ABSTRACT

MCKENNA, MOLLY ELIZABETH. Deep Learning-Based Prediction of Coastal Tide Flood Maps. (Under the direction of Casey Dietrich).

Coastal storm-tide flooding arises from the interactions of storm surge, tides, and coastal geomorphology, and remains one of the most damaging hazards for low-lying shorelines. Predicting the spatial extent and magnitude of storm-tide inundation requires resolving nonlinear hydrodynamics and fine-scale coastal features. However, high-resolution process-based models are expensive, limiting their use in applications that require rapid scenario evaluation or large ensembles. This motivates the development of fast, data-driven surrogate models that can approximate storm-tide behavior while retaining essential physics.

This thesis develops an artificial intelligence (AI) model to predict peak storm-tides as flood maps with sufficient detail to describe hazards at community scales. Unlike prior approaches that predict storm surge or storm tides at discrete points, this model generates continuous inundation maps that consider the geomorphic controls governing coastal flooding. This model is trained on a library of process-based model simulations spanning a realistic range of storm intensities, forward speeds, track geometries, and landfall locations. Storm-tide maps are predicted for coastal North Carolina, which has a complex coastal region with narrow and wide shelves, barrier islands, inlets, sounds, estuaries, and extensive low-lying floodplains, and which is threatened by landfalling tropical cyclones.

A central methodological contribution of this work is a physics-aware tiling and data-augmentation framework that restructures each full-domain simulation into a set of local storm-tide map tiles. Rather than treating each storm as a single, domain-wide field, the augmentation extracts 64 non-overlapping map tiles (0.25° longitude by 0.25° latitude) that retain geospatial reference, coastal morphology, and hydrodynamic structure. This transformation allows the model to learn how individual storms affect specific regions of the coastline, rather than attempting to generalize across the entire domain at once. The model can see diverse, meaningful examples without introducing distortions from naive image-based augmentation, expanding the effective training dataset by more than an order of magnitude. The neural-network architecture fuses two complementary data streams – temporal storm evolution (track, intensity, motion) and static elevation maps encoding pre-storm geomorphology – and is trained using a staged Huber-loss strategy that refines spatial accuracy. Together, these components enable the AI model to learn localized storm-tide responses while maintaining consistency with the underlying coastal physics.

Model performance is evaluated across tens of thousands of test tiles spanning the full

range of coastal settings and flood responses. The surrogate achieves an overall root-mean-square error (RMSE) = 0.2722 m and maintains strong accuracy across the critical 0 m to 3 m elevation range that governs inundation onset, shoreline overtopping, and inland flood propagation. Elevation-dependent analyses reveal predictable patterns: highest accuracy in persistently dry uplands, moderate accuracy in shallow-water transition zones where small vertical differences strongly influence flooding, and larger errors in sparsely sampled or highly nonlinear regions such as tidal channels. Tile-level case studies further show how storm approach angle, intensity, and local geomorphology shape prediction fidelity – parallel approaching storms over barrier islands produce larger errors due to enhanced alongshore gradients, whereas perpendicular tracks yield cleaner cross-shore forcing that the model captures more reliably. These results highlight both the model's strengths in reproducing coherent surge patterns and its limitations in regions with complex hydrodynamics or limited training representation.

By predicting a map of storm tide in milliseconds, this model enables a framework for real-time hazard assessment, which can be scaled to other coastlines. This model demonstrates that neural-network surrogates can reproduce the essential physics of storm-tide dynamics and be useful during storms.

Deep Learning-Based Prediction of Coastal Tide Flood Maps

by
Molly Elizabeth McKenna

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Department of Civil Engineering

Raleigh, North Carolina
2026

APPROVED BY:

Katherine Anarde

John Baugh

Casey Dietrich
Chair of Advisory Committee

DEDICATION

To my family and friends who have supported me every step of the way, I love and appreciate you all.

BIOGRAPHY

Born and raised in North Carolina, the author grew up with a deep love for beaches and an equally deep curiosity about how the world works. A self-described lifelong nerd, she spent her childhood asking too many questions, and finding comfort in science and math. In high school, she majored in engineering and scientific visualization, where she discovered the joy of the scientific process and the power of computational tools to illuminate complex problems.

She attended North Carolina State University for her undergraduate degree in civil engineering. There, she fell in love with coding and the way computational thinking reshaped her approach to engineering challenges. Although she considered switching to computer science, she realized that the computational methods she loved could be applied directly to the civil and environmental problems she cared about most. During this time, she was introduced to coastal engineering and felt an immediate pull toward protecting vulnerable coastal environments.

In her senior year, she joined the Coastal & Computational Hydraulics Team as an undergraduate researcher, an experience that shaped her academic path and led to her master's thesis work. Through this research, she discovered that she could combine her computational skills with her desire to support coastal communities, and she began expanding her work into AI and machine learning applications for coastal modeling.

During her master's program, she presented her research at six conferences, developing a passion for communicating complex ideas to audiences ranging from machine learning specialists to community members with no technical background. She hopes to continue working in research, developing computational tools that support coastal resilience, and creating accessible ways for communities to understand and use scientific information.

Outside of academics, her life in Raleigh became just as formative. She found a home in the city's queer spaces, its coffee shops, and its vibrant art markets, where she spends weekends searching for new pieces that bring beauty and meaning into her life. Raleigh became the place where she built a community, a sense of belonging, and a creative identity that complements her scientific one.

ACKNOWLEDGMENTS

I would first like to thank my advisor, Casey Dietrich. You have been the best mentor I could have asked for. Your guidance pushed me to become the strongest researcher I can be, and I am forever grateful for the time, patience, and care you invested in this work.

Thank you to Katherine Anarde, who first introduced me to coastal engineering and helped me fall in love with coastal processes. I have always looked up to you – both as a scientist and as a person. Your insight into coastal dynamics and your encouragement to stay grounded in the physical processes were essential to this research and deeply appreciated.

Thank you to John Baugh, for being both my undergraduate advisor and a steady source of computational wisdom. Your guidance throughout my academic career has shaped the scientist I am today, and your perspective on the computational implications of this work was invaluable.

To Tomás and Dylan, thank you for the monthly meetings, the thoughtful discussions, and the help in shaping this neural network into a functional model.

To Jorge, thank you for being a wonderful teacher and a constant source of support in classes and in presentations.

To the Coastal Engineering Team – Jenero, Seun, Thomas, Ryan, Nahruma, Brooke, Roya, Sophie, Ebrahim, Juan, Ian, Javier, Nicole, and Sarah Grace – thank you for supporting me throughout these years of research. You became a huge part of my community in Raleigh and an essential part of my found family. I appreciate each of you immensely, and I love you all so much.

Thank you to my parents, who pushed me, supported me, and gave me pep talks on the hardest days. Your belief in me is something I will never take for granted. Thank you to Ian, for being supportive and loving in the very specific way only little brothers can be. And thank you to the rest of my family for your love, encouragement, and well wishes.

To Emma, the best roommate I could ask for – thank you for being there for everything from ridiculous TV shows to listening to me complain about failed cluster jobs. You have been a part of my safe place every time I came home.

To my friends – Jenna, Shannon, Brynn, Macy, Tess, Abby, Anusha, Sanskriti, and Kyleigh – thank you for your friendship, love, and unwavering support.

And finally, to Peanut Butter, my emotional support cat – thank you for waking me up at ungodly hours and brightening my life simply by existing.

TABLE OF CONTENTS

List of Tables . . . . **vii**

List of Figures . . . . **viii**

Authorship Statement . . . . **xiv**

Chapter 1 Introduction **1**
- 1.1 Definitions 1
  - 1.1.1 Storm Tide 1
  - 1.1.2 Process Models 2
- 1.2 Motivation 3
  - 1.2.1 Continued Need for Fast, Physical Predictions 4
  - 1.2.2 Communication of Storm Tide Predictions 5
- 1.3 Background 6
  - 1.3.1 AI Models for Storm Surge & Tide Prediction 6
  - 1.3.2 Image Prediction Approaches 10
- 1.4 Current Limitations 12
  - 1.4.1 Availability of Training Data 12
  - 1.4.2 Design of AI Model Architectures 13
  - 1.4.3 Working with Large Datasets 14
  - 1.4.4 Summary and Research Hypothesis 15
- 1.5 Funding Acknowledgment 15

Chapter 2 Background **16**
- 2.1 Study Area 16
- 2.2 Tropical Cyclones in Coastal NC 20
  - 2.2.1 Observed Storm Tides during Historical Storms 20
  - 2.2.2 Synthetic Storms in Atlantic Basin 22
  - 2.2.3 Sample of Synthetic Storms affecting Coastal NC 24
- 2.3 ADCIRC 24
  - 2.3.1 Unstructured Mesh to Describe Coastal NC 25
  - 2.3.2 Tide Forcing 25
  - 2.3.3 Atmospheric Forcing 27
  - 2.3.4 Predictions of Peak Storm Tides 28
- 2.4 Convolutional Neural Networks 29
  - 2.4.1 One-Dimensional Convolutional Layer 30
  - 2.4.2 Two-Dimensional Convolutional Layer 30
  - 2.4.3 Pooling Layer 31
  - 2.4.4 Dense Layer 31
  - 2.4.5 Data Augmentation 32

Chapter 3 Methods **33**
- 3.1 Storm-Tide Maps as Raster Images 33

3.1.1 Maximum Storm Tide Downscaling to Raster 33
3.1.2 Grayscale Image Process 36
3.1.3 Dry-Land Input 37
3.1.4 Storm Tide Output 38
3.2 Augmentation of Storm-Tide Maps 39
3.2.1 Mean Wetness Variability 41
3.2.2 Maximum Dissimilarity Algorithm 42
3.2.3 Data Shuffling 44
3.3 Development of Temporal Inputs 47
3.3.1 Closest 25 Hours 48
3.3.2 Input parameter Selection 48
3.3.3 Geographic Referencing 50
3.3.4 Temporal Input Summary 50
3.4 Neural Network Architecture 51
3.4.1 Temporal Stream: 1D Convolutional Encoder 53
3.4.2 Spatial Stream: Land-Mask Encoder 54
3.4.3 Feature Fusion and Latent Reshaping 54
3.4.4 Decoder: Multi-Stage Upsampling Network 54
3.4.5 Full Model Architecture Details 54
3.4.6 Training Procedure 56
3.4.7 Weighted Loss 57

Chapter 4 Results and Discussion 58
4.1 CNN for Prediction of Storm Tide Maps 58
4.1.1 Training 58
4.1.2 Training Behavior and Huber Loss Regimes 58
4.1.3 Overall Model Performance 61
4.1.4 Spatial Metrics 66
4.2 Discussion 74
4.2.1 Comparison to Process-Based and AI Models 74
4.2.2 Attempts at Model Improvement 76

Chapter 5 Conclusion 86

References 89

APPENDICES 99
Appendix A Acronyms 100
Appendix B Full Model Architecture 102

LIST OF TABLES

Table 3.1 Storm Parameter Index Table. 51

Table 4.1 Summary of the nine training stages, including epoch ranges, Huber loss thresholds, RMSE and MAE behavior, and qualitative notes on convergence characteristics. 61

Table A.1 A summary of acronyms used in alphabetical order. 100

Table B.1 Full Model Architecture Summary 102

LIST OF FIGURES

Figure 2.1 Elevation distribution of the ground surface across the study area, spanning -1.5 m to 15.5 m NAVD88. The color bar represents elevation ranges from shallow estuarine bathymetry to higher barrier-island and mainland elevations. Although portions of the estuarine system contain depths greater than -1.5 m, the range shown here is truncated to emphasize the elevations most relevant to water-level dynamics and overland flooding. Elevations are derived from the binned 1-m CoNED Topobathy DEM (OCM Partners 2026). 17

Figure 2.2 Digital elevation model (DEM) of coastal North Carolina stored in EPSG:6346 and rendered at its native 1-m horizontal resolution. Elevations range from a maximum of 46.36 m in the upland interior to bathymetric depths of -30.99 m in the offshore waters. The color-shaded DEM highlights the low-relief coastal plain, barrier-island morphology, and submerged estuarine basins that shape regional hydrodynamics and flood pathways. 18

Figure 2.3 Hydrograph at Duck NC during Hurricane Idalia. Observed and forecast water levels at Duck, NC, showing how storm-driven surge interacts with the astronomical tide to produce elevated storm-tide water levels. Peaks occur when surge arrival coincides with high tide, pushing total water levels into minor-to-moderate flooding thresholds. Color-coded flood stages highlight the nonlinear amplification that arises from tide–surge coupling during coastal storms (US Department of Commerce 2023). 21

Figure 2.4 STORM dataset that represents a random period of 1000 years of tropical cyclone tracks. Red boxes are the storm basins, and the generated storms for each basin are colored based on their maximum wind speeds (reproduced from Bloemendaal et al. 2020). 22

Figure 2.5 Selection of synthetic storms that impact coastal NC, identified by the red polygon (reproduced from Cuevas López et al. 2025). 23

Figure 2.6 Local mesh resolution of the ADCIRC computational grid across the study area. Colors indicate the characteristic element size, with darker shades (blue colors) representing finer resolution near the coastline, inlets, and estuarine channels, and lighter shades (green colors) indicating coarser offshore elements. This spatially varying resolution reflects targeted refinement in dynamically important regions. 26

Figure 2.7 Tide sampling added to ADCIRC simulations that represent the two-month period for astronomical tides at Duck NOAA tide gauge (reproduced from Cuevas López et al. 2025). 27

Figure 2.8 Maximum water elevation (`maxele`) field for Storm-0 from the ADCIRC simulation, shown on the native unstructured mesh nodes. Colors indicate the peak water level at each wet node across the study area, highlighting spatial variability in storm-driven surge. Dry-mesh regions appear as gaps where ADCIRC does not compute water levels. 29

Figure 2.9 A schematic overview of a convolutional neural network (CNN) architecture. The 2-D input field is processed through a convolutional layer to extract spatial features, followed by a pooling layer that reduces spatial resolution. The resulting feature maps are flattened and passed through fully connected layers to produce the final output (reproduced from Zang et al. 2020). 31

Figure 3.1 Maximum storm tide (`maxele`) output from ADCIRC for Storm 0, shown as the set of wet mesh vertices above the ADCIRC mesh (gray lines). Each point represents a vertex where ADCIRC computes a maximum water elevation, while the gray regions correspond to dry areas where no water level is defined. 35

Figure 3.2 Comparison between the raw ADCIRC maximum water-level output (left), defined at the vertices of an unstructured, finite element mesh, and the DEM-aligned GeoTIFF generated through linear interpolation (right). The interpolated raster adopts the DEM's grid, affine transform, and CRS, producing a continuous elevation surface suitable for downstream grayscale encoding and machine-learning workflows. . . 36

Figure 3.3 Example of the grayscale-encoding process. The DEM-aligned storm-tide GeoTIFF (left) is transformed into an 8-bit grayscale image (right) by mapping physical water elevations into the 0–254 range and assigning 255 to nodata pixels. This encoding preserves spatial structure while reducing file size and enabling efficient CNN training. 38

Figure 3.4 Example of the preprocessing step used to create the CNN training targets. Left: the interpolated ADCIRC maxele image, where dry regions appear as white gaps because ADCIRC reports values only at wet mesh vertices. Right: the same image after filling dry pixels with elevations from the dry-land DEM, producing a complete, continuous raster. Smaller 512×512 tiles extracted from this filled raster serve as the inputs and prediction targets for the CNN. 39

Figure 3.5 Tile-selection strategy used for storm-tide map augmentation. The full $2^\circ \times 2^\circ$ domain is partitioned into a uniform grid of 64 tiles (cyan boxes), providing complete spatial coverage. An additional 20 tiles (red boxes) are centered on locations selected by a Maximum Dissimilarity Algorithm (MDA) applied to the wetness-variability field, emphasizing regions where the wet/dry interface shifts across storms. Together, these 84 tiles form the augmented representation of each storm-tide map used for model training. 40

Figure 3.6 Mean wetness variability across the ADCIRC storm library. Values range from 0 to 0.25, the maximum possible Bernoulli variance. High-variability regions (yellow–orange) correspond to the dynamic wet/dry interface, where inundation depends strongly on storm characteristics. Low-variability regions (dark blue) represent areas that are consistently wet (open ocean) or consistently dry (inland terrain). 42
Figure 3.7 This figure illustrates the full preprocessing workflow applied to the storm-image dataset. The left block shows the original 3D tensor of storm slices, where each horizontal gridline corresponds to a single storm image. The middle block shows the randomized shuffling of slices, with each arrow mapping an original index to a unique new position; matching colored gridlines highlight the permutation. The right block shows the shuffled dataset partitioned into training, validation, and testing subsets using a 70/15/15 split. 45
Figure 3.8 Schematic of the storm-tide library and the two shuffling strategies used to construct training, validation, and testing datasets. 46
Figure 3.9 Vertical flow chart of the full model architecture with storm track inputs, image inputs, and continuous elevation outputs. 52
Figure 3.10 Temporal stream: 1D convolutional encoder applied to the (25 × 35) storm-track matrix. 53
Figure 3.11 Simplified image up-sampling architecture (each stack of squares represents multiple Conv2DTranspose layers with different channel depths and resizing techniques). Top: first six stacks of Conv2DTranspose layers. Bottom: final three stacks of layers. 55

Figure 4.1 Comparison of RMSE across all nine training stages, shown as a continuous sequence of epochs. Solid lines represent training RMSE and dashed lines represent validation RMSE. Each stage begins where the previous stage ended, illustrating the staged training process and the progressive reduction in error as the model converges. Validation RMSE tracks training RMSE across stages, indicating stable generalization performance and consistent improvement throughout the multi-stage training pipeline. 59
Figure 4.2 Normalized confusion matrix comparing predicted and actual elevations across the full test domain. Each cell shows the percentage of pixels whose predicted elevation falls into the same elevation bin as the true elevation. The strong diagonal structure indicates close agreement across most elevation ranges, with misclassifications concentrated in low-lying transitional zones and negative-elevation regions where hydrodynamic behavior is nonlinear. Higher-elevation bins exhibit minimal confusion, reflecting the model's ability to identify non-inundated terrain. 62

Figure 4.3 (Top) Histogram of actual and predicted elevations (storm-tide or ground-surface) across the full test dataset. The predicted distribution matches the true distribution at most elevations, but shows an under-representation near 1 m, where the true elevation distribution peaks. (Bottom) Root-mean-square error (RMSE) as a function of elevation bin for the full test domain. Errors are largest in the 4 m to 5.5 m elevation range. RMSE decreases sharply in the 0 m to 1 m coastal fringe and remains low through the 1 m to 3 m band, reflecting strong model performance in the low-lying regions that dominate coastal flooding dynamics. RMSE declines again at higher elevations, where the model predicts non-inundated terrain. 64
Figure 4.4 Comparison of model performance for two storms with different track orientations: Storm 562 (top, parallel approach) and Storm 1121 (bottom, perpendicular approach). Each row shows (from left to right) the storm track with category coloring, predicted elevation, actual elevation, and the spatial difference (Predicted minus Actual). The storm-track panels include the full train–validation–test tile configuration, with tiles colored cyan for train, gold for validation, and magenta for test, allowing direct comparison of model performance across regions seen during training and those held out for validation and testing. The difference panel uses a diverging blue–red scale in which blue denotes underprediction and red denotes overprediction. 66
Figure 4.5 Mean absolute error (MAE; left) and root-mean-square error (RMSE; right) computed for each of the 64 tiles that make up the full 4096 × 4096 test-domain extent. Each tile represents a 0.25° × 0.25° region, and the metrics shown here reflect the average elevation error within each tile across all test storms. Spatial patterns in tile-level error highlight regions where the model performs consistently well and regions where prediction uncertainty increases due to complex topography, hydrodynamic variability, or limited training representation. 67
Figure 4.6 Spatial distribution of mean signed error (Predicted minus True) across the test-domain extent. Blue regions indicate systematic under-estimation, particularly along the coastline, while red regions indicate over-estimation in deeper coastal waters and within a localized sector of Pamlico Sound. 68
Figure 4.7 Spatial distribution of test-set sampling density across the 64-tile domain. Tiles along the central coastal plain contain the highest number of test samples, while interior estuarine tiles –including the Pamlico Sound tile with the strongest over-estimation – contain fewer test samples due to the storm-based dataset split. 69
Figure 4.8 Comparison of model performance for Tile 0, with a representative parallel-track – storm 66 (top) and perpendicular-track – storm 806 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual). 70

Figure 4.9 Comparison of model performance for Tile 26, with a representative parallel-track – storm 295 (top) and perpendicular-track – storm 806 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual). 71

Figure 4.10 Comparison of model performance for Tile 9, with a representative parallel-track – storm 626 (top) and perpendicular-track – storm 806 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual). 72

Figure 4.11 Comparison of model performance for Tile 10, with a representative parallel-track – storm 66 (top) and perpendicular-track – storm 70 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual). 73

Figure 4.12 Binned confusion matrix for the static model retrained with the elevation-weighted loss. The diagonal structure is weaker than in the baseline model (Figure 4.2), and new off-diagonal mis-classification patterns appear in the low-elevation bins, indicating degraded stability after switching to the weighted loss. 77

Figure 4.13 (Left) Histogram of actual and predicted elevations for the static weighted-loss test dataset. The predicted distribution retains the same biases as the baseline model, including over-prediction near 1 m and under-prediction at negative elevations. (Right) RMSE as a function of elevation bin for the weighted-loss model. RMSE increases across all elevation ranges, reaching values up to 7 m, indicating an instability introduced by the weighted-loss formulation. 78

Figure 4.14 (Left) Histogram of actual and predicted elevations for the MDA test dataset. The predicted distribution closely matches the true distribution but shows slight over-representation near 1 m. (Right) RMSE as a function of elevation bin for the MDA model. Errors peak in the 4 m to 5.5 m elevation range, while RMSE remains lowest in the –1.5 m to 2 m band, reflecting strong performance in low-lying regions that dominate coastal flooding dynamics. 80

Figure 4.15 Binned confusion matrix for the MDA model comparing predicted and actual maximum water levels (m). The diagonal structure indicates overall agreement, while persistent off-diagonal patterns in low and negative elevation bins reflect the same biases observed in the static model. 81

Figure 4.16 (Left) Histogram of actual and predicted elevations for the storm-wise shuffled MDA test dataset. The predicted distribution follows the true distribution but retains slight over-prediction near 1 m and under-prediction at negative elevations. (Right) RMSE as a function of elevation bin for the storm-wise shuffled MDA model. The mid-elevation error peak (4–5 m) persists, with no clear improvement in the elevation ranges targeted by the MDA tiles. 82
Figure 4.17 Binned confusion matrix for the storm-wise shuffled MDA model comparing predicted and actual maximum water levels (meters). The diagonal structure indicates overall agreement, while the low-elevation bins show similar behavior to earlier MDA experiments, suggesting that the model has not yet leveraged the additional high-variability information provided by the MDA tiles. 84

AUTHORSHIP STATEMENT

Contributions of Molly McKenna and coauthors are listed below for each chapter.

Chapter 1

1. Molly McKenna: sole author
2. Casey Dietrich: editing/revising

Chapter 2

1. Molly McKenna: sole author
2. Casey Dietrich: editing/revising

Chapter 3

1. Molly McKenna: sole author
2. Casey Dietrich: editing/revising

Chapter 4

1. Molly McKenna: sole author
2. Casey Dietrich: editing/revising

Chapter 5

1. Molly McKenna: sole author
2. Casey Dietrich: editing/revising

Use of generative artificial intelligence: No generative artificial intelligence was used to write this dissertation.

Chapter 1

Introduction

1.1 Definitions

During storms, water levels are affected in coastal regions by the interactions of astronomical tides, atmospheric forcings, and coastal geomorphology. Process-based models can predict these interactions and the resulting high waters and coastal flooding.

1.1.1 Storm Tide

Tides are the regular, predictable oscillations in sea level caused by gravitational forcing (Doodson 1921). Storm surge is the rise in water level, above the normal tide levels, generated by wind stress and pressure forcing during a storm (Resio et al. 2009; Westerink 2008). Storm tide is the total water level produced during a coastal storm, defined as the sum of the tide and storm surge (Harris 1963). The astronomical tide has an amplitude and phase that vary with local bathymetry, friction, and resonance characteristics (Proudman 1953). Storm surge is affected by wind stress, pressure forcing, wave setup, and momentum fluxes that drive water toward the coast (Resio et al. 2009; Westerink 2008). Surge magnitude depends on storm size, intensity, forward speed, angle of approach, and the geometry of the continental shelf (Irish et al. 2008; Thomas et al. 2019; Knowles et al. 2025).

Because tides and storm surge arise from distinct physical mechanisms, their interaction is nonlinear in both space and time. When a storm approaches land, the timing of the surge relative to the tidal cycle can modulate the resulting storm tide (Proudman 1955). Constructive interference occurs when peak surge coincides with high tide, producing higher total water levels than either component alone; destructive interference occurs when surge arrives near low tide (Doodson 1956). Along the U.S. Atlantic Coast, where tidal ranges can exceed 1 m, this phasing can shift inundation extent by kilometers and alter flood depths by tens of centimeters or more (Proudman 1955). Because storm tide – not surge alone – determines the flooding actually experienced at the coast, coastal flooding and inundation is expressed in terms of total water level relative to a vertical datum, rather than surge height alone (National Hurricane Center 2016).

1.1.2 Process Models

Storm tides can be predicted by process-based hydrodynamic models, which solve the shallow-water equations under wind, pressure, and tidal forcing (Dietrich et al. 2011; Westerink et al. 2008). Models like SLOSH (Sea, Lake, and Overland Surges from Hurricanes) (Jelesnianski et al. 1992) and ADCIRC (ADvanced CIRCculation) (Luettich et al. 1992) represent the physics governing surge generation and propagation, including nonlinear advection, bottom friction, tidal interactions, and the influence of coastal geometry. These models can resolve complex estuarine systems, barrier islands, and wetting-drying processes, making them the state-of-practice for storm tide simulation.

SLOSH

SLOSH was developed by the National Weather Service to provide fast, operationally reliable storm-surge guidance for real-time forecasting and evacuation planning (Jelesnianski et al. 1992). SLOSH has been applied in operational settings, particularly for basin-scale surge envelopes and rapid scenario generation (Jelesnianski et al. 1992; Glahn et al. 2009; Forbes et al. 2010). However, its simplified physics and coarse resolution limit its ability to represent fine-scale estuarine hydrodynamics, inlet dynamics, and wetting–drying transitions in complex coastal environments. Studies comparing SLOSH to higher-fidelity models have shown that it can underpredict or overpredict surge in regions with narrow channels, barrier islands, or strong tidal interactions (e.g., Zhang et al. 2008; Turan et al. 2018). These limitations motivate the use of more detailed models for site-specific inundation prediction.

A key strength of SLOSH is its computational speed. Its computational efficiency comes from simplifying the governing equations, using coarse polar or curvilinear grids, and parameterizing many small-scale processes. This efficiency enables the National Hurricane Center (NHC) to produce probabilistic surge guidance and evacuation-planning products (National Hurricane Center 2026) that would be infeasible with high-resolution hydrodynamic models. SLOSH can run thousands of hypothetical storm scenarios for probabilistic products such as MEOW (Maximum Envelope of Water) and MOM (Maximum of MEOWs), which underpin FEMA (Federal Emergency Management Agency) evacuation zones and NHC surge risk communication (Glahn et al. 2009).

ADCIRC

ADCIRC (ADvanced CIRCulation) was developed to provide high-resolution simulations of storm surge, tides, and coastal circulation on unstructured meshes (Luettich et al. 1992; Westerink et al. 2008). Its finite-element formulation allows spatial resolution to vary from tens of meters in estuaries to tens of kilometers offshore, enabling accurate representation of coastal geometry, barrier islands, tidal inlets, and floodplain connectivity. ADCIRC solves the depth-integrated shallow-water equations using the Generalized Wave Continuity Equation (Kinnmark 1986) and momentum balance, incorporating wind stress, atmospheric pressure gradients, Coriolis forcing, bottom friction, and nonlinear advection. This flexible formulation allows ADCIRC to represent the multiscale hydrodynamics that govern storm-tide evolution across continental shelves, nearshore zones, and complex estuarine systems.

ADCIRC has been applied in hindcast studies of major storms to evaluate coastal vulnerability, validate model physics, and support infrastructure planning. Westerink et al. (2008) validated a basin- to channel-scale ADCIRC implementation through hindcasts of Hurricanes Betsy (1965) and Andrew (1992), demonstrating that the model can accurately reproduce peak storm-surge heights and overland inundation across the complex coastal and inland waterways of southern Louisiana. Hope et al. (2013) reproduced observed water levels during Hurricane Ike (2008) using a high-resolution unstructured mesh, demonstrating the model's ability to capture surge amplification and coastal inundation pathways along the Texas and Louisiana coasts. Bilskie and Luettich (2024) validated ADCIRC for Hurricane Michael (2018), showing that accurately reproducing the storm's extreme water levels required resolving advection-driven accelerations – particularly Bernoulli and centrifugal effects – arising from regional coastal geometry and the storm's wind field. These studies highlight ADCIRC's ability to reproduce observed water levels, capture nonlinear tide–surge interactions, and resolve localized flooding processes such as barrier-island overtopping, inlet breaching, and estuarine amplification.

1.2 Motivation

As a storm approaches the coast, decisions about evacuations, road closures, and emergency staging must consider the evolving hazards due to storm tides and flooding. Emergency managers depend on storm-tide and inundation guidance to determine which areas are likely to flood, how deep water levels may become, and how quickly conditions may deteriorate (Burston et al. 2015; Becker et al. 2021). The Federal Emergency Management Agency (FEMA) provides long-term hazard and evacuation-planning products, but the real-time storm-tide and inundation guidance used during an approaching storm is generated by National Oceanic and Atmospheric Association (NOAA) and the National Hurricane Center. These products must be fast, reliable, and explicit, because the geographic distribution of flooding – not just its magnitude – determines real-world impacts (FEMA 2026). CERA (Coastal Emergency Risks Assessment) is one example of a platform that provides map-based storm-tide visualizations used in emergency operations (Coastal Emergency Risks Assessment 2024). However, the models capable of producing the most accurate, map-based predictions are also the most demanding, creating a tension between physical fidelity and operational timeliness.

1.2.1 Continued Need for Fast, Physical Predictions

Process models can predict storm tides for a variety of applications, all of which benefit from emphases on accuracy and efficiency. One example is the flood risk maps developed by FEMA and its partners (FEMA 2021), which can require hundreds of simulations of synthetic storms in order to understand the overall flood risk across a coastal region. These maps are used to set insurance rates for coastal homeowners, so accuracy is critical. These large ensembles require substantial computational resources because each simulation involves a full high-resolution storm-tide model run. High-resolution ADCIRC simulations alone may require hours to days of wall-clock time on high-performance computing systems, particularly when meshes include tens of millions of nodes or when the model is coupled with SWAN for wave–current interaction (Dietrich et al. 2012; Tanaka et al. 2011; Pringle et al. 2021). Another example is operational forecasting during storms. The National Hurricane Center issues tropical cyclone forecast advisories every 6 hr, providing updated estimates of storm track, intensity, size, and forward speed (National Hurricane Center 2025). Although these advisories do not predict coastal flooding, they supply storm information that can be used by hydrodynamic models that simulate storm surge, tides, and total water levels (Dresback et al. 2013; Bilskie et al. 2022).

Process models are then used for real-time forecasts of storm tides (National Hurricane Center 2016; Coastal Emergency Risks Assessment 2024), which are then shared with decision-makers. These predictions inform when evacuations should begin, which transportation corridors are likely to close, where to pre-position rescue assets, and how to allocate resources across vulnerable communities. Numerous studies have documented how emergency managers rely on storm-surge and storm-tide guidance when making these protective-action decisions, particularly under tight time constraints (e.g., Morss and Hayden 2010; Meyer et al. 2013). Morss and Hayden (2010) show that households made rapidly evolving preparation and evacuation decisions during Hurricane Ike, frequently seeking updated surge information from multiple sources to judge when conditions would become dangerous. Although focused on residents, their findings highlight the broader operational challenge: surge risk evolves quickly, and protective-action decisions must be supported by timely, clearly communicated guidance. These decisions have a direct effect to protect lives and property during storms, so accuracy is critical. But they can also require a significant computational effort during each advisory cycle. Meyer et al. (2013) similarly find that people adjust their protective actions dynamically as new hurricane information becomes available, with preparation accelerating when forecast graphics depict a clearer or more imminent threat. This behavior highlights the same operational challenge from a different angle: storm-tide guidance must be refreshed and communicated quickly enough to shape time-sensitive decisions before evacuation routes or critical infrastructure become compromised. Together, these studies point to a central tension in operational forecasting. Storm-tide predictions must be both accurate and fast, yet achieving both simultaneously is difficult. High-resolution hydrodynamic models can represent the processes that control coastal flooding but are expensive, while faster models sacrifice spatial detail and process representation. The need for predictions that are accurate, fast, and map-based motivates the exploration of alternative modeling approaches, including data-driven and AI-based surrogates.

1.2.2 Communication of Storm Tide Predictions

While operational speed is essential, the value of storm-tide forecasts ultimately depends on how clearly and intuitively they convey spatial risk to the people who must act on them. A substantial body of research in risk communication shows that the effectiveness of storm-tide predictions depends not only on the accuracy of the underlying model but also on how hazard information is represented and interpreted by users. Storm-tide hazards are inherently spatial: decision-makers must determine *which neighborhoods will flood, which roads will be overtopped, and which facilities are at risk*. Because these impacts depend on the pathways water takes across the landscape, storm-tide information is most useful when communicated through spatial, map-based formats rather than point-based time series. Spatial context transforms a water-level value into a consequence: a predicted peak of 2.3 m is abstract, but a map showing that the only evacuation route from a barrier-island community will be underwater in six hours is actionable (Forbes et al. 2010; National Hurricane Center 2016; Becker et al. 2021).

Research in risk communication shows that spatial visualization improves comprehension, supports faster interpretation of spatial patterns, and increases public understanding of coastal hazards (Morss et al. 2008; Lazo et al. 2010; Meyer et al. 2014; Morrow et al. 2015). Spatial formats also help people form more accurate mental models of how flooding will evolve, which improves their ability to anticipate consequences and evaluate protective actions (Morss et al. 2008). Lazo et al. (2010) shows that visual depictions of inundation increase users' confidence in their understanding of risk and make them more likely to take preparedness actions. Map-based displays reduce the cognitive effort required to integrate multiple pieces of hazard information, allowing users to more easily identify patterns, hotspots, and thresholds of concern (Meyer et al. 2014). Morrow et al. (2015) finds that emergency managers prefer clear, map-based surge products because these tools help them pinpoint vulnerable neighborhoods, communicate risk to elected officials and the public, and determine whether conditions warrant escalating evacuation orders.

More recent work demonstrates that spatial products are especially valuable for emergency managers, who must interpret localized impacts such as access constraints, infrastructure vulnerabilities, and cascading failures; map-based tools align with the inherently spatial nature of these decisions and support faster, more context-specific reasoning under time pressure (Becker et al. 2021). Together, these studies show that effective storm-tide communication depends not only on accurate physical predictions but also on spatially explicit visualizations that match how people perceive, interpret, and act on hazard information. Fast, accurate, spatially explicit storm-tide maps are therefore central to operational forecasting and emergency-management practice.

1.3 Background

1.3.1 AI Models for Storm Surge & Tide Prediction

Overview of AI Methods

Artificial intelligence (AI) and machine learning (ML) have become important tools for prediction, particularly in domains where physical processes are complex and expensive to simulate (Karpatne et al. 2017; Reichstein et al. 2019). In supervised learning, a model learns a mapping from inputs to outputs using labeled examples, while unsupervised learning identifies structure or patterns in unlabeled data (Goodfellow et al. 2016). Many applications in hydrology, meteorology, and coastal flooding rely on supervised learning (Kratzert et al. 2018), where the goal is to approximate the behavior of a physical system using historical observations or simulations as training data.

AI models vary in structure and complexity. Linear regression and decision trees represent simple, low-dimensional relationships between predictors and responses (Hastie et al. 2009).

Linear regression assumes a linear mapping between inputs and outputs, while decision trees partition the input space into discrete regions and assign predictions based on hierarchical rules (Tibshirani 2010). These approaches are interpretable and efficient but struggle with nonlinear or complex processes.

In contrast, neural networks provide a flexible framework for learning nonlinear processes. Feed-forward neural networks (FNNs) consist of stacked layers of neurons that transform inputs through learned weights and activation functions, enabling them to approximate arbitrary nonlinear relationships (Goodfellow et al. 2016). Convolutional neural networks (CNNs) extend this capability to spatial data by applying convolutional filters that detect local patterns, making them well-suited for gridded fields such as wind, pressure, or flood-depth maps (LeCun et al. 1995). Recurrent neural networks (RNNs) and their modern variants – such as long short-term memory (LSTM) and gated recurrent unit (GRU) networks – are designed to capture temporal dependencies by passing information from one time step to the next, making them effective for time-series prediction of water levels, river discharge, or storm evolution (Hochreiter and Schmidhuber 1997; Gu et al. 2018). Autoencoders learn compressed latent representations of high-dimensional data and can reconstruct complex spatial fields from low-dimensional encodings, a property that has been leveraged for surrogate modeling and dimensionality reduction in environmental systems (Hinton et al. 2006).

