One model suits all: data-driven rapid flood prediction with catchment generalizability using convolutional neural networks

Zifeng Guo[1], Vahid Moosavi[2], and João P. Leitão[3]

[1]Swiss Federal Institute of Technology Zurich (ETHZ)
[2]ETH Zurich
[3]Eawag - Swiss Federal Institute of Aquatic Science and Technology

November 23, 2022

Abstract

Data-driven and machine learning models have recently received increasing interest to resolve the bottleneck of computational speed faced by various physically-based simulations. A few studies have explored the application of these models to develop new, and fast, applications for fluvial and pluvial flood extent mapping, and flood susceptibility assessment. However, most studies have focused on model development for specific catchment areas, drainage networks or gauge stations. Hence, their results cannot be directly reused to other contexts unless extra data are available and the models are further trained. This study explores the generalizability of convolutional neural networks (CNNs) as flood prediction models. The study proposes a CNN-based model that can be reused in different catchment areas with different topography once the model is trained. The study investigates two options, patch- and resizing-based options, to process catchment areas of different sizes and different boundary shapes. The results showed that the CNN-based model generalizes well on "unseen" catchment areas with promising prediction accuracy and significantly less computational time when compared to physically-based models. The obtained results also suggest that the patch-based option is more effective than the resizing-based option in terms of prediction accuracy. In addition, all experiments have shown that the prediction of flow velocity is more accurate than water depth, suggesting that the water accumulation is more sensitive to global elevation information than flow velocity. Therefore, one can suggest that CNN-based models for flood prediction should consider large-size inputs and have large receptive field architecture to achieve a better performance.

One model suits all: data-driven rapid flood prediction with catchment generalizability using convolutional neural networks

Z. Guo[1], V. Moosavi[1], and J. P. Leitão[2]

[1]Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland.

[2]Swiss Federal Institute of Aquatic Science and Technology (Eawag), Switzerland.

Corresponding author: Zifeng Guo (guo@arch.ethz.ch)

Key Points:

- Data-driven flood prediction model capable to generalize to different catchment areas.
- Two different spatial discretization options to handle catchment areas of different sizes.

Abstract

Data-driven and machine learning models have recently received increasing interest to resolve the bottleneck of computational speed faced by various physically-based simulations. A few studies have explored the application of these models to develop new, and fast, applications for fluvial and pluvial flood extent mapping, and flood susceptibility assessment. However, most studies have focused on model development for specific catchment areas, drainage networks or gauge stations. Hence, their results cannot be directly reused to other contexts unless extra data are available and the models are further trained. This study explores the generalizability of convolutional neural networks (CNNs) as flood prediction models. The study proposes a CNN-based model that can be reused in different catchment areas with different topography once the model is trained. The study investigates two options, patch- and resizing-based options, to process catchment areas of different sizes and different boundary shapes. The results showed that the CNN-based model generalizes well on "unseen" catchment areas with promising prediction accuracy and significantly less computational time when compared to physically-based models. The obtained results also suggest that the patch-based option is more effective than the resizing-based option in terms of prediction accuracy. In addition, all experiments have shown that the prediction of flow velocity is more accurate than water depth, suggesting that the water accumulation is more sensitive to global elevation information than flow velocity. Therefore, one can suggest that CNN-based models for flood prediction should consider large-size inputs and have large receptive field architecture to achieve a better performance.

1 Introduction

Solving physics-related problems using data-driven and machine learning models has recently become a research field receiving growing attention. Many challenging problems, especially those that relate with dynamic processes, are being tackled by learning from large datasets using machine learning models (e.g., Greydanus et al., 2019; Read et al., 2019). Compared to conventional models that are typically based on a system of equations that describe the physical phenomena, data-driven models, such as artificial neural networks, have two major advantages. First, data-driven models can produce relatively accurate predictions without the need of having the full *a priori* knowledge of the phenomena. The accuracy of the model is related to the amount of data available. This is useful when working with complex phenomena such as weather forecasting (e.g., Xingjian el al., 2015; Cramer el al., 2017). Second, data-driven models can be used as surrogate models for computationally expensive simulations such as fluid dynamics (e.g., Tompson et al., 2017; Raissi et al., 2018), agent-based simulations (e.g., Feng et al., 2016) and topology optimizations (e.g., Li et al., 2019; Sosnovik & Oseledets, 2019). The computational process of data-driven models is independent of the problem context. Therefore, when combined with parallel computing techniques, data-driven surrogate models can significantly accelerate the computational process, especially if considerable number of simulations are required.

Recently, data-driven models have also gained interest for flow and flood modeling applications. A considerable number of studies have been conducted using data-driven methods for tasks such as flood extent mapping (e.g., Gebrehiwot et al., 2019; Moy de Vitry et al., 2019), flood susceptibility assessment (e.g., Zhao et al., 2019; Bui et al., 2020; Zhao et al., 2020; Wang et al., 2020), and pluvial flood predictions (e.g., Huang et al., 2014; Tan et al., 2018; Berkhahn et al., 2019). These studies have shown that machine learning techniques can handle a wide range of flood-related problems with acceptable accuracy when sufficient data are available. However, most of these studies have focused on specific catchment areas or drainage systems. Their results cannot be directly transferred to other locations without adding more data and further training of the models. Although several studies exist for flood prediction in different terrains, these studies are either based on high-level parameters of a terrain generator instead of the raw elevation data (e.g., Mustafa et al., 2018), or consist of multiple site-specific models rather than one general model (e.g., Berkhahn et al., 2019; Kratzert et al., 2019a), which prevent these models from being reused to other scenarios and applications.Recently, data-driven models have also gained interest for flow and flood modeling applications. A considerable number of studies have been conducted using data-driven methods for tasks such as flood extent mapping (e.g., Gebrehiwot et al., 2019; Moy de Vitry et al., 2019), flood susceptibility assessment (e.g., Zhao et al., 2019; Bui et al., 2020; Zhao et al., 2020; Wang et al., 2020), and pluvial flood predictions (e.g., Huang et al., 2014; Tan et al., 2018; Berkhahn et al., 2019). These studies have shown that machine learning techniques can handle a wide range of flood-related problems with acceptable accuracy when sufficient data are available. However, most of these studies have focused on specific catchment areas or drainage systems. Their results cannot be directly transferred to other locations without adding more data and further training of the models. Although several studies exist for flood prediction in different terrains, these studies are either based on high-level parameters of a terrain generator instead of the raw elevation data (e.g., Mustafa et al., 2018), or consist of multiple site-specific models rather than one general model (e.g., Berkhahn et al., 2019; Kratzert et al., 2019a), which prevent these models from being reused to other scenarios and applications.

Despite the recent investigations, data-driven flood prediction models that can generalize to different raw terrain inputs (called *general flood prediction models*) remains rare. The prediction of catchment-level rainfall-runoff relations was presented by Kratzert et al. (2019b) in which a recurrent neural network model was tested on basins that were not included in the training data. Other type of predictions, such as surface water depth and flow velocities, have not yet been well-studied. The lack of general flood prediction models can be justified by two main reasons. First, such model requires a machine learning algorithm that can handle different terrain inputs. The model should be able to systematically process catchment rasters of different size or drainage network graphs with different number of nodes. This is a challenging task as machine learning algorithms such as fully-connected neural networks require input vectors to have the same dimensionality. Second, making an urban-scale general flood prediction model requires large amount of flood data to be available as the training data. Considering the size and spatial and temporal resolutions of a typical urban flood simulation, preparing a large flood dataset is demanding and computationally expensive. Therefore, despite the recent exciting data-driven flood modelling applications, researchers and urban planners still lack proper surrogate models for large-scale simulation-intensive applications such as urban flood risk management, real-time pluvial flood forecast and flood-driven urban planning. This situation emphasizes the need of a data-driven model capable for accurate flood predictions on different catchments.

In this study, we propose a data-driven pluvial flood prediction model that can generalize to different terrain inputs. In other words, once the model is trained, it can be used to different catchment areas that are not included in the training data. The proposed model represents the pluvial flood prediction as an image-to-image translation task that can be handled by convolutional neural networks (CNNs). As CNNs were shown effective to generalize on various rainfall events for urban-scale inundation prediction (Guo et al., 2020a), we mainly focus on the flood prediction of the same event in different catchment areas. Currently, our model only predicts maximum water depths and flow velocities as they are the key factors used by risk assessments and urban planning. The main contributions of our study include:

1. A new data-driven flood prediction model capable to generalize to different catchment areas, i.e., areas with different topography, and to generate flood predictions in several seconds with a promising accuracy compared to physically-based simulations.

2. A set of tests of two different spatial discretization options to handle catchment areas of different sizes, which can be used as reference for further research.

3. A large pluvial flood dataset generated using a simplified physically-based flood model that can contribute to other related flood prediction studies.

2 Flow and Flood Estimation Related Studies

Data-driven models as “surrogates” to accelerate the computational process of physically-based simulations have been intensively discussed in many different fields such as computer graphics and computational fluid dynamics. One of the earliest studies in this areas was presented by Ladicky et al. (2015) who trained a regression forest using simulation data to predict the new states of liquid particles from their previous states. The trained regression model was capable to generate realistic fluid animations consisting of millions of particles in an interactive frame rate. In addition to the particle-based simulations, Guo et al. (2016) shown that the grid-based fluid simulations can also be approximated accurately by machine learning algorithms. They introduced a CNN model which predicted the velocity field of the steady flow from discretized input geometries. Tompson et al. (2017) used a CNN to infer the pressure field from the input geometries and the divergence of the velocity field. The trained CNN replaced the conventional iterative linear solver and thus accelerated the simulation process. Raissi et al. (2018) adopted a fully connected neural network to infer displacement, velocity, and pressure from input space-time specifications. They applied the network to the vibrating cylinder problem and achieved a very high prediction accuracy. However, neither the computational speed nor the generalizability to other scenarios was reported. Amaranto et al. (2018) proposed to use fully-connected neural networks to predict the future ground water level from input factors such as precipitation and current water level. The model was combined with an optimization process for better neural network design. Kim et al. (2019) proposed a novel CNN structure to predict fluid velocity from a set of reduced input parameters such as the source position and the inflow velocity. In their study, multiple CNN instances were trained for different simulation scenarios. Recently, Thuerey et al. (2020) used CNN to directly infer both the velocity and the pressure field from input airfoil geometries. They used a bottleneck neural network structure which convert input array to an output array of the same size.

