EvaNet: Elevation-Guided Flood Extent Mapping on Earth Imagery

Mirza Tanzim Sami[2] , **Da Yan**[1] , **Saugat Adhikari**[1] , **Lyuheng Yuan**[1]
Jiao Han[1] , **Zhe Jiang**[3] , **Jalal Khalil**[4] and **Yang Zhou**[5]

[1]Indiana University Bloomington
[2]University of Alabama at Birmingham
[3]University of Florida
[4]St. Cloud State University
[5]Auburn University

mtsami@uab.edu, {yanda,adhiksa,lyyuan,jiaohan}@iu.edu, zhe.jiang@ufl.edu,
jalal.khalil@stcloudstate.edu, yangzhou@auburn.edu

Abstract

Accurate and timely mapping of flood extent from high-resolution satellite imagery plays a crucial role in disaster management such as damage assessment and relief activities. However, current state-of-the-art solutions are based on U-Net, which cannot segment the flood pixels accurately due to the ambiguous pixels (e.g., tree canopies, clouds) that prevent a direct judgement from only the spectral features. Thanks to the digital elevation model (DEM) data readily available from sources such as United States Geological Survey (USGS), this work explores the use of an elevation map to improve flood extent mapping. We propose, EvaNet, an elevation-guided segmentation model based on the encoder-decoder architecture with two novel techniques: (1) a loss function encoding the physical law of gravity that *if a location is flooded (resp. dry), then its adjacent locations with a lower (resp. higher) elevation must also be flooded (resp. dry)*; (2) a new (de)convolution operation that integrates the elevation map by a location-sensitive gating mechanism to regulate how much spectral features flow through adjacent layers. Extensive experiments show that EvaNet significantly outperforms the U-Net baselines, and works as a perfect drop-in replacement for U-Net in existing solutions to flood extent mapping. EvaNet is open-sourced at https://github.com/MTSami/EvaNet.

1 Introduction

Climate change is drastically increasing the intensity and occurrence of floods [Matgen *et al.*, 2020]. In just the last two decades, flooding has negatively impacted over 2.3 billion people, often disproportionately affecting vulnerable communities [Wahlstrom *et al.*, 2015]. Therefore, accurate and timely mapping of flood extent can be crucial for effectively planning rescue and rehabilitation efforts [Oddo and Bolten, 2019]. However, sending field crews on site for inspection is prohibitively slow, while obtaining data from a limited number of groundwater sensors or UAVs cannot provide a global
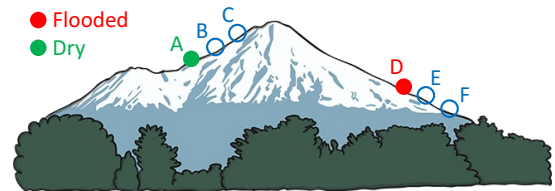

Figure 1: Physical law of gravity: if pixel A is dry, then its higher adjacent pixel B must be dry; while if pixel D is flooded, then its lower adjacent pixel E must be flooded.

coverage. Therefore, we target the accurate and timely mapping of flood extent from high-resolution satellite imagery that provides a global coverage.

There has been a lot of research on using remote-sensing satellite imagery for flood extent mapping. Two commonly used satellite types are radar (active sensor) and optical (passive sensor). For example, Sentinel-1 [Copernicus, 2014] and Sentinel-2 [Copernicus, 2015] are radar and optical satellites, respectively, that produce high resolution earth imagery at high revisit times [Matgen *et al.*, 2020]. Radar is more resilient to noises like clouds, but traditional approach to generating flood maps from radar imagery relies on semi-automated tools, [Rudner *et al.*, 2019] or thresholding techniques [Fraccaro *et al.*, 2022] that are slow and lack generalizability, often requiring expert intervention for parameter tuning. In this paper, we focus on the optical satellite imagery, since radar data such as that from Sentinel-1 [Copernicus, 2014] can also be converted into RGB channels [Fraccaro *et al.*, 2022]. Moreover, both [Konapala *et al.*, 2021] and [Bonafilia *et al.*, 2020] found that using Sentinel-2 data only yields the best segmentation performance for deep learning models, as compared with when Sentinel-1 data is involved.

Thanks to the availability of digital elevation model (DEM) data from sources such as USGS, existing deep learning approaches use the elevation map to improve segmentation performance by treating the map simply as an additional input channel of U-Net [Ronneberger *et al.*, 2015] just like a regular spectral channel [Beni and Gebrehiwot, 2021; Konapala *et al.*, 2021; Fraccaro *et al.*, 2022]. The physical law of gravity inherent to elevations as illustrated in Figure 1 (see its caption) is not encoded into the model design.

Encoding physical law in model design can help better re-

cover the true states ("flooded" or "dry") of noisy pixels covered by clouds and tree canopies, which are often ambiguous just based on spectral features, and it has shown success in graphical models for flood mapping [Jiang and Sainju, 2019; Sainju *et al.*, 2022; Sainju *et al.*, 2020; Xie *et al.*, 2018]. We propose to encode the physics-aware constraints into the more powerful deep learning models to achieve better performance.

Formally, given earth imagery with spectral features (e.g., RGB values) from optical satellites on a terrain surface (defined by an elevation map over a 2D grid), our goal is to conduct image segmentation so that each pixel (i.e., location) is classified as either "flooded" or "dry". In this paper, we propose an Ele̲vation-guided Ne̲twork (EvaNet) for flood extent mapping on earth imagery, which is physics-guided (c.f. Figure 1). Our main contributions are as follows:

- While a conventional loss function for image segmentation (e.g., pixel-wise cross-entropy loss $\mathcal{L}_{CE}$) is only computed from (i) ground-truth pixel labels and (ii) their predictions, we propose an elevation-guided regularization term $\mathcal{L}_{eva}$ to penalize adjacent pixel-pairs that violate the physical law of gravity. Interestingly, we find that $\mathcal{L}_{eva}$ alone can serve as a great loss function, not just as a regularization term.

- We extensively explored various approaches to integrating the elevation map as an input to the convolution operations on the encoder and decoder sides of U-Net, and find that using an independent convolution branch over the elevation map (called the elevation path) to regulate the flow of spectral features by the gated linear unit (GLU) [Dauphin *et al.*, 2017] gives the best performance by striking a balance between model simplicity (to prevent overfitting) and model expressiveness.

- Extensive experiments verify that our both techniques above are effective, and that when being combined, achieve the best performance beating U-Net and other non-deep-learning methods by a large margin. Moreover, we show that EvaNet is an effective drop-in replacement for U-Net to improve deep learning, flood mapping methods that originally rely on U-Net as a basic component.

2 Related Work

Image Segmentation. Flood extent mapping is a semantic segmentation task, which classifies each pixel in an image into a semantic class (flood or dry in our problem). Besides semantic segmentation, there are other segmentation tasks such as instance segmentation [He *et al.*, 2017] which finds the object pixels in the bounding box of an object detected by an object detection model, and panoptic segmentation [Kirillov *et al.*, 2019] which combines semantic segmentation and instance segmentation.

