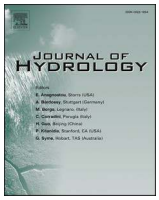
Journal of Hydrology 635 (2024) 131169

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol



Research papers

Real-time flood maps forecasting for dam-break scenarios with a transformer-based deep learning model

Matteo Pianforini [a,*], Susanna Dazzi [a], Andrea Pilzer [b], Renato Vacondio [a]

[a] Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy
[b] NVIDIA AI Technology Center, Italy

ARTICLE INFO

This manuscript was handled by A. Bardossy, Editor-in-Chief, with the assistance of Kourosh Behza, Associate Editor

Keywords:
Real-time flood forecasting
Data-driven model
Inundation maps
Dam-break
Deep learning

ABSTRACT

This paper presents a purely data-driven deep-learning approach for flood maps forecasting. For the first time in this context a Transformer-based algorithm is employed to address one of the main issues in early-warning systems for flood propagation, i.e., the long computational times required to forecast the inundation evolution in real time. The proposed model, named "FloodSformer", is trained to extract the spatiotemporal information from a short sequence of water depth maps and predict the water depth map at one subsequent instant. Then, to forecast a sequence of future maps, we employ an autoregressive procedure based on the trained surrogate model. The method was applied to both synthetic dam-break scenarios and to a real case study, specifically the ideal failure of the Parma River dam (Italy). The training and testing datasets were generated numerically from two-dimensional hydraulic simulations. In the case of the real test case, the average Root Mean Square Error was found to be equal to 10.4 cm. The short computational time (e.g., the forecast of 90 maps, representing a lead time of 3 h, takes less than 1 min) makes the FloodSformer model a suitable tool for real-time emergency applications.

1. Introduction

Floods are natural disasters that cause huge economic damages and casualties. In the last 50 years, 44 % of the natural disasters occurred in the world have been associated with floods. These catastrophes generated over 330′000 deaths and 1 trillion US$ of economic losses in the period 1970–2019 (WMO, 2021). The negative consequence of inundations can be mitigated with structural and non-structural flood risk management strategies. Focusing on river floods, one of the most important and adopted non-structural strategy is the early-warning system based on real-time forecasting of hydrological variables. This methodology, together with efficient emergency action plans, can drastically reduce the impact of extreme floods (Plate, 2002).

Zounemat-Kermani et al. (2020) distinguished three categories of mathematical models for the simulation and forecasting of hydrological variables: physically based, conceptual, and "black-box" models. The first type of models solves partial differential equations describing the physical process in the domain, but the high computational cost of two-dimensional (2D) models with high resolution (1–5 m) and three-dimensional (3D) models prevents their use for real-time forecasting. To overcome this drawback, in the last years, researchers have focused on the parallelization of numerical models, exploiting the high computational efficiency of modern Graphics Processing Units (GPUs). For example, Vacondio et al. (2014) implemented a parallelized 2D Shallow Water Equations (SWE) solver gaining a speedup of two orders of magnitude compared with serial codes. More recently, Ming et al. (2020) developed a forecasting system coupling a numerical weather prediction model with a GPU accelerated hydrodynamic model. Considering a catchment of 2′500 km^2 discretized with a spatial resolution of 10 m (25 M cells), they obtained a ratio of physical to computational time around 20 using 8 GPUs. This suggests that, despite all the efforts on parallelization techniques, the computational cost of physically based models remains significant and requires access to High-Performance Computing (HPC) clusters (Ming et al., 2020; Turchetto et al., 2020), thus preventing their widespread adoption for real-time predictions.

Differently from physically-based schemes, the conceptual models use simplified relationships that emulate the physics of the phenomena, resulting in significantly shorter computational times. However, these models require different physical information, which may not always be

* Corresponding author.
E-mail address: matteo.pianforini@unipr.it (M. Pianforini).

https://doi.org/10.1016/j.jhydrol.2024.131169

Available online 5 April 2024

available, and the calibration of many parameters. Additionally, their ability to simulate the process of flood propagation is limited, particularly in cases involving complex topographies and flood events where momentum conservation is important (Teng et al., 2017). Consequently, conceptual models are often employed for the forecasting of the maximum extent of floods.

Finally, "black-box" models, frequently called "data-driven" or "surrogate" models (Bentivoglio et al., 2022), completely neglect the physical background of the process and learn the nonlinear relationship between the input and output variables directly from observed data. One of the most important advantages of surrogate models is the high computational efficiency. For this reason, in the present work, we focus on data-driven models for flood maps forecasting.

Mosavi et al. (2018) and Bentivoglio et al. (2022) summarized the commonly used machine learning (ML) and deep learning (DL) methods for hydraulic/hydrodynamic studies. They distinguished between two types of prediction tasks. The first is the study of the temporal variation of hydraulic variables (e.g., discharge and/or river stage) in a specific river section, in which the input data to the surrogate model are the rainfall observations in the upstream catchment (Hu et al., 2018; Yin et al., 2022) and/or the water stages observed at the target station and at upstream river sections at previous instants (e.g., Bomers, 2021; Castangia et al., 2023; Dazzi et al., 2021b). The second category of works, to which the present study belongs, focuses on predicting the spatial distribution of flooding in a specific domain. In turn, data-driven models for this purpose can be divided into three categories: the first are the ones that produce flood probability maps, which represent the flood hazard adopting geo-environmental characteristics (e.g., Panahi et al., 2021). Other surrogate models are trained to generate maximum water depths and/or velocities maps, neglecting the time evolution (e.g., Bermúdez et al., 2019; Hofmann & Schüttrumpf, 2021). Finally, the most recent data-driven models forecast the spatial and temporal evolution of inundation for pluvial (Liao et al., 2023) or river floods (e.g., Fraehr et al., 2023; Kabir et al., 2020; Zhou et al., 2022). The knowledge of spatiotemporal information is extremely important to enhance the resilience of flood-prone areas and implement effective emergency measures in real time. For this reason, in the last few years, the use of DL methods dedicated to this task has gained attention in literature. For example, Kabir et al. (2020) used a 1D Convolutional Neural Network (CNN) to predict water depths for a fluvial inundation in a domain discretized with 0.5 million cells. The flood map forecast at a specific instant is obtained from the surrogate model using only the inflow discharge values at previous time-steps as input data. Differently, Zhou et al. (2022) considered the boundary condition inflows to predict the water depths in about 21 k representative locations. Then, these forecasted values were used to reconstruct the flood surface for the entire domain. It follows that, in these two studies, the water depths in the floodplain are predicted considering only the temporal variation of the upstream discharge, whereas the initial conditions are neglected. Moreover, the spatiotemporal correlation between consecutive inundation maps is not considered. Fraehr et al. (2023) adopted a different approach. They utilized a hydrodynamic model with reduced spatial resolution (low-fidelity model) to simulate the flood propagation with a lower computational demand. Then, employing a ML-based model (i.e., Sparse Gaussian Process), they enhanced the spatial resolution of the output maps generated by the low-fidelity model, with the aim of reproducing the outputs of a high spatial resolution hydrodynamic model (high-fidelity model). The bottleneck of this hybrid model lies in the computational time and stability of the low-fidelity model (Fraehr et al., 2023). Therefore, to increase the efficiency, it is advisable to employ a rapid and accurate hydrodynamic model.

The present study focuses on the analysis of inundations resulting from dam-break scenarios, a type of natural disaster characterized by a rapid and unexpected flood propagation downstream of the dam. The release of a considerable volume of water from the dam typically results in an outflow discharge significantly higher than that expected during most severe river floods. Consequently, dam-break events often lead to overflows and subsequent inundation of floodplains. In the realm of hydraulic dam safety studies, there is a lack of real-time forecasting models that specifically address dam-break scenarios. While the literature includes approaches for the rapid prediction of the outflow discharge from reservoirs (e.g., Ma & Fu, 2012), they often do not extend to forecasting the consequent flooding in the downstream area. Furthermore, previous research dedicated to the study of dam-break floods using data-driven models have been restricted to 1D analysis (Li et al., 2023a) or to the development of emulators for computational fluid dynamics (CFD) models, specifically for multiphase flows (Boosari, 2019) and fluid–structure interactions (Li et al., 2023b) for synthetic dam-breaks. Notably, these types of surrogate models are not applicable for the forecasting of real dam-break scenarios. The use of data-driven models for dam-break floods prediction is still an unexplored topic.

Over the last few years, new types of flexible and efficient DL models that consider spatiotemporal information (e.g., Graph Neural Network (GNN) and Transformer) have been presented. For example, a preliminary application of GNN to predict inundation maps on randomly generated bathymetries was presented by Bentivoglio et al. (2023). Another promising method is the Transformer architecture, originally proposed by Vaswani et al. (2017) for natural language processing. Recently, Transformer-based models have brought significant progress in different tasks including image (e.g., Dosovitskiy et al., 2020) and video (e.g., Bertasius et al., 2021) classification, and video frame prediction (e.g., Ye & Bilodeau, 2023). Differently from other types of DL models, the Transformer is based on multi-headed self-attention mechanism, which allows modelling long-range dependencies and attending to different space–time information of the input sequence. Moreover, unlike Recurrent Neural Networks (RNN), Transformer's attention mechanism is parallelizable by design (Vaswani et al., 2017) allowing to scale to thousands of GPUs. Studies concerning the application of this type of data-driven model to video future frames prediction tasks have shown promising results compared to other DL models (e.g., Convolutional Long-Short-Term Memory). In flood forecasting, only a handful of works have taken advantage of Transformer-based architecture. For example, Yin et al., (2022,2023) built two Transformer-based rainfall-runoff models for runoff prediction. Liu et al. (2022) used a double-encoder Transformer to forecast the monthly river streamflow based on the past water levels and other climatological information, while Castangia et al. (2023) proposed a Transformer-based neural network to predict the daily average water level in a river station one day ahead, using the past daily average water levels of upstream hydrological stations. Xu et al. (2023) demonstrated the capability of attention mechanism to make long-time series predictions of lake water level fluctuations. These studies have shown that, in general, Transformer-based models have better forecasting capability compared to well-known RNN (e.g., Long-Short-Term Memory) and CNN models.

This brief literature review shows that Transformer-based models have shown promising results in forecasting tasks. However, previous analyses were limited to time series prediction of scalar values whereas, to the best of our knowledge, no investigations have been conducted with such models on the forecast of 2D inundation maps. In this work, we developed a model named FloodSformer (FS), based on autoencoder (AE) and Transformer architectures for the prediction of the temporal evolution of flood maps. Differently from other works, the FS framework allows extracting the spatiotemporal information from a sequence of consecutive water depth maps (past frames) and predicting water depth maps of subsequent instants (future frames). The model has been tested considering three different test cases of dam-break flows: two dam-breaks over synthetic bathymetries and the hypothetical failure of the Parma River flood mitigation dam (Italy). Most of the previous studies on inundation maps prediction used physically based models to generate the ground-truth maps adopted as samples for the data-driven model (Bentivoglio et al., 2022). This approach allows providing potentially limitless data, overcoming the problem of scarcity of directly observed

data. For this reason, the same strategy was used here, and the dataset samples used to train and validate the FS model were generated through a hydrodynamic model (i.e., PARFLOOD, Vacondio et al., 2014).

The rest of the paper is organised as follows: in Section 2 the surrogate model is described, while the case studies and the model setup are illustrated in Section 3. Section 4 is dedicated to the presentation and discussion of the main results. Finally, conclusions are drawn in Section 5.

2. The FloodSformer model

In this Section, we present the FloodSformer surrogate model (Fig. 1). This data-driven model uses an autoencoder (AE) to analyse spatial information in the input maps and incorporates a video prediction Transformer (VPTR) framework to consider the spatiotemporal relationships between consecutive maps. The design of this surrogate model draws inspiration from transformer-based models employed for video frame prediction (e.g., Ye & Bilodeau, 2023). Given that inundation maps can be seen as matrices with size $H \times W$, in which the value of

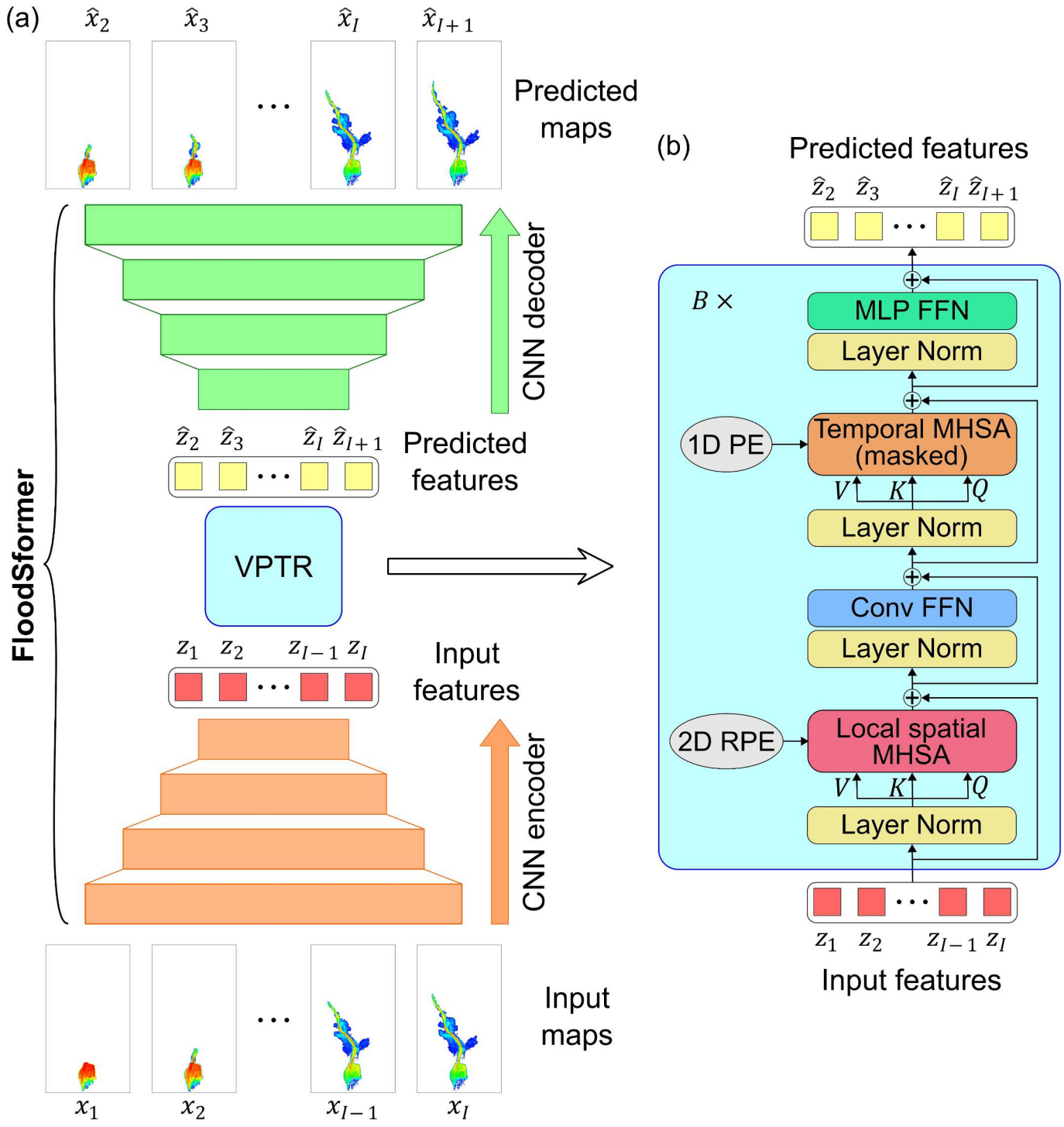


Fig. 1. Architecture of FloodSformer model. (a) General workflow of the proposed model. (b) Sketch of one of the B VidHRFormer blocks forming the video prediction Transformer (VPTR) module.

each pixel corresponds to the water depth in a computational cell of a Cartesian grid, DL algorithms developed for video prediction can be adapted to deal with flood maps. In the FS model, we adopt the same approach presented by Ye & Bilodeau (2023), in which the future frames prediction processes are entrusted to three consecutive blocks (Fig. 1a): an encoder (2D CNN), a video prediction Transformer framework (VPTR), and a decoder (2D CNN). The first and the last block constitute the classical ResNet-based AE from the Pix2Pix model (Isola et al., 2017), while the second block is the fully autoregressive VPTR model based on the Video High-Resolutions Transformer (VidHRFormer) block proposed by Ye & Bilodeau (2023), described in Section 2.2. The encoder uses a sequence of consecutive 2D CNN layers to extract only spatial information from the maps and reduce their dimensionality. The compressed output matrices, named latent features, are sent to the VPTR framework, which analyses the spatiotemporal information and predicts the latent features of the future frames. Finally, the decoder is used to reconstruct the predicted maps, starting from the forecasted latent features. The integration of the AE architecture allows reducing the dimension of the maps that the VPTR block must process, therefore decreasing the memory and time consumption required for its training.