Data-driven models have also been considered for river flow and flood modeling. One of the research directions is the long-term water-level forecast for specific locations based on observational rainfall data. For example, Chang et al. (2004) used a recurrent neural network to forecast the two-step-ahead river stream flow based on the rainfall measurements from several gauge stations. After the training, the neural network was capable to forecast 2-hour ahead stream flow appropriately. This method was later extended to multiple-step-ahead using an expandable neural network architecture. The inputs of the neural networks included not only rainfall measurements but also the historical water depth observations (Chen et al., 2013; Chang et al., 2014). Kratzert et al. (2019a) trained several basin-specific recurrent neural networks with long short-term memory (LSTM) network cells. The trained models outperformed the calibrated traditional hydrology models. Recently, Gude et al. (2020) used recurrent neural networks to predict long-term water depths in rivers as well as the associated uncertainties.

In contrast to these works that focused on long-term predictions of a specific location, another research direction is to predict the water depths or the flood susceptibility within an area of interest. This direction typically uses neural networks to learn the correlation between several designed input terrain features and the output water depths. For example, Berkhahn et al. (2019) used fully-connected neural networks to estimate the water depth of several catchment areas in real-time based on synthetic rainfall events. The catchment areas were discretized to rectangular grids with each cell corresponded to one output of the neural networks. Large catchment areas were modeled by multiple neural networks with zero-value cells neglected. Bui et al. (2020) used a fully-connected neural network to predict the flood susceptibility for the scattered locations within a catchment area. The inputs of the neural network were designed features such as slope, curvature and elevations, and the outputs were binary value indicated the susceptibility. Wang et al. (2020) adopted a similar pipeline for susceptibility mapping. Their model used a CNN instead of a fully-connected network. The neural network was trained using 76 sample locations within the studied catchment area and tested with other locations within the same area. Guo et al. (2020a) showed that CNNs could accurately predict the maximum water depths in specific catchment areas from varying input hyetographs.

Besides these studies that were based on neural networks, a few other exist that used other type of data-driven models. For example, Tehrany et al. (2013) proposed a rule-based decision tree that estimate the flood susceptibility based on selected input factors, such as soil type, terrain curvature and distance to rivers. Other methods such as logistic regressions and support vector machines were also investigated (Tehrany et al., 2017, 2019). Zaghloul (2017) used a ray-shooting method to extract geometric features in different spatial locations. The features were used to train a self-organizing map to predict the velocity field of a steady wind flow. Leitão et al. (2018) used similar feature-extraction and learning methods for flood prediction. The methods were tested in several benchmark cases showing promising accuracy.

3 Problem Statement

Even though data-driven techniques have already been explored for river flow and flood modeling, most studies were limited on specific catchment areas or gauge stations. Therefore, further investigations are needed to study the generalizability of data-driven models on flood predictions on different catchment areas. In this study, we focus on a data-driven model for flood predictions with terrain generalizability, which means once the model is trained, it can be used on different catchment areas not included in the training data. As a first step of this study, we simplify the problem by focusing on the maximum water depths and flow velocities. We also restrict the rainfall event to a designed 100-year storm instead of any events.

In the proposed model, we consider flood prediction as a supervised learning task, meaning that the prediction model is trained using input-output pairs. The inputs are elevation raster data and outputs are raster data of simulation results, namely water depth and velocity in this case. After the training step, the model can predict the maximum water depth and flow velocity from the new elevation data that is fed as input. We use CNNs to implement the prediction model, which, when compared to other machine learning algorithms such as fully-connected neural networks, can utilize the spatial information of adjacent pixels (raster cells) without facing an exponential growth of model parameters. This gives CNNs a huge advantage for handling image-like data such as raster datasets. However, challenges still remain. The major challenge is that, unlike previous studies from computer graphics in which the simulation domains are relatively small (e.g., 256×256 pixels), catchment areas for typical flood prediction tasks are large (e.g., 3,000×3,000 pixels). Running CNNs on large input would be infeasible due to the memory limitation of most graphic cards. To overcome this challenge, we propose and investigate two options: the patch- and the resizing-based options. A baseline experiment is also considered in our study to evaluate the performance of the two proposed options.

3.1 Patch-based option

The patch-based option samples elevation and inundation patches from the catchment areas. The patch sampling process is random and the obtained patches may contain no-data pixels. The patches are used to train and validate the CNN models. After the training step, flood predictions for new catchment areas are also obtained at patch level. The flood patches are then assembled as the final prediction. Furthermore, we oversample the target catchment area to produce patches that overlap. As suggested by the previous study (Guo et al., 2020a), we use the mean value of the overlapped pixels to further reduce errors. The patch-based option was shown effective for describing the original objects, for example, local patches can be used to match different 3D geometries (e.g., Masci et al., 2015), or segment objects from arbitrarily large images (e.g., Ronneberger et al., 2015). Nevertheless, considering the information loss caused by the patch sampling, we chose a relatively large patch size of 1,024×1,024 to preserve as much global information as possible; we have also tested other patch sizes for comparison purposes.

3.2 Resizing-based option

The resizing-based option down-samples large catchment areas, and then up-samples the outputs to their original sizes. The purpose of this option is to study whether CNNs can effectively handle resized or even distorted inputs and make accurate predictions as in other applications, such as detecting the boundary of objects from images taken from different angles (e.g., Badrinarayanan et al., 2017). The resizing-based option preserves global elevation information but destroys local detailed patterns. The lost details were shown by previous studies (e.g., Chu & Thuerey, 2017) re-generatable by synthetic up-sampling methods. We choose a large input size (1024×1024) to preserve as much local information as possible. Also, we only resize catchments that are larger than this size to avoid extra information loss. Catchments that are smaller than the input size are padded with 0s instead of scaled up. The resizing process preserves the aspect ratio of catchment areas.

3.3 Baseline experiment

In addition to the two proposed options, we introduce a baseline experiment to investigate how accurate CNNs are in an ideal situation, i.e., when terrain data for flood simulations have the same size of the input size of the CNNs. These terrain data do not necessarily represent full catchment areas. Therefore, the simulation results produced by these data cannot be applied to real applications. We would like to emphasize that the purpose of this experiment is not for applicational scenarios, rather, it is to study the output difference between a CNN model and a physically-based model when the two models are provided with identical inputs. The experiment excludes the information loss caused by the patch- or resize-based options. Therefore, the result can be interpreted as the potential upper bound of the accuracy of the proposed CNN models, which is a useful reference to assess the performance of the two proposed options.

4 Proposed CNN Model

Water accumulation in a small region is the result of rainfall falling directly in the region, water flowing from an upstream region and water leaving the region to downstream areas. As such, for our CNN model, each pixel of the model's output layer should "see" as many input pixels as possible in order to make accurate predictions. In other words, the CNN model should learn from the global elevation information rather than only from local terrain patterns (Geirhos et al., 2019). The region of the "visible pixels" is called the receptive field (Luo et al., 2016) and it can be effectively increased by (1) adding more network layers and (2) using larger convolutional kernels. Based on these considerations, we design our CNN model using deep networks with relatively large convolutional kernel so that a large receptive field is achieved.

4.1 Model design

The CNN model is designed based on the structure of U-Net (Ronneberger et al., 2015), a neural network architecture that is characterized by the skip-connections between shallow and deep layers. The skip connections of U-Net offer two advantages: 1) deep networks without skip connections are difficult to train and sometimes less accurate (He el al., 2016), and 2) deep networks tend to "smooth" the adjacent pixels in the output layer and destroy the output detail patterns (Long et al., 2015). Although small convolutional kernels can improve the output details (Badrinarayanan et al., 2017), using small kernels is in contradiction with having a desired large receptive field. In contrast, the skip connections can preserve information from the shallow layers, improving the obtained details quality in the output layer.

The structure of the CNN model is shown in Figure 1. The model consists of an encoder and a decoder. The encoder is a series of convolutional and max-pooling layers which compress the input raster to arrays of smaller sizes. The decoder is a series of up-sampling and convolutional layers which decompress the compressed arrays to the output raster. For each up-sampling layer of the decoder, its output array is concatenated with the array of the same size produced by the encoder. The concatenated arrays are fed to the successive layer of the up-sampling layers.

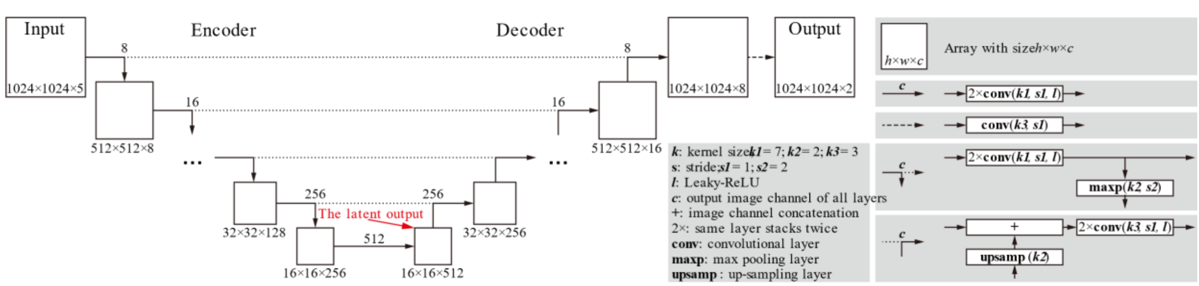

Figure 1. The prediction model. Note that the not all layers are shown for visualization purpose

The number of layers of the CNN model depends on the size of the input. The goal is to have the receptive field in the latent layer (the last layer of the encoder) larger than the input size. The receptive field rn of the n-th hidden layer of the encoder can be calculated using Equation 1.

$$r_n = \begin{cases} k_1, \text{if } n = 1 \\ r_{n-1} + (k_n - 1)\prod_{i=1}^{n-1} s_i \text{ , if } n > 1 \end{cases} \#(1)$$

In the equation, k_n, s_n are the kernel size (the size of the convolutional kernel) and the stride of the *n-th* hidden layer, respectively. For max-pooling layers, $k=s$. Therefore, the larger the input size, the deeper the network.

Based on this formulation, we tested different combinations of kernel size, stride, and number of layers. We found that, for the encoder part, a good combination to efficiently increase the receptive field is two convolutional layers with k=7 and s=1 followed by one max pooling layer with k=2 and s=2. For the decoder part, we used a symmetrical layer sequence and replace all max-pooling layers by up-sampling layers with k=2. All convolutional layers of the decoder part have a k=3 in order to better preserve detail spatial patterns. The activation functions for all except the last convolutional layers are Leaky-ReLU (Maas et al., 2013). The Leaky-ReLU function avoids the “vanishing gradient problem” (Hochreiter et al., 1998) of the sigmoid functions and the dead neuron problem of the rectified linear function (Nair & Hinton, 2010). The output layer has no activation function and produces unbounded values.