Neural networks have been used to emulate rainfall–runoff processes and predict river discharge (Kratzert et al. 2018), CNNs have been applied to estimate wave conditions from atmospheric forcing (Pena and Huang 2021), and autoencoder-based architectures have been used to approximate outputs from intensive numerical models in hydrology, oceanography, and climate science. Their appeal lies in the ability to capture nonlinear relationships and produce fast predictions once trained, making them attractive candidates for real-time or ensemble-based forecasting. This general framework underpins recent efforts to apply AI to storm surge and storm tide prediction.

Point-Based Predictions of Storm Surge

Early applications of machine learning to coastal flooding focused on point-based predictions, in which the model estimates the water levels at a small number of discrete coastal locations. These studies used storm parameters as inputs and observed or simulated surge at tide-gauge locations as outputs. Bezuglov et al. (2016) trained a feed-forward neural network using historical and synthetic hurricane tracks – including storm position, central pressure, maximum wind speed, and radius of maximum winds – as inputs, and observed storm surge at ten tide-gauge locations along the North Carolina coast as outputs. Saviz Naeini and Snaiki (2024) developed a hybrid deep-learning framework that combined a deep autoencoder

(DAE) with a deep neural network (DNN) to predict water levels at 289 coastal locations in New York and New Jersey. Their model used storm parameters and atmospheric forcing fields as inputs, and ADCIRC-simulated water levels as outputs. Together, these and other studies demonstrated that neural networks can approximate hydrodynamic behavior at many coastal points, but they were limited to surge-only, point-wise predictions, and did not produce continuous spatial fields.

A second class of early machine-learning studies predicted storm surge using time series of atmospheric variables measured at fixed stations, rather than storm-track parameters. In these models, the inputs consisted of local meteorological observations – such as wind speed, wind direction, barometric pressure, and sometimes offshore buoy measurements – recorded at hourly or sub-hourly intervals. The outputs were short-lead surge predictions at the same station, usually 1 hr to 6 hr ahead (Kim et al. 2019; Wang et al. 2021; Bai and Xu 2022; Chao and Young 2022; Chen et al. 2022). Because these models relied on local atmospheric time series, they performed well for near-term forecasting but could not represent the broader storm structure, including storm size, asymmetry, or approach angle.

To address this limitation, global studies like Tiggeloven et al. (2021) incorporated coarse atmospheric reanalysis fields surrounding tide-gauge locations. In these models, the inputs included gridded wind and pressure fields extracted from ERA5 (the ECMWF fifth-generation global atmospheric reanalysis, providing hourly fields of winds, pressure, and other variables) or similar datasets, while the outputs remained point-based surge predictions at individual gauges. However, the spatial window was small (e.g. 1–2° around the gauge), and the reanalysis resolution ($\approx$ 25–30 km) was too coarse to capture the full hurricane wind field. Recent work (e.g. Ian et al. 2023) has advanced this class of models by using station-based atmospheric observations – including wind speed, wind direction, barometric pressure, and tide-gauge water levels – as inputs to a bidirectional attention-based LSTM architecture. Although this approach improves short-lead surge prediction by capturing temporal dependencies more effectively, it still lacks explicit representation of storm evolution and coastal geometry.

To incorporate storm structure, a third class of studies adopted hybrid approaches in which hurricane parameters serve as inputs and process model outputs serve as training targets. These models rely on libraries of ADCIRC or SLOSH simulations to learn the relationship between storm characteristics and resulting water levels. Hashemi et al. (2016) trained a neural network using storm track, central pressure, maximum wind speed, and radius of maximum winds as inputs, with ADCIRC-simulated peak surge at multiple coastal locations as outputs. Their results showed that storm-parameter inputs capture broad storm structure more effectively than local atmospheric time series. Lee et al. (2021) incorporated parametric wind-field models and a large synthetic storm library to predict peak surge at hundreds of coastal points along the Gulf of Mexico. Adeli et al. (2022) used SLOSH-generated surge fields as training data and demonstrated that neural networks can approximate surge response across a wide range of hypothetical storms. Ayyad et al. (2022) further refined this approach by including storm size, forward speed, and landfall angle, improving generalization across different storm geometries. Although these studies differ in the details of their storm-parameter sets and simulation libraries, they share a common structure: storm parameters as inputs and point-based peak surge from hydrodynamic models as outputs.

Across this class of models, the reliance on storm parameters allows for improved representation of storm structure relative to atmospheric time-series approaches. However, these models remain limited to surge-only, point-wise predictions, and depend heavily on simulation libraries developed primarily for extreme-event risk analysis. None incorporate tidal phase or produce continuous spatial fields, which limits their ability to predict storm tide or generate inundation maps.

A more sophisticated point-based surrogate was developed by Pachev et al. (2023), who introduced a multi-stage framework for predicting peak storm surge directly on ADCIRC mesh vertices—the individual points that form the unstructured finite-element mesh used by ADCIRC. Their model first classifies whether each vertex becomes inundated and then predicts inundation depth for the subset of vertices classified as wet. A key innovation is a formulation in which surge is predicted at mesh vertices, enabling the model to generalize to locations not present in the training set and reducing the number of trainable parameters relative to structured-grid approaches. Because ADCIRC meshes can contain millions of vertices, Pachev et al. (2023) apply extensive spatial filtering to make training feasible. They restrict predictions to low-lying vertices within 10 km of the coastline, subsample vertices by a factor of ten, and limit predictions to within 150 km of landfall. The resulting surrogate achieves impressive accuracy and is orders of magnitude faster than ADCIRC, but it remains a surge-only, vertex-based model: it does not incorporate tides, does not produce continuous spatial fields, and therefore cannot generate flood maps.

Predictions of Storm Tides or Spatial Maps

In the last two years, researchers have pushed toward predictions of storm tides or spatial maps. The first model to predict storm tide rather than storm surge was introduced by Cuevas López et al. (2025). Their work represents a notable step forward because it incorporates the full storm and tide forcings – including their evolution in time – into a neural network designed to estimate peak storm tide at nine locations along the North Carolina coast. In addition to storm parameters, the model receives tidal inputs, including the local tidal phase and predicted astronomical tide at each site. These tidal inputs allow the network to account for the timing of landfall relative to the tide cycle, which is essential for distinguishing storm tide from storm surge. By conditioning predictions on storm evolution and tidal state, the model captures the nonlinear tide–surge interactions that shape peak water levels. Although the spatial coverage is limited to nine sites, this study is the first to demonstrate that neural networks can learn the combined tide-and-surge response when provided with both storm and tidal information, marking an important conceptual shift in the field.

A separate line of research has attempted to move beyond point predictions by generating storm-surge maps at coarse resolution by using computer-vision architectures. The most notable example is a global storm surge modeling project (Pachev et al. 2026), which provides trained models and processed datasets for predicting storm surge anywhere in the world. This model predicts storm surge on a 128-pixel × 128-pixel grid covering a 2.5° × 2.5° region centered on storm landfall. All input features – including bathymetry, a land–water mask, and wind and pressure fields sampled every 3 hr from 24 hr before landfall, to 12 hr after – are interpolated to this grid, resulting in a total of 41 input channels. The model uses a UNet architecture, a type of convolutional encoder-decoder network commonly used in image segmentation, to learn spatial patterns of surge generation from these gridded inputs. While this approach demonstrates that convolutional networks can produce spatially distributed storm-surge fields, the coarse resolution (grid cells on the order of 20 km to 25 km) and small domain size (275 km×275 km) limit its ability to represent estuarine flooding, barrier-island dynamics, and wetting–drying processes that occur at much finer spatial scales. As a result, the model captures large-scale surge patterns but cannot resolve the coastal geomorphology and hydrodynamic processes required for high-resolution inundation mapping.

1.3.2 Image Prediction Approaches

Recent advances in flood mapping from remote sensing imagery highlight the maturity of image-based segmentation methods but also underscore their difference from storm tide prediction. Awasthi et al. (2026) developed a Maxout-enhanced DeeplabV3+ model for flood detection using Sentinel-1 SAR (synthetic aperture radar) imagery, achieving strong performance on the Sen1Floods11 benchmark dataset (a globally curated collection of paired SAR images and flood-extent labels). These models demonstrate that semantic segmentation networks can produce accurate flood maps when trained on consistent, image-based observations. However, they operate on observed flood extents, not on predictions derived from storm characteristics or hydrodynamic forcing. Satellite-based flood maps cannot be connected to storm characteristics or tides in a predictive way because they provide only inundation extent – not water depth – and are collected at multi-day revisit intervals that miss the rapid evolution of hurricane-driven flooding. Even when the responsible storm is known, SAR imagery contains no information about storm intensity, wind and pressure forcing, tidal phase, or coastal bathymetry, all of which are required to forecast peak storm tide.

Surrogate modeling approaches have also expanded to multi-fidelity learning. Taghizadeh et al. (2025) introduced a multi-fidelity graph neural network (MFGNN) that combines numerous coarse-resolution simulations with a small number of high-fidelity simulations to improve flood hazard mapping accuracy. A key strength of this study is its demonstration that low-fidelity simulations can be used to guide and constrain the learning process, allowing the model to achieve high accuracy with far fewer expensive high-resolution runs. This provides a compelling strategy for reducing computational cost in domains where generating high-fidelity data is expensive. While effective for fluvial and pluvial flooding, these models rely on structured training datasets that are easier to generate than storm tide simulations. Riverine and rainfall-runoff models can be run in large batches on regular grids, with boundary conditions specified directly from precipitation or discharge time series. In contrast, storm-tide simulations require full hurricane wind-field generation, tide–surge coupling, unstructured-mesh hydrodynamics, and coastal wetting–drying processes, making them orders of magnitude more computationally intensive and far less amenable to large-scale multi-fidelity sampling.

Hybrid CNN–hydrodynamic approaches have been proposed for compound flooding as well. Xu et al. (2026) coupled a hydrological–hydrodynamic model with a convolutional neural network (CNN) to predict the evolution of flood depth in an urban setting. A notable strength of this study is its demonstration that a CNN can learn the temporal progression of flood depths from physics-based simulations and reproduce hydrodynamic outputs with high accuracy while operating hundreds of times faster than the underlying model. The work also shows that deep learning can capture nonlinear interactions between rainfall, drainage capacity, and tidal boundary conditions, highlighting the potential of hybrid models to accelerate real-time flood forecasting. The study illustrates that neural networks can emulate complex hydrodynamic responses when provided with physically meaningful inputs, suggesting that similar hybrid strategies could be valuable for coastal applications if appropriate storm-tide forcings were available. However, the model is trained on urban-scale pluvial–tidal interactions, where flooding is driven primarily by rainfall, storm water infrastructure, and local tidal fluctuations. It does not incorporate hurricane wind fields, storm-track evolution, coastal bathymetry, or large-scale tide–surge coupling, and therefore does not address the challenges associated with predicting coastal storm tide.

1.4 Current Limitations

Despite this progress, there remains a fundamental gap: no existing AI model produces continuous, high-resolution storm-tide flood maps comparable to those generated by process models. Current approaches fall short for several reasons. Point-based surrogates lack spatial coherence and cannot generate inundation maps, while image-based models operate on coarse rasters that cannot resolve estuarine channels, barrier-island morphology, or wetting–drying processes. Multi-fidelity and hybrid CNN–hydrodynamic models demonstrate promising strategies for accelerating physics-based simulations, but they have been applied only to fluvial, pluvial, or urban compound flooding—domains where training data are easier to generate and the governing dynamics differ from tide–surge coupling. As a result, no existing AI-based approach integrates storm evolution, tidal phase, and coastal configuration into a single architecture capable of producing high-resolution storm-tide fields at forecasting timescales. These limitations arise from constraints in training data, model architecture, and the computational challenges associated with large spatial datasets, and they collectively prevent current models from achieving the accuracy and spatial fidelity of process-based hydrodynamic simulations. Addressing this gap requires new methods that can learn fine-scale spatial patterns, operate on unstructured coastal geometries, and capture the nonlinear interactions between storm forcing and tidal dynamics that control coastal inundation.

1.4.1 Availability of Training Data

A central limitation in AI-based prediction is the need for large, diverse, and consistent training datasets. Deep-learning models generalize patterns only when exposed to extensive variation in the underlying data distribution, a requirement emphasized across multiple ML domains (Kiranyaz et al. 2021; Goodfellow et al. 2016; Guo et al. 2016). These studies demonstrate that deep networks learn hierarchical representations when trained on broad datasets, and that limited data diversity leads to overfitting and poor generalization. For AI models that classify or characterize images, natural-image datasets can be augmented using arbitrary transformations like rotations or flips to increase dataset size and variability. Such augmentation strategies are well-established in computer vision (He et al. 2015), where geometric transformations preserve semantic content and improve model robustness.

For storm surge and storm tide prediction, training datasets are often derived from process-model simulations. However, the availability of large and diverse simulation libraries remains limited. Many existing datasets were developed for extreme-event risk analysis and therefore emphasize high-return-period storms rather than the full spectrum of tropical cyclone behavior (Lin et al. 2012; U.S. Army Corps of Engineers 2015; Nakamura et al. 2021; FEMA 2021). As a result, models trained on these datasets may generalize poorly to weaker or shorter-duration storms, or to storms with atypical tracks.

Storm tide prediction introduces an additional requirement: training data must include the astronomical tide and its nonlinear interaction with storm surge. Very few existing simulation libraries incorporate tides (e.g. Cuevas and Dietrich 2025), and none provide large collections of high-resolution storm-tide flood maps suitable for training image-based neural networks. This limits the ability of AI models to learn tide–surge coupling, coastal phase dependence, and the spatial structure of inundation.

Finally, for physics-based image-prediction tasks, the challenges are even more pronounced. Data augmentation techniques used in computer vision are difficult to apply to physics-based images, where arbitrary transformations (e.g. rotations, flips, elastic distortions) would violate conservation laws or distort the physical relationships encoded in the data. In natural-image tasks, such transformations are used to increase dataset diversity because they preserve semantic content and do not alter the underlying class structure (He et al. 2015). In contrast, physics-based fields lack these geometric invariances, meaning that standard augmentation strategies would generate inconsistent samples. These constraints reduce the effective size and diversity of available training data, ultimately limiting model performance and generalization.

1.4.2 Design of AI Model Architectures

Deep-learning architectures vary in prior studies, ranging from simple feed-forward networks (Bezuglov et al. 2016) to convolutional encoders (Lee et al. 2021), autoencoder-based models (Saviz Naeini and Snaiki 2024), and UNet-style encoder-decoder architectures (Pachev et al. 2026). However, there is no established standard architecture for storm surge or storm tide prediction, and many models are adapted from unrelated domains such as image segmentation or time-series forecasting. This lack of standardization makes it difficult to compare approaches or identify which architectural components are most effective for coastal flooding.

Additionally, most existing models are trained using loss functions designed for point-wise regression or coarse image reconstruction. Common choices such as mean absolute error (MAE), mean squared error (MSE), and Huber loss treat each pixel and therefore do not account for spatial coherence, hydrodynamic structure, or the sharp gradients associated with wetting-drying transitions (Goodfellow et al. 2016). Image-based similarity metrics like the Structural Similarity Index Metric (SSIM) introduce perceptual comparisons but may penalize acceptable variations while failing to capture meaningful differences in flood extent or depth (Guo et al. 2016). Other metrics used in computer vision – including peak signal-to-noise ratio (PSNR), cosine similarity, and multi-scale SSIM – likewise emphasize global image similarity rather than the localized, threshold-sensitive behavior that governs inundation patterns (He et al. 2015; Kiranyaz et al. 2021). As a result, even models that perform well on standard loss functions or global error metrics may produce spatial artifacts, unrealistic flooding patterns, or incorrect wet/dry boundaries.

1.4.3 Working with Large Datasets

Producing high-resolution flood maps requires models that operate on large spatial domains with fine resolution. However, training neural networks on such datasets is expensive and memory-intensive. Many prior studies avoid this challenge by predicting only peak water levels at discrete points or by using coarse grids (e.g. 128-pixel × 128-pixel, corresponding to hundreds of meters per pixel over typical coastal domains (Pachev et al. 2026)). At these resolutions, estuarine channels, barrier-island morphology, and neighborhood-scale topography—features with characteristic length scales of tens of meters—are not resolved. As a result, coarse-grid surrogates cannot reproduce the fine-scale inundation pathways that govern storm-tide flooding.

In some cases, models generate point-wise predictions and then construct flood maps as a post-processing step, interpolating or gridding the outputs after inference (Soliman et al. 2025). This strategy is common in ML workflows where dense outputs are reconstructed from sparse predictions (Goodfellow et al. 2016; Kiranyaz et al. 2021, e.g. interpolation-based reconstruction in deep networks). While this approach reduces computational burden during training, it does not guarantee spatial coherence, hydrodynamic consistency, or accurate wetting–drying behavior in the resulting maps. Similar issues are well documented in computer-vision tasks where post-processed outputs may introduce artifacts or distort spatial structure (He et al. 2015; Guo et al. 2016, e.g., upsampling and reconstruction challenges). Even when neural networks perform well on point-based metrics, the post-processed flood maps may exhibit unrealistic spatial patterns or fail to capture fine-scale inundation features critical for emergency management.

Temporal prediction adds further complexity. Modeling the full time evolution of storm tide would require networks capable of learning long-range temporal dependencies across many hours of storm forcing. To reduce computational burden, most existing models focus on peak values rather than full time series. While peak prediction is operationally useful, it limits the ability to capture dynamic processes such as timing of inundation, compound flooding, or tide–surge phase interactions.

1.4.4 Summary and Research Hypothesis

Taken together, these limitations explain why no existing AI model has yet produced continuous, high-resolution storm tide flood maps suitable for emergency management. Training datasets are not large or diverse, architectures are not tailored to the physics of coastal flooding, and computational constraints have prevented the development of models capable of learning fine-scale spatial patterns over large domains.

Research Hypothesis: *A convolutional neural network, whose architecture is suited to learning hierarchical and coherent patterns, can extract the physical structure of storm-tide flooding from a large library of high-resolution simulations and generate accurate flood maps at forecast timescales. By encoding storm-scale forcing and high-resolution coastal morphology in a continuous representation, the architecture is tailored to the physics of coastal flooding–capturing multi-scale interactions, flood-propagation pathways, and the land-water connectivity that governs storm-tide response.*

This hypothesis motivates the development of a new modeling framework whose CNN-based architecture is tailored to the physics of coastal flooding by integrating storm evolution, coastal configuration, and tidal dynamics into a coherent representation capable of producing meaningful storm-tide predictions.

1.5 Funding Acknowledgment

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-ND0001-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. This work is also supported by the Department of Civil, Construction, and Environmental Engineering at North Carolina State University (NC State).

Chapter 2

Background

In this chapter, I describe the foundational components for this study's storm-tide prediction framework – datasets, numerical modeling systems, and software tools that provide the physical and computational basis for training the neural network. Because the performance of any data-driven model is constrained by the quality, structure, and representativeness of its training data, it is essential to define the sources and characteristics of the information used in this study. In the context of storm-tide prediction, this information includes the geographic domain and its geomorphological complexity; the storm library and hydrodynamic simulations that predict total water levels under a wide range of storm conditions; and the data-processing and modeling techniques used to prepare these inputs into machine-learning-ready datasets. Together, these components establish the scientific and computational foundation upon which the neural network is trained, and they frame the assumptions, constraints, and capabilities of the modeling approach developed in this work.

2.1 Study Area

The North Carolina (NC) coast is a complex geomorphological environment and is characterized by extensive estuarine systems, shallow sounds, low-lying mainland topography, and a narrow chain of barrier islands that modulate exchange with the Atlantic Ocean. The Outer Banks – a 320-km barrier island system – plays a critical but increasingly vulnerable role in helping attenuate storm surge and wave energy before it reaches coastal NC from tropical cyclones and energetic wave climates. These barrier islands dissipate wave energy from coastal storms and buffer the mainland, but their low elevations and migratory nature make them prone to overwash and breaching during major storm events. These islands have ground elevations a few meters above mean sea level (OCM Partners 2026; Corbett et al. 2008), with dune ridges in some locations reaching 10–12 m (National Oceanic and Atmospheric Administration 2026), but large portions of the system remain low, narrow, and susceptible to overwash and breaching during major storm events.

Behind the barrier islands lies the Albemarle–Pamlico Estuarine System (APES), the

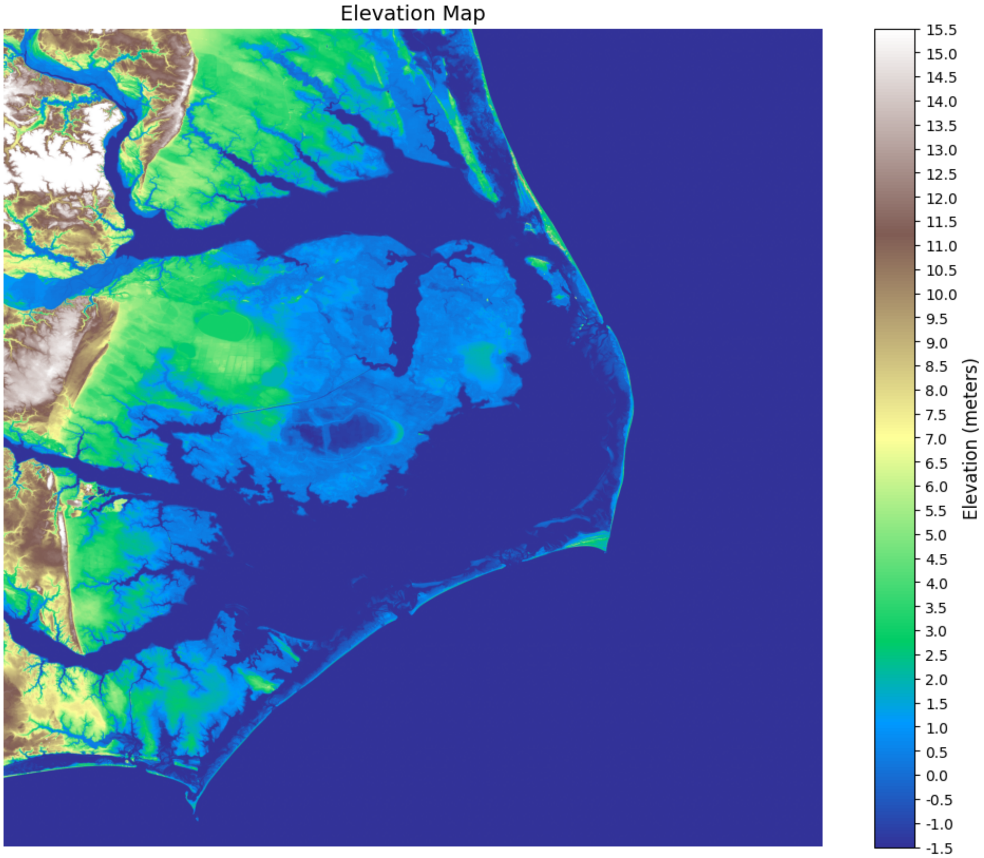
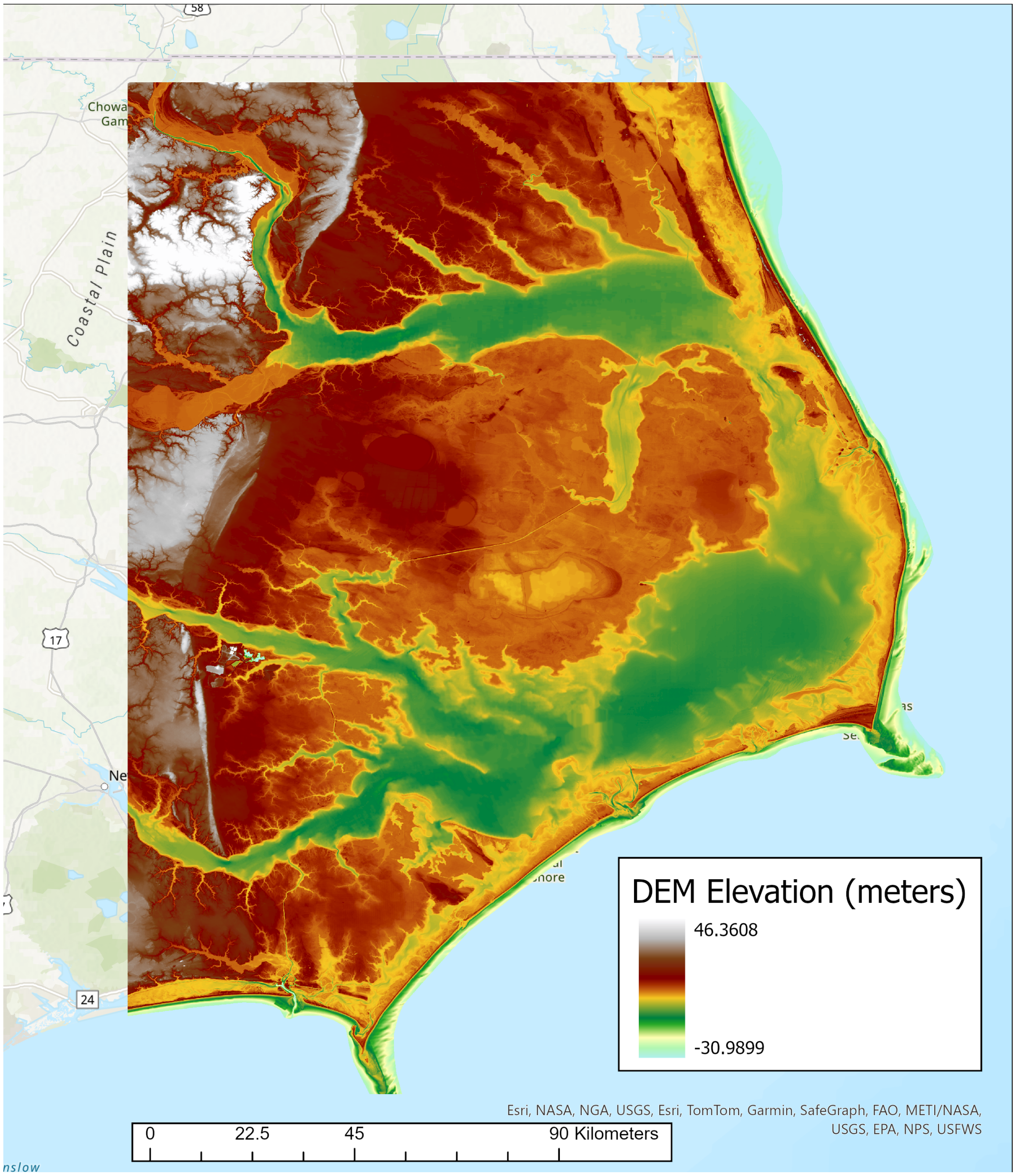


Figure 2.1: Elevation distribution of the ground surface across the study area, spanning -1.5 m to 15.5 m NAVD88. The color bar represents elevation ranges from shallow estuarine bathymetry to higher barrier-island and mainland elevations. Although portions of the estuarine system contain depths greater than -1.5 m, the range shown here is truncated to emphasize the elevations most relevant to water-level dynamics and overland flooding. Elevations are derived from the binned 1-m CoNED Topobathy DEM (OCM Partners 2026).



Esri, NASA, NGA, USGS, Esri, TomTom, Garmin, SafeGraph, FAO, METI/NASA, USGS, EPA, NPS, USFWS

Figure 2.2: Digital elevation model (DEM) of coastal North Carolina stored in EPSG:6346 and rendered at its native 1-m horizontal resolution. Elevations range from a maximum of 46.36 m in the upland interior to bathymetric depths of -30.99 m in the offshore waters. The color-shaded DEM highlights the low-relief coastal plain, barrier-island morphology, and submerged estuarine basins that shape regional hydrodynamics and flood pathways.

second largest estuarine system in the continental United States (Albemarle-Pamlico National Estuary Partnership 2021). Pamlico Sound has average depths of 3 to 8 m, while Albemarle Sound is even shallower, averaging 2 to 6 m (OCM Partners 2026). Hydrodynamic connectivity to the Atlantic Ocean occurs through a small number of tidal inlets – primarily Oregon Inlet, Hatteras Inlet, and Ocracoke Inlet – with typical depths of 6 to 15 m (Army Corps of Engineers 2019). Because these inlets are few and narrow relative to the size of the estuarine system, water exchange is constrained, producing long residence times and strong sensitivity to wind-driven and storm-driven setup (Luettich et al. 2002; Lin et al. 2007). Much of coastal NC's mainland sits at 0 to 2 m (at North American Vertical Datum of 1988 NAVD88), with large expanses of wetlands, agricultural land, and low-lying communities situated only tens of centimeters above mean sea level (Kopp et al. 2015). These low elevations amplify vulnerability to both storm-driven flooding and recurrent tidal flooding under non-storm conditions.

The NC coast is vulnerable to tidal flooding due to its low-lying elevations, but also because storm surge interacts with the astronomical tide. Storm tide can produce peak water levels higher than either component alone, especially when surge arrival coincides with high tide (Federal Emergency Management Agency 2019; Sweet et al. 2022). Prior studies have shown that surge–tide timing can alter total water levels by 0.5–1.5 m along the U.S. Atlantic coast (Horsburgh and Wilson 2007; Thomas et al. 2019). Accurate representations of this interaction are therefore essential for predictive modeling of coastal flooding in NC.

Our study area spans a 2° × 2° region (34.5° to 36.5° latitude, $-75°$ to $-77°$ longitude) including the Outer Banks and Albemarle and Pamlico Sounds, encompassing elevations from 46.36 m to -30.99 m across the DEM domain (Figure 2.2). For hydrodynamic analysis, we restrict this broad elevation range to -1.5 m to 15.5 m (Figure 2.1), isolating the shallow estuarine and low-lying terrestrial surfaces that control storm-driven water-level response, overland flooding pathways, and water-drawback behavior. This study uses the 1851–2020 USGS CoNED Topobathy DEM for the Coastal Carolinas (compiled in 2022), accessed through the North Carolina 1-m CoNED Topobathy dataset (OCM Partners 2026). The DEM provides native 1-m resolution topographic and bathymetric elevations referenced horizontally to NAD83 (2011) and vertically to NAVD88, with all units expressed in meters. We retain the DEM's native 1-m resolution throughout preprocessing to preserve the spatial detail necessary to represent barrier-island morphology, inlet channels, and shallow estuarine bathymetry. Although the DEM is later resampled when interpolated onto the ADCIRC mesh for model-data comparisons, the elevation dataset itself is not downscaled or coarsened. The NC 1-m CoNED Topobathy DEM is a large regional dataset (124.29 GB) that provides continuous, high-fidelity elevation coverage across the study domain.

2.2 Tropical Cyclones in Coastal NC

2.2.1 Observed Storm Tides during Historical Storms

Between 1980 to 2024, coastal NC has experienced 31 tropical cyclones (TCs) causing damages of more than $1 billion (Smith et al. 2020), reflecting its high exposure to TCs and coastal flooding. Across the U.S., TCs are responsible for more fatalities than any other weather-related hazard, with more than 7,000 deaths recorded over this period (Smith et al. 2020). A recent storm that impacted coastal NC was 2018 Florence, which made landfall near Wrightsville Beach on 14 September 2018 as a Category 1 hurricane (National Hurricane Center 2018). Despite its relatively low intensity at landfall, Florence's large wind field and slow forward speed combined to prolong onshore wind forcing along the southern and central NC coastline. This sustained wind stress generated storm-tide water levels of 2.4–3.4 m (National Hurricane Center 2018) along the Neuse River and its tributaries, where easterly winds pushed water toward the western side of Pamlico Sound and backed up river discharge. Although Florence's coastal impacts were severe, its most catastrophic effects occurred inland, where extreme rainfall totals reached 91.3 cm in eastern NC (National Hurricane Center 2018), causing multiple rivers to exceed previous record flood stages. In total, Florence caused $30 billion in damages and resulted in 53 fatalities (Smith et al. 2020), making it one of the most destructive hurricanes in NC's modern record.

Another storm that impacted NC was 2023 Idalia, which made an initial landfall in Florida on 30 August 2023 with maximum sustained winds near 125-130 mph (55-58 m/s) (Smith et al. 2020), reaching Category-4 intensity (Cangialosi and Alaka 2024). By the time it reached NC on 31 August, it had weakened to a tropical storm but still produced sustained onshore winds along the NC shoreline. The storm's forward motion and track kept the circulation close to the coast, which prolonged onshore wind forcing and allowed elevated water levels to persist along the NC barrier islands and estuaries. The coastal-ocean response in NC was driven by the combination of onshore wind stress, low central pressure, and timing with the tidal cycle: tide gauges (Figure 2.3) and local observations recorded a storm tide of 0.3 to 1.2 m (NCEI 2023; Cangialosi and Alaka 2024). The storm surge coincided with a high astronomical tide (a full-moon high tide cycle), which resulted in the storm tide water levels to have increased inundation and extents. This timing amplified water levels along the NC coastline and contributed to $3.6 billion in damages (Smith et al. 2020). The damages from Idalia show that the timing between surge and tide can amplify flooding along the NC coast, making storm tide – not surge alone – the critical variable to understand and predict.

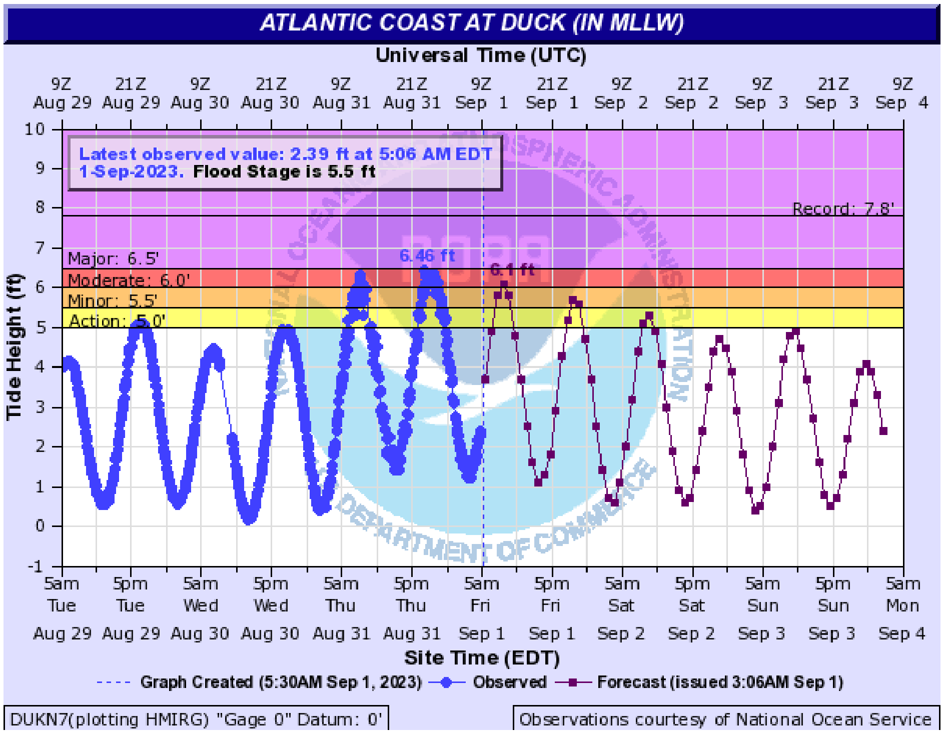


Figure 2.3: Hydrograph at Duck NC during Hurricane Idalia. Observed and forecast water levels at Duck, NC, showing how storm-driven surge interacts with the astronomical tide to produce elevated storm-tide water levels. Peaks occur when surge arrival coincides with high tide, pushing total water levels into minor-to-moderate flooding thresholds. Color-coded flood stages highlight the nonlinear amplification that arises from tide–surge coupling during coastal storms (US Department of Commerce 2023).