The general workflow of the FS model is illustrated in Fig. 1a. Considering a sequence of $I+1$ frames, the CNN encoder takes as input the first I ground-truth maps ($t = 1, \cdots, I$) and, for each of them, extracts the spatial information creating the latent features. These are used by the VPTR framework to predict the latent features at next instants ($t = 2, \cdots, I+1$). Finally, the predicted maps are reconstructed by the CNN decoder. The training strategy of the FS model is presented in Section 2.3.

This model only provides the predicted map at one subsequent time step ($t = I+1$). Then, an autoregressive procedure, described in detail in Section 2.4, is exploited to continue forecasting for further time steps, by using the predicted frames as input for additional runs of the surrogate model. In this way, the model can be applied to predict many future frames using only a limited number of input maps, paving the way for real-time flood forecasting.

2.1. Notation

The structure of the FS model for the training and testing processes can be summarized with the following three equations:

$$\boldsymbol{z}_t = Enc(\boldsymbol{x}_t), \quad t \in [1, \cdots, I] \tag{1}$$

$$\widehat{\boldsymbol{z}}_t = \mathscr{T}(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{t-1}), \quad t \in [2, \cdots, I+1] \tag{2}$$

$$\widehat{\boldsymbol{x}}_t = Dec(\widehat{\boldsymbol{z}}_t), \quad t \in [2, \cdots, I+1] \tag{3}$$

where:

- $\boldsymbol{x} \in \mathbb{R}^{H \times W}$ is the ground-truth map.
- $\widehat{\boldsymbol{x}} \in \mathbb{R}^{H \times W}$ is the predicted map.
- $\boldsymbol{z} \in \mathbb{R}^{h \times w \times d_{model}}$ is the input latent feature of the Transformer blocks.
- $\widehat{\boldsymbol{z}} \in \mathbb{R}^{h \times w \times d_{model}}$ is the output latent feature of the Transformer blocks.
- $Enc(\cdots)$ is the encoder block.
- $Dec(\cdots)$ is the decoder block.
- $\mathscr{T}(\cdots)$ is the VPTR framework.
- H is the number of map cells along the south-north direction.
- W is the number of map cells along the west-east direction.
- $h = H/2^k$ is the height of the latent feature.
- $w = W/2^k$ is the width of the latent feature.
- k is the number of convolutional layers of the AE.
- d_{model} is the number of channels of the latent feature.
- I is the number of input frames for the training and testing procedure.

In the following, we will also use the notations P and F to identify the number of past and future frames for real-time forecasting applications, respectively.

2.2. Transformer model

The self-attention (SA) mechanism is the key to Transformer model (Vaswani et al., 2017). This technique allows to capture global dependencies of an input sequence $\boldsymbol{Z}$ by evaluating the dot-product between the query $\boldsymbol{Q}$ and the key $\boldsymbol{K}$ matrices, scaled by a softmax function, and multiplying it to the value $\boldsymbol{V}$ matrix:

$$SA(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{4}$$

where $\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{W}_{\boldsymbol{Q}}$, $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{W}_{\boldsymbol{K}}$, $\boldsymbol{V} = \boldsymbol{Z}\boldsymbol{W}_{\boldsymbol{V}}$ are matrices with dimensions d_q, d_k, d_v, respectively. These matrices are linearly transformed by the sequence $\boldsymbol{Z}$ through learnable parameters matrices $\boldsymbol{W}_{\boldsymbol{Q}}$, $\boldsymbol{W}_{\boldsymbol{K}}$, $\boldsymbol{W}_{\boldsymbol{V}}$. Furthermore, multiple parallel SA computations, called "heads", are concatenated and once again projected to obtain the multi-head self-attention (MHSA) mechanism:

$$MHSA(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat\left(SA_1, \cdots, SA_p\right)\boldsymbol{W}_{\boldsymbol{mhsa}} \tag{5}$$

where $\boldsymbol{W}_{\boldsymbol{mhsa}}$ is the projection matrix and p is the number of heads. This allows to use multiple heads to attend to information from different representation subspaces simultaneously (Vaswani et al., 2017).

In the original VidHRFormer block (Ye & Bilodeau, 2023), represented in Fig. 1b, the SA computation is separated in space and time using two different layers: a local spatial MHSA and a temporal MHSA. This procedure reduces the overall complexity of a standard Transformer. The VidHRFormer block is then completed with a convolutional feed-forward neural network (Conv FFN), followed by a multilayer perceptron (MLP) and normalization layers. To prevent the prediction at specific time from being conditioned by the maps of the subsequent instants, masking is applied to the attention mechanism in the temporal MHSA layer. As proposed by Ye & Bilodeau (2023), we use a 2D relative positional encoding (RPE) for the spatial MHSA and a fixed absolute 1D positional encoding (PE) for the temporal MHSA. In the FS model, the VPTR module is composed of B consecutive VidHRFormer blocks. For further details, the reader is referred to Ye & Bilodeau (2023).

2.3. Training strategy

The training of the FloodSformer model is conducted using a large dataset constituted of consecutive water depth maps generated by a hydrodynamic model.

The FS framework is a large model with about 135–350 million parameters depending on the use case (see Section 3.5). Consequently, to reduce the memory and time consumption required for computations, we divided the training process in two stages, which are here briefly recalled. First, we address the feature extraction task by training only the AE. Then, we freeze the AE parameters and train the space–time forecasting block (VPTR). As presented in detail in Section 3, each phase of the training process is based on a specific combination of the samples constituting the training dataset. For the AE training, no temporal information is required and consequently the batch is composed of randomly selected single frames. Differently, for the VPTR training, sequences of $I+1$ consecutive maps are used to create the batch.

In the first stage (AE training), for each map $\boldsymbol{x}$ in the training dataset, the encoder extracts the latent feature z and the decoder tries to reconstruct the input map $\widehat{\boldsymbol{x}}$. The AE training procedure aims at minimizing the following loss function:

$$L_{AE} = L_{MSE} + \lambda_{GDL} L_{GDL} + \lambda_{GAN} \; arg \; \min_{G} \max_{D} L_{GAN}(D, G) \tag{6}$$

in which:

$$L_{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}^n - \widehat{\boldsymbol{x}}^n)^2 \tag{7}$$

$$L_{GDL} = \frac{1}{N}\sum_{n=1}^{N}\left[\sum_{i=1}^{W}\sum_{j=1}^{H}\left(\left|\left|\boldsymbol{x}^n_{i,j} - \boldsymbol{x}^n_{i-1,j}\right| - \left|\widehat{\boldsymbol{x}}^n_{i,j} - \widehat{\boldsymbol{x}}^n_{i-1,j}\right|\right|^{\alpha} + \left|\left|\boldsymbol{x}^n_{i,j-1} - \boldsymbol{x}^n_{i,j}\right| - \left|\widehat{\boldsymbol{x}}^n_{i,j-1} - \widehat{\boldsymbol{x}}^n_{i,j}\right|\right|^{\alpha}\right)\right] \tag{8}$$

$$L_{GAN}(D,G) = \mathbb{E}_X[log(D(\boldsymbol{X}))] + \mathbb{E}_{\widehat{X}}[log(1 - D(G(\boldsymbol{X})))] \tag{9}$$

where L_{MSE} is the mean square error (MSE) loss and L_{GDL} is the image gradient difference loss (GDL), which allows preserving the sharpness of the predicted map (Mathieu et al., 2016). L_{GAN} is the generative adversarial network (GAN) loss, which is composed by the generator G (combination of encoder and decoder) and the PatchGAN discriminator D (Isola et al., 2017). $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ and $\widehat{\boldsymbol{X}} = \{\widehat{\boldsymbol{x}}_1, \cdots, \widehat{\boldsymbol{x}}_N\}$ are the original and reconstructed maps, respectively. N denotes the number of samples in the dataset. λ_{GDL}, λ_{GAN} and α are hyperparameters (see Section 3.5).

In the second stage (VPTR training), the AE model parameters are frozen (i.e., we use the optimized values obtained from the first stage), and only the parameters of the VPTR module are optimized (Fig. 2a). The model aims at extracting the spatiotemporal information from a sequence of I input frames to reproduce the flood map at time $t = I+1$. To improve the temporal prediction performance, input frames within the range $2 \leq t \leq I$ are also forecasted and used to update the weights of the Transformer, minimizing the following loss function:

$$L_{VPTR} = \frac{1}{I}\sum_{t=2}^{I+1} L_{MSE}(\boldsymbol{x}_t, \widehat{\boldsymbol{x}}_t) + \lambda_{GDL}\frac{1}{I}\sum_{t=2}^{I+1} L_{GDL}(\boldsymbol{x}_t, \widehat{\boldsymbol{x}}_t) \tag{10}$$

At the end of the training phase, the test dataset is used to evaluate the efficiency and accuracy of the FS model in forecasting unseen sequences of maps.

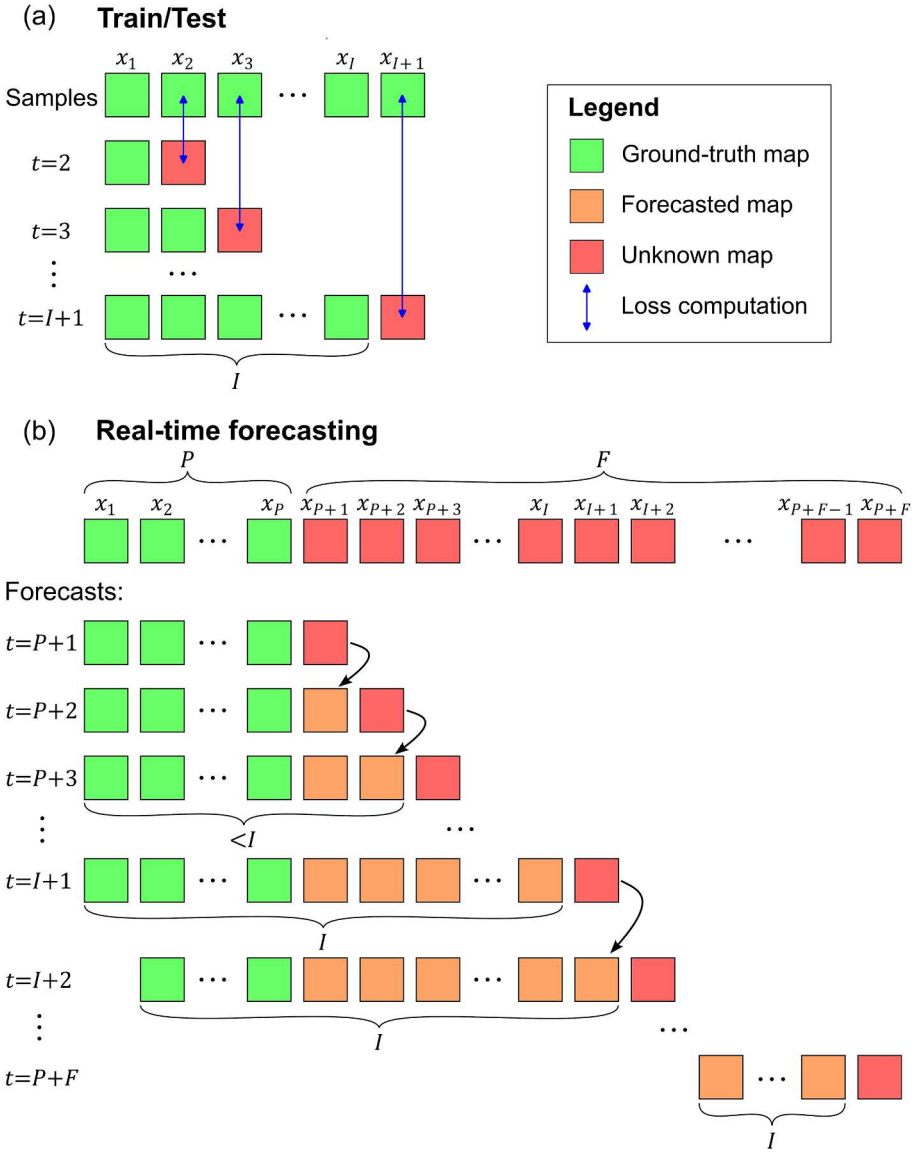


Fig. 2. (a) Sketch of the prediction method for VPTR train and FS test on a sequence of $I+1$ frames. The forecast of the next frame (red square) is achieved considering as inputs all the ground-truth maps of the precedent instants (green squares). (b) Procedure for the autoregressive real-time forecasting of F future maps. Each forecasted future frame (orange square) is used to make the prediction of the next future maps (red squares). In this illustration we assumed $P < I < F$ and, for simplicity, we represented only the input and the output maps of the prediction, neglecting the latent feature computations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.4. Real-time forecasting with autoregression

Once trained, the FS model allows predicting only one frame ahead ($t = I + 1$). However, the application of the surrogate model to real-time flood forecasting cannot be effective unless many frames ahead could be predicted providing only a limited number of input maps. Consequently, an autoregressive procedure was implemented for this purpose. The idea is to exploit the FS model, already trained with a specific value of I, to forecast F future maps by providing only P past ones (with $1 \leq P \leq I$; the influence of P on the prediction accuracy is analysed in Section 4.3.3). The forecast of $F > 1$ future maps is achieved as sketched in Fig. 2b: first, the ground-truth past maps $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_P\}$ (green squares) are used to predict the first unknown map in the future $\widehat{\boldsymbol{x}}_{P+1} = Dec\left(\mathscr{T}\left(Enc\left(\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_P\}\right)\right)\right)$ (red square at $t = P + 1$). Then, the forecasted map $\widehat{\boldsymbol{x}}_{P+1}$ (orange square at $t = P + 2$) is concatenated at the end of the sequence that is fed to the FS model to predict the next map $\widehat{\boldsymbol{x}}_{P+2}$ (red square at $t = P + 2$), and so on. In these first few iterations of the recursive procedure, the VPTR block works even if the number of maps is less than I (its maximum number of input maps according to the training phase), thanks to the approach adopted for training, in which intermediate maps are also predicted and included in the loss computation (Eq. (10)). After a few iterations, when the sum of past maps P and concatenated ones exceeds I, the oldest maps are discarded to constrain the length of the input sequence to I. For example, the prediction of the future frame at instant $t = I + 2$ is computed as follows (see Fig. 2b):

$$\widehat{\boldsymbol{x}}_{I+2} = Dec(\mathscr{T}(Enc(\{\boldsymbol{x}_2, \cdots, \boldsymbol{x}_P, \widehat{\boldsymbol{x}}_{P+1}, \cdots, \widehat{\boldsymbol{x}}_{I+1}\}))) \tag{11}$$

After $I + P$ iterations, all ground-truth maps have been discarded, and all predictions are based on forecasted maps. For example, the prediction of the future frame at instant $t = j$, with $j > I + P$, is computed as follows:

$$\widehat{\boldsymbol{x}}_j = Dec\left(\mathscr{T}\left(Enc\left(\{\widehat{\boldsymbol{x}}_{j-I}, \cdots, \widehat{\boldsymbol{x}}_{j-1}\}\right)\right)\right) \tag{12}$$

This recursive procedure is then repeated until all the F future maps are predicted. Finally, we emphasize that each predicted latent feature is decoded and then encoded back into the latent space before using it to forecast the next temporal frame. This modality strongly reduces the accumulation of error, which characterizes autoregressive schemes (Ye & Bilodeau, 2023).