4.2 Processing elevation data

The raw elevation raster x_{raw} are rescaled to $x = c(\max(x_{raw}) - x_{raw})$ for data normalization, in which max returns the maximum value of x_{raw} and c is a constant. By conducting multiple training process with different c values, we found that smaller c such as 0.01 performs better than large c values in terms of the prediction accuracy of the model. The rescaled elevations are concatenated with additional features that are derived from x_{raw}. The features, which are obtained using the approach of De Smith et al. (2007), include slope, aspect and curvature. All no-data pixels are filled with 0s. We compared training processes with and without terrain features, and found that, although the CNNs can learn from raw data without any designed features, using terrain features makes the training process converge faster. The testing results of different c values and different terrain features are shown in the Appendix. The values of flood simulation rasters are unchanged and are used as the ground truth for training and validation.

5 Experiments

The proposed flood forecasting method was tested in three experiments using real elevation data. These experiments corresponded to the two options described in Sections 3.1 and 3.2: the patch- and the resizing-based options, and the baseline experiment described Section 3.3. For each experiment, several CNN models with different input sizes and kernel sizes were compared. The CNN models were trained separately for different experiments, which means if two models have the same design (input size, kernel size etc.) but used in different experiments, they were trained using different training data.

5.1 Terrain and rainfall data

The elevation data for the experiment were collected from the GeoVITe geodata service of ETH Zurich (https://geovite.ethz.ch/). The data were downloaded as 2 m raster tiles and were processed using GIS software into catchment areas (Figure 2). The collected elevation data consist of two regions. The first region is an area of approximately 90 km × 65 km around the Canton of Zurich, Switzerland. This region contains 649 catchment areas. The second region corresponds to the cities of Lausanne and Geneva, Switzerland, and contains seven catchment areas. We denote the first region as the "main dataset" and the second region as the "validation dataset". The purpose of the validation dataset is to test the performance of our model when "unfamiliar" elevation data are presented. All the catchment areas were used in the patch-based and resizing-based experiments. The baseline experiment, however, used 1,000 elevation patches that were randomly sampled from the main dataset without considering the boarder of catchment areas.

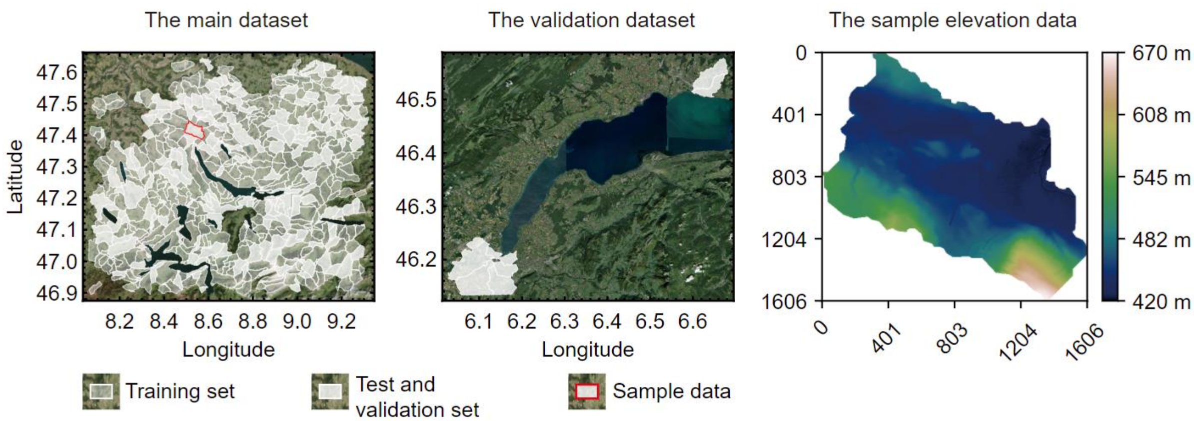

Figure 2. Dataset used in the study.

The ground truth data (i.e., flooding results) for all experiments were created by the five-hour length simulations of a 1-hour duration large design rainfall event using CADDIES model (Guidolin et al., 2016). The design rainfall event was generated using the alternating block method (Te Chow et al., 1988). CADDIES is a cellular-automata-based flood model capable of relatively fast pluvial flood simulations.

For both patch- and resizing-based options, the ground truth was generated per catchment area; 67% catchments from the main dataset were randomly selected as the training sets, whereas the remaining 33% were defined as the test set. All the catchment areas from the validation dataset were used for validation (Figure 2). The CNNs were trained using the training sets and evaluated using both the test set and the validation set. For the baseline experiment, the simulations were conducted using the 1,000 elevation patches, among which 67% patches were the training set and the remaining 33% were the test set. There was no validation set for the baseline experiment. During the training process, data augmentation techniques that randomly flip and rotate the rasters were used to increase the number of training data.

5.2 Tested CNN models

The CNN models tested and compared in this study were named by *input size-kernel size.* The details of these models are shown in Table 1. We propose the *1024-k7* model as our benchmark flood prediction model due to its relatively large input size and receptive field, which reduces information loss and can potentially learn from a larger area. Compared to this model, the other models have smaller receptive fields or smaller input size:

- *1024-k3* model tested the effect of small kernels.
- *1024-plain* model tested the effect of skip connections.
- *512-k7* and *256-k7* models tested different patch size for patch-based option.

Unless otherwise mentioned, all results presented in the paper were obtained by the *1024-k7* model.

All models were implemented using Keras 2.2.2 (Chollet et al., 2015) and Tensorflow 1.14.0 (Abadi et al., 2016), and were trained using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 5×10^{-5}. The batch size for all training was two. We used a small batch size due to the memory limitation of the used graphic card. The mean square loss functions were used during the training step of all the models. All no-data pixels were excluded from the loss functions. Furthermore, as the two options have different number of input-output pairs, i.e., the number of patches (for the patch-based option) is larger than the number of catchment areas (for the resizing-based option). Resizing-based models were trained with more epochs. For all models, we stopped the training process when their test losses converge to stable values.

Table 1. Different models tested in our experiments

Name	Input size	Receptive field	Kernel size	All layers shown in sequence (concatenations are not shown) [1]	Tested in
1024-k7	1,024×1,024	1588	7	**convp**(8); **convp**(16); **convp**(32); **convp**(64); **convp**(128); **convp**(256); 2×**conv**(512); **upconv**(256); **upconv**(128); **upconv**(64); **upconv**(32); **upconv**(16); **upconv**(8); **conv**(2)	All experiments
1024-k3		572	3		
1024-plain	1,024×1,024	1588	7	1024-plain has no skip connections	The baseline experiment
512-k7	512×512	788	7	**convp**(16); **convp**(32); **convp**(64); **convp**(128); **convp**(256); 2×**conv**(512); **upconv**(256); **upconv**(128); **upconv**(64); **upconv**(32); **upconv**(16); **conv**(2)	The patch-based option
256-k7	256×256	388	7	**convp**(32); **convp**(64); **convp**(128); **convp**(256); 2×**conv**(512); **upconv**(256); **upconv**(128); **upconv**(64); **upconv**(32); **conv**(2)	

[1] **conv**(*n*) represents one convolutional layer with the kernel size = 3; **convp**(*n*) is two convolutional layers with the kernel size specified in the network name, followed by one max pooling layer; **upconv**(*n*) is one up-sampling layer followed by two convolutional layers with the kernel size = 3, and *n* is the number of output image channels.

5.3 Model evaluation

The performance of the proposed model was evaluated from the viewpoints of prediction accuracy and computational time. The prediction accuracy was evaluated by calculating the mean absolute error (MAE) between the prediction and the respective ground-truth data and by visually analyzing two-dimensional (2D) error (prediction−simulation) histograms. The MAE assesses the accuracy in general and compares the overall performance of different CNN models tested in the experiments. The MAE has limitations on showing the error distributions when the dataset is imbalanced due to, for example, the different proportion between flooding and non-flooding areas. This issue can be handled by the 2D error histogram which shows the number of raster cells that are y_i by simulation and y_j by prediction by the pixel at row i and column j. The

2D histogram shows the distribution of prediction errors in both shallow and deep-water areas, allowing to analyse if the models tend to under- or over-estimate. Both MAE and 2D error histogram exclude no-data pixels from the assessment results. In addition to these measurements, spatial distribution of errors were also reported.

The computational time was measured by the average prediction time for the entire catchment area. For the baseline experiment and the resizing-based option, the time is equivalent to the process time for one input raster. For the patch-based option, the time depends on how many patches are sampled from the catchment area. The patch sampling process we used determines the patch locations by moving a 1,024×1,024 patch horizontally and vertically with a step of 512 until the entire catchment area is covered. For smaller patch size, the moving step reduces proportionally. In addition to the prediction time, the time for necessary preprocessing and post-processing was also measured.

6. Results

6.1 Evaluating model architecture

The performance of the different prediction models is presented in Table 2, including the MAEs of water depth and flow velocity in test set and validation set. The results show that models that have the same input size are more accurate when the receptive field is larger. This result indicates that the availability of global elevation information is essential for flood predictions. This conclusion is clearer when the results of the patch-based option (*512-k7* and *256-k7*) are compared. Models that process smaller patch size clearly showed higher prediction errors. It can also be seen from the loss curves of the different models (Figure 3) that models with smaller receptive field tend to have larger gaps between the training and test losses. In other words, when two models reach the same training loss (the same accuracy on training data), the one that has larger receptive field has a lower test loss (higher accuracy on test data). This indicates that models without sufficient receptive field tend to "memorize" the training data rather than make good generalization on the test data. The results of the baseline experiment also emphasize the importance of global information. Moreover, the baseline experiment shows that the effect of skip connections is significant.

Table 2. The MAE values for water depth (m) and flow velocity (m/s) of different models on test and validation sets. The values within brackets correspond to the flow velocity.

