Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Transfer learning-enhanced physics-informed neural networks for efficient simulation of river flood in complex large-scale river systems

Kang Zheng

809441140@qq.com

China Institute of Water Resources and Hydropower Research https://orcid.org/0009-0000-0603-9131

Xin Qi

China Institute of Water Resources and Hydropower Research

Dawei Zhang

China Institute of Water Resources and Hydropower Research

Xinzheng Feng

China Institute of Water Resources and Hydropower Research

Weiqi Wang

China Institute of Water Resources and Hydropower Research

Research Article

Keywords: Physics-informed neural network, Saint-Venant equations (SVEs), Transfer learning, Flood simulation

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11382325/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Transfer learning-enhanced physics-informed neural networks for efficient simulation of river flood in complex large-scale river systems

Kang Zheng [1],Xin Qi[1], Dawei Zhang[1, a] , Xinzheng Feng[1],Weiqi Wang[1]
1. State Key Laboratory of Water Cycle and Water Security, China Institute of Water Resources and Hydropower Research, Beijing, 100038, P. R. China
a). Author to whom correspondence should be addressed: zhangdw@iwhr.com

ABSTRACT

Physics-informed neural networks (PINNs) have shown considerable potential for solving hydraulic problems by integrating governing physical laws into neural network optimization. However, their application to real-world river flood remains challenging because of the high computational cost associated with nonlinear and non-convex PINN optimization. Conventional PINNs are generally initialized without task-specific hydraulic knowledge, each new hydraulic scenario requires optimization from an uninformed initial state, resulting in repeated training and substantial computational expense. This study proposes a transfer learning method that can significantly improve the prediction accuracy and training efficiency of PINNs for complex river floods under varying flood events. A transfer learning-enhanced PINN (TL-PINN) framework is developed, in which a data-driven neural network is first trained on a source flood event. The optimized parameters are then transferred to initialize the PINN for target flood events, followed by optimization using a composite physics-informed loss function. The proposed framework is evaluated on a real-world river reach with complex channel geometry and highly unsteady flood processes through three numerical experiments. TL-PINN is compared with three conventionally initialized PINNs using PyTorch Default Initialization, Orthogonal Initialization and Xavier Normal Initialization. TL-PINN consistently maintains superior performance under limited labeled-data conditions and under more severe and complex flood inflows. The results indicate that the transferred parameters provide an effective initialization, leading to improved prediction accuracy and optimization efficiency. These findings demonstrate that transfer learning might significantly enhance the real-world applicability of PINNs for efficient and accurate river flood simulations under varying flood events.

Keywords: Physics-informed neural network; Saint-Venant equations (SVEs); Transfer learning; Flood simulation

1 Introduction

River flood simulation is one of the fundamental problems in computational hydraulics and underpins numerous applications, including flood forecasting, reservoir operation, hydraulic structure design, and river management. River flood processes can be described by the one-dimensional Saint-Venant equations (SVEs) (Cunge et al., 1980), which represent the conservation of mass and momentum for unsteady open-channel flow. Owing to their physical foundation and proven predictive capability, numerical solutions of the SVEs have been widely used for river flood simulation over the past several decades. A variety of numerical methods, including Finite Difference (FD) (Casulli, 1990; Kurganov & Levy, 2002; Molls & Chaudhry, 1995), Finite Volume (FV) (Alcrudo & Garcia‐Navarro, 1993; Bale et al., 2003; Botta et al., 2004; Toro & Garcia-Navarro, 2007; Yoon & Kang, 2004), and Finite Element (FE) (Dawson et al., 2006; Hanert et al., 2005; Lynch & Gray, 1979) have been successfully developed to provide accurate and robust hydraulic solutions under a wide range of flow conditions (LeVeque et al., 2011; Liang & Marche, 2009; Toro & Garcia-Navarro, 2007; Wilson et al., 2007). Nevertheless, repeated simulations under different hydraulic scenarios remain computationally demanding (Bernard et al., 2009; Juez et al., 2014; Liang, 2011), particularly when multiple flood events or boundary conditions must be considered in engineering practice.

Recent advances in scientific machine learning (SciML) have provided new opportunities for solving partial differential equations (PDEs). Among these approaches, physics-informed neural networks (PINNs) (Raissi et al., 2019) have attracted considerable attention because they reduce the reliance on labeled training data for ML models by incorporating governing equations directly into the training process through automatic differentiation (Raissi et al., 2019). This framework provides a promising alternative to conventional SciMLs and has demonstrated encouraging performance in solving a variety of forward and inverse PDE problems, including computational fluid dynamics (CFD) (Eivazi et al., 2022; Iskhakov & Dinh, 2020; Jin et al., 2021; Mao et al., 2020; Yang et al., 2019).

The application of PINNs has expanded rapidly to computational hydraulic. Existing studies have demonstrated their capability in solving various flow-related problems governed by Shollow Water equations (SWEs). For example, Bihlo and Popovych (2022) used physics-informed neural networks for solving the SWEs on the sphere in the meteorological context. Moreover, Li et al. (2023) proposed a physical information-fused deep learning framework with a subregion-specific sampling strategy for flood dynamics prediction, highlighting the potential of PINNs for solving flood-related flow problems. More recently, Qi et al. (2024) developed a convolutional neural network-based PINN framework for solving the 2D SWEs and demonstrated that the proposed architecture could improve prediction accuracy and training efficiency compared with conventional fully connected PINNs. Dazzi (2024) investigated PINNs for the augmented system of SWEs with

complex topography.

Despite these advantages, the practical application of PINNs to real-world hydraulic problems remains challenging, primarily due to the high computational cost and optimization complexity during model training. Unlike conventional numerical solvers, which directly obtain numerical solutions after discretizing the governing equations under prescribed initial and boundary conditions, PINNs formulate the solution as a neural network and iteratively optimize network parameters by minimizing a composite loss function. Before obtaining a reliable solution, PINN often require multiple attempts to determine appropriate model configurations, including network architectures (i.e. hyperparameters such as the number of neurons, hidden layers, and convolutional layers and channels), loss criteria (e.g. the composition of the loss function and the balance among different loss components), optimizer selection, learning rate schedules, and the number of training epochs. These factors significantly influence the performance of PINNs.

Previous studies have attempted to improve PINN performance through enhanced network architectures (Cai, Mao, et al., 2021; Gu et al., 2024; Jagtap, Kharazmi, et al., 2020; Qi et al., 2024), adaptive or automatically weighted loss (Bischof & Kraus, 2025; Kendall et al., 2018; Qi et al., 2025), and improved optimization strategies (Jagtap, Kawaguchi, et al., 2020; Krishnapriyan et al., 2021; Wang et al., 2022). These efforts have significantly enhanced the performance of PINNs under various circumstances. However, an important aspect has received comparatively limited attention: the initialization of network parameters. In conventional PINN training, each new task might starts the optimization process from an uninformed initial state (Wang et al., 2020). As a result, this optimization might introduces considerable computational cost and limits the efficiency of PINNs for hydraulic simulations in the same or similar domain.

Parameter initialization has long been recognized as an important factor affecting neural-network training because the initial parameter state influences the propagation of activations and gradients and, consequently, the stability and convergence of optimization. Inappropriate initialization may lead to activation saturation, vanishing or exploding gradients, slow convergence, or even failure of network training (Glorot & Bengio, 2010; Narkhede et al., 2022; Pascanu et al., 2013). With the development of machine learning and deep learning, several initialization methods have been widely adopted. LeCun et al. (2002) introduced a fan-in-based weight-scaling strategy in which the magnitude of the initial weights is adjusted according to the number of incoming connections. Glorot and Bengio (2010) proposed Xavier initialization, which determines the weight variance jointly from the fan-in and fan-out of each layer to regulate the propagation of activations and gradients. Saxe et al. (2013) investigated orthogonal initialization by constructing orthogonal or semi-orthogonal weight matrices to facilitate signal and gradient propagation. He et al. (2015) proposed a variance-scaling initialization specifically adapted to rectified nonlinearities, while Mishkin and Matas (2015) introduced layer-sequential unit-variance initialization by combining orthogonal initialization with layer-wise variance normalization. Modern deep-learning frameworks

also provide default initialization schemes; for example, PyTorch automatically initializes the weights of fully connected and convolutional layers according to fan-in-dependent random distributions when no explicit initialization method is specified (Paszke et al., 2019).

Although these initialization strategies can improve network training and performance to varying degrees, the effectiveness depends on network architecture, activation function and application scenario, and no single conventional initialization method is universally optimal (de Sousa, 2016; Narkhede et al., 2022). More importantly, their initial parameters are generated according to predefined statistical or structural rules and do not explicitly incorporate knowledge learned from previously solved physical tasks. Transfer learning provides a fundamentally different strategy by reusing knowledge acquired from related tasks to facilitate subsequent learning (Guan et al., 2022; Pan & Yang, 2009). Instead of generating target-model parameters solely from predefined initialization rules, transfer learning uses parameters learned from a related source task to provide a task-informed starting point for optimization. Goswami et al. (2020) developed a transfer-learning-enhanced PINN for phase-field fracture modeling and demonstrated that transferred parameters could accelerate convergence and reduce computational cost. Chakraborty (2021) showed that knowledge learned from previously solved physical problems could be reused in related PDE tasks to improve training efficiency and prediction accuracy. Xu et al. (2023) investigated transfer-learning-based PINNs for inverse problems in engineering structures under different loading scenarios, while Liu et al. (2023) proposed an adaptive transfer-learning strategy to improve knowledge transfer between related physical problems.

Despite the growing interest in transfer learning in deep learning and SciML, its application to PINNs for computational hydraulics remains limited. Its potential may be particularly relevant to river flood simulations under varying flood events. In practical applications, a given river reach is commonly evaluated under multiple flood scenarios, including design floods with different return periods and observed flood events, for which different inflow hydrographs are prescribed as upstream boundary conditions (Cunge et al., 1980; Teng et al., 2017). Although the inflow hydrographs differ among flood events, the channel geometry, roughness characteristics and domain scale are shared across these simulations. Consequently, these river flood problems are closely related physical systems that share substantial hydraulic information. Under such circumstances, conventional initialization strategies still require each target PINN to begin from a newly generated parameter state, whereas transfer learning may offer the possibility of exploiting the shared hydraulic knowledge for PINN among different flood events within the same river reach. Such a strategy has the potential to improve both prediction accuracy and training efficiency.

In this study, we present a transfer learning-enhanced physics-informed neural network (TL-PINN) framework designed to enable efficient one-dimensional river flood simulations under varying flood events.The proposed framework first trains a data-driven neural network using a source flood event and transfers the optimized parameters to initialize the PINN for target flood

event. The initialized TL-PINN is subsequently optimized using a composite physics-informed loss function, allowing the transferred parameters to adapt to the target flood event. The objective of this study is to investigate whether transfer learning can simultaneously improve the prediction accuracy and training efficiency of PINNs for one-dimensional river flood under varying flood events.

The proposed framework is evaluated using a real-world river reach along the main stream of the Xiaoqing River, China, characterized by complex channel geometry, highly unsteady flood process, and a computational length of approximately 10 km. Multiple flood scenarios with different upstream boundary conditions are employed to comprehensively evaluate the effectiveness of the proposed transfer learning strategy. The prediction accuracy and training behavior of the proposed TL-PINN are systematically compared with those of PINNs using the PyTorch Default Initialization, Orthogonal Initialization and Xavier Normal Initialization .

The main contributions of this study are summarized as follows:

- A transfer learning-enhanced physics-informed neural network framework is proposed for efficient one-dimensional river flood simulation under varying flood events.
- The proposed transfer learning strategy provides effective parameter initialization for PINN optimization, significantly improving both prediction accuracy and training efficiency compared with PINNs using conventional initialization strategies, including the PyTorch Default Initialization, Orthogonal Initialization, and Xavier Normal Initialization.
- The proposed framework is validated through the real-world river flood events under varying boundary conditions, demonstrating its applicability to practical computational hydraulic problems.

The remainder of this paper is organized as follows. The study area is introduced in Section 2 and Section 3 describes the governing equations, the PINN framework, and the proposed transfer learning strategy. In Section 4, the prediction accuracy and training behavior of TL-PINN and conventionally initialized PINNs are investigated through three numerical experiments. The main findings are discussed and summarized in Section 5.

2 Study area

As shown in Fig. 1, the study area is located along the downstream reach of the Xiaoqing River, Shandong Province, China. The Xiaoqing River is an important river system in northern Shandong Province, characterized by a typical temperate monsoon climate, with precipitation mainly concentrated during the summer flood season. Previous numerical investigations have shown that variations in upstream inflow and downstream boundary conditions can substantially influence water elevation and flow characteristics along the river channel (e.g., flood and tidal interaction in downstream reaches) (Wenyan et al., 2025). These characteristics might make the Xiaoqing River an appropriate case study for evaluating advanced hydraulic simulation approaches under varying flood

events.

In this study, a representative reach of approximately 10 km along the Xiaoqing River was selected. The computational domain contains 26 measured cross-sections, providing detailed geometric information for constructing the one-dimensional hydraulic model. Multiple flood events with different upstream inflow hydrographs were considered for the same reach, including five flood inflows with varying return periods and one double-peak flood inflow with a highly different shape (Fig. 2).

