

Article

# Flood Uncertainty Estimation Using Deep Ensembles

Priyanka Chaudhary <sup>1,\*</sup>, João P. Leitão <sup>2</sup>, Tabea Donauer <sup>1</sup>, Stefano D’Aronco <sup>1</sup>, Nathanaël Perraudin <sup>3</sup>, Guillaume Obozinski <sup>3</sup>, Fernando Perez-Cruz <sup>3,4</sup>, Konrad Schindler <sup>1</sup>, Jan Dirk Wegner <sup>1,5</sup> and Stefania Russo <sup>1</sup>

<sup>1</sup> EcoVision Lab, Photogrammetry and Remote Sensing Group, ETH Zürich, 8049 Zurich, Switzerland

<sup>2</sup> Department Urban Water Management, Eawag-Swiss Federal Institute of Aquatic Science and Technology, 8600 Dubendorf, Switzerland

<sup>3</sup> Swiss Data Science Center, 8092 Zurich, Switzerland

<sup>4</sup> Institute of Machine Learning, Department of Computer Science, ETH Zürich, 8092 Zurich, Switzerland

<sup>5</sup> Institute for Computational Science, University of Zurich, 8057 Zurich, Switzerland

\* Correspondence: pchaudha@ethz.ch

**Abstract:** We propose a probabilistic deep learning approach for the prediction of maximum water depth hazard maps at high spatial resolutions, which assigns well-calibrated uncertainty estimates to every predicted water depth. Efficient, accurate, and trustworthy methods for urban flood management have become increasingly important due to higher rainfall intensity caused by climate change, the expansion of cities, and changes in land use. While physically based flood models can provide reliable forecasts for water depth at every location of a catchment, their high computational burden is hindering their application to large urban areas at high spatial resolution. While deep learning models have been used to address this issue, a disadvantage is that they are often perceived as “black-box” models and are overconfident about their predictions, therefore decreasing their reliability. Our deep learning model learns the underlying phenomena *a priori* from simulated hydrodynamic data, obviating the need for manual parameter setting for every new rainfall event at test time. The only inputs needed at the test time are a rainfall forecast and parameters of the terrain such as a digital elevation model to predict the maximum water depth with uncertainty estimates for complete rainfall events. We validate the accuracy and generalisation capabilities of our approach through experiments on a dataset consisting of catchments within Switzerland and Portugal and 18 rainfall patterns. Our method produces flood hazard maps at 1 m resolution and achieves mean absolute errors as low as 21 cm for extreme flood cases with water above 1 m. Most importantly, we demonstrate that our approach is able to provide an uncertainty estimate for every water depth within the predicted hazard map, thus increasing the model’s trustworthiness during flooding events.



**Citation:** Chaudhary, P.; Leitão, J.P.; Donauer, T.; D’Aronco, S.; Perraudin, N.; Obozinski, G.; Perez-Cruz, F.; Schindler, K.; Wegner, J.D.; Russo, S. Flood Uncertainty Estimation Using Deep Ensembles. *Water* **2022**, *14*, 2980. <https://doi.org/10.3390/w14192980>

Academic Editor: Paolo Mignosa

Received: 24 August 2022

Accepted: 15 September 2022

Published: 22 September 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** flood prediction; deep learning; uncertainty estimation

## 1. Introduction

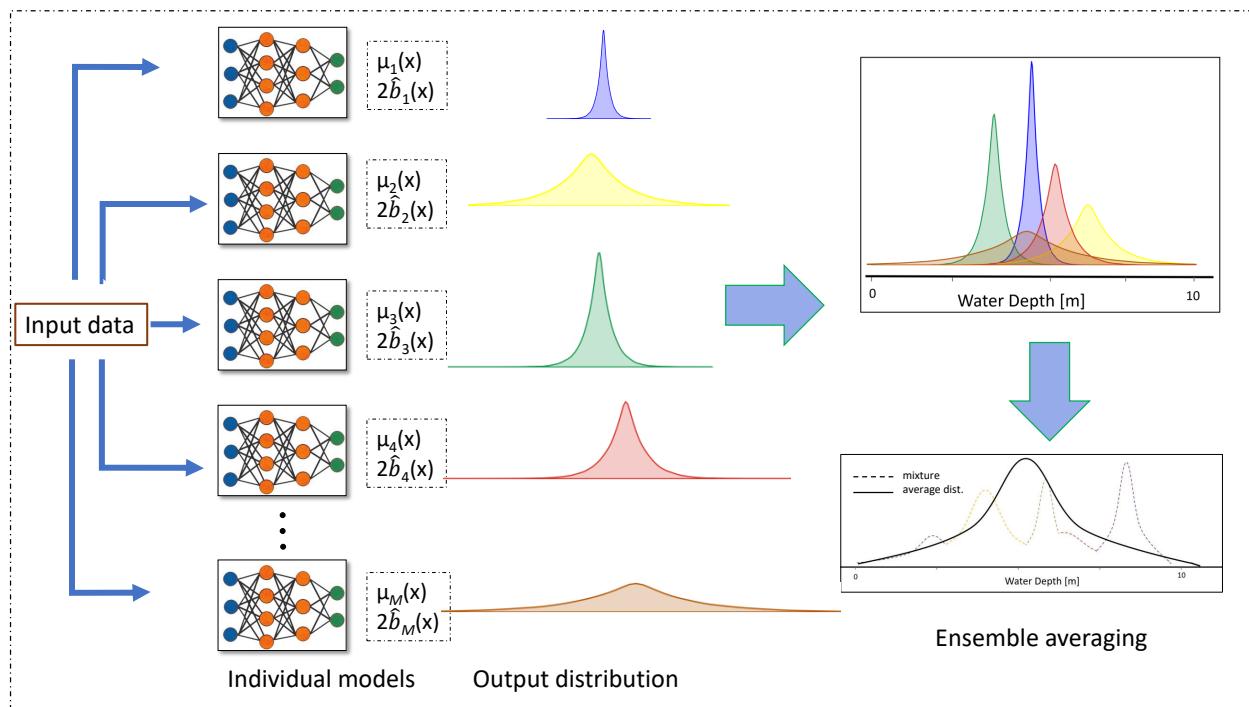
The frequency of urban floods and their impact are escalating at an unprecedented rate due to changes in land use, increase in population, and climate change [1,2]. Floods in urban areas occur when natural water sources or drainage systems in cities lack the capacity to convey excess water (runoff) caused by intense rainfall events (pluvial flooding) [3]. It is a tremendous challenge for urban pluvial flood risk management, which can involve various actions ranging from systematic offline analyses of different rainfall scenarios [4] to real-time flood predictions [5]. Rapid generation of hazard maps is key [6] for supporting flood alert systems and protecting the population.

Surface water flows can be simulated with hydrodynamic models that solve equations derived from the laws of fluid dynamics. Such models can simulate point-wise flood occurrence in urban areas with different degrees of complexity using rainfall forecasts from weather services [7] coupled with digital elevation models (DEMs) or digital surface models (DSMs). However, hydrodynamic simulations are not well resolved in space and time. Their high computational burden hinders their application to large

areas with high spatial resolution [7]. In practice, this limits their use to small areas at a resolution of  $\leq 10$  m [8]. Urban areas require a higher spatial resolution of 1–5 m for meaningful flood predictions because of the large number of urban structures that influence the surface flow [8,9], further constraining their applicability at a useful scale in practice.

In contrast to the slow and computationally intensive hydrodynamic models, machine learning models provide a powerful way to obtain fast predictions at a large scale in case of flooding. They implicitly learn the underlying hydrodynamic phenomena *a priori* from large amounts of offline data [10–12]. Modern deep learning models can learn both linear and non-linear relationships in flood events directly from data [13–15], turning them into a powerful tool for urban flood prediction. A disadvantage is that deep learning models usually do not provide uncertainty estimates. This can jeopardise their applicability in practice: policymakers need a sense of how trustworthy the models' results are [13] to implement meaningful flood risk management and evacuation plans. It is therefore crucial that deep learning models provide well-calibrated uncertainty estimates along with accurate water depth predictions densely at high spatial resolution. Here, the term *well-calibrated* refers to uncertainty estimates that correspond well with the model error. In this work, we take a first step in this direction and propose a probabilistic deep learning framework to rapidly predict maximum water depth in urban flooding scenarios together with uncertainty estimates for every water depth output.

We implement a probabilistic deep learning approach which we name *Deep Flood* based on an ensemble of deep neural networks (Figure 1) that can be viewed as an approximation for Bayesian marginalisation [16,17]. This allows estimating two types of uncertainty: *aleatoric* and *epistemic*, where the first describes the stochasticity inherent to the data and the latter the uncertainty due to modelling approximations and errors, including uncertainty in estimating the parameters of the model [18]. All deep learning models have exactly the same network architecture and loss function but are initialised with different random weights for training, which allows for their interpretation as samples from an approximate posterior distribution. At inference time, we run the test samples through each model separately and compute their individual standard deviations, which are then combined into one, final uncertainty estimate per water depth. We validate the proposed method on a dataset with two catchments in Switzerland and one in Portugal, for 18 different rainfall patterns. Our *Deep Flood* approach is able to provide 1 m resolution hazard water depth maps and dense, well-calibrated uncertainty estimates per pixel for every predicted maximum water depth, eventually increasing the trustworthiness of the system for downstream tasks such as risk assessment and evacuation plans.



**Figure 1.** High-level illustration of our proposed approach, *Deep Flood*, for prediction of flood hazard maps. The input data containing parameters of the terrain and rainfall forecast is fed to the deep learning model during the training phase. All models are identical but trained independently starting from random initial weights. Each model outputs a pixel-wise maximum water depth  $\mu$  and scale parameter  $2b$ . The outputs are then combined providing the final estimate of the water depth map together with its associated predictive uncertainty.

## 2. Related Work

### 2.1. Flood Estimation

Robust and accurate flood predictions support water resource management strategies, policy suggestions and analysis, and additional evacuation modelling [19]. Hydrological models have been in use for a long time to predict water-related events, such as storms [20,21], rainfall/runoff [22,23], shallow water conditions [24], hydraulic models of flow [25,26], and further global circulation phenomena [27], including the consequences of atmosphere, ocean, and floods [28]. Hydrological physically based models can be categorised based on the dimensionality of the flow representation. In the specific case of flood models, they can be categorised as one- (1D), two- (2D), or three-dimensional (3D): 1D models simulate the flow along a centreline and are mainly applicable in confined situations such as in a channel or in a pipe [8]; in 2D models [29–31] the flow is represented as a 2D field: here the assumption is that, compared with the other spatial dimensions, the water depth is shallow; 3D models are required wherever vertical features are crucial, e.g., for studying dam ruptures and tsunamis [32,33]. In order to be developed, such models require expertise and in-depth knowledge about hydrological parameters [34], in addition to intensive computational costs. To this end, a way to speed them up is to reduce the complexity of their associated differential equations. For example, this can be performed by disregarding the inertial and advection terms of the momentum equation [29,30] or by decoupling the flow into orthogonal directions [7,31,35].

Another group of models for flood estimation are the so-called physically simplified approaches [36,37] which predict flood occurrence through simplified hydraulic notions. As a result, these models provide predictions with much less computational costs, with a small loss of accuracy [38,39]. These models are suitable whenever some flow properties, for instance, velocity, are not necessary. The cellular-automata flood models [40] have recently received considerable attention. Instead of solving complex shallow water equations, these

models are able to carry out faster flood modelling by using simple transition rules and a weight-based system. The transition rules work by predicting the new state (e.g., the amount of water) of a cell based on the cell's previous state and its neighbours. Since the transition is applied on all raster cells in parallel, such models can benefit from GPU parallelism, greatly reducing the simulation time [40].

