

Supplementary Information

S1. Choice of Machine Learning Model

These consist of an input layer, a given number of hidden layers, and an output layer. The input layer receives pre-processed data in a form specific to ML model types. The hidden layers consist of artificial neurons. These are fully connected, and through backpropagation update weights and biases between artificial neurons. This in turn affects the impact of feature inputs. Between each hidden layer we commonly have activation functions (Dubey et al., 2022), which allow the model to create non-linear relationships. Here, fully connected means that each neuron is connected to every neuron in the layers on either side of it. The output layer synthesises the work done by the hidden layers, and provides an output, with the form of this output dependent on model type.

S2. Sensitivity Testing

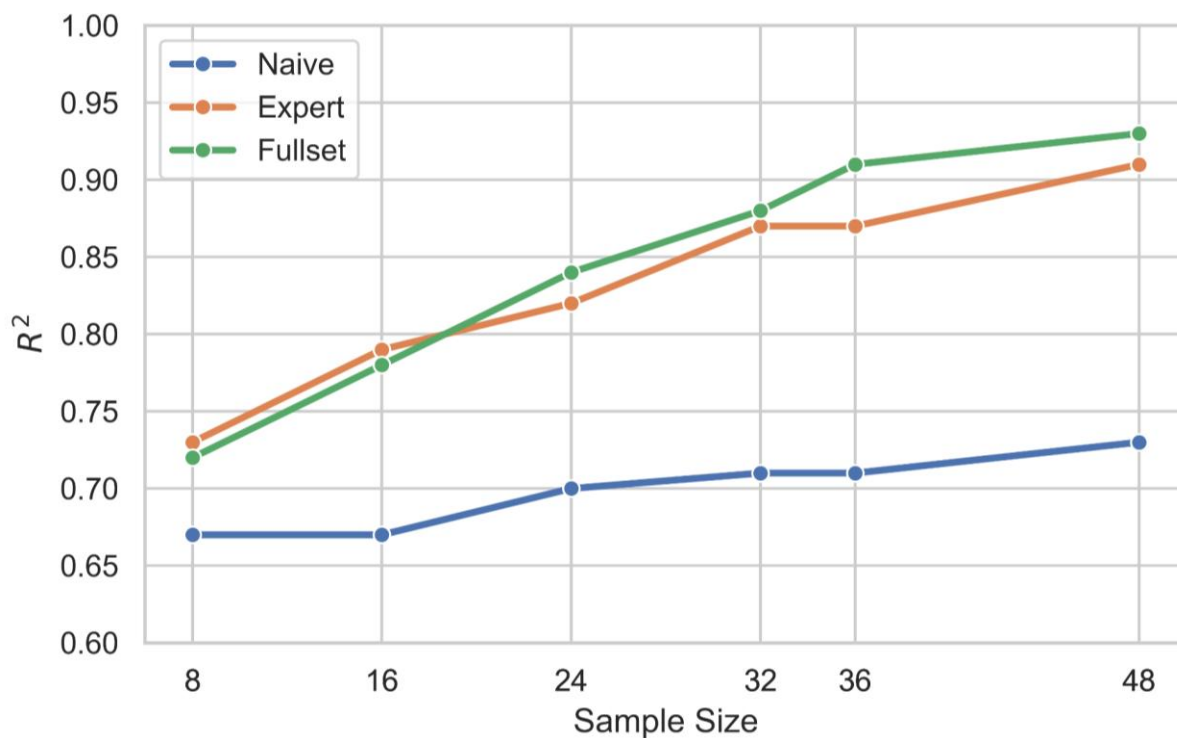


Figure S1 indicating storm sample size sensitivity tests. Note that the sample size of 36 was ultimately used for the training sample used in the paper. This was due to concerns that the similarity of the synthetic trackset used could lead to overfitting if larger sample sizes were utilised. R^2 scores displayed here are calculated using the test set described in the paper.

