Supplemental A Detailed methodology

We use the U.S. National Water Model (NWM) (Salas et al., 2018) as the hydrological foundation for predicting the resulting surface water extent observable from Sentinel-2. In a sense we are proposing an extension to the NWM modeling chain to process atmospheric forcings variables to dynamic surface water extents. We hypothesize that the gridded NWM state values contain rich spatial information for mapping a resulting surface water extent, which includes rivers, lakes, floodplains, gullies, depressions, marshes, etcetera. Frame et al. (2021) post-processed the NWM for streamflow, showing that the NWM states contained more information than was being used for the conversion to streamflow within the NWM modeling chain. Here we use a similar method (Nair et al., 2022), but take advantage of the spatial distribution of the hydrologic states to inform the spatial distribution of surface water and flood characteristics.

NWM-CNN Model

Inputs to this network include two state variables from the U.S. National Water Model (NWM): soil moisture from the land surface component (NOAH-MP) and ponded depth from the terrain router. We also include three static inputs: a digital elevation model derived flow direction raster and flow accumulation raster, and a global surface water raster. NWM inputs include those from the NWM retrospective run (2000-2019; version 2.1) and NWM forecasts (2019-2021). We trained a fully convolutional encoder-decoder network (Ronneberger et al., 2015) to predict the fractional of surface water area (PSWA) (as measured by Sentinel-2) in a 250m x 250m pixel (a grid cell). Fully connected networks are useful for making pixel-wise predictions (Long et al., 2014).

A01 Target flood maps

We selected 3971 Sentinel-2 swaths from 780 flood events from 2015-2022 in the CONtiguous United States (CONUS). Initially, 26,108 Sentinel-2 candidate swaths were identified based on events in the Global Flood Monitoring catalogue (de Bruijn et al., 2019) that contained flood points in the NOAA Storm Events database ("A global database of historic and real-time flood events based on social media", 2023) (n=11,506); overlap with events identified by the Dartmouth Flood Observatory (Brakenridge, 2010), (n=4,259); and a proprietary catalogue of previously mapped floods by Floodbase (n=11,138). We

sub-selected from this large catalogue in several stages, with the overall strategy of showing the model swaths with lots of floodwater from diverse geographies. We excluded swaths where the number of cloud-free pixels were low and pixels which had low amounts of flooding. Note that having a target of fractional water labels, makes the modelling task a regression task, as opposed to binary classification. As this segmentation model is an ensemble, we use the per-pixel standard deviation of the ensemble as an uncertainty estimation. As this segmentation model is an ensemble, we use the per-pixel standard deviation of the ensemble as an uncertainty estimation. Since the segmentation model predicts on a range of [0,100], the standard deviation is bounded on [0,50]. We then exclude examples where the segmentation model had a high amount of uncertainty summed across the chip to avoid training on its mistakes.

Once these uncertain examples are excluded, we compute floodwater statistics from the floodmap labels (amount of water in never-flooded and flooded-before areas, and in urban and suburban pixels). Finally, we subselect chips that are overrepresented in a given geographic group (corresponding approximately to each S2 UTM code).

The motivation of selecting examples based on geographic groups, is to ensure that that frequent floods in the same area are not overrepresented in our dataset, based solely on their chance of being observed. An alternative approach here could be to weight the frequency of examples shown to the model during training, but this adds complexity when evaluating the model, when weights also need to be applied. After this subselection process, we are left with 3971 chips across 780 flood events.

A02 Dynamic Inputs

For each satellite observation, we extract two corresponding hydrologic state variables from the U.S. National Water Model (NWM) (Salas et al., 2018). First, a 1km resolution volumetric soil moisture is defined as a ratio of water volume to soil volume and is calculated from the NWM land surface component which uses the Community Noah Land Surface Model with Multi-Parameterization options (NOAH-MP). The soil moisture value at each grid cell includes interactions with snow, vegetation and surface energy fluxes (Yang et al., 2011). This process is well described elsewhere, but for the purposes of understanding the role on our surface water mode, it can be thought of generally as a function of the previous soil moisture (state value) at that grid cell, land sur-

face parameters and atmospheric forcings:

$$SMC(t) = f(SMC(t-1), u(t), \theta) \tag{A1}$$

where SMC is the soil moisture content state value at a particular grid cell, u is the atmospheric forcing at that grid cell and θ are the land surface parameters of the grid cell. These soil moisture state values provide spatial and temporal context of the precipitation intensity and hydrologic response only where the rainfall hits the ground. There is some influence laterally from the coupling of the land surface model and the terrain router, but in general the soil moisture dynamics are informative on the local surface water response directly from precipitation.