In addition to the models described above, data-driven machine learning models have recently gained more popularity in flood modelling. Various machine learning techniques, such as logistic regression [41,42], artificial neural networks [11,12], support vector machines [43–45], and random forests [46] have been used for urban flood risk prediction during the last decade. Notably, within the machine learning umbrella, deep learning approaches are being presented more often in recent years since they can perform complex tasks without requiring extensive feature engineering. They work well with unstructured data and can support parallel and distributed algorithms thus gaining computational efficiency. In [47], the authors used a recurrent neural network to forecast the two-step-ahead river stream flow based on rainfall measurements from several gauge stations. This method was later extended to multiple-step-ahead using an expandable neural network architecture in [48]. As the traditional hydrological models have the problem of performance degradation when calibrated for multiple basins together instead of a single basin, the authors in [49] trained a single long short-term memory model on 531 basins using meteorological time series data and static catchment attributes and were able to significantly improve performance. Their approach not only significantly outperformed hydrological models that were calibrated regionally but also achieved better performance than hydrological models that were calibrated for each basin individually. For flash flood susceptibility mapping tasks, Tien Bui et al. [50] used a deep neural network model to construct a classification boundary that could determine the susceptibility to flash floods for areas within the studied region. The model took as input a flash flood's influencing factor which was selected based on a literature review and available parameters for flash flood modelling. Finally, Guo et al. [51] showed that the grid-based fluid simulations can also be approximated accurately with a convolutional neural network (CNN), which predicted the velocity field of the steady flow from discretised input geometries.

## 2.2. Uncertainty Estimation in Deep Learning

The ability of deep learning to provide useful flood predictions is clear, but assessing the reliability of such predictions remains a challenge. In fact, estimators obtained from regression in its simplest form only output a prediction without any form of uncertainty measure. In this work, our aim is a more complex form of regression which does not only predict the value of interest but also estimate its uncertainty. The uncertainty estimation should be calibrated, i.e., it should correspond to the “true” expected error of the predictor [52]. Trained deep neural networks should be able to provide a prediction coupled with its uncertainty, whereas “unsure” predictions are associated with high uncertainty. It becomes clear that such behaviour is especially desirable for real-world decision-making systems where downstream actions are taken based on such predictions [53]. Specifically for our flood prediction task, we expect an uncertainty value (e.g., measured in metres) associated with its water-depth predictions for each coordinate of the catchment map.

In the past years, different approaches have been proposed for uncertainty quantification in deep learning. In [54], the authors provided single-model computation of aleatoric and epistemic uncertainty for deep neural networks. To estimate aleatoric uncertainty they proposed simultaneous quantile regression, a loss function to learn all the conditional quantiles of a given target variable. These quantiles were then used to compute well-calibrated prediction intervals. For epistemic uncertainty evaluation, they used orthonormal certificates, a collection of diverse non-constant functions mapping out-of-distribution examples to non-zero values, which indicated epistemic uncertainty. Another approach [55–57], called deterministic uncertainty quantification, was built upon the idea of radial basis function networks, consisting of a deep neural network model and a set of feature vectors. These corresponded to the different classes or centroids. A prediction was made by measur-

ing the distance between the feature vector provided by the model and the centroids. The uncertainty was modelled by computing the distance between the model prediction and the closest centroid. A significant, rapidly increasing family of models is the Dirichlet-based uncertainty (DBU) family [58]. The benefit of DBU models is to provide efficient uncertainty estimates at test time in a single forward pass by directly predicting the parameters of a Dirichlet distribution over categorical probability distributions. DBU models provide both aleatoric and epistemic uncertainty estimates which can be quantified from Dirichlet distributions using different uncertainty measures such as differential entropy, mutual information, or pseudo-counts.

Despite the successful examples presented above, the majority of research revolves around Bayesian formalism for uncertainty estimation [59]. A Bayesian neural network (BNN) model is a regular neural network with a prior placed on the weights and biases [18]. Using the prior and given training data, the posterior distribution over the parameters can be then computed (approximated). The obtained distribution is then employed to compute the posterior distribution of predictions, from which a mean estimate and its uncertainty could be extracted. As the exact Bayesian inference is computationally intractable for neural networks, several approximations have been proposed [59]. Mackay, in [60], suggested a Laplace approximation of the posterior but with limited performance. Markov chain Monte Carlo (MCMC) methods can be used to sample the posterior distribution over the set of model parameters. The main limitation of MCMC methods is their prohibitive storage cost. As the posterior distribution is represented by samples of deep neural network parameters, thousands of samples need to be stored each time. Considering that modern deep neural networks have millions or more parameters, MCMC methods would not be feasible in this case. Variational inference is an alternative Bayesian method which approximates the posterior distribution by a tractable variational distribution  $q_\theta(W)$  indexed by a variational parameter  $\theta$ . The optimal variational distribution can be achieved by minimising the Kullback–Leibler divergence between  $q_\theta(W)$  and the true posterior  $p(W|D)$ . In contrast to MCMC, variational inference methods are more time and space efficient but the gap between the approximate posterior and the true posterior degenerates the model performance [61]. To tackle the limitation of such a slow and computationally expensive method, Monte Carlo (MC) dropout [62] was introduced using dropout [63] as a regularisation term to compute the predictive uncertainty [53].

In our approach, we approximate the posterior over the BNN model with a deep ensemble. In [16,17,64], the authors showed that deep ensembles can be seen as an approximate approach to Bayesian marginalisation, which selects for functional diversity by representing multiple basins of attraction in the posterior. The authors also proved that deep ensembles can provide a better approximation of the Bayesian predictive distribution than standard approaches and discuss that they also outperform some particular approaches to BNNs. Additionally, ensembles are reported to provide better performance in their predictions and reliability of the computed uncertainty estimates [65,66] and have become a standard for uncertainty estimation [67].

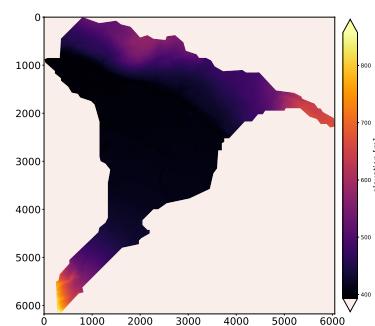
### 3. Datasets

In this work we use a dataset containing two different catchment areas located within Switzerland, using raster data with 1 m spatial grid sampling. The data were simulated using the CADDIES cellular-automata flood model [40] based on [68]. As mentioned earlier, here transition rules are employed through a weight-based system to determine the flow movement, avoiding solving complex and computationally expensive equations. The benefit of using the CADDIES model is that it achieves a faster computational performance when compared with physically based models that solve the shallow-water equations without a significant sacrifice of accuracy [40]. In this way, the CADDIES simulator allows us to conveniently verify our approach with relatively low computational effort. Additionally, we employed a dataset related to a catchment area located in Coimbra, Portugal, using raster data with a spatial grid resolution of 1 m. This dataset was generated using the Infoworks ICM software by Innovyze [69], which is a physically based model that also

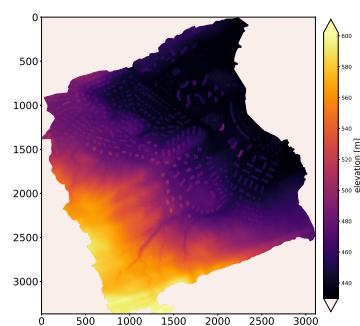
considers pipes and surface flow in urban areas. The characteristics of catchments are summarised in Table 1 and an illustration of the catchments is provided in Figure 2.

**Table 1.** Summary of the catchments' characteristics used in this study.

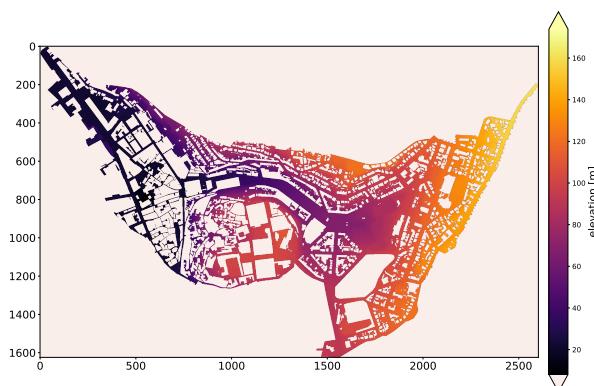
	Area (km <sup>2</sup> )	Minimum Elevation (m)	Maximum Elevation (m)	Maximum Slope (rise/run)
Zurich	37.36	393.7	857.3	5.77
Lucerne	10.48	430.69	602.11	24.36
Portugal	4.23	8.81	173.94	14.77



(a) Zurich



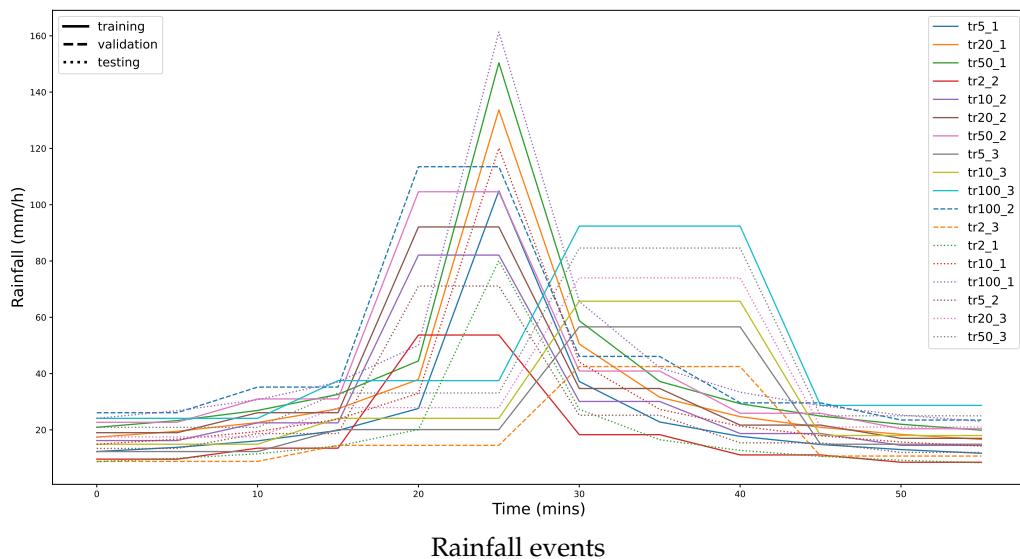
(b) Lucerne



(c) Portugal

**Figure 2.** Illustration of the Zurich, Lucerne, and Portugal catchments in our dataset. The colour represents the terrain elevation data for the catchments.

To conduct the simulations, we selected 18 different one-hour rainfall hyetographs representing different rainfall intensities, with approximately 2, 5, 10, 20, 50, and 100 year return periods, and with different time step discretisation (5 min, 10 min, and 15 min), as shown in Figure 3.



**Figure 3.** The illustration of our hyetographs used for simulations. The first rainfall number corresponds to the return period, while the second to the different events within the same return period in the legend shown in the upper right side of the figure. The different line types, as shown in the upper left legend, refer to the allocation of rainfall events for training, validation, and testing set as described in Table 2.

For both CADDIES and Infoworks ICM, the simulated data (which we consider our “ground truth”) consisted of the maximum water depth for each pair of catchment and hyetograph. No post-processing was applied in order to keep fidelity to the flood simulator we aim to replicate. Predicting maximum water depths maps can be related to anticipating the worst-case scenario (i.e., a hazard map) given a catchment area and a rain forecast. For this specific application, it is clear that providing an uncertainty estimate for each predicted cell in the area is of extreme importance for downstream tasks such as decision making in evacuation plans.

We computed 11 input terrain features as input data for our deep learning model, which are then concatenated as multi-channel images:

- the catchment’s DEM, representing the terrain elevation;
- a spatial differential DEM ( $DEM_{diff}$ ) comprising of four channels. A DEM can be viewed as a 2D grid whose adjacent columns ( $c$ ) or rows ( $r$ ) can be subtracted in four directions: rightward, leftward, downward, and upward. The  $DEM_{diff}$  was obtained using the following equations:

$$DEM_{diff}(1) = c^{(i+1)} - c^{(i)} \quad (\text{Rightward direction})$$

$$DEM_{diff}(2) = c^{(i)} - c^{(i+1)} \quad (\text{Leftward direction})$$

$$DEM_{diff}(3) = r^{(j+1)} - r^{(j)} \quad (\text{Downward direction})$$

$$DEM_{diff}(4) = r^{(j)} - r^{(j+1)} \quad (\text{Upward direction})$$

where  $i$  and  $j$  are the row and column index of the DEM.