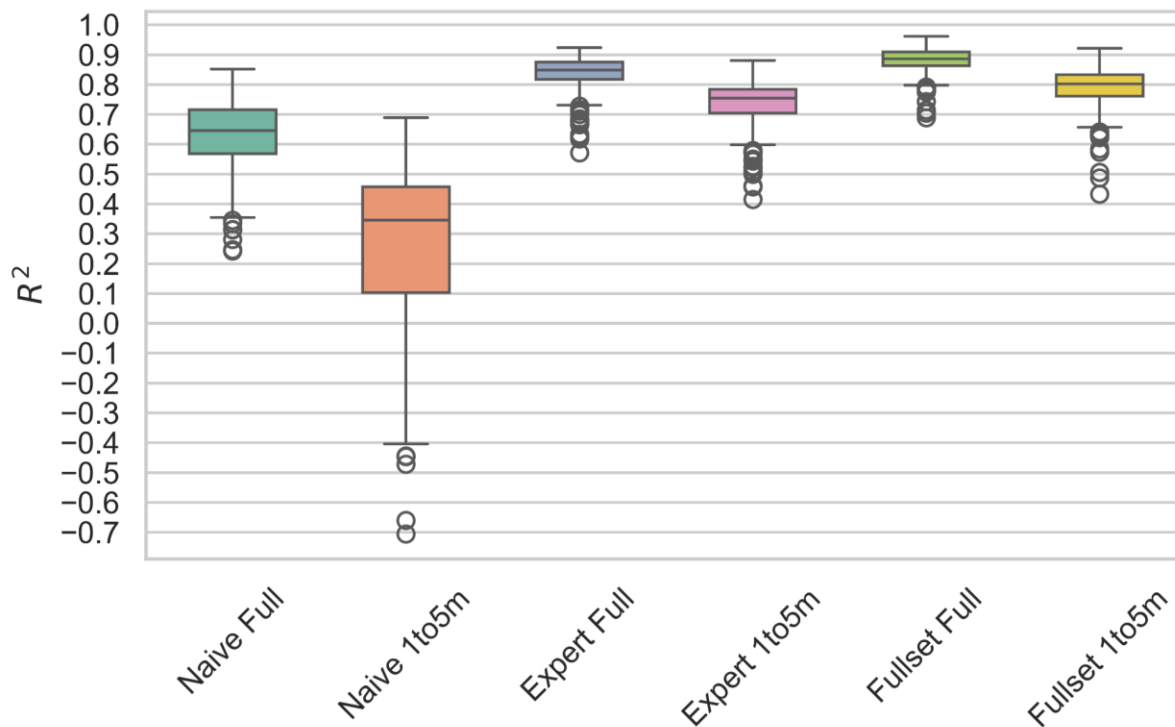


Figure S2 - illustrating model results across all synthetic events which were not utilised within the machine learning framework utilised in the paper. To clarify, these events were not used in the Train/Validation/Test data split of 40 storms, with these excluded to address concerns of overfitting due to the similarity of the synthetic test set.

