Abstract Large-scale river models are being refined over coastal regions to improve the scientific understanding of coastal processes, hazards and responses to climate change. However, coarse mesh resolutions and approximations in physical representations of tidal rivers limit the performance of such models at resolving the complex flow dynamics especially near the river-ocean interface, resulting in inaccurate simulations of flood inundation. In this research, we propose a machine learning (ML) framework based on the state-of-the-art physics-informed neural network (PINN) to simulate the downscaled flow at the subgrid scale. First, we demonstrate that PINN is able to assimilate observations of various types and solve the one-dimensional (1-D) Saint-Venant equations (SVE) directly. We perform the flow simulations over a floodplain and along an open channel in several synthetic case studies. The PINN performance is evaluated against analytical solutions and numerical models. Our results indicate that the PINN solutions of water depth have satisfactory accuracy with limited observations assimilated. In the case of flood wave propagation induced by storm surge and tide, a new neural network architecture is proposed based on Fourier feature embeddings that seamlessly encodes the periodic tidal boundary condition in the PINN’s formulation. Furthermore, we show that the PINN-based downscaling can produce more reasonable subgrid solutions of the along-channel water depth by assimilating observational data. The PINN solution outperforms the simple linear interpolation in resolving the topography and dynamic flow regimes at the subgrid scale. This study provides a promising path toward improving emulation capabilities in large-scale models to characterize fine-scale coastal processes.

1. Introduction

The population growth near coastal regions increases human exposure to natural hazards of hurricanes and flooding (Tellman et al., 2021). Coastal inundation area changes greatly during extreme flooding events. The flow and fluxes at the terrestrial-aquatic interface or tidal rivers are governed by river discharge, tide and storm surge, as well as their complex physical interactions (Hoitink & Jay, 2016), which could also impact sediment dynamics and biogeochemical processes (Lamb et al., 2012; Ward et al., 2020). For example, during landfalling with tropical cyclones bringing heavy precipitation, the flood wave propagation from downstream storm surge and tide impedes the upstream discharge and this backwater effect in turn modulates the water level (Dykstra & Dzwonkowski, 2020, 2021; Gori et al., 2020). Such flood event is referred to as compound flooding as multiple mechanisms can occur simultaneously or in close successions (Eilander et al., 2020; Moftakhar et al., 2017). Climate warming will further exacerbate risks of such hazards because it will create more dynamic flooding zones in river systems over the low-lying regions (Santiago-Collazo et al., 2019) by intensifying extreme storm surge and precipitation (Alfieri et al., 2016; Marsooli et al., 2019), accelerating sea level rise (SLR) (Kulp & Strauss, 2019) and enhancing tidal dynamics (Talke & Jay, 2020). It thus becomes crucial to characterize the complex interactions between these processes in a warming climate.

The mitigation of compound flood risks requires the understanding of tidal river dynamics, which are affected by the nonlinear interaction of hydrological and hydrodynamical processes, river geometry and local basin characteristics (Haigh et al., 2019). Beyond the traditional harmonic tidal analysis, there is a growing application of using numerical models to study the compounding dynamics. For example, hydraulic or hydrodynamic models are configured in tidal estuaries to simulate coastal inundation and flood wave propagation in river networks at the event scale (Helvare et al., 2020; Xiao et al., 2021; Y. J. Zhang, Ye et al., 2020) or under a changing climate (Barnard et al., 2019). The upstream and coastal boundary conditions of these models are provided by hydrological and coastal models, respectively. There are also studies that apply more efficient river hydrological models (Feng et al., 2022; Ikeuchi et al., 2017), coupled with global tide and surge models (Muis et al., 2016, 2020), in larger-scale risk assessments (Eilander et al., 2020). However, these modeling studies have not fully
utilized gauged measurements that may be available for tidal rivers and high-resolution remote sensing data. Such data, if appropriately assimilated, will likely improve the model performance, especially for the large-scale models that cannot fully resolve the river dynamics.

Large-scale river models are a major component of Earth System Models (ESMs) that are the primary tool for climate change research. Such models have been used extensively in global water cycle simulations (L. R. Leung et al., 2020) and large-scale flood risk assessments (Towner et al., 2019). Although local river models are frequently used to assess the compound flooding induced by pluvial, fluvial and coastal processes (Bakhtyar et al., 2020), the large-scale models are conceived to simulate at regional to global scales (Ikeuchi et al., 2017; Yamazaki et al., 2011). However, the approximations in the governing physics and the deficiency in mesh resolutions limit these models in representing the local flooding processes, particularly near the highly dynamic terrestrial-aquatic interface. While the large-scale river models can conserve water mass and simulate reasonable river discharge at the global scale (H. Li et al., 2013), these reduced-physics models employ simplified forms for momentum propagation, which requires less data in river channel topography and lower computational cost than dynamic models (Hodges, 2013). These approximations inevitably limit the representation of physical characteristics of dynamic systems (Santiago-Collazo et al., 2019).

Specifically, in a one-dimensional (1-D) river system, the flow dynamics is governed by the 1-D Saint-Venant equations (SVE). Solving SVE in dynamic river models is computationally demanding, because explicit numerical schemes usually require shorter time steps to converge and implicit schemes need multiple iterations to evaluate SVE equations at every time step (Cunge, 1980; Liu & Hodges, 2014; Tompson et al., 2017). While it appears feasible to apply SVE to a large scale at high spatial resolutions with the advancement of high performance computing (HPC), such dynamic models require high-quality river geometry fields. This requirement cannot be met by the large-scale river models (Wood et al., 2011). As a result, approximations must be made to SVE to relax model requirements. For example, for the two common SVE approximations, the diffusive wave equation neglects the Eulerian inertial acceleration and assumes that gravity, friction and pressure dominate the momentum, while the local inertial equation (Bates et al., 2010) only neglects the advection term. These assumptions, however, are likely not valid for flood-prone river systems. In a flat sloping or tidally-influenced river system, flood wave dominates momentum propagation as velocity changes rapidly with space and time. The inertial approximation reduces the flood propagation speed, attenuates the water surface gradient (De Almeida & Bates, 2013) and underestimates the physical characteristics such as the raising and recession of limbs (Shih & Yeh, 2018).

Even with simplified physics, large-scale river models have to apply low-resolution (5–25 km) meshes to accommodate the high computational cost in large-scale applications. For example, the river component of Energy Exascale Earth System Model (E3SM) applies a resolution of 12.5 km (Caldwell et al., 2019), which is too coarse to resolve the flow dynamics induced by the change of flow regimes over various topographical features. Additionally, in such models, flood inundation over a floodplain is usually simulated using mass-conserved schemes that assume no between-cell exchange of inundated water (Luo et al., 2017; Yamazaki et al., 2012). Such schemes, despite achieving satisfied estimation of flood fraction within a cell, are not meant to represent water depth heterogeneity within each cell, which would require resolving the subgrid flow dynamics.

While extensive efforts have been made to develop efficient numerical schemes to represent a more dynamic river system in large-scale models (C.-W. Yu et al., 2017; C.-W. Yu et al., 2020), one alternative is to derive downscaled solutions at the subgrid scale in regions of interest using statistical or machine learning (ML) approaches. Downscaling coarse-scale numerical models to a fine-scale solution is an established research area in climate and ocean modeling (Bürger, 1996; Maraun & Widmann, 2018; Trotta et al., 2021). The downscaling is often classified into two categories: statistical downscaling and dynamic downscaling. The statistical downscaling usually first builds a statistical relationship between the coarse-scale and fine-scale model outputs or observations and then uses the relationship to produce high spatial- or temporal-resolution outputs (Wilby & Dawson, 2013). In the dynamic downscaling, a high-resolution model simulation is obtained by driving a dynamical model with large-scale model outputs and resolving the subgrid physics (Xue et al., 2014). The application of downscaling to river modeling is still limited because (a) large-scale river models have not been used extensively to study coastal fluvial or compound flooding until the recent advances (Feng et al., 2022; Ikeuchi et al., 2017; Muis et al., 2020; Yang et al., 2021); (b) there lacks a consistent way to integrate both remote sensing data and in-situ measurements. Most previous studies only focused on downscaling a single or grouped gauged measurements to a higher
temporal resolution statistically for assessing the local climate change impacts (Moghim & Gharehroagh, 2022; Tisseuil et al., 2010), or downscaling two-dimensional inundation maps by overlaying water depth computed from a coarse-resolution model or satellite data onto a high-resolution digital elevation model (DEM) (Bermúdez et al., 2020; Fluet-Chouinard et al., 2015). To our knowledge, the only method to downscale the 1-D channel flow is to linearly interpolate the modeled water depth at each grid node and available gauged observations across the cross-sections within a grid cell (Schumann et al., 2014). This method does not require channel topology information and cannot resolve the spatially-varied flow regimes within a grid cell. To fill this gap, this work develops a ML-based downscaling method that assimilates available observations and computes subgrid solutions.

Over recent years, the availability of water level data increases rapidly as such data are consistently collected from in-situ measurements at gauging stations as well as by remote sensing (Elkhrachy, 2022), such as satellite radar altimetry (Durand et al., 2016). In-situ data are continuous in time and sparse in space (Nielsen & Norris, 2007). In contrast, remote sensing data, while measured at large time intervals, can cover great spatial domains at high resolutions (Schumann et al., 2014). For example, the upcoming Surface Water and Ocean Topography (SWOT) will enable direct measurements of water level at a spatial resolution of 50 m for all rivers wider than 100 m (Biancamaria et al., 2016; Domenechetti et al., 2018). Such data could potentially be directly incorporated into ML models to obtain downscaled solutions (Zhu et al., 2021).