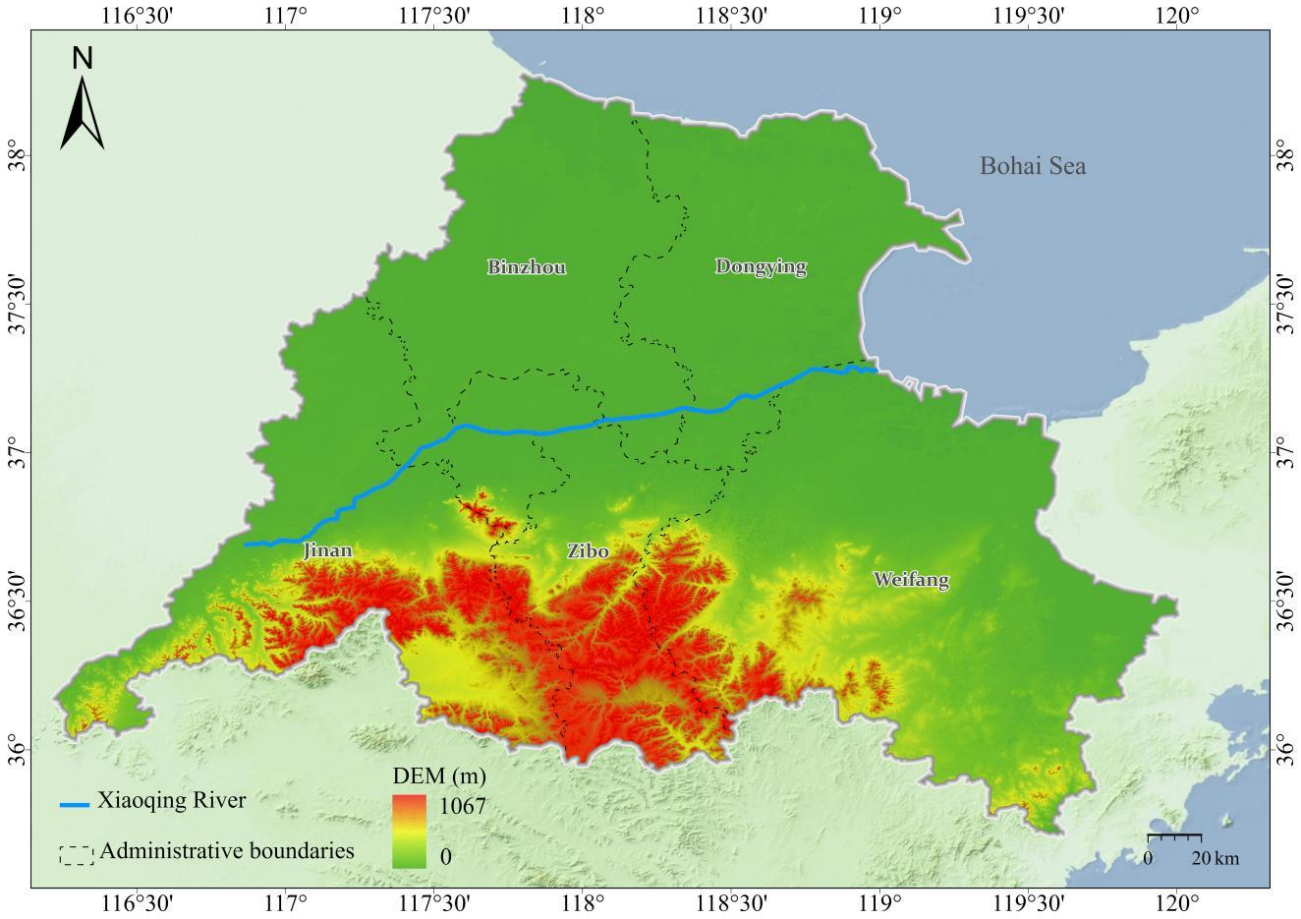


Fig. 1. Location of the study area and the selected river reach in the Xiaoqing River, Shandong Province, China.

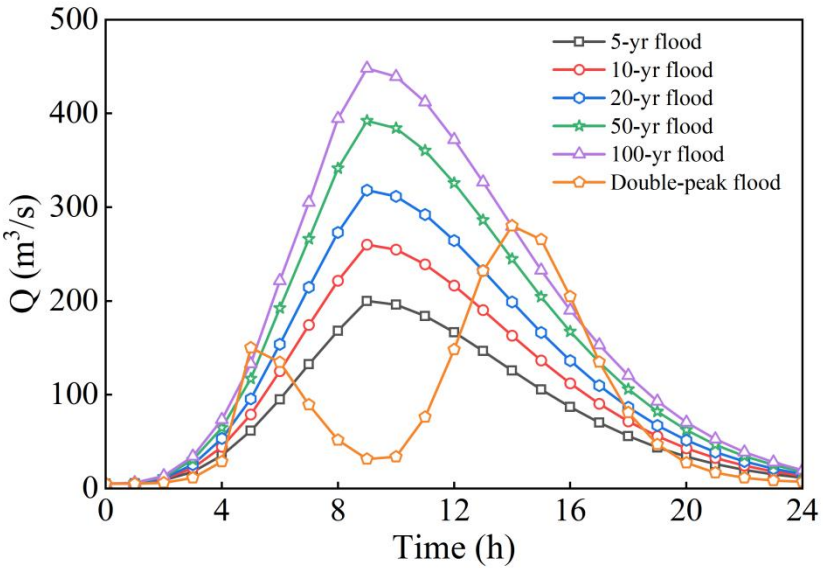


Fig. 2. Upstream inflow hydrographs used for different flood scenarios, including 5-, 10-, 20-, 50-, and 100-year return period flood and the double-peak flood event.

3 Methods

3.1 One-Dimensional Saint-Venant Equations

The one-dimensional Saint-Venant equations (SVEs) are derived from the Navier–Stokes equations under the assumptions of hydrostatic pressure distribution and one-dimensional flow approximation (Chaudhry; Cunge et al., 1980). They represent the conservation of mass and momentum for unsteady open-channel flows and are widely used to describe the propagation of flood waves in natural rivers. The equations govern the temporal and spatial variations of flow depth and discharge along the river channel and can be expressed as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S} \tag{1}$$

$$\mathbf{U} = \begin{bmatrix} A \\ Q \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} Q \\ \frac{Q^2}{A} + gI_1 \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 0 \\ gh(S_0 - S_f) \end{bmatrix} \tag{2}$$

where A is the cross-sectional wetted area; Q is the flow discharge in cross section; gI_1represents the hydrostatic pressure contribution in the momentum equation, and I_1 denotes the hydrostatic pressure integral, which can be numerically evaluated using the discretized cross-sectional geometry; S_f is the friction slope; S_0 is the elevation slope of terrain; t is time; and x is the x-direction coordinate.

The friction slope cannot be obtained analytically in most hydraulic applications. In this study, it is estimated using Manning's equation, which is widely used for open-channel flow. By rearranging Manning's equation, the friction slope can be expressed as Eq. (3a). The bed slope is

calculated from variations in riverbed elevation derived from bathymetric data, and is given by Eq. (3b).

$$S_f = (\frac{nv}{KR^{3/4}})^2 \tag{3a}$$

$$S_0 = -\frac{dz}{dx} \tag{3b}$$

where n denotes Manning's roughness coefficient, which is set to 0.03 in this study to represent the roughness condition of a natural river channel; v is the cross-sectional flow velocity; K is a unit conversion factor (equal to 1 for SI units and 1.486 for imperial units); R is the hydraulic radius; and z is the bed elevation relative to a specified datum.

Furthermore, these hydraulic parameters $(A, P, R, \text{and } B)$ are functions exclusively of the cross-sectional water depth and the terrain shape. Since the terrain shape is derived from DEM and bathymetry data, we denote the raster digital elevation data at all cross-sections as $DEMx$. Consequently, these parameters can be reformulated as a general function involving $DEMx$, expressed as:

$$[A, R, P, B] = f(DEM_x, \eta) \tag{4}$$

where η is the water level.

3.2 PINN Framework

Recently, deep learning architectures have been increasingly used in PINN studies, among which fully connected neural networks (FCNNs) can be regarded as the most common choices. FCNNs are widely adopted because of their simple structure, while Convolutional Neural Networks (CNNs) have also shown good performance in many studies (Cai, Wang, et al., 2021; Li et al., 2021; Zhu et al., 2019). Previous studies have indicated that PINN model based on CNNs might give a more accurate prediction and higher training speed than FCNNs for shallow water problems (Qi et al., 2024). Therefore, this study adopts the CNN-based PINNs for solving the flood problems in the Xiaoqing River.

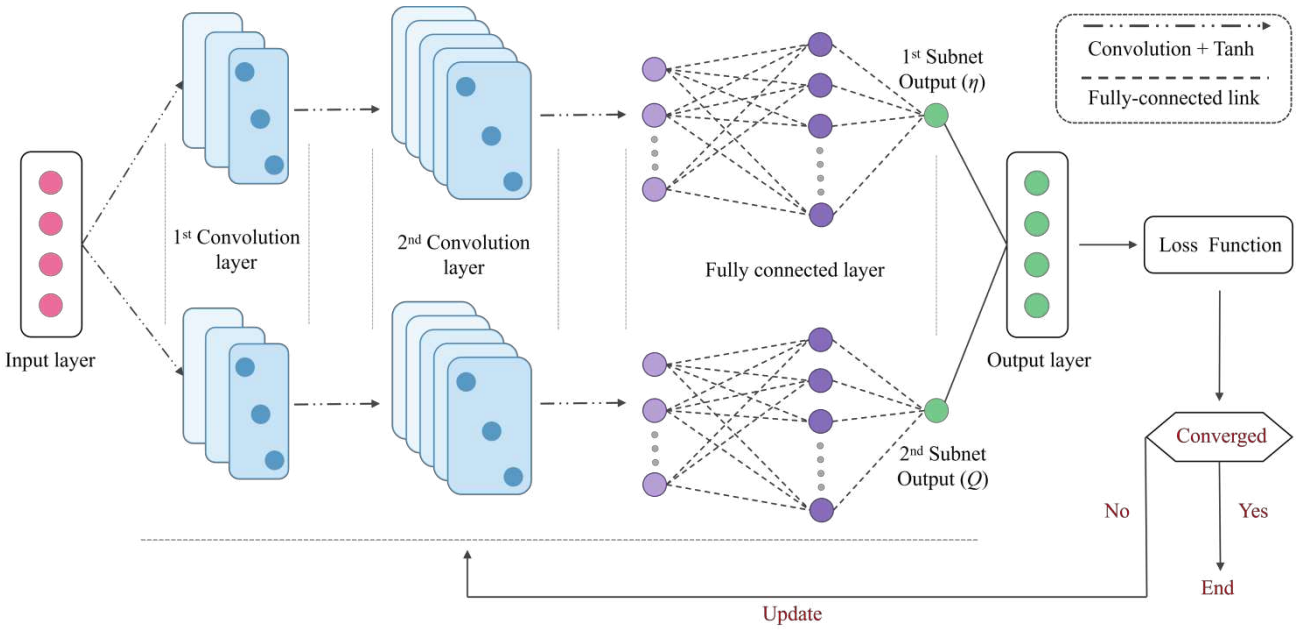


Fig. 3. A schematic diagram of a Physics Informed Neuronal Network (PINN) for finding approximate solutions to the SVEs.

Fig. 3 illustrates the PINN architecture used in this study. The model consists of two independent sub-networks, which are used to predict water level and discharge, respectively. Both sub-networks take the same input variables, but their trainable parameters are not shared. This separate-output design decouples the learning process of different hydraulic variables and can improve prediction performance in multi-output problems, especially when the target variables differ in magnitude or distribution (Gao et al., 2021; Qi et al., 2024; Sun et al., 2020). The spatio-temporal coordinates (x, t) serve as the inputs of the neural network, while $\widetilde{\mathbf{Y}} = (\eta, Q)$ denote the outputs of the neural network. Specifically, the neural network approximation can be expressed as $(\eta, Q) = \aleph(x, t, \Theta)$, here $\aleph$ denotes the trained model and Θ represents the trainable network parameters.

The proposed PINN is trained by minimizing a composite loss function, which is defined as follows:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{ic} + \lambda_2 \cdot \mathcal{L}_{bc_u} + \lambda_3 \cdot \mathcal{L}_{bc_d} + \lambda_4 \cdot \mathcal{L}_{pde} + \lambda_5 \cdot \mathcal{L}_l \tag{5}$$

where $\mathcal{L}$ is the composite loss function to be minimized, and λ_{1-5} are penalty coefficients for the corresponding loss terms. $\mathcal{L}_{ic}$ denotes the initial condition loss, $\mathcal{L}_{bc_u}$ and $\mathcal{L}_{bc_d}$ denote the residual losses of the upstream and downstream boundary conditions, respectively. $\mathcal{L}_{pde}$ represents the physics residual loss associated with the mass and momentum conservation equations, and $\mathcal{L}_l$ denotes the residual loss of the labeled data. These loss terms are defined as follows:

$$\mathcal{L}_{ic} = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \left| (\widetilde{\mathbf{Y}}_{ic} - \mathbf{Y}_{ic}) \right| \tag{6a}$$

$$\mathcal{L}_{bc_u} = \frac{1}{N_{bc_up}} \sum_{i=1}^{N_{bc_u}} \left| (\widetilde{\mathbf{Y}}_{bc_u} - \mathbf{Y}_{bc_u}) \right| \tag{6b}$$

$$\mathcal{L}_{bc_d} = \frac{1}{N_{bc_d}} \sum_{i=1}^{N_{bc_d}} \left| (\widetilde{\mathbf{Y}}_{bc_d} - \mathbf{Y}_{bc_d}) \right| \tag{6c}$$

$$\mathcal{L}_{pde} = \frac{1}{N}\sum_{i=1}^{N}\left|\partial \mathbf{U}(\tilde{\mathbf{Y}})/\partial t + \partial \mathbf{F}(\tilde{\mathbf{Y}})/\partial x - \mathbf{S}(\tilde{\mathbf{Y}})\right| \quad (6d)$$

$$\mathcal{L}_{l} = \frac{1}{N_l}\sum_{i=1}^{N_l}\left|(\tilde{\mathbf{Y}}_l - \mathbf{Y}_{\boldsymbol{l}})\right| \quad (6e)$$

where solutions $\mathbf{Y} = (\eta, Q)$ can be computed by several traditional numerical methods (e.g. FV) and the subscripts ('ic', 'bc_u', 'bc_d', and 'l') denote the collocation points corresponding to the initial condition, upstream boundary condition, downstream boundary condition, and labeled data, respectively. And N is the total number of such points randomly sampled from the spatiotemporal domain.

3.3 Transfer learning

In deep neural networks, one of the most commonly adopted transfer learning strategies is parameter transfer, where pretrained parameters learned from a source task are reused to initialize or adapt models for a target task (Guan et al., 2022; Pan & Yang, 2009; Yosinski et al., 2014; Zhuang et al., 2020). According to the degree of parameter adaptation during target-task training, parameter transfer strategies can generally be categorized into feature extraction and fine-tuning approaches. In feature extraction-based transfer learning, the transferred parameters are partially frozen and only newly added or task-specific layers are optimized using target-task data (Yosinski et al., 2014; Zhuang et al., 2020). In contrast, fine-tuning strategies initialize the target model using pretrained parameters and subsequently update part or all of the transferred parameters to adapt the learned representations to the target task (Howard & Ruder, 2018; Kornblith et al., 2019).

The selection of an appropriate fine-tuning strategy depends mainly on the relationship between the source and target tasks. When the source and target tasks share similar underlying structures, full-network fine-tuning, in which all pretrained parameters are continuously optimized during target-task training, provides greater flexibility for model adaptation and allows the transferred knowledge to be effectively adjusted according to the target conditions (Kornblith et al., 2019). Conversely, partial parameter updating with frozen layers may be more suitable when the source and target tasks exhibit significant differences, as it preserves general features learned from the source task while reducing the risk of over-adaptation.