The second hydrologic state we use is a 250m resolution terrain router, representing a depth of water ponded on the surface. The terrain router uses the diffusive/kinimatic wave equation to route overland flow across the land surface based on a digital elevation model (DEM). The ponded water depth at a grid cell is a function of not only the previous value of the grid cell, but includes input from, and output to, neighboring cells. In general the process can be simplified as:

$$RT_{i,j}(t) = f(\hat{RT}(t-1), u_{i,j}(t), \theta_{i,j}) \tag{A2}$$

where $\hat{RT}$ is the vector of terrain router states from neigboring grid cells. Here it is important to note that the dynamics of flooding are more complex than described with the terrain router. It is possible for a lot of routing to occur on a grid cell without flooding occurring, due to factors including evaporation, soil transmission, and human-made flood infrastructure. Since the terrain router represents the spatial movement of surface water, this NWM state is critical for informing a surface water response from precipitation that may occur outside the area of interest. This type of layer is critical for informing the model about temporo-spatial (e.g., upstream to downstream) hydrologic dynamics, as called for by Dasgupta et al. (2022b).

NWM dynamic data sources are available hourly. In order to capture temporal information about the flood dynamics, we aggregate the previous 72 hours of terrain routing and soil moisture, and provide these as inputs to the model.

A03 Static Inputs

For each satellite observation we also extracted four static layers, to provide the model with context about locations that may be more or less prone to flood. We use two **DEM-Derived** layers from Hydrosheds (Lehner et al., 2008): flow direction ($FDIR$), describing the direction of water flow as one of eight equidistant radians, and a flow accumulation layer ($FACC$), describing the number of upstream grid cells. Both Hydrosheds layers are provided at 15 arcseconds resolution. We use a 30m resolution **Global Surface Water** layer derived from the the Joint Research Council Global Surface Water data v1.1 (GSW) (Pekel et al., 2016b), which describes the frequency of occurrence of water on a grid cell over 32 years and is used to represent permanent water. The layer can be considered as a surface water prior. A sizeable portion of our training examples contain irrigated water in **Agriculture** lands during rainy seasons. Since the dynamic hydrologic state variables do not contain context about human infrastructure that would irrigate this water, we give the model context about heavily irrigated agricultural lands based on a transformation of the the annual, 30m resolution US Department of Agriculture (USDA) Cropland product (USDA National Agricultural Statistics Service, 2023), available 2008-2022. This layer provides the model with context about whether or not a pixel belongs to potentially heavily irrigated agriculture.

A04 Preprocessing

All data required for training is resampled to a 250m grid. Layers that have very heavy tailed distributions are log-transformed (terrain router, flow accumulation), and all layers are clipped to either a max-theoretical value or the max observed in the data (excluding test examples), prior to linearly rescaling to [0, 1].

A05 Intuition for using Convolution Neural Network as a surface water model

Our model, like all other deep learning models is defined by its *architecture*, the series of *layers*, each consisting of several *units*, that together, define a non-linear function that maps input data to an output prediction. As an input, the model ingests NWM dynamic states, along with static layers derived from a DEM. As an output, the model predicts the spatial distribution of surface water. In between are layers referred to as 'hid-

den' layers. In these units, a latent representation of the input data is learned. As the model predict surface water distribution from hydrologic conditions effectively, it suggests that the model has learned some form of physical hydrodynamic process in these latent layers based on the data, as opposed to being explicitly defined as they would be in a set of physics-based hydrodynamic equations. Allowing the model to learn the hydrodynamic process implicitly from the data allows for more flexibility and complexity in the relationship between input states and output surface water than what would be possible in a set of rigid physics based equations.

To motivate the use of a convolutional neural network (CNN) consider the representation of a non-convolutional network. Aside from the input layer, the output of a given unit at layer l, i.e. h^l, is computed by taking a dot product between a vector of the outputs of the previous layer, $\mathbf{h}^{l-1}$, and an vector of weights, $\mathbf{w}$, then adding a constant c^l, and finally applying a non-linear activation function g. This relationship is described in Equation A3 as follows:

$$h^l(\mathbf{h}^{(l-1)}) = g(\mathbf{w}^{\mathrm{T}}\mathbf{h}^{(l-1)} + c^l). \tag{A3}$$

After computing the output for each unit in a layer, they are then used to compute the outputs of the succeeding layer, until the final output is reached.