The ML technique has been explored to alleviate the limitations in the river modeling from various aspects. For example, ML models have typically been employed as surrogates or emulators of hydrological models to perform rapid simulations for real-time predictions and model calibrations (R. Zhang, Zen et al., 2020). ML models are also proposed to predict hydrograph (R. Zhang, Zen et al., 2020), maximum water levels (Berkhahn et al., 2019), infiltration (Crompton et al., 2019), as well as real-time inundation maps (Guo et al., 2021; Qian et al., 2019). Model predictions are made based on a set of input variables, such as precipitation and streamflow. Additionally, training an emulator is usually expensive as it requires abundant data including observations or multiple realizations of high-resolution numerical simulations. As a result, some alternative ML algorithms are proposed to replace computationally demanding components of a numerical scheme. For example, the convolutional neural network (CNN) is used to solve the Euler equation for iterative pressure correction in the velocity update algorithm (Tompson et al., 2017). Similarly, artificial neural networks are used to estimate the velocity in the inertial wave equation (Jamali et al., 2021). However, these ML models are still computationally demanding for training and also they require accurate training data to avoid bad numerical solutions.

Emerged recently are the state-of-the-art physics-informed ML models. Such models are trained by enforcing the relevant physics. Data, mathematical models and domain knowledge are seamlessly integrated and implemented through regression networks (Karniadakis et al., 2021). Physics-informed neural networks (PINN) solve partial differential equations (PDEs) using feed-forward neural network architectures and respect the corresponding physical laws (Raissi, Perdikaris, & Karniadakis, 2019). The unknown solutions of governing equations are inferred from the initial and boundary data of the state variables. The ML models trained with physical constraints require much less data than traditional ML emulators. This approach thus greatly alleviates the limitations of ML in model training caused by data scarcity. The PINN algorithm can also be used in inverse problems to estimate parameters of a PDE with observations of state variables provided (He & Tartakovksy, 2021; Kissas et al., 2020), as well as in developing surrogate models from sparse data (Karpate et al., 2017; Sun et al., 2020). Specialized network architectures have been developed for PINN to meet specific physical laws (W. T. Leung et al., 2022), accelerate the training process (J. Yu et al., 2022) and improve the model accuracy (de Avila Belbute-Peres et al., 2021). As a result, many new variants of PINN were introduced recently, such as multiphysics PINN (He et al., 2020), Parareal PINN (Meng et al., 2020; Shukla et al., 2021), and DeepONet (Lu, Jin, et al., 2021). For a comprehensive review of PINN, please refer to (Karniadakis et al., 2021).

PINN has been explored extensively in solving forward, inverse and data assimilation problems in science and engineering (Karniadakis et al., 2021), such as subsurface transport in porous media (He et al., 2020), acoustic wave propagation (Rashf-Behesht et al., 2021), solid mechanics (Rao et al., 2021) and heat transfer (Cai et al., 2021). In particular, PINN demonstrates tremendous success in simulating flow dynamics (Cai et al., 2022), including high-speed aerodynamic flows (Mao et al., 2020), incompressible flow governed by Navier Stokes equations (Jin et al., 2021; Sun et al., 2020), laminar flows at low Reynolds numbers (Rao et al., 2020), and even fluid-structure interaction (Raiossi, Wang, et al., 2019). Previous applications of PINN to the hydraulic modeling (Cedillo et al., 2022), despite limited, show that it can outperform Artificial Neural Network (ANN) in flood simulations (Mahesh et al., 2022).
The ML-based downscaling is essentially a data assimilation problem with the ML models incorporating measurements and numerical model outputs into the ML training. A key feature of PINN is its simplicity for assimilating observations (He & Tartakovsky, 2021; He et al., 2020). The time-varying observations and/or spatial snapshots at various spatiotemporal scales can be readily incorporated into the PINN training. Based on this feature, we propose a data assimilation approach based on the standard PINN framework and its variant to address problems in Fourier spaces (ff-PINN) (Wang, Wang, & Perdikaris, 2021). The proposed PINN computes the subgrid solution of a coarse model output by assimilating observations without modifying the numerical algorithms or refining the mesh resolution. By incorporating the measured information, PINN is able to emulate the nonlinear interactions of storm surge, tide and discharge within tidal rivers. We do not intend to replace the numerical solver by PINN. Instead, we aim to build the PINN method upon the infrastructure of existing large-scale models to provide downscaled solutions at some specific cells of interest.

The PINN-based method has a few additional advantages over local-scale numerical models. PINN is mesh-free, making it an efficient and flexible tool for downscaling. In practice, the downscaled solution at regions of interest may be obtained by configuring a local numerical model or an emulator at fine scales. Even though PINN itself may not be as efficient as the local model in terms of running speed, PINN does not require discretization and thus saves efforts of generating meshes for the numerical configuration. The meshless formalism enables flexible implementation in various domains as the partial derivatives are evaluated using automatic differentiation in the deep learning platform Tensorflow (Abadi et al., 2016) that computes the gradient of an output variable with respect to the input coordinates (Baydin et al., 2018). Moreover, the PINN-based data assimilation allows exploring the potentials of merging various types of measurements from hydroacoustic meters, remote sensing platforms and other data collection instruments.

This research aims at developing a downscaling approach based on the PINN data assimilation to resolve subgrid variability of river dynamics in coastal regions for large-scale river models. A PINN solver of 1-D SVE is developed to merge in-situ and remote sensing measurements as well as coarse-scale model solutions to obtain a downscaled solution near the coastal interface and improve flood simulation. To the best of our knowledge, this is the first study in literature which develops a data assimilation model for downscaling river flow based on physics-informed ML methods. Several synthetic cases are tested to demonstrate our model’s capability of reproducing flood propagation in a theoretical setup. Importantly, we develop a new neural network architecture to account for periodic tidal oscillations at the river downstream boundary and investigate the effects of the proposed network architecture on the PINN performance. This manuscript is organized as follows. In Section 2, we present the PINN method for solving SVE and data assimilation. The PINN-based downscaling is also discussed. The problem formulation of each case study and the corresponding results are provided in Section 3. Section 4 presents the result discussions, potential limitations in realistic applications and the future path. Finally, the conclusions are provided in Section 5.

2. Methodology
2.1. Saint-Venant Equations

The 1-D SVE consists of a continuity equation and a momentum equation for the dynamics of velocity ($u$) and water depth ($h$) along the river channel. The incompressible continuity equation is defined as

$$\frac{\partial h}{\partial t} + u \frac{\partial h}{\partial x} = q. \quad (1)$$

where $x$ is distance along the river channel, $t$ is time and $q$ is the water inflow per unit length of the channel from land surface and subsurface runoff, groundwater and precipitation. Throughout the study, it is assumed that the river channel does not receive water from the land and atmosphere ($q = 0$). The momentum equation in the full dynamic form is

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + g \frac{\partial h}{\partial x} + g(S_f - S) = 0. \quad (2)$$
where $g$ is gravity, $S$ is riverbed slope and $S_j$ is the friction slope that can calculated using the Chezy–Manning equation:

$$S_f = \frac{n^2|u|u}{R^3}, \quad \text{(3)}$$

in which Manning’s roughness coefficient $n$ is used as the frictional coefficient and $R$ is the hydraulic radius. This study only considers rectangular channels, so $R$ can be expressed as

$$R = bh/(2h + b), \quad \text{(4)}$$

where $b$ is the channel width. The large-scale river models usually use simplified forms of the momentum equation, including the local inertial equation that neglects the convective acceleration term $\left( u \frac{\partial u}{\partial x} \right)$ (De Almeida & Bates, 2013; Yamazaki et al., 2012), the diffusive wave equation that neglects both the convective acceleration term and the local acceleration term $\left( \frac{\partial u}{\partial x} \right)$ (Luo et al., 2017), and the kinematic wave equation that neglects all partial derivative terms (H. Li et al., 2013). None of these simplified schemes include advection that dominates momentum propagation when the flood wave dynamics is strong (Dykstra & Dzwonkowski, 2020).

### 2.2. PINN Approximation of SVE

In this study, instead of using numerical discretization, we take the first attempt to solve SVE using the PINN method. As illustrated in the schematic diagram (Figure 1a), the fully connected feed-forward deep neural network (DNN) in PINN takes spatial and temporal coordinates $x$ and $t$ as inputs and predicts the corresponding unknown variables $u$ and $h$ in the output layer. There are $l$ hidden layers between the input and output layers and $N_l$ neurons in each hidden layer. The neurons between the adjacent layers are fully connected and the inputs of the $l_h$ layer ($z_l$) are fed from the outputs of the previous layer ($z_{l-1}$):

$$z_l = \sigma(W_l z_{l-1} + b), \quad \text{(5)}$$

where the hyper-parameters $W_l$ and $b_l$ are the weight matrix and bias vector at the $l_h$ layer, determined after training, and $\sigma$ is the activation function used to introduce nonlinearity to each output component. In this study, the activation function is selected as $\tanh(x)$, and $W$ and $b$ are initialized using a widely-used initialization scheme, Xavier scheme (Glorot & Bengio, 2010), where initial weights are sampled from a truncated normal distribution.

The partial derivatives in the governing equations are used with the solutions predicted by DNN to approximate the residuals of the governing equations. The partial derivatives are obtained using the automatic differentiation. In PINN, the solutions of SVE, the instantaneous velocity $u(x, t)$ and water depth $h(x, t)$, are estimated with:

$$u(x, t) \approx \hat{u}(x, t, \theta), \quad \text{(6)}$$

$$h(x, t) \approx \hat{h}(x, t, \theta), \quad \text{(7)}$$

where $x$ and $t$ are the space and time vectors, and $\theta$ is the vector of weights and biases (Bishop & Nasrabadi, 2006).