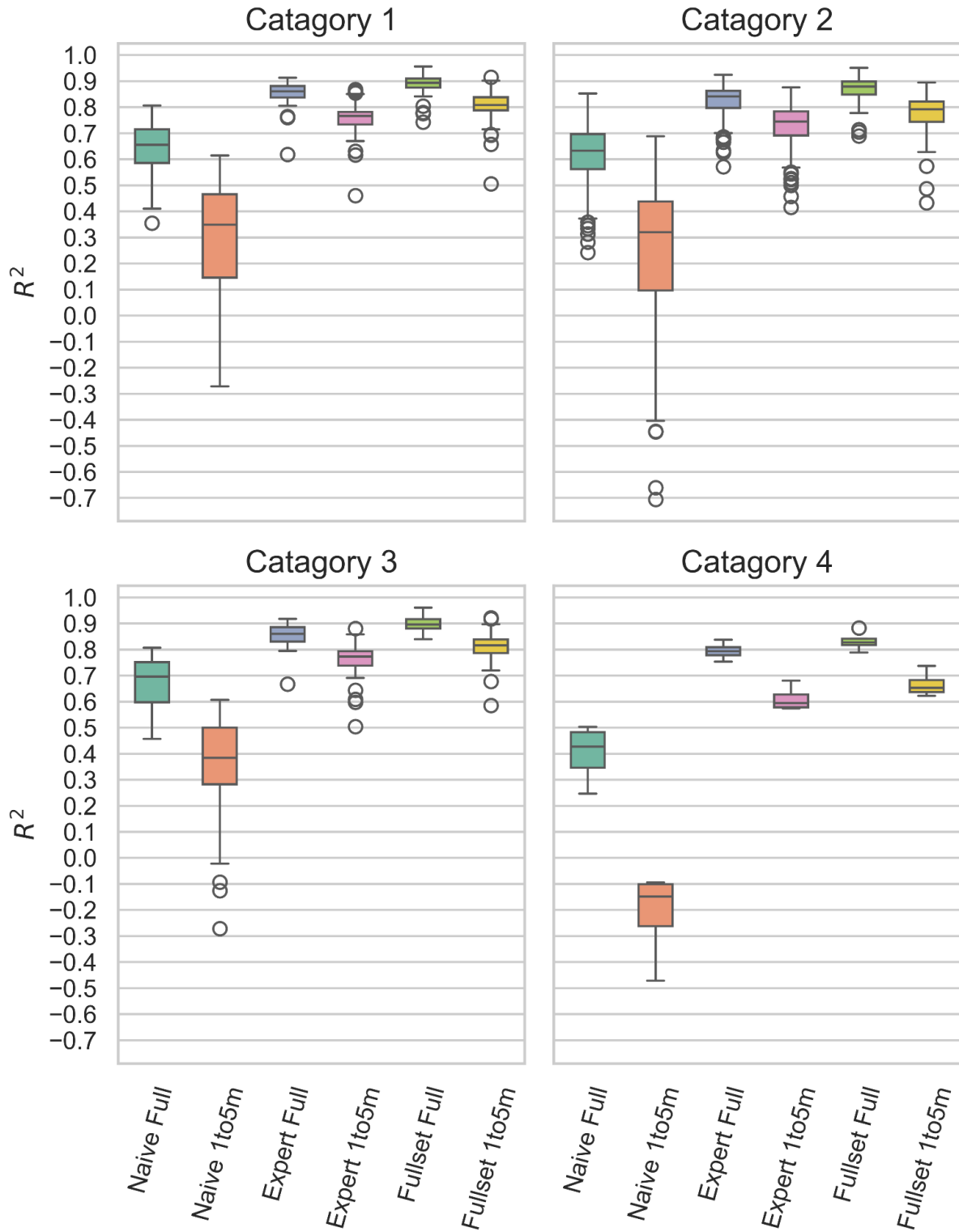


Figure S3 - plotting model results across all synthetic events which were not utilised within the machine learning framework utilised in the paper. The events are plotted by category. These events were not used in the Train/Validation/Test data split of 40 storms within the paper.

S3.1 Feature Engineering

Feature Long Name	Feature Code Name	Description
Fetch over shelf	bathy_fetch	The distance from the edge of the continental shelf that a given pixel is. This is calculated in pixels, with the pixels being of 100m resolution. This is then interpolated (nearest neighbour) onto ADCIRCs mesh.
Coast enclosure metric	coast_geom_transects	Indicates how enclosed a coastal region is. 0 is the most enclosed with 360 the least. It works by calculating 360 transects from each pixel. These are then traced to a maximum of 50 pixels. Where a transect does not encounter land +1 is added to the source pixel. The pixels used for this calculation are 1000m resolution - so the transects extend 50km. This is interpolated onto ADCIRCs mesh using nearest neighbour.
Distance to storm	dis_strm	Maps the distance from the eye of the storm (area of lowest pressure) to each node within the mesh. Distance is in kilometers.
Wind U vector (alt)	windx_alt	The zonal component of wind. This has been altered so that the zero axis is aligned with the path of the storm, as characterised by a straight line between its starting location and its landfall location. This improves direction agnosticity and better captures angle of incidence.
Depth	depth	The distance from ADCIRC mesh mean sea level (NAVD88) to the sea floor.
Charnock's wind stress	wind_stess	A measure of air ocean interaction, quantifying the force per unit area on the sea surface.
Distance from land	dis2land	The distance from a given pixel to land, calculated using a rasterised version of the ADCIRC mesh. Pixels are of 100m resolution.
Harris surge	harris	An empiric surge calculator (Harris, 1963). Gives a surge value at each node.
PRMSL	pressure	Pressure at mean sea level. Derived directly from ADCIRC forcing.
Fetch	fetch	The distance, based on wind direction, from any node to land.
Wind speed	wind_spd	Wind speed in meters per second, calculated from ADCIRC forcing wind u and v vectors.
PRMSL gradient sine	pressure_grad_sin	Provides the sine angle of pressure gradient magnitude. Indicates direction of pressure increase.
Hours to landfall	hrs2landfall	The hours till storm landfall. Positive before landfall, negative prior.
PRMSL gradient cosine	pressure_grad_cos	Provides the cosine angle of pressure gradient magnitude. Indicates direction of pressure increase.
PRMSL temporal gradient (6hrs)	pressure_slope_6hrs	Provides the gradient of pressure change over time, using rise over run.
Wind speed temporal gradient (6hrs)	wind_spd_slope_6hrs	Provides the gradient of wind speed change over time, using rise over run.

Wind speed gradient	wind_spd_grad_mag	The magnitude of wind speed gradient change, measuring the slope between one node and the next.
Wind cosine (alt)	wind_cos_alt	The cosine aspect of wind direction, derived from ADCIRC altered U and V vectors. Note that this alteration aligns the 0 axis with storm path, as defined as storm first location in the domain, and its landfall location.
Wind speed temporal gradient (3hrs)	wind_spd_slope_3hrs	Provides the gradient of wind speed change over time, using rise over run.
34knt wind boolean	knt_34_scalar	A boolean where 1 indicates that wind speeds of 34knts are exceeded and 0 indicates it is not.
Radial Basis Function (-6hr)	rbf_-6hr	A gaussian curve centered on a given point - in this case six hours after storm landfall.
Wind sine (alt)	wind_sin_alt	The sine aspect of wind direction, derived from ADCIRC, altered U and V vectors. Note that this alteration aligns the 0 axis with storm path, as defined as storm first location in the domain, and its landfall location.
Wind speed sq	wind_spd_sq	Wind speed squared.
Time Linear	time_linear	Time, measured as a fraction, from storm entry into the domain until storm end.
50knt wind boolean	knt50_scalar	A boolean where 1 indicates that wind speeds of 50knts are exceeded and 0 indicates it is not.
PRMSL gradient	pressure_grad_mag	The magnitude of pressure gradient change, measuring the slope between one node and the next.
64knt wind boolean	knt64_scalar	A boolean where 1 indicates that wind speeds of 64knts are exceeded and 0 indicates it is not.
Wind stress coefficient	wind_stress_cof	Dimensionless term which represents the efficiency of force transfer between the atmosphere and the ocean. The term has a limiter at 0.0025 above wind speeds of 30m/s. This feeds into Charnock's wind stress.
Storm track direction sine (alt)	tdir_sin_alt	The sine aspect of overall storm direction between timesteps. Note that the axis of measurement has been altered so that zero rests between first storm location and storm landfall.
PRMSL temporal gradient (3hrs)	pressure_slope_3hrs	Provides the gradient of pressure change over time, using rise over run.
Yearly sine	yearly_sin	A sine curve representing time over a year.
Inverse barometer	static_uplift	The impact of pressure on sea level, as defined by the inverse barometer effect.
Wind V vector (alt)	windy_alt	The meridional component of wind. This has been altered so that the zero axis is aligned with the path of the storm, as characterised by a straight line between its starting location and its landfall location. This improves direction agnosticity and better captures angle of incidence.
Yearly cosine	yearly_cosin	A cosine curve representing time over a year.
Storm track direction cosine (alt)	tdir_cosin_alt	The cosine aspect of overall storm direction between timesteps. Note that the axis of measurement has been altered so that zero