Name	The patch-based option		The resizing-based option		The baseline experiment
	Test set	Validation set	Test set	Validation set	
1024-k7	0.0132 (0.0290)	0.0219 (0.0313)	0.0193 (0.0447)	0.0250 (0.0445)	0.0212 (0.0705)
1024-k3	0.0158 (0.0351)	0.0227 (0.0319)	0.0227 (0.0559)	0.0290 (0.0556)	0.0246 (0.0812)
1024-plain	-	-	-	-	0.0394 (0.1230)
512-k7	0.0185 (0.0397	0.0186 (0.0259)	-	-	-
256-k7	0.0215 (0.0469)	0.0228 (0.0365)	-	-	-

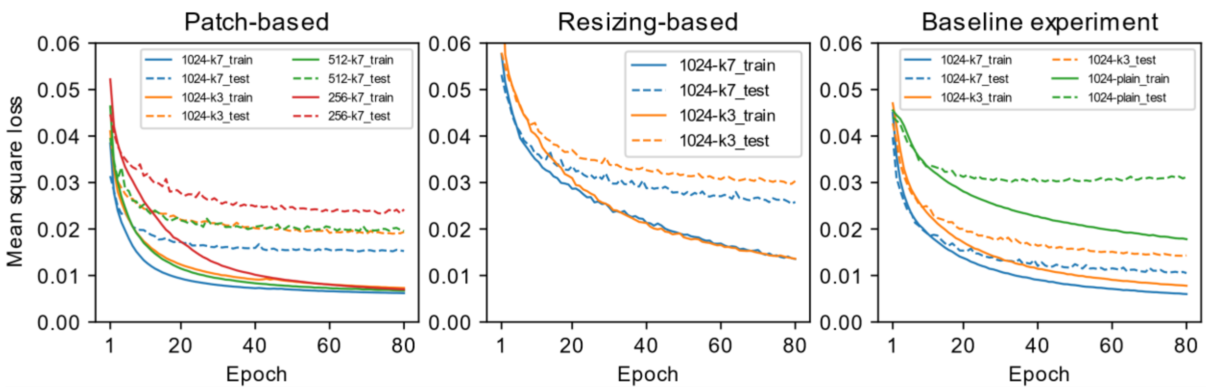

Figure 3. training loss of experiments with different network designs

6.2 Evaluating the prediction accuracy

The error distribution of the results produced by the *1024-k7* model for the two options and the baseline experiment are presented as 2D error histograms in Figure 4. The plot pixel at row i and column j shows the number of raster cells that are y_i by simulation and y_j by prediction. Therefore, a perfect model with no prediction error will produce a histogram in which all pixels except the diagonal are 0. The more diverge the non-zeros pixels from the diagonal, the lower the prediction accuracy. The histograms show that the prediction accuracy obtained is relatively lower in the patch- and resizing-based options when compared with the accuracy obtained using the baseline experiment. This suggests that, as expected, the information loss caused by patch-sampling and resizing reduces the prediction accuracy. Flood predictions on catchment areas that have arbitrary size and irregular boundary is thus more challenging than on terrain data with the same size. However, although the resizing option provides a more “global view” than the patch-based option, the lack of diagonal-liked pattern on validation set indicates that the resizing option does not generalize well. This suggests that learning from elevation data of different scales is more difficult than learning from incomplete elevation data (patches). In addition, both options and the baseline experiment achieved higher accuracy on flow velocity than water depth. This indicates that the flow velocity is affected more by the local elevation pattern than by the global terrain information, thus making flow velocity easier to learn and to predict.

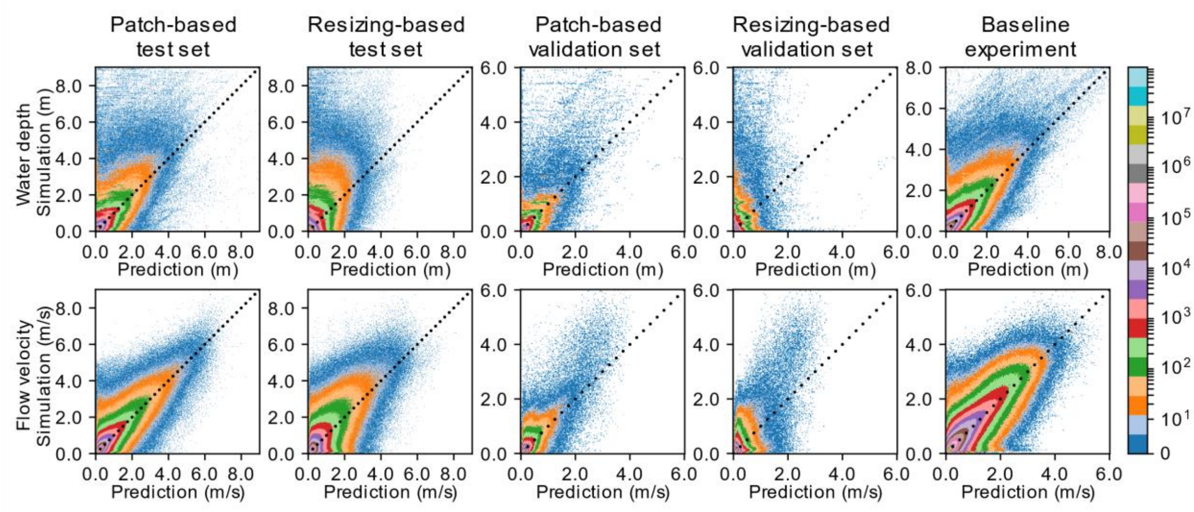

Figure 4. 2D error histograms of all three experiments

6.3 Spatial distribution of errors

Figures 5 and 6 show examples of the 20% most and the 20% least accurate results for the two proposed options, i.e., the results with smaller and larger MAE, respectively. Figure 7 shows three sample results for the baseline experiment. All figures contain enlargements that focus on high-error areas. The spatial patterns between the prediction and the simulation show a high-level visual similarity, suggesting that the neural networks can identify flood extent accurately. The baseline experiment (Figure 7) also suggests that in ideal conditions, i.e., when all terrain data used by physically-based simulations are fed to the CNN model without information loss, the CNN model can approximate the simulator with relatively high accuracy.

Again, as seen in Figure 4, the errors relative to water depth are relatively higher when compared to those of flow velocity. These high-error areas are mainly located in those areas with more than 1 m water depth (Figure 5 and 6). The error of flow velocity in these areas is, however, relatively lower. This observation holds true for the baseline experiment as well (Figure 7). This phenomenon is most likely related to the fact that water accumulation areas can receive water from far upstream regions. Water depth prediction seems to be sensitive to the catchment global elevation information, whereas flow velocity seems to be affected mainly by local elevation patterns, confirming the findings of Tsubaki and Kawahara (2013). Consequently, predicting water depth accurately when elevation data are incomplete is more challenging than predicting flow velocity. Note that all figures use non-linear color maps to visualize small values. The mapping intervals are shown in the color bars. For each interval, linear interpolation is used.

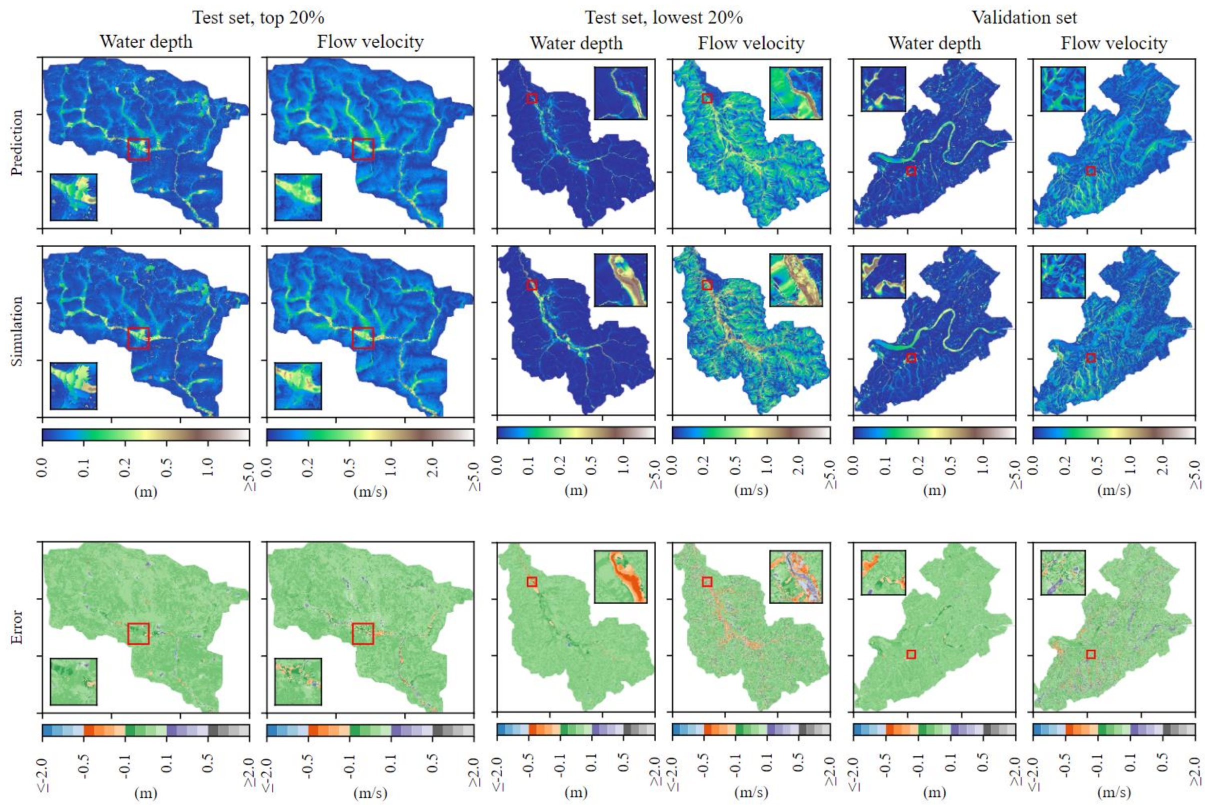

Figure 5. Prediction results of the patch-based option: a 20% most accurate results from test set (left), a 20% least accurate results from test set (middle), and a sample result from validation set (right).