CNNs are also based on the concept of interconnected layers, but also include learnable features that convolve across the input feature map (in our case a 2D image):

$$F_{ij} = \sum_m \sum_n x_{(i+m)(j+n)} K_{mn} \tag{A4}$$

where F_{ij} is the value of the feature map at position (i, j), x is the input image/feature map, K is the kernel/filter and m, nm, n are the dimensions of the kernel. The convelution constricts the design of the network from the completely connected network, making CNNs more efficient, and as a result, more widely in remote sensing, and other computer vision tasks (Goodfellow et al., 2016). The spatial application for 2D images is the key architecture component that make the CNN a suitable architecture for surface water estimation.

The first difference relevant for surface water mapping is that in a CNN, groups of units in a given layer share weights, such that a convolutional operation can be used to calculate the dot product in Equation A3. These layers are referred to as *convolutional*

layers, and they significantly reduce the number of parameters that need to be learned in the network. By reducing the number of parameters in a given layer, we in turn can add more layers to the network, which allows for more a more complex relationship to be learned between inputs and outputs. Weight sharing also results in *translational invariance* property, which means that if a pattern between inputs is observed in the top left corner of an image that indicates surface water, the model will be able to identify it regardless of its relative position in the image (ie. surface water flooding along a river can be identified based on properties of the neighbouring slope and input features, even if it's in the top, middle, or bottom of the input image).

The second key difference is that the input image, instead of being flattened into a vector for the non-convolutional network, retains its shape in the CNN. This, allows the spatial relationship of neighbouring pixels to be retained for the convolution operation, which is extremely important in modelling physical processes. The convolution operation explicitly ties a pixel and its neighbourhood to the output value through the learned weight.

In our case the input images, which include both the dynamic and static images, are projected into the same coordinate system and resolution. The model, however, does not consider geospatial location of any pixel, only relative spatial coordinates of each pixel is considered by the model (Long et al., 2014). The convolution aspect of the neural network is done spatially, not temporally. Between each layer of the network, the convolution of the input layers is done roughly as:

$$x_{ij}^{l+1} = f_{ks}(\{x_{si+\delta i, sj+\delta j}^{l}\}, 0 < \delta i, \delta j < k) \tag{A5}$$

where x are the static and dynamic input features ($SMC, RT, FDIR, FACC\&GSW$), which together represent the potential flood response and floodable landscape, l is the network layer, k is the kernel size, s is the sampling factor and f_{ks} is the type of layer. The model's prediction of flooding is then a function of the last layer of the network, the "output" layer o.

$$y = f(x^{o}) \tag{A6}$$

The resulting layer, y, is the flood map (FSWApp) with the same shape resolution of each of the inputs.

A06 Training the UNet

We split 3971 examples into three cross-validation folds and a single test set. Splits were made stratified by flood water and biome, and assessed for temporal balance afterwards. Additionally, separate splits contain non-overlapping geometries. Figure A1 summarizes example counts across these different geographies. Notably, performance across CONUS is evaluated in 5 Geographic Zones of interest. Four "Coastal zones" which track 10 miles (16km) inland from the corresponding coasts: Pacific, Gulf, Atlantic South, Atlantic North; and one inland zone. The Atlantic zones are divided along the Virginia, North Carolina Border, and are separated due to the different types of flooding we expect to see in these areas.