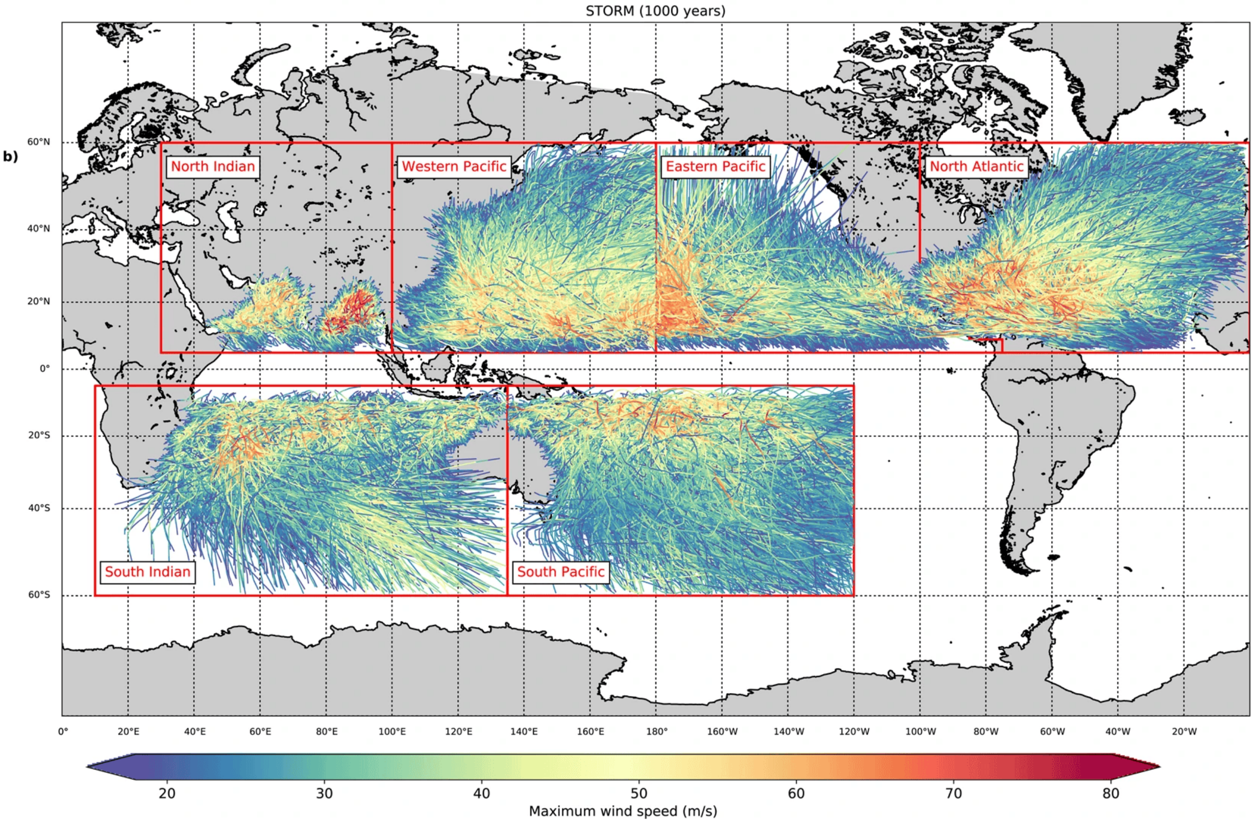


Figure 2.4: STORM dataset that represents a random period of 1000 years of tropical cyclone tracks. Red boxes are the storm basins, and the generated storms for each basin are colored based on their maximum wind speeds (reproduced from Bloemendaal et al. 2020).

2.2.2 Synthetic Storms in Atlantic Basin

Historical tropical cyclone records contain only a few decades of observations, with the International Best Track Archive for Climate Stewardship (IBTRACS; Knapp et al. 2018) providing several hundred storms globally but far fewer within any single basin. This limited sample size constrains the range of storm characteristics available for modeling and often fails to capture the rare, high-impact events most relevant for coastal hazard assessment (Emanuel et al. 2006; Lin et al. 2012; Bloemendaal et al. 2020). As a result, relying solely on historical storms can restrict the diversity of tracks, intensities, and landfall configurations represented in training or evaluation datasets. To address these limitations, coastal hazard studies increasingly use synthetic storms generated from historical observations (Hashemi et al. 2016; Kim et al. 2016; Adeli et al. 2022), which expand the storm catalog and enable exploration of a much broader range of physically plausible tropical cyclone scenarios.

To address these limitations, synthetic storm datasets such as the global Synthetic Trop-

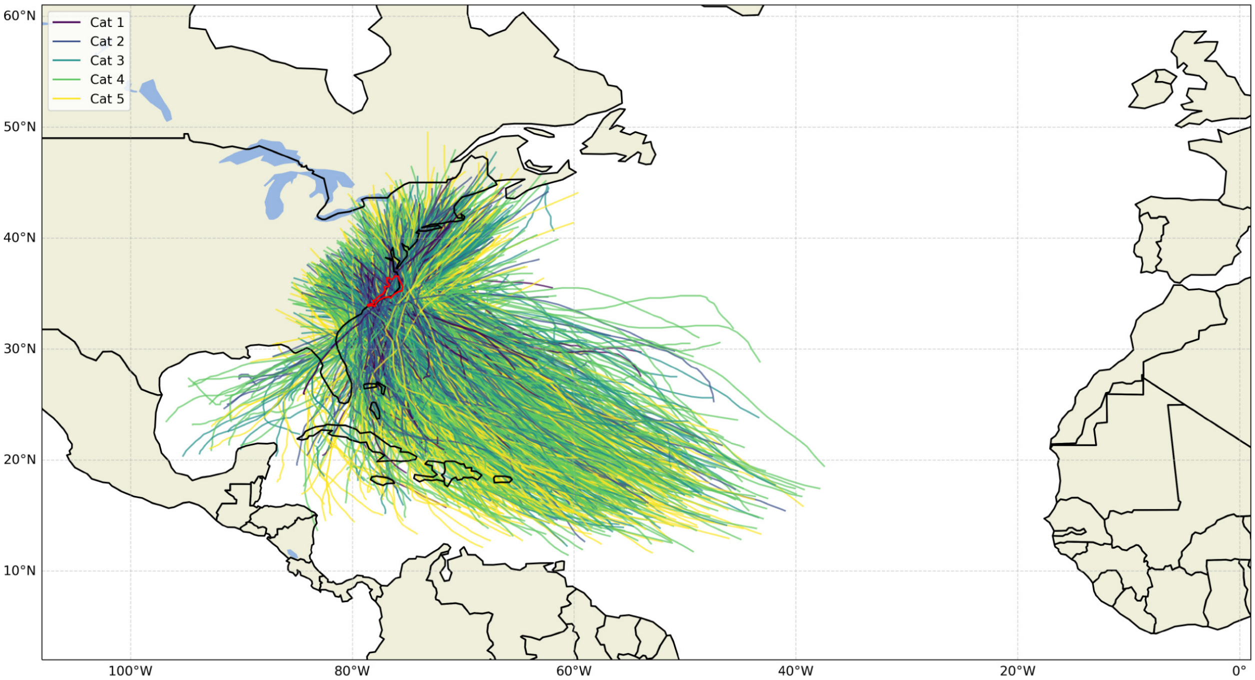


Figure 2.5: Selection of synthetic storms that impact coastal NC, identified by the red polygon (reproduced from Cuevas López et al. 2025).

ical CyclOne geneRation Model (STORM) dataset have been developed to expand the number and diversity of TCs available for analysis (Bloemendaal et al. 2020). The STORM dataset is generated by perturbing historical storms within realistic bounds. STORM converts the multi-decadal IBTrACS baseline into a 10,000 year synthetic archive containing more than 100,000 storms, enabling orders-of-magnitude increases in the frequency of extreme tracks and intensities that are rare or absent in the historical record. Each storm from this dataset (Figure 2.4) includes information about: year, month, TC number, time step (3-hourly), basin, latitude, longitude, minimum pressure, maximum wind speed, radius to maximum winds, category, landfall, and distance to land. Because STORM preserves the statistical structure of real storms while increasing sample size, it provides a more robust basis for training neural networks and evaluating storm surge behavior across a wider range of possible storm conditions. In terms of storm intensity, the STORM dataset reproduces the minimum pressures and maximum wind speeds observed in IBTrACS, with Bloemendaal et al. (2020) noting that “for all basins, these values correspond closely to those found in IBTrACS” and that the MPI-constrained intensity model “succeeds in reproducing the intensities found in IBTrACS.”

2.2.3 Sample of Synthetic Storms affecting Coastal NC

Given the number of synthetic storms within the STORM dataset, a subset of the 100,000 storms is needed because many storms are weak, short-lived, or never approach North Carolina and therefore provide little value for regional hazard assessment. In addition, simulating every storm in the full archive would be computationally prohibitive. Our collaborator Cuevas López et al. (2025) selected storms that impact the NC coast; thus each storm was analyzed to determine if hurricane-strength winds were within the coastline of NC. To approximate the size of the wind field and its interaction with the coastline of NC, the parametric Holland symmetric vortex model (Holland 1980) was used and the distance of the 33 m/s isotach to the coastline was used to determine if the storm would be included in the analysis. This initial reduction created a total of 3,626 'impactful' storms. After removing storms whose maximum wind at closest approach was smaller than 33 m/s (1,792 storms removed) and discarding storms shorter than 24 hours, the final sample comprised 1,813 storms (Figure 2.5).

The selected storms exhibit substantial diversity: durations range from 1.25 to 9.8 days with a mean of 6.76 days and a standard deviation of 3.72 days; maximum wind speeds in the sample span from just below 33 m/s to 83.5 m/s, with an extreme outlier showing a minimum central pressure near 840 hPa; most tracks approach NC from the southwest while 5% are parallel storm tracks. This targeted subset therefore preserves the broad range of approach angles, forward speeds, and intensity-size combinations relevant to NC surge risk while substantially reducing the computational burden of downstream hydrodynamic simulations (Cuevas López et al. 2025; Bloemendaal et al. 2020).

2.3 ADCIRC

ADvanced CIRCulation (ADCIRC; Luettich et al. 1992; Westerink et al. 1993) is a coastal hydrodynamic model that solves the generalized wave-continuity equation (GWCE) using finite-element discretizations on unstructured meshes (Luettich et al. 1992; Westerink et al. 2008). Its ability to represent complex coastlines, variable spatial resolution, and long simulation periods makes it well suited for modeling storm-tide dynamics in coastal NC (Cyriac and Dietrich 2018; Van Cooten et al. 2011). ADCIRC meshes can be applied in complex coastal environments for varied sizes for computational efficiency over paralleled computing environments (Dietrich et al. 2011; Tanaka et al. 2011; Pringle et al. 2021).

Simulating each synthetic storm in ADCIRC required three core inputs – unstructured mesh, tidal time series, and atmospheric forcing – allowing the model to compute water levels throughout the domain and generate a maximum water-elevation surface for every storm.

2.3.1 Unstructured Mesh to Describe Coastal NC

An unstructured mesh was used for coastal North Carolina, with about 70% of its resolution concentrated in coastal NC to reduce computational cost while retaining accuracy. This mesh was developed from the SABv3-60m mesh created by (Woodruff 2023), incorporating the most recent nearshore bathymetry and topography from the CUDEM dataset (CIRES 2014). The original SABv3-60m mesh contains 6.8 million elements and 3.5 million vertices, but because it was not designed for high-resolution modeling in coastal NC, a new mesh – referred to as SABv5 – was constructed.

Our collaborator, Cuevas López et al. (2025) developed SABv5 by removing regions with high resolution outside NC to reduce computational expense. Floodplains and estuaries from Florida through South Carolina were trimmed, resulting in a mesh with similar resolution to SABv3-60m within NC but with only 2.8 million elements and 1.4 million vertices. Element sizes range from 60–100 m in areas such as Oregon Inlet and the Cape Fear River, and 100–200 m in front of and behind the barrier islands (Figure 2.6. The mesh uses the same nodal attributes as SABv3-60m, including parameters for numerical damping, Manning's n, eddy viscosity, GWCE advection control, surface roughness, and wind-canopy coefficients. These attributes are applied across all simulations.

2.3.2 Tide Forcing

Harmonic tidal constituents are provided at the model's open-ocean boundary, which ADCIRC used to reconstruct the astronomical tide time series during each simulation. After identifying July–August 2001 as a representative tidal window, Cuevas López et al. (2025) sampled the tidal record to generate a unique set of astronomical tidal conditions for every storm by selecting constituent phases and amplitudes consistent with the chosen two-month window.

For each of the 1,813 storms, a start day and hour were selected at random within the two-month period, with the constraint that the full storm duration (including the 3-day ramp-up) fit within the window. Because the STORM tracks are provided at 3-hr intervals, the assigned start time defined the complete simulation time range. After the simulation window was set, the tidal harmonic constituents were sampled at 1-s intervals to match the ADCIRC model timestep. The tidal signal at each ocean-boundary node was reconstructed using the eight principal tidal constituents (M_2, S_2, N_2, K_2, K_1, O_1, P_1, Q_1), with constituent amplitudes and phases taken from the mesh boundary and the time-dependent node

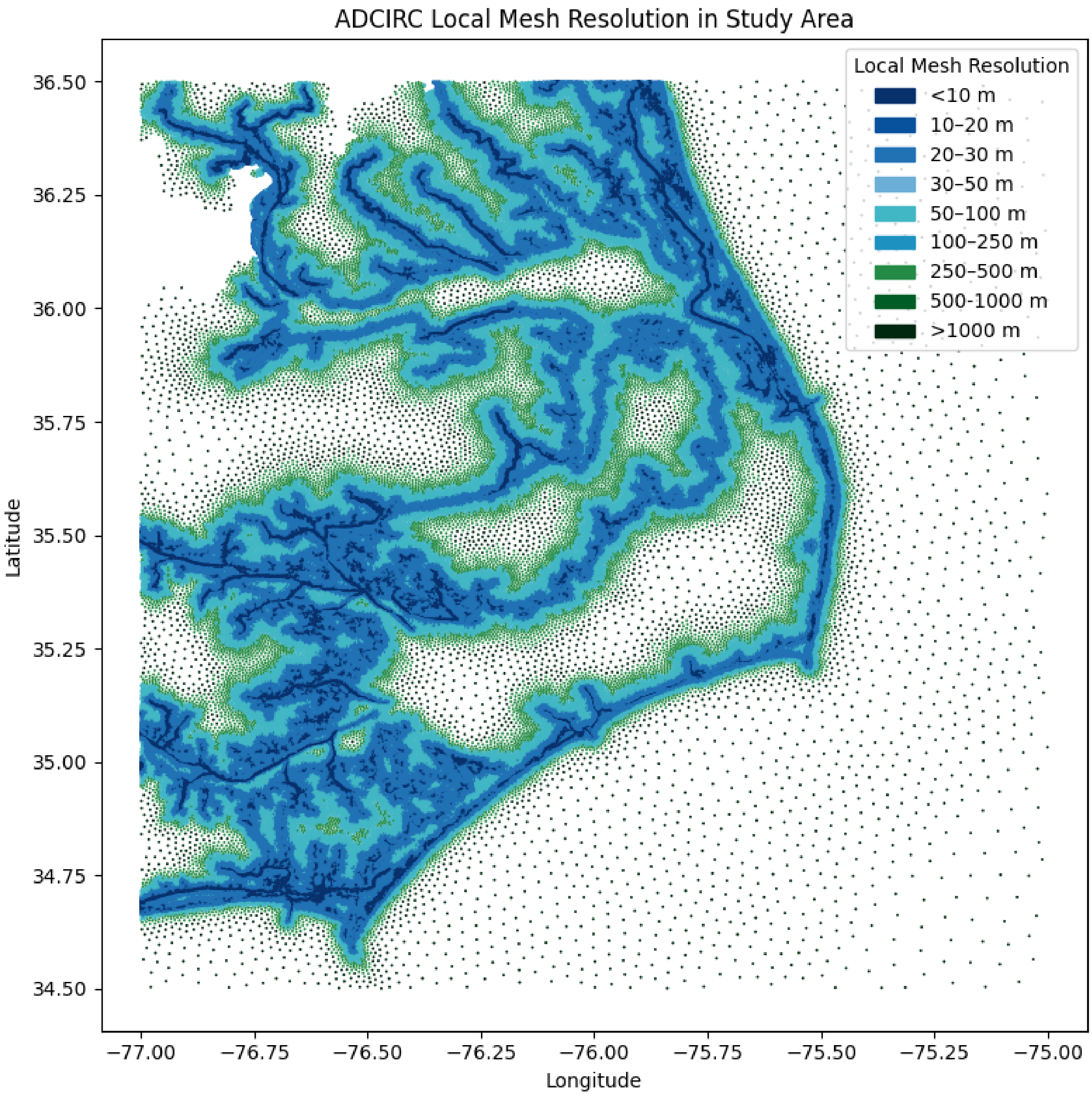


Figure 2.6: Local mesh resolution of the ADCIRC computational grid across the study area. Colors indicate the characteristic element size, with darker shades (blue colors) representing finer resolution near the coastline, inlets, and estuarine channels, and lighter shades (green colors) indicating coarser offshore elements. This spatially varying resolution reflects targeted refinement in dynamically important regions.

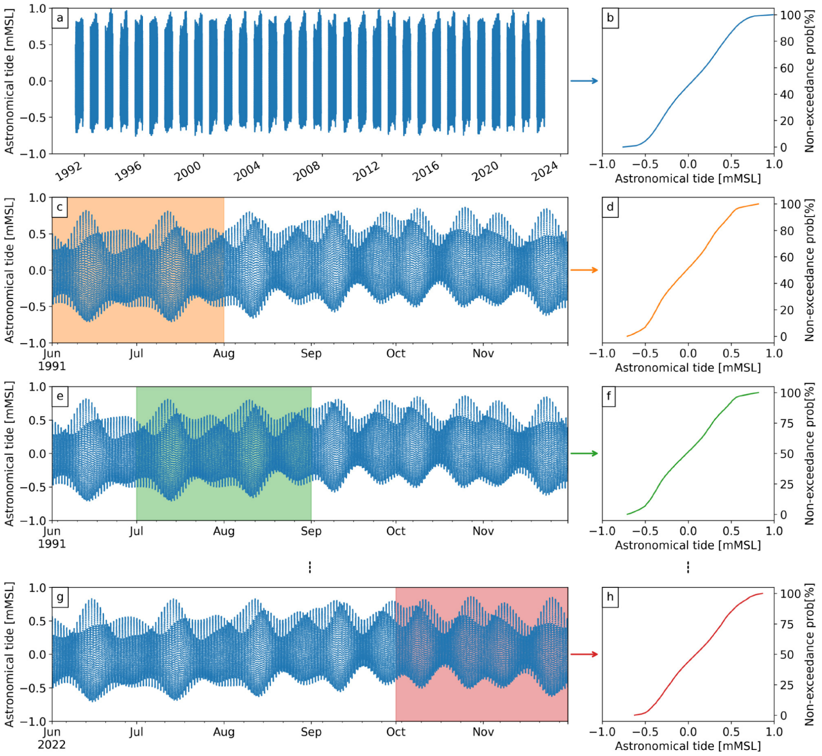


Figure 2.7: Tide sampling added to ADCIRC simulations that represent the two-month period for astronomical tides at Duck NOAA tide gauge (reproduced from Cuevas López et al. 2025).

factors and equilibrium arguments computed for the assigned start date. To match the ADCIRC model timestep, the reconstructed tidal signal was evaluated at 1-second intervals during the simulation.

This procedure produced a continuous tidal time series for each storm that was phase-consistent with the randomly assigned date. Because each storm received a different start time, the ensemble of simulations captures a wide range of tide-surge interactions, including cases where storm arrival coincides with high tide, low tide, or intermediate phases. A tides-only simulation for the same two-month period was also run to provide a reference astronomical tide time series at all mesh nodes.

2.3.3 Atmospheric Forcing

To simulate each storm's effects, ADCIRC must represent the wind and pressure perturbations across the full model domain. The synthetic STORM tracks provide storm-center parameters at 3-hour intervals, so ADCIRC's built-in model to generate full-domain wind and pressure fields from those center parameters was used. The parametric Holland symmetric vortex model (Holland 1980) provides a solution to the gradient-wind balance and approximates the surface pressure profile with a rectangular-hyperbola form whose scaling parameters are derived from V_{max} and R_{max}. This formulation requires parameters available in the synthetic dataset and is therefore well suited for large-scale, reproducible surge studies.

Inputs taken from the STORM records included 3-hourly storm center position, minimum central pressure, maximum sustained wind, and radius to maximum winds, which were formatted into Automatic Tropical Cyclone Forecast (ATCF) style meteorological files. This postprocessing workflow and the generation of the storm-specific ATCF forcing files were implemented by our collaborator Cuevas López et al. (2025). ADCIRC ingested these ATCF files and reconstructed surface wind and pressure fields on the mesh; those fields were converted to wind stress and surface pressure forcing that drive the hydrodynamic response.

2.3.4 Predictions of Peak Storm Tides

Each ADCIRC simulation produces a field of maximum water elevations across the computational mesh, representing the highest total water level reached at each vertex during the storm-tide event. These outputs represent the combined effects of storm surge, tide, wind setup, and local hydrodynamics, and they provide a continuous prediction of peak flooding for each of the 1,813 synthetic storms. For each storm, the location and magnitude of the peak water level were recorded, enabling spatial and statistical characterization of extreme storm-tide responses along the NC coast. These maximum water-elevation fields form the target dataset used to train the neural network model developed in this study.

ADCIRC reports peak values at mesh vertices, and our neural network does accept images as inputs, but those images must be reproducible, consistently scaled, and tied to physical units and georeferencing. Our collaborator produced grayscale raster exports from Kalpana (Cuevas and Dietrich 2025), but we did not use those rasters directly because Kalpana's default exports used variable value-mapping between storms, producing inconsistent pixel quantization and dynamic range that would break reproducibility and bias training. In addition, the Kalpana rasters lacked explicit mesh-to-raster provenance and the fixed, documented scale factors we require, so pixel values could not be unambiguously converted back to floating-point water levels (meters) without storm-by-storm reverse-engineering.

Kalpana's rasters could be repurposed only after (a) restoring accurate georeferencing, (b) converting pixel values back to physical units with documented scale factors, (c) matching the target grid and resolution, and (d) attaching the required metadata and masks – steps that were not performed on the original grayscale outputs and therefore make them unsuitable as

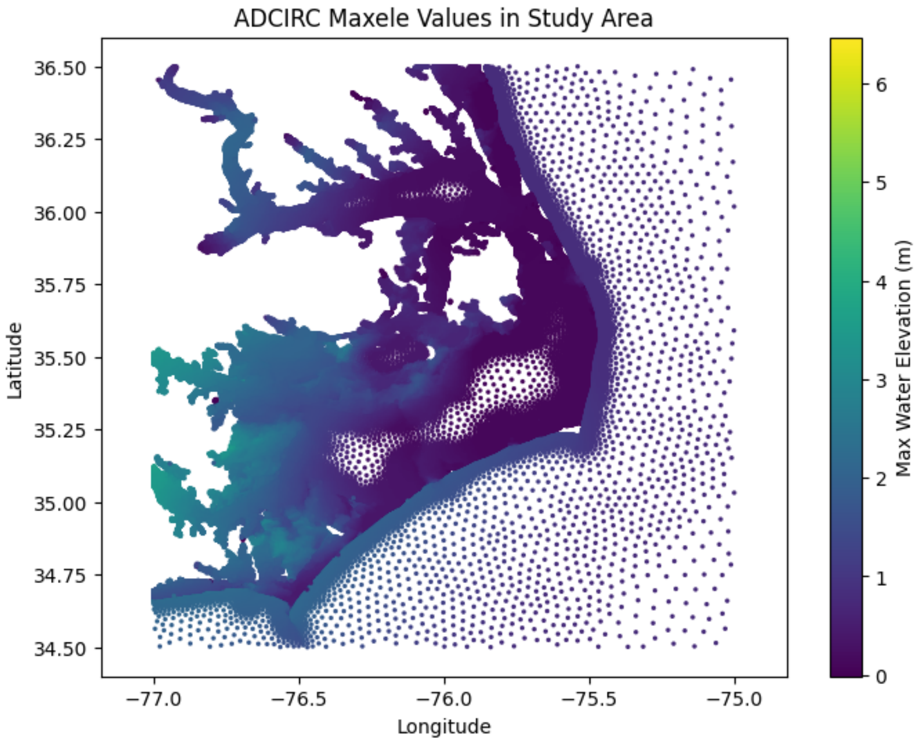


Figure 2.8: Maximum water elevation (`maxele`) field for Storm-0 from the ADCIRC simulation, shown on the native unstructured mesh nodes. Colors indicate the peak water level at each wet node across the study area, highlighting spatial variability in storm-driven surge. Dry-mesh regions appear as gaps where ADCIRC does not compute water levels.

direct training targets. For these reasons, we instead downscale and regrid ADCIRC vertex maxima into georeferenced floating-point rasters (GeoTIFFs) or regular arrays that preserve units, precision, and spatial referencing, include land/water masks and metadata, and match the input format expected by the neural network. This process is described in Section 3.1.

2.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of deep learning models designed to exploit the spatial structure of grid-organized data (Goodfellow et al. 2016). At their core, CNNs apply learned convolutional kernels across an input to produce feature maps that encode local patterns and their spatial relationships; stacking many such layers yields a hierarchical representation in which early layers capture simple, local features and deeper layers capture abstract, large-scale structures (Zeiler and Fergus 2013). Early work established the viability of gradient-based training for convolutional architectures and their effectiveness on visual tasks, and later large-scale demonstrations showed how deep CNNs improved image recognition performance (Krizhevsky et al. 2012; Lecun et al. 1998). Because convolutions are translation-equivariant and parameter-efficient relative to dense connections, CNNs are particularly well suited to image processing, dense prediction, and any task that requires preserving spatial correspondence between input and output (Guo et al. 2016; Gu et al. 2018).

In this study, we use one-dimensional convolutional layers to learn temporal relationships in storm-track time-series predictors, and two-dimensional convolutional layers to extract spatial features from high-resolution elevation fields. These components are integrated into a dual-stream architecture that fuses temporal and spatial information for storm-surge prediction.

2.4.1 One-Dimensional Convolutional Layer

One-dimensional convolutional layers slide a set of 1D kernels along a single axis of the input and are therefore appropriate for sequence and time-series data (Bai et al. 2018). In practice, 1D convolutions are used to learn local temporal patterns, short-range dependencies, and multiscale temporal features with far fewer parameters than many recurrent architectures (Kiranyaz et al. 2021). Stacking 1D convolutional layers increases the effective temporal receptive field while remaining computationally efficient (Bai et al. 2018; Lea et al. 2017). Because 1D convolutional blocks produce compact latent encodings, they are used to preprocess or encode temporal predictors (for example, tide or wind time series) before fusing those encodings with spatial representations in a downstream 2D network (Badrinarayanan et al. 2016; Borovykh et al. 2018; Ismail Fawaz et al. 2019).

2.4.2 Two-Dimensional Convolutional Layer

Two-dimensional convolutional layers apply 2D kernels across height and width and form the backbone of image models (Gregor et al. 2015; Lecun et al. 1998). Typical design uses small kernels (for example 3×3) stacked in depth so that the network attains a large effective receptive field while keeping parameter counts manageable (Figure 2.9); nonlinear activation functions (most commonly Rectified Linear Unit (ReLU); Agarap 2018) introduce nonlinearity and sparsity, and normalization layers (batch or group normalization) stabilize and accelerate training. In a multi-layer stack, early 2D conv layers learn low-level features such as edges and texture, intermediate layers combine those into motifs and shapes, and deep layers capture semantic or context-level patterns; skip connections and multi-scale fusion are often used to preserve fine spatial detail while aggregating broad context (Ronneberger et al. 2015; He et al. 2015; Badrinarayanan et al. 2016).

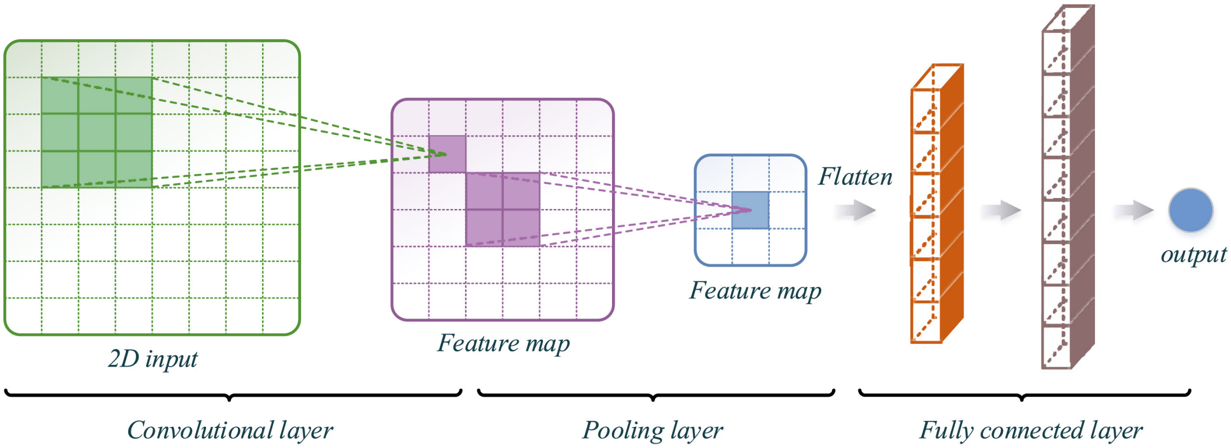


Figure 2.9: A schematic overview of a convolutional neural network (CNN) architecture. The 2-D input field is processed through a convolutional layer to extract spatial features, followed by a pooling layer that reduces spatial resolution. The resulting feature maps are flattened and passed through fully connected layers to produce the final output (reproduced from Zang et al. 2020).

2.4.3 Pooling Layer

Pooling layers reduce the spatial resolution of feature maps to increase translational invariance and to enlarge the network's receptive field with modest computational cost (Gu et al. 2018). Common pooling operations include max pooling, which preserves the strongest activation in a local window, and average pooling, which computes a local mean; both reduce spatial size and can help control overfitting (Guo et al. 2016; Galanis et al. 2022). Because pooling discards spatial detail, modern dense-prediction architectures often use small pooling factors, strided convolutions, or carefully placed pooling combined with skip connections so that fine-scale structure can be recovered in the decoder stage (Long et al. 2015; Ronneberger et al. 2015).

2.4.4 Dense Layer

Dense (fully connected) layers connect every input unit to every output unit and are effective for global reasoning and classification when spatial structure is not required in the output (Goodfellow et al. 2016). In image-to-image or pixelwise regression tasks, however, dense layers are avoided at the output stage because they collapse spatial correspondence and increase parameter counts; instead, convolutional designs replace dense layers with convolutional equivalents so the model can produce coherent outputs and operate on variable-sized inputs (Long et al. 2015). Dense layers remain useful in intermediate multimodal fusion blocks or when a compact global descriptor is required.

2.4.5 Data Augmentation

Data augmentation expands the effective size and diversity of a training set by applying label-preserving transformations to inputs; common image augmentations include flips, small rotations, translations, crops, and modest intensity perturbations (Gu et al. 2018). For geospatial or physically meaningful imagery, augmentations must preserve georeference and physical plausibility, so transformations are chosen conservatively and applied consistently to inputs, targets, and any auxiliary masks. Augmentation reduces overfitting, improves generalization to unseen conditions, and can be combined with normalization, fixed binning/resolution, and careful metadata management to ensure reproducibility and physical interpretability of model inputs and outputs (Shorten and Khoshgoftaar 2019; Ronneberger et al. 2015).

Chapter 3

Methods

This chapter describes the complete workflow used to transform hydrodynamic simulations and synthetic storm information into the spatial and temporal inputs required for training a neural network that predicts peak storm-tide flooding. The process begins by converting ADCIRC's peak water levels from its unstructured mesh into DEM-aligned raster surfaces and encoding them as grayscale flood images, while a parallel workflow generates dry-land elevation images from the DEM to provide terrain information where ADCIRC does not compute water levels. These spatial inputs are then tiled and augmented to create a large, diverse training set that represents both broad coastal patterns and fine-scale shoreline variability. In parallel, a 25-hour window of storm evolution is extracted for each tile and encoded into a structured time series of 35 parameters, forming a compact temporal representation of storm intensity, motion, geometry, and tidal forcing. The neural network integrates these two input streams – a 512-pixel × 512-pixel dry-land mask representing local topography and a 25×35 temporal sequence representing storm evolution – and learns to transform them into a corresponding 512-pixel × 512-pixel continuous elevation image. Through this dual-stream architecture, the model learns how the evolving characteristics of a storm interact with coastal morphology to produce the spatial distribution of storm-tide flooding.

3.1 Storm-Tide Maps as Raster Images

To train a convolutional neural network to predict storm-tide flooding, the model must learn from raster images. However, ADCIRC does not output rasters; it produces water levels on an unstructured triangular mesh. Therefore, a post-processing workflow is required to convert mesh-based maximum storm tide values into DEM-aligned raster images that preserve geospatial accuracy, physical meaning, and consistency across storms.

3.1.1 Maximum Storm Tide Downscaling to Raster

The downscaling procedure is the full sequence of steps that transforms ADCIRC's predictions of peak storm tides from its unstructured mesh into a regularly gridded raster aligned with the study-area DEM. This procedure includes loading inputs, ensuring a shared coordinate system, constructing a target grid, interpolating mesh values, and enforcing physical bounds. The procedure converts ADCIRC's predictions of peak storm tide from its unstructured mesh into a georeferenced raster that is aligned with the clipped DEM (shown in Figure 2.2). First, both inputs are loaded in their natural format (netCDF for the ADCIRC outputs) and the vertex coordinates and maximum storm tides are read from the innate variables (x, y, and $\zeta_{\max}$). The clipped DEM is opened with the Python package `rasterio` (Gillies et al. 2013) to read its inherent CRS (coordinate reference system), transform, bounds, width, and height. This initial loading step establishes the raw data arrays and metadata that define the spatial domain for all subsequent reprojection, interpolation, and raster generation.

The unstructured finite-element mesh spans the full model domain, with finer resolution along the coastline and coarser resolution offshore (Figure 2.6). The unstructured, finite-element mesh spans the full model domain, with finer resolution along the coastline and coarser resolution offshore (Figure 2.6). Maximum storm-tide elevations are computed on this mesh but are only reported at wet nodes (Figure 3.1). These dry regions appear as gaps in the mesh because ADCIRC only reports values at wet vertices. This distinction between wet and dry regions is central to the downscaling workflow: interpolation can only occur within the convex hull of wet nodes, and all remaining gaps must later be filled using the dry-land DEM.

Because the ADCIRC mesh is defined in geographic coordinates (EPSG:4326, Geodetic longitude/latitude for the World) and the DEM is stored in a projected CRS (EPSG:6346, Geodetic longitude/latitude for the U.S.), the two datasets start in different spatial reference frames. The DEM is reprojected from the ADCIRC coordinate system so that both datasets share the same geographic reference frame. The reprojected DEM is then used together with the ADCIRC vertex coordinates as the common spatial grid for interpolation. This approach ensures that interpolation is performed in geographic coordinates and the final storm-tide raster is written using a consistent affine transform and CRS (EPSG:4326).

Following this, the DEM is not only reprojected but also resampled onto a standardized storm-tile grid. The DEM data is updated and resampled using the Python package `rasterio.reproject` (Gillies et al. 2013), which applies a bilinear interpolation to produce a `float32` elevation array on the new grid. This step is necessary because the DEM's original 1-m projected grid does not align with the coarser, uniform grid used for storm-tile processing. Reprojecting and resampling the DEM into EPSG:4326 therefore produces a consistent geographic grid on which all storms can be interpolated and compared.

With the reprojected DEM, the geographic coordinates of every pixel center are computed to define the interpolation target grid. The destination affine transform is used to derive pixel-

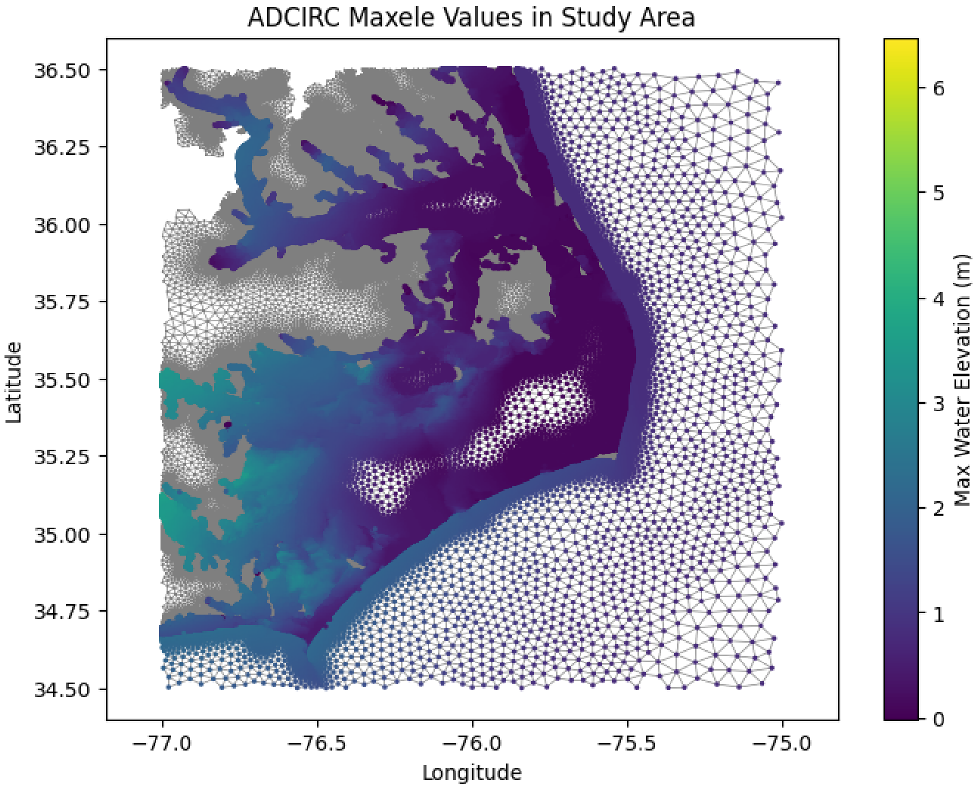


Figure 3.1: Maximum storm tide (`maxele`) output from ADCIRC for Storm 0, shown as the set of wet mesh vertices above the ADCIRC mesh (gray lines). Each point represents a vertex where ADCIRC computes a maximum water elevation, while the gray regions correspond to dry areas where no water level is defined.

center coordinates by using a Python function `xy(transform,row,col)` in a nested loop to populate `dem_lon` and `dem_lat` arrays, and these arrays are flattened and column-stacked to form the regular set of query points. In practical terms, the DEM's affine transform is used to determine the longitude and latitude of each pixel, producing a regular grid of query points where ADCIRC values will be estimated. The ADCIRC vertex triplets (x_i, y_i, ζ_i) are interpolated onto those query points using SciPy's `griddata` with `method='linear'` (Virtanen et al. 2020), producing a continuous surface that is reshaped to the DEM grid shape. Because the ADCIRC mesh is unstructured, griddata fills the convex hull of the node set and returns `NaN` outside that hull.