Flood Extent Mapping by Deep Learning. A number of deep learning models for flood extent mapping have been explored by the remote sensing community recently. For example, [Beni and Gebrehiwot, 2021] first applies FCN [Long *et al.*, 2015a] on only the RGB data to obtain an initial flood map (Stage 1); then the work uses water level data from USGS combined with the DEM data to detect water pixels under tree canopy, by using a method called region growing (Stage 2); the result from Stage 2 is then used to improve the FCN-based flood extent for vegetated areas (Stage 3). We remark that elevation data was not used in Stage 1, and our EvaNet can serve as a drop-in replacement of FCN to improve the performance of Stage 1.

As another example, [Konapala *et al.*, 2021] uses spectral bands SWIR2, NIR and red values from Sentinel-2 to compute new indices as U-Net input, and DEM data is treated simply as an additional channel. Also, [Mateo-Garcia *et al.*, 2021] focuses on the application of onboard segmentation with small, nano satellites to reduce revisit time to disaster areas; they input 13-band Sentinel-2 image into U-Net for segmentation without using DEM data. While these works from the remote sensing community include new input modes that can extend our work, our EvaNet targets the more fundamental problem of effectively integrating DEM data with spectral features guided by the physical law of gravity, and can serve as a drop-in replacement of their naïve U-Net or FCN models to improve performance.

Non-Deep-Learning Methods. Hidden Markov tree (HMT) [Xie *et al.*, 2018] is a state-of-the-art graphical model for flood extent mapping. HMT captures the directed spatial dependency based on flow directions across all locations by a reverse tree structure in the hidden class layer, which is similar to the hidden state sequence in hidden Markov models (HMM); spectral features are read out from the class state (dry or flooded) based on a learned multivariate Gaussian distribution. Later, hidden Markov contour tree (HMCT) [Jiang and Sainju, 2019] further captures complex contour structures on a 3D surface.

Both HMT and HMCT are for transductive learning where training samples are taken near the test region, or where a complete elevation map plus only limited spectral features in a target region are observed [Sainju *et al.*, 2020]. More recently, inductive learning with HMT is explored with the help of U-Net. [Sainju *et al.*, 2022] proposes to use the inference stage of HMCT only as a post-processor of U-Net to cleanse its predicted noisy posterior class probability at all the pixels. The resulting model is called HMCT-PP. Our EvaNet can also serve as a drop-in replacement of the U-Net in HMCT-PP.

Since the size of earth imagery on a terrain surface is often much larger than the input size of popular deep convolutional neural networks, segmentation is usually conducted on patches and the results are then stitched together. In contrast, HMT-based methods build a topological tree on the entire terrain surface so inference of the entire imagery is conducted together all at once. In HMCT-PP, the stitched U-Net predictions serve as the input to HMCT refinement.

Knowledge-Guided Machine Learning (KGML). Unlike our problem that focuses on the Earth imagery at one snapshot to quickly map out flood-inundated area for disaster response, KGML research is usually formulated as a forecast problem fitting historical data or fitting a simulation model (e.g., based on PDE), so requires many temporal snapshots. The goal of KGML is more on geosciences to study climate and environmental phenomena/process for future disaster warning and mitigation, often using meteorological fac-

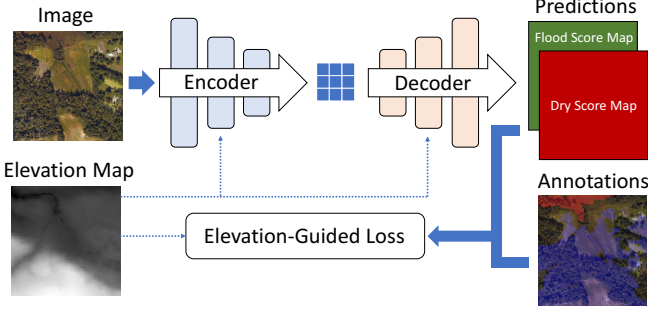| Notation	Meaning
$\mathbf{\Omega}$	The image as a 2D grid
$\mathbf{p}$	A pixel in $\mathbf{\Omega}$
$\mathcal{N}(\mathbf{p})$	The 8 neighboring pixels of $\mathbf{p}$
$\mathbf{p}_n$	A pixel adjacent to $\mathbf{p}$ $(\mathbf{p}_n \in \mathcal{N}(\mathbf{p}))$
$h(\mathbf{p})$	The elevation of $\mathbf{p}$
$\Delta h(\mathbf{p}, \mathbf{p}_n)$	The elevation difference $h(\mathbf{p}) - h(\mathbf{p}_n)$
$s_{flood}(\mathbf{p})$	The predicted flood-score of $\mathbf{p}$
$s_{dry}(\mathbf{p})$	The predicted dry-score of $\mathbf{p}$

Table 1: List of Notations

Figure 2: EvaNet architecture overview.

tors such as air temperature, precipitation, and wind speed. We discuss the related works on KGML in our online appendix [Sami *et al.*, 2024].

3 Method

In this section, we first formally define our problem and overview our techniques used in EvaNet. We then describe both techniques in detail.

Notations and Problem Definition. We regard each image as a 2D grid $\mathbf{\Omega}$, and denote a pixel by $\mathbf{p} = (i, j)$, where i and j are the pixel coordinates in $\mathbf{\Omega}$. We also denote the elevation of pixel $\mathbf{p}$ by $h(\mathbf{p})$. Each pixel $\mathbf{p}$ has 8 adjacent pixels which constitute $\mathbf{p}$'s neighbor set, denoted by $\mathcal{N}(\mathbf{p})$. While pixels on the boundary of $\mathbf{\Omega}$ do not have 8 neighbor pixels, we pad the input image and its elevation map using the reflection of the input boundary, to ensure that all $\mathbf{p} \in \mathbf{\Omega}$ has 8 neighbors. We also use $\mathbf{p}_n$ to denote an arbitrary neighbor of $\mathbf{p}$, i.e., $\mathbf{p}_n \in \mathcal{N}(\mathbf{p})$. We call $(\mathbf{p}, \mathbf{p}_n)$ as a *pixel-pair*, where $\mathbf{p}$ and $\mathbf{p}_n$ are adjacent to each other in $\mathbf{\Omega}$. Finally, we define

$$\Delta h(\mathbf{p}, \mathbf{p}_n) = h(\mathbf{p}) - h(\mathbf{p}_n). \quad (1)$$

In Figure 1, we have $\Delta h(A, B) < 0$ and $\Delta h(D, E) > 0$.

The goal of flood extent mapping is to assign each pixel $\mathbf{p} \in \mathbf{\Omega}$ a binary label indicating whether it is flooded or dry. Table 1 provides the list of notations used in this paper.

EvaNet Overview. Figure 2 overviews the architecture of EvaNet, where the encoder and decoder networks take not only the image but also the elevation map in convolution computation, which we will explain in section "Elevation-Regulated Convolutions"; and where the loss function takes not only the predictions and the ground-truth annotation, but also the elevation map in the loss computation, which we will explain in section "Elevation-Guided Loss".

