowledge, CASPIAN-v2 achieved an AMAE of 0.0567 and ARMSE of 0.2274 on the AD holdout set for 0.5 m SLR, with an average R^2 score of 0.9901. The ARTAE of 2.5225% and low error percentages ($\delta > 0.5$: 0.53% and $\delta > 0.1$: 17.87%) emphasize its precision. For the SF holdout set at 1.0 m SLR, the model achieved an AMAE of 0.0433, ARMSE of 0.2318, and average R^2 score of 0.9685. The ARTAE of 4.6277% and error percentages ($\delta > 0.5$: 0.79% and $\delta > 0.1$: 9.61%) reflect its ability to balance low absolute and relative errors, with an Acc[0] of 99.34%.

Overall, the model achieved an AMAE of 0.0652, an ARMSE of 0.3040, and an average R^2 score of 0.9520, revealing robust generalization abilities of the model across various SLR settings. Further, the model achieved an ARTAE of 4.5871% and low error percentages ($\delta > 0.5\%$: 1.31% and $\delta > 0.1\%$: 12.07%), with a high Acc[0] of 98.69%. These findings highlight the ability of the CASPIAN-v2 model to effectively generalize to new and previously unseen scenarios with minor fine-tuning, making it a reliable tool for real-world inundation prediction.

D.2 Qualitative Results

D.2.1 Visual Performance on Test Set

In this section, we provide a qualitative assessment of the performance of CASPIAN-v2 on the test set. Figure 5 presents two randomly selected scenarios for the AD and SF regions, where it can be observed that the predicted inundation values of the proposed model closely align with the corresponding ground truth values. In single unprotected OLU scenarios (rows 1 and 3), the model accurately captures localized flooding effects, showing sensitivity to minor protection configuration changes. Similarly, CASPIAN-v2 effectively handles the increased complexity of mixed OLU protection statuses (rows 2 and 4). These results highlight the robustness of the model in generalizing across diverse regions and protection patterns. Figure 5(c) shows the absolute error maps, where it can be observed that the CASPIAN-v2 model produced minimal errors, with deviations occurring

mainly in areas with sharp transitions in flood depths. However, these small variations minimally affect the overall prediction accuracy.

To illustrate the local impact of the protection measures on flood dynamics, zoomed-in insets are provided for specific OLUs. For instance, the first inset for AD highlights how inundation patterns are directly controlled by the protection status of the nearest OLU. When OLU-17 is protected, the area behind it remains largely dry, whereas significant flooding occurs inland of the unprotected OLU-14.

D.2.2 Visual Performance on Holdout Set

In this section, we demonstrate the performance of CASPIAN-v2 using a holdout set composed of particularly challenging coastal protection scenarios. Figure 6 showcases the performance of the model on two challenging configurations from the holdout set, which was specifically designed to test generalization across complex protection scenarios. These scenarios feature intricate mixes of protected and unprotected OLUs, creating sharp inundation boundaries where flooded and non-flooded regions meet. CASPIAN-v2 demonstrates high fidelity in these cases, accurately capturing these abrupt changes in local flood behavior. For instance, it correctly captures the inundation dynamics when one side of the SF bay is protected and the other is not (last row of Figure 6).

The strong performance of the model here is particularly noteworthy given that it was trained on only a small subset of the thousands of possible protection combinations (2^n, where n is the number of OLUs). This success on unseen, complex configurations indicates that CASPIAN-v2 is not merely memorizing training data but is learning the underlying spatial logic of how flood defenses influence inundation patterns. This affirms its robustness and reliability for real-world application.

D.2.3 Visual Comparison with SOTA Methods

We qualitatively evaluated the performance of the proposed CASPIAN-v2 by visually comparing its prediction errors with those of key SOTA baselines. Figure 7 presents this analysis for representative scenarios in both Abu Dhabi and San Francisco. Figure 7(b) shows the absolute error map for our proposed CASPIAN-v2 model, demonstrating that errors are generally low and confined to complex hydraulic transition zones. The key insights, however, come from the error difference maps (Figure 7 (c-f)), which directly compare the spatial accuracy of CASPIAN-v2 to each baseline. In these maps, green areas highlight regions where CASPIAN-v2 is more accurate, while red indicates where the baseline had a lower error, and transparent areas denote regions where both models performed similarly.