2.5. Metrics for surrogate model evaluation

To investigate the capability of the FS model to emulate the ground-truth maps, we employed two metrics, the root-mean-square error (RMSE) an the F1 score, defined as follows:

$$RMSE = \sqrt{\frac{1}{N \times T} \sum_{n=1}^{N} \sum_{t=1}^{T} \left(\boldsymbol{x}_t^n - \widehat{\boldsymbol{x}}_t^n\right)^2} \tag{13}$$

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{14}$$

where:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$\boldsymbol{x}_t^n$ and $\widehat{\boldsymbol{x}}_t^n$ are the ground-truth and predicted maps, respectively. N denotes the number of wet cells for the specific map, T is the number of temporal frames. TP is the number of true positives (wet cell both in target and predicted maps), FP is the number of false positives (wet cell in predicted map and dry cell in target map), and FN is the number of false negatives (dry cell in predicted map and wet cell in target map). For the FS test, the RMSE is computed considering only the frame $t = I + 1$ and consequently $T = 1$ in Equation (13). Differently, for the real-time forecasting application, all the future frames are considered ($t \in [P + 1,\ P + F]$). Therefore, the number of temporal frames T is equal to the number of future frames F.

A water depth threshold ε_{wet} was adopted to distinguish between wet and dry cells when calculating the F1 score. For the case studies here considered, the ratio $\varepsilon_{wet}/H_{max}$ ranged from 0.25 % to 0.35 %, where H_{max} represents the maximum water depth expected in the dataset. Additionally, in the RMSE computation (Eq. (13)), we exclusively accounted for cells with a water depth exceeding ε_{wet} in either the ground-truth or the predicted maps. This implies that only cells classified as TP, FP and FN were considered for the RMSE computation. This methodology was adopted to prevent the artificial reduction of the RMSE in scenarios where maps predominantly consist of true negatives TN (dry cells both in target and predicted maps), characterized by null errors.

The RMSE is a regression metric as it considers the differences between target and predicted water depths. Differently, the F1 score (Eq. (14)) is a classification metric used to distinguish between flooded and non-flooded cells and provides an estimate of how well the model can predict the flooded area extent. Differently from precision (Eq. (15)) and recall (Eq. (16)), F1 is suitable even for imbalanced datasets (i.e., in which samples are maps with almost all cells either dry or wet, i.e., TN or TP), because it equally considers both false positives and false negatives, thus avoiding the overestimation of the prediction score.

3. Case studies and model setup

In this Section, a short introduction to the PARFLOOD physically based model, employed only to generate the dataset to train and evaluate the FS model, is provided. Moreover, the three case studies used to train and evaluate the data-driven model (Table 1) are presented. Firstly, a dam-break in a channel with a parabolic cross section is simulated (Section 3.2); followed by a dam-break inside a rectangular tank (Section 3.3). Finally, the last test case considered is the hypothetical collapse of the flood control reservoir dam of the Parma River (Section 3.4). While the first two test cases have simple geometry and bathymetry, in the last one the presence of a real bathymetry together

Table 1
Case studies summary.

	Dam-break in a parabolic channel	Dam-break in a rectangular tank	Dam-break of the Parma River flood-control dam	
			Resolution 20 m	Resolution 10 m
Case study number	1	2	3a	3b
Spatial resolution [m]	10	0.01	20	10
Number of cells	122′880	32′768	114′688	458′752
Temporal resolution [sec]	20	0.02	120	
Samples of the train/ validation/test datasets	504/26/ 106	1082/56/121	532/28/91	
Initial water levels for train/ validation datasets [m a.s.l.]	87–95 every 2 m	0.06–0.20 every 0.02 m	98–100–102–104–105.6	
Initial water level for test dataset [m a.s.l.]	90	0.15	105	
Maximum water depth in test dataset [m]	10.0	0.15	14.2	
Normalization value [m]	16.0	0.22	16.0	
ε_{wet} [m]	0.05	0.0005	0.05	

with urban environments produces a rather complicated flood evolution.

For each case study, the dataset included water-depth maps obtained by running different simulations with the PARFLOOD code, changing the initial water level in the upstream reservoir (Table 1). Furthermore, the output maps of one specific simulation were selected as samples for the test dataset, while the maps of the remaining simulations were randomly split to create the training and validation datasets, with a proportion of 95 % and 5 %, respectively (Table 1). The test dataset was also used for the real-time forecasting application in which the autoregressive procedure was considered.

As already mentioned in Section 2.3, the training procedure is divided into two stages. In the first stage (AE training), the temporal information is neglected, and consequently each batch is composed of randomly selected single frames $\boldsymbol{X} \in \mathbb{R}^{N\times H\times W}$, where N is the batch size. Instead, in the second training stage (VPTR training) the temporal information is relevant, and consequently the batches are generated considering sequences of maps at consecutive instants $\boldsymbol{X} \in \mathbb{R}^{N\times T\times H\times W}$, where T is the total number of frames in the sequence.

For each case study, we considered a specific temporal resolution for the output maps in the dataset (Table 1). Indeed, it is important to choose an appropriate temporal spacing between consecutive frames to ensure that dynamic changes are neither excessively large nor too small.

In Table 1, the value of the water depth threshold ε_{wet}, adopted to distinguish between wet and dry cells (see Section 2.5), is provided for each case study. Please notice that, while cases 1 and 3 are at the field-scale, case 2 is conducted at the laboratory scale, resulting in water depths two orders of magnitude lower. Thus, some parameters, including ε_{wet}, require appropriate scaling.

3.1. Hydrodynamic model (PARFLOOD) for dataset generation

PARFLOOD is a 2D model that solves the fully dynamic SWE with an explicit finite volume scheme (Vacondio et al., 2014). The model is

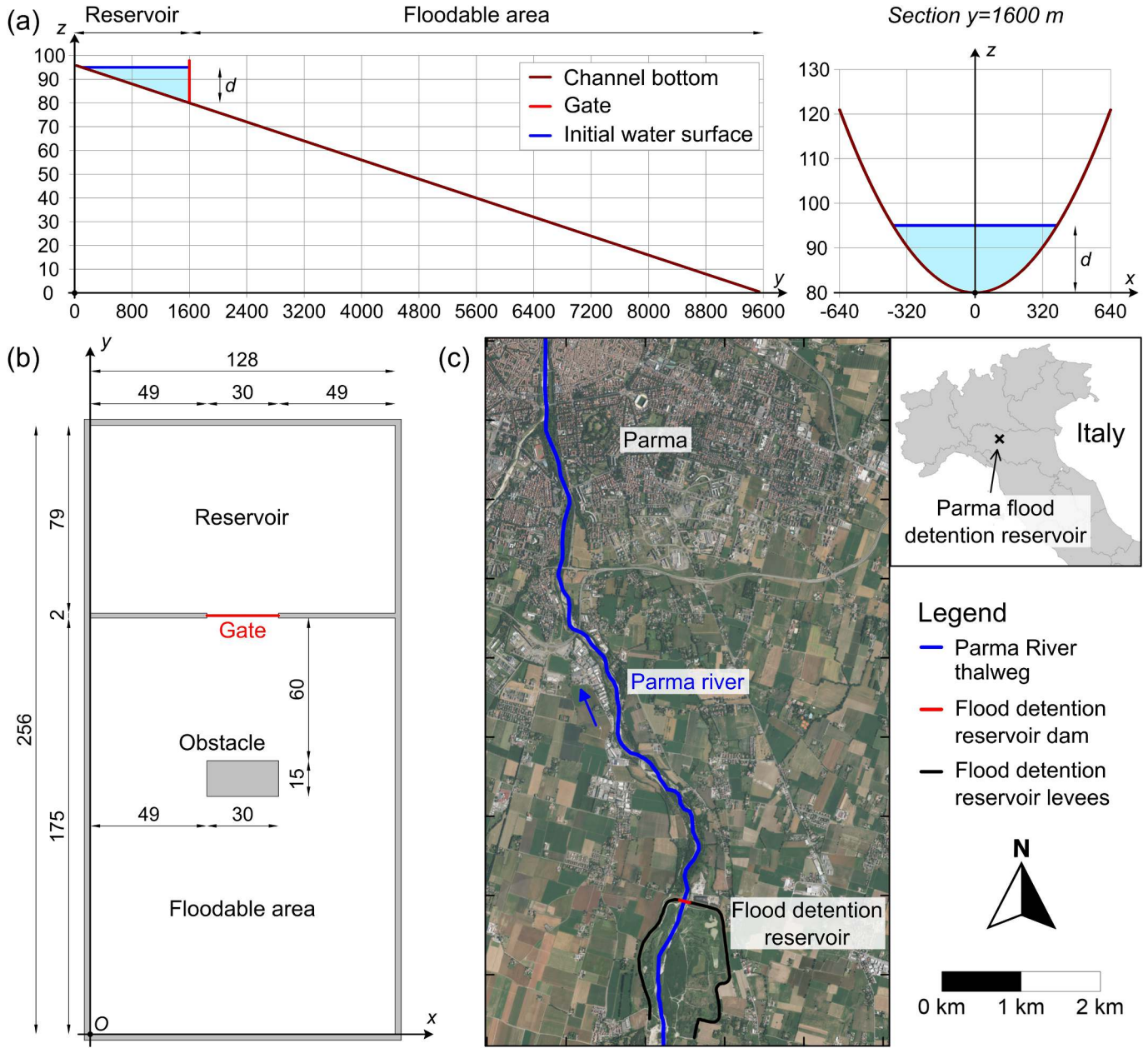


Fig. 3. (a) Case study 1: sketch of the non-horizontal channel with a parabolic cross section (measures in m). (b) Case study 2: sketch of the rectangular tank domain (measures in cm). (c) Study area for the hypothetical collapse of the Parma River flood control reservoir dam (case studies 3a and 3b).

efficiently parallelized using the Computer Unified Device Architecture (CUDA) language, so it can take advantage of the computational power of GPU, drastically reducing the simulation time compared to serial codes (Vacondio et al., 2014). The model's accuracy and efficiency have been extensively tested for challenging case studies, such as synthetic test cases (Vacondio et al., 2014, 2017), river floods (Dazzi et al., 2021a), dam-break inundations (Vacondio et al., 2014) and levee-breach floods (Dazzi et al., 2021a, 2022). In the present work, the PARFLOOD model is used adopting Cartesian grids. For further details on the model description, the reader is referred to Vacondio et al., (2014,2017).

3.2. Case 1: Dam-break in a parabolic channel

A dam-break in a non-horizontal channel with a parabolic cross section was considered as first test case. The domain is 9600 m long and 1280 m wide, and the channel has a 1 % slope (Fig. 3a). Consequently, the terrain profile can be described by means of the following equation:

$$z = 96 + 0.0001x^2 - 0.01y,\ x \in [-640, 640],\ y \in [0, 9600] \quad (17)$$

The parabolic channel is divided into two areas separated by a vertical gate placed at coordinate $y = 1600$ m. The upstream region ($0 < y < 1600$ m) represents the reservoir, while the remaining area ($1600 < y < 9600$ m) identifies the downstream channel. The 12.3 km^2 domain was uniformly discretised with a 10 m resolution grid, consisting of 122′880 cells.

For the hydrodynamic model simulations, we set a uniform Manning roughness coefficient equal to 0.05 $sm^{-1/3}$ and a uniform flow condition was imposed as downstream boundary condition. A constant water surface elevation in the upstream reservoir was adopted as initial condition, whereas the downstream channel was considered initially completely dry.

Five simulations were performed considering an initial water surface elevation in the upstream reservoir in the range between 87 and 95 m (corresponding to maximum water depths between 7 and 15 m, marked as d in Fig. 3a), with steps of 2 m (Table 1). In addition, the case of a water surface elevation of 90 m in the reservoir was also simulated, and the output maps were used as samples for the test dataset. To reproduce the dam-break scenarios, we assumed an instantaneous removal of the gate, and the simulation was concluded when the upstream reservoir was completely emptied, and the flood reached the downstream boundary condition (i.e., between 25 and 50 min after the dam-break). A temporal resolution of 20 s was adopted for the output maps, ensuring an accurate depiction of the flood dynamics.

3.3. Case 2: Dam-break in a rectangular tank

The second test case focuses on a dam-break flow against an obstacle at the laboratory scale, loosely inspired by the experimental facility described by Aureli et al. (2008). The rectangular tank, with dimensions 2.56 × 1.28 m and flat bottom, is divided into two compartments separated by a wall with a 0.30 m wide gate placed in the middle (Fig. 3b). The upstream compartment represents a reservoir with dimension 0.79 × 1.28 m, while the remaining area of the tank identifies the downstream flood plain. A prismatic block with rectangular base 0.30 m wide and 0.15 m long and vertical walls is placed 0.60 m downstream the gate. When the flow impacts this non-submersible obstacle, a hydraulic jump and multiple wave reflections are generated. While this case study is not an experimental analysis, the analogy with the laboratory cases of Aureli et al. (2008) suggests that the flood propagation in the tank can be effectively replicated using a 2D-SWE model, as demonstrated in their study. Therefore, the PARFLOOD code was employed to generate water depth maps, serving as ground truth data.