Post-processing enforces physical bounds and a `nodata` convention in dry regions before writing the georefrenced raster. The implementation defines a `nodata` value (-9999) and applies a physically reasonable water elevation range ζ_{range} (e.g. -2 m to 10 m); any interpolated pixel that is `NaN` or outside this range is replaced with the `nodata` value. Finally, the downscaled masked surface is written as a GeoTIFF using the reprojected DEM's affine

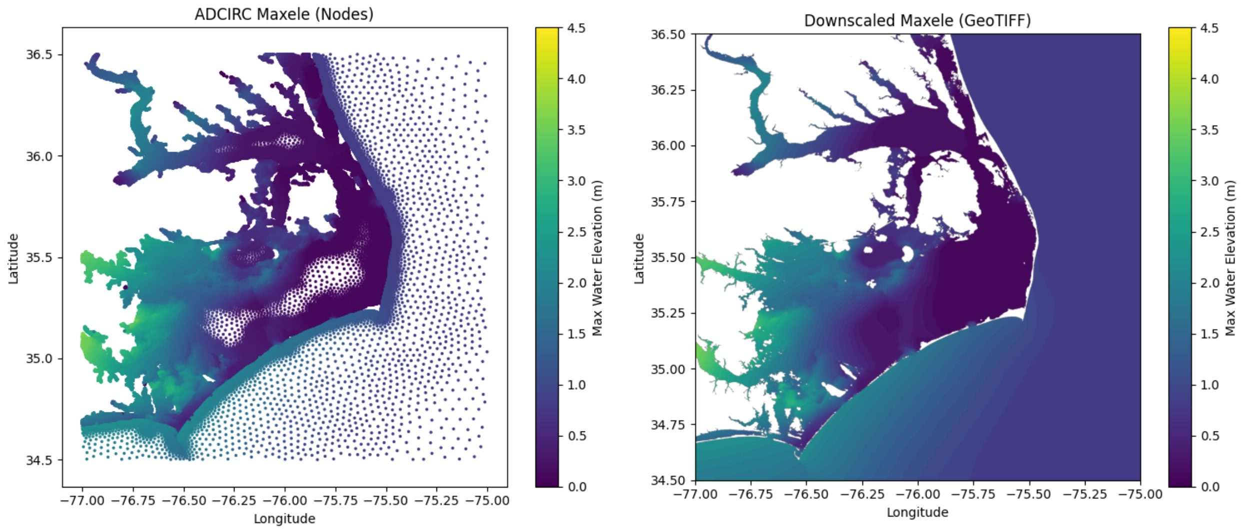


Figure 3.2: Comparison between the raw ADCIRC maximum water-level output (left), defined at the vertices of an unstructured, finite element mesh, and the DEM-aligned GeoTIFF generated through linear interpolation (right). The interpolated raster adopts the DEM's grid, affine transform, and CRS, producing a continuous elevation surface suitable for downstream grayscale encoding and machine-learning workflows.

transform and CRS (Figure 3.2). The resulting GeoTIFF is a reproducible, DEM-aligned raster of ADCIRC $\zeta_{\max}$ values in EPSG:4326 and serves as the intermediate product for all subsequent image generation.

3.1.2 Grayscale Image Process

The motivation for converting the downscaled storm-tide rasters into grayscale images is twofold. First, grayscale encoding provides a compact, standardized representation of the continuous elevation field that is compatible with convolutional neural networks, which expect fixed-size image inputs. Second, the `uint8` format reduces storage requirements and I/O overhead compared to floating-point GeoTIFFs, enabling efficient batching, shuffling, and loading during model training. By mapping water elevations into a consistent 0-255 range, the images preserve the spatial structure of the storm-tide field while remaining lightweight enough to support large-scale machine-learning workflows.

The conversion from floating-point elevation values to `uint8` follows a deterministic linear mapping. Native `nodata` pixels are reassigned so that missing values are encoded as 255 (Figure 3.3). Valid elevations are shifted by +1.5 m so that the minimum expected water level (–1.5 m) maps to zero, and then multiplied by an amplification factor to enhance contrast in the water-level gradients. After scaling, the array is clipped to the 0–254 range and cast to `uint8`, reserving 255 exclusively for nodata. This mapping ensures that the forward and inverse transformations are reproducible, allowing the grayscale images to be decoded back to physical units if needed.

For each valid elevation value z (in meters), the corresponding grayscale pixel value d is computed as

$$d \;=\; \begin{cases} 255, & \text{if the pixel is nodata,} \\ \\ \operatorname{clip}(\alpha\,(z+1.5),\, 0,\, 254)\,, & \text{otherwise,} \end{cases} \tag{3.1}$$

where α is the amplification factor and $\operatorname{clip}(x, 0, 254)$ truncates values outside the 0–254 range. The value 255 is reserved exclusively for nodata.

The inverse mapping, used to recover physical elevations from valid grayscale pixels ($0 \leq d \leq 254$), is

$$z \;=\; \frac{d}{\alpha} - 1.5. \tag{3.2}$$

For each storm, the full $2^\circ \times 2^\circ$ domain is exported as a single 4096×4096-pixel grayscale image, which then allows smaller 512-pixel × 512-pixel patches to be sampled across the area of interest. Because this export represents a substantial downscaling from the original 1-m DEM, much of the fine-scale topographic structure—such as narrow dunes, small channels, and sharp elevation gradients—is smoothed or lost in the coarser grid. Across all storms, the resulting 1,813 images (each 4096-pixel × 4096 pixels) were stored in an HDF5 file with a total size of approximately 450 MB.

3.1.3 Dry-Land Input

A dry-land image is required to provide elevation values in locations where ADCIRC does not compute water levels. Because ADCIRC reports storm-tide elevations only at wet mesh vertices, large portions of the domain – particularly inland areas – contain no modeled values, (e.g. Figure 3.2). To create a continuous elevation surface suitable for image generation and machine-learning training, these dry regions must be filled using the underlying DEM. Encoding the dry-land DEM into the same `uint8` convention as the flood images ensures that the two datasets can be merged seamlessly.

The dry-land image is derived from the original DEM and encoded using the same linear mapping as the flood images. Elevations above 15.5 m are clipped to limit the dynamic range, and the mapping $z = (p/15) - 1.5$m is applied so that pixel value 0 corresponds to –1.5 m and pixel value 255 corresponds to 15.5 m. This yields a vertical resolution of 1/15 m (approximately 0.0667 m) per grayscale level. Unlike the flood images, `nodata` is handled explicitly rather than by reserving a pixel value. The resulting dry-land image provides a

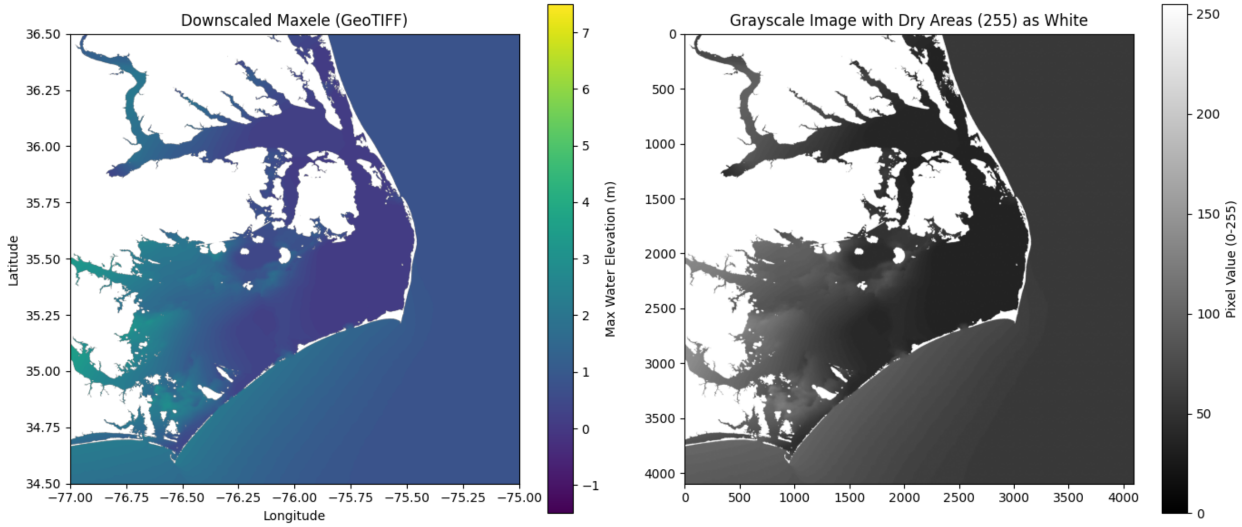


Figure 3.3: Example of the grayscale-encoding process. The DEM-aligned storm-tide GeoTIFF (left) is transformed into an 8-bit grayscale image (right) by mapping physical water elevations into the 0–254 range and assigning 255 to nodata pixels. This encoding preserves spatial structure while reducing file size and enabling efficient CNN training.

complete, DEM-based elevation field that can be used to fill gaps in the interpolated ADCIRC surface.

3.1.4 Storm Tide Output

ADCIRC outputs provide water elevations only at wet mesh vertices, leaving dry locations undefined. After interpolation onto the DEM grid, these undefined pixels appear as `NaN` values (encoded as 255 in the grayscale representation). To produce a continuous, wall-to-wall elevation surface, every pixel where the interpolated ADCIRC field is missing is replaced with the corresponding value from the dry-land image (Section 3.1.3). Because both the flood and dry-land images use the same `uint8` encoding, this substitution can be performed in the encoded domain or, equivalently, by decoding both images to meters, filling missing values, and re-encoding the merged surface (this varied depending on the scale value from meters to pixel).

This merging step removes artificial discontinuities at the wet/dry interface and ensures that the final storm-tide raster contains modeled water elevations where ADCIRC provides wet values and DEM-derived elevations elsewhere (Figure 3.4). The resulting continuous elevation surface is then scaled into the uint8 format used for all storm-tide images, rather than written to a GeoTIFF. This uint8 array serves as the DEM-aligned representation of maximum storm tide used for subsequent grayscale image generation and machine-learning

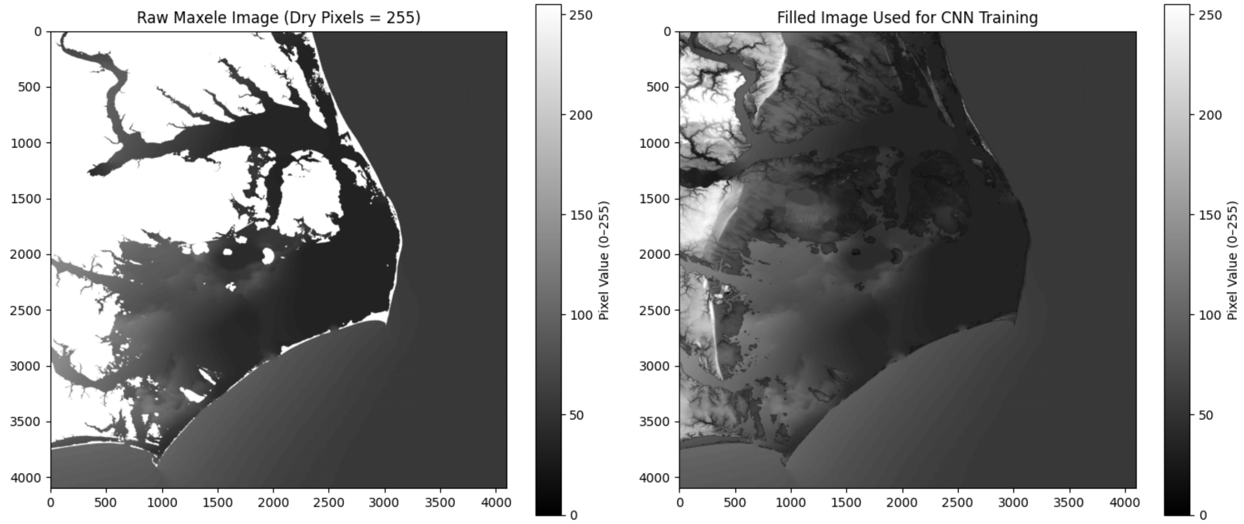


Figure 3.4: Example of the preprocessing step used to create the CNN training targets. Left: the interpolated ADCIRC maxele image, where dry regions appear as white gaps because ADCIRC reports values only at wet mesh vertices. Right: the same image after filling dry pixels with elevations from the dry-land DEM, producing a complete, continuous raster. Smaller 512 × 512 tiles extracted from this filled raster serve as the inputs and prediction targets for the CNN.

training.

3.2 Augmentation of Storm-Tide Maps

Deep-learning models require large, diverse training datasets to generalize, especially when the target phenomenon exhibits strong spatial heterogeneity. Although our storm ADCIRC catalog contains 1,813 high-resolution storm-tide maps, each map represents only a single realization of a complex coastal flooding response. Training directly on full 4096×4096 images would dramatically limit the number of effective training samples and would overweight broad, low-variability regions (e.g. offshore areas or consistently dry inland terrain). To increase the number of training examples and ensure that the model learns both large-scale patterns and fine-scale shoreline dynamics, each storm-tide map is decomposed into a set of storm-tide map tiles.

Our augmentation strategy combines two sampling approaches to create a balanced, meaningful training dataset:

- **A uniform grid of 64 tiles** that covers the entire domain. These tiles ensure that the model is exposed to the full spatial extent of the coastal system, including offshore

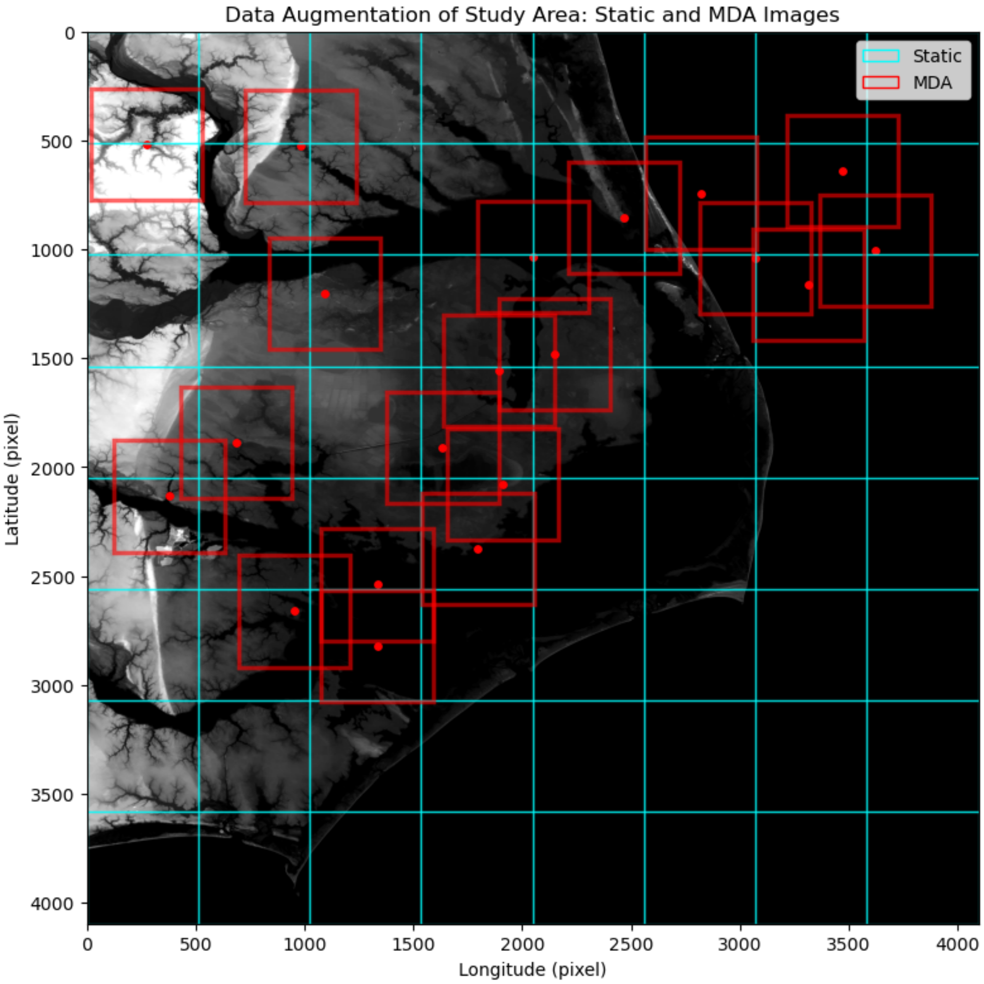


Figure 3.5: Tile-selection strategy used for storm-tide map augmentation. The full 2° × 2° domain is partitioned into a uniform grid of 64 tiles (cyan boxes), providing complete spatial coverage. An additional 20 tiles (red boxes) are centered on locations selected by a Maximum Dissimilarity Algorithm (MDA) applied to the wetness-variability field, emphasizing regions where the wet/dry interface shifts across storms. Together, these 84 tiles form the augmented representation of each storm-tide map used for model training.

regions, inland areas, and low-variability zones.

- **An additional set of 20 tiles** whose centerpoints are chosen using a Maximum Dissimilarity Algorithm (MDA) applied to the wetness-variability field (described in Section 3.2.2). These tiles emphasize the dynamic wet/dry front, where storm-tide gradients are strongest.

Together, the 64 uniform tiles and the 20 MDA tiles form an 84-tile representation of each storm (Figure 3.5), expanding the effective dataset size while emphasizing the important transition zones. This hybrid sampling strategy ensures that the model learns both the global spatial structure of storm tide and the localized, high-variability features that drive flooding impacts.

3.2.1 Mean Wetness Variability

Understanding where storm-tide behavior changes most across storms is essential for designing an effective tile-selection strategy. Some parts of the domain – such as the open ocean or high inland terrain – behave in a consistent manner across all storms, while other regions experience large shifts between wet and dry conditions depending on storm track, intensity, and tide phase. The transitional zones are where storm-tide gradients are steepest and where predictive skill is most critical. To identify these locations, we compute a *wetness variability index* for every pixel in the domain.

Each storm provides a maximum storm-tide raster in which dry pixels are encoded as 255 and wet pixels as values below 255. Treating each pixel as a Bernoulli variable (wet/dry), we count the number of storms in which pixel (i, j) is wet is accumulated across all 1,813 storms. Let n denote the total number of storms, and define the wetness indicator $w_k(i, j)$:

$$w_k(i,j) = \begin{cases} 1, & \text{if pixel } (i,j) \text{ is wet in storm } k, \\ 0, & \text{otherwise.} \end{cases}$$

The fraction of storms in which pixel (i, j) is wet is then

$$\lambda_{ij} = \frac{1}{n} \sum_{k=1}^{n} w_k(i,j).$$

The wetness-variability field is then computed using the Bernoulli variance

$$\mathrm{Var}(i,j) = \lambda_{ij}\,(1 - \lambda_{ij}).$$

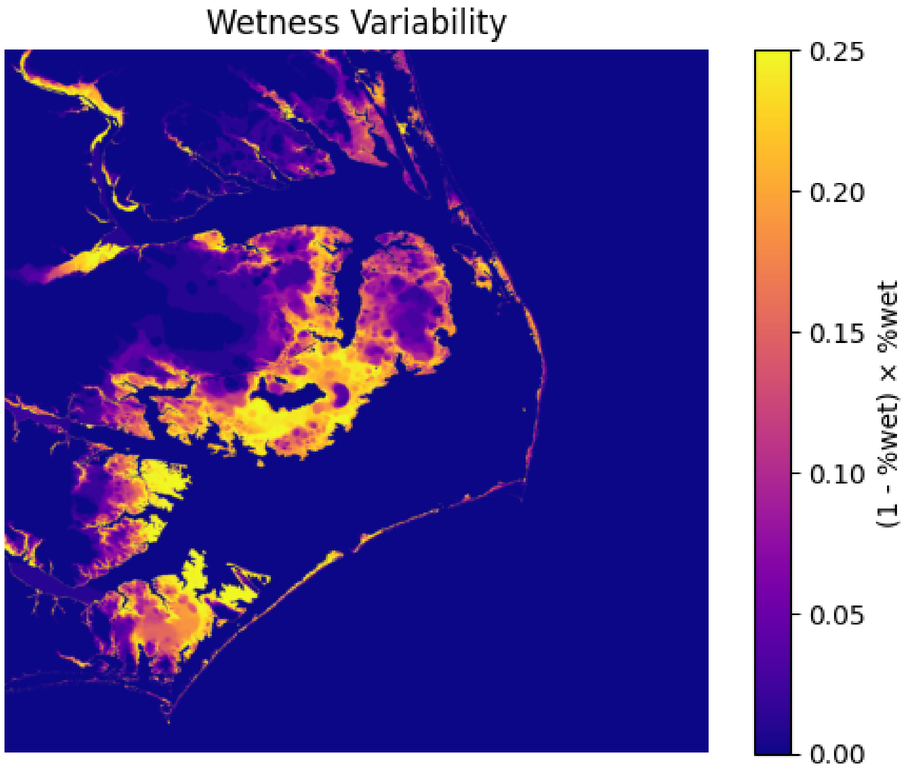


Figure 3.6: Mean wetness variability across the ADCIRC storm library. Values range from 0 to 0.25, the maximum possible Bernoulli variance. High-variability regions (yellow–orange) correspond to the dynamic wet/dry interface, where inundation depends strongly on storm characteristics. Low-variability regions (dark blue) represent areas that are consistently wet (open ocean) or consistently dry (inland terrain).

This metric has maxima for pixels that are wet in half the storms –the dynamic wet/dry front – and minima for pixels that are always wet (open ocean) or always dry (high inland elevations) (Figure 3.6). High-variability bands trace estuaries, barrier islands, and low-lying floodplains, revealing the parts of the coastline most sensitive to storm characteristics. Offshore waters and elevated inland terrain appear as low-variability zones, reflecting their consistent behavior across the storm catalog.

3.2.2 Maximum Dissimilarity Algorithm

The Maximum Dissimilarity Algorithm (MDA) provides a principled way to select tile locations that capture the most informative regions of the domain. While the 64 uniform tiles ensure broad spatial coverage, they do not guarantee that the model sees enough examples from the dynamic wet/dry interface – the region where storm-tide behavior varies most strongly across storms. This interface is precisely where predictive skill is most critical, and where small spatial shifts in inundation can produce large differences in impacts. MDA addresses this gap by sampling locations that are both diverse and physical.

MDA has been used in environmental modeling and geospatial sampling to maximize coverage of heterogeneous fields (Camus et al. 2011). Its core idea is simple: choose points that are as different from one another as possible. In this work, 'difference' is defined using both spatial distance and wetness variability, ensuring that selected tiles are spread across the domain and concentrated in regions where storm-tide behavior changes most across storms.

Candidate Points and Weighting

To ensure that a 512 × 512-pixel tile can be extracted around each centerpoint, candidate points are restricted to a central region of the domain that excludes a 256-pixel border. Each candidate pixel is assigned a weight proportional to its wetness variability:

$$w(i, j) = \frac{\mathrm{Var}(i, j)}{200}.$$

The normalization factor (200) keeps the weighted scores numerically stable while preserving the relative importance of high-variability regions. This weighting ensures that the algorithm preferentially selects centerpoints near the wet/dry interface, where $\mathrm{Var}(i, j)$ is largest, while still enforcing spatial diversity across the domain. In practice, this means that candidate points located in wet offshore waters or dry inland terrain receive very low weights, while pixels along estuaries, barrier islands, and floodplains receive the strongest influence in the selection process.

This normalization step is essential because it prevents the variability term from overwhelming the spatial component of the algorithm. Without normalization, the highest-variability pixels would dominate the selection process and produce clusters of tiles in a small number of coastal hotspots. With normalization, the algorithm balances physical importance (variability) with geometric spread (distance), producing a more representative sampling of the dynamic coastal zone.

Weighted Dissimilarity Selection

The MDA begins by selecting the pixel with the highest wetness variability. At each iteration, the algorithm computes a weighted dissimilarity score for every remaining candidate:

$$\mathrm{score}(i) = d_{\min}(i) \cdot w(i),$$

where $d_{\min}(i)$ is the minimum Euclidean distance from candidate i to any previously selected centerpoint, and $w(i)$ is the normalized variability weight. This ensures that selected points are both spatially diverse and physically meaningful. High-variability pixels that are far from previously selected points receive the highest scores, while low-variability or spatially redundant pixels are deprioritized.

This iterative selection process continues until 20 centerpoints have been chosen for each storm. The resulting set of centerpoints is reshaped into a $(1813, 20, 2)$ array for downstream tiling.

Minimum-Spacing Constraint

A later postprocessing step enforces a minimum spacing requirement (e.g. 250 pixels) between selected centerpoints. This prevents clustering in extremely high–variability regions and ensures that the 20 MDA tiles sample a broad swath of the wet/dry front. This spacing constraint also guarantees that each tile contains unique spatial information, reducing redundancy and improving the diversity of training examples.

Role of MDA Tiles in the Training Dataset

The 20 MDA tiles do *not* replace the 64 uniform tiles; instead, they *augment* them. The uniform tiles guarantee that the model sees the entire domain, while the MDA tiles increase the density of training examples in the dynamic coastal transition zone. This hybrid sampling strategy allows the model to learn both the large-scale spatial structure of storm tide and the fine-scale inundation patterns that occur along the shoreline. As a result, the training dataset becomes both richer and more informative, improving the model's ability to generalize across storms with different tracks, intensities, and surge characteristics.

3.2.3 Data Shuffling

After centerpoints are selected, each storm is tiled using the same set of 84 centerpoints (64 uniform + 20 MDA). These tiles are grouped by centerpoint and shuffled to prevent spatial or temporal ordering from biasing the neural-network training. The shuffled tiles are partitioned into training, validation, and testing datasets (Figure 3.7) and stored in HDF5 format, ensuring efficient loading during model training while preserving spatial consistency across storms.

Because each storm contributes an identical set of tiles, the shuffling strategy affects how the model encounters spatial patterns, storm-to-storm variability, and rare high-impact

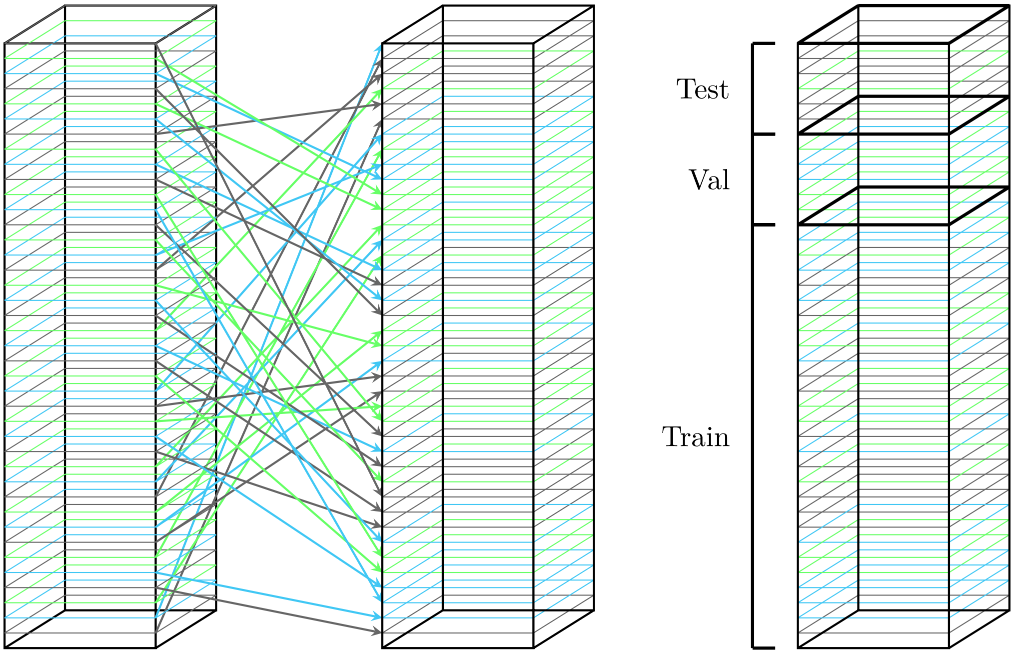


Figure 3.7: This figure illustrates the full preprocessing workflow applied to the storm-image dataset. The left block shows the original 3D tensor of storm slices, where each horizontal gridline corresponds to a single storm image. The middle block shows the randomized shuffling of slices, with each arrow mapping an original index to a unique new position; matching colored gridlines highlight the permutation. The right block shows the shuffled dataset partitioned into training, validation, and testing subsets using a 70/15/15 split.

events. There are two ways we could approach this: random shuffling in-between all storm-tide tiles, or shuffling storms within train/test/val. Random shuffling reduces storm-based biases, while shuffling per storm ensures that the model is seeing the entirety of the study area for the training storms.

In this work, we compare both approaches because they emphasize different learning behaviors:

- **Full random shuffling** treats every tile as an independent sample, maximizing statistical mixing and reducing storm-level autocorrelation. All 152,292 tiles are pooled and permuted randomly. The first 70% are assigned to training, the next 15% to validation, and the final 15% to testing. Tiles from the same storm may appear in different splits (e.g. storm 0 may have some tiles in training and other tiles in testing or val).
- **Storm-wise shuffling** preserves storm integrity within each dataset split, ensuring that the model sees complete spatial coverage for the storms assigned to training. Storms are permuted randomly, and entire storms (all 84 tiles) are assigned to train/val/test. This ensures that no storm appears in more than one dataset split.

This distinction is critical because it determines whether the model is allowed to see partial information from a storm during training and the remainder during validation/testing.

Tiles are stored in HDF5 format with four datasets per tile: the input feature tensor (x), the dry mask (dry), the grayscale elevation tile (img), and the centerpoint metadata (cp). Each tile has dimensions (25 × 35) for the feature tensor and (512-pixel × 512-pixel) for the image-based fields.

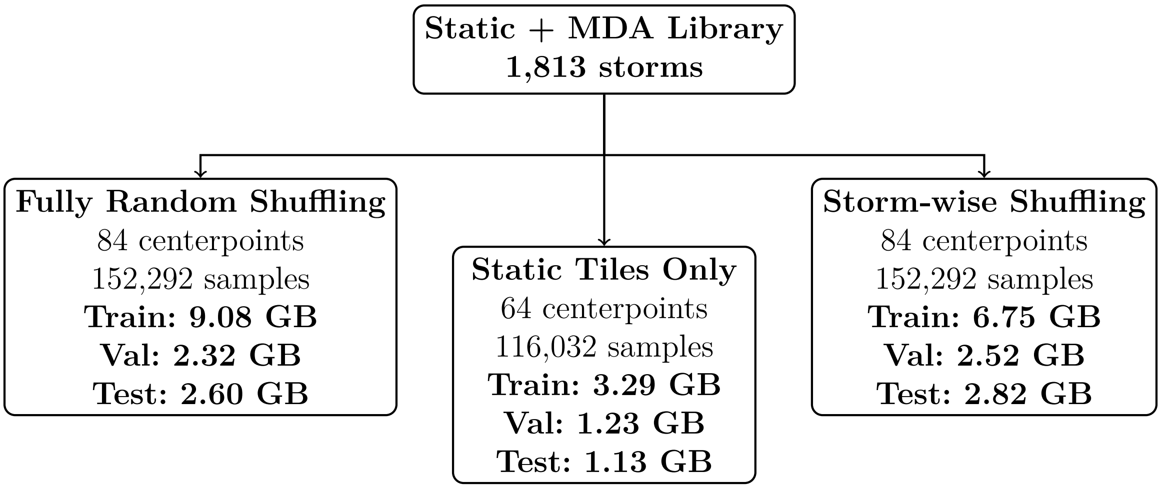


Figure 3.8: Schematic of the storm-tide library and the two shuffling strategies used to construct training, validation, and testing datasets.

The Static + MDA storm-tide library contains 1,813 synthetic storms, each represented by 84 tiles of peak storm tides, for a total of 152,292 samples (Figure 3.8). Data sizes for the different shuffling:

1. Full random shuffling, Static + MDA, 84 centerpoints

    Train (102,035 data points) 8.45 GB

    Val (23,605 data points) 2.17 GB

    Test (26,652 data points) 2.42 GB

    Total library size: 13.04 GB

2. Storm-wise shuffling, Static + MDA, 84 centerpoints

    Train (101,976 data points) 6.29 GB

    Val (23,604 data points) 2.35 GB

    Test (26,712 data points) 2.62 GB

Total library size: 11.26 GB

The Static storm-tide library contains 1,813 synthetic storms, each represented by 64 tiles of peak storm tides, for a total of 116,032 samples (Figure 3.8). Data sizes for the different shuffling:

3. Full random shuffling, Static tiles, 64 centerpoints only)

    Train (77,741 data points) 3.06 GB

    Val (17,985 data points) 1.14 GB

    Test (20,306 data points) 1.05 GB

    Total library size: 5.25 GB

3.3 Development of Temporal Inputs

The storm-track component provides the temporal forcing needed to predict the spatial distribution of storm tide at each tile centerpoint. This temporal encoding allows the neural network to learn how the storm's approach, intensification, weakening, and directional changes shape the resulting spatial distribution of storm tide. By providing the 25 hr surrounding the storm's closest approach, the model is exposed to the sequence of physical events that produce the peak storm-tide response: the buildup of onshore winds, the tightening or loosening of the pressure gradient, the acceleration or deceleration of storm motion, and the rotation of the wind field as the storm passes.

In effect, the temporal input teaches the model how a specific storm 'unfolds' at the tile centerpoint and how that unfolding translates into the final flood extent. The network is not memorizing static storm attributes; it is learning the dynamical mapping from storm evolution to peak water levels. The 25-hr window captures the forcing history that drives surge generation and propagation, enabling the model to associate particular storm behaviors—such as rapid intensification, slow translation, or a shift to a more onshore track—with the spatial footprint of flooding that ultimately develops.

Each synthetic storm includes an hourly sequence of storm-center positions and associated meteorological parameters. For every tile (selected in Section 3.2), we extract a localized, geodesically referenced subset of the storm and convert it into a structured set of input features. This section describes (1) the selection of the 25 closest hours, (2) the construction of the input feature vector, and (3) the geographic referencing procedure used to convert pixel coordinates into latitude, longitude, and relative geodesic quantities.

3.3.1 Closest 25 Hours

Because the hydrodynamic response at a given tile centerpoint is driven by the storm's behavior in the hours surrounding its closest approach, we extract a fixed temporal window of the **25 hr when the storm is nearest to each centerpoint**. For each storm and each tile center, the geodesic distance between the storm center and the tile center is computed at every time step. The time of minimum distance is identified, and a symmetric window of 25 hr (12 hr before and 12 hr after) is selected.

This approach ensures that the model receives a consistent temporal sequence aligned to the physically relevant portion of the storm's evolution. It also avoids including long periods when the storm is far from the domain and has negligible influence on local water levels. In cases where the full 25-hr window cannot be extracted (e.g. storms with short lifespans), the sequence is zero-padded to maintain uniform input dimensions.