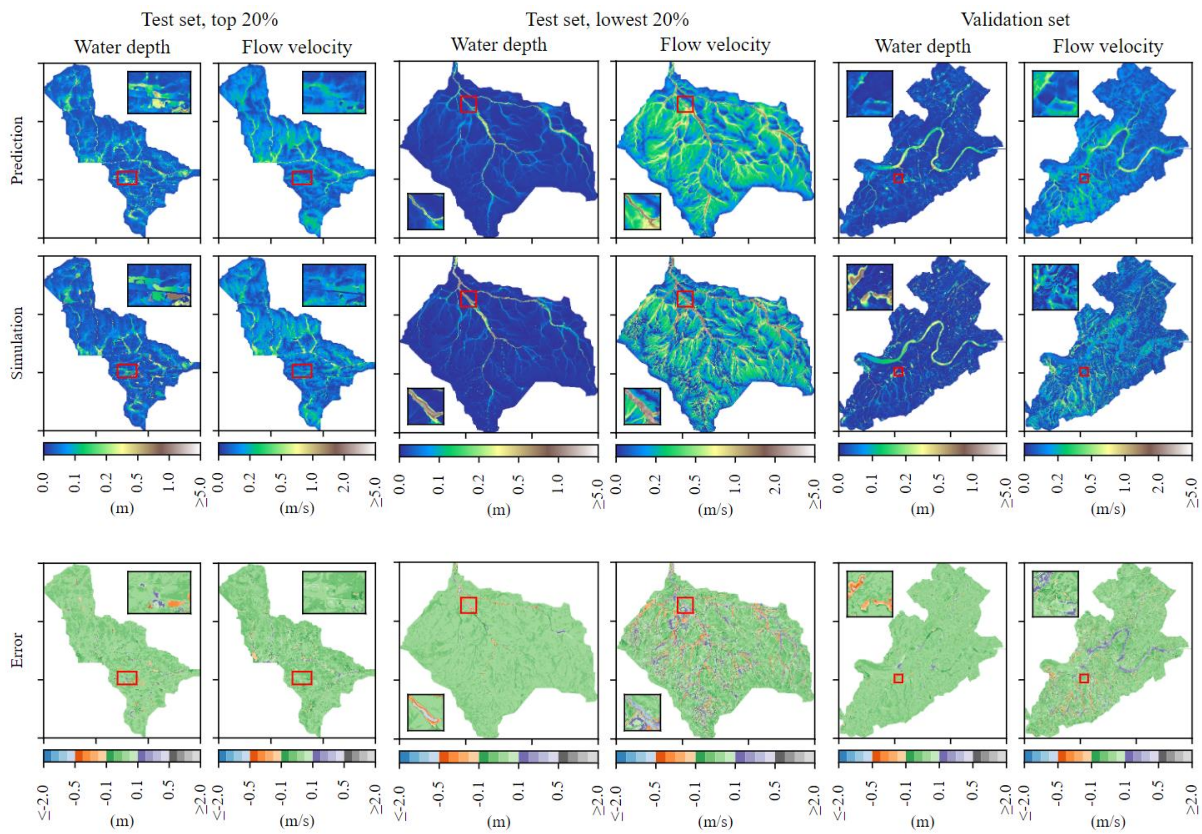

Figure 6. Prediction results of the resizing-based option: a 20% most accurate results from test set (left), a 20% least accurate results from test set (middle), and a sample result from validation set (right).

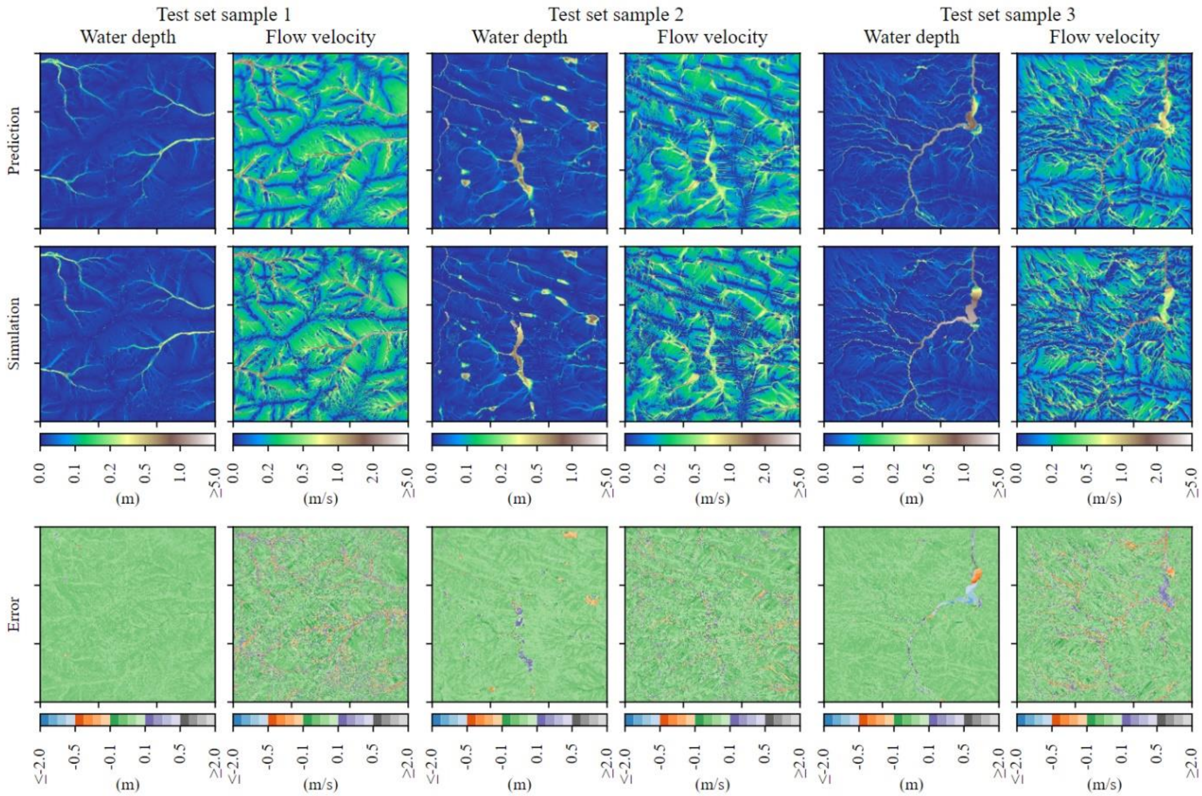

Figure 7. Prediction results of the baseline experiment: three samples from the test set.

In addition, the patch- and resizing-based options show a similar trend of performance on different catchment areas, i.e., both options perform better on some catchment areas than some other catchment areas. This result can be seen in Figure 8, in which the x and y axes represent the MAE values produced by patch- and resizing-based options, respectively. Each dot in the figure represents one catchment area. To further investigate this, case studies were made on four catchment areas (samples A, B, C and D in Figure 8, plotted by flow velocity). These catchment areas were selected as (1) they correspond to higher and lower prediction accuracies, and (2) they are large and contain more terrain features. The case studies suggest that most prediction errors occur in water channels and ponds, and it is not clear that urban areas have in general a lower or higher prediction accuracy when compared to rural areas. The spatial plots of these four catchment areas are presented in Appendix. A hypothesis created from these four case studies is that the lower accuracy on specific catchment areas is due to lack of sufficient terrain variations in our dataset. The results can theoretically be improved if more data are included in the training step. Furthermore, Figure 8 also shows that the resizing-based option tends to be less accurate than the patch-based option, which confirms the previous conclusions.

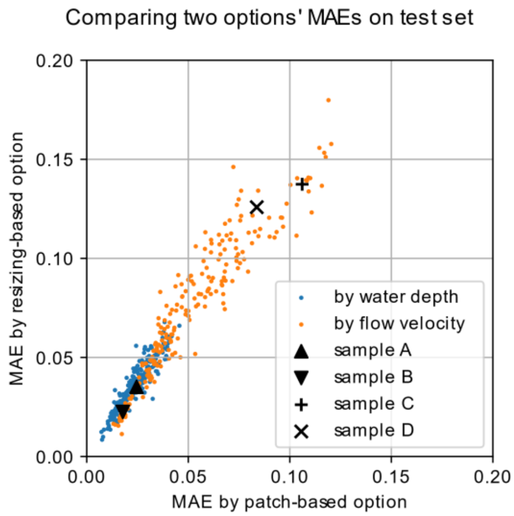

Figure 8. Comparing two options' MAE values, each dot represents one catchment area.

6.4 Evaluating prediction time

The time comparison between physically-based simulations and CNN models is presented in Figure 9 where each point represents one catchment. The x-axes of the plots represent the simulation time and the y-axes show the prediction time. The orange points consider both prediction time and the time for necessary data preprocessing, whereas the blue points consider only the prediction times. The plots clearly show that CNNs achieved a significant improvement on computational speed. Results that take approximately 20,000 seconds by simulations can be obtained by three seconds using CNN based models. For the patch-based option, the prediction time is linearly correlated with the simulation time. This is due to the increasing number of patches sampled from larger catchment areas. For the resizing-based option, the prediction time (blue) is constant, and the data preprocessing time (orange) slightly increases for catchment areas that cost more simulation time, explained by the different size of the catchment areas. The baseline experiment shows that the data-processing time remains constant if all elevation data have same size.

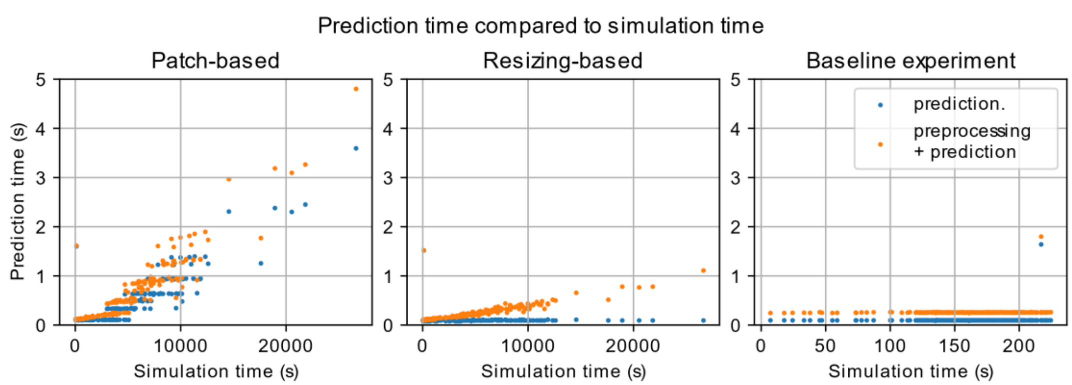

Figure 9. Simulation time vs. prediction times exclude (blue) or include (orange) data processing time, each dot represents one catchment area.

7. Conclusions and Possible Future Research Directions

This study presented a data-driven approach for fast flood prediction using CNNs that is able to generalize on different catchment areas. The study consists of three experiments in which two experiments explored different methods for processing catchment areas and the third one, a theoretical example, serves as the baseline as it is not affected by input data information loss. The results have shown that CNNs exhibit a promising ability to generalize the information learnt from certain terrains to other unseen terrains, suggesting a potential to serve as the "universal" surrogate model for flood predictions of different catchments and scenarios.