		rests between first storm location and storm landfall.
Radial Basis Function (0hr)	rbf_0hr	A gaussian curve centered on a given point - in this case storm landfall.
Slope transect	slope_transect	Uses the distance from each pixel to its closest land pixel to calculate slope using rise over run. Limited to the continental shelf.
Storm path	tc_path	This indicates storm path based on direction. It has 3 categories. 3 indicates within RMax, 2 indicates directly inline, 3 indicates proximal. In practice this creates a cone radiating from the Rmax, with 2 and 3 determined by a given angle.
Wind zone	wind_zone	Combines the Rmax and knt scalar variables to create categories of wind strength.
Wind speed temporal gradient (1hrs)	wind_spd_slope_1hrs	Provides the gradient of wind speed change over time, using rise over run.
Daily sine	daily_sin	A sine curve representing time over a day.
Radial Basis Function (6hr)	rbf_6hr	A gaussian curve centered on a given point - in this case 6hrs before storm landfall.
Wind speed cosine gradient	wind_spd_grad_cos	Provides the cosine angle of wind speed gradient magnitude. Indicates direction of wind speed increase.
Daily cosine	daily_cos	A cosine curve representing time over a day.
Storm Speed	t_speed	The overall storm speed between timesteps, calculated by distance covered over a given time. Unit is kmph.
Slope sine	slope_sin	Sine aspect of slope direction.
Radius of maximum wind boolean	rmax	Defines the radius of maximum winds, with 1 indicating it is within this radius.
Slope	slope_mag	Slope magnitude, defined by rise over run.
Slope cosine	slope_cos	Cosine aspect of slope direction.
Wind U vector	windx	The zonal component of wind angle of incidence.
Wind V vector	windy	The meridional component of wind.

Table S1 - describes the features created and used within this study. Code name refers to the name by which they are coded, and links to SHAP figures.