- the topographic index as the logarithm of the ratio between flow accumulation and local slope. Flow accumulation is related to the upstream drainage area of each raster cell and is computed from the raw DEM using the r.terraflow module in QGIS [70]. The topographic index is commonly used in hydrology as a steady-state wetness index;
- slope, defined as the measure of the rate of change of elevation in the direction of steepest descent. It reflects the steepness of the terrain and is the means by which gravity induces the flow of water;

- aspect as the orientation of the line of steepest descent. Alternatively, it can be defined as for a selected point on a surface, aspect is the direction in which slope is maximised [71,72];
- three curvature attributes computed on second derivatives. Curvature combines the profile curvature, which measures the rate of change of slope down a flow line, and plan curvature, which calculates the rate of change of aspect along a contour [71,73].

In addition to the features described above, we also add rainfall intensities as input channels to the model. The duration of rainfall events used is one hour which results in twelve channels. Note that the selection of these features was inspired by the work of [7]. Compared to [7], we found that adding  $DEM_{diff}$  and the topographic index, as well as the complete rainfall forecast, was beneficial for the performance and led to faster convergence of the model during training.

#### 4. Methodology

##### 4.1. Deep Learning Model

We employ a convolutional neural network (CNN) architecture to predict the maximum flood height map at every location of the DEM. Our model design is inspired by the U-Net architecture, which was originally presented by Ronneberger et al. [74] for biomedical segmentation applications. The architecture is depicted in Appendix A, Figure A1, and consists of a contracting (encoder) and an expansive (decoder) path designed to encode and decode the input to produce an output of the same resolution as the input. The encoder consists of the repeated application of two  $3 \times 3$  convolutions. Each one is followed by a rectified linear unit (ReLU) and batch normalisation and a max pooling operation of  $2 \times 2$  with stride 2 for downsampling and reducing the spatial dimensions. At each downsampling step, the number of feature channels is doubled and the spatial dimensions are reduced by half. Every step in the decoder path consists of an upsampling of the feature map followed by a  $2 \times 2$  transpose convolution, which halves the number of feature channels. Additionally, a concatenation with the corresponding high-resolution feature map from the contracting path is performed, and a successive convolutional layer is then used to learn to assemble a more precise output based on this information. An important feature of U-Net architecture is that, in the upsampling part, we have a large number of feature channels which allows the model to propagate context information to a higher resolution layer. As a result, the expansive path on the right side is nearly symmetric to the contracting path on the left which gives a u-shape to the architecture. To adapt the architecture to our application, we have added two heads for the regression of  $[\mu, \log(2\hat{b})]$  as described later in Section 4.2. Each head consists of two  $1 \times 1$  convolutions with stride 1 with a ReLU layer in between.

##### 4.2. Predictive Uncertainty Estimation

###### 4.2.1. Epistemic Uncertainty

Epistemic uncertainty is modelled by placing a prior distribution over a model's weights, and then expressing the variation of such weights given some data [18]. In regression we predict a single continuous target variable  $y$  from a given dataset of inputs  $\mathbf{X} = \{x_1, x_2, x_3, \dots, x_N\}$  with labels  $\mathbf{Y} = \{y_1, y_2, y_3, \dots, y_N\}$ . We denote the model as  $f^W$ , with output  $f^W(\mathbf{x})$ . We would like to find the parameters  $\mathbf{W}$  of a function  $f^W(x)$  that is likely to have generated our outputs. Taking inspiration from [18], in this work we place a Laplacian prior distribution on the model weights. This distribution represents our prior belief as to which parameters are likely to have generated our data before we observe any input of the dataset. After we observe some data, the prior distribution will be transformed to capture the more or less likely parameters. For this, we further require to define a likelihood distribution, for which we again chose a Laplacian. We then look for

the posterior distribution which captures the set of plausible model parameters given the dataset  $\mathbf{X}, \mathbf{Y}$  by invoking Bayes' theorem:

$$p(\mathbf{W}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})}{p(\mathbf{Y}|\mathbf{X})} \quad (1)$$

A key component to evaluate that posterior is the normalisation constant  $p(\mathbf{Y}|\mathbf{X})$ , the so-called model evidence, which can be obtained by integrating out the parameter distribution. Note that this is equivalent to computing the posterior distribution over the labels, i.e., performing inference for a new input  $\mathbf{x}^*$  sampled from the same distribution:

$$p(\mathbf{Y}|\mathbf{X}) = p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})d\mathbf{W} \quad (2)$$

The marginalisation can be resolved analytically for simple models such as Bayesian linear regression when the prior is conjugate to the likelihood. However, for complex architectures such as deep models, this marginalisation cannot be performed analytically. In such cases approximate techniques are required [75], which were briefly described in Section 2.2. In our approach, we approximate the posterior over the BNN model with an ensemble. Wilson and Izmailov [16] have shown that deep ensembles can be seen as a proxy for Bayesian marginalisation, which selects for functional diversity by representing multiple basins of attraction in the posterior. Using deep ensembles comprises maximum a posteriori (MAP) training of the same architecture many times starting from different random initialisations, in order to find a different local optimum. Therefore, using these models in an ensemble as an approximate Bayesian model is another way to generate a distribution over models, which can be employed to provide an estimate of the variability of the learning procedure. Performing this step corresponds to estimating an essential component of epistemic uncertainty. Instead of using a single point mass to approximate our posterior, as with classical training, we are now using multiple point masses, enabling a better approximation of the integral in Equation (2) that we are trying to solve [64]. Finally, the variance computed over their respective predictions provides an estimate of the epistemic uncertainty.

#### 4.2.2. Aleatoric Uncertainty

We can subcategorise aleatoric uncertainty into *homoscedastic uncertainty* and *heteroscedastic uncertainty*. Homoscedastic uncertainty remains constant for different inputs while heteroscedastic uncertainty is dependent upon the inputs provided to the model and learned through the training process [18,76,77]. Heteroscedastic uncertainty is notably meaningful for computer vision applications [18] and is computed by training the model to estimate the probability density function, by deriving the loss functions for a Laplace prior, as explained below. The probability density function for the Laplace distribution is given by:

$$P(x) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right)$$

where  $\mu$  is a location parameter and  $b > 0$  is a scale parameter, corresponding to the mean absolute deviation from the mean. The latter indicates how much the distribution is spread out. Therefore, the negative log-likelihood of  $\mu_i, b_i$  given a sample  $x_i$  for the Laplace distribution becomes:

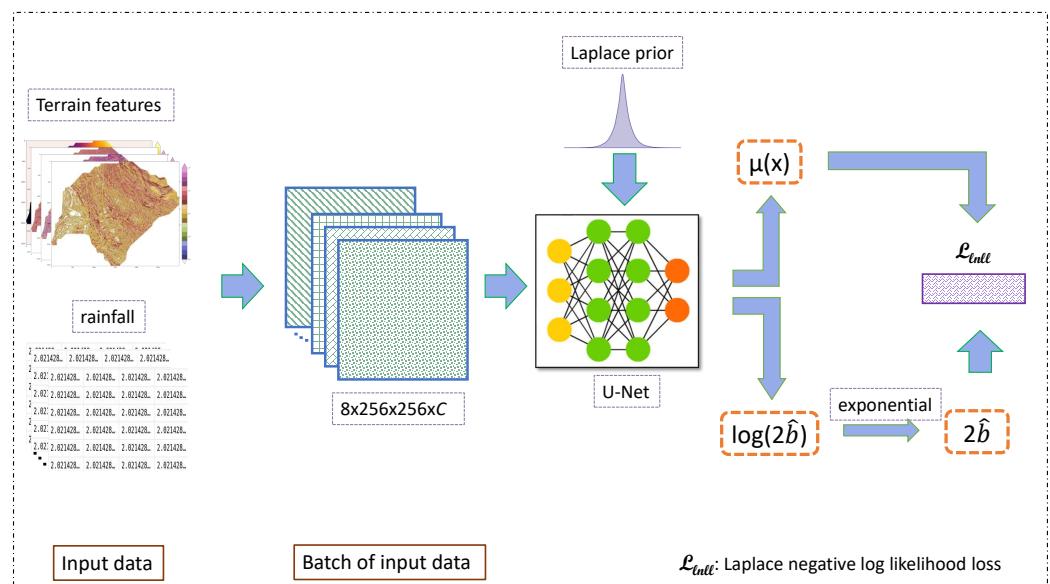
$$-\log p(\mu_i, b_i|x_i) \propto \frac{1}{b_i} |\mu_i - x_i| + \log(2b_i)$$

Note that a Gaussian likelihood could be also used for the above formulation. However, the loss resulting from a Laplace likelihood is more robust than a Gaussian likelihood, when the error is heavy-tailed and not Gaussianly distributed (see [18,78]). Therefore, in

this work, we adapt the standard Gaussian negative log-likelihood loss and formulate our loss function using the Laplacian likelihood

$$\mathcal{L}_{NLL}(\mu_1, \dots, \mu_N, \hat{b}_1, \dots, \hat{b}_N) = \frac{1}{N} \sum_{i=1}^N \frac{|\mu_i - y_i|}{\hat{b}_i} + \log 2\hat{b}_i \quad (3)$$

where  $N$  is the number of output pixels corresponding to the input image indexed by  $i$ ,  $\mu_i$  and  $\hat{b}_i$  are the model outputs at location  $i$ , and  $y_i$  is the ground truth value. Given the formulation of our loss function, labels for the uncertainty are not needed to learn uncertainty. Simply, the learning of the regression task is conducted and the scale parameter  $\hat{b}_i$  is implicitly learned from the loss function. The loss function is composed of two terms,  $\log 2\hat{b}_i$  and  $1/\hat{b}_i$ . The first term discourages the model from predicting high uncertainty for all data as a large uncertainty value increases the contribution of this term and so the loss. Similarly, the model can also learn to neglect the data or predict very low uncertainty value for data points with high error  $|\mu_i - y_i|$ , but, by doing so, the denominator  $\hat{b}_i$  will amplify the contribution of the residual and will penalise the model [18]. Our approach is summarised in Figure 4. The model takes the data input and provides two outputs  $[\mu, \log(2\hat{b})] = f^{\hat{W}}(x)$ . Note that the conversion of the second output using an exponential function guarantees that the estimated  $\hat{b}$  is positive. For numerical stability, we also add a small number  $\epsilon = 1e-7$  to the estimated  $2\hat{b}$  value during training to avoid division by zero. Lastly, since the Laplace distribution has mean  $\mu$  and variance  $2\hat{b}^2$ , this can be easily computed from the model output and will be used for the final estimation of the predictive uncertainty.



**Figure 4.** The illustration of our deep learning approach. Here, we show only one of the  $M$  models included in the ensemble. The model is trained to receive as an input the digital elevation model (DEM), terrain features and rainfall channels. The output consists of a pixel-wise maximum water depth estimation across the DEM and the scale parameter of the Laplacian distribution, through which we then compute the predictive uncertainty.

#### 4.2.3. Combining Aleatoric and Epistemic Uncertainty

We calculate the predictive uncertainty of the deep ensemble by training  $M$  different models and consider them as random samples from the distribution of models. This is performed by applying a different random initialisation of the Laplacian prior weights at the start of the training procedure. The deep ensemble is treated as a uniformly weighted

mixture model of Laplacian distributions [79] and the predictions are combined using the following formula:

$$\bar{y} = \frac{1}{M} \sum_{m=1}^M \mu_m, \quad (4)$$

while the total predictive uncertainty is approximated as:

$$\bar{\sigma} = \sqrt{\underbrace{\frac{1}{M} \sum_{m=1}^M \mu_m^2 - (\frac{1}{M} \sum_{m=1}^M \mu_m)^2}_{\text{Epistemic}}} + \sqrt{\underbrace{\frac{1}{M} \sum_{m=1}^M (2\hat{b}_m^2)}_{\text{Aleatoric}}} \quad (5)$$

for the outputs  $[\mu_m, \log(2\hat{b}_m)]$  of model  $m \in [1, \dots, M]$ . In the above equation, the first term depicts the epistemic uncertainty and the second is the aleatoric uncertainty.