In the present study, the source and target tasks correspond to different flood events occurring within the same river reach. Although the upstream inflow hydrographs vary among flood events, the governing equations, channel geometry, roughness characteristics, and computational domain remain highly consistent. Therefore, making full-network fine-tuning might be an appropriate strategy for transferring knowledge between related tasks.

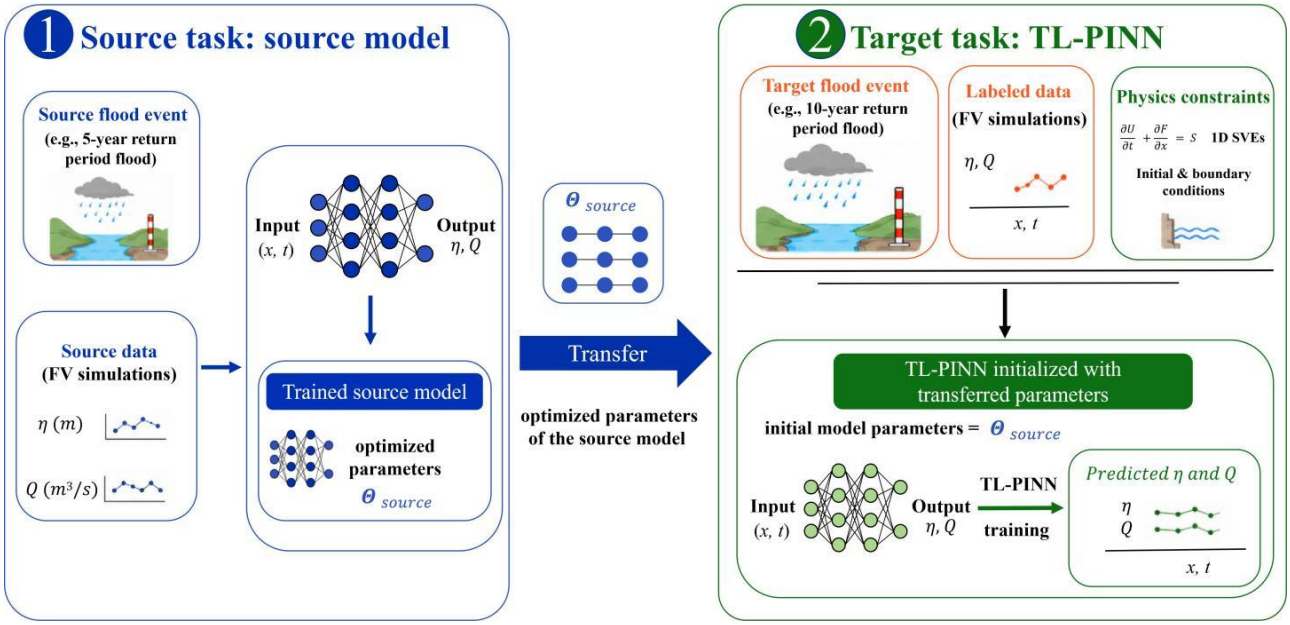


Fig. 4. A schematic workflow of the proposed transfer learning-enhanced PINN framework, including source-task learning, parameter transfer, and target-task optimization.

Fig. 4 illustrates the transfer learning strategy used in this study. The TL-PINN adopts a full-parameter transfer strategy. The optimized parameters of the source neural network is trained by using the purely data-driven method. The parameter of the source model are directly transferred to initialize the TL-PINN. Specifically, the initial parameters of the TL-PINN are assigned as:

$$\Theta_{TL-PINN}^{initial} = \Theta_{source} \tag{7a}$$

where $\Theta_{TL-PINN}^{initial}$ represents the initial parameters of TL-PINN, whereas Θ_{source} denotes the optimized parameters of the source model trained from the source flood event. After parameters transfer, the TL-PINN is optimized using the physics-informed loss function described in Section2.2. During the optimization process, all transferred parameters remain trainable and are updated without additional constraints or manually designed parameter-freezing strategies, expressed as:

$$\Theta_{TL-PINN}^{*} = \arg\min_{\Theta} \mathcal{L} \tag{7b}$$

where $\Theta_{TL-PINN}^{*}$ represents the parameters of TL-PINN during the optimization process by minimizing the total loss.

3.4 PINN design and evaluation

To investigate the influence of parameter initialization on PINN optimization, three representative initialization strategies were selected to initialize the PINN models: the PyTorch default initialization (DE-PINN), orthogonal initialization (OR-PINN), and Xavier normal initialization (XN-PINN), for comparison with TL-PINN. These strategies were selected to provide conventional baselines with distinct parameter-initialization mechanisms (Narkhede et al., 2022),

representing a framework-level default scheme, a structure-based orthogonal scheme, and a variance-scaling scheme, respectively. The relevance of initialization to hydraulic PINNs has been explicitly recognized in previous studies; for example, Qi et al. (2024) identified model initialization as a potential source of sensitivity in PINNs for the two-dimensional SWEs. Xavier initialization has also been directly adopted in PINN applications involving the SVEs and two-dimensional SWEs (Feng et al., 2023; Tian et al., 2025). Beyond hydraulic applications, orthogonal initialization has been investigated in neural PDE solvers for nonlinear evolutionary equations such as the Burgers equation and has shown competitive optimization performance relative to other conventional initialization schemes (Kassaï Koupaï et al., 2024). Table 1 shows the parameter-generation mechanisms of four PINN models.

Table 1. Initialization strategies, parameter distributions, and formulations used in the PINN models

Model	Initialization strategy	Formulation
DE-PINN	PyTorch Default Initialization (Paszke et al., 2019)	$W_{ij} \sim U(-1/\sqrt{fan_{in}}, 1/\sqrt{fan_{in}})$
OR-PINN	Orthogonal Initialization	$W_{ij} = g\,Q_{ij}, QQ^T = I$
XN-PINN	Xavier Normal Initialization	$W_{ij} \sim N(0, \sigma_W^2), \sigma_W = g\sqrt{2/(fan_{in} + fan_{out})}$
TL-PINN	Transfer learning	$\Theta_{TL-PINN}^{initial} = \Theta_{source}$

Note: W_{ij} represents the individual weight connecting the i-th input unit to the j-th output unit. U denotes a uniform distribution, whereas N denotes a normal distribution. fan_{in} and fan_{out} represent the numbers of incoming and outgoing connections of a layer, respectively. g denotes the initialization gain used to adjust the weight scale according to the activation function. For orthogonal initialization, Q denotes an orthogonal or semi-orthogonal matrix satisfying the corresponding orthogonality constrain.

The performance of the DE-PINN, XN-PINN, OR-PINN, and TL-PINN was systematically evaluated and compared. Since there is currently no universal design approach to determine the optimal structure for a neural network (Bihlo &Popovych, 2022), the basic selection rule for PINN design is to find a structure with the lowest possible complexity that achieves the desired accuracy of prediction. In this study, the architecture of the neural network consist of 2 convolutional layers (the first and second layers have 10 and 80 channels, respectively) and 1 fully connected hidden layer with 100 neurons. A total of 2000 epochs were used for all numerical experiments. Additionally, this paper uses the hyperbolic tangent activation function (Tanh) for neural networks. And we employed the Batch Normalization method proposed by Ioffe and Szegedy (2015) to normalize signals between adjacent layers. It can help accelerate PINN training by reducing the internal parameter shift, leading to faster convergence. A maximum of 3000 training epochs was allowed for all numerical experiments, while an early-stopping criterion was employed to automatically terminate training once the model was considered converged.

For the purpose of ensuring reliable and comparable results, it is worth noting that all models investigated in this study—namely, the source data-driven neural network, DE-PINN, OR-PINN, XN-PINN and TL-PINN—share the exact same architecture and training configurations (e.g. loss criterion, optimizer, and number of epochs). The only difference is that TL-PINN employed a

smaller learning rate to enable more conservative fine-tuning of the transferred parameters. The models were implemented on the PyTorch platform (Paszke et al., 2017). Every PINN training was completed on a single NVIDIA GeForce RTX 4090 Laptop GPU.

The accuracy of the solutions is evaluated using the root mean square error (R, Eq. 8) as the performance metrics. These metrics compare the outputs of each model against the benchmark solution (i.e., the solution obtained from the FV model at a fine resolution).Specifically, when assessing the accuracy of the prediction, the performance metric is defined as:

$$R_\eta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{\eta}_i - \eta_i)^2} \tag{8}$$

where N is the discrete number of values used to evaluate the error, $\tilde{\eta}_i$ and η_i refer to the prediction and benchmark solution.

4 Results and discussion

4.1 Overview

To evaluate the effectiveness of the proposed TL-PINN framework, three numerical experiments were conducted using a real-world river reach Xiaoqing River described in Section 2. A 5-year return period flood event was selected as the source task. A purely data-driven neural network was first trained using the benchmark solutions of the source flood event, with 70% of the available data reserved for training. The optimized parameters obtained from the source model were subsequently transferred to initialize all TL-PINN models in the target tasks. All the trained PINN models will be compared in prediction accuracy and training behavior.

The first experiment investigated the performance of TL-PINN under different numbers of labeled-data conditions. A 10-year return period flood event was selected as the target task. The PINN models were subsequently trained under three labeled-data conditions, including labeled data from 5 labeled cross-sections, labeled data from 3 labeled cross-sections, and no labeled data. The prediction accuracy and training behaviors of DE-PINN, OR-PINN, XN-PINN and TL-PINN were compared to evaluate the performance of PINNs under different numbers of labeled-data conditions. The second numerical experiment examined the robustness of PINN performance under different flood magnitudes. Three target tasks with different flood magnitudes were considered, including 20-, 50-, and 100-year return period flood events. The third numerical experiment further evaluated the applicability of TL-PINN under complex unsteady boundary conditions using different shape of boundary conditions(a double-peak flood). And Table 2 denotes the Summary of the numerical experiment configurations, including source tasks, target tasks, and labeled-data conditions used for

PINNs training.

Table 2. Summary of the numerical experiment designs conducted in this study

Experiment	Source Task	Target Task	Labeled-data for training
1	5-year return period flood	10-year return period flood	Case 1: 5 labeled cross-sections (S3, S12, S15, S17, and S21) Case 2: 3 labeled cross-sections (S3, S15, and S21) Case 3: no labeled data
2		20-year return period flood	5 labeled cross-sections (S3, S12, S15, S17, and S21)
		50-year return period flood	5 labeled cross-sections (S3, S12, S15, S17, and S21)
		100-year return period flood	5 labeled cross-sections (S3, S12, S15, S17, and S21)
3		Double-peak flood	5 labeled cross-sections (S3, S12, S15, S17, and S21)

4.2 Numerical experiment 1: Performance under different labeled-data conditions

The computational domain consists of 26 surveyed cross-sections (Fig. 5). In the first numerical experiment, a 10-year return period flood event was selected as the target task. However, only a limited number of labeled data were incorporated during PINNs training. This setting is more consistent with practical hydraulic applications, where fine resolution numerical solutions can be obtained from FV simulations, whereas dense measurements of water level and discharge are generally unavailable. Consequently, three labeled-data conditions were considered.

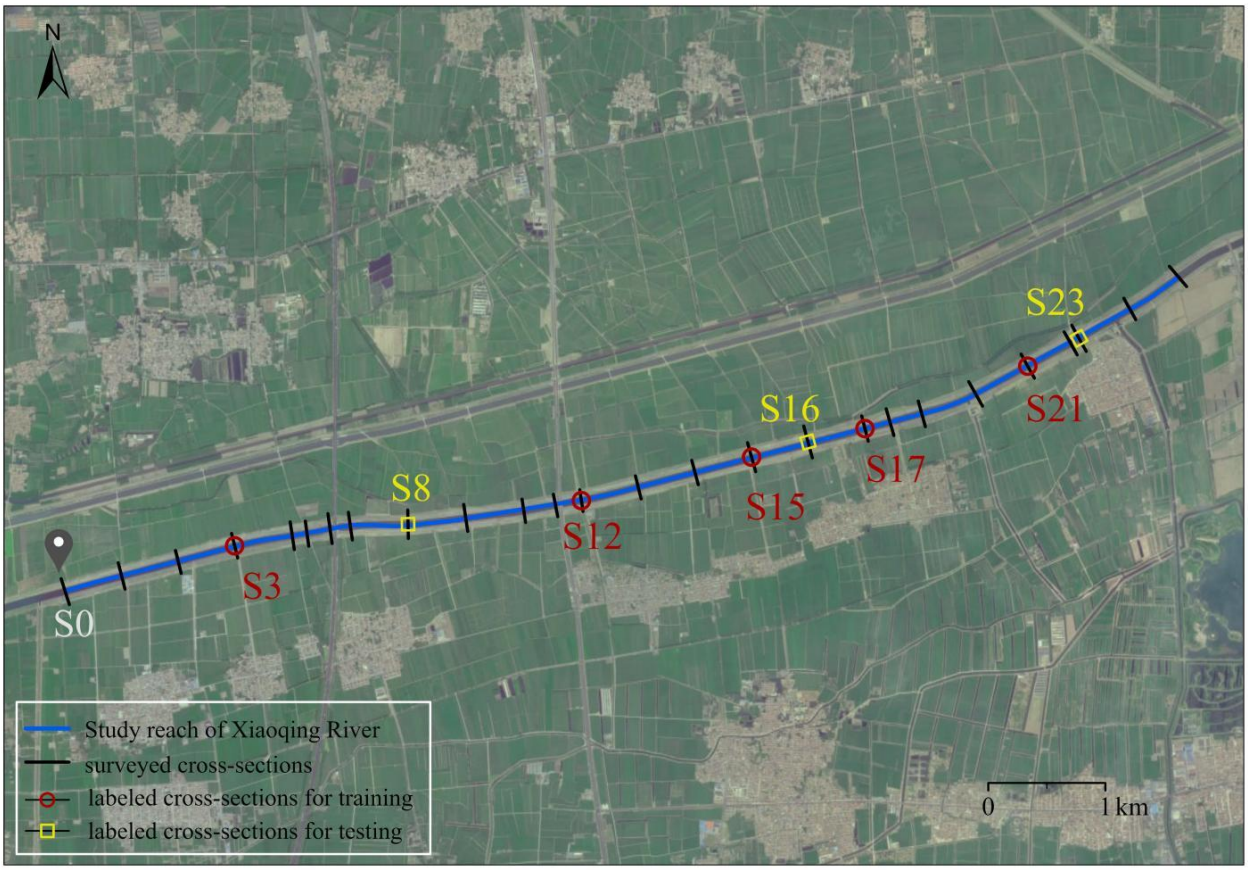


Fig. 5. Study reach and surveyed cross-sections of the Xiaoqing River

In case 1, labeled data form 5 labeled cross-sections (i.e., S3, S12, S15, S17, and S21) were selected for PINNs training. These cross-sections were distributed along the river reach, covering upstream, middle, and downstream regions. In Case 2, the number of labeled cross-sections was reduced to three (i.e., S3, S15, and S21), representing a scenario with more limited hydraulic observations. In Case 3, no labeled data was provided, and the models were trained only by minimizing the composite loss function. The remaining cross-sections were reserved for independent evaluation of model performance.