3.3.2 Input parameter Selection

Each one-hourly storm-track record contains a large number of meteorological and geometric variables. For each of the 25 hr, we extract a fixed set of **35 parameters** that describe the storm's intrinsic structure, its evolving position relative to each tile center, and the tide levels at key coastal locations. These parameters fall into three groups:

- **Storm-intrinsic and geometric parameters (12 features)**. These features describe the storm's intensity, structure, and motion, as well as its evolving geodesic relationship to the tile center:
    - maximum sustained wind speed (m/s)
    - central pressure (hPa)
    - radius to maximum winds (km)
    - heading direction (degrees)
    - forward speed (m/s)
    - forward-speed components (u, v)
    - geodesic distance from the storm center to the tile center (km)
    - projected east-west and north-south distances to the tile center
    - forward azimuth from the storm center to the tile center
    - storm category (derived from wind speed)

- **Tide forcing at coastal and estuarine stations (21 features)**. The storm input includes a set of 21 tide-forcing parameters that describe the background astronomical tide and its spatial gradients across the North Carolina coastline. These features originate from the ADCIRC tidal boundary conditions and from the tidal-gradient components developed in prior work (Cuevas López et al. 2025). The 21 features consist of:

  - **Seven tidal water-level time series** extracted at key coastal and estuarine stations: Duck, Oregon Inlet, Cape Hatteras, Beaufort, Albemarle Sound, Pamlico Sound, and Neuse River.
  - **Directional tidal-forcing terms** inherited from previous feature design. These terms represent the east–west (u_{dist}) and north–south (v_{dist}) components of the storm-to-station distance. Let d_S denote the geodesic distance from the storm center to tidal station S, and let θ_S denote the forward azimuth from the storm center to S. The components are defined as

    $$u_{\text{dist},S} = d_S \sin(\theta_S), \qquad v_{\text{dist},S} = d_S \cos(\theta_S),$$

    where $u_{\text{dist},S}$ represents the east–west component of the storm-to-station distance and $v_{\text{dist},S}$ represents the north–south component. These terms preserve both the magnitude and direction of the storm's position relative to each tidal station and maintain continuity with the feature engineering approach used in prior ADCIRC–ML development.

  Together, these 21 tide-forcing features describe both the local tidal state at multiple coastal stations and the broader spatial structure of the astronomical tide field across the region.

- **Geographic coordinates (2 features)** The storm-center latitude and longitude provide the absolute geographic position of the storm at each time step. These coordinates anchor all other storm-intrinsic and geometric features, allowing the model to relate storm evolution to its location relative to the coastline and the tile centerpoints.

  - storm-center latitude
  - storm-center longitude

Together, these 35 parameters provide a compact but rich representation of storm intensity, structure, motion, and tidal forcing. This reduced feature set balances physical completeness with computational efficiency, ensuring that the model receives the most relevant storm-track information for predicting storm tide at each tile centerpoint.

3.3.3 Geographic Referencing

The synthetic storm tracks provided by ADCIRC are stored in geographic coordinates (latitude and longitude), with one storm-center position recorded at each hourly time step. In contrast, the tile centerpoints (selected in Section 3.2) are defined in pixel coordinates on the 4096 × 4096 computational grid. To compute geodesic distances and azimuths between the storm center and each tile center, these pixel-based centerpoints are converted to geographic latitude and longitude using a linear mapping between grid indices and the domain's bounding box:

$$\begin{aligned}\mathrm{lat}(i) &= \mathrm{lat}_{\mathrm{max}} - \frac{i}{4096}\,(\mathrm{lat}_{\mathrm{max}} - \mathrm{lat}_{\mathrm{min}}),\\ \mathrm{lon}(j) &= \mathrm{lon}_{\mathrm{min}} + \frac{j}{4096}\,(\mathrm{lon}_{\mathrm{max}} - \mathrm{lon}_{\mathrm{min}}),\end{aligned}$$

where (i, j) are the row and column indices of the storm center. For each tile centerpoint (ϕ_c, λ_c), the following geodesic quantities are computed at every time step:

1. **Total geodesic distance** between the storm center and the tile center.
2. **Forward azimuth** (bearing) from the storm center to the tile center.
3. **Projected distances** in the east–west (u) and north–south (v) directions:

   $$u = d\sin(\theta), \qquad v = d\cos(\theta),$$

   where d is the geodesic distance and θ is the azimuth.

These quantities are used to align the 25-hr storm window to the time of closest approach and to construct the storm-intrinsic portion of the 35-parameter feature vector. The final dataset consists of three aligned arrays: geographic coordinates, tide values, and the full parameter matrix for each 25-hr window.

3.3.4 Temporal Input Summary

The temporal input to the model consists of a fixed 25-hr window of storm information, aligned to the time of closest approach for each tile centerpoint. For every storm and every tile, this window is converted into a structured matrix of 35 parameters per hour, yielding

Table 3.1: Storm Parameter Index Table.

Index	**Feature**	**Index (Cont.)**	**Feature**
0	Storm Latitude	20	Neuse Tide
1	Storm Longitude	21–22	Dist. u,v to Neuse
2	Duck Tide	23	Wind Speed
3–4	Dist. u,v to Duck	24	Pressure
5	Oregon Inlet Tide	25	Radius to Max Wind Speed
6–7	Dist. u,v to Oregon Inlet	26	Heading Direction
8	Cape Hatteras Tide	27	Forward Speed
9–10	Dist. u,v to Cape Hatteras	28	Forward Speed (u)
11	Beaufort Tide	29	Forward Speed (v)
12–13	Dist. u,v to Beaufort	30	Distance from storm center to tile
14	Albemarle Tide	31	Distance (u) from storm center to tile
15–16	Dist. u,v to Albemarle	32	Distance (v) from storm center to tile
17	Pamlico Tide	33	Forward Azimuth
18–19	Dist. u,v to Pamlico	34	Category

a final temporal-input tensor of size 25 × 35 per storm-tile. The full list of these 35 parameters, along with their index positions in the feature vector, is provided in Table 3.1. These parameters encode storm intensity, structure, translation, geodesic approach geometry, and tidal forcing at seven coastal and estuarine stations, together providing a compact but information-rich representation of the storm's evolving influence on local water levels.

During training, each temporal-input matrix is loaded as a (25,35) array, normalized, and batched using a streaming HDF5 generator. The resulting input tensor has shape $(\text{batch}, 25, 35)$, ensuring that the model receives a consistent sequence describing how the storm evolves as it approaches, impacts, and recedes from the tile centerpoint. This structure allows the neural network to learn the dynamic mapping from storm evolution to the spatial distribution of peak storm tides.

3.4 Neural Network Architecture

The prediction model is a dual-stream neural network that integrates (1) a temporal representation of storm evolution and tidal forcing and (2) a spatial representation of the land–water configuration surrounding each tile. The model takes two inputs: a 25-hr sequence of 35 storm

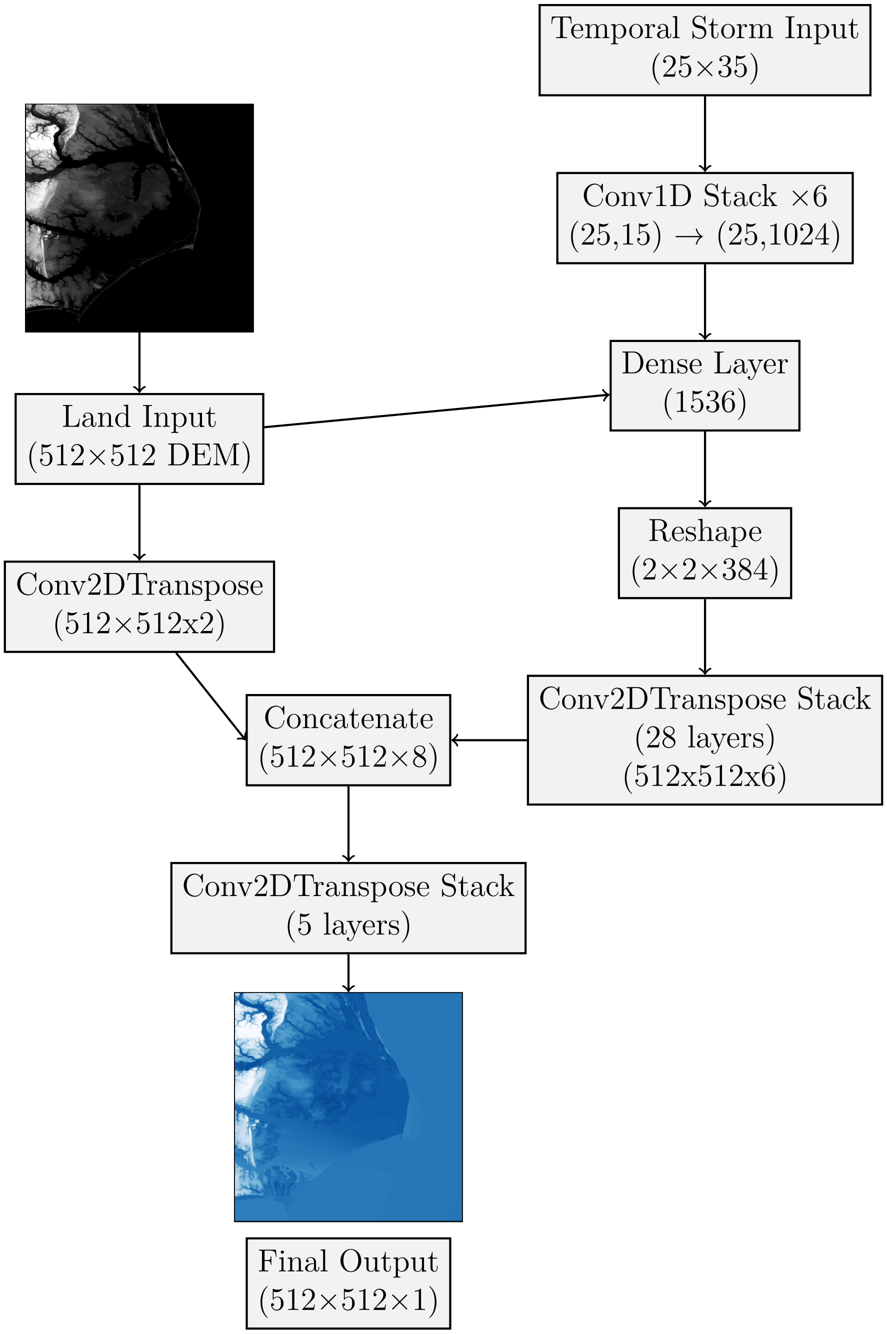


Figure 3.9: Vertical flow chart of the full model architecture with storm track inputs, image inputs, and continuous elevation outputs.

parameters and a 512-pixel × 512-pixel land mask. These inputs are processed through separate encoder streams, fused into a joint latent representation, and then decoded through a multi-stage upsampling network to produce a 512×512-pixel map of peak storm tide (Figure 3.9).

3.4.1 Temporal Stream: 1D Convolutional Encoder

The temporal input is a matrix of shape $(batch, 25, 35)$ containing the 35 storm-track features (Section 3.3.2). Each row corresponds to one hour in the 25-hr window, and each column corresponds to one of the indexed storm parameters (Table 3.1). This sequence is processed using a stack of 1D convolutional layers with increasing channel depth (Figure 3.10).

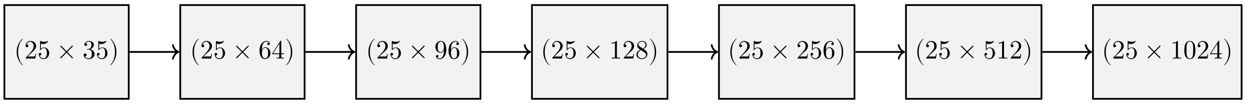


Figure 3.10: Temporal stream: 1D convolutional encoder applied to the (25 × 35) storm-track matrix.

Each layer uses a kernel size of 3 with ReLU activation and "same" padding. A kernel size of 3 was chosen because it is the standard default in most convolutional architectures (He et al. 2015; Simonyan and Zisserman 2015; Bai et al. 2018) and provides an effective balance between model capacity and computational efficiency. It also captures short-range temporal interactions – such as hour-to-hour changes in storm intensity, motion, or tide forcing – without introducing unnecessary parameters. This progression allows the network to extract abstract temporal features describing storm intensity, motion, tide forcing, and their interactions.

The expansion to 1024 channels is motivated by the downstream fusion architecture. The temporal latent vector must be reshaped into a 2 × 2 × 384 tensor for integration with the spatial encoder. A 1024-dimensional vector provides sufficient representational capacity to encode the storm's temporal evolution while matching the dimensionality required by the decoder.

A global average pooling layer compresses the temporal dimension, producing a fixed-length latent vector of dimension 1024. Global average pooling was selected instead of flattening because it is more computationally efficient and avoids over-parameterization. Flattening a (25,1024) tensor would produce 25,600 parameters per sample, which increases memory usage and encourages overfitting. Global average pooling retains the most salient temporal information while reducing the parameter count by more than an order of magnitude.

3.4.2 Spatial Stream: Land-Mask Encoder

The spatial input is a 512 × 512-pixel land elevation map representing the local topography and land–water structure. This field is processed through a shallow convolutional encoder followed by a dense projection: a 3 × 3 transposed convolution producing a two-channel spatial feature map, and a reshape and dense transformation that compresses the 512 × 512 field into a 512-dimensional latent vector. A global average pooling layer produces a compact spatial embedding that summarizes the local land–water configuration relevant to storm-tide response.

3.4.3 Feature Fusion and Latent Reshaping

The temporal and spatial latent vectors (1024 and 512 dimensions, respectively) are concatenated to form a joint 1536-dimensional representation. This fused vector is reshaped into a (2 × 2 × 384) tensor, which serves as the seed for the decoder network.

3.4.4 Decoder: Multi-Stage Upsampling Network

The decoder consists of a sequence of transposed convolution (Conv2DTranspose) layers that progressively upsample the fused latent representation from 2 × 2 to 512 × 512. The channel depth decreases at each stage (Figure 3.11):

$$2\times2 \rightarrow 4\times4 \rightarrow 8\times8 \rightarrow 16\times16 \rightarrow 32\times32 \rightarrow 64\times64 \rightarrow 128\times128 \rightarrow 256\times256 \rightarrow 512\times512.$$

At each resolution, one or more transposed convolution layers with ReLU activation refine the spatial features. In the final stages, the upsampled decoder output is merged with the shallow spatial features extracted from the dry-land elevation input. This provides the model with basic spatial context—where it is predicting and the initial elevations prior to landfall—while helping preserve fine-scale land–water boundaries.

A final 3 × 3 transposed convolution with linear activation produces the output field: representing the predicted peak storm tide at each grid cell.

3.4.5 Full Model Architecture Details

The complete neural network contains a total of 16,391,579 parameters (62.53 MB), of which 5,463,859 are trainable and 10,927,720 belong to the optimizer state. The model is implemented in Keras using the Functional API and consists of three major components: the

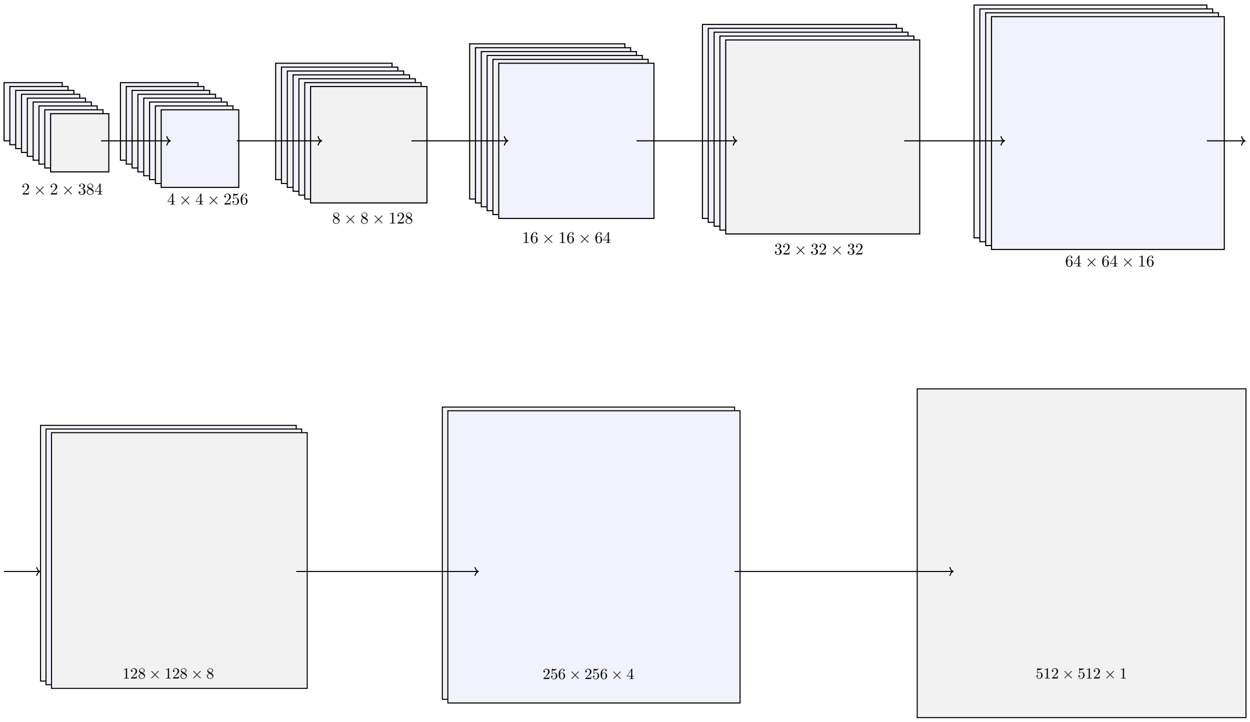


Figure 3.11: Simplified image up-sampling architecture (each stack of squares represents multiple Conv2DTranspose layers with different channel depths and resizing techniques). Top: first six stacks of Conv2DTranspose layers. Bottom: final three stacks of layers.

temporal encoder, the spatial encoder, and the multi-stage decoder. A full layer-by-layer summary is provided in Table B.1.

Temporal Stream Architecture

The temporal encoder processes the (25 × 35) storm-parameter sequence through six 1D convolutional layers with progressively increasing channel depth:

$$35 \to 64 \to 96 \to 128 \to 256 \to 512 \to 1024.$$

Each Conv1D layer uses a kernel size of 3, ReLU activation, and “same” padding. Parameter counts increase with depth, ranging from 6,784 in the first layer to 1,573,888 in the final temporal convolutional layer. A global average pooling layer compresses the (25×1024) tensor into a 1024-dimensional latent vector, reducing over-parameterization while preserving salient temporal features.

Spatial Stream Architecture

The spatial encoder processes the 512×512 land mask through a shallow convolutional transformation followed by a dense projection. A 3×3 transposed convolution produces a two-channel spatial feature map, which is then reshaped and passed through a dense layer containing 262,656 parameters. A global average pooling layer produces the final 512-dimensional spatial latent vector summarizing the local land–water configuration.

Fusion Layer and Latent Reshaping

The temporal (1024-D) and spatial (512-D) latent vectors are concatenated to form a 1536-dimensional joint representation. This fused vector is reshaped into a $(2 \times 2 \times 384)$ tensor, which serves as the seed for the decoder.

Decoder Architecture

The decoder is a multi-stage Conv2DTranspose upsampling network that expands the latent tensor from 2×2 to 512×512. The spatial resolutions progress as:

$$2\times2 \to 4\times4 \to 8\times8 \to 16\times16 \to 32\times32 \to 64\times64 \to 128\times128 \to 256\times256 \to 512\times512.$$

At each resolution, one or more transposed convolution layers refine the spatial features. Early layers contain high channel depth (e.g., 256, 192, 128), while later layers gradually reduce to 32, 24, 16, and 8 channels. For example, the first Conv2DTranspose layer contains 884,992 parameters, while later refinement layers contain between 511 and 13,856 parameters.

In the final stages, the decoder output is merged with shallow spatial features extracted directly from the land mask, helping preserve fine-scale land–water boundaries. A final 3×3 transposed convolution with linear activation produces the output field $\hat{y} \in \mathbb{R}^{(512\times512\times1)}$ representing the predicted peak storm tide.

3.4.6 Training Procedure

The model is trained using the Adam optimizer with a learning rate of 10^{-5}. The loss function is the Huber loss ($\delta = 1.0$), which is robust to outliers in the storm-tide field. Performance is monitored using mean absolute error (MAE) and root mean squared error (RMSE). Early stopping and model checkpointing are used to prevent overfitting and retain the best model weights. Training and validation data are streamed from HDF5 files using a TensorFlow data generator with automatic prefetching for efficient GPU utilization.

3.4.7 Weighted Loss

Loss weighting is a common strategy in machine learning for emphasizing errors in regions of the prediction domain that are physically important or systematically underrepresented in the training data. In geophysical applications, weighted losses have been used to highlight sharp gradients, rare events, or threshold-sensitive processes by assigning larger penalties to errors within user-defined subsets of the input space (Cui et al. 2019; Rasp and Lerch 2018).

Motivated by these approaches, and by preliminary diagnostics showing elevated residuals in low-lying coastal areas, we implemented an elevation-weighted variant of the Huber loss. The weighting scheme increases the contribution of errors within a specified elevation band while reducing the influence of errors at higher elevations. This formulation preserves the robustness of the Huber loss while allowing the model to focus more strongly on regions where small vertical differences have disproportionate impacts on inundation onset. The weighted loss implementation is evaluated in Section 4.2.2. These experiments were conducted by continuing training from the baseline model checkpoint and applying the elevation-dependent weighting during fine-tuning. The goal was to assess whether targeted weighting could reduce residuals in the coastal fringe without degrading performance elsewhere in the domain.

Chapter 4

Results and Discussion

4.1 CNN for Prediction of Storm Tide Maps

4.1.1 Training

The CNN was trained using a staged approach designed to balance early-epoch stability with late-epoch sensitivity to small-scale elevation differences. Due to the large dataset sizes and the model complexity, it was not possible to train to a sufficient accuracy in a single stage; instead, the model was trained in multiple stages by using HPC resources at North Carolina State University (Hazel, `https://hpc.ncsu.edu/main.php`) and Texas A&M University (ACES, `https://hprc.tamu.edu/aces/`). In all stages, the model was trained using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 1×10^{-5} and a varying Huber loss δ (Huber 1992). This configuration provided a stable starting point for learning broad storm-driven elevation patterns while preventing large gradients during the early stages of training. Across all stages, the training process exhibited a characteristic progression in which the model first captured large-scale structural relationships before refining its ability to resolve localized coastal gradients.

On the HPC resources, training was executed by using a mixture of NVIDIA L40 and H100 GPUs, depending on node availability at each stage. Early stages (1–2, with $\delta = 1.0$) ran on 48 to 60 L40 GPUs and required between 2400 and 4000 wall-clock minutes per job, reflecting a heavier computational load. Middle stages (3–7, with $\delta = 0.8$) required 26 to 64 L40 and H100 GPUs and between 508 to 4000 minutes as the model transitioned into finer-scale gradient regimes. Later stages (8–9, with $\delta = 0.5$) were trained on H100 nodes (24–32 GPUs) and each required 4000 minutes, consistent with the increased sensitivity and computational cost of the smallest Huber threshold.

4.1.2 Training Behavior and Huber Loss Regimes

Across all stages (478 total epochs, Figure 4.1), the model was trained using the Adam optimizer with a learning rate of 1×10^{-5} and an initial Huber loss with $\delta = 1.0$. This

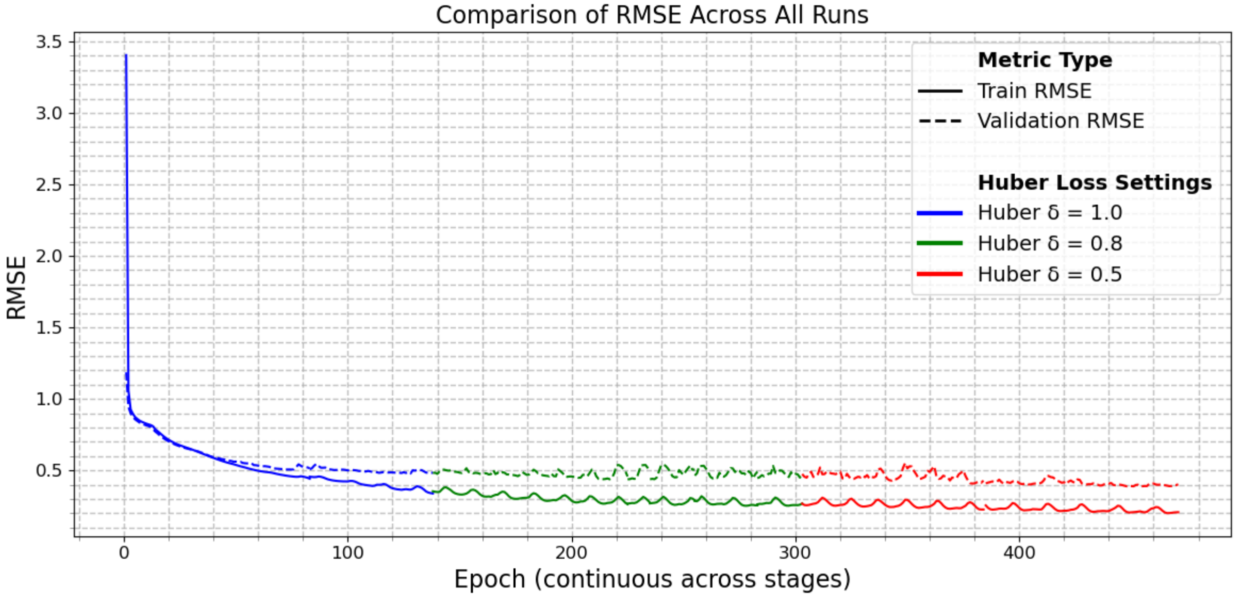


Figure 4.1: Comparison of RMSE across all nine training stages, shown as a continuous sequence of epochs. Solid lines represent training RMSE and dashed lines represent validation RMSE. Each stage begins where the previous stage ended, illustrating the staged training process and the progressive reduction in error as the model converges. Validation RMSE tracks training RMSE across stages, indicating stable generalization performance and consistent improvement throughout the multi-stage training pipeline.

configuration provided a stable foundation for learning broad storm–elevation relationships while preventing large gradient excursions during early training. The Huber threshold was reduced to increase the model's sensitivity to small residuals and to refine fine–scale elevation structure. Each Huber regime produced a distinct convergence pattern, reflecting the changing balance between stability and sensitivity in the loss landscape.

Stage 1: $\delta = 1.0$ — Smooth, Stable Convergence. During the initial stages (epochs 1 to 139, blue line in Figure 4.1), the wide quadratic region of the Huber loss produced smooth and monotonic decreases in both training and validation RMSE. The model first learned large–scale structural relationships between storm forcing and elevation response, with minimal oscillation in the loss curves. Training RMSE decreased from 3.404 m to 0.339 m, while validation RMSE declined from 1.186 m to 0.478 m. Training MAE fell from 2.771 m to 0.187 m, and validation MAE stabilized between 0.859 m to 0.272 m by the end of the stage. These trends indicate that the model learned large-scale storm–elevation structure during this phase, with minimal oscillation in the loss curves and no signs of instability or overfitting.

Stage 2: $\delta = 0.8$ — Increased Sensitivity and Moderate Variability. During the intermediate stages (epochs 140 to 309, green line in Figure 4.1), when the Huber threshold was reduced to $\delta = 0.8$, the training dynamics became more variable. This adjustment improved late-epoch stability, reduced oscillations in the validation metrics, and allowed the model to refine fine-scale spatial features without overfitting to outlier values (Huber 1992; Zhao et al. 2017). The narrower quadratic region increased the model's responsiveness to moderate–magnitude residuals, producing sharper gradient updates and more pronounced oscillations in both training and validation curves. Across the five $\delta = 0.8$ runs, training RMSE decreased from 0.381 m to 0.253 m, while validation RMSE fluctuated between 0.437 m to 0.541 m, reflecting sharper gradient updates and increased sensitivity to moderate-magnitude residuals. Training MAE improved from 0.218 m to 0.128 m, and validation MAE remained in the 0.242–0.335 m range. Although the overall trend continued downward, the RMSE curves were noticeably 'bumpier' than in Stage 1, reflecting the model's transition from learning large–scale structure to resolving finer spatial gradients. This behavior is consistent with the expected shift toward more MAE–like optimization, where the loss surface becomes less smooth and gradient updates respond more to local error patterns.

Stage 3: $\delta = 0.5$ — Highest Sensitivity, Most Variability, Continued Improvement. The final training stages consisted of two stages using the smallest Huber threshold, $\delta = 0.5$, spanning epochs 310–478 (red lines in Figure 4.1). Decreasing δ increased the model's sensitivity to small-magnitude residuals and emphasized fine-scale error correction. These stages are not smoother than the $\delta = 0.8$ regime; instead, they show similar variability but a clearer downward trend, indicating that the model had entered a refinement phase rather than undergoing large structural adjustments. In the first $\delta = 0.5$ run (epochs 310-391), training RMSE decreased from 0.309 m to 0.226 m, while validation RMSE stabilized in the 0.408–0.552 m range. Despite this variability, the overall trend showed continued improvement relative to the preceding $\delta = 0.8$ stage.

The second $\delta = 0.5$ run (epochs 392–478) further refined the model's performance and achieved the lowest errors of the entire training process. The training and validation curves in this stage exhibit less pronounced oscillations, reflecting a smoother convergence profile as the model focused on correcting small-scale residuals. Training RMSE decreased from 0.272 m to 0.201 m, and validation RMSE improved from 0.461 m to 0.389 m. Training MAE reached 0.091–0.140 m, while validation MAE stabilized between 0.201–0.270 m. These results indicate that the reduced Huber threshold successfully enhanced the model's ability to resolve subtle elevation differences while maintaining stable generalization performance. The final checkpoint was selected during this stage, where the model achieved its best combination of

Table 4.1: Summary of the nine training stages, including epoch ranges, Huber loss thresholds, RMSE and MAE behavior, and qualitative notes on convergence characteristics.

Stage	**Epochs**	δ	**RMSE Train**	**RMSE Val**	**MAE Train**	**MAE Val**
1	1–83	1.0	0.442–3.404	0.506–1.186	0.255–2.771	0.295–0.859
2	84–139	1.0	0.339–0.457	0.478–0.545	0.187–0.267	0.272–0.321
3	140–227	0.8	0.264–0.381	0.438–0.538	0.137–0.218	0.242–0.335
4	228–238	0.8	0.268–0.315	0.437–0.538	0.139–0.174	0.243–0.337
5	239–259	0.8	0.259–0.316	0.438–0.541	0.133–0.174	0.242–0.338
6	260–288	0.8	0.252–0.319	0.444–0.532	0.128–0.176	0.254–0.333
7	289–309	0.8	0.253–0.310	0.445–0.532	0.128–0.169	0.256–0.331
8	310–391	0.5	0.226–0.309	0.408–0.552	0.107–0.167	0.216–0.320
9	392–478	0.5	0.201–0.272	0.389–0.461	0.091–0.140	0.201–0.270

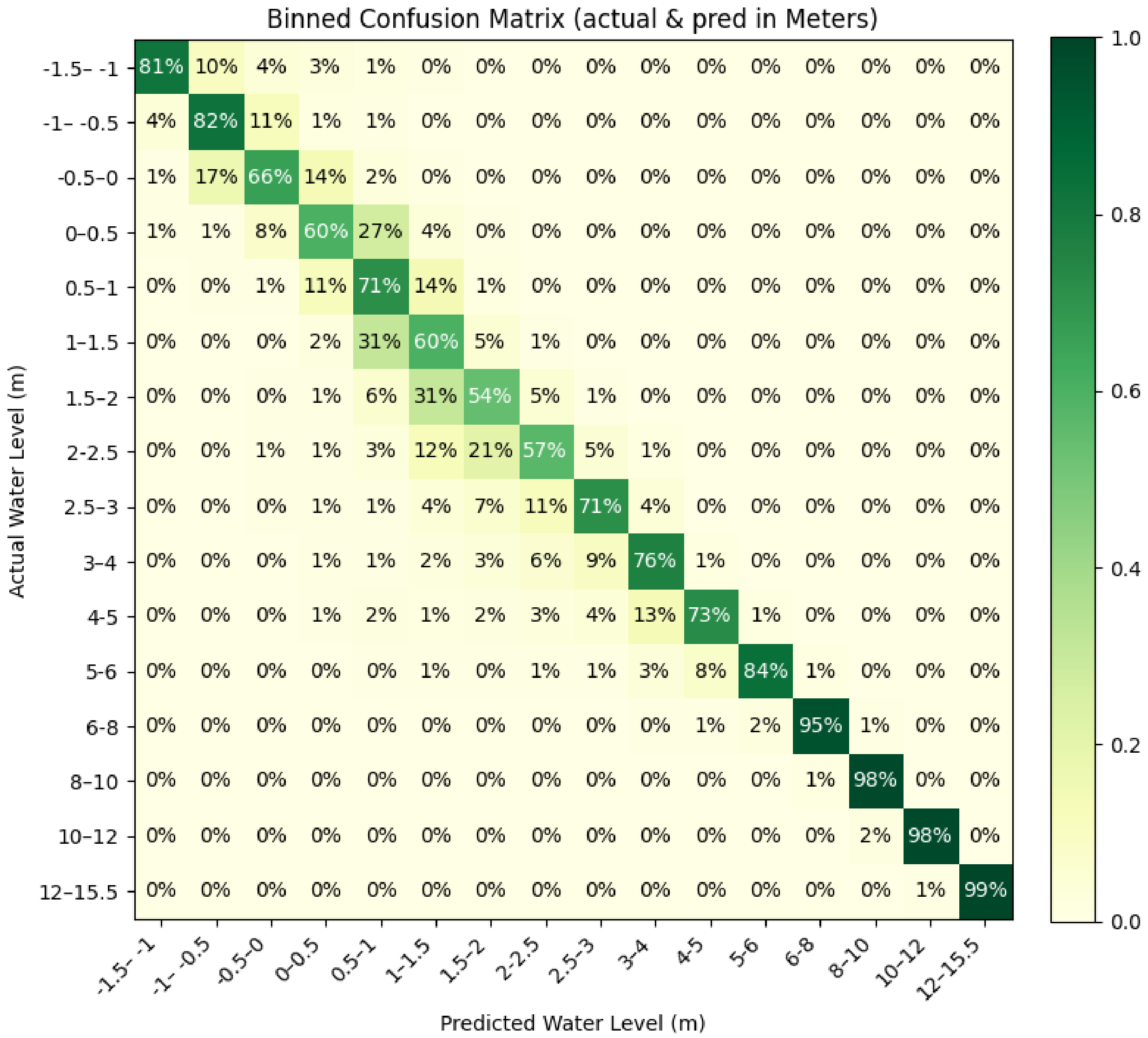
low training error and consistent validation behavior.

4.1.3 Overall Model Performance

The model's performance was evaluated across the full 4096-pixel × 4096-pixel test domain using elevation-specific RMSE, bin-wise accuracy, and a normalized confusion matrix (Figure 4.2). Throughout this analysis, 'elevation' refers to either the storm-tide surface elevation (in wet regions) or the ground-surface elevation (in dry regions), whichever is higher. Evaluating performance as a function of elevation allows us to distinguish how the model behaves in areas that are frequently inundated, intermittently wet, or consistently dry. These diagnostics highlight where the model performs well and where prediction uncertainty increases, particularly in regions with complex topographic or hydrodynamic behavior.

The model performs well in the low-elevation coastal fringe, where accurate predictions are most critical for determining inundation extent. RMSE in the 0 to 0.5 m elevation band is 0.3661 m, indicating that the model captures the fine-scale topographic gradients that govern early inundation, marsh flooding, and shoreline overtopping. Performance remains strong in the slightly higher 1 to 3 m band, with an RMSE of 0.3970 m, a range that corresponds to back-barrier flats, low dune systems, and other geomorphic settings where small vertical errors can meaningfully influence flood propagation.

When restricting evaluation to elevations at or above mean sea level, the model achieves an RMSE of 0.4096 m. This reflects stable performance across the broader coastal plain, where elevation variability increases and storm impacts become more heterogeneous. The consistency of RMSE across these elevation bands demonstrates that the model maintains reliable predictive skill from the lowest coastal surfaces through the upland transition zone.



	-1.5– -1	-1– -0.5	-0.5–0	0–0.5	0.5–1	1–1.5	1.5–2	2-2.5	2.5–3	3-4	4-5	5-6	6-8	8–10	10–12	12–15.5
-1.5– -1	81%	10%	4%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-1– -0.5	4%	82%	11%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-0.5–0	1%	17%	66%	14%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0–0.5	1%	1%	8%	60%	27%	4%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0.5–1	0%	0%	1%	11%	71%	14%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
1–1.5	0%	0%	0%	2%	31%	60%	5%	1%	0%	0%	0%	0%	0%	0%	0%	0%
1.5–2	0%	0%	0%	1%	6%	31%	54%	5%	1%	0%	0%	0%	0%	0%	0%	0%
2-2.5	0%	0%	1%	1%	3%	12%	21%	57%	5%	1%	0%	0%	0%	0%	0%	0%
2.5–3	0%	0%	0%	1%	1%	4%	7%	11%	71%	4%	0%	0%	0%	0%	0%	0%
3-4	0%	0%	0%	1%	1%	2%	3%	6%	9%	76%	1%	0%	0%	0%	0%	0%
4-5	0%	0%	0%	1%	2%	1%	2%	3%	4%	13%	73%	1%	0%	0%	0%	0%
5-6	0%	0%	0%	0%	0%	1%	0%	1%	1%	3%	8%	84%	1%	0%	0%	0%
6-8	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	95%	1%	0%	0%
8–10	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	98%	0%	0%
10–12	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	98%	0%
12–15.5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	99%

Figure 4.2: Normalized confusion matrix comparing predicted and actual elevations across the full test domain. Each cell shows the percentage of pixels whose predicted elevation falls into the same elevation bin as the true elevation. The strong diagonal structure indicates close agreement across most elevation ranges, with misclassifications concentrated in low-lying transitional zones and negative-elevation regions where hydrodynamic behavior is nonlinear. Higher-elevation bins exhibit minimal confusion, reflecting the model's ability to identify non-inundated terrain.