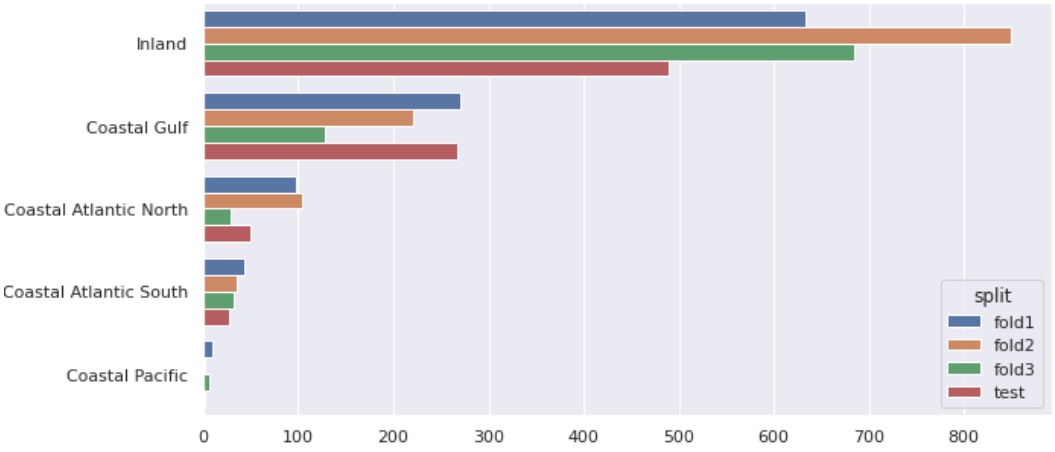

Figure A1. Example counts across dataset splits in different geographies.

The CNN model architecture is based on a U-Net architecture with an EfficientNet-B1 backbone. The encoder is pretrained on ImageNet in a semi-supervised approach (Noisy Student) (Xie et al., 2020). The entire model is fine tuned on our dataset using the Adam optimizer with a learning rate of 1e-4, and a learning rate scheduler that halves the learning rate when the validation loss does not improve for four epochs. We use Mean Squared Error (MSE) as the loss function. Pixels belonging to one of the 5 agricultural bins (corn, soy, rice, cranberry, aquaculture) have a loss down-weighted by a factor of 10.

A1 NWM-HAND model

We include the NWM-HAND model as a benchmark. This model is discussed at length in (Liu et al., 2018; Zheng et al., 2018; Aristizabal et al., 2023), so we will only cover the details that are unique to our implementation.

The HAND procedure run for the HUC6 scale. We download public data from the continental flood inundation mapping (CFIM) framework (Liu et al., 2020).

Once the data is verified or downloaded, the model retrieves links to the HAND TIFF, catchment ID TIFF, and the hydraulic properties table from GCS. The model compiles a list of stream reach IDs within the specified HUC6 domain, which are used for correlating the HAND model data with specific areas within the watershed.

The model is forced with National Water Model (NWM) streamflow estimates, setting a start date and optionally an end date for the analysis. It can generate flood maps for each hour of each day within the date range or produce daily maximum extent maps. For each date and hour within the specified range, the model processes NWM netCDF files to extract discharge estimates, labeled as streamflow, for each catchment, identified by a feature id representing the COMID of the National Hydrographic Dataset (NHD).

The hydraulic properties table is used to convert the NWM discharge estimates into river stage (water depth) for each stream reach. A spatial array representing river stages for each catchment is created, aligning the river stage data with the physical locations of each catchment. Flood depth and extent maps are generated using this spatial stage array and HAND elevation data.

For our comparison we convert the NWM-HAND binary (true/false) extent maps to PSWA. The HAND output is at $10m^2$ spatial resolution, the same as the input static HAND map. We re-sample these based on the average pixel value, meaning that the resulting $250m^2$ pixel value represents the percent of true $10m^2$ pixels within the $250m^2$ area.

A2 Pixel-Wise Performance

Pixel-wise performance across CONUS regions is evaluated by taking the Root Mean Squared Error (RMSE) between S2-observed and predicted fractional water predictions among the held-out test set. As the Fractional Water predictions are bounded on [0,100],

RMSE is also bounded on this range, where lower values indicate a better model. RMSE among completely held-out geographies is provided in Table A1.

To fairly evaluate the model, we further exclude the following pixels from evaluation. Dry-land predictions ("True Negatives"). As most pixels are of dry land, including these would artificially deflate the RMSE. We also exclude known heavily irrigated agriculture crops (Aquaculture, Corn, Cranberries, Soy, Rice). Including these would otherwise artificially increase the RMSE on pixels that the model cannot be expected to predict water on, given the lack of forcing data.

Geographic Subgroup	**Test Set RMSE**
Inland	3.68
Coastal Gulf	7.59
Coastal Atlantic North	2.29
Coastal Atlantic South	3.67
Coastal Pacific	5.67
Across Regions	4.58 +/- 2.07

Table A1. RMSE among non-Permanent Water pixels (GSW $\in$ [0, 0.30]) across geographic groups

A3 Proprietary Sentinel-2 Satellite Segmentation Model

Our proprietary Sentinel-2 water segmentation model was designed to produce binary water labels at 10m resolution given a multispectral Sentinel-2 observation. To demonstrate the effectiveness of this model, a test set of 892 hand-labelled examples were produced from held-out geographies. A summary of the model's performance on this held-out test set is provided in Table A2.