The longitudinal residuals of water level and discharge are used to evaluate the spatial reconstruction capability of the models along the river reach at selected times, thereby identifying reaches with relatively large prediction errors. Figs. 6 and 7 illustrate the longitudinal profiles of the benchmark (purple curves) and the corresponding residuals (ϵ_η and ϵ_Q) predicted by DE-PINN (blue lines), OR-PINN (orange lines), XN-PINN (green lines), and TL-PINN (pink lines) at t = 6, 12 and 18 h under the three labeled-data conditions. Overall, TL-PINN almost consistently provides smaller residuals than other PINN models. Specifically, the ϵ_η of TL-PINN remain within approximately ± 0.008 m, while ϵ_Q are generally controlled within ± 5 m³/s. Moreover, the residual magnitude of TL-PINN show no significant increase with decreasing labeled data, whereas

other PINN models exhibit larger deviations. The residuals become particularly significant in case 3 (no labeled data). For example, at t = 12 h, the ϵ_η of DE-PINN and OR-PINN exceeds 0.25 m at most locations along the river reach, while the ϵ_Q exceeds approximately 30 m³/s.

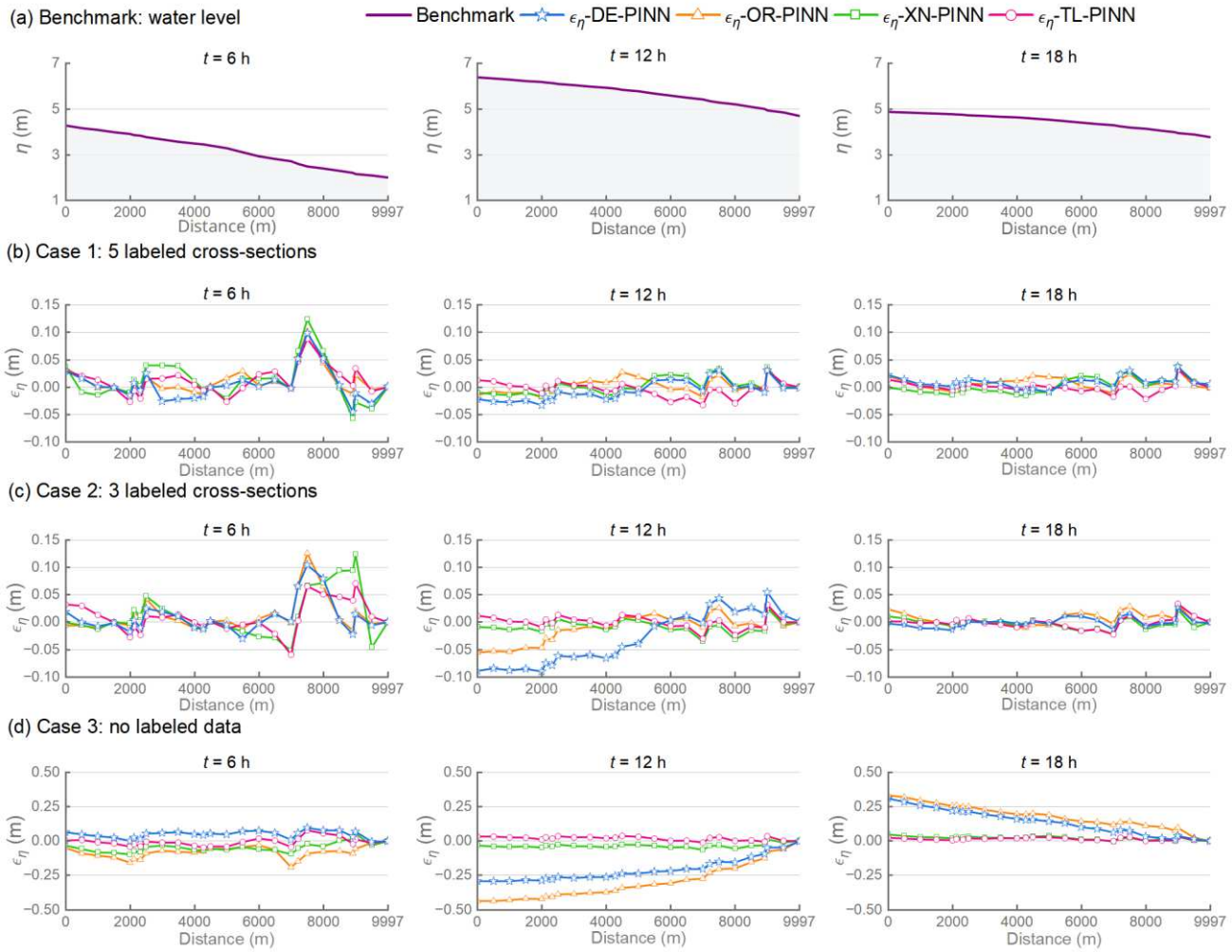


Fig. 6. Numerical experiment 1: Longitudinal profiles of benchmark water level η and water-level prediction residuals ϵ_η from DE-PINN, OR-PINN, XN-PINN, and TL-PINN at t = 6, 12 and 18 h under different labeled-data conditions: (a) benchmark water level; (b) Case 1, five labeled cross-sections; (c) Case 2, three labeled cross-sections; and (d) Case 3, no labeled data.

The temporal comparisons at representative cross-sections S8, S16 and S23 (Figs. 8 and 9) further reveal performance of the models throughout the flood process, including the rising, peak, and recession stages under different labeled-data conditions. The purple curves represent the benchmark and the corresponding residuals (ϵ_η and ϵ_Q) predicted by DE-PINN (blue lines), OR-PINN (orange lines), XN-PINN (green lines), and TL-PINN (pink lines) under the three labeled-data conditions. Overall, TL-PINN demonstrates excellent errors regardless of the labeled-data conditions. In contrast, the deviations of other PINN models are mainly associated with the underestimation of peak discharge, and their predictions deteriorate further during the rising limb (4 h $\leq$ t $\leq$ 8 h). In case 3 (no labeled data), another significant error emerges during the flood

recession stage. Specifically, after approximately 16 h, the discharge predictions of DE-PINN, OR-PINN and XN-PINN tend to gradually approach a unsteady state, failing to accurately reproduce the flood recession process.

The quantitative results in Table 3 further confirm these observations. Note that R_η and R_Q are calculated using all collocation points. In case 1, TL-PINN achieves R_η = 0.018 m and R_Q = 0.398 m³/s, whereas the residuals obtained from conventionally initialized PINNs are within the range of 0.022 m$\leq$ R_η $\leq$ 0.025 m and 3.465 m³/s $\leq$ R_Q $\leq$ 5.932 m³/s, respectively. When no labeled data are available for PINN training (i.e., Case 3), the advantage of TL-PINN becomes more significant. For example, TL-PINN maintains substantially lower errors of R_η = 0.035 m and R_Q = 1.014 m³/s, whereas the value of R_η within the range of 0.256 m$\leq$ R_η $\leq$ 0.405 m and 22.049 $\leq$ R_Q $\leq$ 42.063 m³/s. Overall, these results demonstrate that transfer learning provides more effective initial parameters for PINN optimization, thereby improving both prediction accuracy and training efficiency while substantially reducing the optimization difficulty of PINNs when labeled data are unavailable.

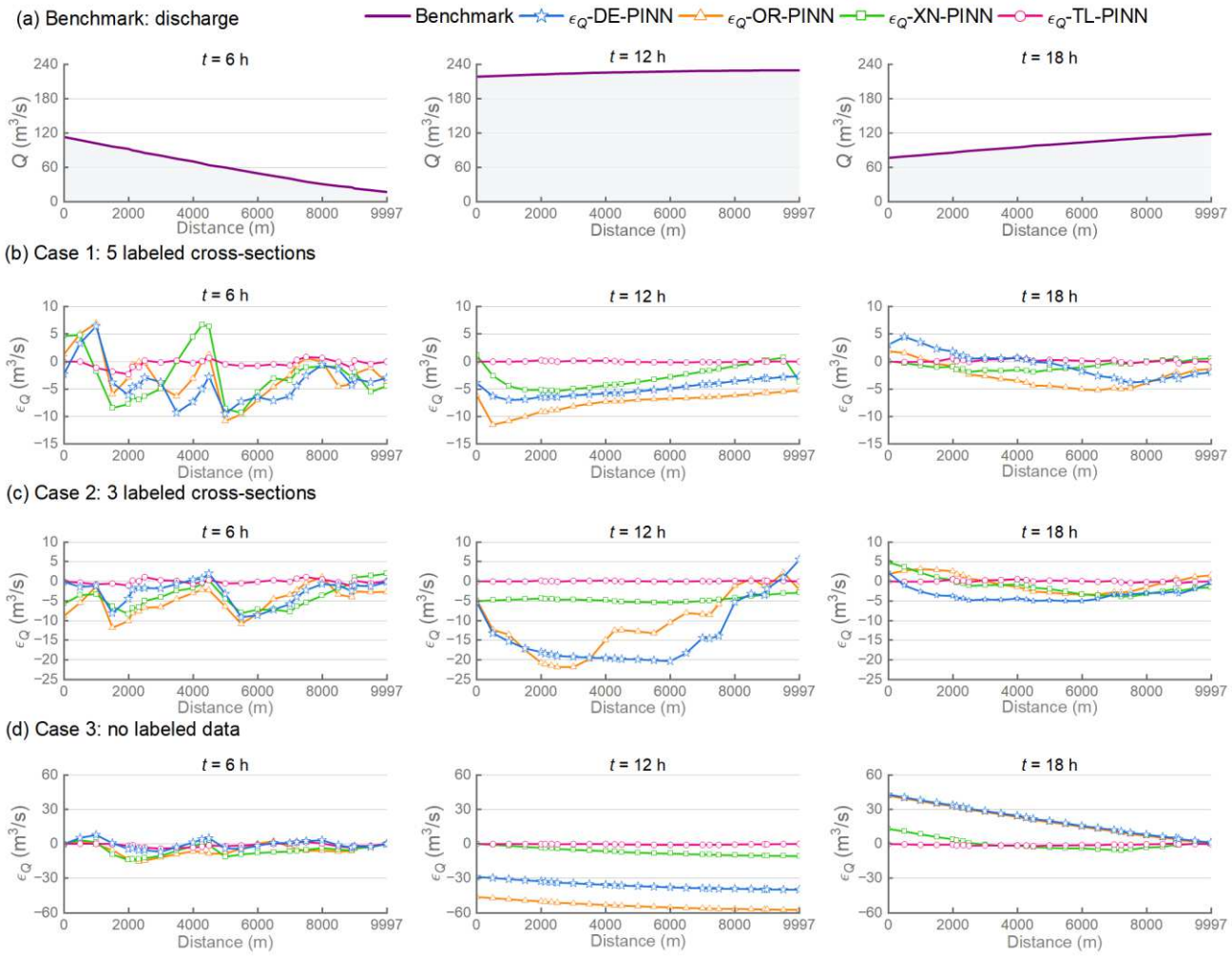


Fig. 7. Numerical experiment 1: Longitudinal profiles of benchmark discharge Q and discharge prediction residuals ϵ_Q from

DE-PINN, OR-PINN, XN-PINN, and TL-PINN at t = 6, 12 and 18 h under different labeled-data conditions: (a) benchmark discharge; (b) Case 1, five labeled cross-sections; (c) Case 2, three labeled cross-sections; and (d) Case 3, no labeled data.

Table 3. Numerical experiment 1: Training time ($\mathcal{T}$) and prediction accuracy of DE-PINN, OR-PINN, XN-PINN, and TL-PINN.

Model	Case 1: 5 labeled cross-sections			Case 2: 3 labeled cross-sections			Case 3: no labeled data		
	$\mathcal{T}$ (s)	R_η (m)	R_Q (m^3/s)	$\mathcal{T}$ (s)	R_η (m)	R_Q (m^3/s)	$\mathcal{T}$ (s)	R_η (m)	R_Q (m^3/s)
DE-PINN	323	0.024	5.048	308	0.029	8.172	268	0.392	38.736
OR-PINN	329	0.025	5.932	304	0.029	6.59	289	0.405	42.063
XN-PINN	292	0.022	3.465	277	0.027	4.923	275	0.256	22.049
TL-PINN	44	0.018	0.398	39	0.019	0.541	54	0.035	1.014

Fig. 10 presents the evolution of R_η and R_Q from four PINN models (i.e. all collocation points) during the training process under different labeled-data conditions. The blue, orange, green and pink curves represent DE-PINN, OR-PINN, XN-PINN, and TL-PINN, respectively. In contrast, TL-PINN consistently exhibits lower R_η and R_Q and faster error reduction throughout training. Across the three cases, TL-PINN achieves a substantial reduction in training time, requiring only about 50 s to complete training, whereas the other three models require approximately 300 s. It is worth noting that the optimization of DE-PINN, OR-PINN, and XN-PINN becomes particularly challenging in case 3. After approximately 50 s, both R_η and R_Q exhibit limited further reduction, suggesting that the PINN models using conventional initialization strategy may become trapped in an inefficient optimization region due to the lack of labeled data.