The loss function consists of the residuals of PDE (i.e., the summed residual errors of Equations 1 and 2 when using $\hat{u}$ and $\hat{h}$), and the mean square errors of the DNN approximation against the boundary condition (BC) and the snapshot data ($S$) associated with $u$ and $h$:

$$J(\theta) = J_f(\theta) + \sum_j w_j J_j(\theta), \quad \text{(8)}$$

The subscript $j$ is $BC_u$, $BC_h$, $S_u$ and $S_h$. $J$ and $w$ represent the loss terms and weighting coefficients correspondingly. The PDE loss ($J_f$) is defined as

$$J_f(\theta) = \frac{1}{N_f} \sum_{j=1}^{N_f} |r_f^j(x', t', \theta)|, \quad \text{(9)}$$
where \( N_f \) denotes the number of collocation points in the computational domain. The residual of SVE (\( r_f \)) is obtained by summing the continuity Equation \( 1 \) and the momentum Equation \( 2 \) with \( \hat{u} \) and \( \hat{h} \) substituted. The loss functions corresponding to BC and spatial snapshot data are defined as:

\[
J_{BC_s}(\theta) = \frac{1}{N_{BC_s}} \sum_{i=1}^{N_{BC_s}} |\hat{u}(x', t', \theta) - u(x', t')|^2,
\]

\[
J_{BC_h}(\theta) = \frac{1}{N_{BC_h}} \sum_{i=1}^{N_{BC_h}} |\hat{h}(x', t', \theta) - h(x', t')|^2, \quad x \in \Omega_{BC}, t \in [0, T]
\]
$$J_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} |\hat{\theta}(x', 0, \theta) - u(x', 0)|^2,$$

$$J_{sc}(\theta) = \frac{1}{N_{sc}} \sum_{i=1}^{N_{sc}} |\hat{h}(x', 0, \theta) - h(x', 0)|^2, \quad x \in \Omega_S$$

where $N_{bc}$ refers to the number of points sampled at the boundaries of the computational domain and $N_{sc}$ is the number of points sampled at the spatial snapshots, and $T$ is the simulation period. Time-varying data, such as the observations of $u$ and $h$, can be used in the PINN training to add further constraints. In this study, we add two additional terms (i.e., $w_{obs}J_{obs}$ and $w_{obs}J_{obs}$) to the total loss in Equation 8. The unweighted loss functions corresponding to the observations are

$$J_{obs}(\theta) = \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} |\hat{\theta}(x', t', \theta) - u(x', t')|^2,$$

$$J_{obs}(\theta) = \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} |\hat{h}(x', t', \theta) - h(x', t')|^2, \quad x \in \Omega_{obs}, t \in [0, T]$$

Previous studies showed that the weighting coefficients of the loss functions ($w$) in Equation 8 are critical to the PINN training as these coefficients are used to balance the contribution of different loss terms He & Tartakovsky, (2020, 2021; van der Meer et al., 2022; Wang, Wang, & Perdikaris, 2021). Imbalanced loss terms can lead to failure in achieving convergence of the PINN solution. The selection of weights is problem-specific because the optimal combination of weights varies across different flow scenarios, depending on system properties and conditions. The weights are typically tuned via trial-and-error procedures before training to search for the optimal solution, which is also referred to as nonadaptive weighting, but such method can be very time-consuming (Jin et al., 2021). Recently, a self-adaptive weight approach for PINN has been proposed in McClenny and Braga-Neto (2020) that trains the weights by autonomously focusing on the difficult regions of the solution. The method attains satisfactory accuracy but significantly increases the computational cost in training. To address this issue, in this study we adopt the strategy of adaptive weights (Wang, Teng, & Perdikaris, 2021) that scales different loss terms by updating the weighting coefficients to balance gradients during the back-propagation of the DNN training. The weights at every iteration of the training are updated based on the loss ($J$) obtained in the previous iteration. This scheme is demonstrated efficient and effective in our numerical study (Table 2). The formulation of the DNN parameters at each iterative step is

$$\theta_{k+1} = \theta_k - \eta |\nabla_\theta J_f(\theta_k) + \sum_j \nabla_\theta J_f(\theta_k)|,$$

where $k$ is the iteration step and $\eta$ is the learning rate. The estimated weighting coefficients ($\hat{w}_j$) at the ($k + 1$)th iteration are computed as

$$\hat{w}_j = \frac{\max_i |\nabla_\theta J_f(\theta_i)|}{\sum_i |\nabla_\theta J_f(\theta_i)|}.$$  

The new weighting coefficients are then updated with a moving average

$$w_j = (1 - a)w_j + a\hat{w}_j.$$  

The hyper-parameter $a$ determines the decay rate of the previous weight and is taken as 0.9 to ensure a stable adjustment during the training (Jin et al., 2021; Wang, Teng, & Perdikaris, 2021).

In the PINN training, the DNN is trained using the stochastic gradient descent method by minimizing the loss function of Equation 8 using an adaptive optimization algorithm, Adam optimizer (Kingma & Ba, 2014). The learning rate is a hyper-parameter that determines how much $\theta$ is updated at each iteration step of the training process. Here we apply the learning rate annealing algorithm with an initial learning rate of 0.9 (Rasht-Behesht et al., 2021; Wang, Teng, & Perdikaris, 2021). Normalization of the input coordinates is important to ensure the convergence to the correct solution. So the space and time are both mapped to $[-1, 1]$:

$$X = \frac{2X - \min(X)}{\max(X) - \min(X)} - 1,$$
where $X$ represents the input vector of $x$ and $t$.

### 2.3. Fourier Feature Embedding

In practice, the training of a PINN model is challenging. The PINN solutions may be trapped to a local minimum (i.e., a trivial solution), which satisfies the PDE residual loss but is far from the true solution that should also satisfy initial and boundary conditions (Wong et al., 2022). This is because the trivial solution of a PDE results in flat outputs under the activation function, for example, tanh. This issue can be mitigated by transforming the inputs of PINN into a multi-scale feature space, such as through Fourier feature mapping (Tancik et al., 2020; Wang, Wang, & Perdikaris, 2021) or the sinusoidal mapping (Wong et al., 2022). Such transformation can modulate the levels of the output variability to match the high-frequency patterns of emulated physics in time and space.

As proposed by Wang, Wang, and Perdikaris (2021), prior to the fully connected neural network, we introduce a multi-scale Fourier feature architecture to apply the Fourier feature embeddings to input coordinates:

$$ y(v) = \begin{bmatrix} \cos(2\pi Bv) \\ \sin(2\pi Bv) \end{bmatrix} $$

(17)

where $y$ is the Fourier mapping function, $v$ is the input coordinates, $v = [x, t]^T$, and $B$ is the mapping matrix corresponding to the frequency of the multi-scale Fourier features. The values in $B$ are sampled from a Gaussian distribution $N(0, s)$. The hyper parameter $s$ is referred to as bandwidth in Wong et al. (2022) and is related with the frequency range of the Fourier features and directly affects the output variability. This parameter is problem-dependent and needs to be adjusted to match the frequency of the problem. The value of $s$ can either be optimized during the PINN training or manually fixed based on the prior knowledge of the boundary data. With the Fourier feature embeddings, the first layer of the neural network becomes

$$ H_1 = \phi(W_1 \cdot y(v) + b_1), $$

(18)

where $\phi$ denotes the activation function, $W_1$ and $b_1$ represent the weights and biases, respectively, in the first hidden layer.

In this study, we devise a new neural network architecture that only applies the Fourier feature embeddings to the temporal coordinate to account for the periodic feature of the tidal boundary forcing. As Fourier feature embedding provides the initial guess to the target solutions, the physical characteristics, such as the periodic tidal patterns, are seamlessly encoded into PINN. At the downstream end of a river, the dynamic tidal wave generated by the ocean tides and storm surge can propagate far upstream into the river network (Ikeuchi et al., 2017), inducing high-frequency oscillations in flow velocity and water depth. We address the tidal wave propagation over multiple tidal cycles by mapping the time into a multi-scale Fourier space. The feed forward pass for time and space is now defined as

$$ y_i^{(i)}(t) = \begin{bmatrix} \cos(2\pi B_i^{(i)}t) \\ \sin(2\pi B_i^{(i)}t) \end{bmatrix}, \quad H_{x,i}^{(i)} = \phi(W_1 \cdot y_i^{(i)}(t) + b_1), \quad \text{for } i = 1, 2, \ldots $$

(19)

$$ H_{x,1} = \phi(W_1 \cdot x + b_1), $$

(20)

where $i$ represents the index of the Fourier dimension. In a multi-scale problem, it is possible to map the time input to several Fourier features with different $s$ to reproduce variability sources of different frequencies. The typical range of $s$ is between $10^{-4}$ and $10^2$ (Wang, Wang, & Perdikaris, 2021; Wong et al., 2022). The value of $s$ may be determined by trial and error or performing spectral analysis on the training data (Wang, Wang, & Perdikaris, 2021). The parameter may also be treated as a trainable hyper-parameter and tuned in the PINN training by monitoring the loss function (Wong et al., 2022).

### 2.4. The PINN-Based Downscaling

This section introduces the PINN-based downscaling method that provides an efficient subgrid solution of a large-scale river model. In such cases, the numerical solution of the coarse-scale model, remote sensing data and
in-situ measurements provide the BC, spatial snapshots and observations, respectively, required for the PINN training and validation.

Figure 1b shows the diagram of using PINN to produce a downscaled solution within a grid cell of a 1-D numerical model over multiple time steps. The cell of interest with nodes $x_i$ and $x_{i+1}$ is the domain of a PINN model. The numerical solutions at the grid nodes are used as the boundary data for the PINN training, which are provided at the time interval $\Delta t$ that is usually set as less than one second to hundreds of seconds in river modeling.

The spatial snapshots overlapping the grid cell ($x \in [x_i, x_{i+1}]$) can be extracted from high-resolution remote sensing data whenever available, for example, at time steps $t, t_\ast, t_{+1}$. The remote sensing snapshots usually have a much higher spatial resolution (O($\sim$100 m)) than the large-scale river model (O(>1,000 m)) and a much lower temporal resolution ($\sim$ several day) than the model's output frequency (e.g., 1 hr). Within the river model's grid cell, since PINN does not require grid discretization, the collocation points are directly made from the remote sensing pixels. In the case when remote sensing data are unavailable at the exact location of the model's grid node, the nearest data will be used. It is worth noting that it is difficult to estimate flow velocity accurately from remote sensing. To mimic this data limitation, we only consider the water level information in the spatial snapshots and focus on the downscaled solution of water depth that is of most significance to riverine/coastal flooding.