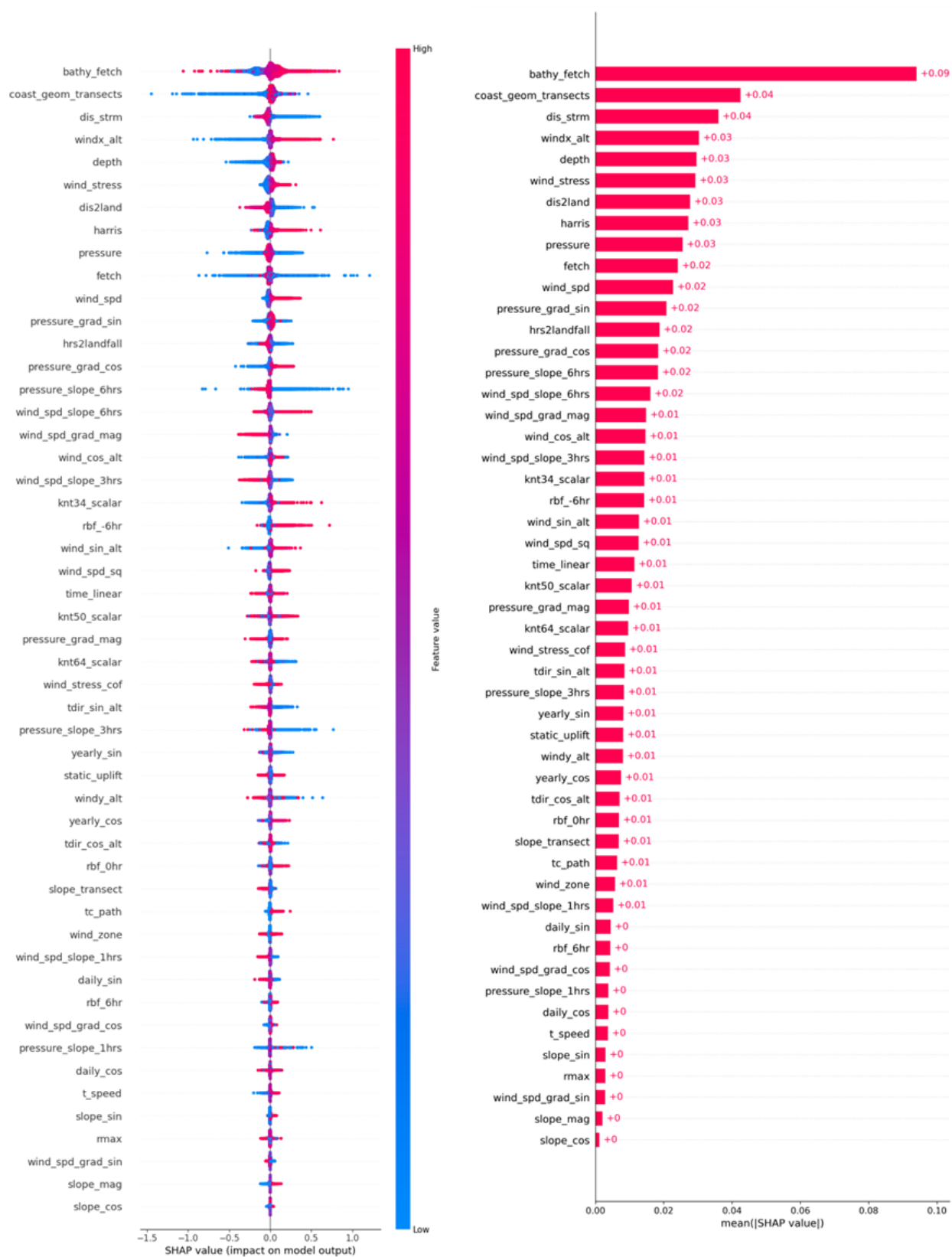


Figure S4 - illustrating the full range of SHAP values for the Fullset model across the full domain with coordinates excluded. See table 1 for details on each variable.

S3.2

In essence this is not a recurrent neural network, and so does not see any timesteps but the one being predicted – this is in contrast to a LSTM, a common recurrent neural network, which can be fed a time series of data, and then make a prediction for next step in the timeseries (Krichen and Mihoub, 2025). Nor is it a computer vision model, and so does not see surge as an image, like a CNN would, limiting its spatial context (O'shea and Nash, 2015). Despite this we can produce features, which while at a single point in time and space, can provide information on other times and other spaces, for example by using gradients.

S3.3

For context Yeo-Johnson normalisation is an algorithm which uses power transformations to make a given dataset's distribution more gaussian, in this context it is useful as it automates selection of the most suitable exponent. Following this normalisation, Standard Scaling is performed to transform features to have a mean of zero and a unit standard deviation. The purpose of this is to ensure features are on a comparable scale, which reduces training noise and speeds up convergence (Koukaras and Tjortjis, 2025).

S4.1 Neural Network Architecture and Training

Batch normalisation improves training speed and stability, while dropout helps to prevent overfitting by randomly deactivating neurons during training.

S4.2

ReLU works by setting negative inputs to 0, i.e. $f(x) = \max(0, x)$.

S4.3

Learning rate refers to the step size taken when adjusting the weights and biases in each training step. The batch size determines the number of records which are used in each training step. Training steps make up epochs, with one epoch being reached once all input training data has been fed through the model. The Adam optimiser (Kingma, 2014) is an algorithm which optimises convergence by making per-parameter learning rate adjustments during training.

S4.4

The benefit of Huber Loss is that it acts like MSE for error below the delta threshold, while acting like mean absolute error (MAE) (Equation 3) above it (Huber, 1992). In essence this combines the effectiveness of MSE while utilising MAEs robustness to outliers.

S4.5

Training data is used to update weights and biases to minimise loss. The model does not train on validation data instead uses it to prevent overfitting. For example, if training loss continues to improve but validation loss flatlines or increases, this indicates overfitting. Test data is withheld from the training process entirely and is used to generate the results described in this paper.

S5.1 Feature Interpretability

These employ game theory to determine the contribution of each feature to a single model prediction. The Shapley values are determined by removing not just the target feature, but also all possible combinations of the target feature and other features present in the model, thereby producing marginal contributions.

S5.2

Classically, feature permutation importance works by perturbing the value of a single feature and so breaking the relationship between the feature and the target, allowing calculation of impact on model performance. Our use of Shapley values here is more robust, as it considers feature interdependence.

S5.3

For context, PCA reduces dimensionality by creating an axis which captures maximum possible variance within the data. Subsequent axes (components) must be perpendicular to the first, ensuring they capture information not captured in prior axes. Additional components are added until the target threshold value of variance is captured (Hotelling, 1933). Cluster K-means works by reducing the within-cluster sum of squares – essentially the sum of squared distances between each point and its assigned cluster centroid (Arthur and Vassilvitskii, 2006).