#### 4.3. Model Training

Different rainfall events were allocated for training, validation, and testing of our approach, as shown in Table 2. As a common practice in machine learning applications, the validation set is not employed in the optimisation process but for hyper-parameter tuning and for loss monitoring on unseen data, and the models were only evaluated on the test sets. The data input fed to the model is described in Section 3. The output consists of hazard maps across the DEM corresponding to predicting the maximum water depth that can occur within the forecasted rainfall period. Additionally, we output the scale factor of the Laplacian distribution, through which we then compute the predictive uncertainty.

**Table 2.** Allocation of rainfall events for training, validation, and test sets.

Dataset	Rainfall Events
Training Set	tr5_1, tr20_1, tr50_1, tr2_2, tr10_2, tr20_2, tr50_2, tr5_3, tr10_3, tr100_3
Validation Set	tr100_2, tr2_3
Test Set	tr2_1, tr10_1, tr5_2, tr20_3, tr50_3, tr100_1

As the catchment size is too large to be fed to each model, we generate catchment patches. At each training iteration, a random patch of size  $256 \times 256$  with a batch size of 8 data samples is used, which are common choices in deep learning applications. We train each of the five individual model of the ensemble with prior weights initialisation using a Laplace distribution, in this way, the optimisation process can start from a different point, potentially resulting in a diverse final set of weights and performance characteristics. We then train each U-Net model in the ensemble using a Laplacian negative log-likelihood as a loss function Equation (3) for 500 epochs. We use the Adam optimisation algorithm and train the model using back-propagation, which allows the information from the objective function to flow backwards through the model in order to compute the gradient [80]. To update the model weights in the negative gradient direction we start with a learning rate of 0.001 and decrease it by a factor of ten at different epochs. These numbers were chosen empirically during the training and validation of our models, following standard hyperparameter settings in deep learning applications.

## 5. Results

### 5.1. Hazard Map

Recall that *Deep Flood* represents an independent ensemble of five U-Net models of identical architecture, as described in Section 4.3. The output of each individual U-Net in an ensemble is used to compute the final prediction and uncertainty per model. For all

the experiments, we report the mean absolute error (MAE) as a performance metric which quantifies the deviation between the predicted water depth and ground truth.

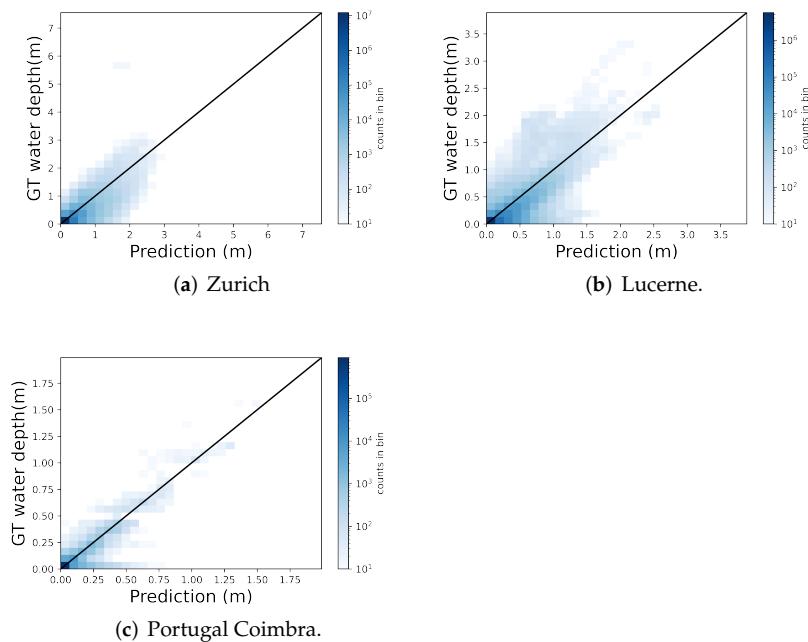
We first evaluate the predicted hazard map and consider the distribution of MAEs for the studied catchments at different ranges of water depth, i.e., all pixels, pixels with water depth  $> 10$ ,  $> 20$ ,  $> 50$ ,  $> 100$  cm in Table 3. Here, the MAEs represent the average between all the rainfall events in the test set. From Table 3, we can observe that at test time the Zurich and Lucerne catchments perform generally worse compared to the Portugal one, suggesting that the deep learning model learns more effectively the flood behaviour when presented with a complex, physically based dataset. Specifically for the Portugal dataset, the MAE values for elements  $> 50$  and  $> 100$  cm are particularly low, which makes these results attractive for hazard prediction, where deep water depths correspond to a higher risk for the population.

**Table 3.** Mean absolute errors (MAEs) in cm for *Deep Flood* models for different ranges of water depth and all rainfall events.

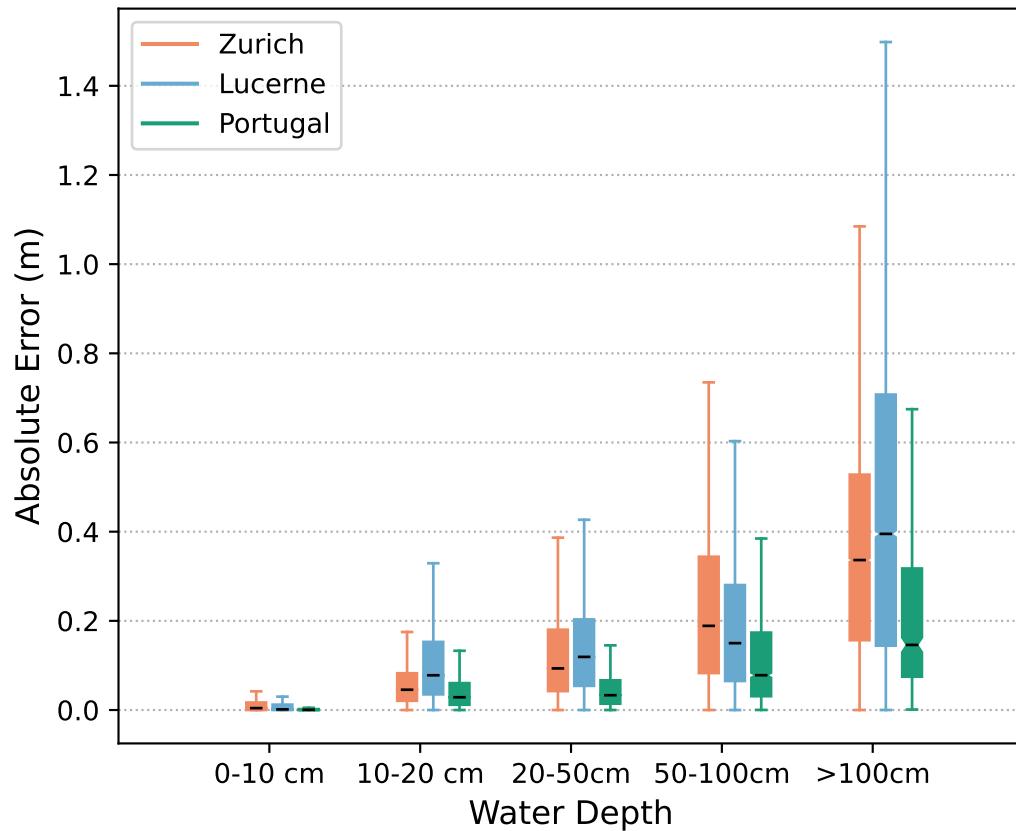
	Mean Absolute Errors (cm)				
	all	>10 cms	>20 cms	>50 cms	>100 cms
Zurich	2.70	10.97	19.09	33.87	51.87
Lucerne	3.45	18.60	25.0	42.95	69.83
Portugal	0.72	7.87	10.20	18.86	21.21

In Figure 5, we qualitatively illustrate the results by plotting the ground truth water depth against the predictions. Note that, while the results in Table 3 were combined across all rainfall events, we will now focus on only the *tr2\_1* rainfall event in the test set to avoid excessive averaging. Here, the results for the rainfall event are divided into bins and then plotted. The shade of each bin represents the count of bins for that pixel. We also ignored bins presenting less than ten elements. The black diagonal line represents the ideal prediction, where the trained model predicts exactly the same ground truth water depth value: the more the plot diverges from the diagonal, the more it indicates a decrease in accuracy. The plots agree with the results presented in Table 3. We notice from Figure 5b that the Lucerne catchment presents a wide pixel distribution plot along the upper area of the diagonal, which suggests that the prediction errors for this catchment are caused by an underestimation of the predicted water depth. This effect is opposite (and reduced) for the Zurich terrain, (Figure 5a), with the difference that the pixels here are much less spread. The Portugal catchment, on the other hand, presents a plot closer to the diagonal, indicating an improved accuracy compared to the other catchments.

To further analyse the results of this rainfall event, we divide the pixels into five different categories based on their ground truth water depth value and show the error in the form of a box plot in Figure 6, where the x-axis represents the different binned water depth categories and the y-axis the associated absolute error value. As we move from shallow water depth to deeper areas, the absolute error value increases. Specifically, the plot shows that in reality, most of the errors are very low (between a few cm to less than 20 cm median for high water depth). Few points show very high errors, especially for deep water depths. This can be explained by the behaviour of the U-Net model which, as it is based on convolution operations, tends to smooth the prediction when pixels present sudden, small changes in the water depth (which could be caused by artefacts in the DEM). This is discussed in the next paragraph.



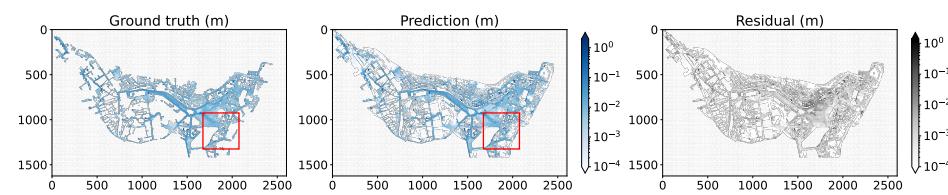
**Figure 5.** Ground-truth vs. predicted flood hazard maps for rainfall event *tr2\_1* in (a) Zurich, (b) Lucerne and (c) Portugal catchment. The black diagonal line represents the ideal prediction.



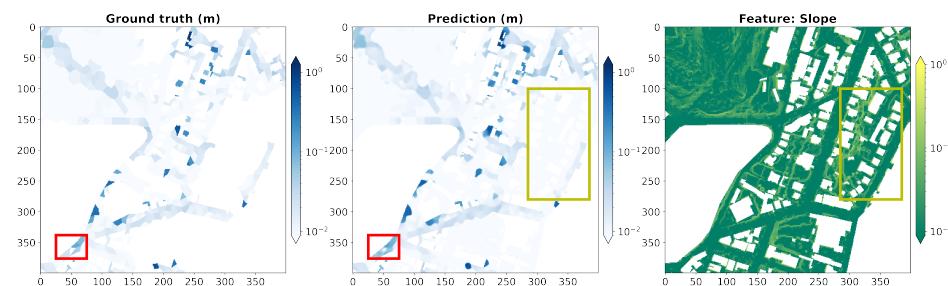
**Figure 6.** The box plot for the predicted water depth values and their associated absolute errors. The predictions are reported for our *Deep Flood* tested on steep test catchment and rainfall event *tr2\_1*.