Compared to the Lasso with polynomial features Figure 7(c) and MLP Figure 7(d) baselines, CASPIAN-v2 offers a dramatic improvement, with vast green areas indicating its superior ability to capture the fundamental flood patterns that these simpler models miss. The comparison with the more advanced Swin-Unet Figure 7(e) and the original CASPIAN Figure 7(f) models is also convincing. While these models are more competitive, the difference maps still show a clear and consistent advantage for CASPIAN-v2, which successfully reduces errors in many of the most deeply inundated and complex areas.

Moreover, Figure 8 visualizes the flood extents predicted by CASPIAN-v2 against the best-performing ML and DL baseline model. The map breaks down the predictions into correctly matched areas (green), over-predicted areas (orange), and under-predicted areas (purple). The visualization reveals that while the baseline models produce a more fragmented prediction with significant patches of both over- and under-prediction, the output of the proposed CASPIAN-v2 model aligns much more closely with the ground truth. Its predicted flood extent is more coherent and captures the true inundation boundaries with far fewer spatial errors. These qualitative comparisons align with the quantitative results in Table 5, highlighting the ability of the proposed model to achieve higher accuracy and visually superior predictions.

D.2.4 Visual Assessment of Generalizability

We next evaluate the generalizability of CASPIAN-v2 under different environmental conditions by fine-tuning the model on two additional SLR data of 0.5 m and 1.5 m. Figure 9 shows the prediction results, illustrating that while the fine-tuned model exhibits some localized discrepancies (Figure 9(c)), these deviations remain modest given the minimal training data and limited fine-tuning epochs. In the 0.5 m SLR scenario, the model yields relatively lower absolute errors in predicting flood