S6. Feature Selection

For context, Spearman's Rank assesses the strength and direction of the relationship between the target and the predictor, but is limited in that it assumes the relationship is monotonic, however, it does not assume the relationship is linear. Mutual Information measures the amount of information gained from knowledge of a predictor in respect to the target and does not require the relationship to be linear.

S7. Computational Cost

In statistics the maxim of finding the best performing and simplest model, following the principle of Occam's Razor, is preferable. In ML frameworks this maxim arguably does hold the same power. Through the process of gradient descent, ML models are capable of ignoring irrelevant features, with these features therefore not detrimental to model performance in terms of error (although poorly chosen features may result in higher local minimas, and slower convergence). However, beyond error statistics there are other reasons for reducing the size of a feature set. Primary amongst these is reduced computational cost and so reduced runtime.

S8.1 Model Fit

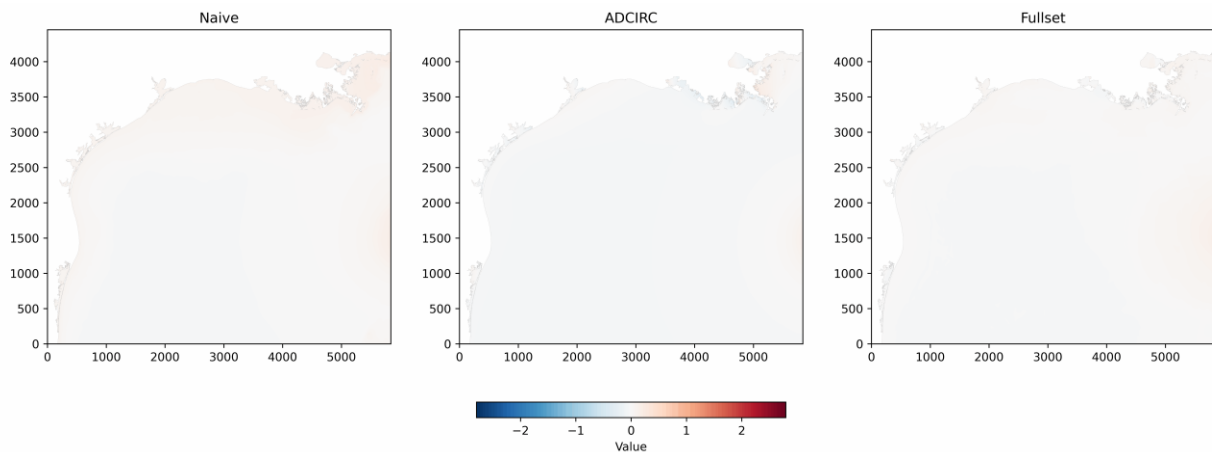


Figure S5 - shows a gif illustrating the spatiotemporal evolution of a single tropical cyclone event. ADCIRC illustrates the hydrodynamic model, which both the Naive and Fullset models are emulating. For further information refer to section 3.2.3 in the main text.