In-situ measurements of velocity and water level exist extensively in river systems, particularly coastal river reaches where tidal gauges are installed. For example, United States Geological Survey (USGS) maintains over 8,200 continuous records of streamgages for over 115 years (Turnipseed & Sauer, 2010). The water stage is continuously recorded in USGS gauges usually every 15 min with less than 0.2% error. The instruments such as current meter and Acoustic Doppler Current Profiler (ADCP) are used to measure flow velocity periodically, which is then used to derive and maintain the semi-empirical stage-discharge relation for continuous velocity estimates. Many other state agencies and private sectors are also consistently contributing to in-situ measurements. The in-situ measurements usually have continuous coverage over the simulation period ($t \in [t_\ast, t_{+1}]$) and small time interval, usually from 6 min to 1 hr. These time-series data wherever available (e.g., at $x_{i,1}, x_{i,2}, \ldots, x_{i,n}$) are readily assimilated into PINN. Table 1 summarizes the information of the assimilated data used in the PINN training. We will demonstrate the effectiveness of using PINN to solve the downscaled solutions in Case 5 and 6 of the following section.

### Table 1

| Data source          | Loss function | Location | Period | Time interval |
|----------------------|---------------|----------|--------|---------------|
| Large-scale model    | $J_{BC}$      | $x = x_i, x_{i+1}$ | $t \in [t_\ast, t_{+1}]$ | 1 hr |
| Remote sensing       | $J_s$         | $x \in [x_i, x_{i+1}]$ | $t = t_\ast, t_{+1}, \ldots$ | $\sim$10 days |
| In-situ measurements | $J_{obs}$     | $x = x_{obs,1}, x_{obs,2}, \ldots, x_{obs,n}$ | $x_i, x_{i+1}$ | $t \in [t_\ast, t_{+1}]$ | 6 min to 1 hr |

### Table 2

| Experimental Configurations |
|-----------------------------|
| **Domain (m)** | **Simulation period** | **Numerical setup** | **Pseudo observations** | **NN structure** |
|-----------------|----------------------|---------------------|--------------------------|-----------------|
| Case 1          | 0–3,600              | 0–3,600 s           | Equation 26              | $h = 0$ at $x = ut$ | $x = 1,200, 2,400 m$ | $t = 3,600 s$ | 3 | 32 |
| Case 2          | 0–3,600              | 0–3,600 s           | Equation 30              | $h = 0$ at $x = ut$ | $x = 1,200, 2,400 m$ | $t = 3,600 s$ | 3 | 32 |
| Case 3          | 0–305                | 0–24 hr             | Equation 30              | $5.7 \text{ m/s}$ | $x = 100, 200 m$ | $t = 0$ hr | 3 | 64 |
| Case 4          | 0–305                | 0–240 hr            | Tide and surge (Figure 3) | $5.7 \text{ m/s}$ | $x = 100, 200 m$ | $t = 0$ and 10 days | 4 | 64 |
| Case 5          | 0–914                | 0–240 hr            | $S = 0.001$              | Discharge (Figure 3) | $x = 100, 200 m$ | $t = 0$ and 240 hr | 5 | 64 |
| Case 6          | 0–914                | 0–240 hr            | Tide and surge (Figure 3) | Discharge (Figure 3) | $x = 100, 150, 200, 300, 600 m$ | $t = 0$ and 240 hr | 5 | 64 |
3. Problem Formulation

In this section, we used six synthetic experimental cases, building up in complexity, to explore the feasibility of using PINN to downscale a large-scale river model for tidal rivers. The flow simulations are performed over an initially-dry floodplain and along an idealized open channel forced by upstream discharge and downstream tide and storm surge. We first evaluate the capability of the PINN data assimilation in resolving flow dynamics (Case 1–3) and that of the Fourier feature embedding scheme in capturing the high-frequency tidal variations (Case 4). The PINN solutions are compared with analytical or numerical solutions. We then compare the PINN's performance in downscaling the spatially-varied channel flow with that of the traditional linear interpolation method (Case 5–6).

The first two test cases in Sections 3.1 and 3.2 follow the numerical experiments performed in De Almeida and Bates (2013), Hunter et al. (2005), which simulated flood wave propagation over an initially-dry plane with zero and uniform adverse slope, respectively. In these cases, the velocity \( u \) is assumed to be constant. The SVE is simplified and the water depth \( h \) is the single unknown variable. Thus, both the analytical solution and a 4th-order Runge-Kutta (RK4) solution can be used to calculate \( h \). The third and fourth test cases in Sections 3.3 and 3.4 simulate flood propagation induced by a pseudo storm surge and a combination of tide and surge, respectively, in a tilted open channel. Both \( u \) and \( h \) are unknown. The reference solutions of \( u \) and \( h \) are simulated by a high-fidelity hydraulic model. The first four cases are used to demonstrate the capability of PINN in solving 1-D SVE at unsteady flow conditions, which is required to accurately downscale river models in coastal regions. Finally, in Case 5 and 6, we explore to use PINN to solve the downscaled solutions with changing river topology and flow regimes. The size of NN increases with the problem size and the number of unknown variables (Raissi, Wang, et al., 2019). In this study, we use 32 neurons for each state variable and manually tune the number of hidden layers in each test case. The PINN configurations and numerical setups are summarized in Table 2.

Among all cases, the workflow is organized as:

- Set up the analytical or numerical models in the idealized domain to simulate the reference solutions of \( u \) and/or \( h \). The experimental configurations are summarized in Table 2.
- Prepare the training data. While the boundary data \( J_{BC_1} \) and \( J_{BC_2} \) are directly obtained from the reference solutions, the pseudo observations, such as spatial snapshots \( J_{ab_a} \) and \( J_{ab_b} \) and in-situ measurements \( J_{S_h} \), are estimated from the reference solutions by adding a small level (0.2%) of random white noise. This noise level is determined as the typical error in the USGS water level measurements (Turnipseed & Sauer, 2010).
- Formulate the PINN scheme for each test case with manually-tuned NN layers and number of neurons per layer (Table 2).
- Train the PINN model as outlined in Figure 1a.
- Evaluate the PINN solutions against the reference solutions (Case 1–6) and the linear-interpolated downscaled solutions (Case 5 and 6) using two model skill metrics: the relative \( L_2 \) error

\[
\epsilon_h = \sqrt{\frac{\sum_{i=1}^{N} (\hat{h}_i - h_i)^2}{\sum_{i=1}^{N} h_i^2}}, \tag{21}
\]

and root mean squared error (RMSE)

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{h}_i - h_i)^2}, \tag{22}
\]

where \( \hat{h} \) and \( h \) are the predicted and referenced water depth, respectively, and \( N \) is the number of data points.

3.1. Case 1: Nonbreaking Wave Propagation Over a Horizontal Plane

The first case explores the PINN's capability to simulate the flood wave propagation over a horizontal plane, in which the inundation can be considered as an advancing front moving at the speed of \( u \). The analytical solution of SVE is available by assuming \( u \) to be constant over space and time (Hunter et al., 2005). Thus, Equations 1 and 2 are rewritten as
\[ \frac{\partial h}{\partial t} + u \frac{\partial h}{\partial x} = 0 \quad (23) \]
\[ \frac{\partial h}{\partial x} = -\left( S + \frac{n^2 u^2}{h^{4/3}} \right) \quad (24) \]
At \( x = ut \), the moving boundary condition is \( h(ut, t) = 0 \). As \( S = 0 \), the analytical solution of \( h \) can be obtained by directly integrating equation and imposing the moving boundary, which yields
\[ h(x, t) = -\frac{7}{3} \left[ n^2 u^2 (x - ut) \right]^{3/7} \quad (25) \]
At \( x = 0 \), the boundary condition is derived from the analytical solution:
\[ h(0, t) = \left( \frac{7}{3} n^2 u^3 t \right)^{3/7} \quad (26) \]
The reference solution (25) is solved by setting \( u = 1 \) m/s, \( n = 0.005 \) m \(^{−1/3}\) s, \( \Delta x = 30 \) m and \( \Delta t = 30 \) s. The simulation period is 3,600 s.

The PINN solution associated with Equations 23 and 24 is obtained using a DNN with 3 hidden layers and 32 neurons per hidden layer. At \( t = 0 \), the inundation front has not yet moved. Only the water depth at \( x = 0 \) is known at the initial state. So the spatial snapshot is only enforced at \( t = 3,600 \) s. The computational domain is restricted by the far end of the inundation front that represents the propagation distance it has moved over time \( t \). Thus, the BC derived from Equation 26 is enforced at \( x = 0 \), and \( h = 0 \) is imposed at the moving boundary \( (x = ut) \) over the simulation, \( t \in [0, 3,600] \). Note that it is straightforward to impose moving boundary condition under the PINN framework owing to its space-time data assimilation capacity, see Equation 12. The resulting \( N_{bc} \) and \( N_{bc} \) are 121 and 242, respectively. Although PINN does not require mesh generation, the collocation points to compute the loss functions are specified at the nodes of the discretized mesh for the comparison with the analytical solution or the numerical solutions in the following sections. The learning rate is specified with an initial value of \( 10^{-4} \) that is reduced exponentially every 5,000 iterations at the rate of 0.9. The in-situ observations are assimilated at the point of \( x = 1,200 \) m and \( 2,400 \) m over the entire simulation period (\( N_{obs} = 242 \)).

### 3.2. Case 2: Nonbreaking Wave Propagation Over an Adverse Slope

In the second test case we simulate the flood propagation over an adverse slope \( (S \neq 0) \). While the Equations 23 and 24 are still valid, an analytical solution is no longer available. As a result, the numerical solution of \( h \) is solved at each time step using the RK4 scheme (De Almeida & Bates, 2013; Hunter et al., 2005):
\[ h_{i+1} = h_i + \frac{1}{6} \Delta x (k_1 + 2k_2 + 2k_3 + k_4), \quad (27) \]
where
\[ k_1 = f(h_i), \]
\[ k_2 = f \left( h_i + \Delta x \frac{k_1}{2} \right), \]
\[ k_3 = f \left( h_i + \Delta x \frac{k_2}{2} \right), \]
\[ k_4 = f(h_i + \Delta x k_3), \quad (28) \]

\[ f(h_i) = -\left( S + \frac{n^2 u^2}{h_i^{4/3}} \right). \quad (29) \]
Since there is no analytical boundary condition, we use a sinusoidal wave to emulate an advancing inundation front on the floodplain:
\[ h(0, t) = A \sin(2\pi t/T_{bc}). \quad (30) \]
The amplitude $A$ is 4 m and the period $T_B C$ is 4 hr. Please note that the Manning's $n$ determines the maximum travel distance, beyond which the floodplain is dry. However, considering the maximum distance that the flood wave can travel is $x = u t$, we can impose a vanishing condition at the moving boundary $x = u t$, that is $h(u t, t) = 0$. In the simulation, the Manning's $n$ is specified as 0.025 m$^{-1/3}$s and $S$ is $10^{-3}$ m⋅m$^{-1}$. The simulation period is 3,600 s. The RK4 solution is obtained with setting $\Delta x = 20$ m and $\Delta t = 30$ s. The DNN is configured the same as that in Section 3.1. We evaluated the PINN solutions with assimilating observations at $x = 1,200$ and 2,400 m. The values of $N_S$, $N_{BC}$ and $N_{obs}$ are the same as those in Case 1.