3.1 Elevation-Guided Loss

Given an input image of size $W \times H$, a flood segmentation model (U-Net or EvaNet) outputs a tensor of size $W \times H \times 2$ with 2 channels: one is a dry score map $s_{dry} \in \mathbb{R}^{W \times H}$, and the other is a flood score map $s_{flood} \in \mathbb{R}^{W \times H}$, as illustrated in Figure 2. To compute the loss function, we apply pixel-wise softmax where the probability that a pixel $\mathbf{p}$ is flooded (resp. dry) is given by $p_{flood}(\mathbf{p}) = \frac{e^{s_{flood}(\mathbf{p})}}{e^{s_{flood}(\mathbf{p})} + e^{s_{dry}(\mathbf{p})}}$ (resp. $p_{dry}(\mathbf{p}) = \frac{e^{s_{dry}(\mathbf{p})}}{e^{s_{flood}(\mathbf{p})} + e^{s_{dry}(\mathbf{p})}}$)[1].

We define the ground-truth label of a pixel $\mathbf{p}$ as follows:

$$gt(\mathbf{p}) = \begin{cases} 1 & \mathbf{p} \text{ is flooded,} \\ 0 & \mathbf{p} \text{ is unlabeled,} \\ -1 & \mathbf{p} \text{ is dry.} \end{cases}$$

Note that a pixel may be unlabeled since it is covered by tree canopy. We provide domain experts with a 3D visualization tool where they can navigate the terrain surface by shifting and rotation, and where they can annotate the pixels using tools like 'brush' if they are confident of the pixel labels. Elevation-guided breadth-first search (BFS) [Adhikari *et al.*, 2022] was adopted to speed up annotation, where (i) when an annotator marks an individual pixel $\mathbf{p}$ as flooded, the label propagates to nearby pixels with lower elevations by 'pit-filling' breadth-first search stopping when reaching pixels with elevation higher than $h(\mathbf{p})$, and (ii) when an annotator marks an individual pixel $\mathbf{p}$ as dry, the label propagates to nearby pixels by 'hill-climbing' breadth-first search stopping when reaching pixels with elevation starting to drop. The labels of some pixels covered by tree canopy may be derived in this way from nearby pixels using the physical law of gravity.

U-Net uses only pixel-wise cross-entropy loss:

$$\mathcal{L}_{CE} = - \sum_{\mathbf{p} \in \mathbf{\Omega} \, \wedge \, gt(\mathbf{p})=1} \log p_{flood}(\mathbf{p}) - \sum_{\mathbf{p} \in \mathbf{\Omega} \, \wedge \, gt(\mathbf{p})=-1} \log p_{dry}(\mathbf{p}).$$

Elevation-Guided Regularization. However, $\mathcal{L}_{CE}$ does not take the pixel elevations into consideration. To utilize the elevation map, we propose a regularization term $\mathcal{L}_{eva}$ to penalize predictions that violate the physical constraints shown in Figure 1. Unlike $\mathcal{L}_{CE}$ which is computed over individual pixels $\mathbf{p}$, $\mathcal{L}_{eva}$ is computed over individual pixel-pairs $(\mathbf{p}, \mathbf{p}_n)$:

$$\mathcal{L}_{eva} = \sum_{\mathbf{p} \in \mathbf{\Omega} \, \wedge \, gt(\mathbf{p}) \neq 0} \sum_{\mathbf{p}_n \in \mathcal{N}(\mathbf{p})} \Big(w(\mathbf{p}, \mathbf{p}_n) \cdot \delta(\mathbf{p}, \mathbf{p}_n) \Big), \quad (2)$$

where (i) $w(\mathbf{p}, \mathbf{p}_n)$ is a weight term indicating how much the pixel-pair $(\mathbf{p}, \mathbf{p}_n)$ is relevant to the physical constraints, and (ii) $\delta(\mathbf{p}, \mathbf{p}_n)$ measures how much the label predictions of $\mathbf{p}$ and $\mathbf{p}_n$ deviate from the physical constraints. We next explain the formulations of $w(\mathbf{p}, \mathbf{p}_n)$ and $\delta(\mathbf{p}, \mathbf{p}_n)$.

Note that if a pixel $\mathbf{p}$ is unlabeled (i.e., $gt(\mathbf{p}) = 0$), it has no term in the summation formulations of $\mathcal{L}_{CE}$ and $\mathcal{L}_{eva}$, so it will not provide any supervision during training.

---

[1]Alternatively, since our label is binary, one may let the segmentation model output only flood score map s_{flood}, and compute $p_{flood} = \text{sigmoid}(s_{flood})$ and $p_{dry} = 1 - p_{flood}$. However, we find this formulation to be less numerically stable in PyTorch than using softmax.
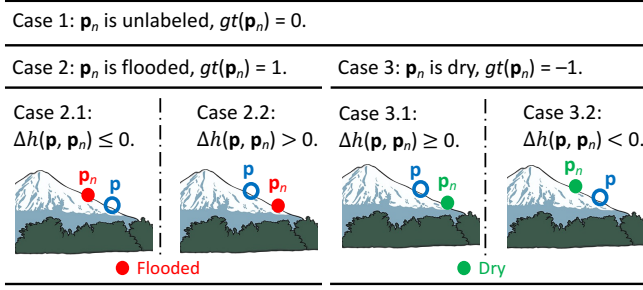
Figure 3: Five cases for pixel-pair $(\mathbf{p}, \mathbf{p}_n)$.

Formulation of $\delta(\mathbf{p}, \mathbf{p}_n)$. We define $\delta(\mathbf{p}, \mathbf{p}_n)$ as follows, and we will explain its intuition soon:

$$\delta(\mathbf{p}, \mathbf{p}_n) = 1 - gt(\mathbf{p}_n) \cdot f(\mathbf{p}), \qquad (3)$$

where $f(\mathbf{p}) \in (-1, 1)$ is defined as follows:

$$f(\mathbf{p}) = \begin{cases} \text{sigmoid}(s_{flood}(\mathbf{p})) & f_{flood}(\mathbf{p}) \geq f_{dry}(\mathbf{p}), \\ -\text{sigmoid}(s_{dry}(\mathbf{p})) & f_{flood}(\mathbf{p}) < f_{dry}(\mathbf{p}). \end{cases}$$

In other words, the higher confidence that our prediction is for $\mathbf{p}$ to be flooded (resp. dry), the closer $f(\mathbf{p})$ is to 1 (resp. -1). An alternative definition that achieves this goal is to let the segmentation model output only a flood score map s_{flood}, and compute $f(\mathbf{p}) = \tanh(s_{flood})$. However, we find this formulation less numerically stable in PyTorch.