The study area was discretised with a spatial resolution of 0.01 m and consequently the computational grid had 32′768 cells. Given the similarity with the work by Aureli et al. (2008), the same Manning roughness coefficient was adopted, i.e., 0.007 $sm^{-1/3}$. As initial conditions for the hydrodynamic model, we considered 8 different uniform water levels in the upstream reservoir, in the range between 6 and 20 cm, with steps of 2 cm (Table 1). In addition, the 15 cm water level was simulated to create the test dataset. The downstream floodable area was initially dry in all different simulations. To reproduce the dam-break scenarios, we assumed an instantaneous removal of the gate, and we stopped the simulation when the water was reflected by the downstream wall, occurring between 2.2 and 3.9 s after the dam-break. The surrogate model dataset was composed of temporally consecutive water depth maps with a temporal resolution of 0.02 s.

3.4. Case 3: Dam-break of the Parma river flood-control dam

The third test case focuses on the hypothetical collapse of the Parma River flood control reservoir dam. The in-line detention reservoir is located a few kilometres upstream from the city of Parma (Italy) and it has a storage capacity of about $12{\cdot}10^6$ m^3. The maximum retaining water level is 105.6 m a.s.l., which corresponds to a maximum water depth of about 14.5 m just upstream of the dam. The study domain covers about 45 km^2 and corresponds to the floodable area between the control reservoir and the Parma city centre (Fig. 3c). Differently from the previous case studies, here a real bathymetry is considered.

For the topography, we used two digital terrain models (DTMs) with spatial resolution of 10 m (458′752 cells) and 20 m (114′688 cells), derived from a LiDAR survey. A uniformly distributed value of the Manning coefficient equal to 0.05 $sm^{-1/3}$ was adopted. In the downstream section, a far-field boundary condition was imposed. As initial conditions, we considered different water levels in the upstream reservoir (Table 1): values within the range of 98 to 104, with increments of 2 m, were used for the numerical simulations to generate samples for the training and validation datasets. Additionally, the maximum retaining water level of the Parma River Dam (105.6 m a.s.l.) was incorporated to create the training/validation samples. The inclusion of the maximum expected water level in the upstream reservoir is useful to avoid the issue of extrapolating beyond the range of the training data when the model is applied for predictions. Finally, an initial level of 105 m a.s.l. was selected to simulate water depth maps for the test dataset. We considered an initially dry downstream river region. We set a specific simulation time (between 2 and 3 h) for each initial condition to ensure that the flood propagation in the study area had ended (i.e., emptying of the reservoir and reaching of the maximum flood extent). The water depth maps were sampled at 2-min intervals. This time is adequate to correctly reproduce the rapid flood propagation generated by a real dam-break.

3.5. Implementation details and hyperparameters definition

As suggested by Ye & Bilodeau (2023), in the present model 12 VidHRFormer Transformer blocks, with 8 parallel attention heads and a local patch size equal to 4, were adopted. The output layer of the network was the Sigmoid function (which returns values in range between 0 and 1). This activation function, together with a suitable dataset normalization, automatically prevents the formation of nonphysical negative water depths. The datasets were normalized to the interval (0, 1) to ensure proper functioning of the Sigmoid, and to align the water depth values for different test cases. All the samples in each dataset were divided by a value slightly higher than the maximum water depth simulated by the PARFLOOD model for the specific case study (Table 1). This normalization strategy aims to prevent the potential saturation of the activation function, particularly for values near the dataset's maximum water depth.

In this work, the dataset samples consist of raster grids with a significantly larger number of cells than the 64 × 64 pixels of the video clips used to train the original VPTR model (up to two orders of

magnitude larger). For this reason, to limit the latent features dimension, the original number of convolutional layers of the autoencoder (i.e., $k = 3$) was increased to 4 and 5. Additionally, we adjusted the number of channels in the latent feature d_{model} to 512 and 768, depending on the study case (see Table 2). These adaptations, together with the use of separated space–time SA computation and patches in the local spatial MHSA calculation, are the keys to overcome the quadratic complexity of the SA computation, and consequently to reduce the training time and the GPU memory consumption.

For the first training stage (AE training), we used the Adam optimizer with beta values of (0.5, 0.999) and learning rates of 2e-4 for case studies 1, 3a, and 3b, and 1e-5 for case study 2, while a learning rate of 1e-3 was adopted in the second training phase (VPTR training). Furthermore, an early stopping technique was introduced for the Transformer training stage to halt the training process if the model's performance did not improve after a specified number of epochs (e.g., 20), in order to prevent overfitting. During the surrogate model training and inference, we considered a batch size in the range 4–12, depending on the case study and the training phase. Furthermore, we filtered out insignificant water depths in the simulated maps by setting the water depth in cells with values lower than a specified threshold ($1e^{-3}$ m for the first and third test case and $1e^{-4}$ m for the second one) to zero.

In computing the training losses (Eq. (6) and Eq. (10)), the values of the hyperparameters λ_{GDL} and λ_{GAN}, which generated the highest prediction accuracy, were identified through trial-and-error. The value of these hyperparameters changed during the AE training (Eq. (6)): for the first E epochs we set $\lambda_{GAN} = 0.01$ and $\lambda_{GDL} = 1.0$, then we neglected the influence of L_{GAN} in the total loss computation setting $\lambda_{GAN} = 0$ and $\lambda_{GDL} = 0.01$. E was the number of epochs required for the L_{GAN} loss (Eq. (9)) convergence. Its value depends on the test case and varies approximately between 30 and 50 epochs. For the VPTR training (Eq. (10)) we considered λ_{GDL} equal to 0.1 for the synthetic study cases, and to 0.01 for the real test case application. Generally, the value of λ_{GDL} must be calibrated to achieve an order of magnitude of L_{GDL} loss (Eq. (8)) equal or greater than L_{MSE} loss (Eq. (7)). For the GDL computation we set $\alpha = 1$.

Training a Transformer based model from scratch is challenging, and usually leads to a reduced prediction accuracy if not enough data are available (Bertasius et al., 2021). Furthermore, the datasets employed in this study consist of a notably smaller number of samples in comparison to the datasets normally used for video classification and video future frames prediction tasks. To mitigate the risk of overfitting and minimize training costs, we initialized the AE and VPTR models with weights pretrained on the MovingMNIST dataset (Ye & Bilodeau, 2023).

In this work, all simulations were run using a NVIDIA A100 GPU with 80 GB memory.

4. Results and discussion

In this Section, the results of the FS model testing are presented for the three case studies. The unseen sequences of maps of the testing dataset were used to evaluate the forecasting performance of the surrogate model.

Table 2
Surrogate model hyperparameter settings and number of model parameters for different case studies.

	Case study 1	Case study 2	Case study 3a	Case study 3b
CNN layers (k)	4	4	4	5
Latent feature size ($h \times w \times d_{model}$)	60 x 8 x 512	16 x 8 x 512	28 x 16 x 512	28 x 16 x 768
AE parameters [million]	45.5	45.5	45.5	105
VPTR parameters [million]	129	90	125	244

Table 3 provides a summary of the average performance metrics (RMSE and F1) obtained setting $P = I = 8$ and F equal to 98, 113 and 83 for the three test cases, respectively. We emphasize that these metrics are computed considering the frames $t > I$ (see Section 2.5). As shown in Table 3, the autoregressive procedure of the real-time forecasting application generates higher errors compared to the FS test, where only the frame at time $t = I + 1$ is predicted. This behaviour is expected due to the error accumulation resulting from using predicted maps as input to forecast subsequent ones during the autoregressive procedure (see Fig. 2b). It is also important to note that, for case study 2, the errors are two orders of magnitude lower than the errors observed in the other cases, due to the different scales involved.

The detailed results for each case study will be thoroughly discussed in the following subsections.

4.1. Case 1: Dam-break in a parabolic channel

Considering the first case study, Fig. 4 shows the RMSEs computed for the predicted frames of the FS test and the real-time forecasting application. The RMSE of the test procedure (black line) is in the range 0.4–2.5 cm. The higher errors are observed in predicting the first 10 future frames (i.e., $t \in [9, \cdots, 18]$) due to the rapid dynamic of the flood and the high velocity of the wet/dry front propagation. In subsequent instants, where the flood evolution slows down, the data-driven model improves its accuracy, reducing the error near the front of the flood. The average RMSE is equal to 0.7 cm. This value corresponds to a small fraction of the maximum water depth of the testing dataset (i.e., 10 m). Consequently, we can assert that the SA mechanism of the Transformer block is able to extract spatiotemporal information from the latent features and predict frame at instant $t = I + 1$. The average F1 score close to 1 (Table 3) confirms the capability of the FS model to correctly identify the flooded area in this case study.

For the real-time forecasting application, we considered the number of past P and future F frames equal to 8 and 98, respectively. This configuration allows to consider the entire flood event represented by the 106 samples of the testing dataset, corresponding to about 35 min of real time. The red line in Fig. 4 represents the RMSE computed for each recursively predicted future frame, while Fig. 5 shows the comparison between ground-truth and forecasted maps using the autoregressive procedure, for some representative instants. The red line in Fig. 4 shows a particular trend. In the recursive prediction of the first 15 future frames (i.e., $t \in [9, \cdots, 23]$), the RMSE increases from 2.5 to 12.5 cm. This trend is mainly associated with the higher errors in forecasting the first future frames, similarly to what was observed in the FS test (black line). Furthermore, the error accumulation accentuates the increase of the RMSE. In subsequent instants, approximately until $t = 72$, the error gradually decreases due to the reduction of the flood propagation velocity and the associated error near the wet/dry front. Subsequently, the RMSE rises again due to the error accumulation until frame $t = 100$, when the flood inundation reaches the downstream boundary, and consequently the wet/dry front disappears, resulting in a sudden drop of the RMSE of about 5 cm. The average RMSE computed for all the 98 future frames is reported in Table 3. Fig. 5 shows that the differences between the ground-truth and predicted maps are everywhere lower than 25 cm, except for cells near the front of the flood, where differences can reach up to 80 cm in some frames. These errors are acceptable for the purpose of real-time forecasting. Furthermore, Fig. 5 shows that the symmetry of the inundation is well preserved in almost all the predicted maps. This demonstrates the capability of FS model to correctly reproduce symmetrical floods. We stress that no additional information about the symmetry had been provided to the surrogate model during the training process.

As expected, the autoregressive procedure of the FS model generates higher RMSEs compared to the simple prediction of one frame ahead (FS test), due to the error accumulation. Nevertheless, the average F1 score decreases slightly for the recursive prediction (see Table 3), keeping a

Table 3
Performance metrics for test and real-time forecasting application of the FS model on each case study. Number of frames: $I = P = 8$ and F equal to 98, 113 and 83 for the three test cases, respectively. The unit of RMSE is [m].

	Case study 1		Case study 2		Case study 3a		Case study 3b	
	RMSE	F1	RMSE	F1	RMSE	F1	RMSE	F1
FS test	0.007	0.998	0.0003	0.998	0.042	0.967	0.059	0.928
Real-time forecasting	0.084	0.986	0.0029	0.994	0.104	0.937	0.154	0.886

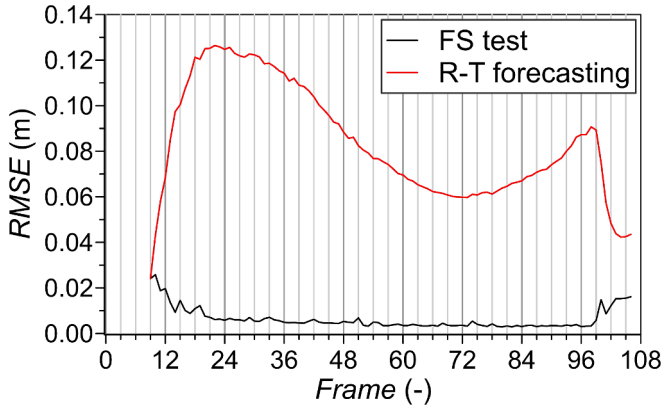


Fig. 4. Case study 1: RMSE computed for the FS test and the real-time (R-T) forecasting application. Number of frames: $I = P = 8$ and $F = 98$.

very high level of confidence.

4.1.1. Sensitivity analysis to the number of input frames

We analysed the influence of the hyperparameter I on the quality of the training result. In particular, we conducted a sensitivity analysis to determine the number of input frames I to efficiently train the FS model for case study 1. We considered values of I equal to 1, 2, 4, 8, 12 and 16. Since the number of input frames affects only the VPTR module, we used the same optimized AE parameters for all configurations analysed. The model performance was assessed by comparing the RMSE and the overall VPTR training times. The computational time required by the training process depends on different factors, such as the batch size. Generally, reducing the GPU memory consumption due to a lower number of frames in the input sequence for the Transformer allows increasing the batch size. While preliminary experiments showed that increasing the batch size reduces the training time per epoch, convergence to the optimal solution is slower and requires a greater number of epochs. Consequently, to mitigate the influence of this hyperparameter, a constant batch size of 4 was considered for all the configurations examined in the present sensitivity analysis. Furthermore, the values of the other hyperparameters were kept as described in Section 3.5.

The results are summarized in Table 4. To ensure a fair comparison, the average RMSE of the FS test was computed considering the predicted maps at the same instants for all simulations (i.e., $t \in [17, 76]$ since the maximum value of I considered was equal to 16). Generally, the average RMSE remains almost constant as I varies, except for $I = 1$ which generates a slightly higher value. Fig. 6a shows the RMSE of the FS test for frames $t \in [2, 30]$. The case $I = 1$ generates higher RMSEs compared to the other configurations. Furthermore, the model trained with $I = 4$ determines a lower error in predicting the first three future frames (i.e., $t \in [5, 7]$), compared to the same maps predicted setting $I = 2$. Similar improvements are observed with $I = 8$ and $I = 4$ for $t \in [9, 11]$, while a further increase in the number of input frames (e.g., $I = 12$ or 16) produces a modest RMSE improvement. Focusing on the training time, the 4th column of Table 4 shows that increasing the number of input frames from 1 to 2 leads to a slight increase in the average epoch training time. Furthermore, the computational time changes almost linearly by varying the number of input frames in range 4–16.

Given that the ultimate goal of the FS model is the application on real flood scenarios, in this sensitivity analysis we also take into account the outcomes of the real-time forecasting procedure adopting the autoregressive method described in section 2.4. In the present study we set the number of past frames $P = I$ and number of future frames $F = 60$. To facilitate a comparison of results across different values of the hyperparameter I, all the configurations analysed focus on the prediction of the same 60 future maps (i.e., $t \in [17, 76]$). Fig. 6b shows the RMSE computed on the recursively forecasted maps. Notably, the model trained with $I = 8$ achieved the lowest RMSE during the autoregressive procedure. The use of a higher value of the hyperparameter (i.e., $I = 12$ or 16) led to a lower accuracy due to the redundancy of information in the input sequence. The average RMSEs summarized in Table 4 confirm that the surrogate model trained with $I = 8$ outperforms the other configurations studied. In conclusion, it can be reasonably assumed that the number of input frames $I = 8$ can also be considered adequate to efficiently train the FS model for the other dam-break case studies considered in the present work.