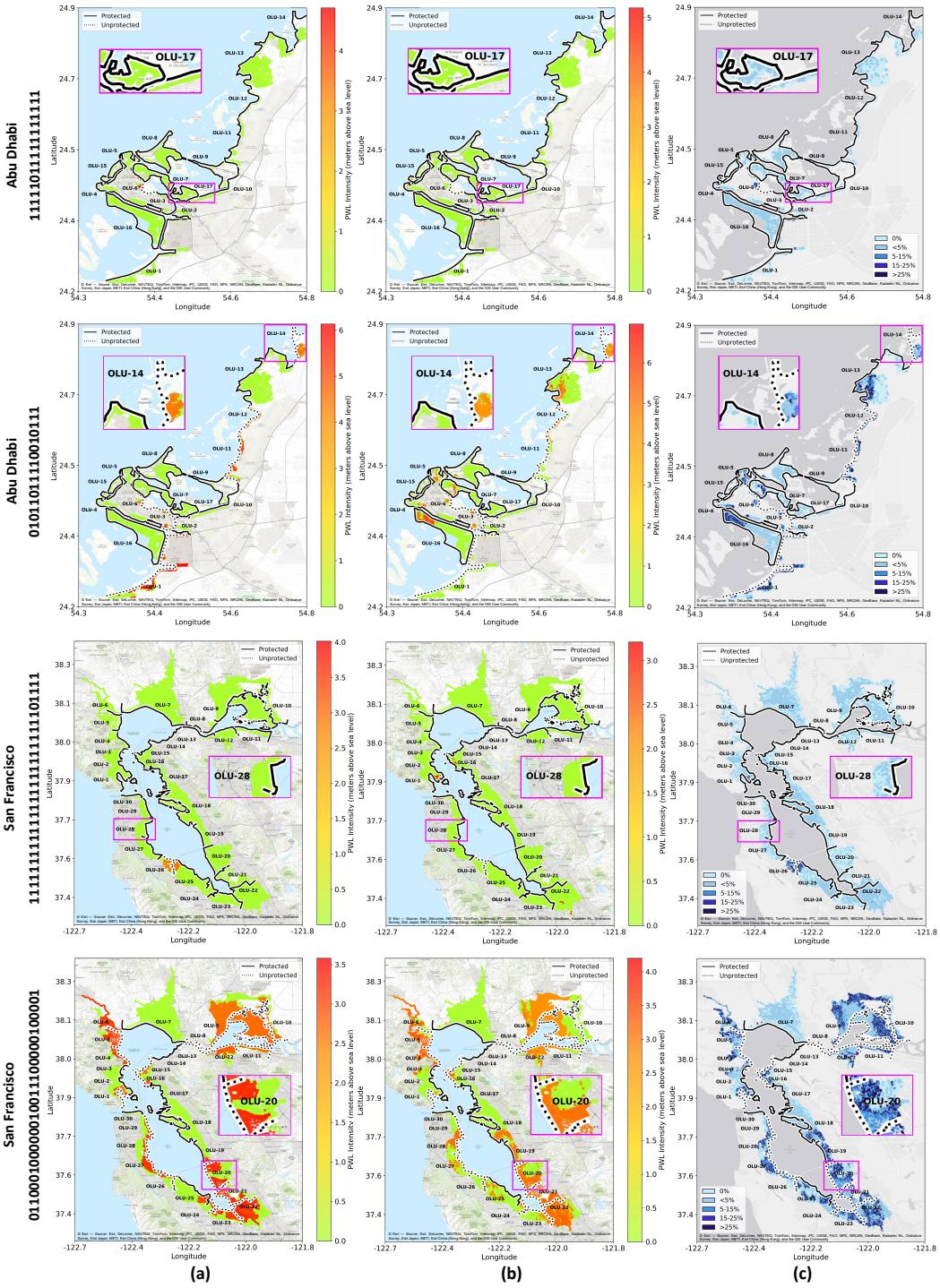


Figure 5: Evaluation of CASPIAN-v2 on the test datasets. (a) Ground truth inundation maps for representative AD and SF scenarios. (b) Predicted inundation values. (c) Absolute error distributions of predicted inundation values. Darker shades of blue indicate higher absolute errors, ranging from near 0% to greater than 25%. The magenta insets provide zoomed-in views of specific OLUs to illustrate the effect of protection measures. For instance, the inundation is shown to be minimal inland of the protected OLU-17 in AD, whereas significant flooding occurs near the unprotected OLU-20, a dynamic that the model precisely captures.