Formulation of $w(\mathbf{p}, \mathbf{p}_n)$. We define $w(\mathbf{p}, \mathbf{p}_n)$ as follows, and we will explain its intuition soon:

$$w(\mathbf{p}, \mathbf{p}_n) = \mathbf{1}\{-gt(\mathbf{p}_n) \cdot \Delta h(\mathbf{p}, \mathbf{p}_n) \geq 0\}, \qquad (4)$$

where $\mathbf{1}\{.\}$ is an indicator variable that equals 1 if the condition holds, and 0 otherwise. Note that only $\delta(\mathbf{p}, \mathbf{p}_n)$ involves prediction since it uses $f(\mathbf{p})$; in contrast, $w(\mathbf{p}, \mathbf{p}_n)$ is purely decided by $gt(\mathbf{p}_n)$, $h(\mathbf{p})$ and $h(\mathbf{p}_n)$, i.e., annotations and elevations. Also note that if $\mathbf{p}_n$ is unlabeled (i.e., $gt(\mathbf{p}_n) = 0$), then $w(\mathbf{p}, \mathbf{p}_n) = 0$ so the term in the summation formulation of $\mathcal{L}_{eva}$ in Eq (2) is 0, i.e., pixel-pair $(\mathbf{p}, \mathbf{p}_n)$ provides no supervision during training.

Intuition Behind the Formulations. We next provide the intuition of our formulations for $\delta(\mathbf{p}, \mathbf{p}_n)$ and $w(\mathbf{p}, \mathbf{p}_n)$. We divide the possible scenarios for a pixel-pair $(\mathbf{p}, \mathbf{p}_n)$ into 5 cases shown in Figure 3, based on $gt(\mathbf{p}_n)$ as the primary partitioning key and $\Delta h(\mathbf{p}, \mathbf{p}_n)$ as the secondary key.

The physical constraints illustrated in Figure 1 correspond to 2 sub-cases in Figure 3: (1) Case 2.1 where we can derive that $\mathbf{p}$ should also be flooded, and (2) Case 3.1 where $\mathbf{p}$ should also be dry. Note that based on Eq (4), $w(\mathbf{p}, \mathbf{p}_n) = 1$ for these 2 cases, and $w(\mathbf{p}, \mathbf{p}_n) = 0$ for the other 3 cases where $\mathbf{p}$ can be either dry or flooded without violating any constraints. As a result, $w(\mathbf{p}, \mathbf{p}_n)$ in Eq (4) acts as a switch that is on only for those pixel-pairs $(\mathbf{p}, \mathbf{p}_n)$ that belong to Case 2.1 and Case 3.1, and excludes the other pixel-pairs when computing $\mathcal{L}_{eva}$ using Eq (2). We call this weighting scheme given by Eq (4) as **binary weighting**.

Based on Eq (3), (1) $\delta(\mathbf{p}, \mathbf{p}_n) = 1 - f(\mathbf{p})$ in Case 2.1, so if $f(\mathbf{p})$ is close to 1 (i.e., predicted closer to be flooded) as aligned with the physical constraint, then $\delta(\mathbf{p}, \mathbf{p}_n)$ is close to

0; while if $f(\mathbf{p})$ is close to -1 (i.e., predicted closer to be dry) which violates the physical constraint, then $\delta(\mathbf{p}, \mathbf{p}_n)$ is close to the supremum 2 so the penalty term in $\mathcal{L}_{eva}$ is maximized. Also, (2) $\delta(\mathbf{p}, \mathbf{p}_n) = 1 + f(\mathbf{p})$ in Case 3.1 by Eq (3), so if $f(\mathbf{p})$ is close to -1 (i.e., predicted closer to be dry) as aligned with the physical constraint, then $\delta(\mathbf{p}, \mathbf{p}_n)$ is close to 0; while if $f(\mathbf{p})$ is close to 1 (i.e., predicted closer to be flooded) which violates the physical constraint, then $\delta(\mathbf{p}, \mathbf{p}_n)$ is close to 2 to maximize the penalty term.

Figure 5 in our online appendix [Sami $et\ al.$, 2024] shows the pixel distribution among the 5 cases on our datasets, where Cases 2.1 and 3.1 are dominating where the physical constraints apply, so we expect sufficient regularization.

Alternative Formulations of $w(\mathbf{p}, \mathbf{p}_n)$. The binary weighting scheme in Eq (4) only switches on the pixel-pairs satisfying Case 2.1 and Case 3.1, and treats their penalty terms as equally important. We also explored the adjustment of weights based on $|\Delta h(\mathbf{p}, \mathbf{p}_n)|$. The intuition is that, for example in Case 2.1, the higher $\mathbf{p}_n$ is than $\mathbf{p}$, the more penalty weight will be imposed if $\mathbf{p}$ is misclassified as dry.

Based on this idea, we define the **elevation-difference weighting scheme** as follows:

$$w(\mathbf{p}, \mathbf{p}_n) = \max\{-gt(\mathbf{p}_n) \cdot \Delta h(\mathbf{p}, \mathbf{p}_n), 0\}, \qquad (5)$$

so that $w(\mathbf{p}, \mathbf{p}_n) = |\Delta h(\mathbf{p}, \mathbf{p}_n)|$ for Cases 2.1 and 3.1, and 0 otherwise. We also explored a **log-elevation-difference weighting scheme** where $w(\mathbf{p}, \mathbf{p}_n) = \log\left(1 + |\Delta h(\mathbf{p}, \mathbf{p}_n)|\right)$ for Cases 2.1 and 3.1, and 0 otherwise:

$$w(\mathbf{p}, \mathbf{p}_n) = \log\Big(1 + \max\{-gt(\mathbf{p}_n) \cdot \Delta h(\mathbf{p}, \mathbf{p}_n), 0\}\Big). \quad (6)$$

However, we find that these weighting schemes are not better than binary weighting. The results of comparison can be found in Table 7 of the appendix [Sami $et\ al.$, 2024].

Implementing $\mathcal{L}_{eva}$. Recall that Eq (2) sums over pixel-pairs rather than individual pixels, but both the predicted score maps s_{flood} and s_{dry} and the ground-truth map gt are on pixels. Fortunately, PyTorch supports an "unfold" operation that can expand each pixel to include its local 3×3 pixel-neighborhood. We utilize "unfold" to expand each pixel $\mathbf{p}$ in s_{flood}, s_{dry} and gt with its neighbors $\mathbf{p}_n$ to facilitate the computation of $w(\mathbf{p}, \mathbf{p}_n)$ and $\Delta h(\mathbf{p}, \mathbf{p}_n)$.