4.2. Case 2: Dam-break in a rectangular tank

For the second case study, Fig. 7 shows the RMSEs computed for the predicted frames of the FS test and the real-time forecasting application. The RMSE for the FS test is lower than 0.5 mm for all the predicted frames. This value is less than 0.5 % of the initial water level in the upstream reservoir (i.e., 15 cm). Similar to the previous case study, we can assert that the FS model is able to efficiently forecast one frame ahead.

For the real-time forecasting application, we considered the number of past P and future F frames equal to 8 and 113, respectively. Consequently, all the 121 samples of the testing dataset were included. Fig. 8 shows the comparison between the ground-truth and forecasted maps using the autoregressive procedure, for a few representative instants. The temporal evolution of the inundation in the floodable area of the tank is reproduced quite correctly for the first 65–70 future frames. Indeed, the differences between the ground-truth and predicted maps are almost everywhere lower than 2.5 mm. The highest errors are in the floodable area close to the lateral boundaries, where the water depths are higher due to the reflection against the walls. Generally, the differences in the area close to the prismatic block are less than 5 mm, although the water depth reaches values up to 10 cm in this portion of the domain. Furthermore, the surrogate model perfectly reproduces the presence of the block. We emphasize that no additional information about the existence of this element had been provided during the training of the FS model. Consequently, the surrogate model acquired this information from the samples of the training dataset. Conversely, the prediction of the water levels in the upstream reservoir presents larger errors. The discrepancies in this part of the domain can be due to the type of flow generated in the reservoir. Initially, a rarefaction wave travels upstream; then, the wave is reflected by the walls and consequently the flow dynamics is affected by the interaction with the boundaries. Therefore, the low gradients of water depth and the complex interaction with the walls drastically increase the influence of the error accumulation in the recursively predicted maps. It follows that the differences between ground-truth and forecasted maps in the upstream reservoir are up to 13–14 mm (see Fig. 8). Nevertheless, the prediction of the temporal evolution of the flood in the downstream area is not

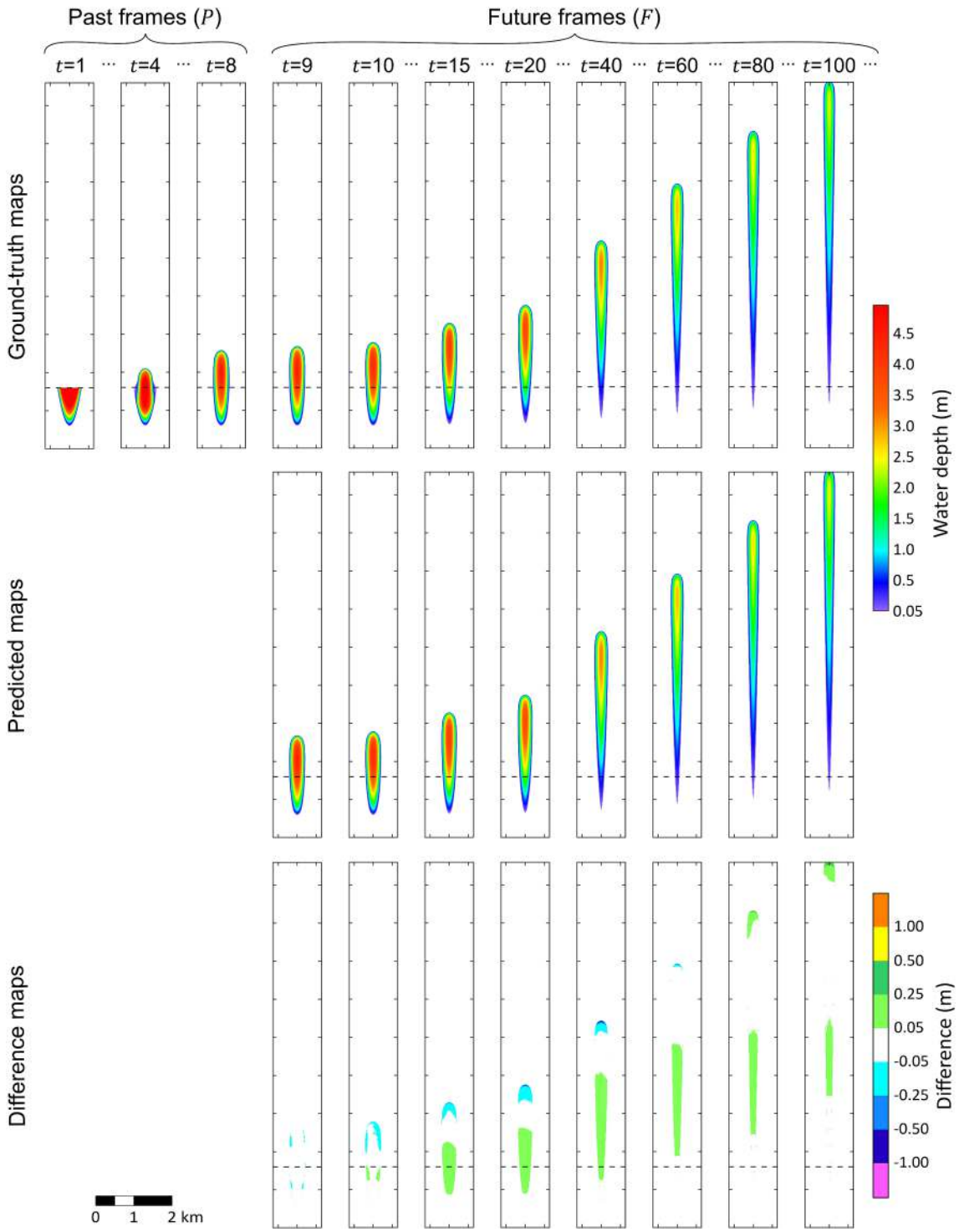


Fig. 5. Case study 1: comparison between ground-truth and forecasted maps for some representative frames of the testing dataset. The predicted maps are obtained through the autoregressive procedure of the FS model. The last row represents the difference between the predicted and ground-truth maps. The dashed black lines represent the position of the gate ($y = 1600$ m). Number of frames: $I = P = 8$.

affected by these errors.

In addition to the standard RMSE computed for the entire domain, Fig. 7 shows the RMSE calculated separately for the cells in the upstream reservoir and in the downstream floodable area. Clearly, the RMSE for the entire domain is influenced by the error in the upstream reservoir for the first 80–85 frames. Indeed, the RMSE computed for the cells in the upstream basin increases rapidly from 0.5 mm to about 5.6 mm in predicting the first 16 future frames (i.e., $t \in [9, \cdots, 24]$), due to high prediction errors in the north edge of the tank (see Fig. 8), and then maintains an oscillatory trend. Differently, the RMSE computed for the downstream area remains lower than 1.0 mm until frame $t = 60$, and then progressively increases, reaching values up to 3.0–3.3 mm. The deterioration of the forecasting process is a result of the error accumulation in the southern area of the tank. Indeed, starting approximately from frame $t = 70$, the inundation reaches the downstream boundary, and consequently a local increase in water depth (up to 75 mm) is

Table 4
Comparison of prediction performance varying the number of input frames I. We set $P = I$ and $F = 60$. The number of training epochs was automatically defined by the early stopping technique. The average RMSE of the FS test and the real-time forecasting application was computed on frames $t \in [17, 76]$. The final configuration ($I = 8$) is in bold.

Input frames I	Training epochs	Training time [min]	Time/ epochs [min]	RMSE FS test [m]	RMSE real-time forecasting [m]
1	104	65	0.63	0.0060	0.080
2	110	75	0.68	0.0056	0.085
4	100	90	0.90	0.0056	0.075
8	**116**	**125**	**1.08**	**0.0054**	**0.045**
12	120	155	1.29	0.0053	0.059
16	136	200	1.47	0.0053	0.056

generated by the wall reflection. The data-driven model tends to overestimate the speed of the front propagation of the reflected wave, producing water depth differences up to 35 mm.

The average RMSE of the 113 recursively predicted frames is equal to 2.9 mm, which is less than 2 % of the initial water level in the upstream reservoir (i.e., 15 cm). Furthermore, the average RMSEs computed separately for the cells in the upstream reservoir and in the downstream floodable area are about 3.4 and 1.6 mm, respectively. The F1 score is extremely high for both the FS test and the real-time forecasting application (see Table 3). Consequently, we can conclude that the FS model correctly reproduces the extension of the flooded area for this case study. Furthermore, similarly to the case study 1, the symmetry of the flooding is quite well preserved in the downstream floodable area (Fig. 8). The major asymmetries can be found in the area close to the lateral walls, where the flood dynamics is affected by the wave reflection, and in the upstream reservoir, where the error accumulation is notable.

4.3. Case 3: Dam-break of the Parma river flood-control dam

In this Section, dedicated to the hypothetical dam-break of the Parma River reservoir dam, we first present the results of the FS training process with a spatial resolution of 20 m. Then, we compare these results with the outcomes of the surrogate model trained using the 10 m resolution maps. Finally, we demonstrate that the FS model can predict tens of future maps requiring only a few (e.g., 1–2) past frames, promoting the integration of our surrogate model in early-warning systems.

4.3.1. Spatial resolution of 20 m

For case study 3a, the solid lines in Fig. 9 represent the RMSEs computed for the predicted frames of the FS test and of the real-time forecasting application. As expected, the latter is affected by the error accumulation, and consequently the recursively predicted maps have higher RMSEs compared to the frames forecasted by the test procedure in which only the $I + 1$ map is predicted.

The RMSEs of the FS test are in the range 3–6 cm, with an average value equal to 4.2 cm. These values are less than 0.5 % of the maximum water depth for the present case study (i.e., 14.2 m). Furthermore, the F1 score for the test procedure is close to 1 (see Table 3). Consequently, also for a dam-break on a real bathymetry, the FS model can forecast one frame ahead with a high level of accuracy.

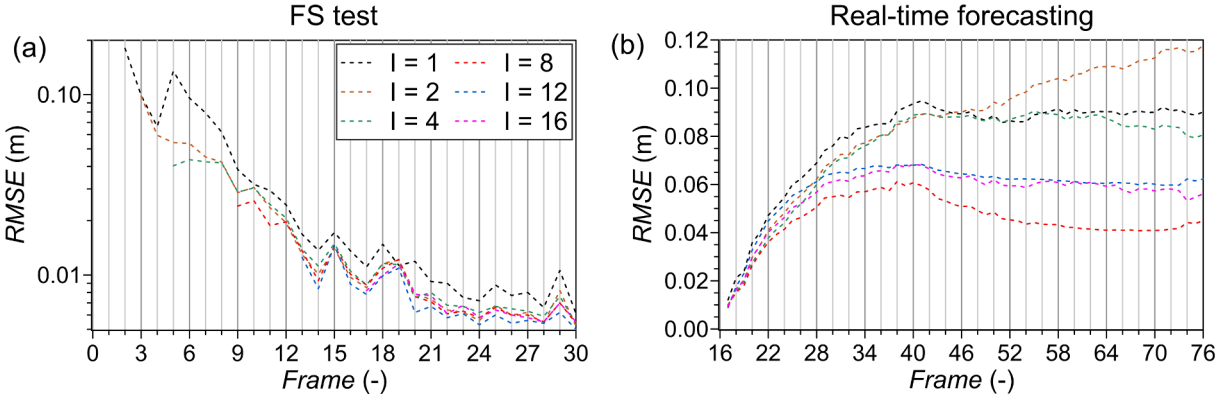


Fig. 6. (a) Comparison of the RMSE computed for the FS test varying the number of input frames I. For clarity of representation, only the first 30 frames are shown. Number of frames: $I = [1, 2, 4, 8, 12, 16]$ (b) Comparison of the RMSE computed for the real-time forecasting procedure varying the number of input frames I. All the simulations forecast the same temporal frames $t \in [17, 76]$. Number of frames: $I = P = [1, 2, 4, 8, 12, 16]$ and $F = 60$.

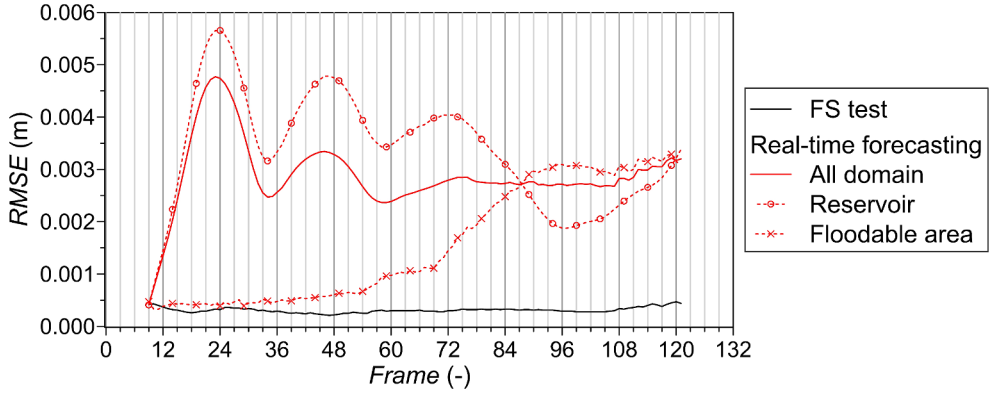


Fig. 7. Case study 2: RMSE computed for the FS test and the real-time forecasting application. For this last case, we split the metric computation for the upstream reservoir and the downstream floodable area. Number of frames: $I = P = 8$ and $F = 113$.