The overall test RMSE across the full domain is 0.2722 m, demonstrating that the model maintains consistent accuracy even when evaluated across the entire elevation range.

The elevation histogram comparing actual and predicted values (Figure 4.3 top) provides additional insight into how the model distributes elevation predictions across the domain. The two distributions align overall, indicating that the model preserves the large-scale structure of the terrain without introducing systematic elevation biases. However, the histogram reveals a pronounced under-representation of predicted elevations near 1 m, which correspond to the most densely populated elevation band in the domain. This discrepancy reflects the strong spatial gradients and hydrodynamic sensitivity of the shallow-water transition zone, where small vertical differences determine whether a pixel is classified as wet or dry. Elevations below 0 m are also under-represented in the predictions, consistent with the limited training data and the nonlinear hydrodynamics of tidal channels and depressions. These patterns are consistent with the RMSE-by-elevation-bin results (Figure 4.3 bottom), which show elevated errors in negative-elevation regions and a distinct RMSE peak in the 4 m to 5.5 m mid-elevation band. In contrast, RMSE remains low throughout the 0 m to 3 m coastal fringe, demonstrating strong model performance in the low-lying regions that exert the greatest control on inundation dynamics. Together, these analyses show that the model captures the overall elevation distribution while exhibiting predictable deviations in regions with sparse representation, steep gradients, or nonlinear storm-driven behavior.

Accuracy for each elevation bin is computed as the proportion of pixels whose predicted elevation falls into the same bin as the true elevation, i.e.

$$\text{Accuracy}_b = \frac{\text{Number of pixels with predicted and true elevations both in bin } b}{\text{Total number of pixels whose true elevation lies in bin } b}.$$

Bin-wise accuracy reveals clear elevation-dependent patterns in model performance. Accuracy is highest in the -1.5 m to -0.5 m bins (81.35% and 82.47%) and decreases to 65.90% in the -0.5 m to 0 m bin. Although these values seem large, these negative-elevation regions correspond to tidal creeks, channels, and bathymetric depressions that occupy a small fraction of the domain. Their hydrodynamic behavior is nonlinear, and the updated histogram shows that these elevations are under-represented in the predictions. As a result, even small deviations in predicted elevation can shift pixels into adjacent bins, making these accuracies to be meaningful despite their magnitude.

Accuracy in the 0 m to 1 m range is moderate, with values of 60.09% in the 0 m to 0.5 m bin and 71.16% in the 0.5 m to 1 m bin. This shallow-water transition zone contains the highest pixel densities and exhibits strong spatial gradients, so small vertical errors can shift pixels between adjacent bins, reducing bin-wise accuracy even when the underlying elevation

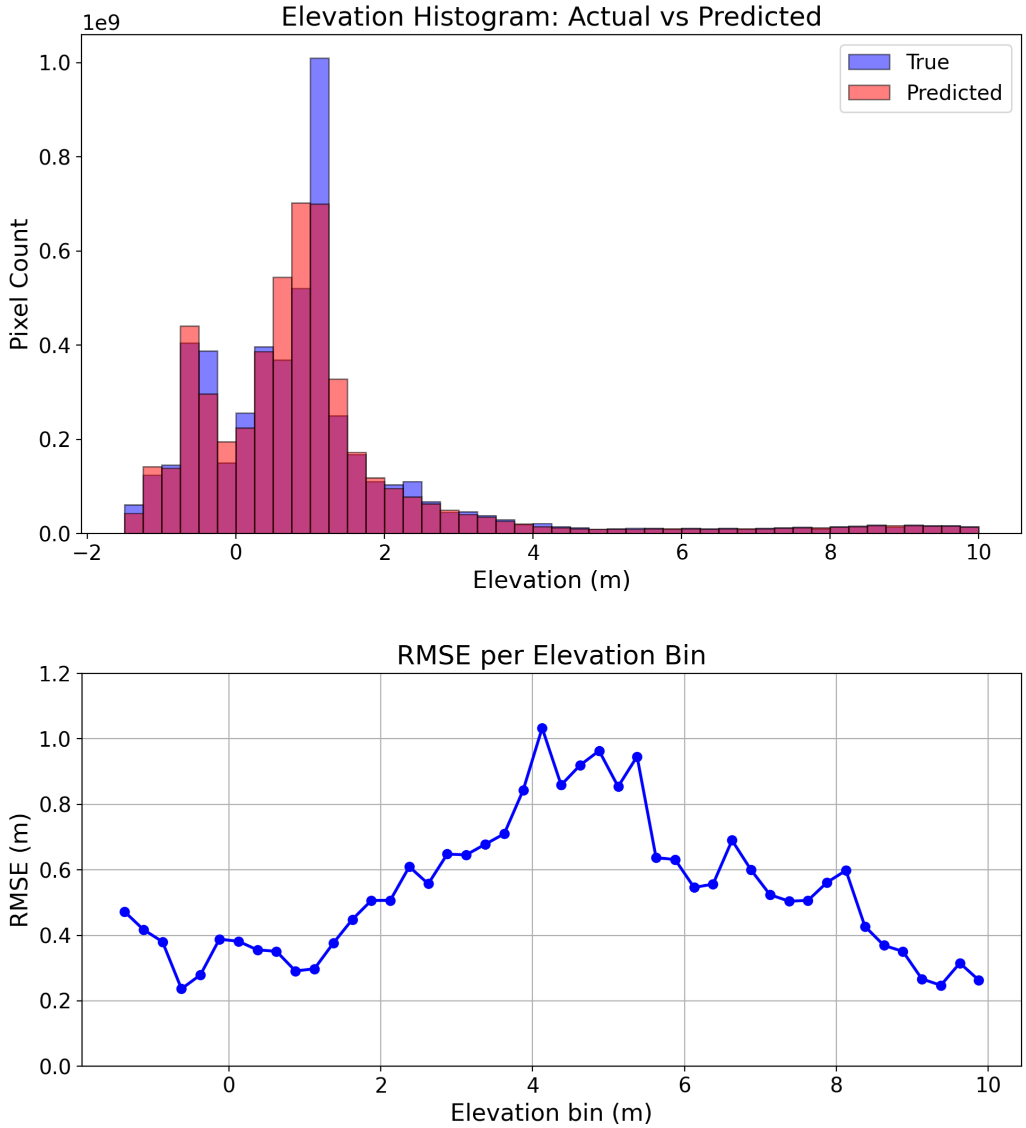


Figure 4.3: (Top) Histogram of actual and predicted elevations (storm-tide or ground-surface) across the full test dataset. The predicted distribution matches the true distribution at most elevations, but shows an under-representation near 1 m, where the true elevation distribution peaks. (Bottom) Root-mean-square error (RMSE) as a function of elevation bin for the full test domain. Errors are largest in the 4 m to 5.5 m elevation range. RMSE decreases sharply in the 0 m to 1 m coastal fringe and remains low through the 1 m to 3 m band, reflecting strong model performance in the low-lying regions that dominate coastal flooding dynamics. RMSE declines again at higher elevations, where the model predicts non-inundated terrain.

structure is well captured.

The 1 m to 3 m bin exhibits the lowest accuracies in the domain, with values of 60.02%, 54.24%, 56.68%, and 71.16% across the 1 m to 3 m bins. This band contains the highest pixel density in the dataset and corresponds to extensive marsh platforms, back-barrier flats, and gently sloping uplands. These areas experience the greatest spatial and hydrodynamic variability during storm events. The updated histogram shows a pronounced under-representation of predicted elevations near 1 m, which contributes to reduced accuracy in this densely populated elevation band. Small vertical differences in this range strongly influence inundation onset, shoreline overtopping, and inland flood propagation, making this region inherently more challenging to predict.

Accuracy improves in the 3 m to 5 m elevation range, reaching 75.91% in the 3 m to 4 m bin and 72.57% in the 4 m to 5 m bin. These elevations represent transitional upland terrain where storm impacts vary substantially between events. The RMSE-per-bin curve shows a distinct error peak in the 4 m to 5.5 m band, reflecting the sensitivity of these elevations to storm-specific forcing and the reduced representation of these terrain types in the training data. Despite this RMSE peak, bin-wise accuracy remains moderate, indicating that most predictions fall within the correct or adjacent elevation bins.

At elevations above 5 m, accuracy increases sharply, reaching 83.73% in the 5 m to 6 m bin and exceeding 95% in the 6 m to 8 m, 8 m to 10 m, 10 m to 12 m, and 12 m to 15.5 m bins. These upland regions remain dry during most storms, and because ground-surface elevation is one of the model inputs, the model reliably preserves the correct non-inundated state. The high accuracy in these bins reflects the model's ability to distinguish dry terrain from storm-affected areas with minimal ambiguity.

The normalized confusion matrix (Figure 4.2) reinforces these elevation-dependent trends. The matrix exhibits strong diagonal structure across most elevation bins, indicating close agreement between predicted and actual values. Misclassifications are concentrated in low-lying transitional zones–particularly near the shoreline and in the 1 m to 3 m band–where small vertical errors can shift a pixel between adjacent bins. Higher-elevation bins show minimal confusion, reflecting the model's ability to reliably identify non-inundated terrain.

Overall, the model performs best in the 0 m to 3 m coastal fringe, which exerts the strongest control on coastal flooding dynamics. Errors are largest in deep or highly variable bathymetric regions and in sparsely sampled mid-elevation terrain (4 m to 5.5 m), but these areas represent a small fraction of the domain. The model's strong performance in the low-elevation coastal fringe demonstrates its suitability for predicting storm-driven water levels and inundation patterns across a wide range of storm conditions.

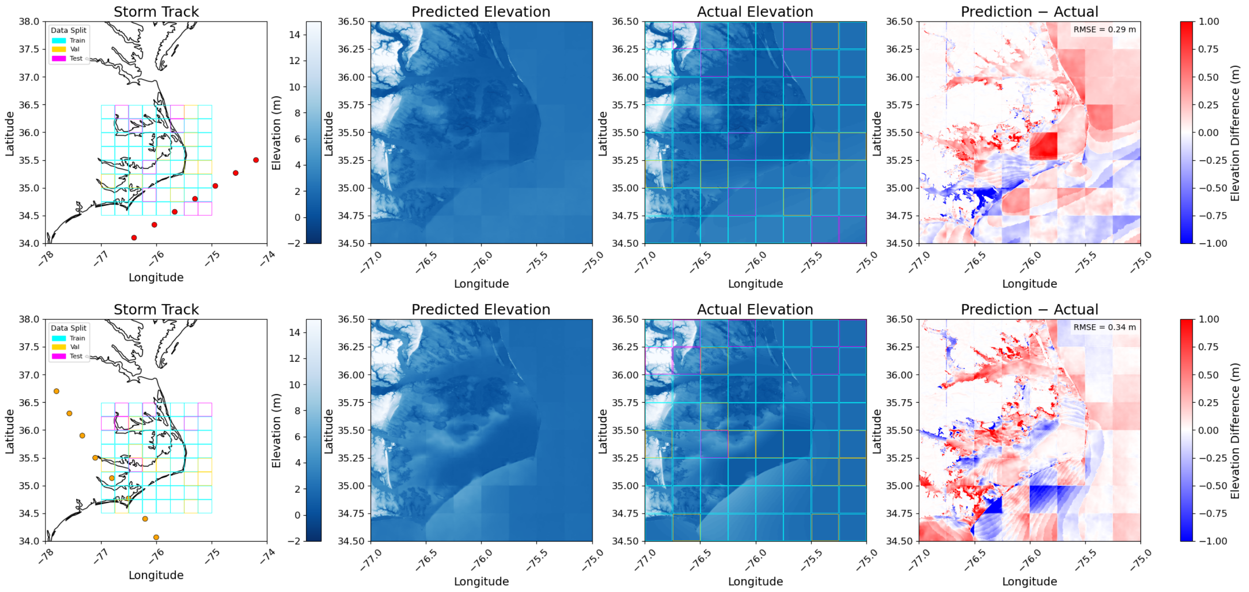


Figure 4.4: Comparison of model performance for two storms with different track orientations: Storm 562 (top, parallel approach) and Storm 1121 (bottom, perpendicular approach). Each row shows (from left to right) the storm track with category coloring, predicted elevation, actual elevation, and the spatial difference (Predicted minus Actual). The storm-track panels include the full train–validation–test tile configuration, with tiles colored cyan for train, gold for validation, and magenta for test, allowing direct comparison of model performance across regions seen during training and those held out for validation and testing. The difference panel uses a diverging blue–red scale in which blue denotes underprediction and red denotes overprediction.

4.1.4 Spatial Metrics

Spatial evaluation provides insight into how the model performs across the full geographic extent of the domain and how prediction accuracy varies with storm characteristics, landscape setting, and elevation structure. Whereas global metrics such as RMSE summarize overall performance, spatial diagnostics reveal where the model tends to over- or under-estimate water levels, how errors cluster across geomorphic features, and how well the model generalizes to unseen storm conditions. To support this analysis, two additional spatial visualizations are included: a domain-wide comparison of predicted and actual elevations, and a corresponding spatial error map (Figure 4.4). Across the full 64-tile mosaic, the spatial error field is dominated by low-magnitude deviations, with most of the domain falling within ±0.1–0.2 m of the true elevation and a domain-wide RMSE of approximately 0.3 m. In several locations, the training tiles actually contain the largest contiguous patches of bias—typically on the order of 0.2–0.3 m—whereas many of the validation and test tiles show smaller, more spatially diffuse deviations. The magnitude and spatial organization of these errors

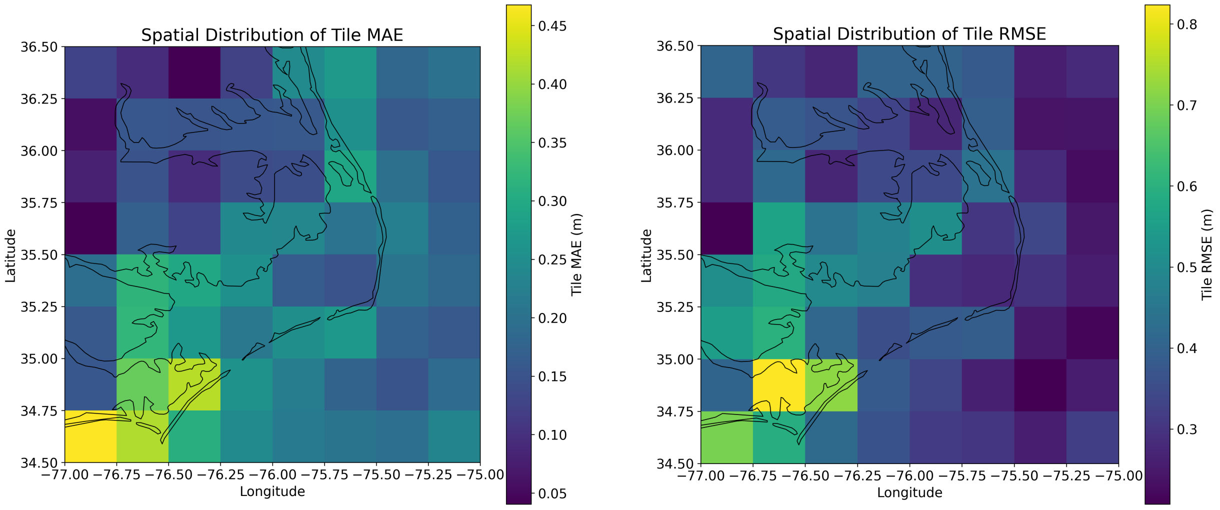


Figure 4.5: Mean absolute error (MAE; left) and root-mean-square error (RMSE; right) computed for each of the 64 tiles that make up the full 4096 × 4096 test-domain extent. Each tile represents a 0.25° × 0.25° region, and the metrics shown here reflect the average elevation error within each tile across all test storms. Spatial patterns in tile-level error highlight regions where the model performs consistently well and regions where prediction uncertainty increases due to complex topography, hydrodynamic variability, or limited training representation.

are broadly consistent across the train, validation, and test tiles: the held-out test regions do not exhibit systematically larger or more coherent error structures than the areas used for training or validation. This pattern suggests that the dominant error structures arise from unresolved geomorphic complexity rather than overfitting to the training domain or degradation in unseen regions. Importantly, these visualizations display the full train–validation–test tile configuration, allowing direct comparison of model behavior across all data splits and illustrating how performance differs between regions the model has seen during training and those reserved for validation and testing. The following analyses examine spatial performance at three complementary scales in the test dataset: (1) domain-wide behavior across all 64 tiles, (2) storm-specific behavior on a representative tile, and (3) systematic spatial patterns of over- and under-prediction.

Domain-Wide Spatial Performance

Across the full 4096-pixel × 4096-pixel domain, the model exhibits strong spatial consistency, with low errors across most of the coastal plain and higher errors concentrated in regions characterized by steep elevation gradients or complex hydrodynamics. Tile-level MAE and

RMSE maps (Figure 4.5) show that the majority of tiles fall within MAE values of 0.05 m to 0.20 m and RMSE values of 0.30 m to 0.45 m, indicating stable performance across barrier islands, back-barrier marshes, tidal channels, and upland terrain. Elevated errors appear in tiles containing sharp coastal transitions or deep estuarine basins, where storm-driven water levels respond nonlinearly to small changes in elevation or bathymetry; these tiles reach MAE values of 0.35 m to 0.45 m and RMSE values approaching 0.7 m to 0.8 m, consistent with the bright-yellow regions in Figure 4.5.

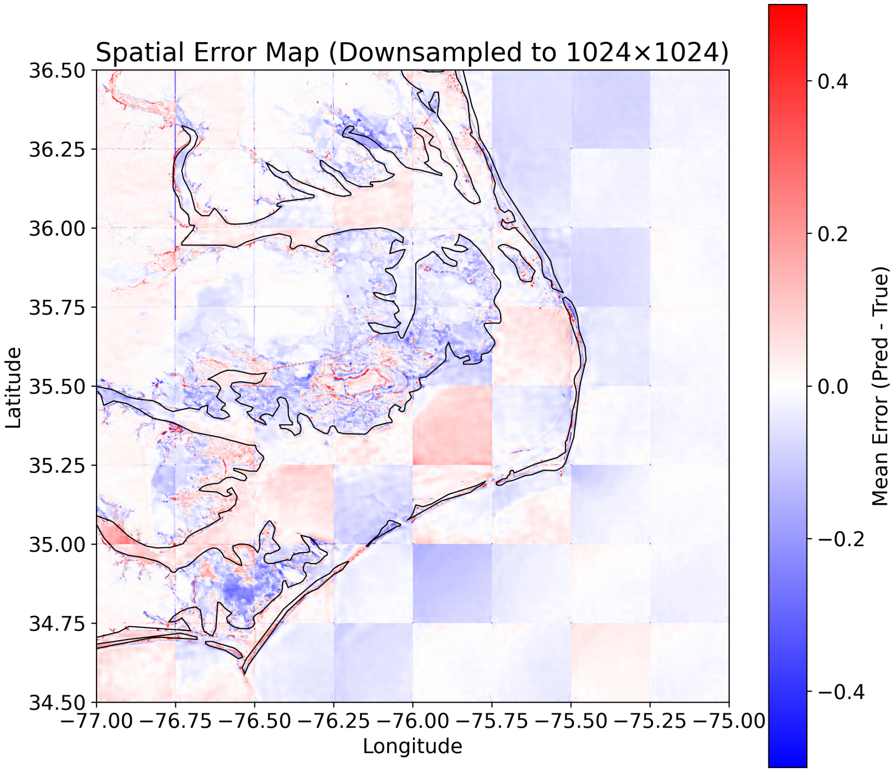


Figure 4.6: Spatial distribution of mean signed error (Predicted minus True) across the test-domain extent. Blue regions indicate systematic under-estimation, particularly along the coastline, while red regions indicate over-estimation in deeper coastal waters and within a localized sector of Pamlico Sound.

Systematic spatial patterns of over- and under-estimation provide additional insight into the model's behavior. The spatial signed-errors (Figure 4.6) shows that most of the domain lies within ±0.2 m of zero bias, indicating that large-scale prediction errors are generally small across the coastal plain. A coherent band of under-estimation appears along the immediate coastline, particularly in narrow shoreline fringes where storm-tide gradients are steep and inundation onset is highly sensitive to small vertical differences; in these regions, mean signed errors typically reach –0.05 m to –0.2 m. In contrast, deeper coastal waters exhibit a modest but spatially consistent under-prediction on the order of 0 to -0.2 m and slight over-prediction around +0.1 m, reflecting areas where large-scale hydrodynamics dominate and terrain constraints exert weaker control on water-level response. The map also reveals a distinct tile within Pamlico Sound that displays the strongest localized positive error in the domain, with mean signed errors of 0.2 m to 0.25 m, indicating a localized breakdown in model performance that may reflect unresolved bathymetry, or storm-specific forcing.

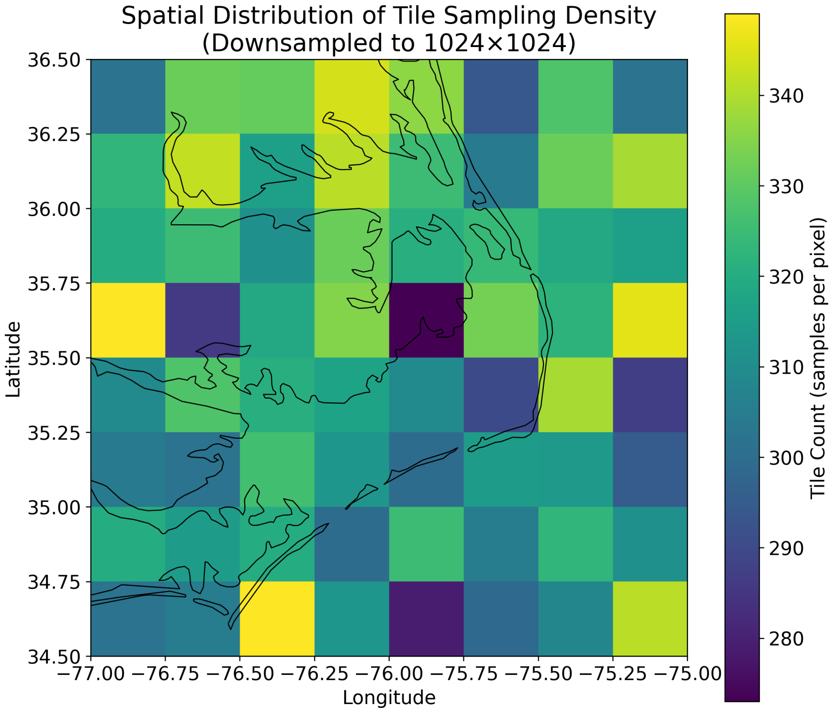


Figure 4.7: Spatial distribution of test-set sampling density across the 64-tile domain. Tiles along the central coastal plain contain the highest number of test samples, while interior estuarine tiles –including the Pamlico Sound tile with the strongest over-estimation – contain fewer test samples due to the storm-based dataset split.

The tile exhibiting the strongest over-estimation in Pamlico Sound also has the fewest samples in the test dataset (Figure 4.7, note the dark-blue tile in central Pamlico Sound). Because each tile receives one sample per storm, a lower test-set count implies a higher proportion of training and validation samples for that tile. Thus, the elevated error in this

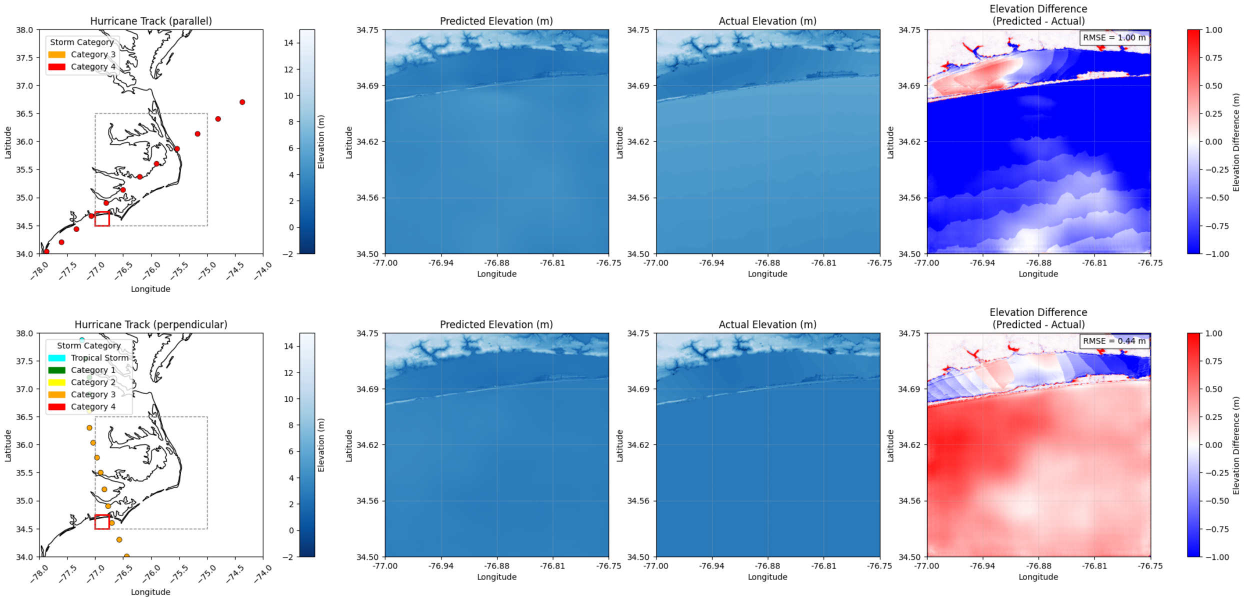


Figure 4.8: Comparison of model performance for Tile 0, with a representative parallel-track – storm 66 (top) and perpendicular-track – storm 806 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual).

region is not caused by limited training exposure. Instead, the reduced number of test samples increases the influence of each individual prediction on the tile-level mean error, making localized biases more pronounced. The strong positive bias in this tile therefore reflects a combination of estuarine hydrodynamic complexity, unresolved bathymetric structure, and the statistical sensitivity introduced by a smaller test-set denominator.

Together, these spatial diagnostics demonstrate that the model represents the dominant patterns of storm-driven inundation while exhibiting predictable deviations in regions with steep coastal gradients, complex estuarine dynamics, or reduced test-set sampling. The combination of tile-level error metrics, signed-error mapping, and sampling-density analysis provides a comprehensive understanding of the model's spatial performance across the coastal landscape.

Representative Storm Behavior on a Single Tile

To illustrate how the model responds to different storm geometries, four tiles (0, 9, 10, and 26) were evaluated using one parallel-track event and one perpendicular-track event for each location (Figures 4.8, 4.9, 4.10 and 4.11, respectively). These tiles were selected because tile 9 exhibits the highest RMSE, tile 0 shows the highest MAE, tile 10 performs poorly across both metrics, and tile 26 performs around testing average. Although each pair of storms intersects the same spatial domain, the model's behavior varies with the storm's approach angle.

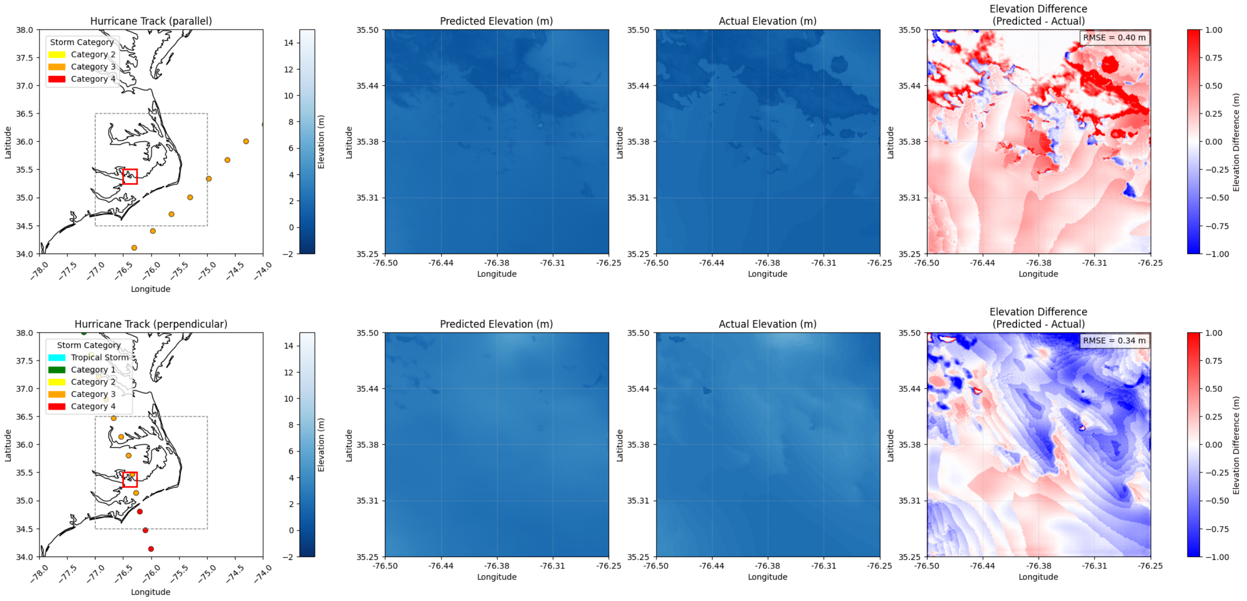


Figure 4.9: Comparison of model performance for Tile 26, with a representative parallel-track – storm 295 (top) and perpendicular-track – storm 806 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual).

For tile 0, the two storms differ in their approach geometry (Figure 4.8), and in this case the parallel storm is the stronger event, reaching the tile as a Category 4 while the perpendicular storm arrives as a more moderate Category 3. Despite the higher intensity of the parallel track, the model performs substantially worse for that case (RMSE = 1.0 m) than for the perpendicular approach (RMSE = 0.44 m). This behavior is consistent with the training dataset, which contains fewer high-intensity storms and therefore provides the model with fewer examples of the extreme surge and wave-driven dynamics associated with stronger events. This tile spans a barrier-island setting with large open-ocean exposure, beach-face topography, and areas of mid- to higher-elevation terrain-features that amplify alongshore gradients during parallel storms and likely contribute to the larger errors. Much of the elevated error also occurs offshore, beyond the barrier island, where deeper water and more energetic dynamics introduce additional variability. In contrast, the more direct, cross-shore forcing of the perpendicular storm produces a cleaner and more coherent surge signal that the model captures with much higher fidelity.

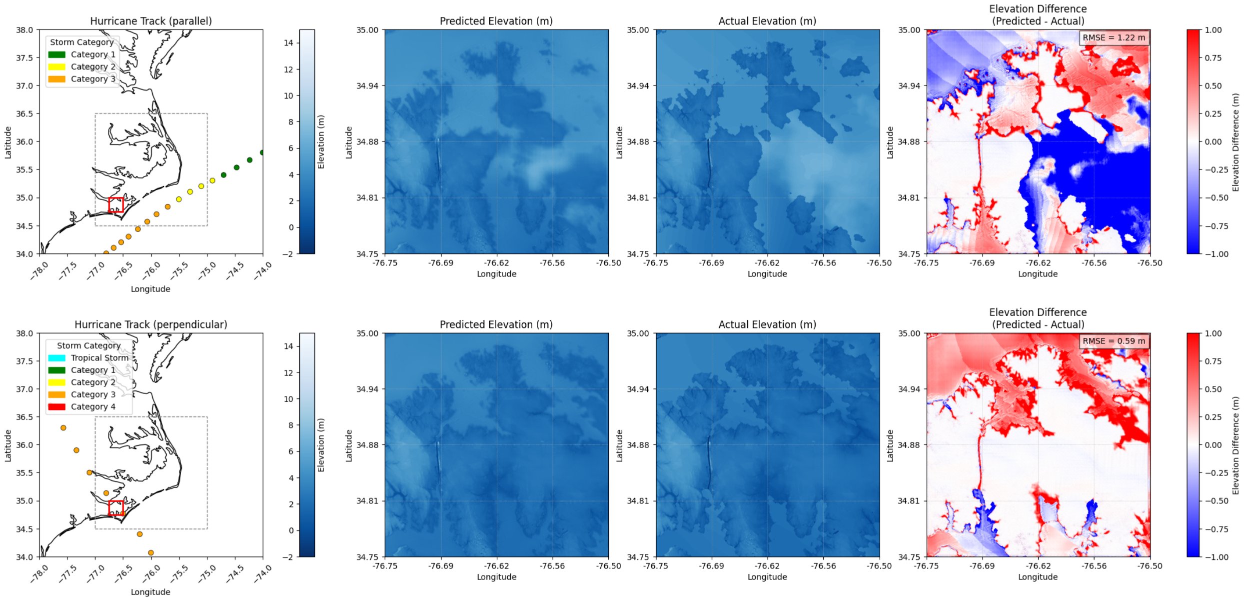


Figure 4.10: Comparison of model performance for Tile 9, with a representative parallel-track – storm 626 (top) and perpendicular-track – storm 806 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual).

For tile 26 (Figure 4.9), another Pamlico sound coastal location with low-lying elevation, the parallel and perpendicular storm tracks impact the area differently. For the perpendicular storm the model predicts RMSE = 0.34 m around the test RMSE = 0.29 m, this storm approaches the barrier islands with a hurricane category 4 which when it hits landfall of our tile it's at a category 3. For the parallel storm its an offshore storm that is a Category 3 for the entire time it nears the tile, the model predicts well here too, with a slight over-prediction of flooding, with RMSE = 0.40 m. This tile performs very well for two strong storms.

For tile 9, the two storms arrive with comparable local intensity (both reaching Category 3 near the tile), but their approach angle leads to very different model performance (Figure 4.10). The parallel track produces a much larger elevation error (RMSE = 1.22 m), suggesting that the alongshore-oriented forcing and more gradual change in storm proximity are harder for the model to represent. In contrast, the perpendicular track approaches the tile more directly, makes landfall shortly afterward, and yields lower error (RMSE = 0.59 m). This contrast indicates that, for tile 9, cross-shore approaches generate a more coherent and predictable surge response, whereas parallel motion introduces spatial gradients and timing offsets that the model struggles to resolve. The magnitude of the RMSE difference also implies that approach angle can dominate over storm intensity in shaping local prediction accuracy for this location.

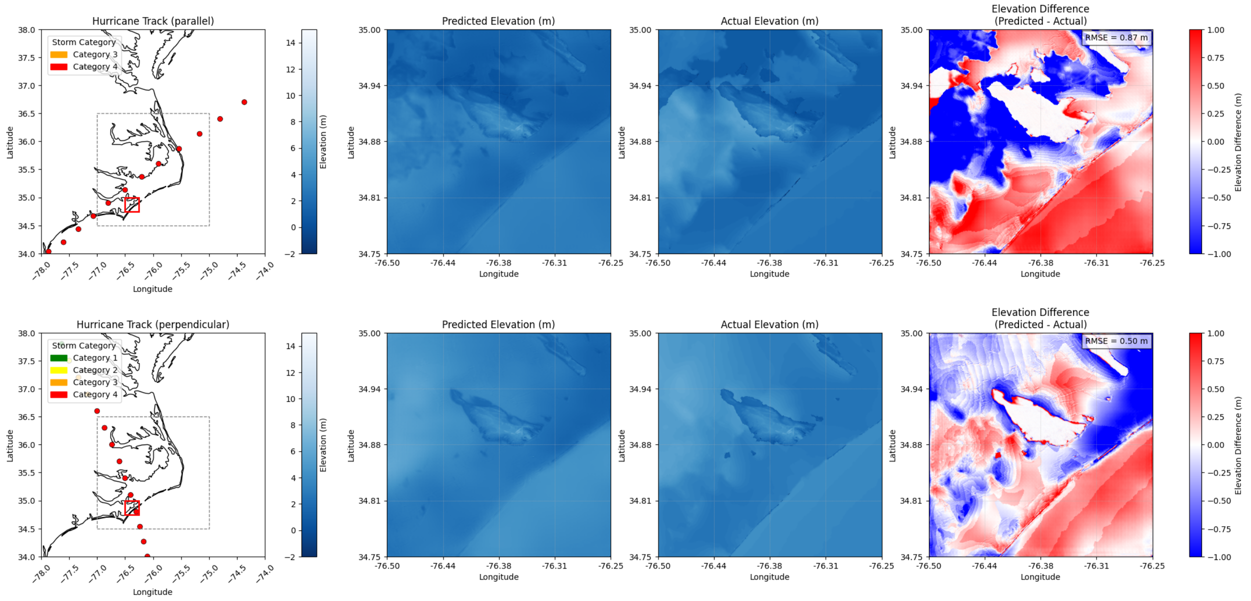


Figure 4.11: Comparison of model performance for Tile 10, with a representative parallel-track – storm 66 (top) and perpendicular-track – storm 70 (bottom) hurricane. Each panel shows the storm track with category coloring, predicted elevation, actual elevation, and the elevation difference (Predicted minus Actual).