### 3.3. Case 3: Storm Surge Propagation Along a Sloping Open Channel

This case examines the PINN's capability of solving the full SVE in a more dynamic flow condition. Because both analytical solution and RK4 solution are no longer available, the reference solution is obtained from a widely used finite difference based hydraulic model, HEC-RAS (Brunner, 1995). The numerical solutions from HEC-RAS provide the BC, pseudo spatial snapshots and in-situ observational data for the PINN training. The convergence criteria/tolerance of the maximum norm is set as 0.006 m (0.02 ft).

The experiment is set up in an open channel with a uniform width of 3 m (Figure 2). The slope is approximately 0.007, 0.004, and 0.01 m⋅m$^{-1}$ over the three sections. The flow regime changes with the channel slope. In HEC-RAS, the channel is divided into segments connected at nodes and the spatial resolution is 6 m. The state variables ($h$ and $u$) are outputted every 15 min. A constant flow of 5.7 m$^3$/s (200 cfs) is imposed at the upstream boundary, $x = 914$ m (3,000 ft), while a pseudo storm surge is enforced at the downstream boundary, $x = 0$ m. The storm surge is approximated using a periodic sinusoidal wave (Equation 30) with an 2.4 m amplitude and a 50-hr period. The Manning’s $n$ is 0.022 m$^{-1/3}$ s. The simulation period is 24 hr. Please note that US customary Units used in HEC-RAS simulations are converted to standard international units.

![Figure 2. The diagram of the computational domain.](image)

![Figure 3. The upstream boundary of flow (dash lines) and the downstream boundary of water stage (solid lines) in Case 4, 5, and 6.](image)
The DNN architecture consists of 3 hidden layers and 64 neurons per hidden layer. The collocation points are specified at each computational node. The learning rate is initialized at $5 \times 10^{-4}$ and then reduces exponentially every 5,000 iterations at the rate of 0.9. For Case 3 and 4, we only focus on the flood propagation in the downstream section, $x = 0–305$ m (0–1,000 ft) where tide and storm surge dominate the momentum. The collocation points in the remaining domain are not used in the model training. The upstream and the downstream BC are imposed at $x = 305$ m and $x = 0$ m, respectively. The spatial snapshot of $h$ is only specified at $t = 0$ as the revisit time of satellites is usually greater than a day. The time-series observations are assumed available every 15 min and are assimilated at $x = 100$ and 200 m. The resulting $N_s$, $N_{BC}$ and $N_{in}$ are 51, 192 and 96, respectively. The BC of $u$ and $h$ are obtained from the HEC-RAS output. The snapshot of $h$ and observational data of $u$ and $h$ are obtained by adding 0.2% noise to the HEC-RAS solutions.

3.4. Case 4: Tide and Surge Propagation Along a Sloping Open Channel

In the test Case 4, we simulated flood wave propagation forced by a combination of tide and storm surge along the same open channel as used in Section 3.3, with a focus on examining the effect of Fourier feature embedding. The diurnal tidal signal is set by $A = 1.5$ m and $T_{BC} = 25$ hr in Equation 30. A storm surge is imposed to the tide during a randomly-selected 48-hr period. The upstream flow is 5.7 m$^3$/s. The simulation is extended to 10 days. The upstream and downstream boundaries specified in the HEC-RAS for the test Case 4 are provided in Figure 3. The state variables ($h$ and $u$) are outputted every 1 hr. The DNN architecture of 4 hidden layers is employed with each hidden layer consisted of 64 neurons. The other configurations are the same as those in Section 3.3, except that the snapshots are added at the initial state and the last time step assuming satellite data are available every 10 days. The corresponding $N_s$ is 102, and $N_{BC}$ and $N_{in}$ are 480 and 240 according to the updated output interval. We evaluate the PINN solutions under two different NN architectures: the standard NN architecture and the NN architecture using the Fourier feature embeddings. In both solutions, observations are added at $x = 100$ and 200 m. We use $s =$ of 0.25 and 10 (in Equation 19) to account for the tidal frequency in this case. These numbers are determined by trial and error.

3.5. Case 5: PINN Downscaling

The aforementioned cases have shown that PINN can achieve reasonable accuracy when constrained by BC and data of spatial snapshots and in-situ observations. In Case 5 and 6, we perform the simulation over the full channel (Figure 2) and explore the PINN’s capability to resolve the channel topography and variable flow regimes at the subgrid scale. The PINN-based downscaled solutions are also compared with the solutions obtained from the linear interpolation.

At the upstream boundary, a time-varying discharge is imposed with a base flow of 17 m$^3$/s (600 cfs) and a peak flow of 28.3 m$^3$/s (1,000 cfs), which follows the sinusoidal curve (Equation 30) with a period of 20 days and an amplitude of 11.3 m$^3$/s (400 cfs) as shown in Figure 3. The downstream is configured with a normal depth boundary, where a constant slope ($S$) of 0.001 is added. This slope is slightly smaller than the slope of the downstream section and ensures that the water slowly drains.

As the channel is considered as a grid cell of a large-scale river model, the downstream and upstream ends of the channel (at $x = 0$ and 914 m) are the cell’s nodes, corresponding to $x_i$ and $x_{i+1}$ in Figure 1b, where the HEC-RAS solutions of $h$ and $u$ are enforced as the boundary condition in PINN. The pseudo in-situ observations over the simulation period are imposed at $x = 100$ and 200 m. The pseudo snapshots of $h$ along the channel are added at $t = 0$ and 240 hr, corresponding to $t_{in}$ and $t_{in+1}$ in Figure 1b. The linear-interpolation solution is also obtained by linearly interpolating water depth $h$ across the two boundary nodes and the in-situ observations at each time step. A fully-connected DNN is employed with 5 hidden layers and 64 neurons per hidden layer. The other configurations are the same as those in Section 3.4. Since no tides are enforced in this case, PINN applies the standard NN architecture.

3.6. Case 6: PINN Downscaling With a Periodic Boundary

This case simulates a more dynamic condition at the river-ocean interface where oceanic and fluvial processes dominate the downstream and upstream sections separately. The impact of in-situ observations is assessed by adding additional gauging stations to the domain. In addition to the two stations in the downstream section ($x = 100$ and 200 m), we sequentially add gauges 150 m, 300 and 600 m from the upstream boundary (Table 4).
This allows us to evaluate the performance of linear interpolation downscaling with more uniformly distributed in-situ observations.

The upstream boundary is configured the same as that in Case 5 (Section 3.5). The downstream water stage boundary, as shown in Figure 3, is a combination of tide and storm surge as specified in Section 3.4. The PINN solutions apply the NN architecture with the Fourier feature embeddings. The other configurations are the same as those in Case 5.

4. Result and Discussion

This section demonstrates and discusses the results of the six test cases that are configured following the descriptions in Section 3. The limitations of the PINN-based downscaling and future developments of the method are also discussed.

4.1. Test Results

4.1.1. Case 1 and 2

We compare the spatiotemporal evolution of water depth ($h$) in the analytical solution and the PINN solutions (Figure 4) along the flood propagation distance at four different time instants (Figure 5). The result shows that the PINN solution achieves high accuracy. The pattern of the flood wave propagation and the shape of the inundation front are both well reproduced, with small $\epsilon_h$ and RMSE (Table 3).

The comparisons between the RK4 solution and the PINN solution over the spatiotemporal domain and along the travel distance are provided in Figures 6 and 7, respectively. PINN is able to capture the flood propagation and the water surface profile well. The small bias only occurs near the end of the inundation front, which is most likely

Figure 4. The spatiotemporal evolution of the water depth in Case 1 simulated by the analytical scheme (a) and the physics-informed neural network (PINN) model (b) and the error of the PINN model simulation (c). The green dots, black circles and red stars in (a) of this figure and Figures 6, 8, 10, 12, and 14 represent the training data locations of BC, in-situ observation and spatial snapshots, respectively.

Figure 5. The propagation of flood water along a conceptual flat floodplain simulated by the analytical method (reference) and the physics-informed neural network (PINN) model (PINN) at four time instants (i.e., 1,200, 1,800, 2,400, and 3,600 s from the initial time) in Case 1.
because there is no information enforced at the far end and the boundary of \( h = 0 \) is specified at \( x = ut \), a distance greater than the true travel distance. Overall, the resulting \( \epsilon_h \) and RMSE imply reasonable accuracy of the PINN solution (Table 3).

In Case 1 and 2, we tested the PINN's capability of solving simplified SVE with a moving boundary, demonstrating the flood propagation over a floodplain can be well simulated by PINN. Also, provided with adequate observations and topology information, PINN can be used to compute the floodplain inundation at the subgrid scale, resulting in smaller errors than those in numerical studies (De Almeida & Bates, 2013; Hunter et al., 2005). Even though our intention is not to replace the efficient inundation scheme of a large-scale river model that covers the entire domain (Luo et al., 2017; Yamazaki et al., 2011), PINN can provide a more detailed inundation map at local regions of interest as the within-cell heterogeneity of water depth is resolved.

### 4.1.2. Case 3 and 4

The PINN solutions are evaluated against the HEC-RAS reference solutions in terms of the water depth \( (h) \) over the spatiotemporal domain (Figure 8) and their along-channel profiles at four different time instants (Figure 9). The PINN solution agreed well with the reference solution, with small simulation bias near the upstream end of the along-channel profile. The corresponding \( \epsilon_h \) and RMSE also imply good performance (Table 3). The error is close to the value of 0.01 m, an accurate reproduction of the reference solution (Cunge, 1980).