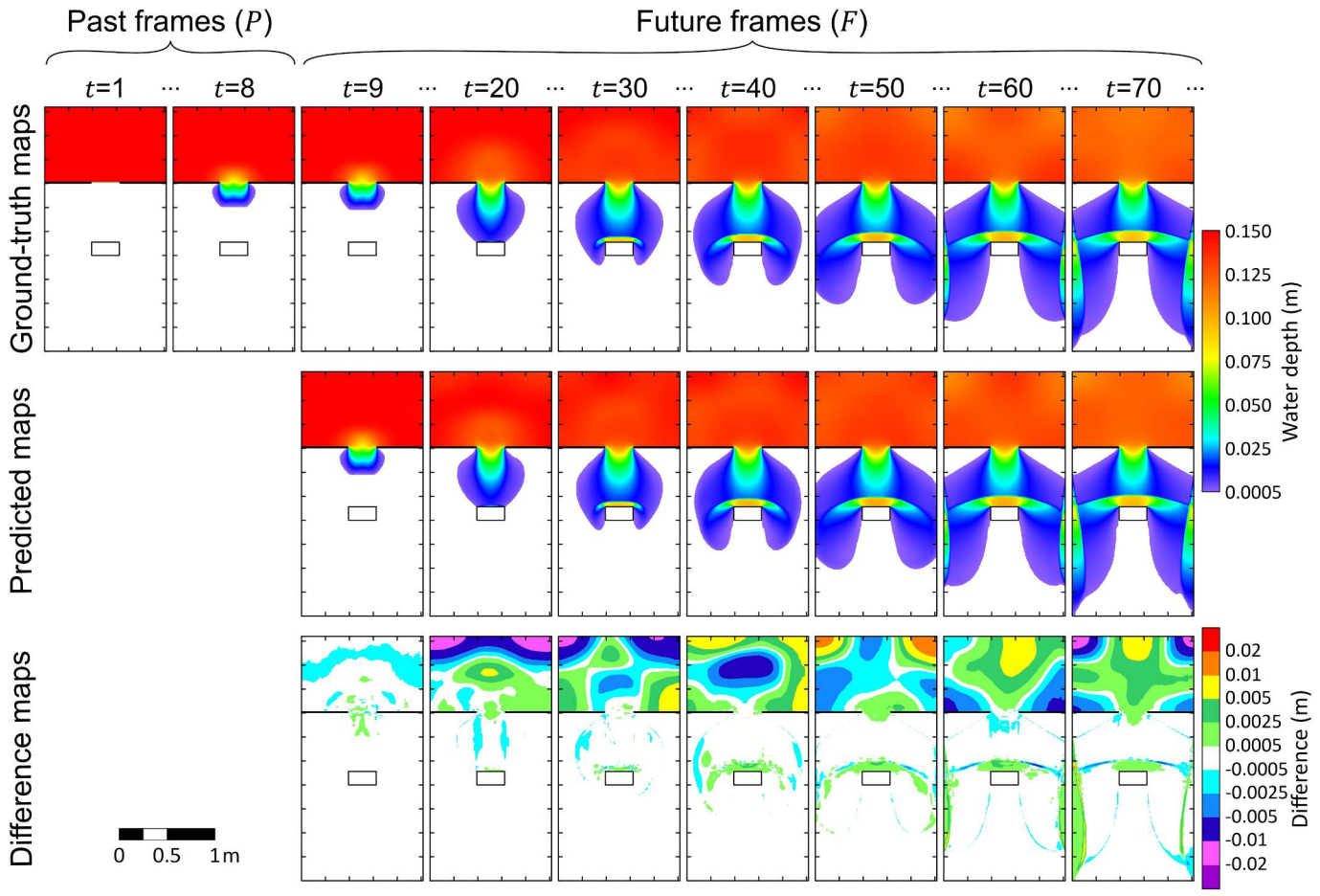


Fig. 8. Case study 2: comparison between ground-truth and forecasted maps for some representative frames of the testing dataset. The predicted maps are obtained through the autoregressive procedure of the FS model. The last row represents the difference between the predicted and ground-truth maps. The black rectangle in the downstream floodable area is the prismatic block. Number of frames: $I = P = 8$.

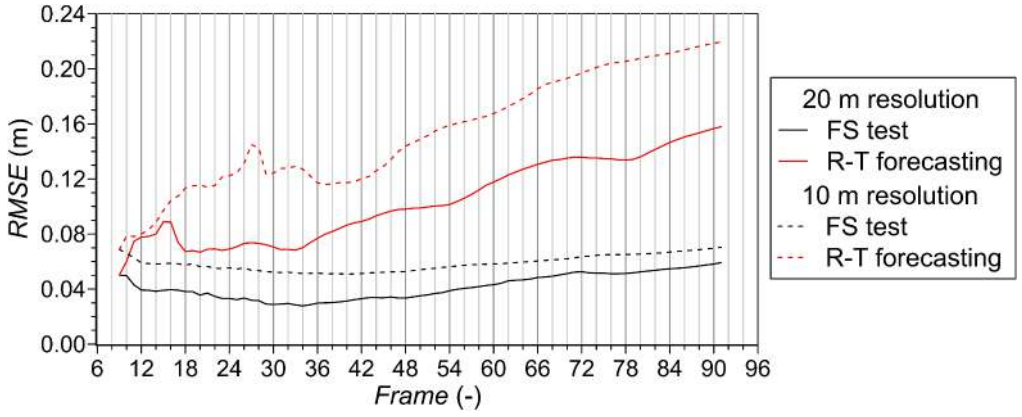


Fig. 9. Case studies 3a and 3b: RMSE computed for the FS test and the real-time (R-T) forecasting application. Number of frames: $I = P = 8$ and $F = 83$.

For the real-time forecasting application, we set $I = P = 8$ and $F = 83$, and all 91 samples from the testing dataset are considered. This lead time, which corresponds to approximately 3 h of real-time, ensures the conclusion of the flood event within the study area (i.e., emptying of the reservoir and reaching of the maximum flood extent). The first predicted frame (i.e., $t = 9$) has a RMSE of about 5 cm (Fig. 9). For frames in range $t = 10$-16 (2nd-8th predicted frames) the error gradually increases up to 9 cm due to the underestimation of the water depths near the wet/dry front of the flood in the river region (see Fig. 10). Starting from frame $t = 17$, the inundation reaches the downstream boundary, and consequently the error correlated to the wet/dry front disappears. This involves a RMSE reduction of about 2 cm (Fig. 9). Then, the RMSE starts to increase again due to the error accumulation, reaching a value of about 16 cm for the last predicted frame ($t = 91$). Fig. 10 shows that, in general, the differences between predicted and ground-truth maps in the area outside the river region are lower than 25 cm, except for some specific locations, where the errors increase up to 1 m. These high differences start appearing approximately from the 60th predicted frame,

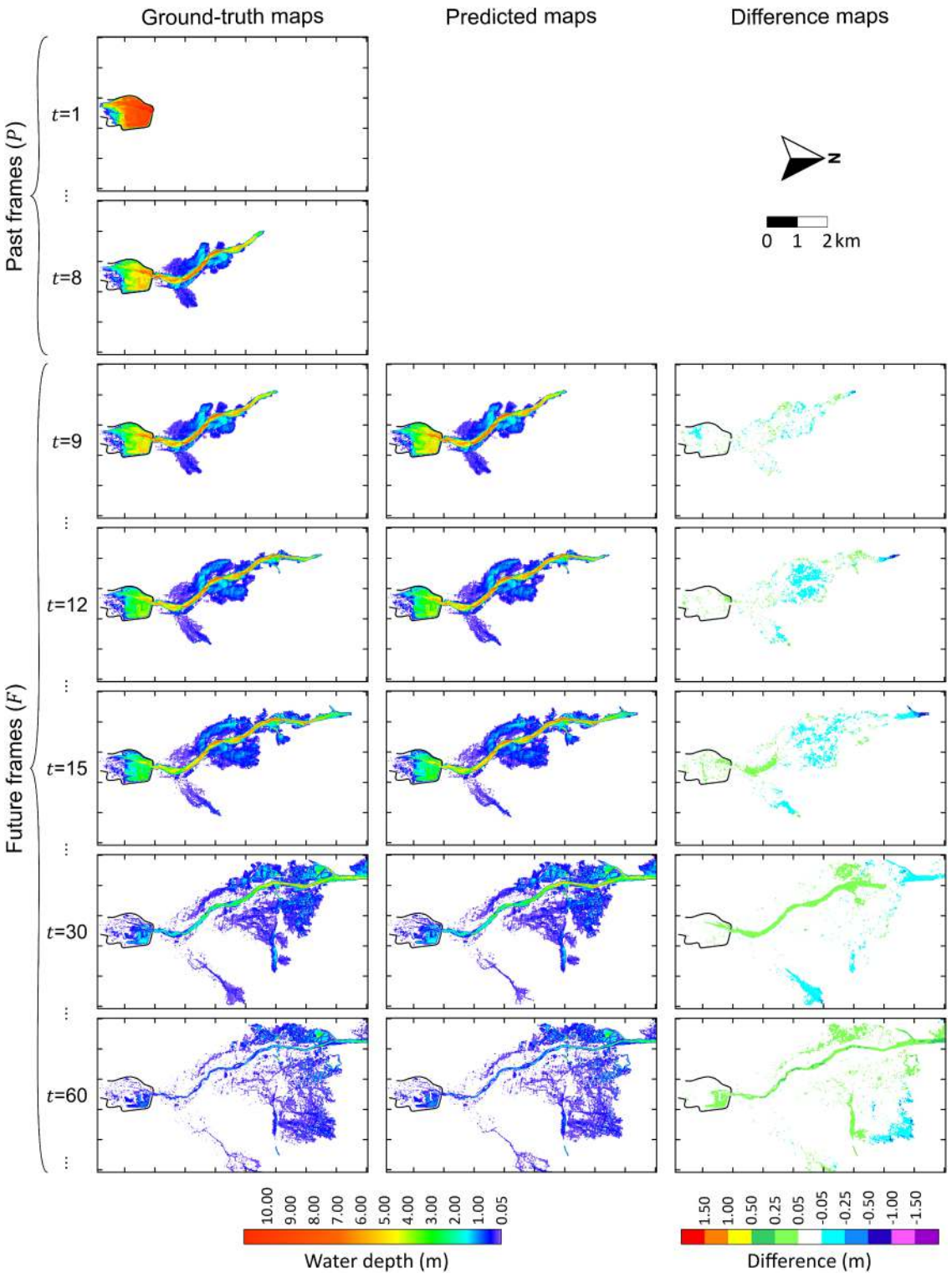


Fig. 10. Case study 3a: comparison between ground-truth and forecasted maps for some representative frames of the testing dataset. The predicted maps are obtained through the autoregressive procedure of the FS model. The last row represents the difference between the predicted and ground-truth maps. Number of frames: $I = P = 8$.

due to the error accumulation at the north-eastern edge of the flood. Furthermore, for these frames, the progressive increase in the RMSE is mainly due to the overestimation of water depths in the river region during the recession limb of the flood. Indeed, considering only the domain outside the river region, the RMSE of the frame $t = 90$ is equal to 9.8 cm, which is 6.1 cm lower than the metric computed for the entire study area, while for $t = 60$ it is equal to 8.5 cm (Fig. 9). For the last frames, the dynamics of the flood propagation in the lowland area is not affected by the overestimated water depths in the river region because the water surface elevations in this area are lower than the riverbanks elevations, and consequently the overtopping is prevented. The average RMSE and F1 score of the 83 recursively predicted frames are equal to

10.4 cm and 0.937, respectively (Table 3). Considering that the present case study is an application to a dam-break on a real bathymetry, the model provides accurate enough forecasts for practical purposes.

For the present case study, the overall training time of the FS model is lower than 3.5 h using an NVIDIA A100 GPU, while the computational time required by the surrogate model to recursively forecast 90 future maps is lower than 1 min. This execution time is comparable to the runtime of the efficient PARFLOOD code, and it is negligible compared to the physical time of the flood simulated (i.e., 3 h). Furthermore, the current implementation of the FS model allows running the real-time forecasting procedure also using Central Processing Unit (CPU) instead of GPU. For example, using an Intel Xeon CPU E5-2680v4 2.4 GHz, the

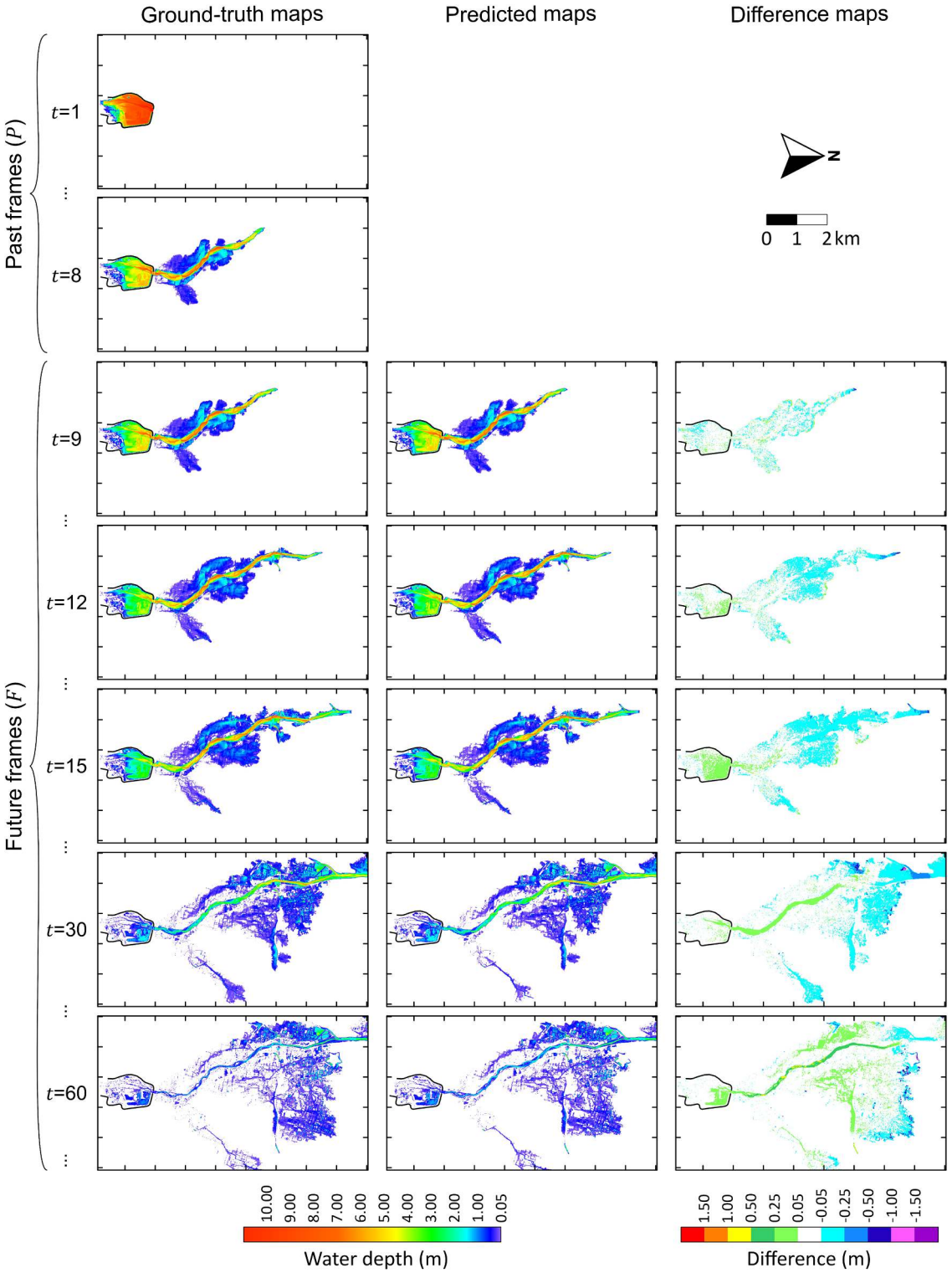


Fig. 11. Case study 3b: comparison between ground-truth and forecasted maps for some representative frames of the testing dataset. The predicted maps are obtained through the autoregressive procedure of the FS model. The last row represents the difference between the predicted and ground-truth maps. Number of frames: $I = P = 8$.

autoregressive procedure takes about 2 min, even if the model was not optimized for inference. This suggests that, while the training process requires high-performance computing hardware, real-time forecasting could be performed even with standard laptops/workstations in a fraction of the real event duration.

4.3.2. Spatial resolution of 10 m

In this Section, we compare the results of case study 3a (20 m resolution, see Section 4.3.1) with the outcomes of the surrogate model trained considering the 10 m resolution maps (case study 3b). The purpose is to analyse the influence of the map dimensions on the performance of the surrogate model, verifying its robustness and scalability with different spatial resolutions. Fig. 9 shows the comparison of the RMSEs for the two configurations. Generally, the errors obtained for the 10 m resolution are slightly larger than the ones obtained with the 20 m grid. Indeed, as summarized in Table 3, the average RMSEs for the FS test and real-time forecasting procedure are respectively 1.7 and 5.0 cm higher for the finer resolution.