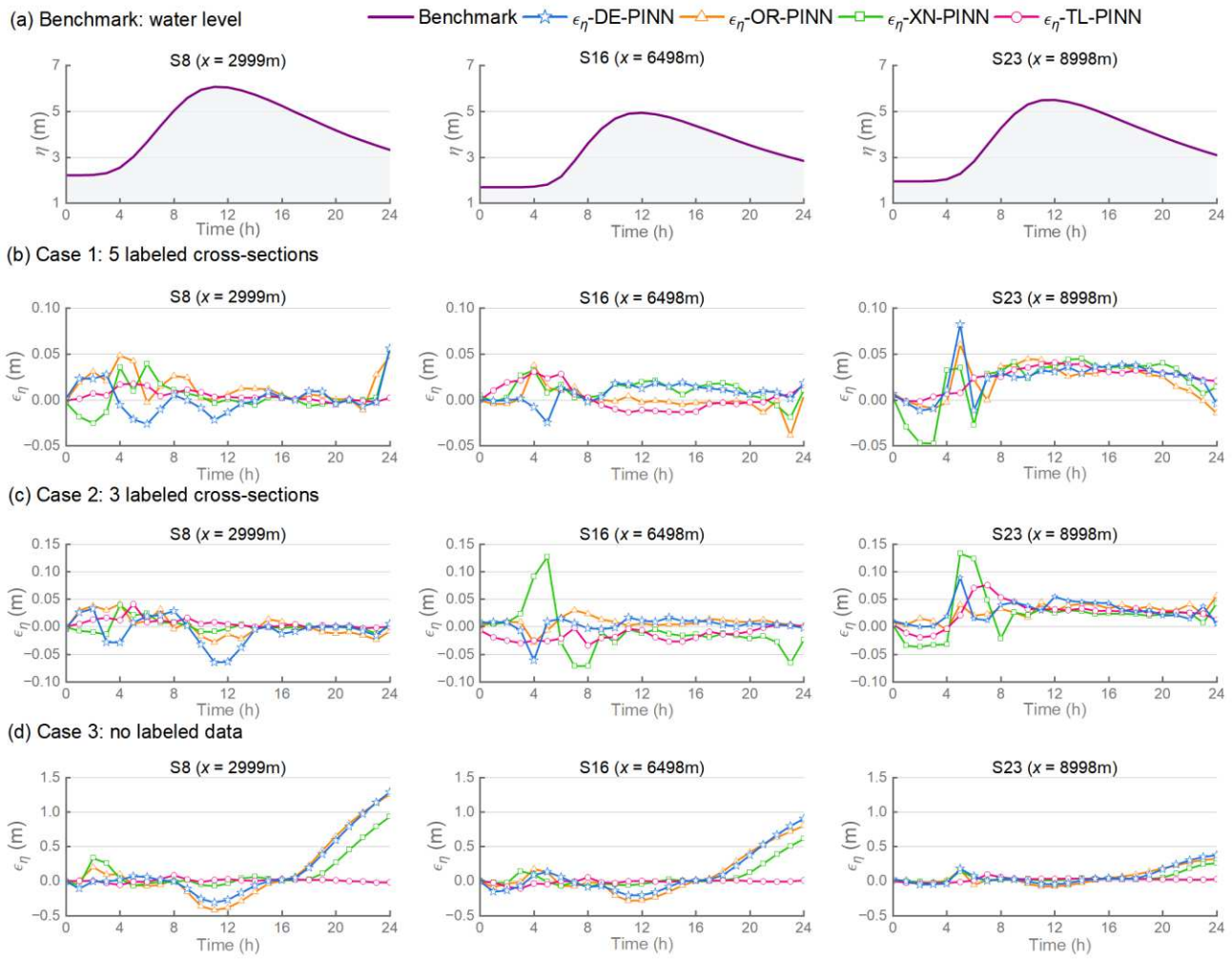


Fig. 8. Numerical experiment 1: Temporal variations of benchmark water level η and water-level prediction residuals ϵ_η from DE-PINN, OR-PINN, XN-PINN, and TL-PINN at representative cross-sections S8, S16, and S23 under different labeled-data conditions: (a) benchmark water level; (b) Case 1, five labeled cross-sections; (c) Case 2, three labeled cross-sections; and (d) Case 3, no labeled data.

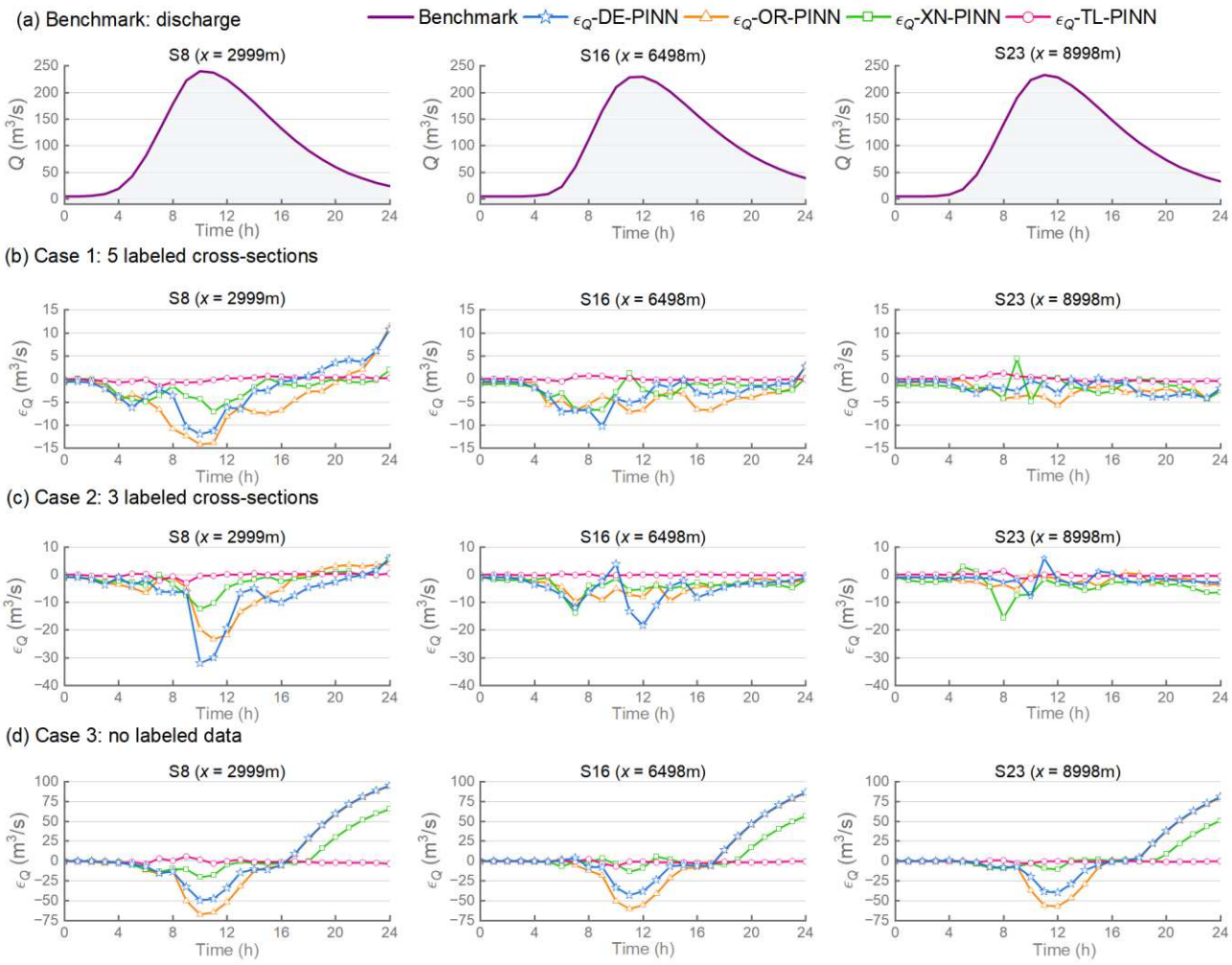


Fig. 9. Numerical experiment 1: Temporal variations of benchmark discharge Q and discharge prediction residuals ϵ_Q from DE-PINN, OR-PINN, XN-PINN, and TL-PINN at representative cross-sections S8, S16, and S23 under different labeled-data conditions: (a) benchmark water level; (b) Case 1, five labeled cross-sections; (c) Case 2, three labeled cross-sections; and (d) Case 3, no labeled data.

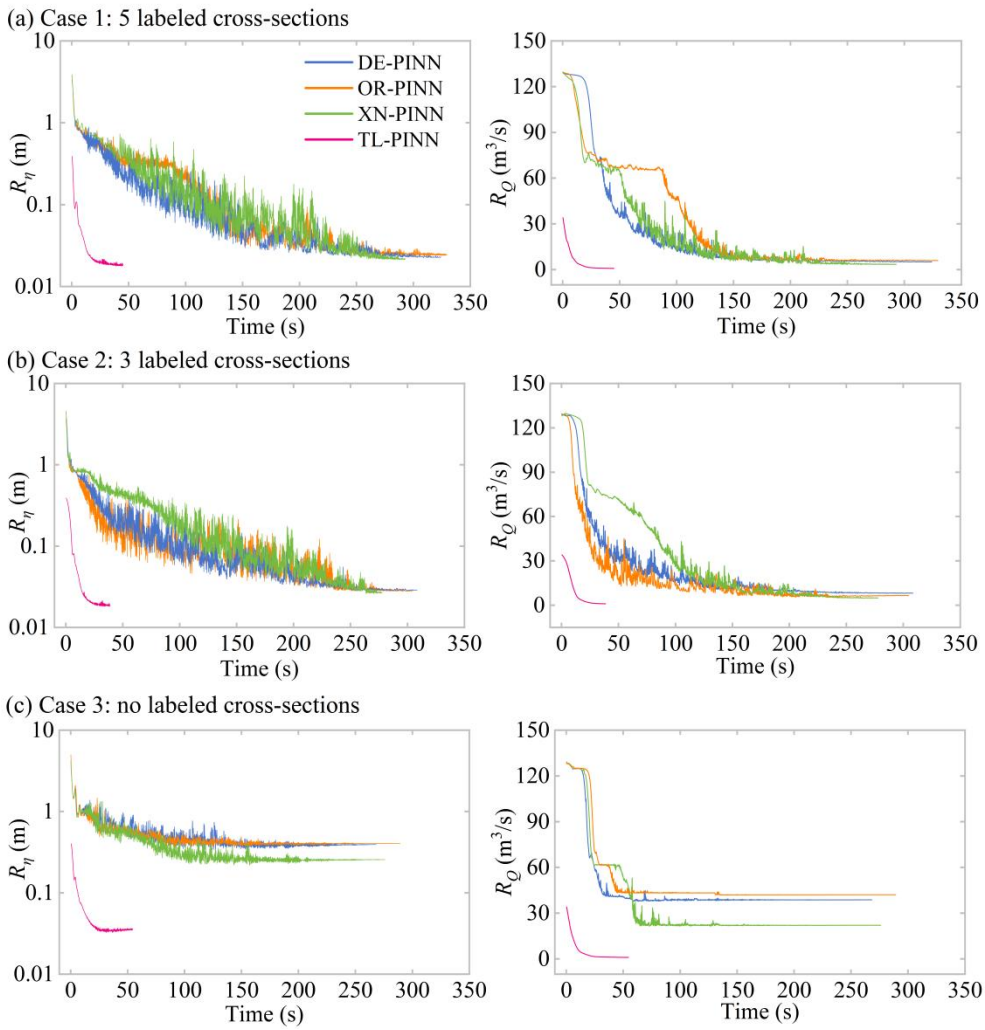


Fig. 10. Numerical experiment 1: Evolution of R_η and R_Q during training for DE-PINN, OR-PINN, XN-PINN, and TL-PINN under different labeled-data conditions: (a) Case 1, five labeled cross-sections; (b) Case 2, three labeled cross-sections; and (c) Case 3, no labeled cross-sections.

4.3 Numerical experiment 2: Performance under different flood magnitudes

The second numerical experiment was designed to investigate the adaptability of TL-PINN under increasingly severe flood conditions. This numerical experiment focuses on the influence of flood magnitude by increasing the return period of the target flood event. Three target tasks were considered for flood events with return periods of 20, 50, and 100 years, while the source task (5-yr return period flood event) remained unchanged. Given that DE-PINN, OR-PINN, and XN-PINN cannot accurately capture the flood process without labeled data, as shown in numerical experiment 1, labeled data from five labeled cross-sections (i.e., S3, S12, S15, S17, and S21) were incorporated into the PINN training process in this numerical experiment. Therefore, this experiment evaluates whether the transferred parameters obtained from the source task can provide effective initialization and maintain reliable prediction performance and stable training behavior under more severe flood

inflow conditions.

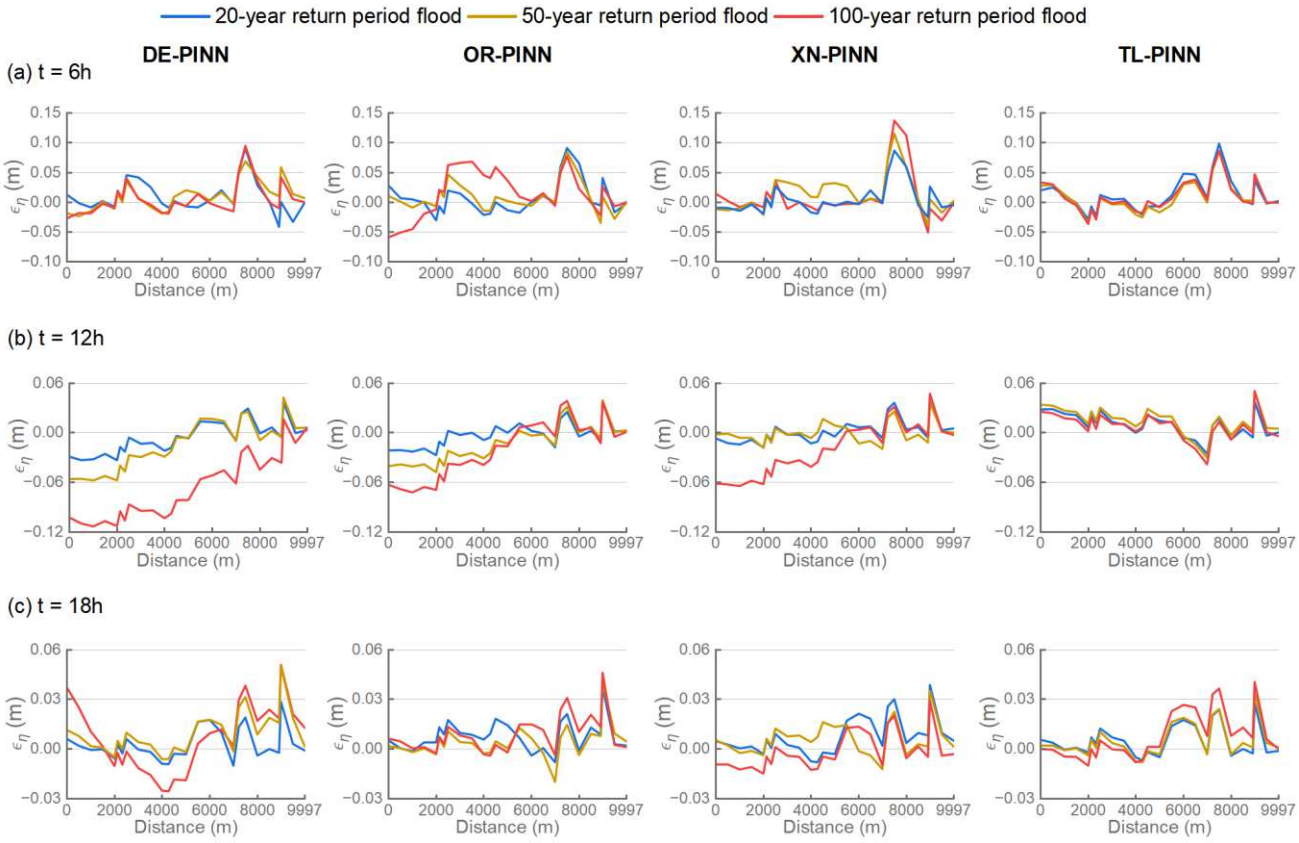


Fig. 11. Numerical experiment 2: Longitudinal distributions of water-level prediction residuals ϵ_η from DE-PINN, OR-PINN, XN-PINN, and TL-PINN under different flood magnitudes, represented by 20-, 50-, and 100-year return period floods, at (a) t = 6 h, (b) t = 12 h, and (c) t = 18 h.