The results of the experiments also suggest that the major challenge for data-driven flood prediction is how to systematically encode catchment areas of arbitrary sizes and shapes. Compared to the baseline experiment, both patch- and resizing-based options showed a lower prediction accuracy. The patch-based option showed significantly better performance than the resizing-based option on validation data, which means the information loss caused by subsampling is more critical than by incomplete terrain data. The resizing-based option tends to "memorize" the training data rather than generalize. The high-error areas are more likely to exist around deep-water regions than shallow-water regions. This suggests that water accumulation is sensitive to the global pattern of the catchment area; this was also found by Tsubaki and Kawahara (2013). This conclusion is also supported by the results of the comparison of the different CNN models investigated in our study. Models with larger receptive field and larger input size higher accuracy in all the experiments conducted. Another interesting result is that all experiments achieved higher prediction accuracy on flow velocities than water depths.

The question of how to effectively encode different catchment areas still remains a major challenge. Possible solutions include testing new neural network architecture, modifying the loss functions, or sampling patches based on flow movement rather than spatial locations (e.g., Chu & Thuerey, 2017). Another interesting direction of future research would be to estimate the flow dynamic based on input constraints such as spatial rainfall intensity. Also, the rapid development of sensor networks has made it possible to collect data by crowdsourcing methods (Zheng et al., 2018) or computer vision techniques (e.g., Moy de Vitry et al., 2019; Gebrehiwot et al., 2019), opening new possibilities to produce observational flood data to be used in the training step of data-driven flood prediction models.

Acknowledgments, Samples, and Data

This study was funded by the China Scholarship Council grant 201706090254. The elevation and simulation data of the catchment areas used by this study can be obtained from the data repository (Guo et al., 2020b) hosted by the Research Collection of ETH Zurich with DOI link 10.3929/ethz-b-000453305. The authors declare no conflicts of interest.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th Symposium on Operating Systems Design and Implementation*. USENIX, Savannah, USA. pp. 265-283. doi: 10.1029/2018WR024301

Amaranto, A., Munoz-Arriola, F., Solomatine, D. P., & Corzo, G. (2019). A spatially enhanced data-driven multimodel to improve semiseasonal groundwater forecasts in the High Plains aquifer, USA. *Water Resources Research*, *55*(7), 5941-5961. doi:

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495. doi: 10.1109/TPAMI.2016.2644615

Bui, D. T., Hoang, N. D., Martínez-Álvarez, F., Ngo, P. T. T., Hoa, P. V., Pham, T. D., ... & Costache, R. (2020). A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Science of the Total Environment, 701*, 134413. doi: 10.1016/j.scitotenv.2019.134413

Chang, L. C., Chang, F. J., & Chiang, Y. M. (2004). A two-step-ahead recurrent neural network for stream-flow forecasting. *Hydrological Processes*, *18*(1), 81-92. doi: 10.1002/hyp.1313

Chang, F. J., Chen, P. A., Lu, Y. R., Huang, E., & Chang, K. Y. (2014). Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *Journal of Hydrology*, *517*, 836-846. doi: 10.1016/j.jhydrol.2014.06.013

Chen, P. A., Chang, L. C., & Chang, F. J. (2013). Reinforced recurrent neural networks for multi-step-ahead flood forecasts. *Journal of Hydrology, 497*, 71-79. 10.1016/j.jhydrol.2013.05.038

Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Chu, M., & Thuerey, N. (2017). Data-driven synthesis of smoke flows with CNN-based feature descriptors. *ACM Transactions on Graphics (TOG), 36*(4), 1-14. doi: 10.1145/3072959.3073643

Cramer, S., Kampouridis, M., Freitas, A. A., & Alexandridis, A. K. (2017). An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. *Expert Systems with Applications*, *85*, 169-181. doi: 10.1016/j.eswa.2017.05.029

De Smith, M. J., Goodchild, M. F., & Longley, P. (2007). *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador Publishing Ltd.

Gebrehiwot, A., Hashemi-Beni, L., Thompson, G., Kordjamshidi, P., & Langan, T. E. (2019). Deep convolutional neural network for flood extent mapping using unmanned aerial vehicles data. *Sensors*, *19*(7), 1486. doi: 10.3390/s19071486

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations* (ICLR 2019). New Orleans, USA. Oral presentation. https://openreview.net/forum?id=Bygh9j09KX

Greydanus, S., Dzamba, M., & Yosinski, J. (2019). Hamiltonian neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing System 2019*. Vancouver, Canada, pp. 15353-15363.

Gude, V., Corns, S., & Long, S. (2020). Flood Prediction and Uncertainty Estimation Using Deep Learning. *Water, 12*(3), 884. doi: 10.3390/w12030884

Guo, X., Li, W., & Iorio, F. (2016). Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, New York, USA. pp. 481-490. doi: 10.1145/2939672.2939738

Guo, Z., Leitao, J. P., Simoes, N. E., & Moosavi, V. (2020a). Data-driven Flood Emulation: Speeding up Urban Flood Predictions by Deep Convolutional Neural Networks. *Journal of Flood Risk Management*, e12684. doi: 10.1111/jfr3.12684

Guo, Z., Leitao, J. P., & Moosavi, V. (2020b). Flood simulation data of a 100-year designed storm in 656 catchment areas of Switzerland. *ETH Zurich Research Collection*. doi: 10.3929/ethz-b-000453305

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the 29th IEEE conference on computer vision and pattern recognition (CVPR 2016)*, Las Vegas, USA. pp. 770-778. doi: 10.1109/CVPR.2016.90

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, *Fuzziness and Knowledge-Based Systems, 6*(02), 107-116. doi: 10.1142/S0218488598000094

Huang, S., Chang, J., Huang, Q., & Chen, Y. (2014). Monthly streamflow prediction using modified EMD-based support vector machine. *Journal of Hydrology, 511*, 764-775. doi: 10.1016/j.jhydrol.2014.01.062

Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., & Solenthaler, B. (2019, May). Deep fluids: A generative network for parameterized fluid simulations. *Computer Graphics Forum, 38*, 59-70. doi: 10.1111/cgf.13619

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, USA. Poster presentation.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019a). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology & Earth System Sciences, 23*(12). doi: 10.5194/hess-23-5089-2019

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., & Nearing, G. S. (2019b). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research, 55*(12), 11344-11354. doi: 10.1029/2019WR026065

Leitão, J. P., Zaghloul, M., & Moosavi, V. (2018). Modelling overland flow from local inflows in “almost no-time” using Self-Organizing Maps, In *11th International Conference on Urban Drainage Modelling*, Palermo, Italy. Oral presentation.

Li, B., Huang, C., Li, X., Zheng, S., & Hong, J. (2019). Non-iterative structural topology optimization using deep learning. *Computer-Aided Design, 115*, 172-180. doi: 10.1016/j.cad.2019.05.038

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the 28th IEEE conference on computer vision and pattern recognition (CVPR 2015)*. Boston, USA, pp. 3431-3440.doi: 10.1109/CVPR.2015.7298965

Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona, Spain. pp. 4905–4913.

Masci, J., Boscaini, D., Bronstein, M., & Vandergheynst, P. (2015). Geodesic convolutional neural networks on riemannian manifolds. In Proceedings of the IEEE international conference on computer vision workshops. Santiago, Chile. pp. 37-45. doi: 10.1109/ICCVW.2015.112

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL 2013)*. Atlanta, USA.

Moy de Vitry, M., Kramer, S., Wegner, J. D., & Leitão, J. P. (2019). Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network. *Hydrology and Earth System Sciences*, *23*(11), 4621-4634. doi: 10.5194/hess-2018-570

Mustafa, A., Wei Zhang, X., Aliaga, D. G., Bruwier, M., Nishida, G., Dewals, B., ... & Teller, J. (2018). Procedural generation of flood-sensitive urban layouts. *Environment and Planning B: Urban Analytics and City Science*, doi: 10.1177/2399808318812458

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML 2010)*, Haifa, Israel, pp. 807-814.

Raissi, M., Wang, Z., Triantafyllou, M. S., & Karniadakis, G. E. (2019). Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, *861*, 119-137. doi: 10.1017/jfm.2018.872

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., ... & Kumar, V. (2019). Process-guided deep learning predictions of lake water temperature. *Water Resources Research*, *55*(11), 9173-9190. doi: 10.1029/2019WR024922

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, LNCS, Vol.9351, pp. 234-241. doi: 10.1007/978-3-319-24574-4_28

Sosnovik, I., & Oseledets, I. (2019). Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, *34*(4), 215-223. doi: 10.1515/rnam-2019-0018

Tan, Q. F., Lei, X. H., Wang, X., Wang, H., Wen, X., Ji, Y., & Kang, A. Q. (2018). An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. *Journal of hydrology*, *567*, 767-780. doi: 10.1016/j.jhydrol.2018.01.015

Te Chow, V., Maidment, D.R., Mays, L.W. (1988). *Applied Hydrology*. McGraw-Hill. ISBN: 0-07-100174-3

Tehrany, M. S., Pradhan, B., & Jebur, M. N. (2013). Spatial prediction of flood susceptible areas using rule based decision tree (DT) and a novel ensemble bivariate and multivariate statistical models in GIS. *Journal of Hydrology, 504*, 69-79. doi: 10.1016/j.jhydrol.2013.09.034

Tehrany, M. S., Shabani, F., Neamah Jebur, M., Hong, H., Chen, W., & Xie, X. (2017). GIS-based spatial prediction of flood prone areas using standalone frequency ratio, logistic regression, weight of evidence and their ensemble techniques. *Geomatics, Natural Hazards and Risk*, *8*(2), 1538-1561. doi: 10.1080/19475705.2017.1362038

Tehrany, M. S., Jones, S., & Shabani, F. (2019). Identifying the essential flood conditioning factors for flood prone area mapping using machine learning techniques. *Catena, 175*, 174-192. doi: 10.1016/j.catena.2018.12.011

Thuerey, N., Weißenow, K., Prantl, L., & Hu, X. (2020). Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows. *AIAA Journal, 58*(1), 25-36. doi: 10.2514/1.J058291

Tompson, J., Schlachter, K., Sprechmann, P., & Perlin, K. (2017). Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*. Sydney, Australia. pp. 3424-3433

Tsubaki, R., & Kawahara, Y. (2013). The uncertainty of local flow parameters during inundation flow over complex topographies with elevation errors. *Journal of Hydrology, 486*, 71-87. doi: 10.1016/j.jhydrol.2013.01.042

Turner, S. W., Doering, K., & Voisin, N. (2020). Data-Driven Reservoir Simulation in a Large-Scale Hydrological and Water Resource Model. *Water Resources Research, 56*(10). doi: 10.1029/2020WR027902

Wang, Y., Fang, Z., Hong, H., & Peng, L. (2020). Flood susceptibility mapping using convolutional neural network frameworks. *Journal of Hydrology*, *582*, 124482. doi: 10.1016/j.jhydrol.2019.124482

Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. Montreal, Canada. pp. 802-810.