Loss Schemes. Since $\mathcal{L}_{eva}$ is formulated as a regularization term that penalizes pixel-pairs $(\mathbf{p}, \mathbf{p}_n)$ in Cases 2.1 and 3.1 based on how much the label prediction of $\mathbf{p}$ violates the physical constraints, a natural loss function is given by

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \cdot \mathcal{L}_{eva}, \qquad (7)$$

where hyperparameter λ balances the importance between $\mathcal{L}_{CE}$ and $\mathcal{L}_{eva}$. U-Net simply uses $\mathcal{L} = \mathcal{L}_{CE}$ (i.e., $\lambda = 0$). By setting $\lambda = 1$, we obtain a representative hybrid loss $\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{eva}$. Another extreme is using $\mathcal{L} = \mathcal{L}_{eva}$ (i.e., $\lambda = \infty$), which could be a bit surprising since we exclude $\mathcal{L}_{CE}$ and only use the regularization term to penalize the violation of physical constraints. Surprisingly, this last loss turns out to perform well in our experiments (c.f., Table 6 in our appendix [Sami $et\ al.$, 2024]), probably because Cases 2.1 and 3.1 are well spread out on the entire terrain (c.f., Figure 5 in our appendix [Sami $et\ al.$, 2024]). In other words, $\mathcal{L}_{eva}$ by itself serves as a good loss function for flood mapping.
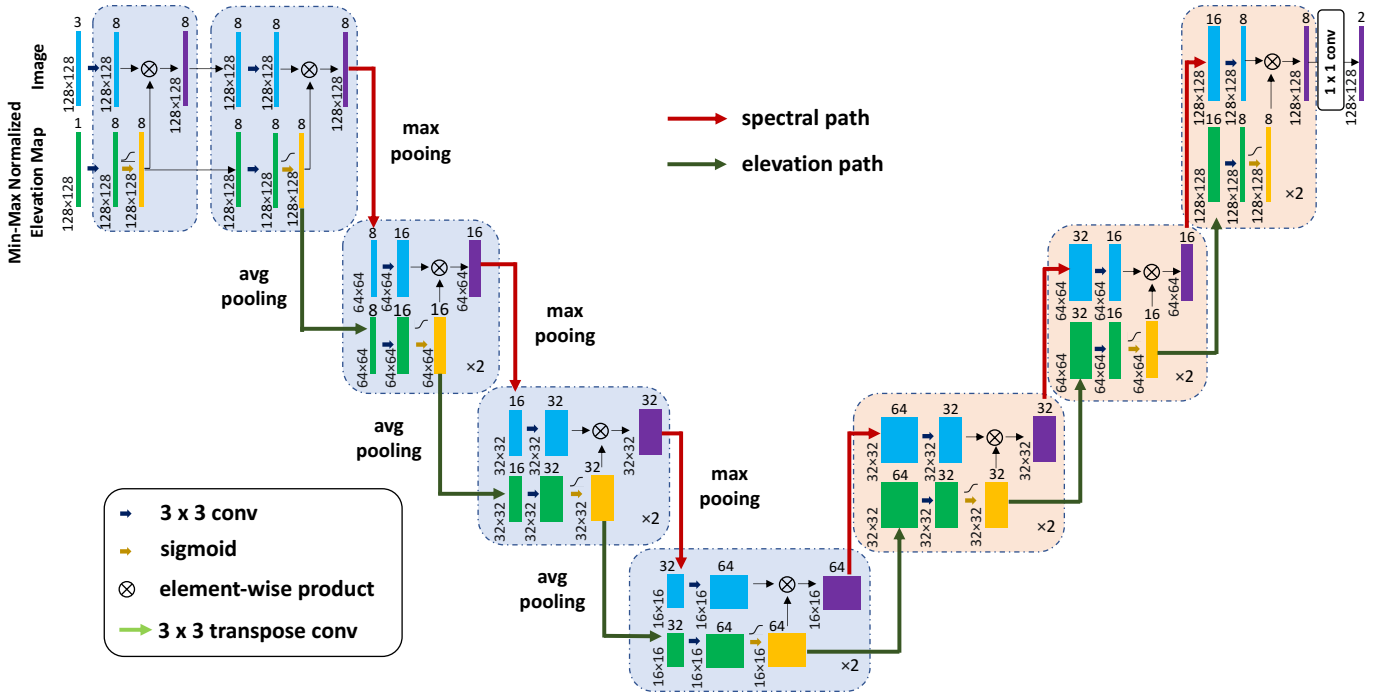
Figure 4: EvaNet encoder-decoder network architecture.

3.2 Elevation-Regulated Convolutions

Figure 4 shows the architecture of EvaNet's encoder-decoder network, where the encoder and decoder each consists of 3 convolution blocks, each with two novel elevation-regulated convolution (ERC) layers to be introduced soon.

EvaNet takes two inputs: (1) a 128×128 image patch and (2) the associated elevation map of the patch. Here, we partition a large high-resolution satellite image into 128×128 patches, since an entire satellite image is too large to input into any convolution neural network given the current GPU memory capacity. For this purpose, we pad each image on its boundary using the reflection mode, so that both the width and height of the padded image is divisible by 128. For example, for an 1856×4104 input image, we pad 32 pixels on the left and right boundaries, and pad 60 pixels on the upper and lower boundaries, giving a padded image of size 1920×4224 which gives 15×33 patches.

Since neural networks take normalized inputs, we conduct min-max normalization of the elevations and input the normalized elevation patch to our encoder. Note that in contrast, raw elevations are used when we calculate $w(\mathbf{p}, \mathbf{p}_n)$ (using Eq (4), Eq (5) or Eq (6)) to compute $\mathcal{L}_{eva}$.

As Figure 4 shows, each convolution block consists of two ERC layers. Each ERC layer takes a spectral tensor (blue) $\mathbf{X}$ and an elevation-derived tensor (green) $\mathbf{X}_e$, and outputs a regulated spectral tensor (purple) $\mathbf{Y}$ and a gated elevation-derived tensor (yellow) $\mathbf{Y}_e$. Specifically, both inputs first pass through a 3×3 convolution, and the two resulting tensors are combined by gated linear unit (GLU) [Dauphin *et al.*, 2017]:
$$\mathbf{Y}_e = \text{sigmoid}\big(\text{conv2d}(\mathbf{X}_e)\big), \mathbf{Y} = \text{conv2d}(\mathbf{X}) \otimes \mathbf{Y}_e, \text{ where}$$
conv2d(.) is a 3×3 convolution with a 'replicate' padding mode, and $\otimes$ is element-wise product. The outputs $(\mathbf{Y}, \mathbf{Y}_e)$ then serve as the inputs to the next ERC layer.

Intuitively, $\mathbf{Y}_e \in (0, 1)^{H \times W \times C}$ serves as an elevation-based gate to control the flow of conv2d($\mathbf{X}$) to $\mathbf{Y}$, and since $\mathbf{Y}_e$ is computed from $\mathbf{X}_e$ with a 3×3 convolution, it is able to capture the local terrain topology. We choose GLU for elevation-based regulation since it is found to outperform other gating mechanisms in recent works [Wu *et al.*, 2019; Hui *et al.*, 2021b; Hui *et al.*, 2021a]. We also tested other alternative gating mechanisms such as [Arevalo *et al.*, 2017; Kim *et al.*, 2018] and found GLU to perform the best. The related experiments are reported in Table 5 of our online appendix [Sami *et al.*, 2024].

Also, we use 'replicate' padding for conv2d(.) to ensure the quality of predictions on patch boundary. This is in contrast to zero padding, which we tested and found that the abrupt value drops at the patch boundary lead to poor prediction quality for pixels at the boundary.