Figures 10 and 11 compare the simulated spatiotemporal evolution of \( h \) and the along-channel profiles between the HEC-RAS reference model and the two PINN models. The PINN solution that encodes the Fourier feature is able to capture the periodic variation induced by the downstream tide. The tidal phase and the amplitude of the tide and surge are reproduced (Figure 10c). The rising and ebbing of the water surface are also well simulated. Figure 11 shows two time instants during the propagation of tide and storm surge and two time instants during the tidal ebbing. The larger bias only occurs during low tides when water flows relatively fast toward the downstream end and between the two observational sites (Figure 11). The biases at \( x = 100 \) and \( x = 200 \) m are likely due to the weighting of the loss function in Equation 8 that balances the observation loss \( (J_{obs}) \) with the other loss terms (i.e., \( J_f, J_{BCa}, \) and \( J_3 \)). In contrast, the PINN solution using the standard architecture can only predict the storm surge and a few tidal cycles and results in poor performance when the tidal variations are not captured (Figures 11b and 11c). The solution with Fourier feature embeddings has a much smaller \( \epsilon_h \) and RMSE than the standard PINN solution (Table 3). Overall, the accuracy of PINN prediction for Case 4 is worse than the Case 3, even with the enhancement by the feature encoding, because of higher oscillatory behaviors (see Figure 11a) created by tides and boundary conditions. This implies more physical constraints and observation data are required in the PINN training. The impact of assimilating more observations is discussed in Section 3.6.

### 4.1.3. Case 5 and 6

The PINN predicted \( h \) is compared with the HEC-RAS and linear-interpolation solutions over the full spatiotemporal space (Figure 12) and along the channel at a few time instants (Figure 13). The linear interpolation yields a

| Case no. | PINN scheme | \( \epsilon_h \)   | RMSE (m) |
|----------|-------------|-------------------|-----------|
| Case 1   | Standard    | 3.075 x 10^{-3}  | 0.001     |
| Case 2   | Standard    | 3.685 x 10^{-2}  | 0.076     |
| Case 3   | Standard    | 6.635 x 10^{-3}  | 0.054     |
| Case 4   | Standard    | 3.093 x 10^{-3}  | 1.771     |
| ff       |             | 7.176 x 10^{-2}  | 0.411     |

**Table 3.** Information of the Simulation Errors in Case 1–4

Note: RMSE, root mean squared error; PINN, physics-informed neural network.
poor downscaled solution, with the interpolated $h$ only agreeing with the reference over the downstream section. The accuracy of the linear interpolation reduces significantly in the other two sections where there are no in-situ data. Neither the spatiotemporal patterns nor the along-channel profiles are reproduced in these regions. The PINN solution, on the other hand, has a more resolved flow dynamics over the full subgrid sections. Particularly at the middle section where the subcritical flow dominates and the water depth is greater, PINN provides a more accurate solution (Figure 13). The errors of the PINN solution only arises where the flow regime changes. In general, the errors (i.e., $\epsilon_h$ and RMSE) of the downscaled solution in PINN are less than 1/5 of those in the linear-interpolation solution.

The PINN solutions are compared with the HEC-RAS reference and the linear-interpolation solutions over the spatiotemporal space (Figure 14) and along the channel (Figure 15). While both downscaling methods show improved performance with the increase in in-situ data, the PINN solutions are more accurate and are less sensitive to the observations. When there are only gauges at the downstream section ($x = 0–305$ m), PINN is able to capture the tidal phase and amplitude reasonably well (Figure 14b). The periodic tide is characterized in the Fourier space, although there are small oscillations in the PINN solution. These oscillations are caused by the high-frequency component ($s = 10$) defined in the Fourier feature embedding. The propagation of the storm surge and the gentle variation induced by the upstream discharge, however, are underestimated. In contrast, the linear interpolation overestimates the tidal influence in the two upstream sections ($x = 305–914$ m), where the flow regimes cannot be resolved due to the lack of data (Figure 14c). As a result, the linearly interpolated water depth is much smaller than the reference and PINN solutions (Figure 15). The error metrics also indicate that the PINN downscaling outperforms the linear interpolation (Table 4): the errors are reduced to one third. When five in-situ stations are available (Figures 14h and 14i), PINN still has more accurate solutions but the difference

Figure 7. The propagation of flood water along a conceptual sloping floodplain simulated by the RK4 scheme (reference) and the physics-informed neural network (PINN) model (PINN) at four time instants (i.e., 1,200, 1,800, 2,400, and 3,600 s from the initial time) in Case 2.

Figure 8. The spatiotemporal evolution of the water depth in Case 3 simulated by the HEC-RAS model (a) and the physics-informed neural network (PINN) model (b) and the simulation error of the PINN model (c).
between the two methods is reduced. PINN shows slight improvements in the upstream stretch of the tides. The accuracy of the linear interpolation increases significantly over the two upstream sections. The water depth is better estimated except that near the upstream boundary which requires more data.

The comparison of the two downscaling methods re-emphasizes the limitation of using the linear interpolation to downscale channel flow in variable flow regime conditions. This method has a large dependence on the amount of gauged data. Otherwise, the subgrid features cannot be resolved, resulting in poor downscaled solutions. In contrast, the PINN downscaling can achieve reasonable accuracy even when the observations are limited.

### 4.1.4. Computational Effort

The simulation time of the numerical and PINN solutions in Case 1–6 and the linear-interpolation downscaling in Case 5 and 6 are summarized in Table 5. The simulations and linear interpolations are performed on an Intel Xeon CPU. The size of the problem and the NN structure regulate the PINN’s computational time. It increases with the number of collocation points ($N_f$) and the training data ($N_{\text{data}}$), the NN size (e.g., number of hidden layers and neurons per layer) and the complexity in NN architecture. For example, the training time of the cases using the Fourier feature embeddings is over two times larger than that of those using the standard NN.

PINN, despite having a higher computational cost than the numerical methods, shows promising performance when the training data set has a relatively small size and the problem size is comparable to realistic conditions. More importantly, to obtain the subgrid solution of multiple grid cells, a local numerical model has to be configured separately for each corresponding cell. Considering such efforts, PINN can be more efficient and flexible than a local numerical model. PINN also has multiple advantages over the surrogate-type ML models in efficiency. Although the surrogate models may be faster after sufficiently training, the training of a surrogate model could require extensive efforts including a larger training data set. The training data usually come from setting up and pre-running local numerical models for each cell of interest. In contrast, the PINN solutions can be obtained directly from the training process. Case 5 and 6 are designed at the size similar to realistic problems of downscaling one grid cell of a coarse-scale model: the study domain is 900 m, that is, the typical resolution in large-scale river models (∼O(1 km)); the domain is discretized to 6 m, that is, the high resolution of SWOT (O(10 m)); the 10-day simulation period is also determined based on the average revisit time of satellites at mid-latitudes. Our results imply that the downscaled solution of the interested cell can be obtained in 10–50 min. This time is
longer than a single numerical run. But PINN with its meshless feature reduces the effort of mesh generation in the numerical setup. Normally, the generated mesh in models, such as HEC-RAS, has to be adjusted manually multiple times prior to a successful run, which requires significant efforts when the downscaled cells are multiple. Moreover, the computational cost of PINN can be reduced using parallel computing and GPUs. While it is straightforward to deploy PINN to downscale multiple cells simultaneously, more sophisticated parallelism are now available (Meng et al., 2020; Shukla et al., 2021) to accelerate the process.

### 4.2. Discussion of the Test Results

The results in Case 6 indicate that the availability of observations for assimilation plays an important role in determining the accuracy of the PINN solutions. With limited collocation points, the performance of PINN is degraded with increased variations of the state variables in space and time. For example, the temporal variations of water depth in Case 4 are greater than those in Case 3, resulting in a larger bias. Similarly, the spatially varied flow regime in Case 6 causes the PINN to give less satisfied predictions than the uniform flow regime in Case 4. This issue is related to a common challenge in training PINN. The solutions of the cells that are far from the domain boundaries and have drifted from the initial state can be easily trapped to a trivial local minimum during the PINN inference (Wong et al., 2022) because the constraints of the prescribed boundary and initial conditions.
on the solutions of those cells damp with time. In practical problems, additional constraints can be enforced by assimilating more observational data wherever available. Adding a time series of observations at a location between the boundaries of the computational domain is equivalent to subdividing the domain into subdomains in 1-D problems. The variations of each state variable become less significant in each subdomain, facilitating the learning process. However, even with spatially-varied flow regimes, PINN is less dependent on observations than the linear interpolation in downscaling 1-D channel flow. In practical applications, in-situ observations are likely limited to main gauging stations that are concentrated near the coastal zone (e.g., the downstream section in Case 6). In such a case, PINN is still able to provide reasonable downscaled solutions due to its capability of encoding the governing physics to resolve the topology and assimilating the remote sensing snapshots. The resolved variation of the water depth is of critical importance for delineating more accurate flood inundation maps.

Our proposed Fourier feature embeddings are sufficient to encode the physical characteristics of tide and address the periodicity in BC. However, the PINN solution in Case 6 suggests that the performance is degraded when the flow varies dramatically along the channel. Even though the periodic tidal variations are formulated into the Fourier space, the along-channel dynamics is problem-dependent, as the flow regime is determined by various factors, including the channel topology, the upstream and downstream BCs and their interactions. In Case 6, while the flow is determined by the high-frequency tide in the downstream section, the variation of flow is much smaller in the upstream and middle sections, where the tidal influence is limited and the low-frequency discharge dominates. Because Fourier transformation may only be performed on partial training data, it is challenging to encode the space variable directly in the NN architecture. Moreover, Fourier feature embeddings modulate the variability in data to match the high-frequency patterns. As such, the nonlinear interactions between high-frequency and low-frequency processes are not well reproduced, especially when there are limited observations. In Case 6 for example, the PINN solutions have larger biases near the tips of the tidal propagation where the discharge-tide interaction occurs (Figure 14). Such interaction is expected to be stronger at the upstream reaches of real tidal rivers, and more complex than what we represent in the case since other mechanisms also occur in the context of compound flooding (Dykstra & Dzwonkowski, 2020). This poses further challenges in the realistic application of PINN. Fortunately, the accuracy of PINN depends on the training data. That means, if the proposed PINN solves SVE with sufficient data assimilated, the flow dynamics and discharge-tide interaction would be reasonably

![Figure 11.](image-url)
Figure 12. The spatiotemporal evolution of the water depth in Case 5 simulated by the HEC-RAS model (a), the physics-informed neural network (PINN) model with Fourier feature embeddings (b) and the linear-interpolation scheme (c), and the simulation error of the PINN model (d) and the linear-interpolation scheme (e).