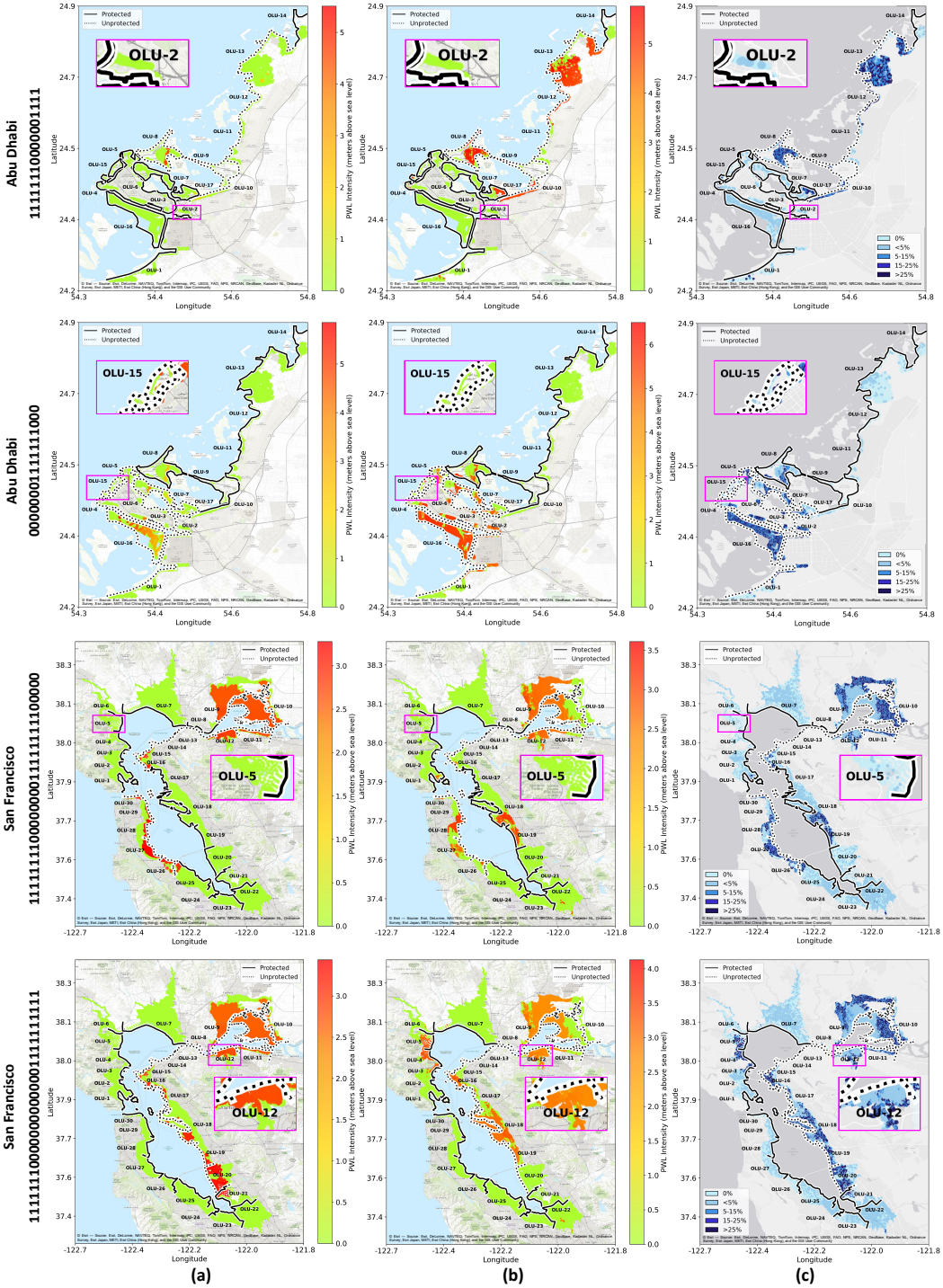


Figure 6: Evaluation of CASPIAN-v2 on the holdout datasets. (a) Ground truth inundation maps for representative AD and SF scenarios. (b) Predicted inundation values. (c) Absolute error distributions of predicted inundation values. Darker shades of blue indicate higher absolute errors, ranging from near 0% to greater than 25%. The zoomed-in insets highlight fine-grained hydrodynamic effects. For instance, the successful prevention of inundation by a protected OLU-2 in AD, versus the widespread inland flooding resulting from an unprotected OLU-12 in SF.