Figs. 11 and 12 illustrate the longitudinal distributions of the prediction residuals (ϵ_η and ϵ_Q) predicted by DE-PINN, OR-PINN, XN-PINN and TL-PINN under flood scenarios of different magnitudes. The blue, orange, and red lines represent the prediction residuals (ϵ_η and ϵ_Q) of each model under 20-, 50-, and 100-year return period flood task, respectively. Overall, TL-PINN maintains lower errors under almost all flood scenarios, and . Notably, as the flood magnitude increases, DE-PINN, OR-PINN, and XN-PINN generally exhibit a marked upward trend in errors, whereas TL-PINN remains essentially stable in terms of prediction accuracy. For example, the value of ϵ_η obtained form conventionally initialized PINNs are within the range of -0.03 m < ϵ_η < 0.09 m (under 20-year return period flood event) and -0.12 m < ϵ_η < 0.14 m (under 100-year return period flood event), whereas the range for TL-PINN remains within -0.03 m to 0.1 m. In terms of discharge, the prediction residuals of TL-PINN remain close to zero, substantially outperforming the other models.

Figs. 13 and 14 further compare the temporal evolution of the prediction residuals (ϵ_η and ϵ_Q) provided by all PINN models at representative cross-sections under 20- (blue lines), 50- (orange

lines), and 100-year return period flood task (red lines). Similar to Experiment 1, all models exhibit larger water level residuals ϵ_η during the rising limb (2 h < t < 8 h) and larger discharge errors during the peak stage (2 h < t < 8 h). As the flood magnitude increases, DE-PINN, OR-PINN, and XN-PINN exhibit varying degrees of error growth, with DE-PINN being the most pronounced under the 100-year return period flood scenario, whereas the error performance of TL-PINN is almost unaffected by flood magnitude.

The quantitative findings presented in Table 4 further corroborate the above observations. Note that R_η and R_Q are over all collocation points.Under larger flood magnitude (i.e. 100-year return period flood), the values of R_η and R_Q obtained from conventionally initialized PINNs increase significantly, particularly for discharge prediction. The changes in R_η and R_Q reach 0.025 m and 7.239 m^3/s, respectively. The R_η remains approximately unchanged, and R_Q only slightly increases from 0.594 m^3/s to 0.755 m^3/s.

Table 4. Numerical experiment 2: Training time ($\mathcal{T}$) and prediction accuracy of DE-PINN, OR-PINN, XN-PINN, and TL-PINN.

Model	20-year return period flood			50-year return period flood			100-year return period flood		
	$\mathcal{T}$ (s)	R_η (m)	R_Q (m^3/s)	$\mathcal{T}$ (s)	R_η (m)	R_Q (m^3/s)	$\mathcal{T}$ (s)	R_η (m)	R_Q (m^3/s)
DE-PINN	304	0.024	6.374	307	0.026	9.007	312	0.049	13.613
OR-PINN	297	0.028	7.624	321	0.026	9.833	304	0.034	14.770
XN-PINN	283	0.023	5.169	284	0.024	7.194	295	0.028	9.398
TL-PINN	55	0.019	0.594	68	0.020	0.628	63	0.020	0.755

Fig. 15 presents the evolution of R_η and R_Q from four PINN models (i.e. all collocation points) during the training process under different flood magnitudes. The blue, orange, green and pink lines represent DE-PINN, OR-PINN, XN-PINN, and TL-PINN, respectively. Similar to Experiment 1, TL-PINN consistently shows lower R_η and R_Q and faster error reduction throughout training. In addition, the training time of TL-PINN remains substantially shorter than that of the other three PINN models.

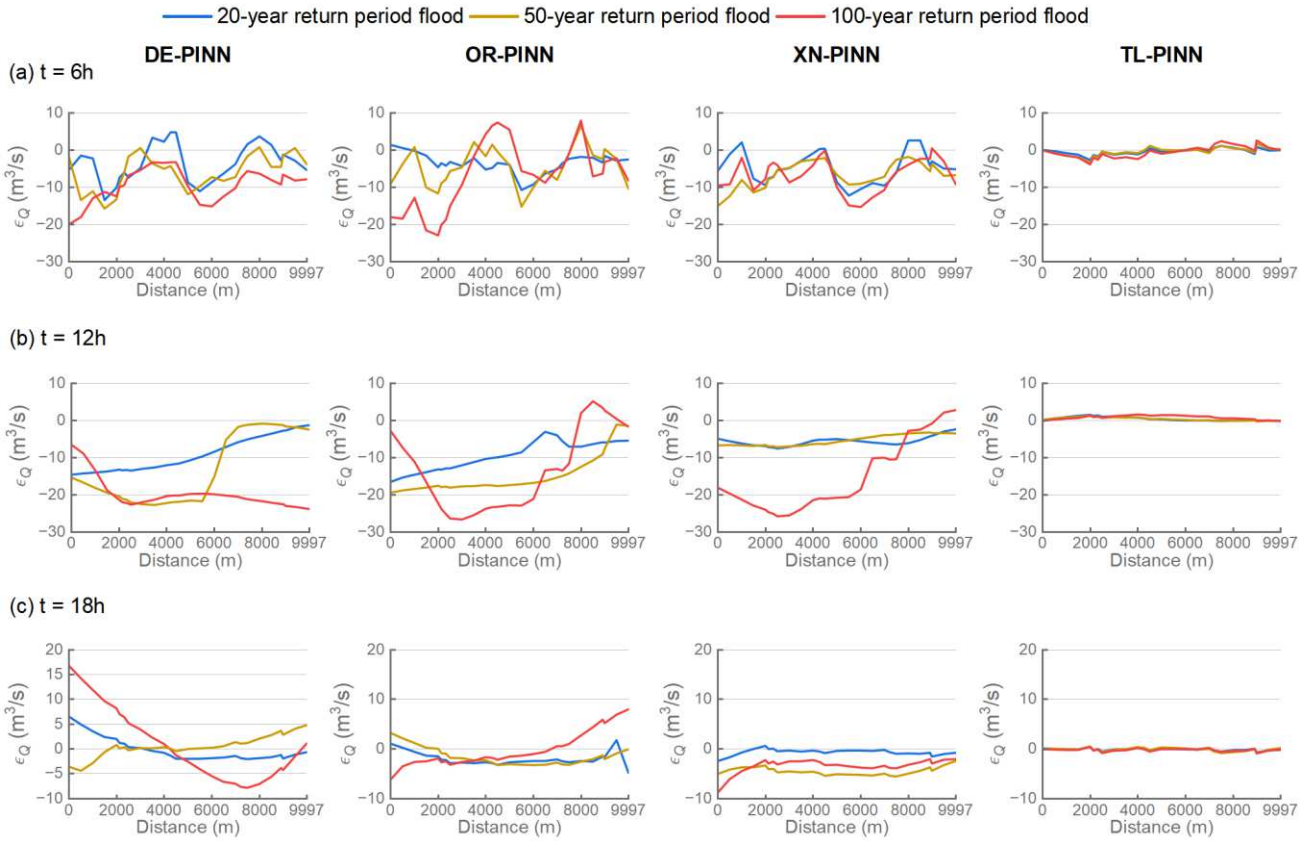


Fig. 12.. Numerical experiment 2: Longitudinal distributions of discharge prediction residuals ϵ_Q from DE-PINN, OR-PINN, XN-PINN, and TL-PINN under different flood magnitudes, represented by 20-, 50-, and 100-year return period floods, at (a) t = 6 h, (b) t = 12 h, and (c) t = 18 h.

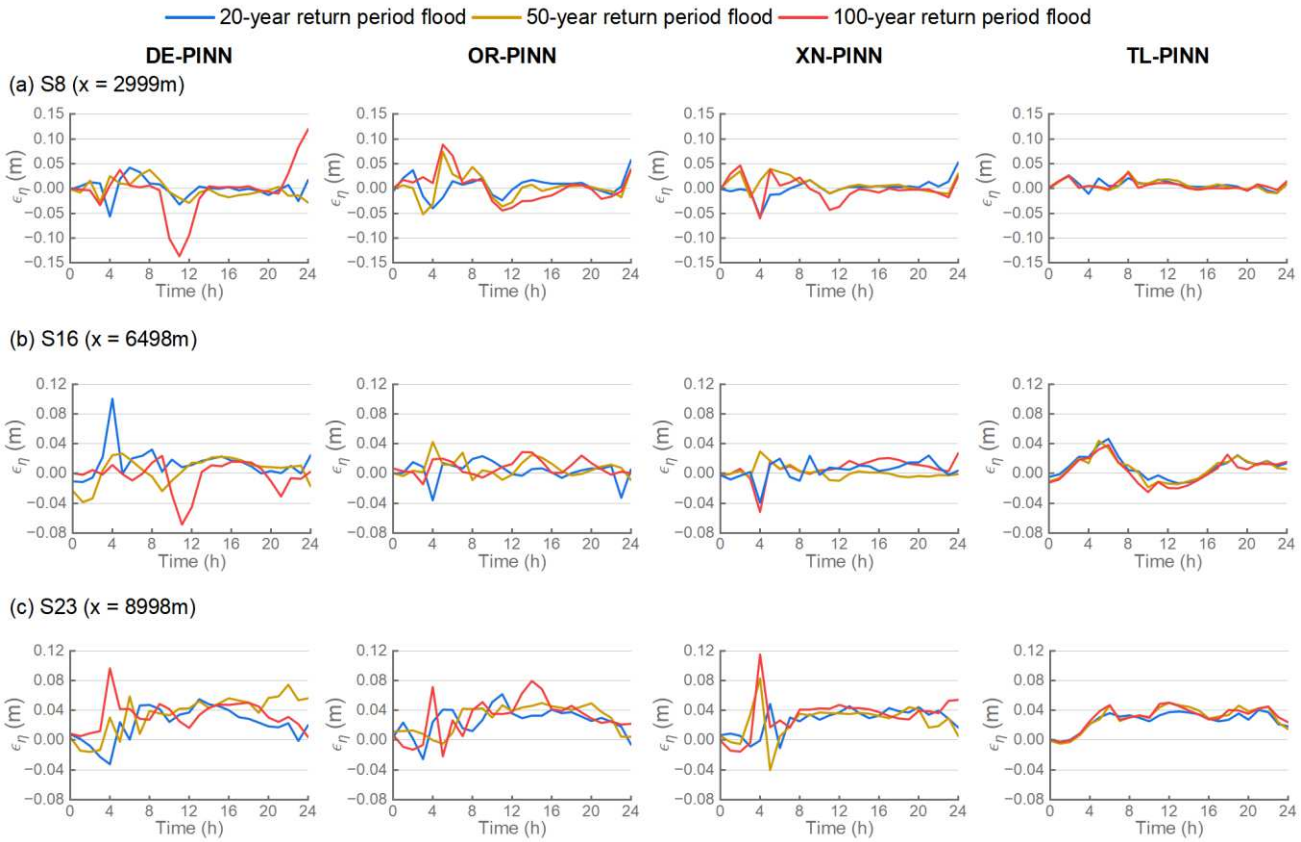


Fig. 13. Numerical experiment 2: Temporal variations of water-level prediction residuals ϵ_η from DE-PINN, OR-PINN, XN-PINN, and TL-PINN under different flood magnitudes, represented by 20-, 50-, and 100-year return period floods, at representative cross-sections (a) S8, (b) S16, and (c) S23.

Overall, increasing flood magnitude introduces additional challenges for PINNs optimization. While DE-PINN, OR-PINN and XN-PINN become increasingly sensitive to these changes and suffers from larger prediction errors, especially in discharge estimation, TL-PINN maintains stable accuracy and training behavior across different return-period flood events. These results further demonstrate that transfer learning could effectively improve the adaptability of PINNs to the flood events with more severe flood inflows conditions.

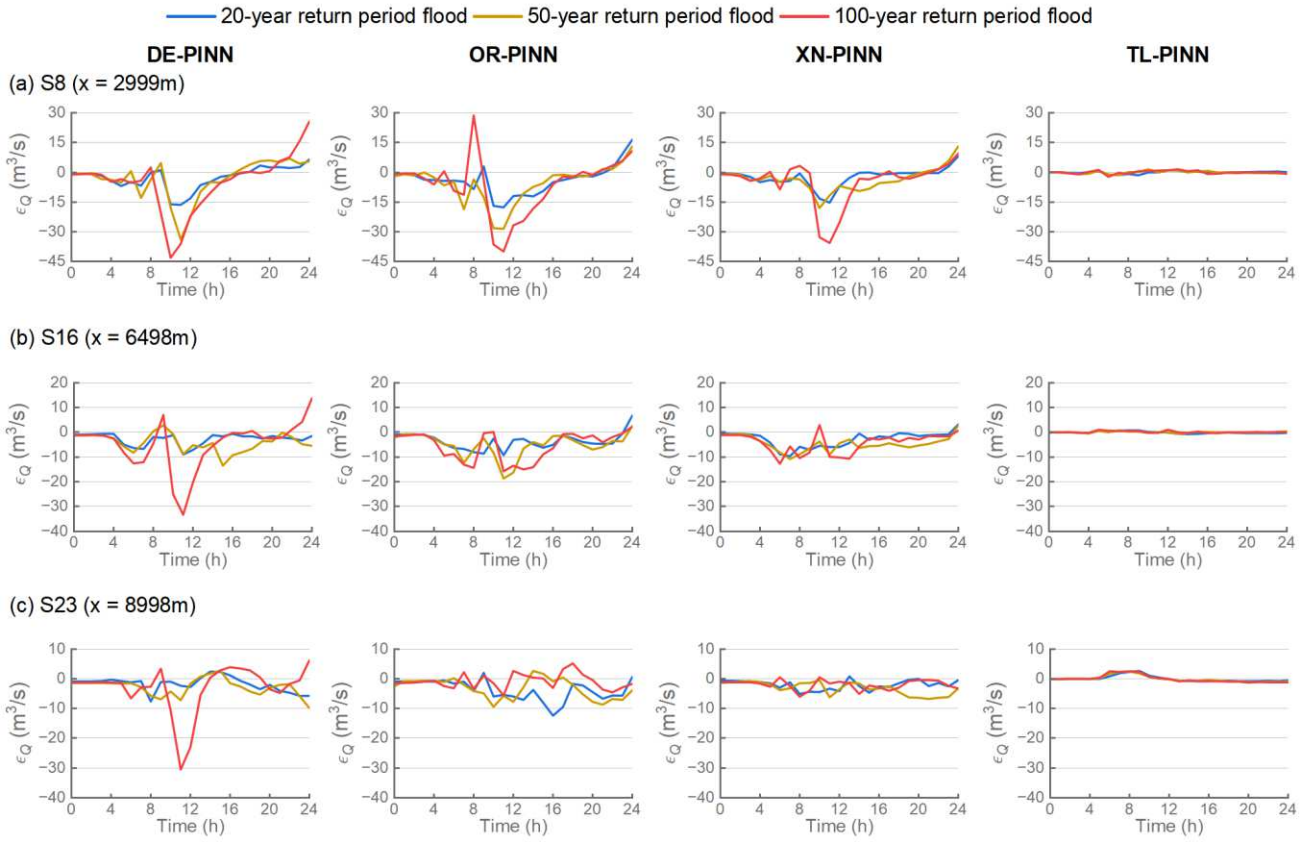


Fig. 14. Numerical experiment 2: Temporal variations of discharge prediction residuals ϵ_Q from DE-PINN, OR-PINN, XN-PINN, and TL-PINN under different flood magnitudes, represented by 20-, 50-, and 100-year return period floods, at representative cross-sections (a) S8, (b) S16, and (c) S23.