Figure 13. The water stage along a conceptual sloping open river channel simulated by HEC-RAS (reference), the physics-informed neural network (PINN) model without Fourier feature embeddings (PINN), and the linear-interpolation method (interpolation) at four time instants of different discharge conditions (i.e., 20, 50, 100, and 200 hr from the initial time) in Case 5.
emulated. Our results show the accuracy improved with the increase in the amount of assimilated observations. The PINN-based data assimilation provides a new path towards tackling the challenges of compound flood modeling in large-scale earth system models.

The most valuable insight developed in this work is a relatively efficient way of downscaling a coarse-scale model at the subgrid domain using the physics-based ML/AI techniques. The PINN-based downscaling is proved quite handy for assimilating both in-situ and remote sensing data. The latter, in particular, provides a key addition to the numerical modeling community for better resolving the physics without generating subgrid meshes and modifying the model code, thus preserving the computational efficiency. The downscaling method is mesh independent and can always be used to solve a subgrid solution for the interested sub-regions of a large domain, regardless of the original mesh resolution. The downscaling can be potentially extended to the full domain of a coarse-scale model by deploying PINN at all grid cells simultaneously with more advanced computational resources. With the increase in data abundance, the method enables the investigation of subgrid physics even in a high-resolution (∼O(1 m)) model, alleviating limitations in numer-
Figure 15. The water stage along a conceptual sloping open river channel simulated by HEC-RAS (reference), the physics-informed neural network (PINN) model with Fourier feature embeddings (PINN), and the linear-interpolation method (interpolation) at four time instants of different tidal and discharge conditions (i.e., 20, 50, 100, and 200 hr from the initial time) in Case 6. The four panels correspond to the case of 2–5 observations in Table 4, respectively.

Table 4
Information of Simulation Errors in Case 5 and 6

| Case no. | Number of obs. | Location (x) | Downscaling method | \( \epsilon_h \) | RMSE (m) |
|----------|----------------|--------------|--------------------|-----------------|-----------|
| Case 5   | 2              | 100, 200     | PINN               | \( 9.876 \times 10^{-2} \) | 0.387     |
|          |                |              | Linear interpolation | \( 5.181 \times 10^{-1} \) | 2.033     |
| Case 6   | 2              | 100, 200     | PINN               | \( 1.396 \times 10^{-1} \) | 0.674     |
|          |                |              | Linear interpolation | \( 3.586 \times 10^{-1} \) | 1.782     |
|          | 3              | 100, 200, 764 | PINN               | \( 1.121 \times 10^{-1} \) | 0.541     |
|          |                |              | Linear interpolation | \( 1.410 \times 10^{-1} \) | 0.873     |
|          | 4              | 100, 200, 614, 764 | PINN | \( 8.560 \times 10^{-2} \) | 0.413     |
|          |                |              | Linear interpolation | \( 1.288 \times 10^{-1} \) | 0.763     |
|          | 5              | 100, 200, 314, 614, 764 | PINN | \( 8.142 \times 10^{-2} \) | 0.393     |
|          |                |              | Linear interpolation | \( 1.128 \times 10^{-1} \) | 0.557     |

Note. RMSE, root mean squared error; PINN, physics-informed neural network.
Table 5

| Case no. | Scheme | $N_f$ | $N_{data}$ | Reference (s) | PINN (s) | Interpolation (s) |
|----------|--------|-------|------------|---------------|----------|-------------------|
| Case 1   | Standard | 7,380 | 605 | 0.012 | 440 | |
| Case 2   | Standard | 10,980 | 605 | 0.184 | 620 | |
| Case 3   | Standard | 4,896 | 339 | 21 | 158 | |
| Case 4   | Standard | 12,189 | 822 | 36 | 338 | |
| Case 5   | ff | 36,567 | 1,262 | 38 | 615 | 0.016 |
| Case 6   | ff | 36,567 | 1,262 | 42 | 1,529 | 0.018 |
|          | ff | 36,567 | 1,501 | 42 | 1,546 | 0.018 |
|          | ff | 36,567 | 1,740 | 42 | 2,438 | 0.018 |
|          | ff | 36,567 | 1,979 | 42 | 3,147 | 0.018 |

including the channel meandering and widening toward the coast have not been considered either. The latter, in particular, has been found to increase the channel capacity and flow gradient (Fuller et al., 2013). Additionally, general issues, such as observation uncertainty, inconsistent data time series and lack of observations, are common in realistic conditions. While intermittent data are applicable in training without any modifications, a more robust noise-resistance algorithm will be needed to handle data uncertainties or errors. In this study, the noise level is set by USGS in-situ measurement errors. However, the remote sensing data errors are subject to large variability. While the local-scale remote sensing measurements can achieve high accuracy (Bhattacharjee et al., 2021), the large-scale satellite images may suffer larger errors (Biancamaria et al., 2016). Finally, it is necessary to improve the accuracy of PINN in resolving the nonlinear interaction at the river-ocean interface. Overall, the application of PINN in realistic conditions is challenging and has not yet been explored extensively. Even though we have tested and developed algorithms best fitting our purpose, in the future we will also explore state-of-the-art physics-based ML techniques and tools that are emerging in the ML community, such as the general PINN library DeepXDE (Lu, Meng, et al., 2021), the Fourier Neural Operator that models turbulent flows (Z. Li et al., 2020) and algorithms that could potentially improve the PINN robustness to noise (Bajaj et al., 2021).

This study demonstrates the PINN’s capability of solving the 1-D SVE by assimilating data. However, the 1-D approach neglects the vertical stratification and baroclinic forcing during a storm tide propagation (Orton et al., 2012) and limits the application in 2-D flood mapping. As the SWOT satellite imagery provides both 1-D vector and 2-D raster products for water surface elevations, we expect to extend the PINN-based data assimilation approach to a 2-D domain. Even though the 2-D application will bring in the velocity in a different direction and stimulate more interactions, the success of PINN in solving Navier-Stokes equations (Jin et al., 2021; Rao et al., 2020) provides the theoretical basis for tackling 2-D river dynamics in both depth-averaged and longitudinal-averaged form.

In addition to being integrated in a numerical model, PINN can also be used to create a digital twin targeting on regions with complex flow regimes, where large-scale models at their current state are unable to resolve. For example, at the river-ocean interface, 1-D hydraulic models (e.g., HEC-RAS) are usually employed. The hydraulic models take freshwater discharge from hydrologic models at the upstream boundary and water stage information from ocean models at the downstream boundary (Torres et al., 2015). These models are more computationally efficient than 2-D/3-D hydrodynamic models and resolve the flow dynamics and topography better than hydrologic models. Such hydraulic models may be replaced by a digital twin based on PINN, to exchange the information from both sides at the river-ocean interface. Such a digital twin can be developed following a similar procedure as shown in Case 6: (a) the BC data are obtained from the linked river and ocean models at the domain boundaries; (b) the snapshots are provided by satellite images and/or other remote sensing data; (c) in-site measurements are applied as additional constraints.

4.3. Limitations and Future Applications of PINN

In the future, we aim at applying the PINN downscaling to a broader domain with more realistic flow conditions. Our test cases are relatively simple. First, we only apply a constant friction coefficient ($n$). However, $n$ may be considered as a trainable variable in NN because the variable $n$ is often used to account for enhanced friction effects from the nonlinear interaction among tides, storm surge and river discharge (Godin, 1999; Godin & Gonzalez, 1991; Kukulka & Jay, 2003). Second, the mapping matrix representing the Fourier features are sampled from the Gaussian distribution with the bandwidth determined manually through trial and error. As tidal frequencies are usually known in local regions, these parameters may be approximated from the range of tidal frequencies, which adds additional physical constraints on the PINN training. Third, the effects of river channel geometry
5. Conclusion

This research makes the first attempt to apply the state-of-the-art PINN to downscale a large-scale river model to a subgrid solution. The conservation physics of the governing equations regularizes the training data set to a manageable size, which alleviates the burden in data-intensive neural network models. The proposed PINN framework can (a) solve the full dynamic 1-D SVE equation, (b) encode the periodic variation in tidal boundaries using Fourier feature embeddings, (c) assimilate various data types and (d) be integrated to a large-scale river model for downscaled solutions. This work tests the applicability of PINN in flood modeling for delineating more accurate inundation maps, and creates a promising path for applying the method to real-world problems with the emergence of high-quality satellite images in surface water storage and fluxes and more comprehensive in-situ gauge networks.

Data Availability Statement

The data and code to reproduce the results and figures are publicly available at the Zenodo repository (Feng, 2022).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In 12th UNEXSI symposium on operating systems design and implementation (OSDI 16) pp. 265–283.

Alfieri, L., Feyen, L., & Di Baldassarre, G. (2016). Increasing flood risk under climate change: A pan-european assessment of the benefits of four adaptation strategies. Climatic Change, 136(3), 507–521. https://doi.org/10.1007/s10584-016-1461-1

Bajji, C., McLennan, L., Andeen, T., & Roy, A. (2021). Robust learning of physics informed neural networks. arXiv:2110.13330 arXiv preprint.