In Figure 7a, we qualitatively show the complete reconstruction of the hazard map for the Portugal catchment. Reconstructions for the Zurich and Lucerne catchments are

shown in Appendix A, Figures A2 and A3. The red bounding box depicts the patch areas in Figure 7b, which is zoomed in. We can observe that the prediction clearly follows the ground truth water depth pattern. However, some edges of the prediction are smoothed out compared to the ground truth, especially where it presents sharp water depth changes. This might explain the presence of outliers discussed earlier and is highlighted within the red box in Figure 7b (left and centre). Additionally, we highlight within the yellow box in Figure 7b (centre) an area of the reconstruction which presents small artefacts (accounting for the few cm errors) which are most probably caused by the feature *Slope* (Figure 7b on the right). These errors were already visible in Figure 5c, in the area of the plot close to the axes' origin. Nevertheless, we found that the inclusion of the *Slope* feature still improves the results, especially for high water depths, which is the reason why we decided to keep this feature for the training of the model.



(a) On the left, the ground truth maximum water depth for Portugal catchment for rainfall event *tr2\_1*. In the middle, the complete predicted reconstruction performed by our *Deep Flow*. On the right, the residual of ground truth and predicted reconstruction for the same area.

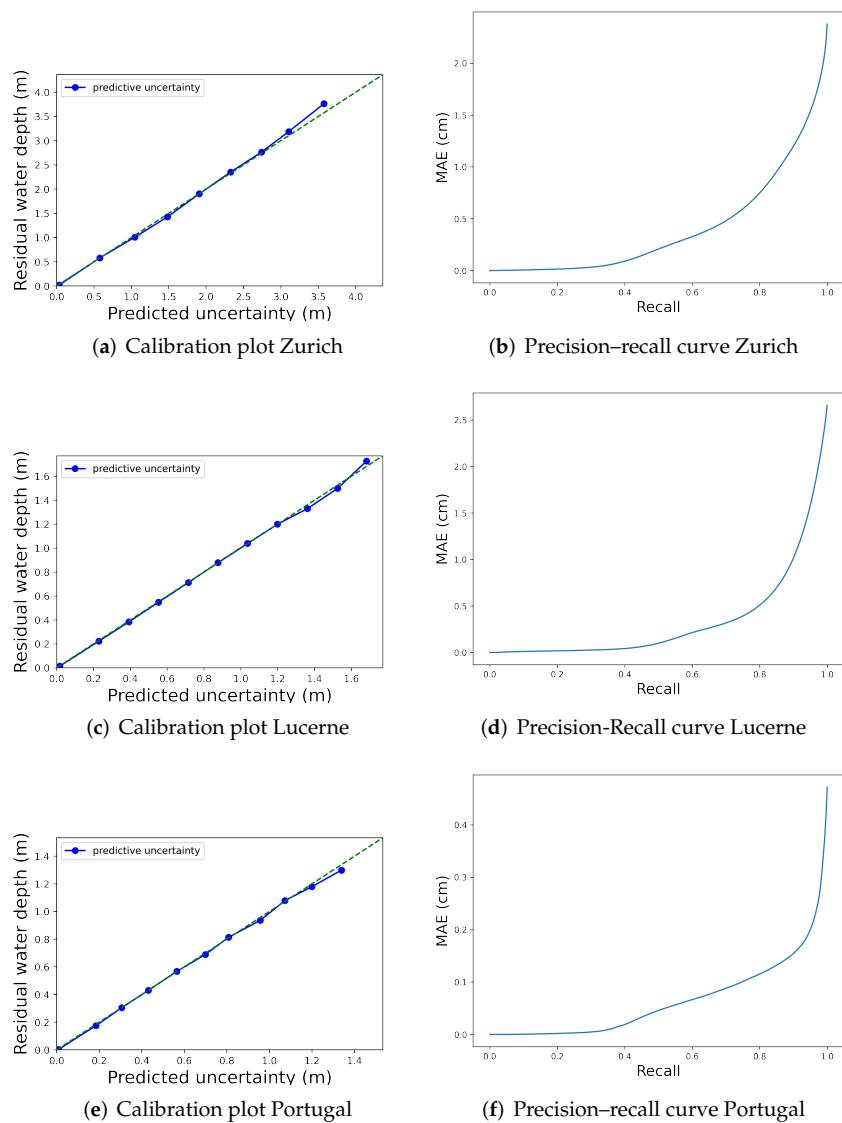


(b) On the left, one enlargement area of the ground truth. In the centre, the same area predicted by our *Deep Flow*. On the right, the feature *Slope* for the same area.

**Figure 7.** Ground truth maximum water depth and prediction for steep catchment at one specific time step of rainfall event *tr2\_1*, with a bounding box in red depicting the patch area shown in (b). In the bottom figure, we highlight in red the areas where the edges around water basins are smoothed out; in yellow, an area of the reconstruction which presents small artefacts.

### 5.2. Uncertainty Evaluation

Evaluation of the predictive uncertainties is a difficult task as the ground-truth or reference uncertainty estimates are unavailable. An approach to analysing the quality of our predictive uncertainties estimates is to use calibration plots by measuring the discrepancy between subjective forecasts and empirical long-run frequencies [59]. To form the calibration plot for our ensemble, we first take the mean value for each data point in our test set from the five trained U-Nets Equation (4). After that, we discretise the computed predictive uncertainty values in Equation (5) into bins. We then plot the mean estimated value of residual water depth for each bin. The more the resulting plot correlates with the identity line, the better the uncertainties are calibrated. In fact, as for regression tasks we have continuous output values, and the predictive uncertainty is expected to follow the residuals. having a well-calibrated predictive uncertainty means that we can safely decide, based on the uncertainty, whether to trust the predictions or not. Figure 8a shows the calibration of our approach.

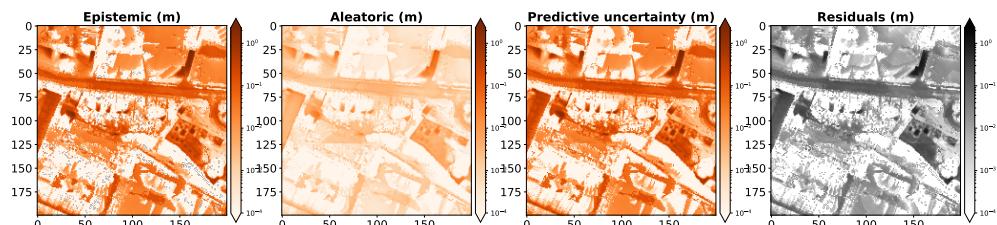


**Figure 8.** Evaluation of the predictive uncertainties predicted by Deep Flood for rainfall event *tr2\_1* on the Zurich, Lucerne, and Portugal test catchments. (a) The residuals in metres are plotted versus the grouped predicted uncertainties. (b) The test data are filtered based on the predictive uncertainty, showing a decrease in mean absolute error when high uncertainty test data points are removed.

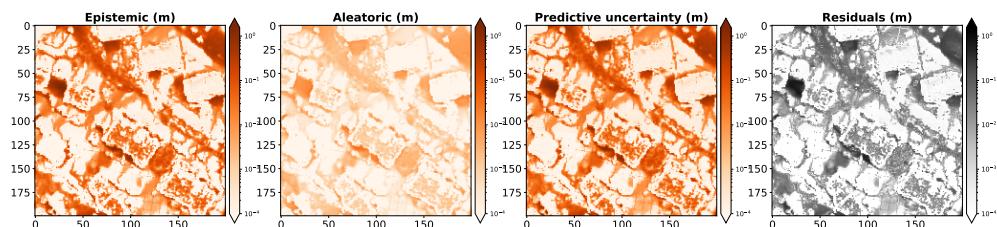
The second way to analyse the uncertainty is with precision–recall curves [18]. These curves depict in which way our model performance improves by discarding data points associated with an uncertainty larger than various percentile thresholds. To create a precision–recall curve, we sort the test samples by their predictive uncertainties and then filter out the test samples associated with the highest uncertainties. In such a way, we can reduce the overall error for the test set. We can observe this scenario in Figure 8b,d,f, where the precision (MAE) quickly decreases as the recall decreases, then flattens and becomes stable. This shows that the predictive uncertainty measurements are correlated well with MAE. For instance, we observe that for a recall between 1.0 and 0.8 in Figure 8b the MAE value decreases from 2.5 cm to around 0.5 cm. As the uncertainty is correlated with data points with high residuals, we see a sharp drop in the MAE value by removing such points. In Figure A4 in the Appendix A, we present the same results for another rainfall event, *tr100\_1*.

Finally, in Figure 9 we show the qualitative results for our predictive uncertainty by plotting the epistemic, aleatoric and predictive uncertainties together with the residuals

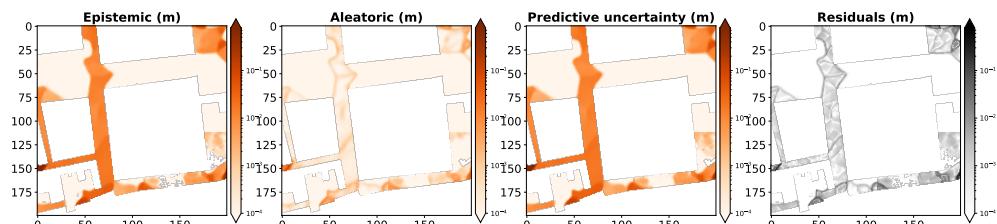
in a specific area of the Zurich, Lucerne, and Portugal catchments for the *tr2\_1* rainfall event. We observe that epistemic, aleatoric and combined predictive uncertainties follow the pattern of the residuals. For higher residual values, we also observe higher values of uncertainty. Additionally, epistemic uncertainty is much sharper around the edges and aleatoric uncertainty is more fuzzy. This behaviour is in accordance with the results such uncertainties presented in [18].



(a) Visual comparison of epistemic, aleatoric, total predictive uncertainty, and residuals for an area of Zurich test catchment and rainfall *tr2\_1*.



(b) Visual comparison of epistemic, aleatoric, total predictive uncertainty, and residuals for an area of Lucerne catchment and rainfall *tr2\_1*.



(c) Visual comparison of epistemic, aleatoric, total predictive uncertainty, and residuals for an area of Portugal test catchment and rainfall *tr2\_1*.

**Figure 9.** Qualitative evaluation of the epistemic, aleatoric and total predictive uncertainty with respect to the residual error in different catchment areas for the *tr2\_1* rainfall event. We can observe that both uncertainties follow the pattern of the residuals, with epistemic uncertainty being slightly higher around the edges of high water depth areas.

## 6. Discussion

### Comparison with previous works

In the last decades, a large body of literature has been dedicated to flood prediction and management strategies. Emerging advances in computing technologies and big data are just some of the key elements which have favoured data-driven approaches such as machine learning and deep learning, offering flexibility and scalability. Some recent works have applied deep learning for flood forecasting and prediction [14,81]. Ref. Song [81] presents a CNN model to perform a daily runoff simulation and creation method of two extra features representing the topographical and rainfall conditions. Analogously, Löwe et al. [14] trained a CNN to predict the maximum water depth maps in urban pluvial flood events using hydrodynamic flood simulations as ground truth data. Another similar work is [7], where the generalisation capabilities of CNNs as flood prediction models are explored on the same dataset used in this work. In contrast to these studies, here we presented a new framework to provide well-calibrated uncertainty estimates densely at the same spatial resolution along with the model's maximum water depth predictions.

Recently, in [13], deep learning techniques were used for predicting gauge height as well as for evaluating the associated uncertainty. The authors show that their deep learning model was more accurate than the physical and statistical models currently in use. To provide an estimation of the uncertainty for the predicted outputs, the forecasting process was repeated several times and then averaged. It is different from our approach, however, in that study, the authors did not report point-wise water depth estimates and generalisation capabilities for different rainfall events. To the best of our knowledge, our work is the first data-driven approach that is able to jointly provide a two-dimensional flood occurrence prediction at high spatial resolution together with well-calibrated uncertainty estimates.