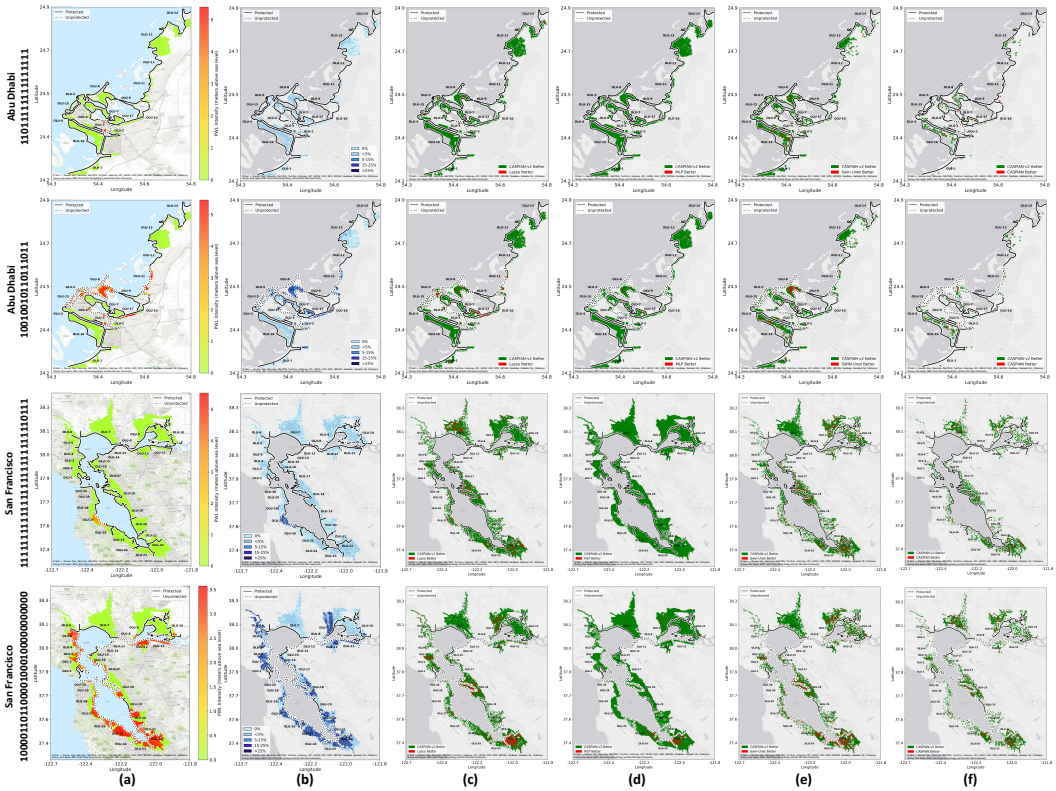


Figure 7: Qualitative comparison of CASPIAN-v2 with SOTA approaches in predicting coastal flood inundation (a) Ground truth inundation maps for representative AD and SF scenarios. (b) Absolute error map for our proposed CASPIAN-v2 model, with darker blue indicating higher error. (c-f) Error difference maps comparing CASPIAN-v2 to key baselines. In these maps, green indicate regions where CASPIAN-v2 is more accurate than the baseline, red areas show where the baseline performed better, and transparent regions denote similar performance. The visualization clearly shows that CASPIAN-v2 provides a substantial improvement over the (c) Lasso, (d) MLP, (e) Swin-Unet, and (f) original CASPIAN models.

extents. By contrast, the 1.5 m scenario exhibits slightly higher errors, likely due to the increased variability in PWL values. Nonetheless, the predictions generally align well with the ground truth inundation patterns.

Overall, these findings underscore adaptability of the proposed model to evolving coastal conditions, suggesting that with sufficient training data and appropriately tuned hyperparameters, the model can maintain robust performance across a broad range of SLR scenarios.

E Holdout Dataset

To thoroughly evaluate the ability of our model to handle challenging conditions, we curated a specialized holdout set for both the AD and SF regions. The scenarios below were chosen based on the spatial configuration and proximity of the OLUs to ensure diverse yet demanding test cases for the model. Tables 7 and 8 list all holdout scenarios for the AD and SF regions, respectively.

F Generalizability Dataset

We further evaluated generalizability of the CASPIAN-v2 model for unseen SLR conditions of 0.5 m and 1.5 m. This dataset contains 32 protection scenarios for the SF region: 30 scenarios each with exactly one protected OLU (and the remaining unprotected), a completely unprotected scenario, and