Zaghloul, M. (2017). *Machine-Learning aided Architectural Design - Synthesize Fast CFD by Machine-Learning*. Phd Diss. ETH Zurich. doi: 10.3929/ethz-b-000207226

Zhao, G., Pang, B., Xu, Z., Peng, D., & Xu, L. (2019). Assessment of urban flood susceptibility using semi-supervised machine learning model. *Science of The Total Environment, 659*, 940-949. doi: 10.1016/j.scitotenv.2018.12.217

Zhao, G., Pang, B., Xu, Z., Peng, D., & Zuo, D. (2020). Urban flood susceptibility assessment based on convolutional neural networks. *Journal of Hydrology, 590*, 125235. doi: 10.1016/j.jhydrol.2020.125235

Zheng, F., Tao, R., Maier, H. R., See, L., Savic, D., Zhang, T., et al. (2018). Crowdsourcing methods for data collection in geophysics: State of the art, issues, and future directions. *Reviews of Geophysics, 56*, 698– 740. doi: 10.1029/2018RG000616

Appendix

The case studies of four selected catchment areas are presented in Figures A1 to A4. Each figure contains three columns that correspond to the simulation results, the patch-based predictions and the resizing-based predictions. The first and second rows of each figure are water depth results and errors. The third and last rows of each figure are flow velocity results and errors. The elevation data are attached at the bottom right corner of each figure.

Figures A1 and A2 show the case studies that correspond to sample A and B in Figure 8. The two samples represent rural and urban areas, respectively. For these samples, both patch- and resizing-based options reached a relatively high accuracy. On a detailed level, the patch-based option made less mis-prediction than the resizing-based option in terms of numerical errors and visual patterns. An example is presented through the enlargement areas of both samples (Figures A1 and A2) where the difference of visual patterns can be clearly seen. Figure A1 shows another mis-prediction of resizing-based option. The mis-prediction is located on the right side of the red rectangle, near the boundary of the catchment area. The resizing-based option consider this location filled with deep waters (last plot of the first row), whereas these waters do not exist at the same location of the simulation result and the patch-based options (first two plots of the first row).

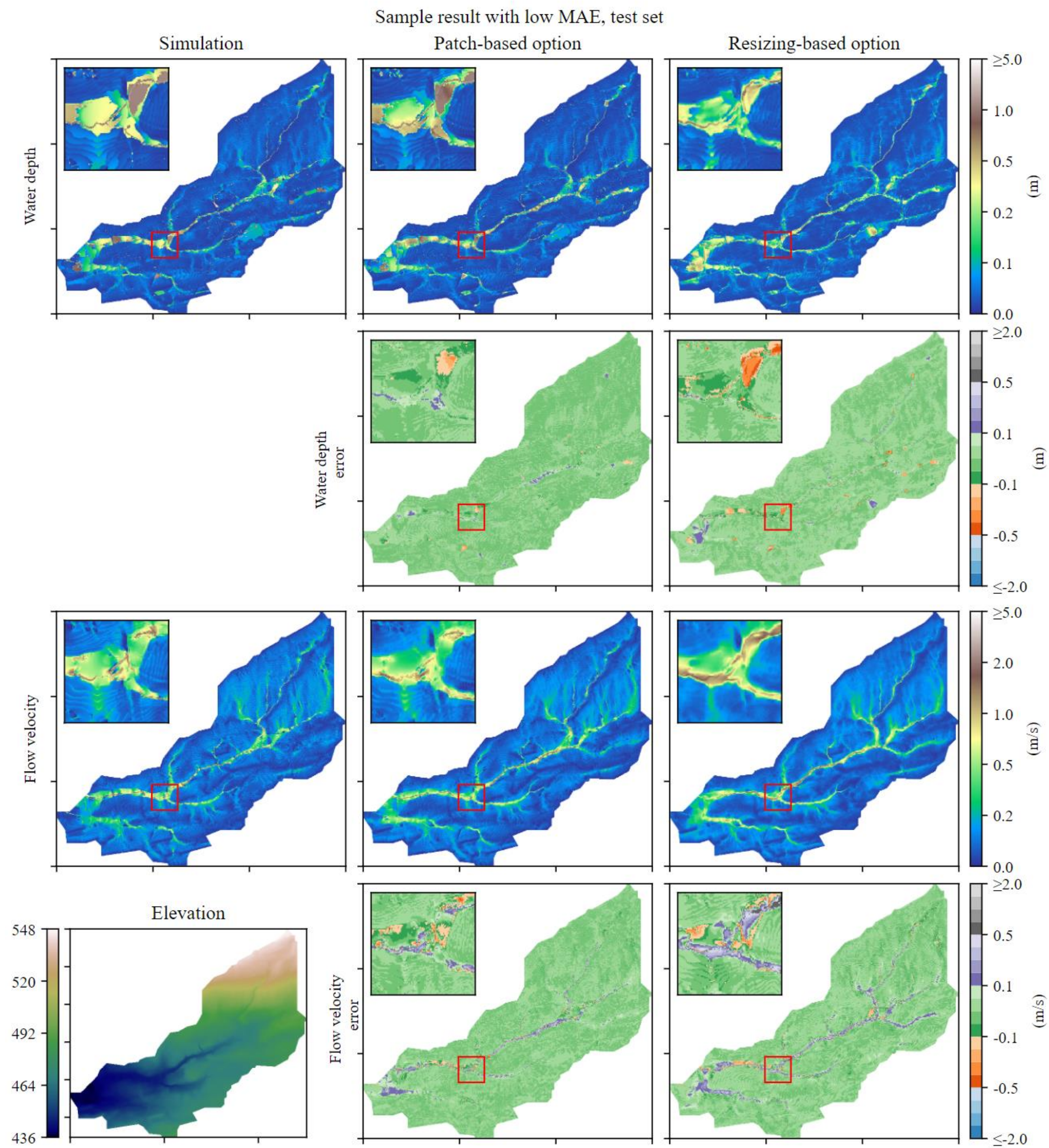

Figure A1. Case study sample A

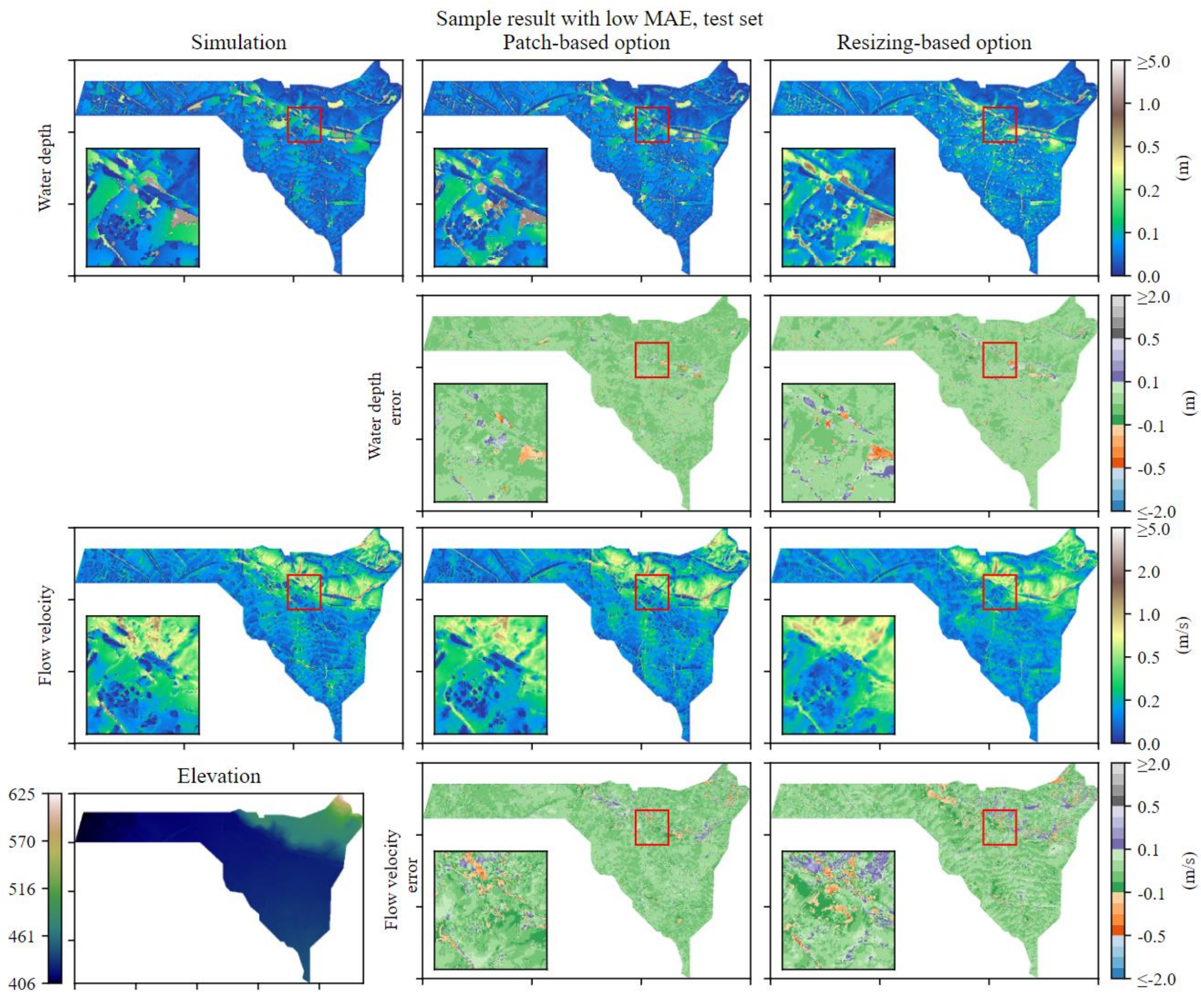

Figure A2. Case study sample B

Figures A3 and A4 show the case studies of sample C and D in Figure 8. These two samples also represent rural and urban areas. Compared with samples A and B, the prediction accuracy obtained for samples C and D is lower. The most significant misprediction shown in the enlargement area of sample C is the missing flood in the urban area (Figure A3). This misprediction exists in both patch- and resizing-based options. For sample D (Figure A4), the patch-based option successfully identified the flood extent of the downstream areas, whereas the resizing-based option mis-predicted most downstream areas.