### Advantages of the proposed approach

In this work, we proposed a probabilistic deep learning approach to predict urban flood occurrence at high spatial resolution. Especially compared to physically based flood models, a major advantage of our data-driven approach is that it provides predictions much faster than the simulator. While physically based flood simulations for very small catchments can still be computed efficiently, we expect the computational load to raise drastically with larger urban catchments. Real-time flood prediction for large and complex scenarios becomes almost infeasible [7]. Additionally, we have shown that our probabilistic deep learning approach can provide well-calibrated uncertainty predictions per water-depth prediction. This value indicates how “unsure” the model is for each prediction. We believe that those findings advocate for deep learning models as a promising alternative to physics-based simulations for flood prediction in practice: the more such systems are reliable and trustworthy, the better they can be used to implement meaningful flood risk management and evacuation plans.

### Predictive uncertainty estimation using deep ensembles

There is an ongoing debate in the literature on whether to place deep ensembles within the family of Bayesian approaches or rather view them as an alternative, outside the Bayesian framework. In this work, we have followed the method proposed by [18], who view deep ensembles as an approximation of Bayesian marginalisation, therefore placing them within the family of Bayesian methods. This interpretation is supported by [16], who argue that deep ensembles are effective at providing approximate Bayesian marginalisation as well as [17], who explicitly identify deep ensembles in the framework of Bayesian deep learning. Likewise, the authors of [64] state that deep ensembles should not be seen as an alternative approach outside the family of Bayesian methods, but as approximate Bayesian marginalisation. Finally, Izmailov et al. [82] discuss that deep ensembles provide better Bayesian than standard approximate inference procedures. On the other hand, Lakshminarayanan et al. [59] describe deep ensembles as an alternative to Bayesian neural networks thus placing them outside the Bayesian framework. Ovadia et al. [83] include deep ensembles within the non-Bayesian methods together with bootstrapping, despite showing how well ensembles perform empirically [84]. Overall, we found that despite the ongoing debate on where to theoretically place deep ensembles with respect to the Bayesian framework, they do work well empirically for calibrating uncertainties as demonstrated by our experimental evaluation.

### Limitations of the proposed approach and future works

In Figure 6 we have shown the box plots for the predicted water depth values and their associated absolute errors. While most of the errors are very low, it can be noted that some outliers go up to 150 cm. This is explained by the convolution operations in our deep learning model which tend to smooth the predictions when the pixels present sudden changes in the water depth. This drawback can negatively alter the prediction for such important and dangerous sharp changes and could be addressed in future works by testing different types of deep learning architectures.

Another strategy worth investigating for reducing severe outliers would be adding physical constraints in the design and training of the deep learning architecture. For

example, hydraulic laws could be incorporated in the loss of the model such that training in an end-to-end fashion would simultaneously constrain the gradient flow to physically plausible values and improve the quality of the predictions.

An interesting research question for future work would be to include the intrinsic uncertainty of the rainfall forecast in the predictions i.e., determining how such uncertainty propagates to the outputs of the model, as well as testing the results on real-world data.

## 7. Conclusions

We have presented *Deep Flood*, a new probabilistic deep learning approach based on deep ensembles for dense flood prediction at high spatial resolution. *Deep Flood* predicts accurate water depths and assigns well-calibrated uncertainty estimates to every predicted water depth. Our approach uses terrain features of a catchment and a rainfall forecast as input to provide a water depth map. Through experiments over three catchments and eighteen rainfall patterns, we have shown that the deep learning model is able to accurately predict maximum water depth maps over the full rainfall events at 1 m resolution in large catchment areas. Importantly, our approach also generalises well to new rainfall events with mean absolute errors of as low as 21 cm for the extreme, catastrophic cases where flooding is above 1 m. For the first time in flood prediction, our approach also returns pixel-wise well-calibrated uncertainty estimates for every predicted water depth. Quantifying the predictive uncertainty increases the trustworthiness of the system, which translates into better downstream tasks such as risk assessment and evacuation plans.

**Author Contributions:** F.P.-C.: Conceptualisation, Data curation, Investigation, Methodology, Software, Visualisation, Writing—original draft. J.P.L.: Conceptualisation, Resources, Supervision, Writing—review and editing. T.D.: Data curation. S.D.: Conceptualisation, Software. N.P.: Writing—review and editing. G.O.: Funding acquisition, Writing—review and editing. F.P.-C.: Funding acquisition, Writing—review and editing. K.S.: Conceptualisation, Supervision, Writing—review and editing. J.D.W.: Conceptualisation, Supervision, Writing—review and editing. S.R.: Conceptualisation, Software, Validation, Methodology, Visualisation, Supervision, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

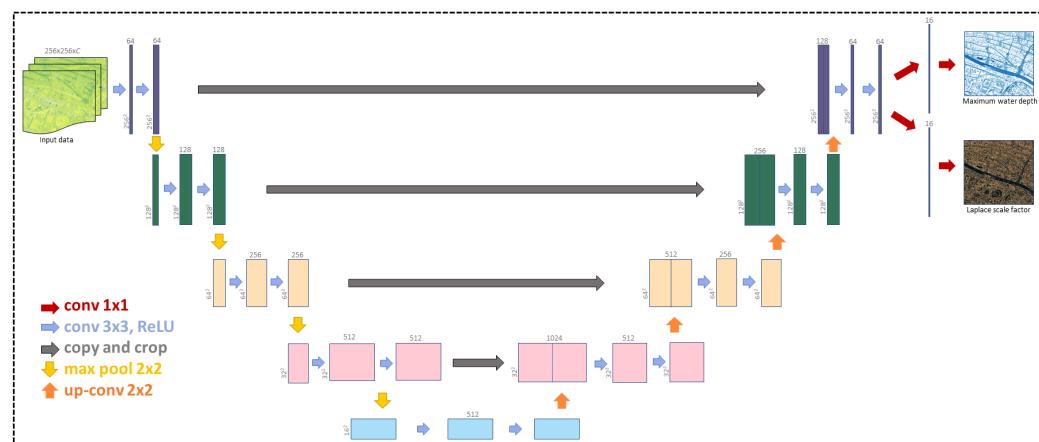
**Funding:** This study was funded by the Swiss Data Science Center (SDSC), EPFL and ETH Zürich under project MLATEM (grant number C19-07).

**Data Availability Statement:** To download the dataset please refer to the link <https://doi.org/10.3929/ethz-b-000365484> (accessed on 2 February 2022).

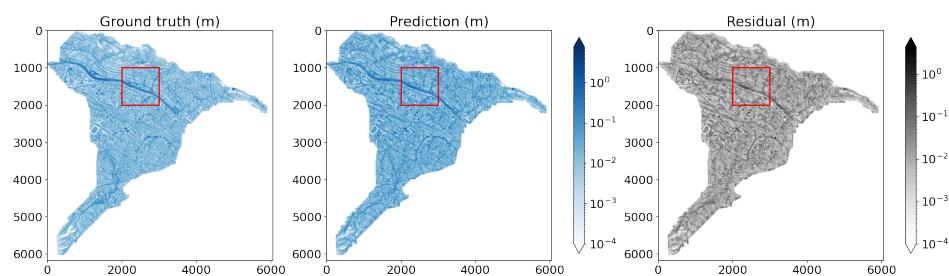
**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

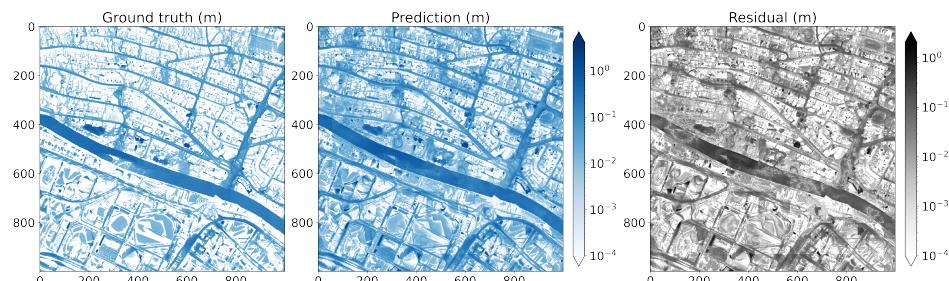
An overview of the network architecture used for the experiments in this paper is shown in Figure A1, and more visualisations of results are provided below.



**Figure A1.** Illustration of the U-Net architecture as used in this paper.

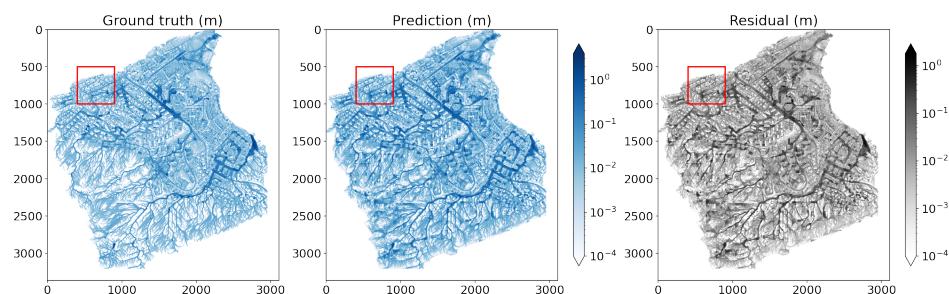


**(a)** On the left, the ground truth maximum water depth for Zurich catchment of rainfall event *tr2\_1*. In the middle, the complete predicted reconstruction performed by our *Deep Flow*. On the right, the residual of ground truth and predicted reconstruction for the same area.

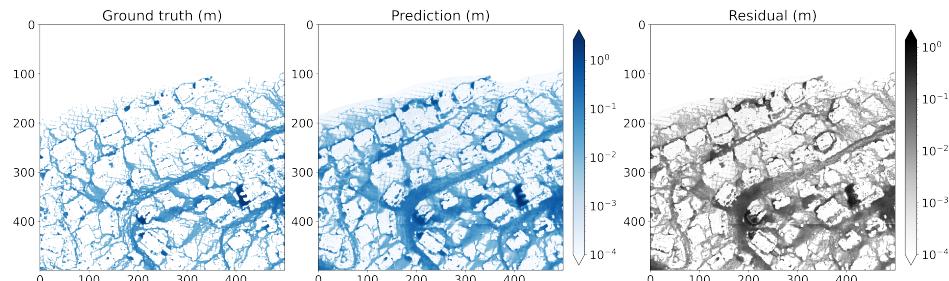


**(b)** On the left, one enlargement area of the ground truth. In the middle, the same area predicted by our *Deep Flow*. On the right, the residual of ground truth and predicted reconstruction for the same area.

**Figure A2.** Ground truth maximum water depth and prediction for Zurich catchment and rainfall event *tr2\_1*, with bounding box depicting the patch area shown in **(b)**.