Bakhtyar, R., Maitaria, K., Velissariou, P., Trimble, B., Mashriqui, H., Moghimi, S., et al. (2020). A new 1D/2D coupled modeling approach for a riverine-estuarine system under storm events: Application to Delaware river basin. Journal of Geophysical Research: Oceans, 125(9), e2019JC015822. https://doi.org/10.1029/2019JC015822

Barnard, P. L., Erikson, L. H., Foxgrover, A. C., Hart, J. A. F., Limber, P., O’Neill, A. C., et al. (2019). Dynamic flood modeling essential to assess the coastal impacts of climate change. Scientific Reports, 9(1), 1–13. https://doi.org/10.1038/s41598-019-40742-z

Bates, P. D., Horritt, M. S., & Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. Journal of Hydrology, 387(1–2), 33–45. https://doi.org/10.1016/j.jhydrol.2010.03.027

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. Journal of Machine Learning Research, 18, 1–43.

Berkhahn, S., Fuchs, L., & Neuweiler, I. (2019). An ensemble neural network model for real-time prediction of urban floods. Water Resources Management, 34(14), 4345–4363. https://doi.org/10.1007/s11269-020-02673-7

Bhattacharjee, J., Rabbil, M., Fazel, N., Darabi, H., Choubin, B., Khan, M. M. R., et al. (2021). Accuracy assessment of remotely sensed data to analyze lake water balance in semi-arid region. Science of the Total Environment, 793, 149034. https://doi.org/10.1016/j.scitotenv.2021.149034

Biancamaria, S., Lettenmaier, D. P., & Pavelsky, T. M. (2016). The swot mission and its capabilities for land hydrology. Water Resources Management, 30(7), 307–337. https://doi.org/10.1007/s11269-015-9346-y

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4). Springer.

Brunner, G. W. (1995). Hec-ras river analysis system. hydraulic reference manual version 1.0. (Tech. Rep.). Hydrologic Engineering Center Davis CA.

Burger, G. (1996). Expanded downscaling for generating local weather scenarios. Climate Research, 7(2), 111–128. https://doi.org/10.3354/ cr007111

Cai, S., Mao, Z., Wang, Z., Yin, M., & Karnaikadak, G. E. (2022). Physics-informed neural networks (PINNs) for fluid mechanics: A review. Acta Mechanica Sinica, 37, 1–12. https://doi.org/10.1007/s10409-021-01148-1

Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karnaikadak, G. E. (2021). Physics-informed neural networks for heat transfer problems. Journal of Heat Transfer, 143(6), 060801. https://doi.org/10.1115/1.4050542

Caldwell, P. M., Mametjanov, A., Tang, Q., Van Roekel, L. P., Golaz, J.-C., Lin, W., et al. (2019). The doe E3SM coupled model version 1: Description and results at high resolution. Journal of Advances in Modeling Earth Systems, 11(12), 4095–4146. https://doi.org/10.1029/2019ms001870

Cedillo, S., Sánchez-Cordero, E., Samaniego, E., & Alvarado, A. (2022). Exploratory study of physic informed deep learning applied to a step-pool for different flow magnitudes. In Communication, smart technologies and innovation for society (pp. 275–284). Springer.

Crompton, O., Sysma, A., & Thompson, S. (2019). Emulation of the saint venant equations enables rapid and accurate predictions of infiltration and overland flow velocity on spatially heterogeneous surfaces. Water Resources Research, 55(8), 7108–7129. https://doi.org/10.1029/2019wr025146

Cunge, J. (1980). Practical aspects of computational river hydraulics (p. 420). Pitman Publishing Ltd. London, (17 CUN).

de Avila Belbute-Pereira, F., Chen, Y-F., & Sha, F. (2021). HyperPINN: Learning parameterized differential equations with physics-informed hypernetworks. In The symposium of deep learning and differential equations. Retrieved from https://openreview.net/forum?id=LxUaRDUhRjM

De Almeida, G. A., & Bates, P. (2013). Applicability of the local inertial approximation of the shallow water equations to flood modeling. Water Resources Research, 49(8), 4833–4844. https://doi.org/10.1002/wrcr.20366

Domeneghetti, A., Schumann, G.-P., Frasson, R. D. M., Wei, R., Pavelsky, T., Castellarin, A., et al. (2018). Characterizing water surface elevation under different flow conditions for the upcoming swot mission. Journal of Hydrology, 561, 848–861. https://doi.org/10.1016/j.jhydrol.2018.04.046
Torres, J. M., Bass, B., Izza, N., Fang, Z., Proft, J., Dawson, C., et al. (2015). Characterizing the hydraulic interactions of hurricane storm surge and rainfall–runoff for the Houston–Galveston region. Coastal Engineering, 106, 7–19. https://doi.org/10.1016/j.coastaleng.2015.09.004

Towner, J., Cloke, H. L., Zouter, E., Flamig, Z., Hoch, J. M., Bazo, J., et al. (2019). Assessing the performance of global hydrological models for capturing peak river flows in the amazon basin. Hydrology and Earth System Sciences, 23(7), 3057–3080. https://doi.org/10.5194/hess-23-3057-2019

Trotta, F., Federico, I., Pinardi, N., Coppini, G., Causio, S., Jansen, E., et al. (2021). A relocatable ocean modeling platform for downscaling to shelf-coastal areas to support disaster risk reduction. Frontiers in Marine Science, 8, 317. https://doi.org/10.3389/fmars.2021.642815

Turnipseed, D. P., & Sauer, V. B. (2010). Discharge measurements at gaging stations. (Tech. Rep.). US Geological Survey.

van der Meer, R., Oosterlee, C. W., & Borovyskh, A. (2022). Optimally weighted loss functions for solving PDEs with neural networks. Journal of Computational and Applied Mathematics, 405, 113887. https://doi.org/10.1016/j.cam.2021.113887

Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM Journal on Scientific Computing, 43(5), A3055–A3081. https://doi.org/10.1137/20m1318043

Wang, S., Wang, H., & Perdikaris, P. (2021). On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering, 384, 113938. https://doi.org/10.1016/j.cma.2021.113938

Ward, N. D., Megonigal, J. P., Bond-Lamberty, B., Bailey, V. L., Butman, D., Canuel, E. A., et al. (2020). Representing the function and sensitivity of coastal interfaces in Earth system models. Nature Communications, 11(1), 1–14. https://doi.org/10.1038/s41467-020-16236-2

Wilby, R. L., & Dawson, C. W. (2013). The statistical downscaling model: Insights from one decade of application. International Journal of Climatology, 3(7), 1707–1719. https://doi.org/10.1002/joc.3544

Wong, J. C., Ooi, C., Gupta, A., & Ong, Y.-S. (2022). Learning in sinusoidal spaces with physics-informed neural networks. IEEE Transactions on Artificial Intelligence.

Wood, E. F., Roundy, J. K., Troy, T. J., Van Beek, L., Bierkens, M. F., Blyth, E., et al. (2011). Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth’s terrestrial water. Hydrology and Earth System Sciences, 15(1), 1–14. https://doi.org/10.1016/j.coastaleng.2015.09.004

Xue, Y., Janjic, Z., Dudhia, J., Vasic, R., & De Sales, F. (2014). A review on regional dynamical downscaling in intraseasonal to seasonal simulation/prediction and major factors that affect downscaling ability. Atmospheric Research, 147, 68–85. https://doi.org/10.1016/j.atmosres.2014.05.001

Yamazaki, D., Kanae, S., Kim, H., & Oki, T. (2011). A physically based description of floodplain inundation dynamics in a global river routing model. Water Resources Research, 47(4), W04501. https://doi.org/10.1029/2010wr009726

Yamazaki, D., Lee, H., Alsdorf, D. E., Dutra, E., Kim, H., Kanae, S., & Oki, T. (2012). Analysis of the water level dynamics simulated by a global river model: A case study in the amazon river. Water Resources Research, 48(9), 2012. https://doi.org/10.1029/2012wr1011869

Yang, Y., Pan, M., Lin, P., Beck, H. E., Zeng, Z., Yamazaki, D., et al. (2021). Global reach-level 3-hourly river flood reanalysis (1980–2019). Water Resources Research, 47(4), 4001–4024. https://doi.org/10.1029/2010wr010090

Xiao, Z., Yang, Z., Wang, T., Sun, N., Wingham, M., & Judd, D. (2021). Characterizing the non-linear interactions between tide, storm surge, and river flow in the Delaware bay estuary, United States. Frontiers in Marine Science, 8, 715557. https://doi.org/10.3389/fmars.2021.715557

Xue, Y., Janjic, Z., Duda, J., Vasic, R., & De Sales, F. (2014). A review on regional dynamical downscaling in intraseasonal to seasonal simulation/prediction and major factors that affect downscaling ability. Bulletin of the American Meteorological Society, 105(11), E2086–E2105. https://doi.org/10.1175/bams-d-20-0057.1

Yu, C.-W., Hodges, B. R., & Liu, F. (2020). A new form of the saint-venant equations for variable topography. Hydrology and Earth System Sciences, 24(8), 4001–4024. https://doi.org/10.5194/hess-24-4001-2020

Yu, C.-W., Liu, F., & Hodges, B. R. (2017). Consistent initial conditions for the saint-venant equations in river network modeling. Hydrology and Earth System Sciences, 21(9), 4959–4972. https://doi.org/10.5194/hess-21-4959-2017

Yu, J., Lu, L., Meng, X., & Karnaiaakis, G. E. (2022). Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. Computer Methods in Applied Mechanics and Engineering, 393, 114823. https://doi.org/10.1016/j.cma.2022.114823

Zhang, R., Zen, R., Xing, J., Arsa, D. M. S., Saha, A., & Bressan, S. (2020). Hydrological process surrogate modelling and simulation with neural networks. In Advances in knowledge discovery and data mining. 12085 (pp. 449–461). https://doi.org/10.1007/978-3-030-47436-2_34

Zhang, Y. J., Ye, F., Yu, H., Sun, W., Moghim, S., Myers, E., et al. (2020). Simulating compound flooding events in a hurricane. Ocean Dynamics, 70(5), 621–640. https://doi.org/10.1007/s10236-020-01351-x

Zhu, L., Zhang, Y., Wang, J., Tian, W., Liu, Q., Ma, G., et al. (2021). Downscaling snow depth mapping by fusion of microwave and optical remote-sensing data based on deep learning. Remote Sensing, 13(4), 584. https://doi.org/10.3390/rs13040584