As can be seen in Figure 8, the model tends to perform better in certain catchment areas and worse in other catchment areas; however, the analysis of the four sample areas does not suggest a clear relationship between the prediction accuracy and certain terrain features or type of area (i.e. urban and rural areas). In most cases, the path-based option outperforms the resizing-based option, whereas for certain catchment areas, both options show less accurate results at same locations. We suspect that this is due to lack of sufficient terrain feature variations in our dataset, making the model failing to generalize on these catchments.

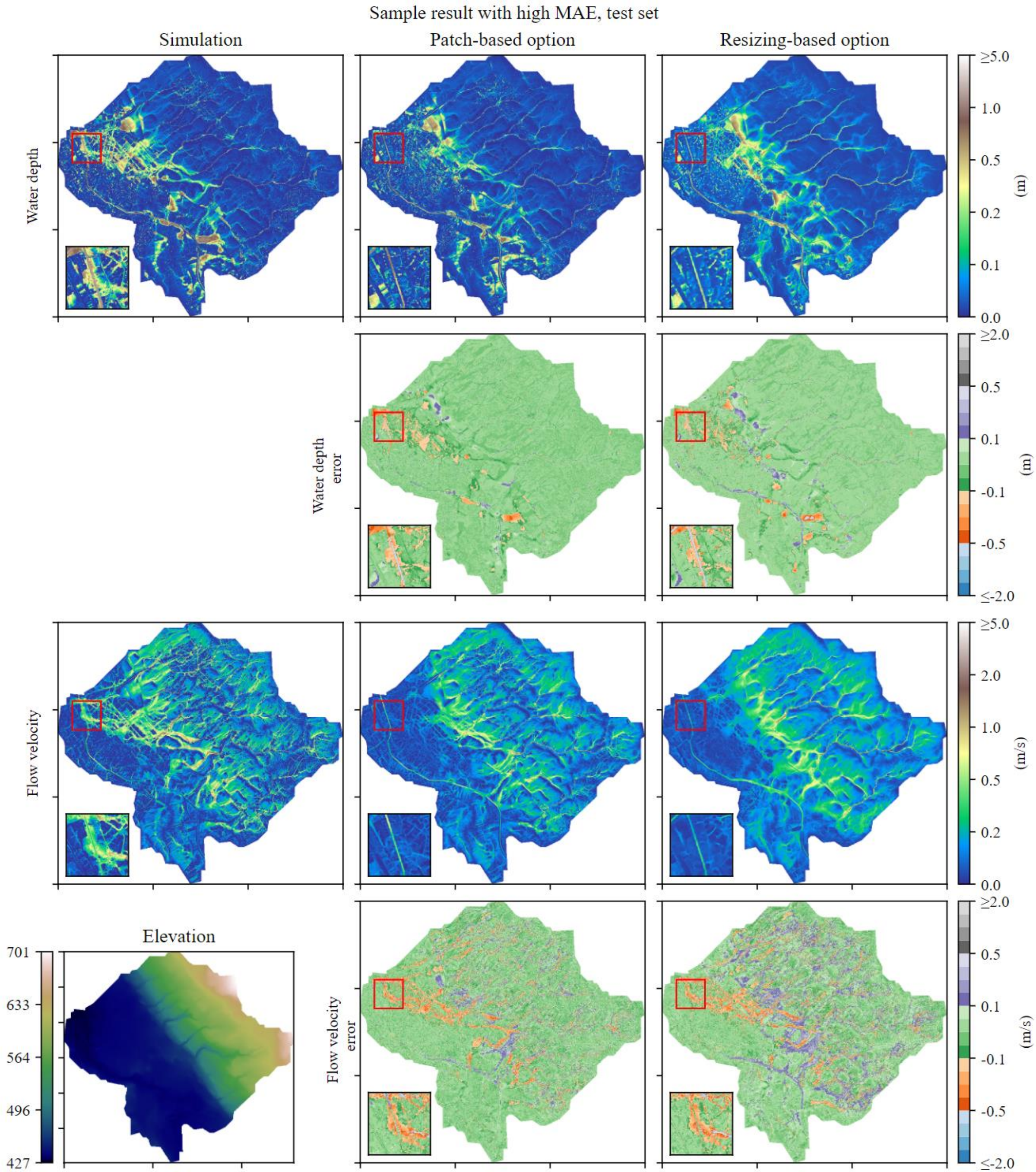

Figure A3. Case study sample C

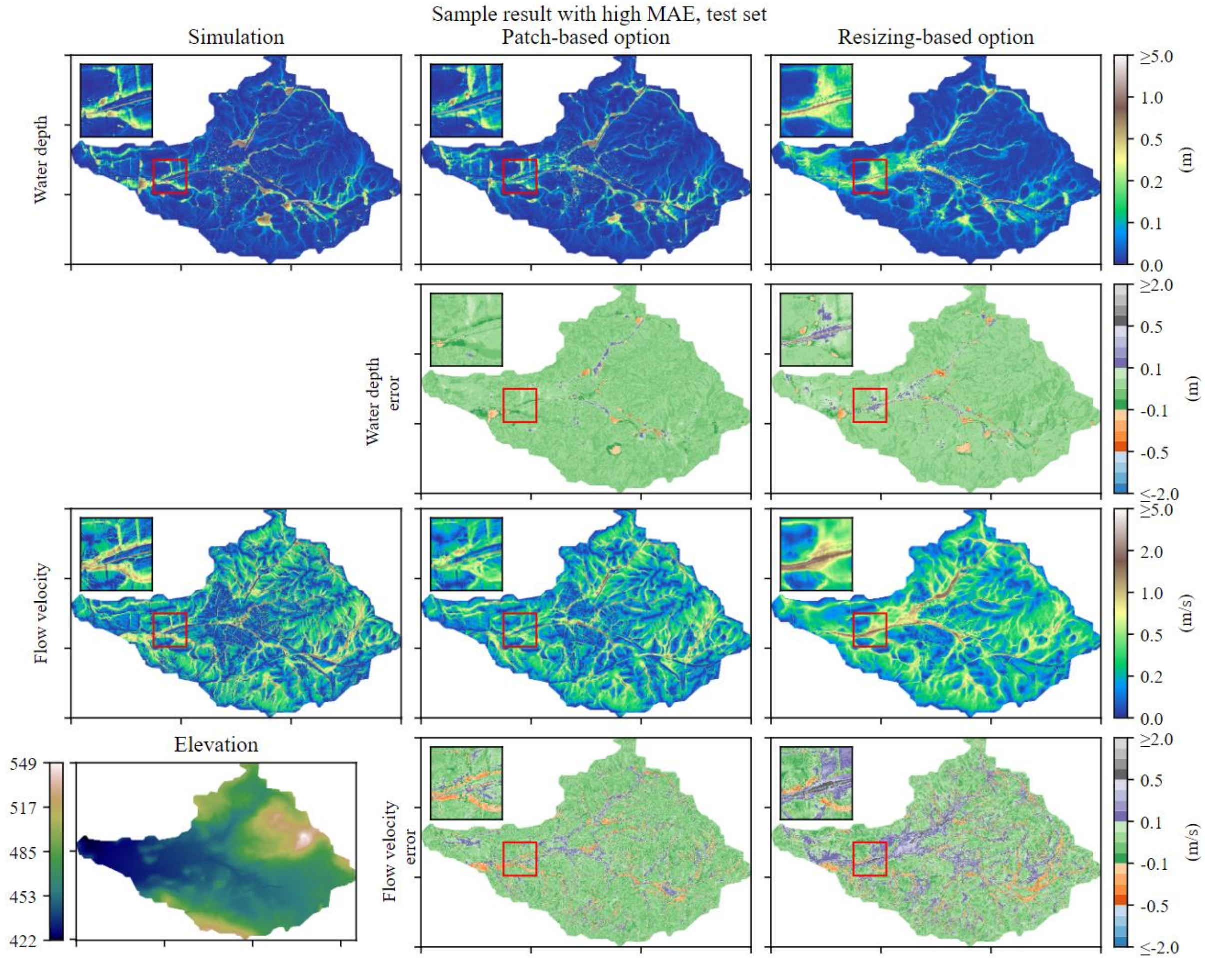

Figure A4. Case study sample D

As mentioned in Section 4.2, several tests with and without terrain features were conducted for the baseline experiment to determine the optimal data-processing parameters. The results of these tests are presented as loss curves in Figure A5, in which the left plot shows the result of different *c* values, and the right plot shows the result of different terrain features. It is clear from the left plot that the model converges faster as the *c* decreases. However, the improvement on convergence speed becomes less significant when $c < 0.01$. The right plot suggests that models using multiple features converge faster than those using one feature or those without any feature.

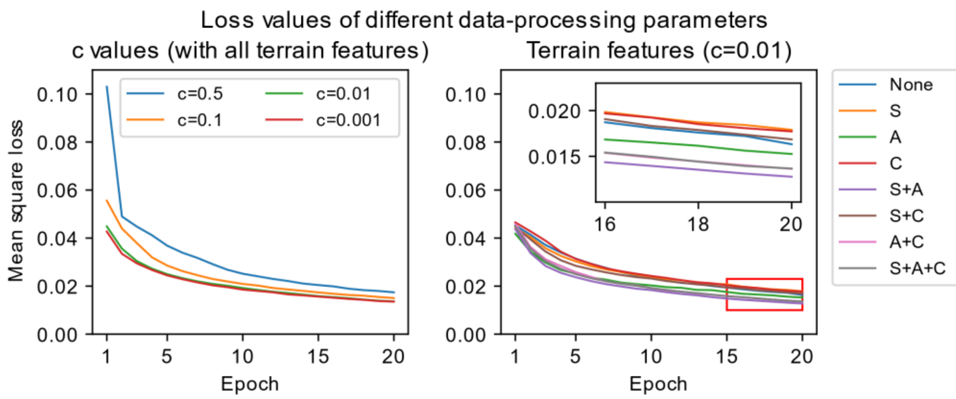

Figure A5. The effect of data-processing parameters on model convergence. The S, A, and C represent slope, aspect, and curvature, respectively.