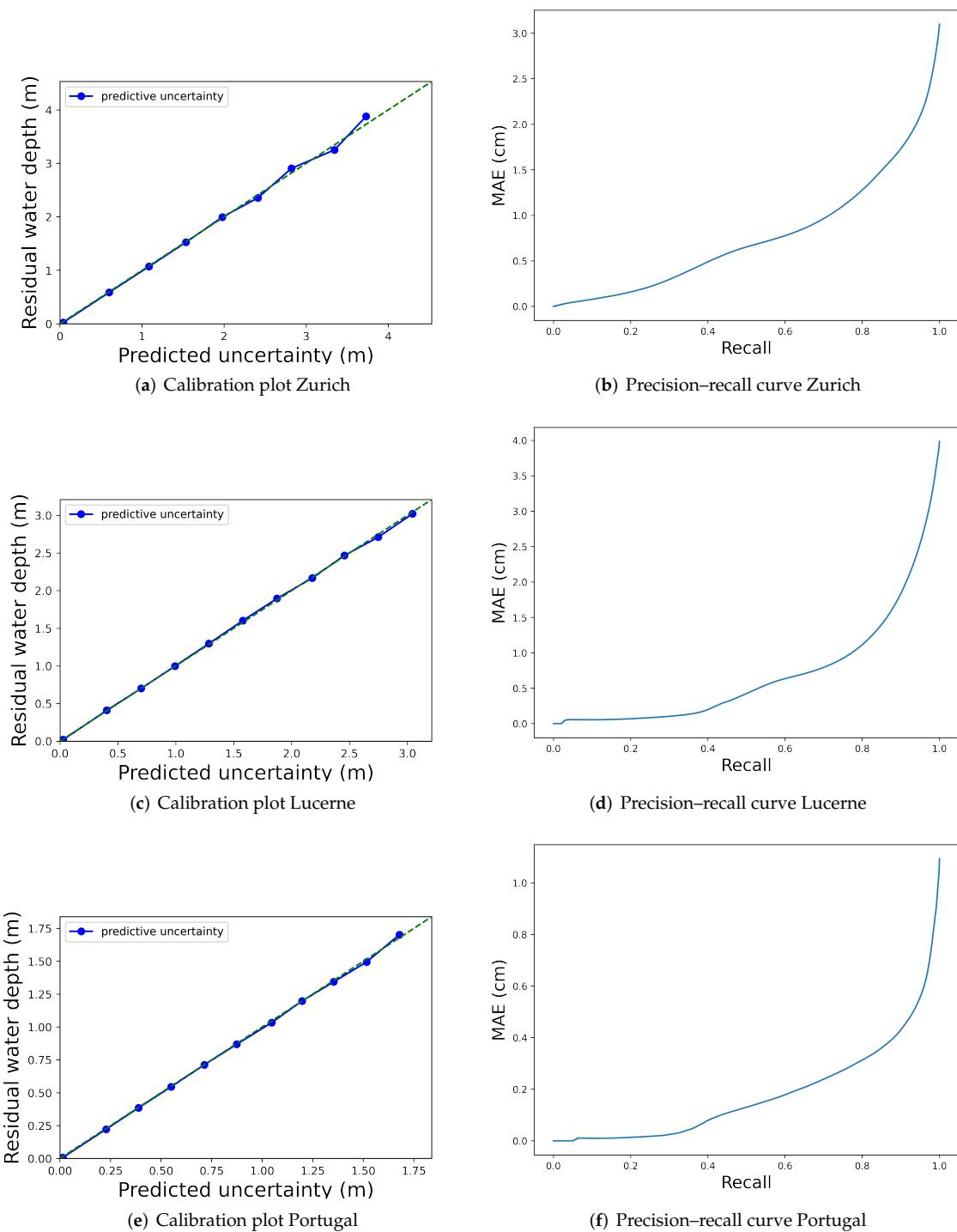


**(a)** On the left, the ground truth maximum water depth for Lucerne catchment of rainfall event *tr2\_1*. In the middle, the complete predicted reconstruction performed by our *Deep Flow*. On the right, the residual of ground truth and predicted reconstruction for the same area.

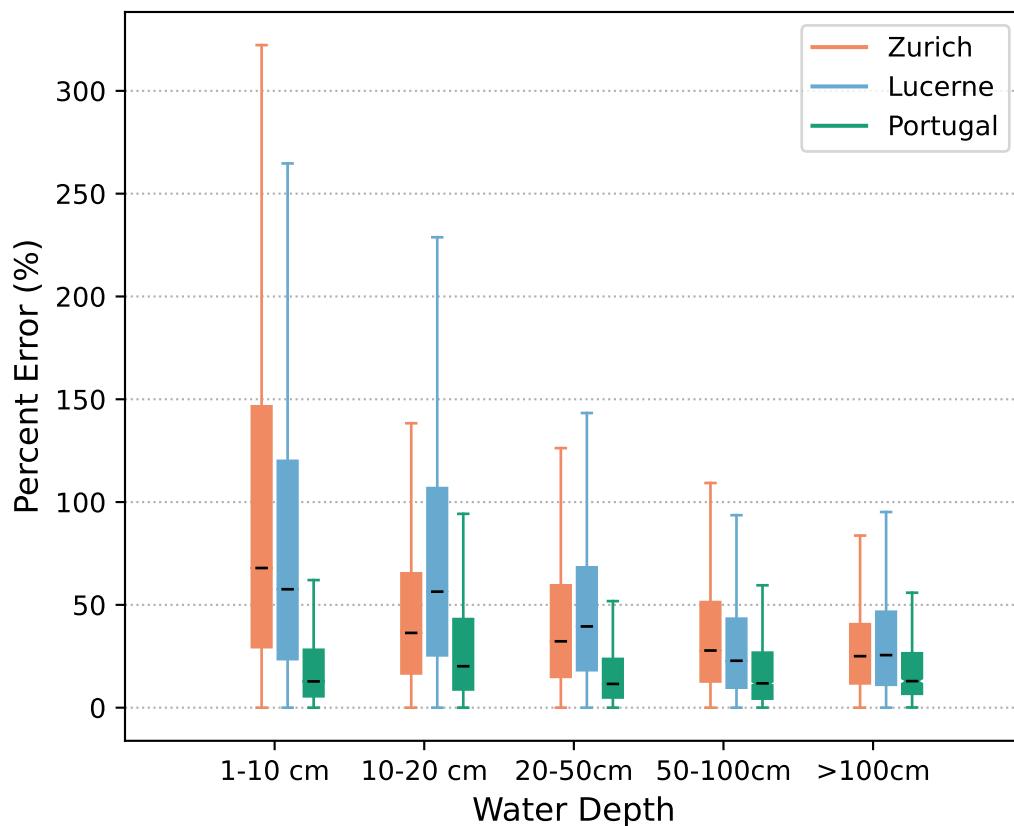


**(b)** On the left, one enlargement area of the ground truth. In the middle, the same area predicted by our *Deep Flow*. On the right, the residual of ground truth and predicted reconstruction for the same area.

**Figure A3.** Ground truth maximum water depth and prediction for Lucerne catchment and rainfall event *tr2\_1*, with bounding box depicting the patch area shown in **(b)**.



**Figure A4.** Evaluation of the predictive uncertainties predicted by *Deep Flood* for rainfall event *tr100-1* on the Zurich, Lucerne, and Portugal test catchments. (a) The residuals in metres are plotted versus the grouped predicted uncertainties. (b) The test data are filtered based on the predictive uncertainty, showing a decrease in mean absolute error when high uncertainty test data points are removed.



**Figure A5.** The box plot for the predicted water depth values and their associated per cent error. The predictions are reported for our *Deep Flood* tested on a steep test catchment and rainfall event *tr2\_1*.

## References

- Misra, A.K. Climate change and challenges of water and food security. *Int. J. Sustain. Built Environ.* **2014**, *3*, 153–165. [CrossRef]
- Rosenzweig, B.R.; McPhillips, L.; Chang, H.; Cheng, C.; Welty, C.; Matsler, M.; Iwaniec, D.; Davidson, C.I. Pluvial flood risk and opportunities for resilience. *WIREs Water* **2018**, *5*, e1302. [CrossRef]
- Adikari, Y.; Yoshitani, J. Global Trends in Water-Related Disasters: An Insight for Policymakers, 2009. Available online: <https://unesdoc.unesco.org/ark:/48223/pf0000181793> (accessed on 23 August 2022).
- Kubal, C.; Haase, D.; Meyer, V.; Scheuer, S. Integrated urban flood risk assessment – adapting a multicriteria approach to a city. *Nat. Hazards Earth Syst. Sci.* **2009**, *9*, 1881–1895. [CrossRef]
- Leitão, J.P.; Peña-Haro, S. Leveraging Video Data to Assess Urban Pluvial Flood Hazard. 2022. Available online: [https://udm2022.org/wp-content/uploads/2021/11/1428\\_Leitao\\_REV-4d457576.pdf](https://udm2022.org/wp-content/uploads/2021/11/1428_Leitao_REV-4d457576.pdf) (accessed on 2 February 2022).
- Plate, E.J. Flood risk and flood management. *J. Hydrol.* **2002**, *267*, 2–11. Advances in Flood Research. [CrossRef]
- Guo, Z.; Leitão, J.P.; Simões, N.E.; Moosavi, V. Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks. *J. Flood Risk Manag.* **2021**, *14*, e12684. [CrossRef]
- Teng, J.; Jakeman, A.; Vaze, J.; Croke, B.; Dutta, D.; Kim, S. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environ. Model. Softw.* **2017**, *90*, 201–216. [CrossRef]
- Mark, O.; Weesakul, S.; Apirumanekul, C.; Aroonmet, S.B.; Djordjević, S. Potential and limitations of 1D modelling of urban flooding. *J. Hydrol.* **2004**, *299*, 284–299. Urban Hydrology. [CrossRef]
- Kim, T.; Yang, T.; Gao, S.; Zhang, L.; Ding, Z.; Wen, X.; Gourley, J.J.; Hong, Y. Can artificial intelligence and data-driven machine learning models match or even replace process-driven hydrologic models for streamflow simulation?: A case study of four watersheds with different hydro-climatic regions across the CONUS. *J. Hydrol.* **2021**, *598*, 126423. [CrossRef]
- Li, Y.; Martinis, S.; Wieland, M. Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 178–191. [CrossRef]
- Berkhahn, S.; Fuchs, L.; Neuweiler, I. An ensemble neural network model for real-time prediction of urban floods. *J. Hydrol.* **2019**, *575*, 743–754. [CrossRef]
- Gude, V.; Corns, S.; Long, S. Flood Prediction and Uncertainty Estimation Using Deep Learning. *Water* **2020**, *12*, 884. [CrossRef]
- Löwe, R.; Böhm, J.; Jensen, D.G.; Leandro, J.; Rasmussen, S.H. U-FLOOD—Topographic deep learning for predicting urban pluvial flood water depth. *J. Hydrol.* **2021**, *603*, 126898. [CrossRef]