As Figure 4 shows, at the encoder side, $\mathbf{Y}$ is downsampled using 2×2 max pooling which is commonly used for spectral tensors, and $\mathbf{Y}_e$ is downsampled using 2×2 average pooling as a lower-resolution terrain approximation. We tried other pooling combinations but this setting gives the best predictions as demonstrated in Table 8 of the online appendix [Sami *et al.*, 2024]. At the decoder side, we use 3×3 transpose convolution for both $\mathbf{Y}$ and $\mathbf{Y}_e$.

4 Experiments & Results

Data. We obtain high-resolution aerial imagery from NOAA National Geodetic Survey during Hurricane Matthew in North Carolina (NC) in 2016 [NOAA, 2016]. The accompanied DEM data are obtained from the University of North Carolina Libraries [NCSU, 2023]. Futhermore, to test if our models trained on the Matthew 2016 dataset can generalize to a different flooding event, we also obtained two more

	Height	Width	Region	%Dry	%Flood	%Annotated
R1	1856	4104	Grimesland, NC	19.97%	41.22%	61.19%
R2	2240	4704	Greenville-Central, NC	52.32%	34.43%	86.75%
R3	3136	6472	Falkland, NC	58.35%	26.54%	84.89%
R4	2782	5500	Kinston, NC	49.20%	40.95%	90.15%
R5	2800	5400	Greenville-West, NC	45.18%	27.63%	72.81%
R6	2828	5286	Baytown, TX	37.56%	39.48%	77.04%
R7	2306	4802	Richmond, TX	44.44%	36.44%	80.88%

Table 2: Regions and their statistics.

regions in Baytown, TX, and Richmond, TX during Hurricane Harvey in 2017 from [NOAA, 2017], and the corresponding DEM data was obtained from USGS data downloader [USGS, 2023]. The aerial imagery from both events are 0.3 m × 0.3 m in resolution, the DEM data for Matthew 2016 are 6 m × 6 m and for Harvey 2017 are 3 m × 3 m in resolution. We resampled them into a resolution of 2 m × 2 m, which is fine enough for the purpose of flood extent mapping. Table 2 shows the regions we used, where R1–R5 are from Matthew 2016, and R6–R7 is from Harvey 2017. While high-resolution Earth imagery and DEM data are abundant, the ground-truth flood maps need to be annotated by domain experts. So, we provide a 3D visualization tool to domain experts for annotating pixels.

In our online appendix [Sami *et al.*, 2024], Figure 6 shows the imagery for Kinston, NC during Hurricane Matthew in 2016, its annotation where pixels are labeled as flooded or dry, and the normal-time imagery from Google Earth. We expect that inputting the normal-time image into a model is effective in helping identify flooded regions by providing a contrast. This is confirmed our experiments reported in Table 10 of our online appendix, which also aligns with findings from prior works [Drakonakis *et al.*, 2022; Rudner *et al.*, 2019]. Some pixels are ambiguous (e.g., covered with tree canopies) so remain unlabeled. Unlabeled pixels are not included in the loss computation during training, and not included during test when calculating the evaluation measures. As Table 2 shows, most pixels of each region are annotated. Without loss of generalization, in Table 2, we use R1 and R2 for training, and use R3, R4, R5, R6 and R7 for test.

Models. We consider three U-Net baselines: (1) "U-Net 3C" with input using the RGB channels of the disaster-time image; (2) "U-Net 4C" with input adding an additional channel for the elevation map; (3) "U-Net 7C" with input also adding the RGB channels of the normal-time image.

For EvaNet, we consider two versions: (1) "EvaNet 4C" using only the disaster-time RGB image for the leftmost cyan 'image' input tensor shown in Figure 4, and (2) "EvaNet 7C" using both disaster-time and normal-time images for that input tensor. We remark that our new elevation-regulated convolution (denoted by **Eva-Conv**) and elevation-guided loss (denoted by **Eva-Reg**) can serve as drop-in replacements of the regular convolution and binary cross-entropy loss in any encoder-decoder model for semantic segmentation, and we choose the popular U-Net model just as a representative.

For physics-guided graphical models, we compare with hidden Markov tree (HMT) [Xie *et al.*, 2018; Jiang and Sainju, 2019]. Since HMT uses transductive learning, we use the same region for training and test. For U-Net and EvaNet, we use two regions R1 and R2 in Table 2 for training and the other regions for test. For example, when reporting the performance measures on R3, HMT is trained on R3 and tested on R3, while all other models are trained with the patches of R1 and R2, and then tested on patches of R3 and stitched.

By default, EvaNet uses only $\mathcal{L}_{eva}$ in Eq (2) as the loss function, where $w(\mathbf{p}, \mathbf{p}_n)$ uses the binary weighting scheme in Eq (4). We use 3 convolution blocks of Eva-Conv for both the encoder and the decoder as shown in Figure 4, where the input has 7 channels (disaster-time and normal-time RGB plus elevation map), and the encoder downsamples the spectral path using max pooling, and downsamples the elevation path using average pooling. These default configurations are found to be the best-performing, as demonstrated by our experiments in the online appendix [Sami *et al.*, 2024] (see Tables 6–10). We trained our EvaNet models and the U-Net baselines on a cluster with NVIDIA-P100 GPUs for 100 epochs with a learning rate of 1×10^{-7} and batch size of 4. These default settings are found to be the best-performing through extensive tests.

Performance Measures. Accuracy, precision, recall and F1-score are commonly used measures for binary classification. Following prior works such as [Xie *et al.*, 2018; Jiang and Sainju, 2019], we consider our flood segmentation problem as a pixel-wise binary classification problem, and report the above four measures in two contexts: (1) "Dry" where dry (resp. flood) is the positive (resp. negative) class, and (2) "Flood" where dry (resp. flood) is the positive (resp. negative) class. These measures are computed only from labeled pixels in the test images.

Comparison with Baselines. Here, we consider 3 representative baselines: (1) HMT [Xie *et al.*, 2018], which is a graphical model for transductive learning that considers both the disaster-time RGB image and the elevation map, (2) U-Net 3C, which is a U-Net model that only considers the disaster-time RGB image, (3) U-Net 7C, which is a U-Net model that considers both the RGB images and the elevation map. They are compared with EvaNet 7C in Table 3. We skip 4C models since they are dominated by 7C ones (c.f. Table 10 in our online appendix [Sami *et al.*, 2024]).