Pixel Subset	**IoU (mean +/- std)**
Never Flooded Pixels (GSW ∈ [0])	76.3% +/- 3.3
Flooded Before Pixels (GSW ∈ (0, 0.30])	88.6% +/- 4.2

Table A2. Intersection over Union of Proprietary Sentinel-2 Segmentation Model

A4 Proprietary Sentinel-1 Satellite Segmentation Model

Our proprietary Sentinel-1 water segmentation model was designed to produce binary water labels at 10m resolution given a Sentinel-1 observation. To demonstrate the effectiveness of this model, a test set of 1573 hand-labelled examples were produced from held-out geographies. A summary of the model's performance on this held-out test set is provided in Table A2.

Pixel Subset	**IoU (mean +/- std)**
Never Flooded Pixels (GSW ∈ [0])	64.8% +/- 15.2
Flooded Before Pixels (GSW ∈ (0, 0.30])	94.2% +/- 4.3

Table A3. Intersection over Union of Proprietary Sentinel-1 Segmentation Model

Supplemental B Binary metrics

Here we'll explain in detail why binary metrics aren't suitable for our model.

B1 Sensitivity to threshold values

In the main analysis we use the trivial thresholds of PSWApp > 0 for calculating binary statistics. This provides a general understanding of how NWM-CNN behaves under a threshold value for determining "flooded" and "not flooded", however, this determination should be done on a case-by-case basis, as some areas with relatively low PSWApp values may be considered flooded, while other areas may have a high PSWApp value before the characterization of flooded is appropriate.

We ran the metrics on a range of thresholds, 1 through 99 for the NWM-CNN values. he figures show that with low model threshold values, our model does quite well. But with higher threshold values, our model does poorly. This is directly related to the distribution of pixel values.

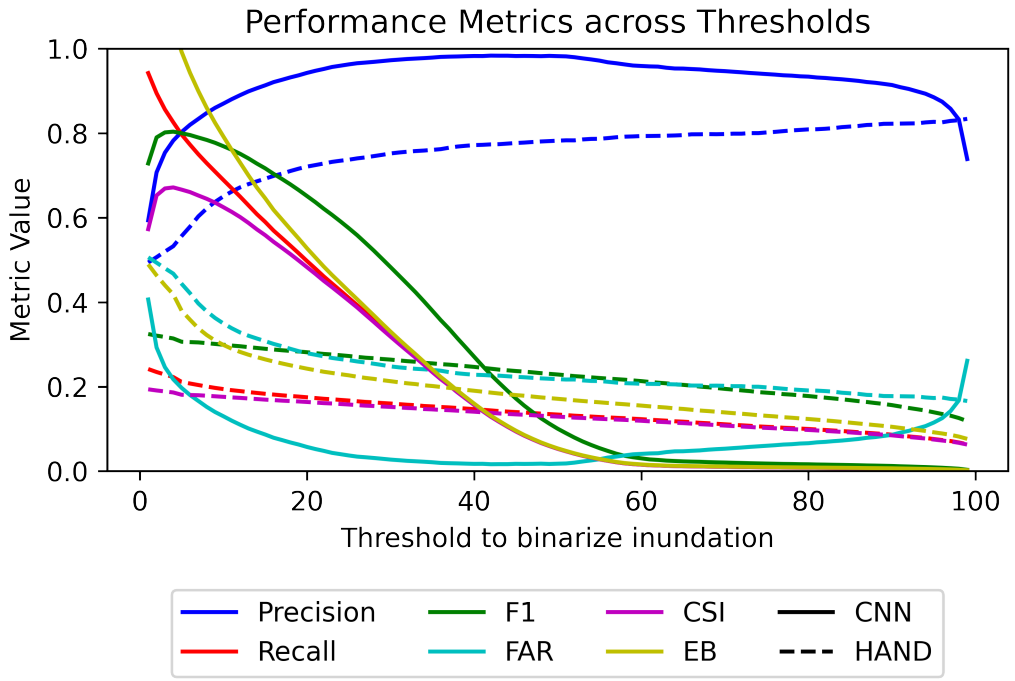

Figure B1. Binary metrics sensitivity to arbitrary threshold values.