For tile 10, both storms reach the domain as high-intensity Category 4 events, but their approach geometry is fundamentally different: one track is roughly parallel to the coastline, while the other is more perpendicular (Figure 4.11). Despite comparable intensity, the model performs slightly better for the perpendicular storm (RMSE = 0.5 m) than for the parallel storm (RMSE=0.87 m). This pattern suggests that the model captures the surge and elevation response more accurately when the storm forces the tile in a direct cross-shore manner, producing a cleaner and more coherent signal. In contrast, the along-track motion of the parallel storm introduces additional complexity—such as stronger alongshore gradients, shifting peak-forcing timing, and more spatially variable surge amplification—that is harder for the model to reproduce. These differences highlight how approach geometry, even for storms of the same intensity, can strongly influence local prediction accuracy.

Spatial Patterns of Over- and Under-Estimation

Systematic spatial patterns of over- and under-estimation provide additional insight into the model's behavior. The spatial mean signed-error map (Figure 4.6) shows that the model tends to under-predict elevations along the coastlines, particularly in narrow coastal fringes and nearshore zones where storm-tide gradients are steep. These regions coincide with the transition between frequently inundated lowlands and deeper coastal waters, where small vertical differences strongly influence inundation extent and timing.

In contrast, the model exhibits a slight but coherent over-prediction in deeper coastal waters offshore of the primary shoreline. These areas are less constrained by topographic variability and more influenced by large-scale hydrodynamics, so modest positive biases in predicted water levels can persist over broad regions.

The spatial error map also reveals a distinct tile within Pamlico Sound where the model massively over-predicts elevations, indicating a localized breakdown in model performance that may reflect unresolved bathymetry, atypical storm forcing, or inconsistencies between the training and test conditions in that sector.

Together, these spatial diagnostics demonstrate that the model captures the dominant patterns of storm-driven inundation while exhibiting predictable deviations in regions with steep coastal gradients, complex estuarine dynamics, or localized data limitations. The combination of domain-wide signed-error mapping, elevation-dependent accuracy, and RMSE-by-bin analysis provides a comprehensive view of the model's spatial performance across the coastal landscape.

4.2 Discussion

4.2.1 Comparison to Process-Based and AI Models

In the context of existing storm-surge and storm-tide modeling systems, the RMSE achieved by the present model compares favorably to both operational numerical models and emerging machine-learning approaches. Traditional physics-based systems like SLOSH, which is designed for computational efficiency and rapid ensemble forecasting, have water-level errors on the order of 0.5 m to 1.0 m, with larger deviations in narrow estuaries, barrier-island inlets, and regions with strong tide-surge interactions (e.g., Forbes et al. 2010; Kerr et al. 2013). In contrast, high-resolution ADCIRC hindcasts – particularly those using unstructured meshes refined to tens of meters in complex coastal settings – achieve RMSE values in the 0.15 m to 0.35 m range for well-observed storms such as Katrina, Ike, and Florence (Dietrich et al. 2011; Pringle et al. 2023). These studies demonstrate that ADCIRC's detailed physics allow it to reproduce observed water levels with high fidelity, while SLOSH provides faster but less spatially resolved guidance. Within this context, the RMSE values achieved by the present model – averaging 0.2722 m for the static configuration – fall between these two operational benchmarks: substantially more accurate than typical SLOSH guidance and approaching the error levels of ADCIRC hindcasts, despite requiring only a fraction of the computational cost.

When compared to prior machine-learning approaches for storm surge and storm tide prediction, the present model occupies a distinct position in terms of spatial resolution, physical completeness, and predictive accuracy. Early neural-network models like Bezuglov et al. (2016), Hashemi et al. (2016), and Lee et al. (2021) produced point-based surge predictions at tide-gauge locations or selected coastal points, with reported errors typically ranging from 0.2 m to 0.6 m depending on storm intensity and gauge location. More recent hybrid frameworks, including Saviz Naeini and Snaiki (2024) and Pachev et al. (2023), achieved similar or smaller errors by training on large libraries of ADCIRC simulations, but these models remained limited to surge-only predictions and did not generate continuous spatial fields. Saviz Naeini and Snaiki (2024) reported a test-set MSE of 0.07 m^2, corresponding to an RMSE of 0.26 m, for peak storm-surge prediction across 289 coastal stations. Likewise, Pachev et al. (2023) demonstrated that their multi-stage surrogate achieved RMSE values of 0.31 m to 0.33 m for inundation-depth predictions at ADCIRC mesh vertices, with slightly higher errors near the wet–dry boundary. The first storm-tide neural network, introduced by Cuevas López et al. (2025), incorporated tidal phase and storm-track evolution but still produced predictions at only nine coastal sites. Their model achieved sub-meter accuracy at all locations, with RMSE values ranging from 0.08 m to 0.19 m, demonstrating that neural networks can learn nonlinear tide–surge interactions when provided with both storm and tidal inputs. Although these point-based and vertex-based models provide valuable insight into the feasibility of data-driven coastal water-level prediction, they do not generate continuous spatial fields and therefore cannot produce inundation maps or resolve fine-scale geomorphic controls on storm tide.

In contrast, the present model produces high-resolution, domain-wide storm-tide fields at a 4096×4096-pixel resolution across 64 tiles, capturing fine-scale geomorphic features and wetting–drying transitions that are not represented in coarse-grid AI surrogates. Its RMSE values place it between operational SLOSH guidance and high-fidelity ADCIRC hindcasts, while offering orders-of-magnitude speedups relative to process-based models. This comparison highlights both the promise and the challenge of spatially explicit neural-network surrogates: while current accuracy approaches that of point-based AI models and exceeds that of SLOSH, further architectural refinement—such as multi-scale encoder–decoder networks, attention mechanisms, or physics-guided training—may be required to match ADCIRC-level fidelity across complex coastal environments.

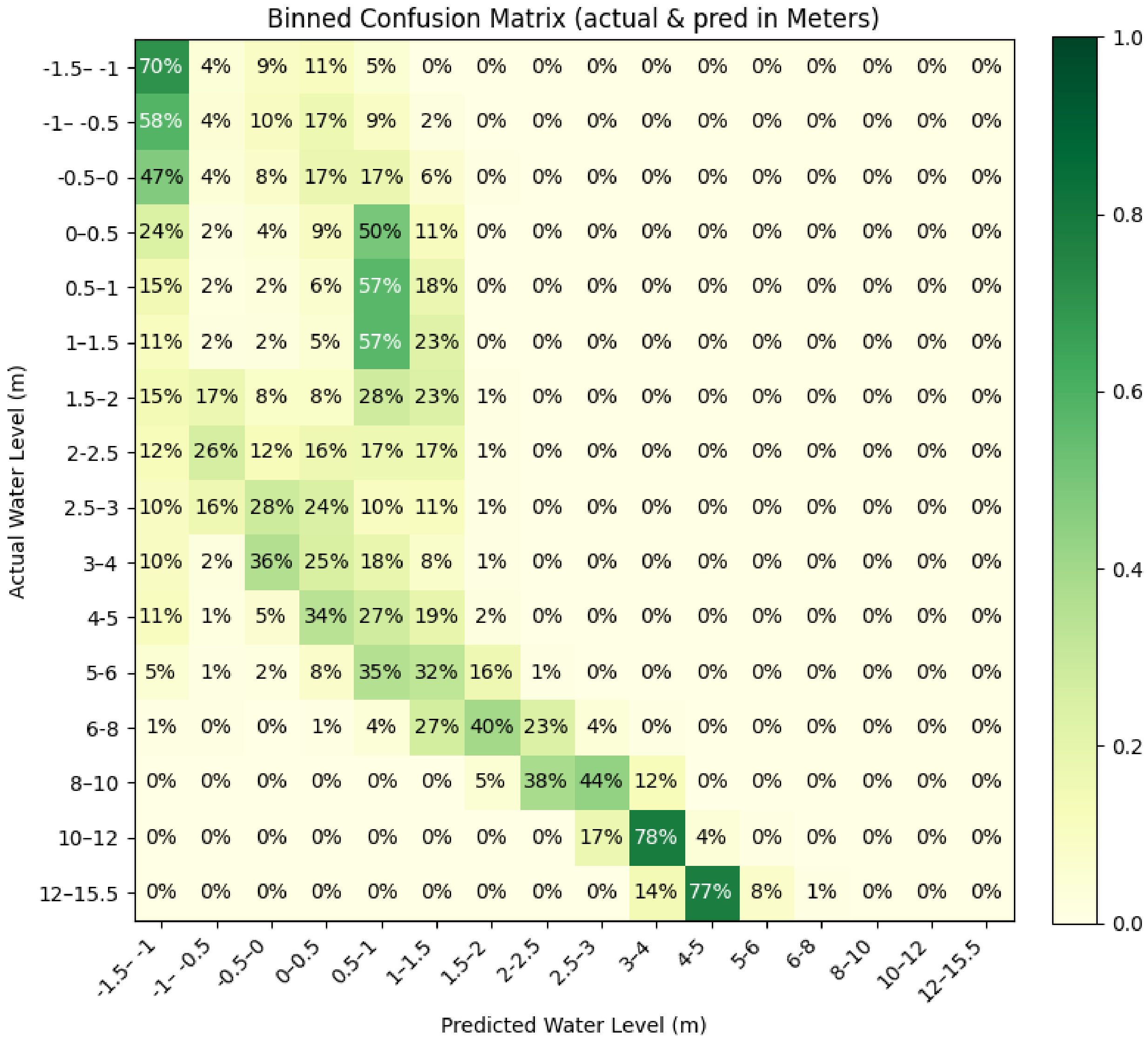
4.2.2 Attempts at Model Improvement

The development of the final model involved evaluating several alternative strategies designed to improve spatial generalization, reduce elevation-dependent error, and enhance performance in the low-lying coastal fringe where inundation dynamics are most sensitive to small elevation differences. Although the static-tile model provided the most stable and accurate predictions, multiple experimental approaches were explored during model development, including elevation-weighted loss functions, the incorporation of Maximum Dissimilarity Algorithm (MDA) tiles, and alternative dataset-shuffling strategies. Each of these approaches targeted a specific limitation observed during early training – such as underrepresentation of certain elevation ranges, insufficient sampling of high-variability regions, or potential storm-specific biases – and therefore provides insight into the model's behavior and the challenges inherent in learning storm-driven coastal inundation patterns. The following subsections summarize these experiments, explain why certain approaches did not yield the expected improvements, and contextualize the performance of the final model within this broader methodological exploration.

Weighted Loss Metrics

Elevation-weighted loss functions have been used in other machine learning applications to emphasize critical regions of the prediction domain-such as sharp gradients, rare events, or physically sensitive thresholds-by assigning larger penalties to errors in user-defined subsets of the data (Cui et al. 2019; Rasp and Lerch 2018). Motivated by this literature, and by earlier diagnostics showing persistent residuals in the –1.5 m to 3 m elevation band, we implemented a weighted Huber loss that increased the penalty for errors in low-lying coastal regions while reducing the penalty at higher elevations (see Methods, 3.4.7). In principle, this approach should have improved performance in the coastal fringe, where small vertical differences strongly influence inundation onset and where many physics-informed ML studies have found targeted weighting to be beneficial.

To ensure a fair comparison and avoid retraining from scratch, the weighted-loss model was hot-started from the 139-epoch static model, which already exhibited strong large-scale skill. This checkpoint represented a stable region of the loss landscape, making it an appropriate point to introduce a more aggressive, elevation-weighted objective without relearning coarse-scale storm-tide structure. However, this also meant that the model entered the weighted-loss regime with a loss landscape shaped by earlier unweighted training, making it more susceptible to instability when the gradient emphasis was abruptly shifted. The weighted-loss model was then trained for an additional 52 epochs, during which both training



	-1.5– -1	-1– -0.5	-0.5–0	0–0.5	0.5–1	1–1.5	1.5–2	2-2.5	2.5–3	3–4	4-5	5-6	6-8	8–10	10–12	12–15.5
-1.5– -1	70%	4%	9%	11%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-1– -0.5	58%	4%	10%	17%	9%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-0.5–0	47%	4%	8%	17%	17%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0–0.5	24%	2%	4%	9%	50%	11%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0.5–1	15%	2%	2%	6%	57%	18%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
1–1.5	11%	2%	2%	5%	57%	23%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
1.5–2	15%	17%	8%	8%	28%	23%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
2-2.5	12%	26%	12%	16%	17%	17%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
2.5–3	10%	16%	28%	24%	10%	11%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
3–4	10%	2%	36%	25%	18%	8%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4-5	11%	1%	5%	34%	27%	19%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%
5-6	5%	1%	2%	8%	35%	32%	16%	1%	0%	0%	0%	0%	0%	0%	0%	0%
6-8	1%	0%	0%	1%	4%	27%	40%	23%	4%	0%	0%	0%	0%	0%	0%	0%
8–10	0%	0%	0%	0%	0%	0%	5%	38%	44%	12%	0%	0%	0%	0%	0%	0%
10–12	0%	0%	0%	0%	0%	0%	0%	0%	17%	78%	4%	0%	0%	0%	0%	0%
12–15.5	0%	0%	0%	0%	0%	0%	0%	0%	0%	14%	77%	8%	1%	0%	0%	0%

Figure 4.12: Binned confusion matrix for the static model retrained with the elevation-weighted loss. The diagonal structure is weaker than in the baseline model (Figure 4.2), and new off-diagonal mis-classification patterns appear in the low-elevation bins, indicating degraded stability after switching to the weighted loss.

and validation curves showed persistent oscillations and no sustained improvement. Across these 52 epochs, training RMSE ranged from 2.77 m down to 0.68 m, while validation RMSE fluctuated between 2.35 m and 0.67 m, demonstrating that the weighted loss introduced substantial instability rather than guiding the model toward improved performance in the targeted elevation range. The fact that training RMSE spiked to 2.77 m—far above the values at the end of the 139-epoch static run—shows that the weighted-loss model effectively abandoned the pretrained solution and re-entered a high-error regime, behaving more like a newly initialized model than one continuing from a stable checkpoint.

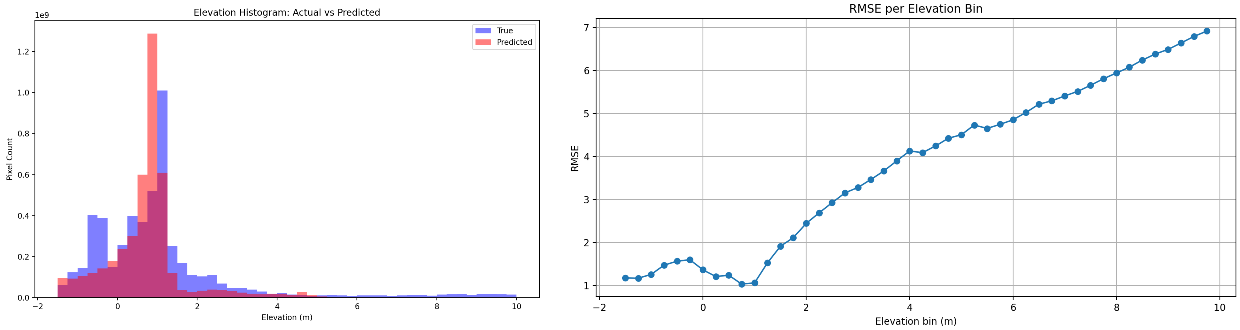


Figure 4.13: (Left) Histogram of actual and predicted elevations for the static weighted-loss test dataset. The predicted distribution retains the same biases as the baseline model, including over-prediction near 1 m and under-prediction at negative elevations. (Right) RMSE as a function of elevation bin for the weighted-loss model. RMSE increases across all elevation ranges, reaching values up to 7 m, indicating an instability introduced by the weighted-loss formulation.

After switching to the weighted loss, the optimization became unstable, and the validation metrics showed large, erratic fluctuations rather than improvement. This degradation is reflected in the supporting diagnostics. The RMSE-per-elevation curve, which remained below 1.2 m for the baseline model, reached values as high as 7 m, indicating a complete breakdown in elevation-conditioned accuracy (Figure 4.13). The characteristic mid-elevation error peak (4 m to 5 m) not only persisted but was amplified.

The elevation histogram likewise showed no correction of the known biases: the model continued to under-predict negative elevations and over-predict near 1 m, with the weighted loss failing to reshape the distribution in any meaningful way. The binned confusion matrix further confirmed the degradation (Figure 4.12). Diagonal agreement did not improve, and new off-diagonal misclassifications pattern emerged in the low-elevation bins that were intended to benefit from the weighting. Instead of tightening predictions in the coastal fringe, the model spread probability mass into adjacent bins, indicating over-correction and loss of stability.

These results show that introducing elevation-dependent weighting late in training distorted the loss landscape enough to destabilize convergence. Because the static model had learned a coherent representation of the coastal fringe under the standard Huber loss, the abrupt shift in gradient structure caused the model to diverge rather than refine. For this reason, the weighted-loss experiment is presented first in the discussion: it was the least successful modification and demonstrated that post-hoc elevation weighting was not a viable refinement strategy for this architecture.

MDA Tiles

The Maximum Dissimilarity Algorithm (MDA) tiles were introduced to increase sampling density in regions with high spatial heterogeneity—particularly along wet–dry boundaries, tidal channels, and barrier-island inlets. As described in the Methods (Section 3.2.2), the MDA procedure selects tile centers by maximizing dissimilarity in a wetness-variability field, ensuring that the training set includes examples from locations where inundation patterns vary sharply across storms. This approach follows earlier work demonstrating that dissimilarity-based sampling can produce representative subsets of complex coastal and wave-climate regimes (e.g., Camus et al. 2011), and aligns with broader geoscientific ML efforts that use diversity-based sampling to expose models to rare or high-gradient physical conditions. The overall intent was for the MDA tiles to help the model learn these transitions more effectively than the uniform 64-tile grid alone.

The MDA-augmented model was trained for 53 epochs, beginning from the same initialization used for the static-tile model. During this run, training RMSE decreased from 4.88 m to 0.36 m, and validation RMSE decreased from 4.80 m to approximately 0.60 m, but both curves plateaued after the first 10–15 epochs. The epoch-by-epoch trajectories showed no sustained downward trend beyond this early phase, indicating that the additional high-variability samples did not meaningfully alter the optimization trajectory or reduce the persistent elevation-dependent errors observed in the static-tile configuration. In contrast to the static-tile model, which underwent several hundred epochs of progressively refined training, the MDA-augmented model was trained for only 53 epochs. This shorter training horizon may have prevented the model from fully integrating the additional high-variability samples, suggesting that the lack of improvement may reflect limited training rather than an inherent shortcoming of the MDA approach.

The elevation-conditioned RMSE curve (Figure 4.14 right) reinforced this interpretation: errors peak in the 4 m to 5 m elevation band – precisely where wet-dry transitions and com-

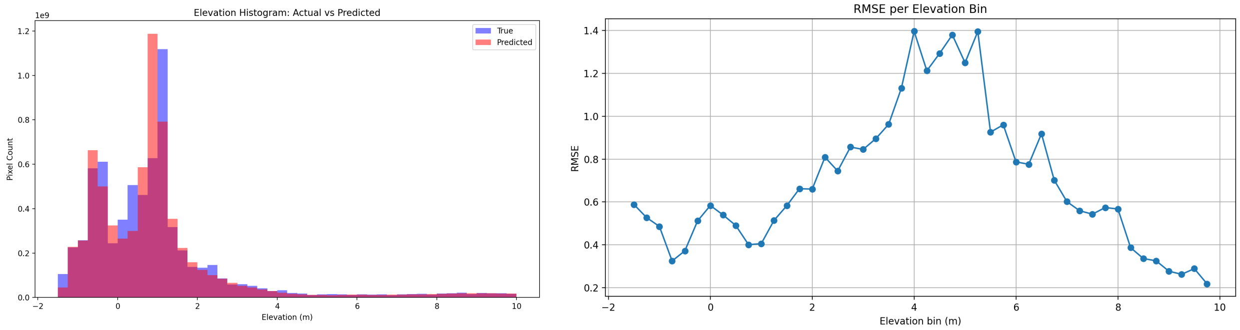


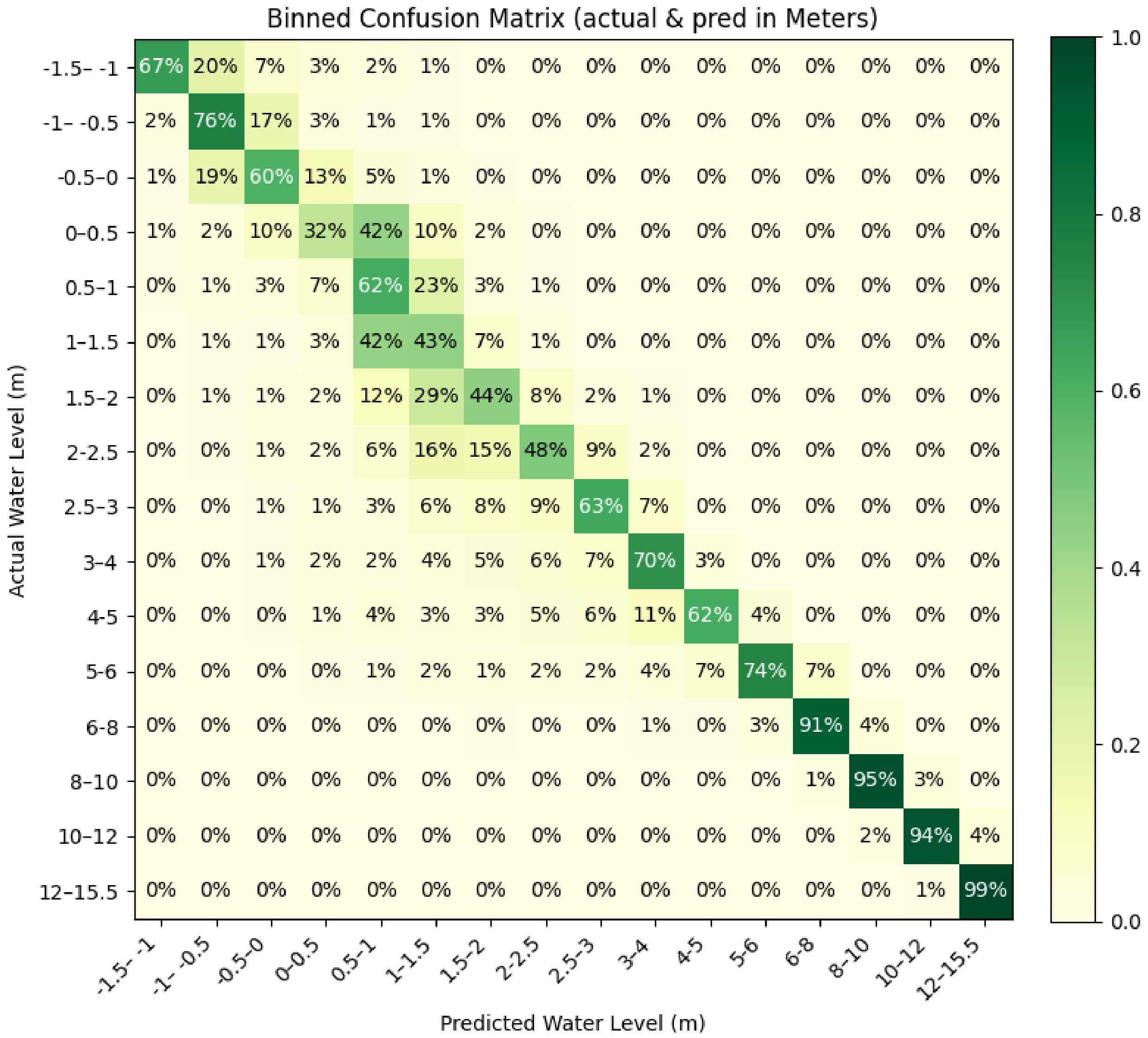
Figure 4.14: (Left) Histogram of actual and predicted elevations for the MDA test dataset. The predicted distribution closely matches the true distribution but shows slight over-representation near 1 m. (Right) RMSE as a function of elevation bin for the MDA model. Errors peak in the 4 m to 5.5 m elevation range, while RMSE remains lowest in the –1.5 m to 2 m band, reflecting strong performance in low-lying regions that dominate coastal flooding dynamics.

plex channel geometries dominate-and the inclusion of MDA tiles did not reduce this peak. The elevation histogram (Figure 4.14 left) shows that the predicted distribution remains close to the true distribution but does not correct the systematic biases at low and negative elevations.

However, model performance indicates that the added MDA tiles did not produce the expected improvements. The binned confusion matrix (Figure 4.15) shows that the MDA-augmented model still predicted most water-level bins, but the off-diagonal structure remained similar to the static-tile model (shown previously in Figure 4.2). The model continued to over-predict low-elevation bins and under-predict negative-elevation bins—patterns consistent with the spatial error maps.

Two structural limitations likely explain this outcome. First, the MDA tiles cover only a small fraction of the domain and are isolated. Because convolutional filters are learned from spatially contiguous patterns, the influence of these isolated high-variability patches is diluted during training. Second, the MDA tiles target regions with extreme wetness variability. While this approach increases exposure to sharp gradients, it also increases exposure to noise, which may have hindered generalization rather than improving it.

Given these results, the final model was trained on the 64 uniform static tiles, which provided more stable and domain-consistent performance. Nonetheless, the MDA approach remains promising. With additional training epochs, or with architectural adjustments that better leverage sparse high-variability samples, the model may learn to generalize the sharp gradients captured within the MDA tiles and improve predictions along the wet-dry front.



	-1.5– -1	-1– -0.5	-0.5–0	0–0.5	0.5–1	1–1.5	1.5–2	2-2.5	2.5–3	3–4	4-5	5-6	6-8	8–10	10–12	12–15.5
-1.5– -1	67%	20%	7%	3%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-1– -0.5	2%	76%	17%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-0.5–0	1%	19%	60%	13%	5%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0–0.5	1%	2%	10%	32%	42%	10%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0.5–1	0%	1%	3%	7%	62%	23%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%
1–1.5	0%	1%	1%	3%	42%	43%	7%	1%	0%	0%	0%	0%	0%	0%	0%	0%
1.5–2	0%	1%	1%	2%	12%	29%	44%	8%	2%	1%	0%	0%	0%	0%	0%	0%
2-2.5	0%	0%	1%	2%	6%	16%	15%	48%	9%	2%	0%	0%	0%	0%	0%	0%
2.5–3	0%	0%	1%	1%	3%	6%	8%	9%	63%	7%	0%	0%	0%	0%	0%	0%
3–4	0%	0%	1%	2%	2%	4%	5%	6%	7%	70%	3%	0%	0%	0%	0%	0%
4-5	0%	0%	0%	1%	4%	3%	3%	5%	6%	11%	62%	4%	0%	0%	0%	0%
5-6	0%	0%	0%	0%	1%	2%	1%	2%	2%	4%	7%	74%	7%	0%	0%	0%
6-8	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	0%	3%	91%	4%	0%	0%
8–10	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	95%	3%	0%
10–12	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	94%	4%
12–15.5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	99%

Figure 4.15: Binned confusion matrix for the MDA model comparing predicted and actual maximum water levels (m). The diagonal structure indicates overall agreement, while persistent off-diagonal patterns in low and negative elevation bins reflect the same biases observed in the static model.

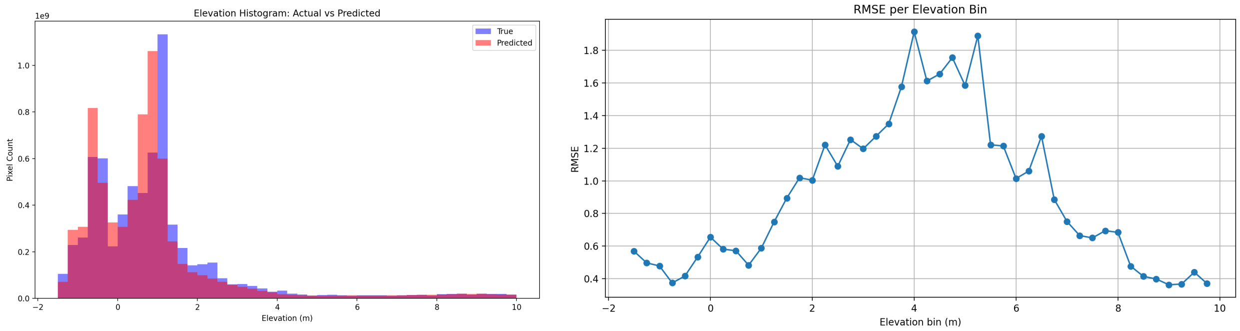


Figure 4.16: (Left) Histogram of actual and predicted elevations for the storm-wise shuffled MDA test dataset. The predicted distribution follows the true distribution but retains slight over-prediction near 1 m and under-prediction at negative elevations. (Right) RMSE as a function of elevation bin for the storm-wise shuffled MDA model. The mid-elevation error peak (4–5 m) persists, with no clear improvement in the elevation ranges targeted by the MDA tiles.

MDA and Storm Shuffling

A final experiment evaluated the performance of the MDA-augmented model when the dataset was partitioned by storm rather than by individual tiles. As described in the Methods (Section 3.2.3), this configuration assigns entire storms to the training, validation, or test sets, ensuring that the model is evaluated on storm events it has never encountered during training. This storm-wise shuffling provided a more meaningful assessment because storm-driven inundation patterns are coherent; tile-wise random shuffling could leak storm-specific spatial structure across splits, potentially inflating validation performance.

The storm-wise shuffled MDA model was trained over four sequential runs, with the fourth run extending training to 171 total epochs. During this final stage, training RMSE decreased from 0.51 m to 0.45 m, while validation RMSE fluctuated between 0.66 m and 0.76 m without exhibiting a sustained downward trend. Although the model showed small improvements during the earliest epochs of this run, the loss curves quickly settled into a narrow oscillatory band, indicating that even with extended training, the model did not converge toward a representation that leveraged the additional high-variability MDA tiles. This behavior mirrors the earlier MDA experiments and suggests that the added tiles did not meaningfully alter the optimization trajectory under storm-wise partitioning.

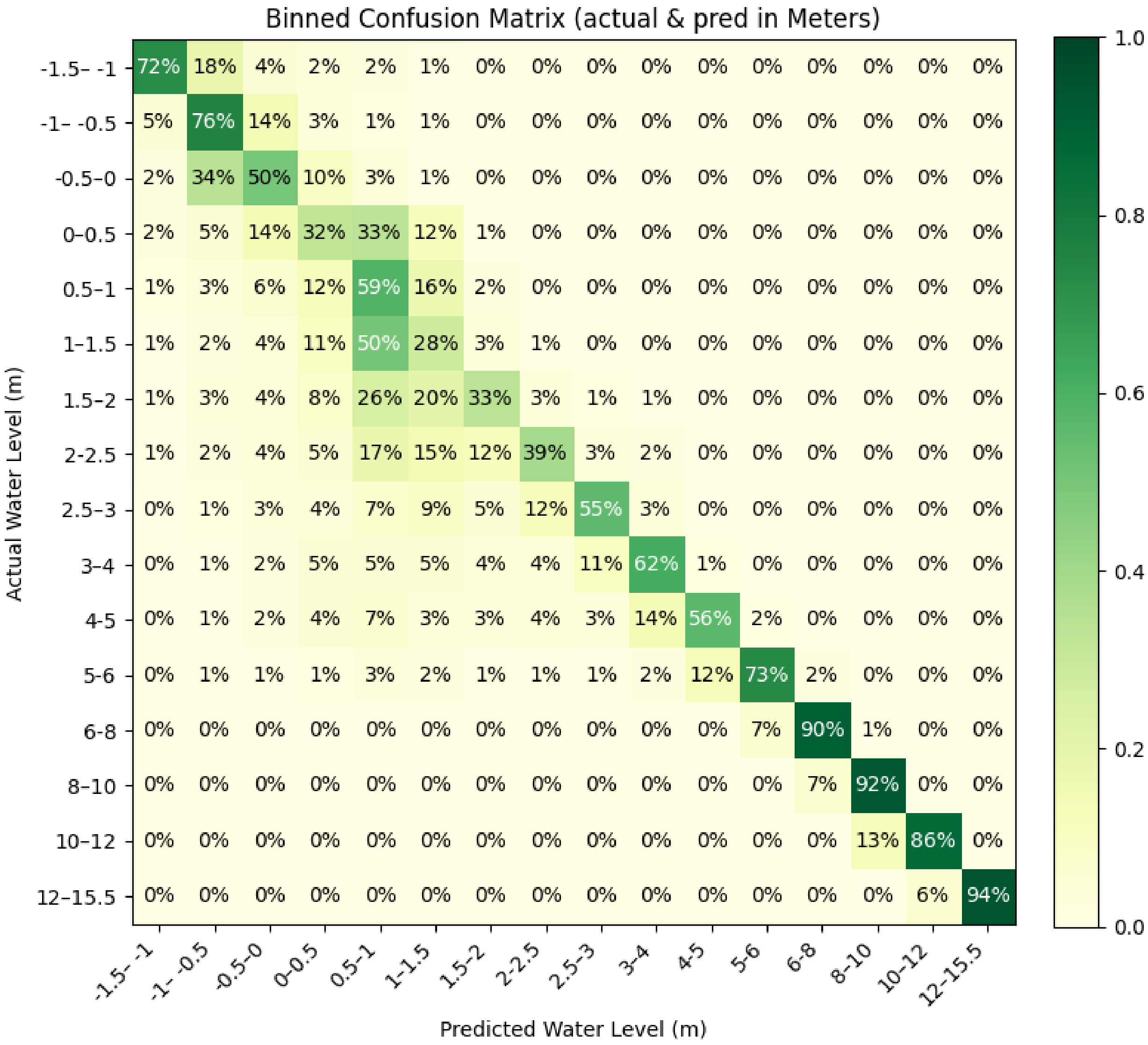
Under storm-wise shuffling, the MDA model exhibited behavior consistent with the earlier MDA experiments: the additional tiles did not improve predictive skill. The elevation histogram shows that the predicted elevation distribution followed the true distribution but retains the same systematic biases – slight over-prediction near 1 m and under-prediction at negative elevations. The RMSE-per-elevation curve maintained the characteristic mid-elevation peak (around 4 m to 5 m), with no reduction in error in the elevation ranges targeted by the MDA tiles. The binned confusion matrix likewise shows strong diagonal structure but no improvement in the low-elevation bins where the MDA tiles were expected to help (Figure 4.17).

These results should not be interpreted as evidence that the MDA tiles are ineffective. Rather, at this stage of training and with the current architecture, the model has not yet learned to leverage the additional high-variability information provided by the MDA tiles. Because the MDA tiles emphasize sharp spatial gradients and wet–dry transitions, they may require more training epochs, different regularization, or architectural adjustments to influence the learned representations meaningfully. With further refinement, the MDA storm shuffled model may outperform the static-only configuration, particularly in regions where spatial gradients dominate storm-driven inundation dynamics.

Summary of Other Models

Taken together, these experiments highlight both the strengths of the static-tile model and the challenges inherent in improving storm-driven inundation prediction through targeted modifications. The weighted-loss experiment demonstrated that late-stage, elevation-dependent penalties can destabilize optimization, a behavior consistent with findings in other environmental ML studies where aggressive spatial or physics-aware weighting can distort the loss landscape or amplify noise in sensitive regions (e.g. Beucler et al. 2021; Karpatne et al. 2017) The MDA tiles – though conceptually well-motivated and aligned with diversity-based sampling strategies shown to improve representation of rare or high-gradient regimes in coastal and wave-climate studies (e.g. Camus et al. 2011) – did not yet yield measurable gains under the current architecture or training duration. The storm-wise shuffling results further emphasize that generalization to unseen events remains a demanding task, echoing broader challenges in spatial and temporal generalization documented in hydrologic, ecological, and atmospheric ML applications (e.g. Mahoney et al. 2023). None of these outcomes suggest that these strategies are ineffective; rather, they indicate that the present configuration has not yet reached the level of representational capacity or training depth needed to benefit from them.