We can see that EvaNet 7C consistently gives the highest accuracy and F1-score, much higher than all the baselines. Among the other models, HMT performs well on R3, R4 and R7, but very poorly on R5 and R6 where more pixels are predicted as dry, which shows that HMT is region-sensitive and not very robust. U-Net 7C is the second best on R3, R5, and R6 and generally better than U-Net 3C and HMT, but U-Net

Region	Method	Dry				Flood			
		Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
R3: Falkland, NC	HMT	91.91	95.31	92.80	94.03	91.91	85.03	89.95	87.42
	U-Net 3C	81.26	95.48	76.34	84.85	81.26	63.90	92.05	75.44
	U-Net 7C	94.94	95.91	96.77	96.34	94.94	92.76	90.93	91.83
	EvaNet 7C	**97.15**	**97.63**	**98.23**	**97.93**	**97.15**	**96.06**	**94.77**	**95.41**
R4: Kinston, NC	HMT	84.33	81.14	92.88	86.61	84.33	89.64	74.06	81.11
	U-Net 3C	90.70	**93.57**	89.08	91.27	90.70	87.60	**92.64**	90.05
	U-Net 7C	83.90	78.45	97.20	86.82	83.90	95.28	67.92	79.31
	EvaNet 7C	**92.16**	88.64	**98.22**	**93.18**	**92.16**	**97.54**	84.88	**90.77**
R5: Greenville-West, NC	HMT	61.91	62.21	**98.39**	76.23	61.91	46.18	2.25	4.30
	U-Net 3C	86.12	96.93	80.18	87.76	86.12	74.73	95.84	83.98
	U-Net 7C	95.97	98.49	94.96	96.70	95.97	92.21	97.62	94.84
	EvaNet 7C	**97.75**	**98.65**	97.70	**98.18**	**97.75**	**96.30**	97.81	**97.05**
R6: Baytown, TX	HMT	63.30	57.08	**99.65**	72.59	63.30	**98.86**	28.73	44.52
	U-Net 3C	63.68	88.50	29.31	44.0	63.68	58.90	96.38	73.12
	U-Net 7C	74.41	88.76	54.40	67.46	74.41	68.30	93.45	78.90
	EvaNet 7C	**91.50**	**98.46**	83.89	**90.59**	**91.50**	86.56	**98.75**	**92.26**
R7: Richmond, TX	HMT	74.67	68.61	**99.38**	81.17	76.67	**98.33**	44.55	61.32
	U-Net 3C	53.95	81.36	20.99	33.38	53.95	49.42	**94.13**	64.81
	U-Net 7C	56.13	74.17	30.92	43.65	56.13	50.77	86.87	64.08
	EvaNet 7C	**84.06**	**90.07**	79.79	**84.62**	**84.06**	78.37	89.28	**83.46**

Table 3: Model comparison (unit: %).

Method	Dry				Flood			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
U-Net 7C	83.90	78.45	97.20	86.82	83.90	95.28	67.92	79.31
U-Net 7C w/ Eva-Conv	85.56	79.75	98.56	88.17	85.56	97.61	69.93	81.48
U-Net 7C w/ Eva-Reg	86.59	80.96	**98.64**	88.93	86.59	**97.78**	72.13	83.02
EvaNet 7C	**92.16**	**88.64**	98.22	**93.18**	**92.16**	97.54	**84.88**	**90.77**

Table 4: Ablation study (unit: %) on test region R4.

7C is even worse than U-Net 3C on R4. We explored the data and found that our pixels in training regions all have elevation below 80 meters, but quite a few pixels in R4 are higher than 90 meters, leading to poor model generalization when the elevation map is treated as an additional input channel backfiring on performance. However, our way of using the elevation map in EvaNet 7C does not cause any generalization issue, and EvaNet 7C still gives the best performance on R4; it also outperforms all other models on R6 and R7, which is from a different flooding event. This shows that our EvaNet design is robust thanks to the elevation guidance.

Ablation Study. To verify the effectiveness of both our Eva-Conv and Eva-Reg designs, we conduct an ablation study using 7-channel input, by exploring the performance of 4 models: (1) U-Net 7C (i.e., with loss $\mathcal{L}_{CE}$), (2) U-Net with our loss function $\mathcal{L}_{eva}$ (denoted by U-Net 7C + Eva-Reg), (3) U-Net 7C with our new Eva-Conv operations in both encoder and decoder (denoted by U-Net 7C + Eva-Conv), and (4) EvaNet 7C, i.e., with both Eva-Conv and Eva-Reg. Without loss of generality, Table 4 shows the performance results on test region R4, where we can see that (i) both Eva-Reg and Eva-Conv improve the model accuracy and F1-score as compared with U-Net 7C, (ii) Eva-Reg (i.e., our new loss $\mathcal{L}_{eva}$)

improves the performance more significantly, and (iii) our EvaNet 7C model combining both designs provides the highest accuracy and F1-score. In our online appendix [Sami *et al.*, 2024], we show that our techniques are effective also in other ConvNet architectures using Table 11, and **we also visually inspect the flood maps from different models in Figure 7 to confirm our conclusions**.

Effect as a U-Net Drop-in Replacement. We also find that EvaNet can improve the performance of an existing model such as HMCT-PP [Sainju *et al.*, 2022] (i.e., U-Net + HMCT) by replacing its U-Net component. Due to space limit, please refer to Table 12 in our online appendix [Sami *et al.*, 2024] for more details.

5 Conclusion

We presented EvaNet, a physics-guided segmentation model for flood extent mapping on satellite imagery. EvaNet is based on the popular encoder-decoder architecture with two novel techniques: (1) an elevation-guided loss function, and (2) an elevation-regulated convolution. Extensive experiments show that both techniques are effective, and that EvaNet works as a perfect drop-in replacement for U-Net in flood mapping solutions such as HMCT-PP.

Acknowledgements

References

[Adhikari *et al.*, 2022] Saugat Adhikari, Da Yan, Mirza Tanzim Sami, Jalal Khalil, Lyuheng Yuan, Bhadhan Roy Joy, Zhe Jiang, and Arpan Man Sainju. An elevation-guided annotation tool for flood extent mapping on earth imagery. In *SIGSPATIAL*. ACM, 2022.

[Arevalo *et al.*, 2017] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017.

[Beni and Gebrehiwot, 2021] Leila Hashemi Beni and Asmamaw Gebrehiwot. Flood extent mapping: An integrated method using deep learning and region growing using UAV optical data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 14:2127–2135, 2021.

[Bonafilia *et al.*, 2020] Derrick Bonafilia, Beth Tellman, Tyler Anderson, and Erica Issenberg. Sen1floods11: A georeferenced dataset to train and test deep learning flood algorithms for sentinel-1. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 210–211, 2020.

[Copernicus, 2014] Copernicus. Sentinel-1 - missions - sentinel online - sentinel online. https://sentinel.esa.int/web/sentinel/missions/sentinel-1, 2014. (Accessed on 07/29/2023).

[Copernicus, 2015] Copernicus. Sentinel-2 - missions - sentinel online - sentinel online. https://sentinel.esa.int/web/sentinel/missions/sentinel-2, 2015. (Accessed on 07/29/2023).

[Dauphin *et al.*, 2017] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR, 2017.

[Drakonakis *et al.*, 2022] Georgios I Drakonakis, Grigorios Tsagkatakis, Konstantina Fotiadou, and Panagiotis Tsakalides. Ombrianet—supervised flood mapping via convolutional neural networks using multitemporal sentinel-1 and sentinel-2 data fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:2341–2356, 2022.