S8.2

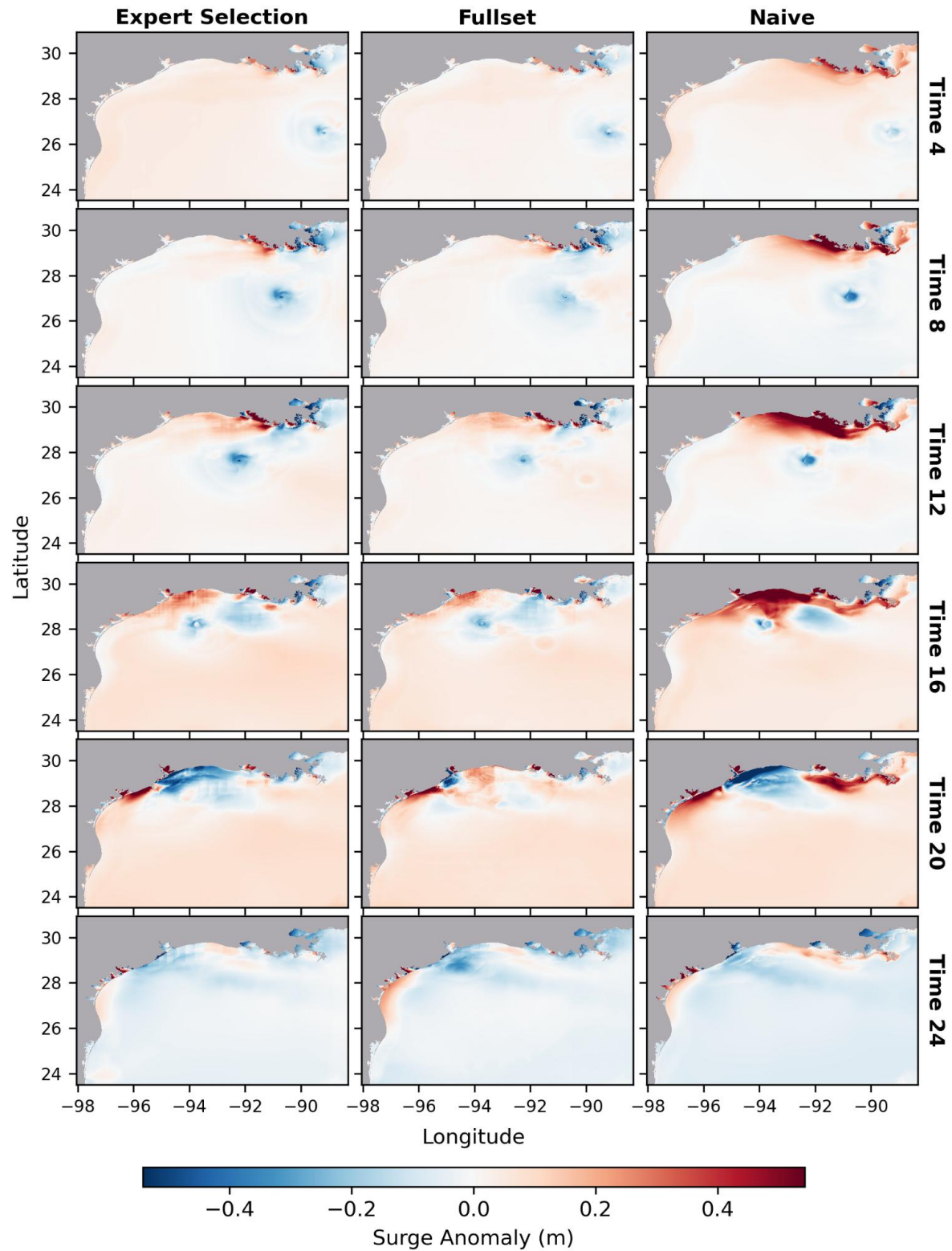


Figure S6 - illustrating spatiotemporal surge anomaly. Each column represents a model and each row a timestep, with time progressing from top to bottom. Anomaly is calculated as $ML - ADCIRC$.

S9. Historic and synthetic tracks

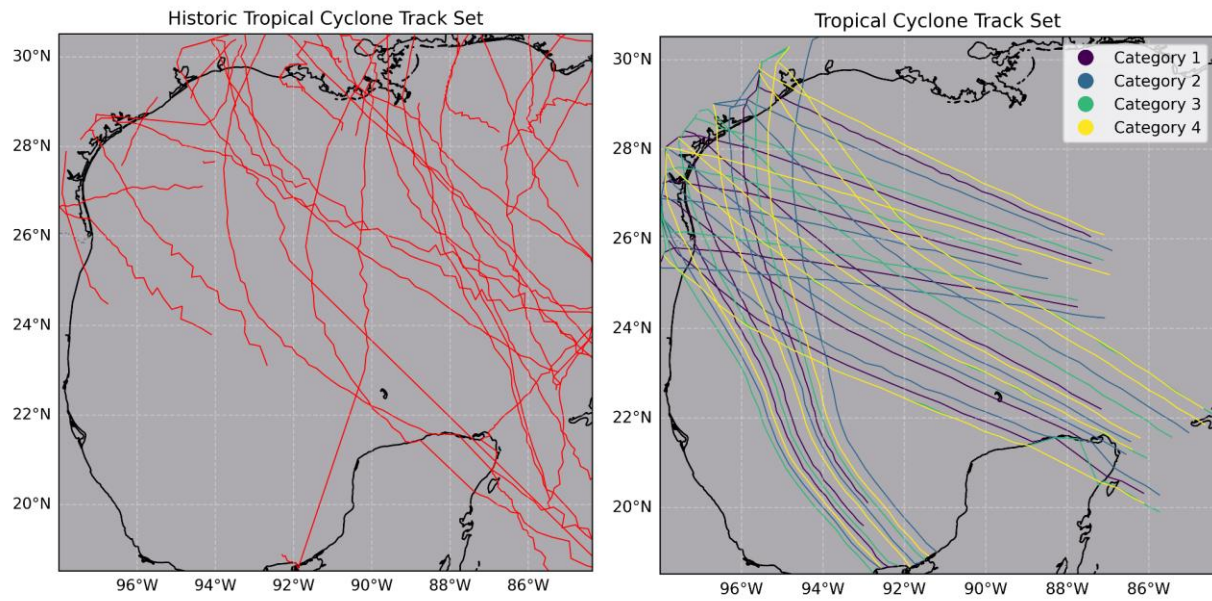


Figure S7 - illustrating that the available historic set is much more chaotic, and much more sparsely populated than the synthetic training set which this paper utilises.