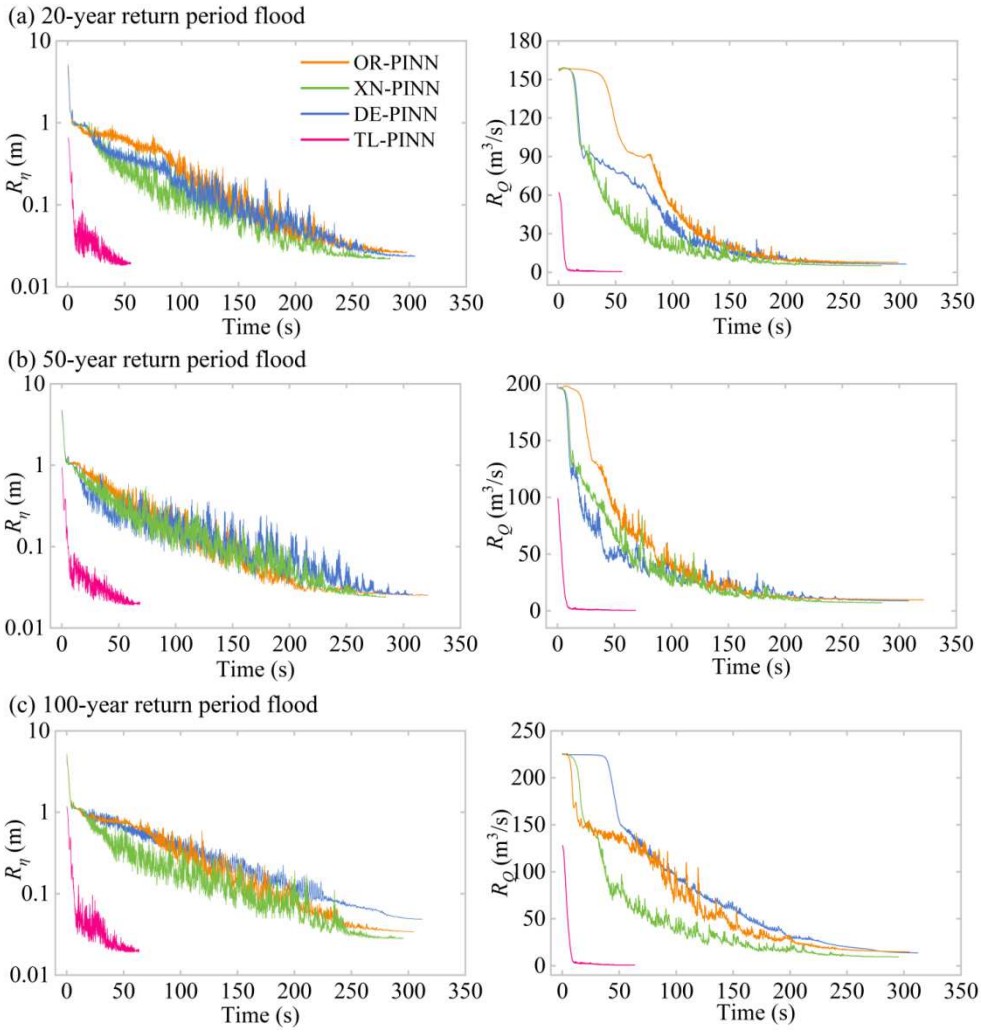


Fig. 15. Numerical experiment 2: Evolution of R_η and R_Q during training for DE-PINN, OR-PINN, XN-PINN, and TL-PINN under different flood magnitudes, represented by (a) 20-year, (b) 50-year, and (c) 100-year return period floods.

4.4 Numerical experiment 3: Performance under a double-peak flood event

To further evaluate the capability of transfer learning under more complex boundary conditions, a double-peak flood event was selected as the target task. Compared with the source task (5-yr return period flood event), the target task differs substantially not only in flood magnitude but also in hydrograph shape, providing a more challenging scenario for evaluating the generalization capability of TL-PINN. Following the same approach as in Numerical Experiments 1 and 2, labeled data from five cross-sections (i.e., S3, S12, S15, S17, and S21) were used for PINNs training in this numerical experiment.

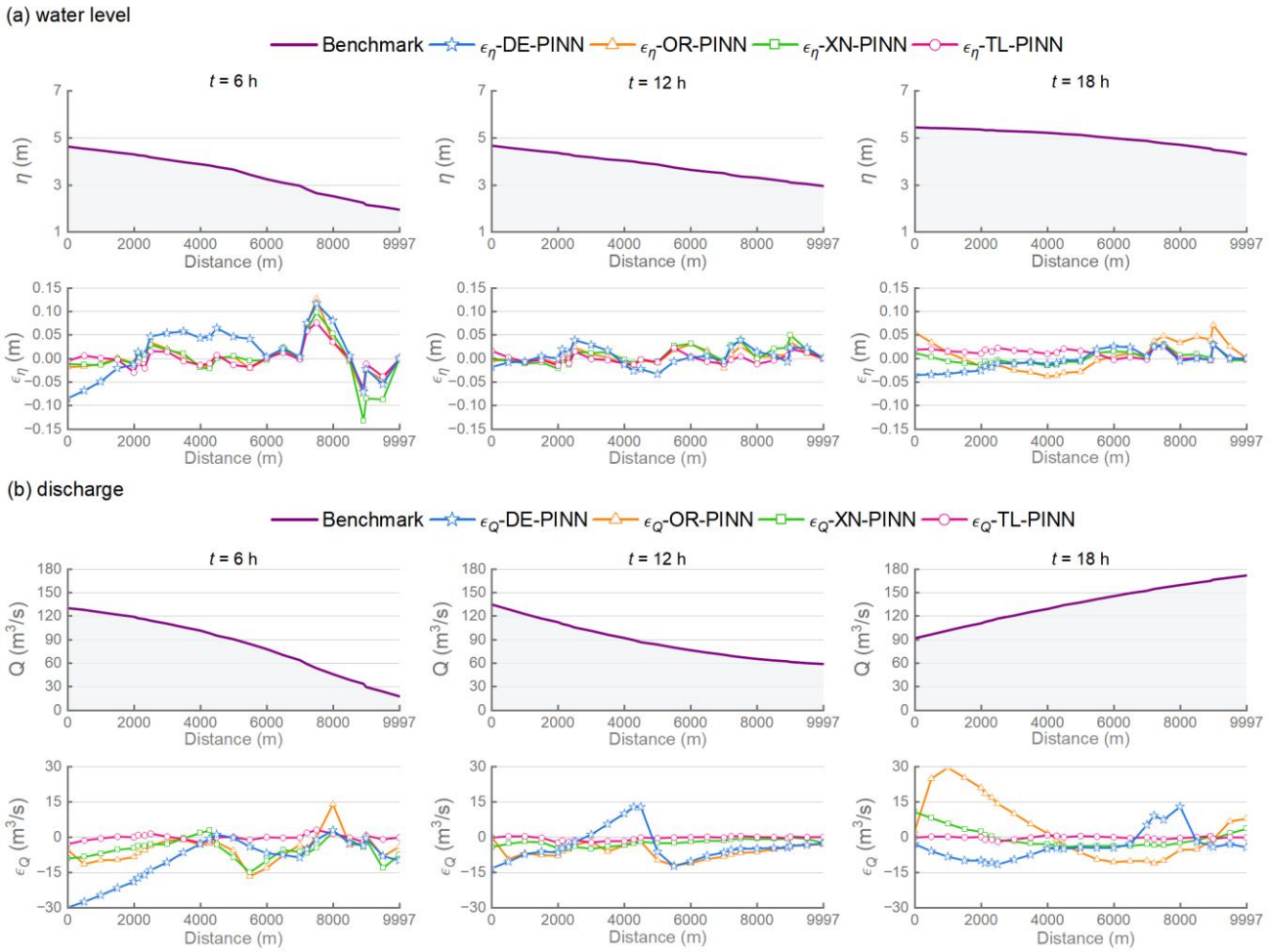


Fig. 16. Numerical experiment 3: Longitudinal profiles of benchmark water level η and discharge Q, together with the corresponding prediction residuals ϵ_η and ϵ_Q from DE-PINN, OR-PINN, XN-PINN, and TL-PINN at t = 6, 12, and 18 h under the double-peak flood event: (a) water level and (b) discharge.

Fig. 16 presents the longitudinal distributions of benchmark water level η and discharge Q (purple curves), along with the prediction residuals (ϵ_η and ϵ_Q) obtained form DE-PINN (blue lines), OR-PINN (orange lines), XN-PINN (green lines), and TL-PINN (pink lines). Overall, TL-PINN almost consistently provides lower prediction errors than other PINN models. Remarkably, the ϵ_Q values of DE-PINN and OR-PINN reach approximately -30 m³/s and 30 m³/s, respectively, while that of XN-PINN reaches about -15 m³/s. In contrast, TL-PINN remains within approximately ±3 m³/s throughout. For water level residuals ϵ_η.

The temporal evolution of benchmark (η and Q) at representative cross-sections (i.e. S8, S16 and S23) is shown in Fig. 17. The blue, orange, and green lines represent the prediction residuals (ϵ_η and ϵ_Q) obtained form the three conventionally initialized PINNs, respectively, and the pink lines denote TL-PINN. For water level, TL-PINN generally maintains the prediction residuals within a relatively narrow range of approximately ± 0.05 m, whereas the other three PINN models exhibit substantially larger deviations at t = 6, with errors approaching 0.2 m. For discharge, the conventionally initialized PINNs show a stronger tendency to underestimate the two flood peaks. At

cross-section S8 (t = 15 h), the discharge errors of XN-PINN, OR-PINN, and DE-PINN reach approximately 25, 50, and 70 m³/s, respectively. In contrast, TL-PINN successfully reproduces the two rising stages, peak values, and recession process of the double-peak flood hydrograph. Table 5 further confirms the above observations. Note that R_η and R_Q are calculated using all collocation points. TL-PINN still achieves lower R_η and R_Q, and in all experiments, This is particularly evident for R_Q, where the other three PINN models yield above 6.488 m³/s. It is worth noting that TL-PINN consistently maintains R_η within 0.02 m when labeled data are available for training.

The evolution of R_η and R_Q (i.e. all collocation points) from DE-PINN (blue line), OR-PINN (orange line), XN-PINN (green line), and TL-PINN (pink line) during the training process is shown in Fig. 18. As in numerical experiment 1 and 2, TL-PINN consistently achieves lower R_η and R_Q and faster error reduction throughout training. In addition, TL-PINN requires substantially less training time than the other three models, further demonstrating its higher training efficiency.

Table 5. Numerical experiment 3: Training time ($\mathcal{T}$) and prediction accuracy of DE-PINN, OR-PINN, XN-PINN, and TL-PINN.

Model	$\mathcal{T}$ (s)	R_η (m)	R_Q (m³/s)
DE-PINN	305	0.047	15.008
OR-PINN	304	0.038	13.058
XN-PINN	285	0.025	6.488
TL-PINN	65	0.019	0.651

Overall, Numerical Experiment 3 demonstrates that the proposed TL-PINN retains its advantages in prediction accuracy and training efficiency under a substantially different inflow-hydrograph shape, indicating that the transferred hydraulic information remains useful beyond target events that differ only in flood magnitude.