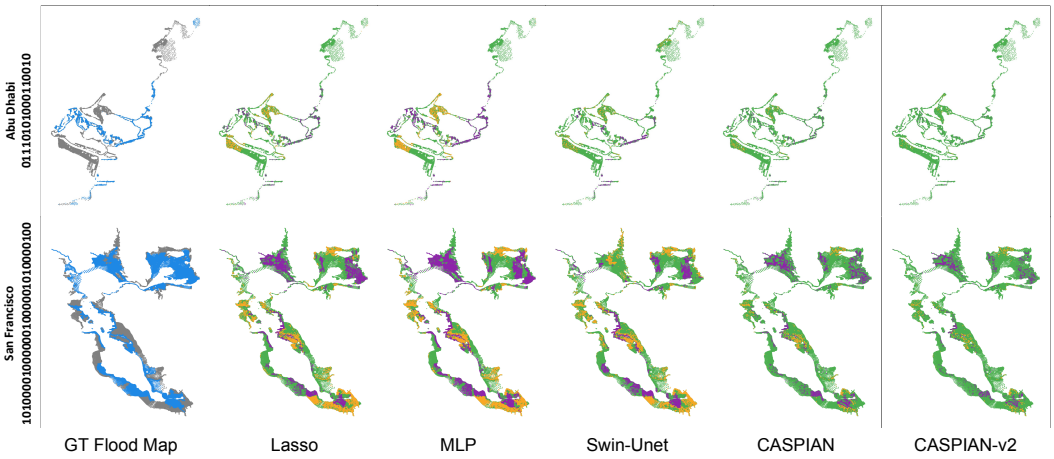


Figure 8: Visual comparison of spatial prediction accuracy for CASPIAN-v2 versus the top-performing baseline model on a representative test case. Green indicates correctly predicted inundated areas (true positives), orange indicates over-prediction (false positives), and purple indicates under-prediction (false negatives). CASPIAN-v2 demonstrates a larger matched area and more coherent flood boundaries.

a fully protected scenario. Table 9 lists these configurations in binary form, where `0` and `1` denote unprotected and protected OLUs, respectively.

G Model Interpretability and Uncertainty Quantification

To ensure CASPIAN-v2 is a trustworthy and practical tool for real-world decision-making, we evaluated two critical aspects beyond standard accuracy metrics: model interpretability and predictive uncertainty.

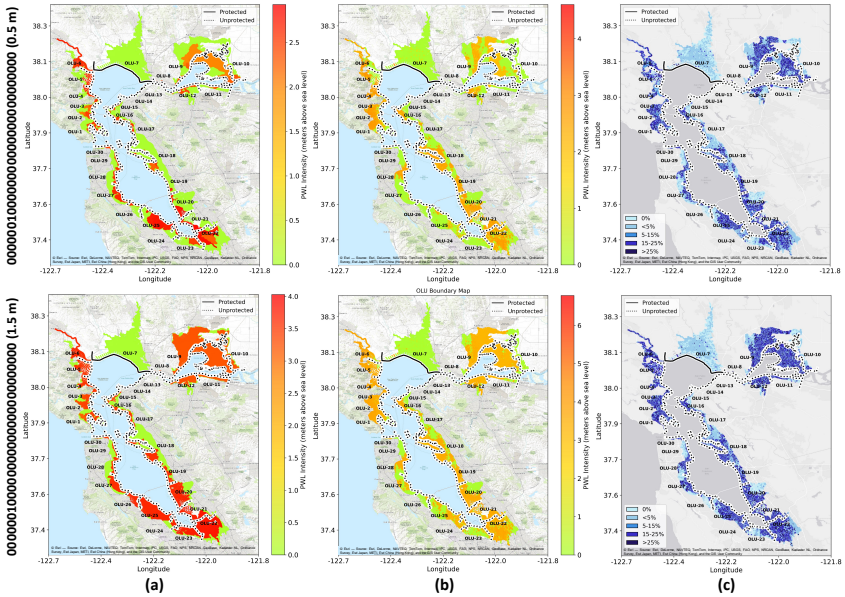


Figure 9: Generalizability evaluation of CASPIAN-v2 fine-tuned for 0.5 m and 1.5 m SLR scenarios. (a) Ground truth inundation maps. (b) Predicted inundation values. (c) Absolute error distributions of predicted inundation values. Darker shades of blue indicate higher absolute errors, ranging from near 0% to greater than 25%.

Table 7: Holdout set scenarios for the AD region. 1 indicates a protected OLU, while 0 denotes an unprotected OLU.

AD Scenarios			
00000001111110000	00000001111111000	00000011111000000	00000011111100000
00000111100000111	00000111110000011	00000111111100000	00001110000111000
00001111000011110	00001111111110000	00011000110001100	00011100011100011
00011111111111000	00110011001100110	00111111111111100	01010101010101010
10101010101010101	11000000000000011	11001100110011001	11100000000000111
11100011100011100	11100111001110011	11110000000001111	11110000111100001
11110001111000111	11111000000011111	11111000001111100	11111000011111000
11111100000011111	11111100000111111	11111110000000111	11111110000001111

Table 8: Holdout set scenarios for the SF region. 1 indicates a protected OLU, while 0 denotes an unprotected OLU.