S10. Preliminary Historic

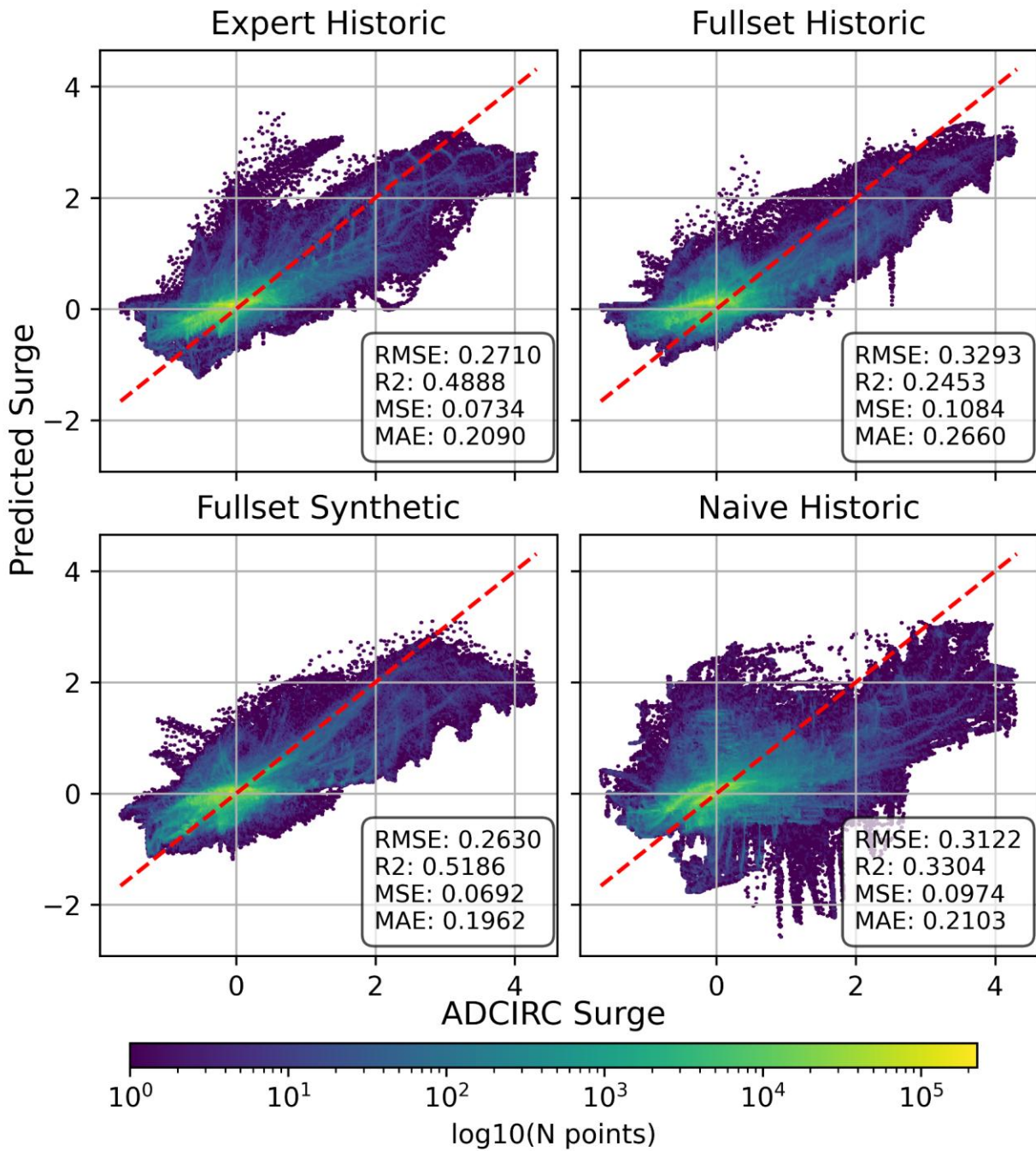


Figure S8 - illustrates results when applied to a subset of historic storms. “Historic” refers to models which were trained on historic hydrodynamic simulations (Kaiser et al., 2023). “Synthetic” refers to the model described in the main text. Results are shown predicting hindcasts for tropical cyclones Charley, Dolly, Hermine and Rita.

Reference List

- Arthur, D. and Vassilvitskii, S.: k-means++: The advantages of careful seeding, Stanford, 2006.
- Dubey, S. R., Singh, S. K., and Chaudhuri, B. B.: Activation functions in deep learning: A comprehensive survey and benchmark, *Neurocomputing*, 503, 92-108, <https://doi.org/10.1016/j.neucom.2022.06.111>, 2022.
- Harris, D. L.: Characteristics of the hurricane storm surge, 48, Weather Bureau.1963.
- Hotelling, H.: Analysis of a complex of statistical variables into principal components, *Journal of educational psychology*, 24, 417, 1933.
- Huber, P. J.: Robust estimation of a location parameter, in: *Breakthroughs in statistics: Methodology and distribution*, Springer, 492-518, 1992.
- Kingma, D. P.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, 2014.
- Koukaras, P. and Tjortjis, C.: Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices, *AI*, 6, 257, 2025.
- Krichen, M. and Mihoub, A.: Long Short-Term Memory Networks: A Comprehensive Survey, *AI*, 6, 215, 2025.
- O'Shea, K. and Nash, R.: *An Introduction to Convolutional Neural Networks*, ArXiv e-prints, 2015.