15. Wu, Z.; Zhou, Y.; Wang, H.; Jiang, Z. Depth prediction of urban flood under different rainfall return periods based on deep learning and data warehouse. *Sci. Total. Environ.* **2020**, *716*, 137077. [[CrossRef](#)] [[PubMed](#)]
16. Wilson, A.G.; Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates Inc.: Red Hook, NY, USA 2020; Volume 33, pp. 4697–4708.
17. Gustafsson, F.K.; Danelljan, M.; Schon, T.B. Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 14–19 June 2020.
18. Kendall, A.; Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA 2017; Volume 30.
19. Butler, D.; Digman, C.; Makropoulos, C.; Davies, J.W. *Urban Drainage*; CRC Press: Boca Raton, FL, USA, 2018. [[CrossRef](#)]
20. Borah, D.K. Hydrologic procedures of storm event watershed models: A comprehensive review and comparison. *Hydrol. Process.* **2011**, *25*, 3472–3489. [[CrossRef](#)]
21. Costabile, P.; Costanzo, C.; Macchione, F. A storm event watershed model for surface runoff based on 2D fully dynamic wave equations. *Hydrol. Process.* **2013**, *27*, 554–569. [[CrossRef](#)]
22. Cea, L.; Garrido, M.; Puertas, J. Experimental validation of two-dimensional depth-averaged models for forecasting rainfall-runoff from precipitation data in urban areas. *J. Hydrol.* **2010**, *382*, 88–102. [[CrossRef](#)]
23. Fernández-Pato, J.; Caviedes-Voullième, D.; García-Navarro, P. Rainfall/runoff simulation with 2D full shallow water equations: Sensitivity analysis and calibration of infiltration parameters. *J. Hydrol.* **2016**, *536*, 496–513. [[CrossRef](#)]
24. Caviedes-Voullième, D.; García-Navarro, P.; Murillo, J. Influence of mesh structure on 2D full shallow water equations and SCS Curve Number simulation of rainfall/runoff events. *J. Hydrol.* **2012**, *448–449*, 39–59. [[CrossRef](#)]
25. Costabile, P.; Costanzo, C.; Macchione, F. Comparative analysis of overland flow models using finite volume schemes. *J. Hydroinform.* **2011**, *14*, 122–135. [[CrossRef](#)]
26. Xia, X.; Liang, Q.; Ming, X.; Hou, J. An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations. *Water Resour. Res.* **2017**, *53*, 3730–3759. [[CrossRef](#)]
27. Liang, X.; Lettenmaier, D.P.; Wood, E.F.; Burges, S.J. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *J. Geophys. Res. Atmos.* **1994**, *99*, 14415–14428. [[CrossRef](#)]
28. Costabile, P.; Macchione, F. Enhancing river model set-up for 2-D dynamic flood modelling. *Environ. Model. Softw.* **2015**, *67*, 89–107. [[CrossRef](#)]
29. Bradbrook, K.; Lane, S.; Waller, S.; Bates, P. Two dimensional diffusion wave modelling of flood inundation using a simplified channel representation. *Int. J. River Basin Manag.* **2004**, *2*, 211–223. [[CrossRef](#)]
30. Chen, A.; Djordjević, S.; Leandro, J.; Savić, D. The urban inundation model with bidirectional flow interaction between 2D overland surface and 1D sewer networks. In Proceedings of the Novatech Sixth International Conference on Sustainable Techniques and Strategies in Urban Water Management; Lyon, France, 25–28 June 2007, pp. 465–472.
31. Bates, P.D.; Horritt, M.S.; Fewtrell, T.J. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *J. Hydrol.* **2010**, *387*, 33–45. [[CrossRef](#)]
32. Monaghan, J. Simulating Free Surface Flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406. [[CrossRef](#)]
33. Ye, J.; McCorquodale, J.A. Simulation of Curved Open Channel Flows by 3D Hydrodynamic Model. *J. Hydraul. Eng.* **1998**, *124*, 687–698. [[CrossRef](#)]
34. Mosavi, A.; Ozturk, P.; Chau, K.W. Flood Prediction Using Machine Learning Models: Literature Review. *Water* **2018**, *10*, 1536. [[CrossRef](#)]
35. Hunter, N.M.; Horritt, M.S.; Bates, P.D.; Wilson, M.D.; Werner, M.G. An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. *Adv. Water Resour.* **2005**, *28*, 975–991. [[CrossRef](#)]
36. Lhomme, J.; Sayers, P.; Gouldby, B.; Samuels, P.; Wills, M.; Mulet-Martí, J. Recent development and application of a rapid flood spreading method. In *Flood Risk Management: Research and Practice*; Taylor & Francis Group: London, UK, 2008.
37. Jamali, B.; Bach, P.M.; Cunningham, L.; Deletic, A. A Cellular Automata Fast Flood Evaluation (CA-ffé) Model. *Water Resour. Res.* **2019**, *55*, 4936–4953. [[CrossRef](#)]
38. Hunter, N.M.; Bates, P.D.; Neelz, S.; Pender, G.; Villanueva, I.; Wright, N.G.; Liang, D.; Falconer, R.A.; Lin, B.; Waller, S.; et al. Benchmarking 2D hydraulic models for urban flooding. *Proc. Inst. Civ. Eng. Water Manag.* **2008**, *161*, 13–30. [[CrossRef](#)]
39. Neelz, S.; Pender, G. *Benchmarking of 2D Hydraulic Modelling Packages SC080035/SR2*; Environment Agency: Bristol, UK, 2010. Available online: [https://books.google.ch/books/about/Benchmarking\\_of\\_2D\\_Hydraulic\\_Modelling\\_P.html?id=ghoZYAAACAAJ&redir\\_esc=y](https://books.google.ch/books/about/Benchmarking_of_2D_Hydraulic_Modelling_P.html?id=ghoZYAAACAAJ&redir_esc=y) (accessed on 2 February 2022).
40. Guidolin, M.; Chen, A.S.; Ghimire, B.; Keedwell, E.C.; Djordjević, S.; Savić, D.A. A weighted cellular automata 2D inundation model for rapid flood analysis. *Environ. Model. Softw.* **2016**, *84*, 378–394. [[CrossRef](#)]
41. Hernandez, O.J.; Alamia, J.A. Precision stabilization simulation of a ball joint gimbaled mirror using advanced MATLAB® techniques. In Proceedings of the IEEE Southeastcon 2009, Atlanta, GA, USA, 5–8 March 2009; pp. 72–77. [[CrossRef](#)]
42. Zhao, G.; Pang, B.; Xu, Z.; Peng, D.; Xu, L. Assessment of urban flood susceptibility using semi-supervised machine learning model. *Sci. Total. Environ.* **2019**, *659*, 940–949. [[CrossRef](#)] [[PubMed](#)]

43. Tehrany, M.S.; Pradhan, B.; Mansor, S.; Ahmad, N. Flood susceptibility assessment using GIS-based support vector machine model with different kernel types. *CATENA* **2015**, *125*, 91–101. [[CrossRef](#)]
44. Choubin, B.; Moradi, E.; Golshan, M.; Adamowski, J.; Sajedi-Hosseini, F.; Mosavi, A. An ensemble prediction of flood susceptibility using multivariate discriminant analysis, classification and regression trees, and support vector machines. *Sci. Total. Environ.* **2019**, *651*, 2087–2096. [[CrossRef](#)] [[PubMed](#)]
45. Kisi, O.; Choubin, B.; Deo, R.C.; Yaseen, Z.M. Incorporating synoptic-scale climate signals for streamflow modelling over the Mediterranean region using machine learning models. *Hydrol. Sci. J.* **2019**, *64*, 1240–1252. [[CrossRef](#)]
46. Sadler, J.; Goodall, J.; Morsy, M.; Spencer, K. Modeling urban coastal flood severity from crowd-sourced flood reports using Poisson regression and Random Forest. *J. Hydrol.* **2018**, *559*, 43–55. [[CrossRef](#)]
47. Chang, L.C.; Chang, F.J.; Chiang, Y.M. A two-step-ahead recurrent neural network for stream-flow forecasting. *Hydrol. Process.* **2004**, *18*, 81–92. [[CrossRef](#)]
48. Chang, F.J.; Chen, P.A.; Lu, Y.R.; Huang, E.; Chang, K.Y. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* **2014**, *517*, 836–846. [[CrossRef](#)]
49. Kratzert, F.; Klotz, D.; Shalev, G.; Klambauer, G.; Hochreiter, S.; Nearing, G. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrol. Earth Syst. Sci.* **2019**, *23*, 5089–5110. [[CrossRef](#)]
50. Tien Bui, D.; Hoang, N.D.; Martínez-Álvarez, F.; Ngo, P.T.T.; Hoa, P.V.; Pham, T.D.; Samui, P.; Costache, R. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Sci. Total. Environ.* **2020**, *701*, 134413. [[CrossRef](#)]
51. Guo, X.; Li, W.; Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, San Francisco, CA, USA, 13–17 August 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 481–490. [[CrossRef](#)]
52. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML'17, Sydney, Australia, 6–11 August 2017*; Volume 70, pp. 1321–1330.
53. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML'16, New York, NY, USA, 19–24 June 2016*; Volume 48, pp. 1050–1059.
54. Tagasovska, N.; Lopez-Paz, D. Single-Model Uncertainties for Deep Learning. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2019; Volume 32.
55. Van Amersfoort, J.; Smith, L.; Teh, Y.W.; Gal, Y. Uncertainty Estimation Using a Single Deep Deterministic Neural Network. In *Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020*; III, H.D., Singh, A., Eds.; 2020; Volume 119, pp. 9690–9700.
56. Liu, J.Z.; Lin, Z.; Padhy, S.; Tran, D.; Bedrax-Weiss, T.; Lakshminarayanan, B. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7498–7512.
57. Mukhoti, J.; Kirsch, A.; van Amersfoort, J.; Torr, P.H.S.; Gal, Y. Deterministic Neural Networks with Appropriate Inductive Biases Capture Epistemic and Aleatoric Uncertainty. *arXiv* **2021**, arXiv:2102.11582.
58. Kopetzki, A.K.; Charpentier, B.; Zügner, D.; Giri, S.; Günnemann, S. Evaluating Robustness of Predictive Uncertainty Estimation: Are Dirichlet-based Models Reliable? In *Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021*; Meila, M., Zhang, T., Eds.; 2021; Volume 139, pp. 5707–5718.
59. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30.
60. MacKay, D.J. Bayesian Methods for Adaptive Models. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 1992.
61. Shen, G.; Chen, X.; Deng, Z. Variational Learning of Bayesian Neural Networks via Bayesian Dark Knowledge. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020*; IJCAI-20; Bessiere, C., Ed.; International Joint Conferences on Artificial Intelligence Organization, 2020; pp. 2037–2043. Main track. [[CrossRef](#)]
62. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer: Berlin/Heidelberg, Germany, 1996.
63. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
64. Wilson, A.G. The Case for Bayesian Deep Learning. *arXiv* **2020**, arXiv:2001.10995.
65. Ganaie, M.A.; Hu, M.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *arXiv* **2021**, arXiv:2104.02395.
66. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; pp. 770–778. [[CrossRef](#)]
67. Ashukha, A.; Lyzhov, A.; Molchanov, D.; Vetrov, D. Pitfalls of In-Domain Uncertainty Estimation and Ensembling in Deep Learning. *arXiv* **2020**, arXiv:2002.06470.
68. Ghimire, B.; Chen, A.S.; Guidolin, M.; Keedwell, E.C.; Djordjević, S.; Savić, D.A. Formulation of a fast 2D urban pluvial flood model using a cellular automata approach. *J. Hydroinform.* **2012**, *15*, 676–686. [[CrossRef](#)]
69. Innovyz. InfoWorks ICM. 2019. Available online: <https://www.innovyz.com/en-us/products/infoworks-icm> (accessed on 27 July 2022),

70. QGIS Development Team. *QGIS Geographic Information System*; Open Source Geospatial Foundation, 2022. Available online: <https://gis.stackexchange.com/questions/23622/citing-qgis-in-formal-publications#:~:text=A%3A%20To%20cite%20QGIS%20software,qgis.osgeo.org%22> (accessed on 2 February 2022).
71. Wilson, J.P.; Gallant, J.C. *Terrain Analysis: Principles and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2000.
72. Horn, B. Hill shading and the reflectance map. *Proc. IEEE* **1981**, *69*, 14–47. [CrossRef]
73. Zevenbergen, L.W.; Thorne, C.R. Quantitative analysis of land surface topography. *Earth Surf. Process. Landforms* **1987**, *12*, 47–56. [CrossRef]
74. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
75. Kendall, A. Geometry and Uncertainty in Deep Learning for Computer Vision. PhD Thesis, University of Cambridge, Cambridge, UK, 2018.
76. Nix, D.; Weigend, A. Estimating the mean and variance of the target probability distribution. In Proceedings of the 1994 IEEE International Conference on Neural Networks (ICNN'94), Orlando, FL, USA, 28 June–2 July 1994; Volume 1, pp. 55–60. [CrossRef]
77. Le, Q.V.; Smola, A.J.; Canu, S. Heteroscedastic Gaussian Process Regression. In Proceedings of the 22nd International Conference on Machine Learning, ICML '05, Bonn Germany, 7–11 August 2005; Association for Computing Machinery: New York, NY, USA, 2005; pp. 489–496. [CrossRef]
78. Ronchetti, E.M.; Huber, P.J. *Robust Statistics*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
79. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012.
80. Hecht-Nielsen. Theory of the backpropagation neural network. In Proceedings of the International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 1989; Volume 1, pp. 593–605. [CrossRef]
81. Song, C.M. Data construction methodology for convolution neural network based daily runoff prediction and assessment of its applicability. *J. Hydrol.* **2022**, *605*, 127324. [CrossRef]
82. Izmailov, P.; Vikram, S.; Hoffman, M.D.; Wilson, A.G. What Are Bayesian Neural Network Posteriors Really Like? *CoRR* **2021**, *abs/2104.14421*.
83. Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.V.; Lakshminarayanan, B.; Snoek, J. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. *arXiv* **2019**, arXiv:stat.ML/1906.02530.
84. Osband, I.; Blundell, C.; Pritzel, A.; Van Roy, B. Deep Exploration via Bootstrapped DQN. In *Advances in Neural Information Processing Systems*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2016; Volume 29.