SF Scenarios		
000001111111111111100000000000	000011111111100001111111010101	000111000111000111000111000111
001000000110101010111001011111	001001100011010100010000110101	001100110011001100110011001100
001100111100101000111010000010	001101101111101000010100001001	001111111010111000001001001100
010011000111110100101010000000	010100000101110111010010 1100001	010101010101010101010101010101
010101100100010101111100010000	011000001010000011110001111000	011000111000100011000001110010
011000111100001111001101001110	011010010000101000111110110100	011011000000011111011000100101
011100010110000001100011001011	011101000010011011111110001010	011101011000011111110101011001
011110110100101000001001101110	011111000111000101011010001001	100000000010100101001101111111
100010101001111100110000100100	100100000111111000001001001110	100101011111010111011101001100
101001101011010011000100100110	101010000010001101100100001010	101010101010101010101010101010
101011000111111100110100001100	101100111110011101010100111101	101110001110001001111001001001
110000101001111101011001111101	110001101101101111101101000110	110011001100110011001100110011
110111100111100111000010100001	110111110011100101000010001100	111000111000111000111000111000
111001111010010011111010110010	111100100010111111110110010100	111101111001000101111101100011
111110001000010010110111011100	111111100000000000000111111111	111111100011111000011110000000
111111110000000011111111100000		

G.1 Interpretability via Grad-CAM

To understand the model's decision-making process, we employed Gradient-weighted Class Activation Mapping (Grad-CAM). This technique produces heatmaps that highlight which regions of the input the model focused on most when making a prediction. As shown in Figure 10, the visualizations confirm that the model's attention (warmer colors) aligns with physically relevant and vulnerable areas, particularly the unprotected shoreline segments where inundation originates. This alignment validates that the model is learning meaningful spatial logic rather than spurious correlations, providing crucial transparency for stakeholders and planners.

G.2 Predictive Uncertainty Quantification

To measure the model's confidence in its predictions, we implemented a deep ensemble method, training five independent models and calculating the pixel-wise standard deviation of their outputs. This standard deviation serves as a direct proxy for model uncertainty. The resulting maps in Figure

Table 9: Generalizability set scenarios for the SF region under 0.5 m and 1.5 m SLR. Each row contains binary strings of length 30, with 1 indicating a protected OLU and 0 indicating an unprotected OLU.

SLR Generalizability Scenarios (SF)		
000000000000000000000000000000	000000000000000000000000000001	000000000000000000000000000010
000000000000000000000000000100	000000000000000000000000001000	000000000000000000000000010000
000000000000000000000000100000	000000000000000000000001000000	000000000000000000000010000000
000000000000000000000100000000	000000000000000000001000000000	000000000000000000010000000000
000000000000000000100000000000	000000000000000001000000000000	000000000000000010000000000000
000000000000000100000000000000	000000000000001000000000000000	000000000000010000000000000000
000000000000100000000000000000	000000000001000000000000000000	000000000010000000000000000000
000000000100000000000000000000	000000001000000000000000000000	000000010000000000000000000000
000000100000000000000000000000	000001000000000000000000000000	000010000000000000000000000000
000100000000000000000000000000	001000000000000000000000000000	010000000000000000000000000000
100000000000000000000000000000	111111111111111111111111111111	