For the real-time forecasting application, we considered the same number of frames used for case study 3a (i.e., $I = P = 8$ and $F = 83$). For frames in range $t \in [9, 17]$ (1st-9th predicted frames), the surrogate model trained with the 10 m resolution grid reduces the error near the wet/dry front. However, it generates more spatially distributed differences (in the range 5–25 cm) in the flooded area (see Fig. 10 and Fig. 11). Starting from frame $t = 18$ (10th predicted frame) to frame $t = 36$ (28th predicted frame), the differences increase up to 0.8–1.2 m in some specific locations on the hydraulic left of the river, where the local flow field is particularly complex. For the following frames, similarly to the results obtained considering the coarser grid, the water depths in the river region are overestimated up to 0.5–0.8 m, while the highest errors (up to 1.0–1.2 m) are in the north-eastern edge of the flooded area. Differently from case study 3a, these differences are mainly due to the underestimation of the flooded area near the front of the propagation. Indeed, as we can see from Table 3, the average F1 score for the case study 3b is slightly lower than case study 3a. For the 10 m resolution maps, the accumulation of the error in the river region generates a progressive increasing in the RMSE, which reaches values up to 22 cm for the last predicted frames (Fig. 9). Considering only the cells outside the river banks, the RMSE of frame $t = 90$ is equal to 14.8 cm, which is 7.3 cm lower than the metric computed for the entire study area (Fig. 9), while for $t = 60$ it is equal to 12.5 cm. Nevertheless, these values are 4–6 cm higher than the results of case study 3a. Consequently, refining the map resolution produces a more relevant influence of the error accumulation for long-time series predictions. On the other hand, it allows describing the temporal evolution of the inundation with a better spatial resolution, due to the discretization of the domain with a larger number of cells.

For the 10 m resolution maps, the overall training time is about 6.5 h, which is less than two times higher compared to the coarsened grid. This increase is a result of the higher dimension of the FS model, which requires fitting a greater number of parameters (see Table 2). Nevertheless, the recursive forecast of 90 future maps (i.e., 3 h of lead time) takes less than 1.4 min for the refined resolution. Differently from physically based models, the computational efficiency for the prediction phase of the FS model is only marginally influenced by the map dimension. Indeed, the inference time increases only by 40 % when doubling the spatial resolution.

4.3.3. Sensitivity analysis to the number of past frames

In the previous sections, we presented the results of the real-time forecasting of dam-break scenarios, using the same value for both the number of past frames and input frames (i.e., $P = I = 8$). Consequently, with this configuration, the application of the FS model to forecast the temporal evolution of the inundation maps for a real emergency application would require the availability of the first 8 water depth maps of the flood event, which can be obtained by coupling numerical and surrogate models. This procedure might reduce the overall computational efficiency and it is not an easy task to perform in real time. For this reason, we tested the capability of the FS model to forecast water depth maps by assuming that the number of past frames P was set to a value lower than the number of input frames I assigned during the training process. Theoretically, the forecasting of the temporal evolution of floods setting $P = 1$ would only require the initial water level in the upstream reservoir. Normally, this value is recorded by a gauge station located near the dam, and consequently it is easily available and convertible to a water depth map in lake-at-rest conditions.

For this analysis, we considered the dam-break of the Parma River dam with a spatial resolution of 20 m (case study 3a). We set $I = 8$, consequently we used the previously trained FS parameters. In addition to the real-time forecasting procedure obtained setting the number of past frames $P = 8$ (Section 4.3.1), we also considered $P = 4$, 2 and 1. The corresponding number of future frames F are respectively equal to 87, 89 and 90 (i.e., $P + F = 91$, which is the number of samples in the testing dataset). Fig. 12 shows the RMSEs for the recursively predicted future frames for different values of P, while Fig. 13 represents the differences between the forecasted and ground-truth maps for each configuration analysed. The use of a lower number of past frames generates a RMSE increase for the first frames of the sequence. For $P = 1$, the prediction of the first future frame ($t = 2$) has a relatively high RMSE (approximately 31 cm). As we can see from Fig. 13, the water depths in this frame are particularly high (up to 11–12 m), and the major differences (0.9–1.1 m) are located near the dam and in the southern area of the reservoir. The RMSE value for the subsequent predicted maps progressively decreases, reaching about 7 cm for frame $t = 18$ (17th predicted frame). We obtained comparable results setting $P = 2$ but in this case the RMSE is lower, especially for frames in range $t = 10$–17. For $P = 4$, the RMSE value of the forecasted frames $t = 5$–10 is 1.5–3.5 cm lower than the RMSE for $P = 2$. Similar results are obtained comparing $P = 8$ and $P = 4$, with differences of 1.5–2.5 cm for frames $t = 9$–13. Starting from frame $t = 18$ (approximately the instant the flood reaches the downstream boundary) the RMSEs are comparable for all the simulations. The average RMSEs and F1 scores of the four simulations are summarized in Table 5. The average F1 is almost the same for all the simulations, thus the ability of the FS model to predict the flooded area extension is only marginally influenced by the number of past frames. Similarly, the average RMSE is nearly constant except for $P = 1$, which generates an increase of about 1 cm. Despite these small discrepancies, the results are acceptable for the purpose of real-time forecasting of dam-break scenarios on real bathymetry. The surrogate model computational time is not affected by the variation of the number of past frames (P).

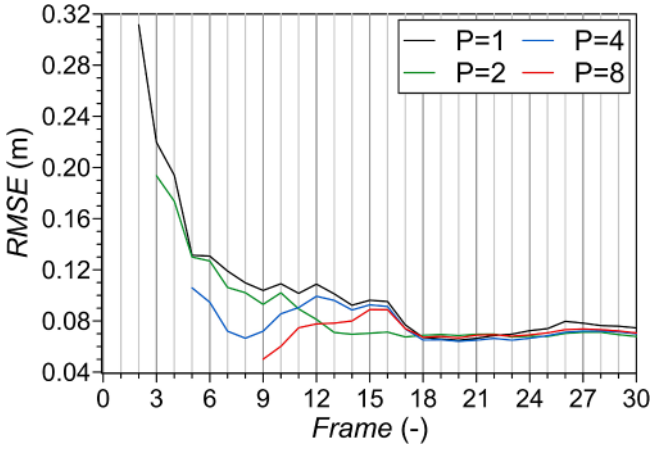


Fig. 12. Comparison of RMSE for different number of past frames (P). Number of frames: $I = 8$, $P = [1, 2, 4, 8]$ and $F = [90, 89, 87, 83]$. Only the RMSE of the first 30 frames is represented.

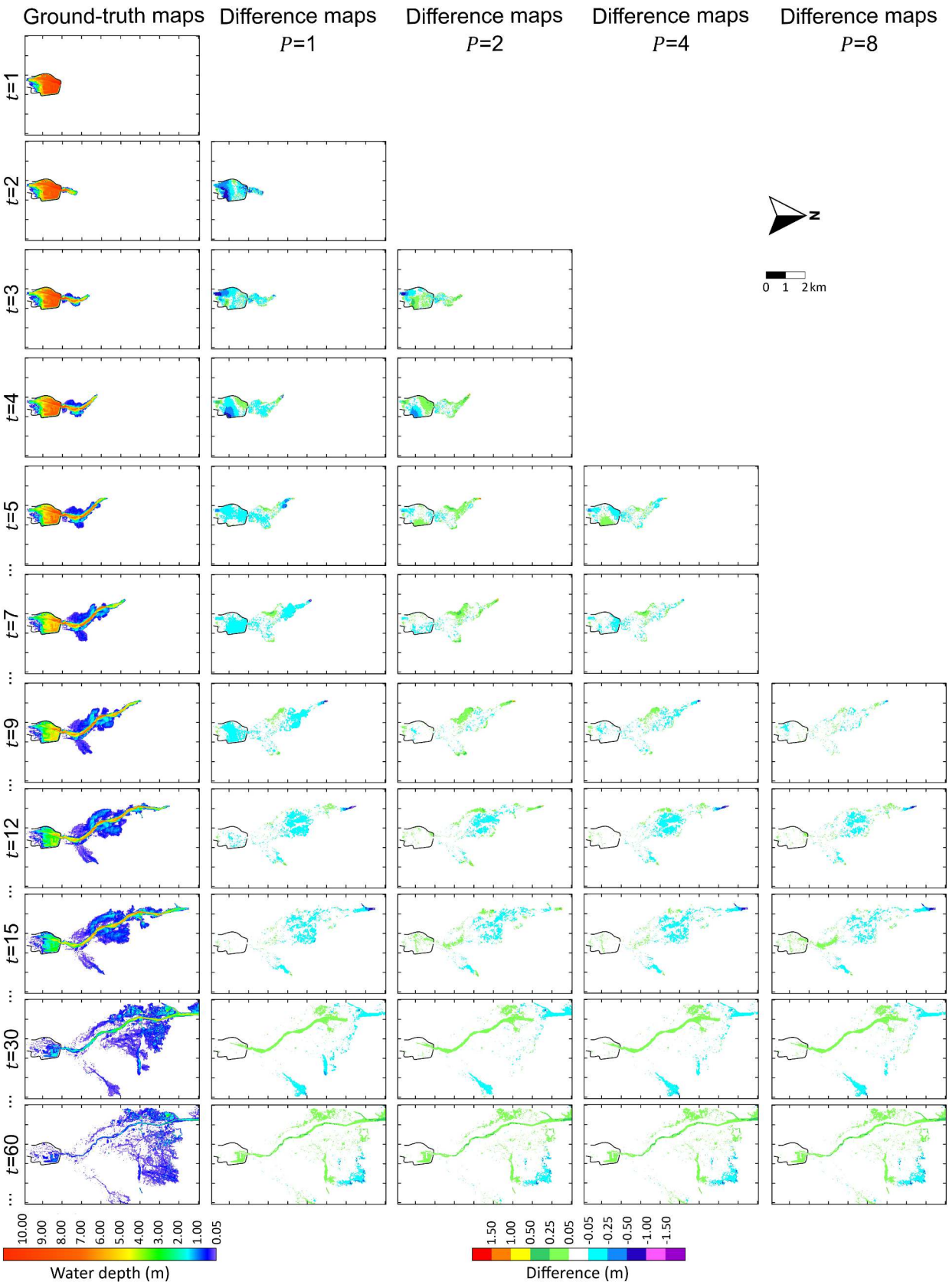


Fig. 13. Comparison of prediction performance with different number of past frames (P). The maps represent the differences between the predicted and ground-truth maps. Number of frames: $I = 8$, $P = [1, 2, 4, 8]$.

Table 5
Comparison of prediction performance with different number of past frames P. Number of frames: $I = 8$, $P = [1, 2, 4, 8]$ and $F = [90, 89, 87, 83]$.

Past frames P	RMSE [m]	F1 [-]
1	0.114	0.936
2	0.106	0.939
4	0.105	0.937
8	0.104	0.937

4.4. Advantages, limitations, recommendations, and future work

Differently from the previously developed surrogate models for flood predictions, the FloodSformer model integrates autoencoder and transformer architectures. The results presented show its ability in predicting long time-series of water depth maps for dam-break scenarios with acceptable accuracy, even when providing a limited number of past frames as input data. This efficiency is principally attributed to the multi-headed self-attention mechanism, which allows modelling long-range dependencies and attending to different space–time information of the input sequence (Bertasius et al., 2021). As a result, for problems where the spatiotemporal correlation is fundamental, the adoption of a transformer-based model provides superior accuracy compared to traditional ML and DL models (e.g., RNN, CNN). Furthermore, as presented in Section 2.3, the loss of the VPTR training procedure is computed by considering all the predicted frames (i.e., not only the future instant $t = I + 1$ but also the input frames within the range $2 \leq t \leq I$), as illustrated in Fig. 2a. This approach reduces errors when employing the surrogate model for autoregressive predictions that start with the number of past frames P lower than I.

A drawback associated with the proposed surrogate model is the error accumulation in the predicted maps due to the autoregressive procedure (see Section 2.4). To address this issue, a non-autoregressive model can be utilized. Indeed, as demonstrated by Ye & Bilodeau (2023), a non-autoregressive model that directly forecasts K future frames can mitigate the accumulation of error in predicted frames and enhance inference speed. However, this type of model comes with several drawbacks. Firstly, training a transformer-based non-autoregressive model is more complicated than its autoregressive counterpart, necessitating a sophisticated loss function to effectively capture and reproduce spatiotemporal information in the maps, along with a higher number of parameters (Ye & Bilodeau, 2023). Secondly, the GPU memory and time consumption during training increase with the raise in the number of forecasted frames K. This imposes a limitation on the number of frames that the non-autoregressive model can predict in a single step. Consequently, if the total required lead time (i.e., F future frames) exceeds K, the real-time forecasting procedure entails using the initially predicted K frames as input data for forecasting the subsequent K frames, repeating this process until all F future frames are predicted. Therefore, for practical applications, a non-autoregressive model also introduces a recursive prediction, potentially resulting in error accumulation similar to autoregressive models. For these reasons, the strategy of implementing a non-autoregressive model was discarded for the present study. Further analyses regarding the suitability of transformer-based non-autoregressive models for predicting the temporal evolution of flood maps can be conducted in future studies.

In this work, the number of input frames I was determined through a sensitivity analysis involving a comparison of metrics for the prediction of test samples (see Section 4.1.1). Differently, the selected number of future frames F ensures that the flood propagation in the study area is concluded (i.e., the reservoir is almost empty, and the maximum flood extent is reached).

In the implementation of a data-driven surrogate model, it should be recalled that, generally, it exhibits optimal performance when interpolating information within the range of the training data. Conversely, the accuracy of predictions significantly diminishes for extrapolations, i.e., when performing predictions of values outside the training range (Fraehr et al., 2024). Therefore, it is important that the training dataset contains a broader spectrum of values than those expected during the inference process. To overcome this significant problem, in the context of dam-break prediction, it is recommended to include in the training dataset the scenario corresponding to an initial water level equal to the maximum allowable in the dam.

The dataset creation is a fundamental step for the implementation of the FS model (and of data-driven models in general). The accuracy of actual flood predictions highly depends on the reliability of ground-truth maps in the training dataset, which can only be generated numerically, because observed maps representing the inundation dynamics at the adequate temporal resolution are never available in the practice. The dataset quality depends on the accuracy of the hydrodynamic model used to generate the maps, and of the model calibration process, which guarantees that simulated results are in close agreement with field observations during past flood events. As previously mentioned in Section 3.1, the PARFLOOD code, employed in this work, is a robust and accurate 2D SWE solver, largely validated for real events (e.g., Vacondio et al., 2014), and therefore meets the requirements as a numerical tool to generate the ground-truth water depth maps. Moreover, for the case studies considered in this work, the maps were generated with an adequate spatial and temporal resolution to provide results of good quality for the training phase. As regards the calibration, the main parameter for hydraulic models is the roughness coefficient. Typically, calibration is carried out when observed data (e.g., recorded water levels at a gauging station, arrival time of the flood, etc.) are available, which is rarely the case for dam-break scenarios. Alternative strategies are therefore necessary to determine reasonable values for the roughness coefficient. For case 1, where a synthetic domain is considered, a value consistent with the friction of natural channels was simply adopted. For case 2, we considered a roughness coefficient derived from a calibration procedure carried out on a similar case by Aureli et al. (2008). Lastly, for the real-field dam-break case study (case 3), observed data were unavailable, as this phenomenon had never occurred in the Parma River. Consequently, the Manning coefficient derived from a prior calibration process based on past flood events was used for the river region. Furthermore, the same value was also assumed for the floodable areas outside the river region. It is important to emphasize that, for dam-break scenarios, the rapid flooding dynamics are only marginally affected by the specific value of the roughness coefficient chosen for the study area (Ferrari et al., 2023), which reduces the impact of the lack of calibration on the results. Obviously, in the context of developing real-time forecasting models intended for practical emergency applications, a rigorous analysis should be conducted to determine the optimal roughness coefficient for each specific case study.