Future work could explore architectures better suited to capturing multi-scale spatial structure, drawing on advances in geoscientific machine learning. UNet-style encoder–decoder networks, originally developed for biomedical image segmentation (Ronneberger et al. 2015), have been successfully adapted for hydrologic and coastal applications requiring fine-scale spatial detail. Attention-based models (Vaswani et al. 2023), have demonstrated



Actual \ Predicted	-1.5– -1	-1– -0.5	-0.5–0	0–0.5	0.5–1	1–1.5	1.5–2	2-2.5	2.5–3	3-4	4-5	5-6	6-8	8–10	10–12	12–15.5
-1.5– -1	72%	18%	4%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-1– -0.5	5%	76%	14%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
-0.5–0	2%	34%	50%	10%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0–0.5	2%	5%	14%	32%	33%	12%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
0.5–1	1%	3%	6%	12%	59%	16%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%
1–1.5	1%	2%	4%	11%	50%	28%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%
1.5–2	1%	3%	4%	8%	26%	20%	33%	3%	1%	1%	0%	0%	0%	0%	0%	0%
2-2.5	1%	2%	4%	5%	17%	15%	12%	39%	3%	2%	0%	0%	0%	0%	0%	0%
2.5–3	0%	1%	3%	4%	7%	9%	5%	12%	55%	3%	0%	0%	0%	0%	0%	0%
3-4	0%	1%	2%	5%	5%	5%	4%	4%	11%	62%	1%	0%	0%	0%	0%	0%
4-5	0%	1%	2%	4%	7%	3%	3%	4%	3%	14%	56%	2%	0%	0%	0%	0%
5-6	0%	1%	1%	1%	3%	2%	1%	1%	1%	2%	12%	73%	2%	0%	0%	0%
6-8	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	7%	90%	1%	0%	0%
8–10	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	7%	92%	0%	0%
10–12	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	13%	86%	0%
12–15.5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	6%	94%

Figure 4.17: Binned confusion matrix for the storm-wise shuffled MDA model comparing predicted and actual maximum water levels (meters). The diagonal structure indicates overall agreement, while the low-elevation bins show similar behavior to earlier MDA experiments, suggesting that the model has not yet leveraged the additional high-variability information provided by the MDA tiles.

strong performance in learning long-range spatial dependencies and are increasingly used in Earth-system modeling. Recent hybrid designs that combine convolutional encoders with Transformer-style attention-such as FourCastNet, a global high-resolution data-driven weather model (Pathak et al. 2022)—show that multi-scale physical processes can be learned effectively when local and global spatial features are modeled jointly. Additional improvements may come from longer training schedules, improved regularization, or physics-guided loss formulations, which have been shown to enhance generalization and physical consistency in environmental modeling tasks (e.g., Karpatne et al. 2017; Read et al. 2019). With these refinements, the model may ultimately surpass the static-tile baseline and more effectively leverage elevation-dependent information and high-gradient coastal features.

Chapter 5

Conclusion

Storm-tide map prediction is a demanding problem that requires representing nonlinear hydrodynamics, complex coastal geometry, and the spatial coherence of storm-driven flooding. This thesis developed and evaluated a machine-learning surrogate model capable of reproducing high-fidelity storm-tide predictions at high spatial resolution and with computational efficiency far exceeding process-based models. By learning from ADCIRC simulations, the model demonstrates that coherent storm-tide dynamics can be approximated by neural networks. The following key findings summarize the contributions of this research:

1. *This surrogate model achieves high accuracy while learning the spatial structure of storm-tide flooding.* The model attains RMSE = 0.2722 m, comparable to other AI storm-surge and storm-tide models (with examples in Section 4.2.1). Because the model is trained on full storm-tide maps, it learns spatial interactions that govern flooding within each tile, including channelized flow, floodplain connectivity, and barrier-island overtopping. This is a significant advance over point-based models, which cannot predict the full extent of spatial dependencies or inundation pathways.

2. *A library of process-model simulations was augmented by using a physics-aware strategy for storm-tide map tiles.* The tiling and augmentation framework developed in this work expands the effective training dataset while preserving the physical integrity of the storm-tide tiles. By focusing the model on how the storms impact geographic areas (tiles), we were are able to expand our ADCIRC results and avoid transformations that would violate coastal geometry or hydrodynamic constraints. The geospatial-referenced input arrays ensure that each tile retains its physical context, allowing the model to learn fine-scale flooding behavior. The 64 static tiles – though not physics-aware – preserve coastal geometry and hydrodynamic structure by maintaining consistent spatial reference. The MDA tiles represent a promising extension of this idea, but additional work is needed to fully leverage their high-variability information. Together, these approaches demonstrate that tile-based sampling can expand training datasets without introducing distortions that violate coastal morphology or inundation dynamics.

3. *The model produces high-resolution, spatially continuous storm-tide fields that describe behavior at community scales.* By learning from full inundation maps, this surrogate model captures fine-scale geomorphic controls that shape storm-tide behavior. It resolves wetting and drying transitions, tidal variability, and localized surge pathways, enabling the generation of spatially explicit inundation fields at 50-m spatial resolution per pixel. These capabilities contrast with recent coarse-resolution data releases and demonstrate the potential for ML surrogates to provide high-fidelity spatial predictions at scales relevant to coastal analysis. This model therefore produces continuous inundation fields that preserve the small-scale coastal morphology and hydrodynamic pathways governing storm-tide evolution, demonstrating the potential for ML surrogates to deliver high-fidelity spatial predictions at relevant scales.

4. *The surrogate model is highly efficient relative to process-based and AI-based alternatives.* Whereas ADCIRC and other physics based simulations require hours to days on high-performance computing systems, this model produces full spatial fields in seconds. In comparison to SLOSH, this model predicts with higher accuracy and within 100 milliseconds. This efficiency exceeds that of SLOSH and is competitive with or faster than other deep-learning storm-surge models. The computational speed enables rapid scenario exploration, ensemble generation, and uncertainty quantification—tasks that are prohibitively expensive with traditional process models.

5. *Machine-learning surrogates can preserve physical realism when trained on high-fidelity models.* Although the surrogate does not explicitly solve the shallow-water equations, it inherits physical structure from ADCIRC through supervised learning. The model reproduces spatial inundation patterns, surge pathways, and nonlinear tide–surge interactions, demonstrating that ML surrogates can augment–rather than replace–physics-based modeling by providing fast approximations of complex hydrodynamic behavior.

6. *Experiments with alternative model configurations provide guidance for future surrogate development.* Experiments with weighted losses, MDA tiles, and storm-wise shuffling show that targeted modifications do not automatically improve performance. Weighted losses can destabilize optimization; MDA tiles require additional architectural or training refinements; and storm-wise shuffling highlights the difficulty of generalizing to unseen storms. These results align with broader challenges in geoscientific ML – such as spatial leakage, limited training diversity, and sensitivity to high-gradient regions – and provide practical guidance for researchers developing coastal surrogate models. They underscore the importance of spatially coherent validation strategies and the need for architectures capable of leveraging high-variability training samples.

In the short term, several changes will strengthen both the surrogate model and a forthcoming journal manuscript. First, adopting storm-wise shuffling as the primary evaluation strategy is essential for avoiding spatial leakage and ensuring that model skill is assessed under realistic generalization conditions. Second, the MDA-augmented model should be retrained for longer durations, as the current results may reflect under-training rather than conceptual limitations; extended training schedules, staged learning rates, or hot-starting from the static-tile model may allow the network to better exploit the high-variability information encoded in the MDA tiles. Finally, incorporating uncertainty quantification – through dropout or lightweight ensemble methods – will provide a more complete characterization of surrogate behavior and strengthen the study's methodological rigor.

Looking further ahead, several avenues offer promising opportunities for advancing storm-tide surrogate modeling. Expanding the training dataset to include a larger number of storms, particularly high-impact and catastrophic events, would improve the model's ability to learn nonlinear surge amplification mechanisms and generalize across a broader range of meteorological conditions. Increasing the spatial resolution of the training maps represents another important direction; moving from 50 m toward finer grids would enable the surrogate to resolve additional physical processes such as narrow inlet dynamics, small-scale overtopping pathways, and marsh-channel interactions. Architectural improvements – such as UNet-style encoder–decoders, attention-based models, or hybrid CNN–Transformer designs – may further enhance the model's ability to capture multi-scale spatial structure, especially when paired with higher-resolution inputs or MDA-based sampling. Additional long-term opportunities include integrating physics-guided loss terms or constraints, developing multi-resolution or nested-tile strategies to capture both basin-scale and fine-scale dynamics, and exploring transfer learning across coastlines to build more generalizable coastal surrogate models. Together, these directions outline a clear path toward more accurate and more applicable storm-tide prediction tools.

REFERENCES

Adeli, E., Sun, L., Wang, J., and Taflanidis, A. A. (2022). An advanced spatio-temporal convolutional recurrent neural network for storm surge predictions. *arXiv preprint arXiv:2204.09501*.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Albemarle-Pamlico National Estuary Partnership (2021). Our Estuary. `https://apnep.nc.gov/our-estuary`. [Retrieved 18 October 2021].

Army Corps of Engineers, U. (2019). Gridded Surface Subsurface Hydrologic Analysis. `https://chl.erdc.dren.mil/gridded-surface-subsurface-hydrologic-analysis/`. [Retrieved 26 April 2019].

Awasthi, S., Phartiyal, G. S., Varade, D., and Jain, K. (2026). A Maxout-enhanced robust deep convolutional neural network model for flood mapping using Sentinel-1 SAR data. *Physics and Chemistry of the Earth, Parts A/B/C*, 143.

Ayyad, M., Hajj, M. R., and Marsooli, R. (2022). Artificial intelligence for hurricane storm surge hazard assessment. *Ocean Engineering*, 245:110435.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2016). Segnet: A deep convolutional encoder-decoder architecture for image segmentation.

Bai, L.-H. and Xu, H. (2022). Accurate storm surge forecasting using the encoder–decoder long short term memory recurrent neural network. *Physics of Fluids*, 34(1):016601.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling.

Becker, A., Hallisey, N., Kalaidjian, E., Stempel, P., and Rubinoff, P. (2021). The hazard consequence prediction system: A participatory action research approach to enhance emergency management. *Journal of Homeland Security and Emergency Management*.

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., and Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*.

Bezuglov, A., Blanton, B., and Santiago, R. (2016). Multi-Output Artificial Neural Network for Storm Surge Prediction in North Carolina.

Bilskie, M. V., Asher, T. G., Miller, P. W., Fleming, J. G., Hagen, S. C., and Luettich, R. A. (2022). Real-time simulated storm surge predictions during hurricane michael (2018). *Weather and Forecasting*, 37:1085 – 1102.

Bilskie, M. V. and Luettich, R. A. (2024). The role of advection in storm surge for hurricane michael (2018). *Journal of Geophysical Research: Oceans*, 127:e2024JC021105.

Bloemendaal, N., Haigh, I. D., de Moel, H., Muis, S., Haarsma, R. J., and Aerts, J. C. J. H. (2020). Generation of a global synthetic tropical cyclone hazard dataset using storm. *Scientific Data*, 7(40).

Borovykh, A., Bohte, S., and Oosterlee, C. W. (2018). Conditional time series forecasting with convolutional neural networks.

Burston, J., Ware, D., and Tomlinson, R. (2015). The real-time needs of emergency managers for tropical cyclone storm tide forecasting: results of a participatory stakeholder engagement process. *Natural Hazards*.

Camus, P., Mendez, F. J., Medina, R., and Cofiño, A. S. (2011). Analysis of clustering and selection algorithms for the study of multivariate wave climate. *Coastal Engineering*, 58(6):453–462.

Cangialosi, J. P. and Alaka, L. (2024). National hurricane center tropical cyclone report, hurricane idalia. Technical report, National Hurricane Center.

Chao, W.-T. and Young, C.-C. (2022). Accurate storm surge prediction with a parametric cyclone and neural network hybrid model. *Water*, 14(1):96.

Chen, K., Kuang, C., Wang, L., Chen, K., Han, X., and Fan, J. (2022). Storm surge prediction based on long short-term memory neural network in the east china sea. *Applied Sciences*, 12(1):181.

CIRES (2014). Continuously Updated Digital Elevation Model (CUDEM) - 1/9 Arc-Second Resolution Bathymetric-Topographic Tiles. `https://coast.noaa.gov/htdata/raster2/elevation/NCEI_ninth_Topobathy_2014_8483/`. Accessed 15 July 2021.

Coastal Emergency Risks Assessment (2024). `https://cera.coastalrisk.live/`. [Retrieved 30 September 2024].

Corbett, D., Walsh, J., Riggs, S., Ames, D., and Culver, S. (2008). Shoreline change within the albemarle-pamlico estuarine system, north carolina.

Cuevas, T. and Dietrich, J. (2025). Library of Peak Storm Tides for 1800 Synthetic Scenarios of Tropical Cyclones.

Cuevas López, T. A., Tucker, B. J., Dietrich, J. C., Anderson, D. L., and Mariegaard, J. S. (2025). Neural network predictions of peak storm tides due to tropical cyclones. *Ocean Modelling*, 197.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples.

Cyriac, R. and Dietrich, J. C. (2018). CARTHE-II: Coupled wave-circulation results from the SWAN+ADCIRC hindcast models for the North Carolina coast during Hurricane Arthur (2014). Distributed by: Gulf of Mexico Research Initiative Information and Data Cooperative (GRIIDC), Harte Research Institute, Texas A&M University – Corpus Christi. Available from: `https://data.gulfresearchinitiative.org/data/R4.x265.246:0001`.

Dietrich, J. C., Tanaka, S., Westerink, J. J., Dawson, C. N., Luettich, R. A., Zijlema, M., Holthuijsen, L. H., Smith, J. M., Westerink, L. G., and Westerink, H. J. (2012). Performance of the unstructured-mesh, SWAN+ADCIRC model in computing hurricane waves and surge. *Journal of Scientific Computing*, 52:468–497.

Dietrich, J. C., Zijlema, M., Westerink, J. J., Holthuijsen, L. H., Dawson, C. N., Luettich, R. A., Jensen, R. E., Smith, J. M., Stelling, G. S., and Stone, G. W. (2011). Modeling hurricane waves and storm surge using integrally-coupled, scalable computations. *Coastal Engineering*, 58:45–65.

Doodson, A. T. (1921). The harmonic development of the tide-generating potential. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character.*

Doodson, A. T. (1956). Tides and storm surges in a long uniform gulf. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences.*

Dresback, K. M., Fleming, J. G., Blanton, B. O., Kaiser, C., Gourley, J. J., Tromble, E. M., Luettich, R. A., Kolar, R. L., Hong, Y., Van Cooten, S., Vergara, H. J., Flamig, Z., Lander, H. M., Kelleher, K. E., and Nemunaitis-Monroe, K. L. (2013). Skill assessment of a real-time forecast system utilizing a coupled hydrologic and coastal hydrodynamic model during Hurricane Irene (2011). *Continental Shelf Research*, 71:78–94.

Emanuel, K., Ravela, S., Vivant, E., and Risi, C. (2006). A statistical deterministic approach to hurricane risk assessment. *Bulletin of the American Meteorological Society*, 87:299–314.

Federal Emergency Management Agency (2019). Guidance for flood risk analysis and mapping: Determination of wave characteristics. Technical report, Federal Emergency Management Agency.

FEMA (2021). Flood risk study engineering library. `https://hazards.fema.gov/wps/portal/frisel`. [Retrieved 4 February 2021].

FEMA (2026). National hurricane program tools.

Forbes, C., Luettich, R. A., Mattocks, C. A., and Westerink, J. J. (2010). A retrospective evaluation of the storm surge produced by Hurricane Gustav (2008): Forecast and hindcast results. *Wea. Forecasting*, 25:1577–1602.

Galanis, N.-I., Vafiadis, P., Mirzaev, K.-G., and Papakostas, G. A. (2022). Convolutional Neural Networks: A Roundup and Benchmark of Their Pooling Layer Variants. *Algorithms*.

Gillies, S. et al. (2013). Rasterio: geospatial raster i/o for Python programmers.

Glahn, B., Taylor, A., Kurkowski, N., and Shaffer, W. A. (2009). The role of the SLOSH model in National Weather Service storm surge forecasting. *National Weather Digest*, 33:3–14.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). DRAW: A Recurrent Neural Network For Image Generation.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*.

Harris, D. L. (1963). Characteristics of the hurricane storm surge. Technical Report Technical Paper No. 48, U.S. Department of Commerce and Weather Bureau, Washington D.C.

Hashemi, M. R., Spaulding, M. L., Shaw, A., Farhadi, H., and Lewis, M. (2016). An efficient artificial intelligence model for prediction of tropical storm surge. *Natural Hazards*, 82:471–491.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. 2nd edition.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Holland, R. W. (1980). An analytic model of the wind and pressure profiles in hurricanes. *Monthly Weather Review*, 108:1212–1218.

Hope, M. E., Westerink, J. J., Kennedy, A. B., Kerr, P. C., Dietrich, J. C., Dawson, C. N., Bender, C. J., Smith, J. M., Jensen, R. E., Zijlema, M., Holthuijsen, L. H., Luettich Jr, R. A., Powell, M. D., Cardone, V. J., Cox, A. T., Pourtaheri, H., Roberts, H. J., Atkinson, J. H., Tanaka, S., Westerink, H. J., and Westerink, L. G. (2013). Hindcast and Validation of Hurricane Ike (2008) Waves, Forerunner, and Storm Surge. *Journal of Geophysical Research: Oceans*, 118:4424–4460.

Horsburgh, K. J. and Wilson, C. (2007). Tide-surge interaction and its role in the distribution of surge residuals in the North Sea. *Journal of Geophysical Research*, 112:1–13.

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

Ian, V.-K., Tse, R., Tang, S.-K., and Pau, G. (2023). Bridging the Gap: Enhancing Storm Surge Prediction and Decision Support with Bidirectional Attention-Based LSTM. *Atmosphere*.

Irish, J. L., Resio, D. T., and Ratcliff, J. J. (2008). The influence of storm size on hurricane surge. *Journal of Physical Oceanography*, 38:2003–2013.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.

Jelesnianski, C. P., Chen, J., and Shafer, W. A. (1992). SLOSH: Sea, Lake, and Overland Surges from Hurricanes. Technical report, U.S. Dep. of Commer., Natl. Oceanic and Atmos. Admin., Natl. Weather Serv., Silver Spring, Maryland.

Karpatne, A., Watkins, W., Read, J., and Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling.

Kerr, P. C., Martyr, R. C., Donahue, A. S., Hope, M. E., Westerink, J. J., Luettich Jr, R. A., Kennedy, A. B., Dietrich, J. C., Dawson, C. N., and Westerink, H. J. (2013). US IOOS coastal and ocean modeling testbed: Evaluation of tide, wave, and hurricane surge response sensitivities to mesh resolution and friction in the Gulf of Mexico. *Journal of Geophysical Research: Oceans*, 118:4633–4661.

Kim, S., Matsumi, Y., Pan, S., and Mase, H. (2016). A real-time forecast model using artificial neural network for after-runner storm surges on the tottori coast, japan. *Ocean Engineering*, 122:44–53.

Kim, S., Pan, S., and Mase, H. (2019). Artificial neural network-based storm surge forecast model: Practical application to sakai minato, japan. *Applied Ocean Research*, 91:101871.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kinnmark, I. (1986). The shallow water wave equations: Formulation, analysis and application. In Brebbia, C. A. and Orszag, S. A., editors, *Lecture Notes in Engineering*, volume 15, pages 12–26. Springer-Verlag.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*.

Knapp, K. R., Diamond, H. J., Kossin, J. P., Kruk, M. C., Schreck, C. J., et al. (2018). International best track archive for climate stewardship (ibtracs) project, version 4. *NOAA National Centers for Environmental Information*, 10.

Knowles, J. S., Dietrich, J. C., Elkut, A. E., Puleo, J. A., Shi, F., and Tateosian, L. G. (2025). Ranges of peak storm tides between open-coast and bay locations. *Journal of Geophysical Research: Oceans*, 130.

Kopp, R. E., Horton, B. P., Kemp, A. C., and Tebaldi, C. (2015). Past and future sea-level rise along the coast of north carolina, usa. *Climatic Change*, 132:693–707.

Kratzert, F., Klotz, D., Brenner, C., Karsten, S., and Herrnegger, M. (2018). Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems.*

Lazo, J. K., Waldman, D. M., Morrow, B. H., and Thacher, J. A. (2010). Household Evacuation Decision Making and the Benefits of Improved Hurricane Forecasting: Developing a Framework for Assessment.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE.*

Lee, J.-W., Irish, J. L., Bensi, M. T., and Marcy, D. C. (2021). Rapid prediction of peak storm surge from tropical cyclone track time series using machine learning. *Coastal Engineering*, 170:104024.

Lin, J., Xie, L., Pietrafesa, L. J., Ramus, J. S., and Paerl, H. W. (2007). Water quality gradients across Albemarle-Pamlico Estuarine system: Seasonal variations and model applications. *Journal of Coastal Research*, 23:213–229.

Lin, N., Emanuel, K., Oppenheimer, M., and Vanmarcke, E. (2012). Physically based assessment of hurricane surge threat under climate change. *Nature Climate Change*, 2:462–467.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation.

Luettich, R. A., Carr, S. D., Reynolds-Fleming, J. V., Fulcher, C. W., and McNinch, J. E. (2002). Semi-diurnal seiching in a shallow, micro-tidal lagoonal estuary. *Continental Shelf Research*, 22:1669–1681.

Luettich, R. A., Westerink, J. J., and Scheffner, N. W. (1992). ADCIRC: An advanced three-dimensional circulation model for shelves coasts and estuaries, report 1: Theory and methodology of ADCIRC-2DDI and ADCIRC-3DL. Technical report, United States Army Corps of Engineers.

Mahoney, M. J., Johnson, L. K., Silge, J., Frick, H., Kuhn, M., and Beier, C. M. (2023). Assessing the performance of spatial cross-validation approaches for models of spatially structured data.

Meyer, R., Baker, J., Broad, K., Czajkowski, J., and Orlove, B. (2014). The dynamics of hurricane risk perception: Real-time evidence from the 2012 atlantic hurricane season. *Bull. Amer. Meteor. Soc.*, 95:1389–1404.

Meyer, R., Broad, K., Orlove, B., and Petrovic, N. (2013). Dynamic simulation as an approach to understanding hurricane risk response: Insights from the stormview lab. *Risk Analysis*.

Morrow, B., Lazo, J., Rhome, J., and Feyen, J. (2015). Improving storm surge risk communication: Stakeholder perspectives. *Bulletin of the American Meteorological Society*, 96:35–48.

Morss, R. E., Demuth, J. L., and Lazo, J. K. (2008). Communicating uncertainty in weather forecasts: A survey of the us public. *Weather and forecasting*, 23(5):974–991.

Morss, R. E. and Hayden, M. H. (2010). Storm surge and "certain death": Interviews with texas coastal residents following hurricane ike. *Weather, Climate, and Society*.

Nakamura, J., Lall, U., Kushnir, Y., Harr, P. A., and McCreery, K. (2021). Early Season Hurricane Risk Assessment: Climate-Conditioned HITS Simulation of North Atlantic Tropical Storm Tracks. *Journal of Applied Meteorology and Climatology*.

National Hurricane Center (2016). Potential storm surge flooding map. `http://www.nhc.noaa.gov/surge/inundation/`. [Retrieved 20 July 2016].

National Hurricane Center (2018). Hurricane florence: September 14, 2018. `https://www.weather.gov/ilm/HurricaneFlorence`. [Retrieved 10 May 2021].

National Hurricane Center (2025). 2025 tropical cyclone advisory archive.

National Hurricane Center (2026). How to read the forecast/advisory.

National Oceanic and Atmospheric Administration (2026). Data access viewer. [Retrieved 14 March 2026].

NCEI, N. C. f. E. I. (2023). State of the Climate Monthly Overview - National Overview.

OCM Partners (2026). 1851 - 2020 usgs coned topobathy dem (compiled 2022): Coastal carolinas from 2010-06-15 to 2010-08-15. `https://www.fisheries.noaa.gov/inport/item/67013`. [Retrieved 1 Sept 2025].

Pachev, B., Arora, P., del Castillo-Negrete, C., Valseth, E., and Dawson, C. (2023). A framework for flexible peak storm surge prediction. *Coastal Engineering*, 186:104406.

Pachev, B., Arora, P., and Zhao, J. (2026). UNet5 and Other Vision Architectures for Peak Storm Surge, in Global Storm Surge Machine Learning Models.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, P., Kashinath, K., and Anandkumar, A. (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators.

Pena, B. and Huang, L. (2021). Wave-GAN: A deep learning approach for the prediction of nonlinear regular wave loads and run-up on a fixed cylinder. *Coastal Engineering*.

Pringle, W. J., Burnett, Z., Sargsyan, K., Moghimi, S., and Myers, E. (2023). Efficient probabilistic prediction and uncertainty quantification of tropical cyclone-driven storm tides and inundation. *Artificial Intelligence for the Earth Systems*, pages 1–40.

Pringle, W. J., Wirasaet, D., Roberts, K. J., and Westerink, J. J. (2021). Global storm tide modeling with ADCIRC v55: unstructured mesh design and performance. *Geoscientific Model Development*, 14:1125–1145.

Proudman, J. (1953). *Dynamical Oceanography.* John Wiley and Sons, New York.

Proudman, J. (1955). The propagation of tide and surge in an estuary. In *Proceedings of the Royal Society of London*, volume A231, pages 8–24.

Rasp, S. and Lerch, S. (2018). Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11).

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., Karpatne, A., Hansen, G. J. A., Hanson, P. C., Watkins, W., Steinbach, M., and Kumar, V. (2019). Process-guided deep learning predictions of lake water temperature. *Water Resources Research.*

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat, M. (2019). Deep learning and process understanding for data-driven earth system science. *Nature.*

Resio, D. T., Irish, J. L., and Cialone, M. A. (2009). A surge response function approach to coastal hazard assessment: Part 1, basic concepts. *Natural Hazards*, 51:163–182.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

Saviz Naeini, S. and Snaiki, R. (2024). A novel hybrid machine learning model for rapid assessment of wave and storm surge responses over an extended coastal region. *Coastal Engineering*, 190.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data.*

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Smith, A., Lott, N., Houston, T., Shein, K., Crouch, J., and Enloe, J. (2020). U.s. billion-dollar weather and climate disasters 1980-2025. Technical report, NOAA National Centers for Environmental Information (NCEI).

Soliman, M., Morsy, M. M., and Radwan, H. G. (2025). Generalized methodology for two-dimensional flood depth prediction using ml-based models. *Hydrology.*

Sweet, W. V., Hamlington, B. D., Kopp, R. E., Weaver, C. P., Barnard, P. L., Bekaert, D., Brooks, W., Craghan, M., Dusek, G., Frederikse, T., Garner, G., Genz, A. S., Krasting, J. P., Larour, E., Marcy, D., Marra, J. J., Obeysekera, J., Osler, M., Pendleton, M., Roman, D., Schmied, L., Veatch, W., White, K. D., and Cuzak, C. (2022). Global and Regional Sea Level Rise Scenarios for the United States: Updated Mean Projections and Extreme Water Level Probabilities Along U.S. Coastlines. Technical report, NOAA Technical Report NOS 01.

Taghizadeh, M., Zandsalimi, Z., Shafiee-Jood, M., and Alemazkoor, N. (2025). Multi-fidelity graph neural networks for efficient and accurate flood hazard mapping. *Environmental Modelling & Software*, 193.

Tanaka, S., Bunya, S., Westerink, J. J., Dawson, C. N., and Luettich, R. A. (2011). Scalability of an Unstructured Grid Continuous Galerkin Based Hurricane Storm Surge Model. *Journal of Scientific Computing*, 46:329–358.

Thomas, A., Dietrich, J. C., Asher, T. G., Blanton, B. O., Cox, A. T., Dawson, C. N., Fleming, J. G., and Luettich, R. A. (2019). Influence of storm timing and forward speed on tide-surge interactions during Hurricane Matthew. *Ocean Modelling*, 137:1–19.

Tibshirani, R. (2010). The Elements of Statistical Learning: Data Mining, Inference, and Prediction by HASTIE, T., TIBSHIRANI, R., and FRIEDMAN, J. *Biometrics*.

Tiggeloven, T., Couasnon, A., van Straaten, C., Muis, S., and Ward, P. J. (2021). Exploring deep learning capabilities for surge predictions in coastal areas. *Scientific reports*, 11(1):17224.

Turan, C. K., Kinfu, Y. P., Samad, M. A., Farhadzadeh, A., and Ng, K. (2018). *Comparison of ADCIRC and SLOSH Model Simulations for Hurricanes Andrew and Irma near Miami, Florida*, pages 176–187. ASCE.

U.S. Army Corps of Engineers (2015). North atlantic coast comprehensive study: Resilient adaption to increasing risk. Technical report, U.S. Army Corps of Engineers.

US Department of Commerce, N. (2023). Summary of Tropical Cyclone Idalia's Impacts on the Wakefield CWA. `https://www.weather.gov/akq/Aug312023_Idalia`. [Retrieved 25 Feb 2026].

Van Cooten, S., Kelleher, K., Howard, K., Zhang, J., Gourley, J., Kain, J., Berry, K., Flamig, Z., Grams, H., Arthur, A., Langston, C., Kolar, R., Hong, Y., Dresback, K., Tromble, E., Vergara, H., Luettich, Richard, J., Blanton, B., Lander, H., and Spence, L. (2011). The CI-FLOW Project: A System for Total Water Level Prediction from the Summit to the Sea. *Bulletin of the American Meteorological Society*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

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

Wang, B., Liu, S., Wang, B., Wu, W., Wang, J., and Shen, D. (2021). Multi-step ahead short-term predictions of storm surge level using cnn and lstm network. *Acta Oceanologica Sinica*, 40:104–118.

Westerink, J. J. (2008). Flood Insurance Study: Southeastern Parishes, Louisiana, Intermediate Submission 2: Offshore Water Levels and Waves. Technical report, FEMA, U.S. Army Corps of Engineers.

Westerink, J. J., Luettich, R. A., and Scheffner, N. W. (1993). ADCIRC: An Advanced Three-Dimensional Circulation Model for Shelves Coasts and Estuaries, Report 3: Development of a Tidal Constituent Database for the Western North Atlantic and Gulf of Mexico. Technical Report DRP-92-6, U.S. Army Corps of Engineers, Vicksburg, MS.

Westerink, J. J., Luettich Jr, R. A., Feyen, J. C., Atkinson, J. H., Dawson, C. N., Roberts, H. J., Powell, M. D., Dunion, J. P., Kubatko, E. J., and Pourtaheri, H. (2008). A Basin to Channel Scale Unstructured Grid Hurricane Storm Surge Model Applied to Southern Louisiana. *Monthly Weather Review*, 136:833–864.

Woodruff, J. L. (2023). *Subgrid Corrections in Storm-Driven Coastal Flooding*. PhD thesis, North Carolina State University.

Xu, K., Tian, Y., Bin, L., Lai, C., and Yang, W. (2026). Rapid Prediction of Compound Flood Based on Hydrological-Hydrodynamic Model and Convolution Neural Network. *Water Resources Management*, 40(4).

Zang, H., Liu, L., Sun, L., Cheng, L., Wei, Z., and Sun, G. (2020). Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations. *Renewable Energy*.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks.

Zhang, K., Xiao, C., and Shen, J. (2008). Comparison of the CEST and SLOSH Models for Storm Surge Flooding. *Journal of Coastal Research*, 24:489–499.

Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2017). Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57.

APPENDICES

Chapter A

Acronyms

A summary of all acronyms from this Thesis is documented in Table A.1.

Table A.1: A summary of acronyms used in alphabetical order.

Term	**Definition**
Advanced Computing for Engineering and Science	ACES
ADvanced CIRCulation Model	ADCIRC
Albemarle–Pamlico Estuarine System	APES
Albemarle–Pamlico National Estuary Partnership	APNEP
Annual Exceedance Probability	AEP
Automated Tropical Cyclone Forecasting	ATCF
Artificial Intelligence	AI
Convolutional Neural Network	CNN
Coastal National Elevation Database	CoNED
Coordinate Reference System	CRS
Continuously Updated Digital Elevation Model	CUDEM
Digital Elevation Model	DEM
European Petroleum Survey Group (spatial reference code)	EPSG
Federal Emergency Management Agency	FEMA
Feed-Forward Neural Network	FNN
Gated Recurrent Unit	GRU
Generalized Wave Continuity Equation	GWCE
Geo-referenced Tagged Image File Format	GeoTIFF
Graphics Processing Unit	GPU
Hierarchical Data Format version 5	HDF5
High-Performance Computing	HPC
International Best Track Archive for Climate Stewardship	IBTRACS
Input/Output	I/O
Long Short-Term Memory network	LSTM

Table A.1: Continued

Term	**Definition**
Mean Absolute Error	MAE
Maximum Dissimilarity Algorithm	MDA
Maximum Envelope of Water	MEOW
Maximum of MEOWs	MOM
Maximum Potential Intensity	MPI
Machine Learning	ML
North American Datum of 1983	NAD83
North American Vertical Datum of 1988	NAVD88
National Centers for Environmental Information	NCEI
North Carolina State University	NC State
National Oceanic and Atmospheric Administration	NOAA
National Hurricane Center	NHC
National Weather Service	NWS
North Carolina	NC
Not a Number	NaN
Root-Mean-Square Error	RMSE
Simulating WAves Nearshore	SWAN
Synthetic Tropical CyclOne geneRation Model	STORM
Tropical Cyclone	TC
United States	U.S.
United States Army Corps of Engineers	USACE
United States Geological Survey	USGS
Unsigned 8-bit Integer	uint8
World Geodetic System 1984	WGS84

Chapter B

Full Model Architecture

Table B.1: Full model architecture summary showing each layer, output shape, parameter count, and connectivity.

Layer (type)	Output Shape	Param #	Connected to
temporal_stream (InputLayer)	(None, 25, 35)	0	–
conv1d (Conv1D)	(None, 25, 64)	6784	temporal_stream
conv1d_1 (Conv1D)	(None, 25, 96)	18528	conv1d
conv1d_2 (Conv1D)	(None, 25, 128)	36992	conv1d_1
conv1d_3 (Conv1D)	(None, 25, 256)	98560	conv1d_2
conv1d_4 (Conv1D)	(None, 25, 512)	393728	conv1d_3
conv1d_5 (Conv1D)	(None, 25, 1024)	1573888	conv1d_4
global_average_pooling1d (GAP)	(None, 1024)	0	conv1d_5
land_stream (InputLayer)	(None, 512, 512, 1)	0	–
reshape (Reshape)	(None, 512, 512)	0	land_stream
dense (Dense)	(None, 512, 512)	262656	reshape
global_average_pooling2d (GAP)	(None, 512)	0	dense
concatenate (Concatenate)	(None, 1536)	0	GAP_temporal, GAP_spatial
reshape_1 (Reshape)	(None, 2, 2, 384)	0	concatenate
conv2d_transpose_1	(None, 2, 2, 256)	884992	reshape_1
conv2d_transpose_2	(None, 4, 4, 256)	590080	conv2d_transpose_1
conv2d_transpose_3	(None, 4, 4, 192)	442560	conv2d_transpose_2
conv2d_transpose_4	(None, 8, 8, 192)	331968	conv2d_transpose_3

Table B.1: Continued

Layer (type)	**Output Shape**	**Param #**	**Connected to**
conv2d_transpose_5	(None, 8, 8, 128)	221312	conv2d_transpose_4
conv2d_transpose_6	(None, 16, 16, 128)	147584	conv2d_transpose_5
conv2d_transpose_7	(None, 16, 16, 96)	110688	conv2d_transpose_6
conv2d_transpose_8	(None, 16, 16, 80)	69200	conv2d_transpose_7
conv2d_transpose_9	(None, 32, 32, 80)	57680	conv2d_transpose_8
conv2d_transpose_10	(None, 32, 32, 72)	51912	conv2d_transpose_9
conv2d_transpose_11	(None, 32, 32, 64)	41536	conv2d_transpose_10
conv2d_transpose_12	(None, 64, 64, 64)	36928	conv2d_transpose_11
conv2d_transpose_13	(None, 64, 64, 48)	27696	conv2d_transpose_12
conv2d_transpose_14	(None, 64, 64, 32)	13856	conv2d_transpose_13
conv2d_transpose_15	(None, 128, 128, 32)	9248	conv2d_transpose_14
conv2d_transpose_16	(None, 128, 128, 28)	8092	conv2d_transpose_15
conv2d_transpose_17	(None, 128, 128, 24)	6072	conv2d_transpose_16
conv2d_transpose_18	(None, 256, 256, 24)	5208	conv2d_transpose_17
conv2d_transpose_19	(None, 256, 256, 20)	4340	conv2d_transpose_18
conv2d_transpose_20	(None, 256, 256, 16)	2896	conv2d_transpose_19
conv2d_transpose_21	(None, 512, 512, 16)	2320	conv2d_transpose_20
conv2d_transpose_22	(None, 512, 512, 12)	1740	conv2d_transpose_21
conv2d_transpose_23	(None, 512, 512, 10)	1090	conv2d_transpose_22
conv2d_transpose_24	(None, 512, 512, 9)	819	conv2d_transpose_23
conv2d_transpose_25	(None, 512, 512, 8)	656	conv2d_transpose_24
conv2d_transpose_26	(None, 512, 512, 7)	511	conv2d_transpose_25
conv2d_transpose	(None, 512, 512, 2)	20	land_stream

Table B.1: Continued

Layer (type)	**Output Shape**	**Param #**	**Connected to**
conv2d_transpose_27	(None, 512, 512, 6)	384	conv2d_transpose
concatenate_1 (Concatenate)	(None, 512, 512, 8)	0	conv2d_transpose_27, conv2d_transpose_26
conv2d_transpose_28	(None, 512, 512, 8)	584	concatenate_1
conv2d_transpose_29	(None, 512, 512, 6)	438	conv2d_transpose_28
conv2d_transpose_30	(None, 512, 512, 4)	220	conv2d_transpose_29
conv2d_transpose_31	(None, 512, 512, 2)	74	conv2d_transpose_30
img (Conv2DTranspose)	(None, 512, 512, 1)	19	conv2d_transpose_31