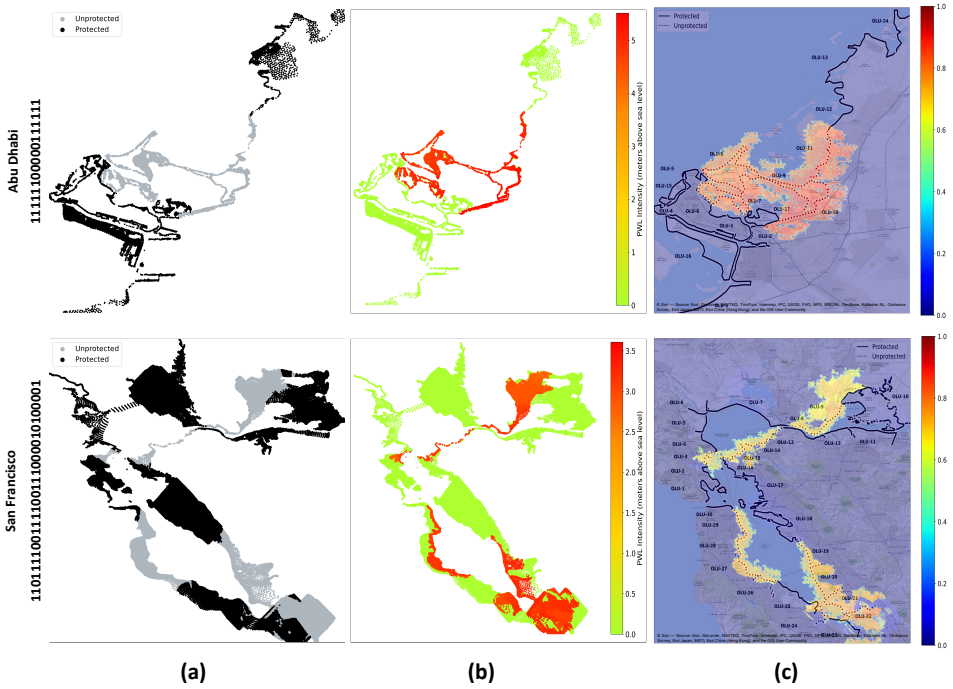


Figure 10: CASPIAN-v2 inundation prediction and interpretability for AD (top) and SF (bottom): (a) Input maps, (b) Predicted inundation, (c) Grad-CAM visualizations highlighting model attention, which aligns with unprotected and vulnerable areas.

11 reveal a strong spatial correlation between high predictive uncertainty (lighter colors) and high prediction error. This indicates that the model effectively learns to identify regions where its own predictions are less reliable. This self-awareness is invaluable for coastal planners, as it allows them to trust high-certainty predictions while flagging high-uncertainty zones as areas requiring a greater margin of safety or further detailed study.

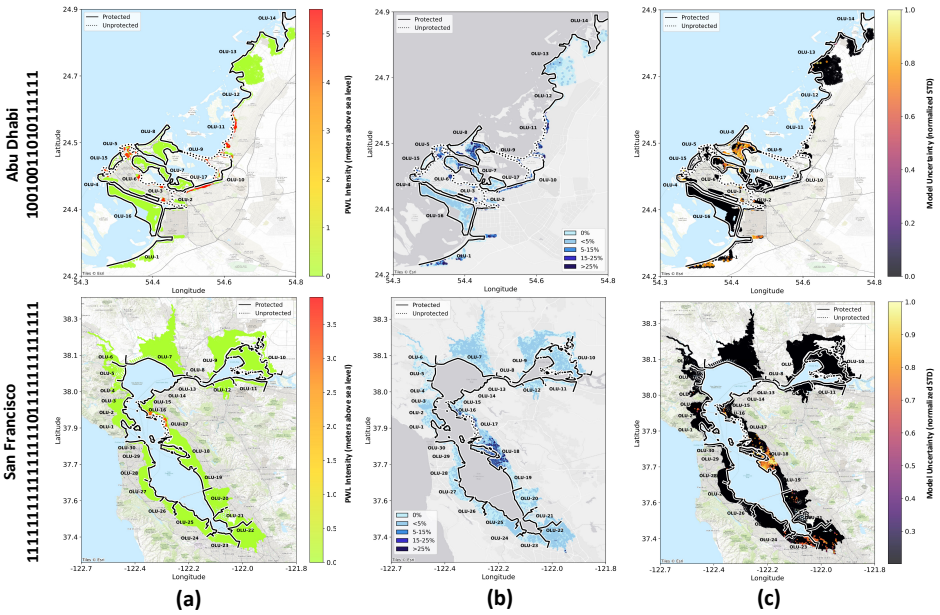


Figure 11: Predictive uncertainty maps for AD and SF scenarios. (a) Ground truth inundation. (b) Absolute error of the ensemble mean prediction. (c) Pixel-wise predictive uncertainty, where lighter colors indicate higher uncertainty and align with areas of higher error.