In this work, we focused on the analysis of inundations resulting from dam-break scenarios. However, it's noteworthy that the FS model has the potential to be applied to various types of river floods, provided that open boundary conditions are implemented in the surrogate model. Further developments of the FS model, which may expand its range of applicability and/or improve its predictive accuracy, include the integration of additional input maps (e.g., velocity, terrain elevations, etc.) in the training process and of time series of scalar values (e.g., discharge, levels) to account for open boundary conditions.

5. Conclusions

In this work, a new Transformer-based surrogate model was used to efficiently predict the temporal evolution of inundation maps for dam-break scenarios. The overall goal was to develop a real-time forecasting model that emulates physically based schemes. The data-driven model assessment was performed based on the prediction of flood maps of three case studies: two dam-breaks over synthetic bathymetries (case studies 1 and 2) and the hypothetical collapse of a dam on a real bathymetry (case studies 3a and 3b). The ground-truth water depth maps,

used as samples to train and evaluate the data-driven model, were generated through a hydrodynamical code (PARFLOOD).

Our results showed the following:

- For the first two case studies, the autoregressive procedure of the FS model shows a predictive performance in forecasting the temporal evolution of inundation that can be considered accurate enough for real-time forecasting applications. Furthermore, the symmetry of the inundation is quite properly preserved for both case studies. In case study 2, slightly larger errors were observed in the upstream reservoir, due to the low gradients of water depth and the complex interaction of the flow with the walls.
- For the real-field dam-break case study (case 3a), average RMSE and F1 of about 10.4 cm and 0.94 were obtained, respectively. Generally, the differences between ground-truth and predicted maps are less than 25 cm in the floodable area for the first 60 predicted frames (RMSE $<$ 14 cm), corresponding to 2 h of real-time. The highest errors are located near the wet/dry front of the flood for the first frames, and in the river region for the later frames, due to the overestimation of the water depths during the recession limb of the flood. The FS model's prediction accuracy for real-time applications is considered entirely acceptable.
- When refining the spatial resolution of the maps (case study 3b) a more pronounced influence of the error accumulation in long-time series predictions was observed. However, the errors can be still considered low for real-time forecasting applications. Doubling the spatial resolution (from 20 to 10 m) results in approximately 5 cm increase of the RMSE. Conversely, the temporal evolution of the inundation was better described, thanks to the discretization of the domain with a larger number of cells, without significantly affecting the computational time required for real-time forecasting.
- Reducing the number of past frames P (i.e., maps used as initial condition) minimizes the prior information needed for the real-time forecasting. In the case study 3a, the consequent increase in prediction error is restricted to the first forecasted frames. The average RMSE remains almost unchanged until $P = 2$, while it increases by about 1 cm for $P = 1$. The average F1 score is not influenced by the variation of P. Consequently, our data-driven model can be used to predict long lasting real-time dam-break scenarios, with just 1–2 initial water depth maps as input data.
- As regards the computational times, considering the real-field dam-break case study with a spatial resolution of 20 m (case 3a), the recursive prediction of 90 future frames, which corresponds to 3 h of lead time, takes less than 1 min. Furthermore, even doubling the spatial resolution (case 3b), the forecast takes less than 1.4 min.

In conclusion, the proposed strategy, leveraging an autoregressive procedure to make long-term forecasts of flood maps inferred from the FS model, can be considered a suitable tool for real-time emergency applications, thanks to its short computational time (e.g., about 1 min for the prediction of 90 maps) and its ability to provide accurate forecasts for at least 60 frames ahead.

CRediT authorship contribution statement

Matteo Pianforini: Data curation, Formal analysis, Methodology, Software, Writing – original draft. **Susanna Dazzi:** Conceptualization, Funding acquisition, Methodology, Writing – review & editing. **Andrea Pilzer:** Methodology, Software, Writing – review & editing. **Renato Vacondio:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets generated for this study and the weights of the FS trained model are available in an online repository (Pianforini, M., Dazzi, S., Pilzer, A., & Vacondio, R., 2024a. FloodSformer: datasets&checkpoints [Data set]. Zenodo. https://doi.org/10.5281/zenodo.10878384). Additionally, the Python code of the FS model is accessible via a Zenodo repository provided by the Authors (Pianforini , M. , Dazzi , S. , Pilzer , A. , & Vacondio , R. , 2024b . FloodSformer: python code (v1.0.0). Zenodo . https://doi.org/10.5281/zenodo.10895200).

Acknowledgements

This research was granted by University of Parma through the action "Bando di Ateneo 2022 per la ricerca" co-funded by MUR-Italian Ministry of Universities and Research - D.M. 737/2021 - PNR - PNRR – NextGenerationEU. Renato Vacondio and Susanna Dazzi acknowledge financial support from the PNRR MUR project ECS_00000033_ECOSISTER. This research also benefits from the HPC facility of the University of Parma. The support of CINECA under project AMNERIS (ID: HP10CZG1DV) is also gratefully acknowledged.

References

Aureli, F., Maranzoni, A., Mignosa, P., Ziveri, C., 2008. Dam-Break Flows: Acquisition of Experimental Data through an Imaging Technique and 2D Numerical Modeling. J. Hydraul. Eng. 134 (8), 1089–1101. https://doi.org/10.1061/(asce)0733-9429(2008)134:8(1089).

Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R., 2022. Deep learning methods for flood mapping: a review of existing applications and future research directions. Hydrol. Earth Syst. Sci. 26 (16), 4345–4378. https://doi.org/10.5194/hess-26-4345-2022.

Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R., 2023. Rapid spatio-temporal flood modelling via hydraulics-based graph neural networks. Hydrol. Earth Syst. Sci. 27, 4227–4246. https://doi.org/10.5194/hess-27-4227-2023.

Bermúdez, M., Cea, L., Puertas, J., 2019. A rapid flood inundation model for hazard mapping based on least squares support vector machine regression. J. Flood Risk Manage. 12 (S1) https://doi.org/10.1111/jfr3.12522.

Bertasius, G., Wang, H., Torresani, L., 2021. Is Space-Time Attention All You Need for Video Understanding? Proceedings of the International Conference on Machine Learning (ICML).

Bomers, A., 2021. Predicting outflow hydrographs of potential dike breaches in a bifurcating river system using NARX neural networks. Hydrology 8 (2), 87. https://doi.org/10.3390/hydrology8020087.

Boosari, S.S.H., 2019. Predicting the Dynamic Parameters of Multiphase Flow in CFD (Dam-Break Simulation) Using Artificial Intelligence-(Cascading Deployment). *Fluids 2019, Vol. 4, Page 44, 4*(1), 44. https://doi.org/10.3390/FLUIDS4010044.

Castangia, M., Grajales, L.M.M., Aliberti, A., Rossi, C., Macii, A., Macii, E., Patti, E., 2023. Transformer neural networks for interpretable flood forecasting. Environ. Modell. Softw. 160 (2022) https://doi.org/10.1016/j.envsoft.2022.105581.

Dazzi, S., Shustikova, I., Domeneghetti, A., Castellarin, A., Vacondio, R., 2021a. Comparison of two modelling strategies for 2D large-scale flood simulations. Environ. Model. Softw. 146, 105225 https://doi.org/10.1016/j.envsoft.2021.105225.

Dazzi, S., Vacondio, R., Mignosa, P., 2021b. Flood stage forecasting using machine-learning methods: a case study on the Parma river (Italy). Water (Switzerland) 13 (12), 1612. https://doi.org/10.3390/w13121612.

Dazzi, S., Vacondio, R., Mignosa, P., Aureli, F., 2022. Assessment of pre-simulated scenarios as a non-structural measure for flood management in case of levee-breach inundations. Int. J. Disaster Risk Reduct. 74, 102926 https://doi.org/10.1016/j.ijdrr.2022.102926.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*.

Ferrari, A., Vacondio, R., Mignosa, P., 2023. High-resolution 2D shallow water modelling of dam failure floods for emergency action plans. J. Hydrol. 618, 129192 https://doi.org/10.1016/j.jhydrol.2023.129192.

Fraehr, N., Wang, Q.J., Wu, W., Nathan, R., 2023. Development of a Fast and Accurate Hybrid Model for Floodplain Inundation Simulations. e2022WR033836 Water Resour. Res. 59 (6). https://doi.org/10.1029/2022WR033836.

Fraehr, N., Wang, Q.J., Wu, W., Nathan, R., 2024. Assessment of surrogate models for flood inundation: The physics-guided LSG model vs. state-of-the-art machine learning models. Water Res. 252, 121202 https://doi.org/10.1016/J.WATRES.2024.121202.

Hofmann, J., Schüttrumpf, H., 2021. floodGAN: Using deep adversarial learning to predict pluvial flooding in real time. Water (switzerland) 13 (16). https://doi.org/10.3390/w13162255.

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., Lou, Z., 2018. Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water (switzerland) 10 (11), 1543. https://doi.org/10.3390/w10111543.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2017.632.

Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., Pender, G., 2020. A deep convolutional neural network model for rapid prediction of fluvial flood inundation. J. Hydrol. 590, 125481 https://doi.org/10.1016/j.jhydrol.2020.125481.

Li, C., Han, Z., Li, Y., Li, M., Wang, W., Chen, N., Hu, G., 2023a. Data-driven and echo state network-based prediction of wave propagation behavior in dam-break flood. J. Hydroinf. 25 (6), 2235–2252. https://doi.org/10.2166/HYDRO.2023.035.

Li, S., Yang, J., Ansell, A., 2023b. Data-driven reduced-order simulation of dam-break flows in a wetted channel with obstacles. Ocean Eng. 287, 115826 https://doi.org/10.1016/J.OCEANENG.2023.115826.

Liao, Y., Wang, Z., Chen, X., Lai, C., 2023. Fast simulation and prediction of urban pluvial floods using a deep convolutional neural network model. J. Hydrol. 624 https://doi.org/10.1016/j.jhydrol.2023.129945.

Liu, C., Liu, D., Mu, L., 2022. Improved Transformer Model for Enhanced Monthly Streamflow Predictions of the Yangtze River. IEEE Access 10, 58240–58253. https://doi.org/10.1109/ACCESS.2022.3178521.

Ma, H., Fu, X., 2012. Real time prediction approach for floods caused by failure of natural dams due to overtopping. Adv. Water Resour. 35, 10–19. https://doi.org/10.1016/J.ADVWATRES.2011.08.013.

Mathieu, M., Couprie, C., LeCun, Y., 2016. Deep multi-scale video prediction beyond mean square error. 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings.

Ming, X., Liang, Q., Xia, X., Li, D., Fowler, H.J., 2020. Real-time flood forecasting based on a high-performance 2-D hydrodynamic model and numerical weather predictions. e2019WR025583 Water Resour. Res. 56. https://doi.org/10.1029/2019WR025583.

Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review. In *Water (Switzerland)* (Vol. 10, Issue 11, p. 1536). Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/w10111536.

Panahi, M., Jaafari, A., Shirzadi, A., Shahabi, H., Rahmati, O., Omidvar, E., Lee, S., Bui, D.T., 2021. Deep learning neural networks for spatially explicit prediction of flash flood probability. Geosci. Front. 12 (3), 101076 https://doi.org/10.1016/j.gsf.2020.09.007.

Plate, E.J., 2002. Flood risk and flood management. J. Hydrol. 267 (1–2), 2–11. https://doi.org/10.1016/S0022-1694(02)00135-X.

Teng, J., Jakeman, A.J., Vaze, J., Croke, B.F.W., Dutta, D., Kim, S., 2017. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. Environ. Model. Softw. 90, 201–216. https://doi.org/10.1016/J.ENVSOFT.2017.01.006.

Turchetto, M., Dal Palù, A., Vacondio, R., 2020. A General Design for a Scalable MPI-GPU Multi-Resolution 2D Numerical Solver. IEEE Trans. Parallel Distrib. Syst. 31 (5), 1036–1047. https://doi.org/10.1109/TPDS.2019.2961909.

Vacondio, R., Dal Palù, A., Mignosa, P., 2014. GPU-enhanced finite volume shallow water solver for fast flood simulations. Environ. Model. Softw. 57, 60–75. https://doi.org/10.1016/j.envsoft.2014.02.003.

Vacondio, R., Dal Palù, A., Ferrari, A., Mignosa, P., Aureli, F., Dazzi, S., 2017. A non-uniform efficient grid type for GPU-parallel Shallow Water Equations models. Environ. Model. Softw. 88, 119–137. https://doi.org/10.1016/j.envsoft.2016.11.012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*, 5999–6009. https://doi.org/10.48550/arxiv.1706.03762.

World Meteorological Organization. (2021). *WMO atlas of mortality and economic losses from weather, climate and water extremes (1970-2019)* (Issue WMO-No. 1267). https://library.wmo.int/index.php?lvl=notice_display&id=21930.

Xu, J., Fan, H., Luo, M., Li, P., Jeong, T., Xu, L., 2023. Transformer Based Water Level Prediction in Poyang Lake, China. Water (switzerland) 15 (3), 576. https://doi.org/10.3390/w15030576.

Ye, X., Bilodeau, G.A., 2023. Video prediction by efficient transformers. Image Vis. Comput. 130 https://doi.org/10.1016/j.imavis.2022.104612.

Yin, H., Guo, Z., Zhang, X., Chen, J., Zhang, Y., 2022. RR-Former: Rainfall-runoff modeling based on Transformer. J. Hydrol. 609, 127781 https://doi.org/10.1016/j.jhydrol.2022.127781.

Yin, H., Zhu, W., Zhang, X., Xing, Y., Xia, R., Liu, J., Zhang, Y., 2023. Runoff predictions in new-gauged basins using two transformer-based models. J. Hydrol. 622, 129684 https://doi.org/10.1016/j.jhydrol.2023.129684.

Zhou, Y., Wu, W., Nathan, R., Wang, Q.J., 2022. Deep Learning-Based Rapid Flood Inundation Modeling for Flat Floodplains With Complex Flow Paths. e2022WR033214 Water Resour. Res. 58 (12). https://doi.org/10.1029/2022WR033214.

Zounemat-Kermani, M., Matta, E., Cominola, A., Xia, X., Zhang, Q., Liang, Q., & Hinkelmann, R. (2020). Neurocomputing in surface water hydrology and hydraulics: A review of two decades retrospective, current status and future prospects. In *Journal of Hydrology* (Vol. 588, p. 125085). Elsevier. https://doi.org/10.1016/j.jhydrol.2020.125085.