[Fraccaro *et al.*, 2022] Paolo Fraccaro, Nikola Stoyanov, Zaheed Gaffoor, Laura Elena Cue La Rosa, Jitendra Singh, Tatsuya Ishikawa, Blair Edwards, Anne Jones, and Komminist Weldermariam. Deploying an artificial intelligence application to detect flood from sentinel 1 data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12489–12495, 2022.

[Ghosh *et al.*, 2022] Rahul Ghosh, Arvind Renganathan, Kshitij Tayal, Xiang Li, Ankush Khandelwal, Xiaowei Jia, Christopher J. Duffy, John Nieber, and Vipin Kumar. Robust inverse framework using knowledge-guided self-supervised learning: An application to hydrology. In *KDD*, pages 465–474. ACM, 2022.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988. IEEE Computer Society, 2017.

[Hui *et al.*, 2021a] Bo Hui, Da Yan, Haiquan Chen, and Wei-Shinn Ku. Trajectory wavenet: A trajectory-based model for traffic forecasting. In *ICDM*, pages 1114–1119. IEEE, 2021.

[Hui *et al.*, 2021b] Bo Hui, Da Yan, Haiquan Chen, and Wei-Shinn Ku. Trajnet: A trajectory-based deep learning model for traffic prediction. In *KDD*, pages 716–724. ACM, 2021.

[Jiang and Sainju, 2019] Zhe Jiang and Arpan Man Sainju. Hidden markov contour tree: A spatial structured model for hydrological applications. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *KDD*, pages 804–813. ACM, 2019.

[Jiang *et al.*, 2021] Peishi Jiang, Nis Meinert, Helga Jordão, Constantin Weisser, Simon Holgate, Alexander Lavin, Björn Lütjens, Dava Newman, Haruko Wainwright, Catherine Walker, and Patrick Barnard. Digital twin earth – coasts: Developing a fast and physics-informed surrogate model for coastal floods via neural operators. *arXiv preprint arXiv:2110.07100*, 2021.

[Kim *et al.*, 2018] Jaekyum Kim, Junho Koh, Yecheol Kim, Jaehyung Choi, Youngbae Hwang, and Jun Won Choi. Robust deep multi-modal learning based on gated information fusion network. In *Asian Conference on Computer Vision*, pages 90–106. Springer, 2018.

[Kirillov *et al.*, 2019] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, pages 9404–9413. Computer Vision Foundation / IEEE, 2019.

[Konapala *et al.*, 2021] Goutam Konapala, Sujay V Kumar, and Shahryar Khalique Ahmad. Exploring sentinel-1 and sentinel-2 diversity for flood inundation mapping using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 180:163–173, 2021.

[Long *et al.*, 2015a] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440. IEEE Computer Society, 2015.

[Long *et al.*, 2015b] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Mahesh *et al.*, 2022] Ragini Bal Mahesh, Jorge Leandro, and Qing Lin. Physics informed neural network for spatial-temporal flood forecasting. In *Climate Change and Water Security: Select Proceedings of VCDRR 2021*, pages 77–91. Springer, 2022.

[Mateo-Garcia *et al.*, 2021] Gonzalo Mateo-Garcia, Joshua Veitch-Michaelis, Lewis Smith, Silviu Vlad Oprea, Guy Schumann, Yarin Gal, Atılım Güneş Baydin, and Dietmar Backes. Towards global flood mapping onboard low cost satellites with machine learning. *Scientific reports*, 11(1):1–12, 2021.

[Matgen *et al.*, 2020] Patrick Matgen, Sandro Martinis, Wolfgang Wagner, Vahid Freeman, Peter Zeil, Niall McCormick, et al. Feasibility assessment of an automated, global, satellite-based flood-monitoring product for the copernicus emergency management service. *Luxembourg: Publications Office of the European Union*, 2020.

[NCSU, 2023] NCSU. NCSU Libraries. https://www.lib.ncsu.edu/gis/elevation, 2023.

[Nearing *et al.*, 2020] Grey Nearing, Frederik Kratzert, Daniel Klotz, Pieter-Jan Hoedt, Günter Klambauer, Sepp Hochreiter, Hoshin Gupta, Sella Nevo, and Yossi Matias. A deep learning architecture for conservative dynamical systems: Application to rainfall-runoff modeling. In *AI for Earth Sciences Workshop at NeurIPS*, 2020.

[NOAA, 2016] NOAA. Hurricane matthew imagery. https://geodesy.noaa.gov/storm_archive/storms/matthew/, 2016. (Accessed on 08/02/2023).

[NOAA, 2017] NOAA. Hurricane harvey imagery. https://storms.ngs.noaa.gov/storms/harvey/index.html#7/28.411/-96.691, 2017. (Accessed on 08/02/2023).

[Oddo and Bolten, 2019] PC Oddo and JD Bolten. The value of near real-time earth observations for improved flood disaster response. frontiers in environmental science, 2019.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[Rudner *et al.*, 2019] Tim GJ Rudner, Marc Rußwurm, Jakub Fil, Ramona Pelich, Benjamin Bischke, Veronika Kopačková, and Piotr Biliński. Multi3net: segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 702–709, 2019.

[Sainju *et al.*, 2020] Arpan Man Sainju, Wenchong He, Zhe Jiang, and Da Yan. Spatial classification with limited observations based on physics-aware structural constraint. In *AAAI*, pages 898–905. AAAI Press, 2020.

[Sainju *et al.*, 2022] Arpan Man Sainju, Wenchong He, and Zhe Jiang. A hidden markov contour tree model for spatial structured prediction. *IEEE Trans. Knowl. Data Eng.*, 34(4):1530–1543, 2022.

[Sami *et al.*, 2024] Mirza Tanzim Sami, Da Yan, Saugat Adhikari, Lyuheng Yuan, Jiao Han, Zhe Jiang, Jalal Khalil, and Yang Zhou. Evanet: Elevation-guided flood extent mapping on earth imagery. *arXiv preprint arXiv:2404.17917*, 2024.

[Sun *et al.*, 2023] Alexander Y. Sun, Zhi Li, Wonhyun Lee, Qixing Huang, Bridget R. Scanlon, and Clint Dawson. Rapid flood inundation forecast using fourier neural operator. In *ICCV*, pages 3735–3741. IEEE, 2023.

[USGS, 2023] USGS. Tnm download v2. https://apps.nationalmap.gov/downloader/, 2023. (Accessed on 08/02/2023).

[Wahlstrom *et al.*, 2015] Margareta Wahlstrom, Debarati Guha-Sapir, et al. The human cost of weather-related disasters 1995–2015. *Geneva, Switzerland: UNISDR*, 2015.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*, pages 1907–1913. ijcai.org, 2019.

[Xie *et al.*, 2018] Miao Xie, Zhe Jiang, and Arpan Man Sainju. Geographical hidden markov tree for flood extent mapping. In *KDD*, pages 2545–2554. ACM, 2018.