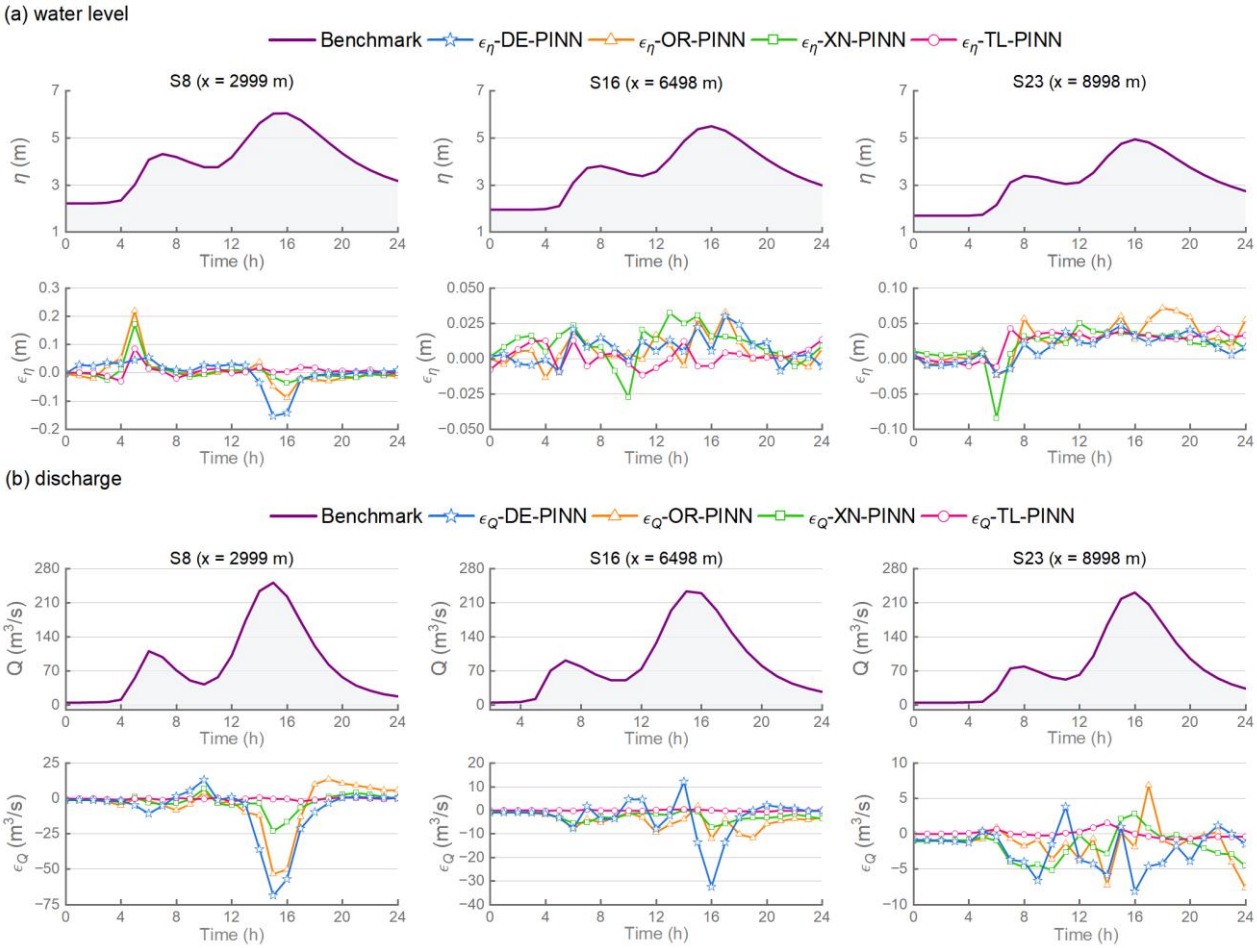


Fig. 17. Numerical experiment 3: Benchmark water level η and discharge Q, together with the corresponding prediction residuals ϵ_η and ϵ_Q from DE-PINN, OR-PINN, XN-PINN, and TL-PINN at cross-sections S8, S16, and S23 under the double-peak flood event: (a) water level and (b) discharge.

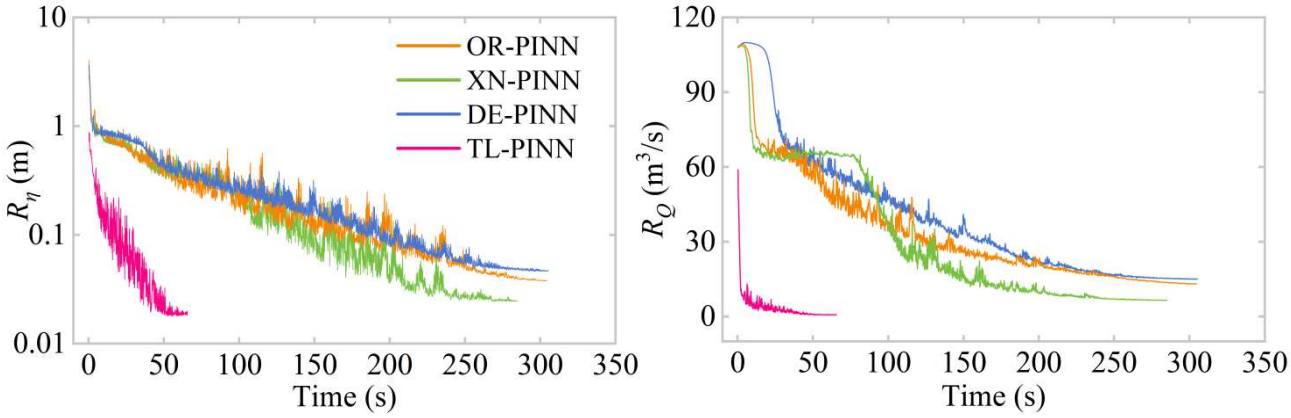


Fig. 18. Numerical experiment 3: Evolution of R_η and R_Q during training for DE-PINN, OR-PINN, XN-PINN, and TL-PINN under the double-peak flood event.

5 Concluding remarks

Physics-informed neural networks (PINNs) provide a promising framework for hydraulic

modeling by embedding governing physical laws into neural-network training. However, their application to real-world river-flood problems remains constrained by the difficulty and computational cost associated with nonlinear and non-convex optimization. In conventional PINN training, each new hydraulic scenario is generally initialized without task-specific information learned from previously solved events, which may lead to repeated optimization from an uninformed parameter state. To address this limitation, this study developed a transfer learning-enhanced physics-informed neural network (TL-PINN) framework for one-dimensional river flood simulation under varying flood events. A data-driven neural network was first trained using a source flood event, and the optimized source-model parameters were subsequently transferred to initialize the target PINNs, followed by physics-informed fine-tuning based on a composite loss function.

The proposed framework was evaluated on a real-world reach of the Xiaoqing River with complex channel geometry and unsteady flood processes. Three numerical experiments were designed to examine TL-PINN under progressively different target-task conditions, including reduced labeled-data availability, increasing flood magnitude, and a double-peak inflow hydrograph with a markedly different temporal pattern from the source event. TL-PINN was compared with PINNs initialized using PyTorch Default Initialization (DE-PINN), Orthogonal Initialization (OR-PINN), and Xavier Normal Initialization (XN-PINN). Across all tested conditions, TL-PINN consistently achieved higher accuracy (lower R_η and R_Q) with less training time, particularly for discharge prediction. This indicates that conventionally initialized PINNs still require further improvement in their ability to capture discharge. In addition, TL-PINN maintained stable performance not only without labeled data for training but also under more complex flood scenarios (i.e. flood magnitude and hydrograph shape).

The results further suggest that the effectiveness of transfer learning arises from the physical similarity among flood events occurring within the same river reach. Although the upstream inflow hydrographs vary, the source and target tasks share the same governing equations, channel geometry, roughness characteristics, computational domain, and fundamental hydraulic-response mechanisms. The transferred parameters therefore provide a task-informed initial state that already contains reusable hydraulic information. Instead of reconstructing the hydraulic response entirely from a newly generated initialization, TL-PINN can adapt the transferred parameter state to the target flood event through physics-informed optimization (i.e. composite loss function). This provides a plausible explanation for the consistently faster convergence and superior prediction accuracy observed for TL-PINN. Moreover, the parameters obtained from purely data-driven neural network training can be successfully transferred to the PINN model and remain effective across different target tasks. This indicates that hydraulic representations learned from data-driven pretraining can be retained and further refined under physical constraints. This combination of data-driven pretraining and physics-informed fine-tuning provides a practical pathway for improving PINN initialization.

From an engineering perspective, the proposed framework is particularly relevant to repeated

flood simulation applications. In practical river management, the same river reach is often evaluated under multiple design floods, forecast hydrographs, or alternative boundary-condition scenarios. Training a separate PINN from scratch for each case would repeatedly incur the cost of identifying a suitable parameter state. By reusing information learned from a previously solved flood event, TL-PINN reduces this repeated optimization burden and provides a more efficient strategy for a family of hydraulically related simulations. Its reduced sensitivity to labeled-data availability also suggests potential value in applications where dense hydraulic observations are difficult or costly to obtain.

Overall, the proposed TL-PINN framework provides a more efficient and reliable strategy for repeated one-dimensional river flood simulations. By reusing hydraulic information learned from a source flood event, transfer learning reduces the optimization difficulty of PINNs, improves training efficiency and prediction accuracy. The results suggest that TL-PINN has considerable potential for practical hydraulic applications involving multiple flood scenarios. Therefore, it might emerge as a compelling candidate for supplanting conventional PINN training strategies in addressing flood simulations in the real-world river reach governed by the one-dimensional SVEs under varying flood events.

Despite the promising performance of TL-PINN, several limitations remain. First, although transfer learning significantly improves the training efficiency of PINNs, the overall computational cost is still higher than that of traditional numerical methods, Second, this study focused on the river flood governed by the one-dimensional SVEs. Future research could extend the proposed transfer learning strategy to two-dimensional flow problems.

Funding

This study was supported by the National Key Research and Development Program of China (Grant No. 2024YFC3214802), the Special Fund for Scientific Achievements Transformation of IWHR (No. JZ121003A0052025), and the IWHR Research & Development Support Program (No. JZ110145B0022025).

Conflict of Interest

The authors declare that they have no conflict of interest

Author Contributions

CRediT: **Kang Zheng**: Conceptualization, Data curation, Methodology, Software, Investigation, Formal analysis, Visualization, Writing – original draft; **Xin Qi**: Conceptualization, Methodology,

Validation, Formal analysis, Writing – review and editing; **Dawei Zhang**: Conceptualization, Project administration, Supervision, Funding acquisition, Writing – review and editing; **Xinzheng Feng**: Validation, Resources, Writing – review and editing; **Weiqi Wang**: Data curation, Resources, Writing – review and editing.

Data availability

Data will be made available on request.

Ethics Approval and Consent to Participate

Not applicable.

Acknowledgments

Not applicable.

Reference

Alcrudo, F., & Garcia-Navarro, P. (1993). A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids*, *16*(6), 489–505.

Bale, D. S., Leveque, R. J., Mitran, S., & Rossmanith, J. A. (2003). A wave propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM Journal on Scientific Computing*, *24*(3), 955–978.

Bernard, P.-E., Remacle, J.-F., Comblen, R., Legat, V., & Hillewaert, K. (2009). High-order discontinuous Galerkin schemes on general 2D manifolds applied to the shallow water equations. *Journal of Computational Physics*, *228*(17), 6514–6535.

Bihlo, A., & Popovych, R. O. (2022). Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics*, *456*, 111024.

Bischof, R., & Kraus, M. A. (2025). Multi-objective loss balancing for physics-informed deep learning. *Computer Methods in Applied Mechanics and Engineering*, *439*, 117914.

Botta, N., Klein, R., Langenberg, S., & Lützenkirchen, S. (2004). Well balanced finite volume methods for nearly hydrostatic flows. *Journal of Computational Physics*, *196*(2), 539–565.

Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. (2021). Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, *37*(12), 1727–1738.

Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, *143*(6), 060801.

Casulli, V. (1990). Semi-implicit finite difference methods for the two-dimensional shallow water equations. *Journal of Computational Physics*, *86*(1), 56–74.

Chakraborty, S. (2021). Transfer learning based multi-fidelity physics informed deep neural network. *Journal of Computational Physics*, *426*, 109942.

Chaudhry, M. H. Chaudhry, 2008. Open-Channel Flow.

Cunge, J. A., Holly, F. M., & Verwey, A. (1980). *Practical aspects of computational river hydraulics*.

Dawson, C., Westerink, J. J., Feyen, J. C., & Pothina, D. (2006). Continuous, discontinuous and coupled discontinuous–continuous Galerkin finite element methods for the shallow water equations. *International Journal for Numerical Methods in Fluids*, *52*(1), 63–88.

Dazzi, S. (2024). Physics-informed neural networks for the augmented system of shallow water equations with topography. *Water Resources Research*, *60*(10), e2023WR036589.

de Sousa, C. A. (2016). An overview on weight initialization methods for feedforward neural networks. 2016 International Joint Conference on Neural Networks (IJCNN),

Eivazi, H., Tahani, M., Schlatter, P., & Vinuesa, R. (2022). Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. *Physics of Fluids*, *34*(7).

Feng, D., Tan, Z., & He, Q. (2023). Physics-informed neural networks of the Saint-Venant equations for downscaling a large-scale river model. *Water Resources Research*, *59*(2), e2022WR033168.

Gao, H., Sun, L., & Wang, J.-X. (2021). PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics*, *428*, 110079.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics,

Goswami, S., Anitescu, C., Chakraborty, S., & Rabczuk, T. (2020). Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, *106*, 102447.

Gu, L., Qin, S., Xu, L., & Chen, R. (2024). Physics-informed neural networks with domain decomposition for the incompressible Navier–Stokes equations. *Physics of Fluids*, *36*(2).

Guan, Y., Chattopadhyay, A., Subel, A., & Hassanzadeh, P. (2022). Stable a posteriori LES of 2D turbulence using convolutional neural networks: Backscattering analysis and generalization to higher Re via transfer learning. *Journal of Computational Physics*, *458*, 111090.

Hanert, E., Le Roux, D. Y., Legat, V., & Deleersnijder, E. (2005). An efficient Eulerian finite element method for the shallow water equations. *Ocean Modelling*, *10*(1-2), 115–136.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision,

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning,

Iskhakov, A. S., & Dinh, N. T. (2020). Physics-integrated machine learning: embedding a neural network in the Navier-Stokes equations. Part I. *arXiv preprint arXiv:2008.10509*.

Jagtap, A. D., Kawaguchi, K., & Karniadakis, G. E. (2020). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, *404*, 109136.

Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, *365*, 113028.

Jin, X., Cai, S., Li, H., & Karniadakis, G. E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, *426*, 109951.

Juez, C., Murillo, J., & García-Navarro, P. (2014). A 2D weakly-coupled and efficient numerical model for transient shallow flow and movable bed. *Advances in water resources*, *71*, 93–109.

Kassaï Koupaï, A., Mifsut Benet, J., Yin, Y., Vittaut, J.-N., & Gallinari, P. (2024). Boosting generalization in parametric pde neural solvers through adaptive conditioning. *Advances in neural information processing systems*, *37*, 70659–70692.

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. Proceedings of the IEEE conference on computer vision and pattern recognition,

Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do better imagenet models transfer better? Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., & Mahoney, M. (2021). Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, *34*, 26548–26560.

Kurganov, A., & Levy, D. (2002). Central-upwind schemes for the Saint-Venant system. *ESAIM: Modélisation mathématique et analyse numérique*, *36*(3), 397–425.

LeVeque, R. J., George, D. L., & Berger, M. J. (2011). Tsunami modelling with adaptively refined finite volume methods. *Acta Numerica*, *20*, 211–289.

Li, C., Han, Z., Li, Y., Li, M., Wang, W., Dou, J., Xu, L., & Chen, G. (2023). Physical information-fused deep learning model ensembled with a subregion-specific sampling method for predicting flood dynamics. *Journal of Hydrology*, *620*, 129465.

Li, R., Lee, E., & Luo, T. (2021). Physics-informed neural networks for solving multiscale mode-resolved phonon Boltzmann transport equation. *Materials Today Physics*, *19*, 100429.

Liang, Q. (2011). A structured but non-uniform Cartesian grid-based model for the shallow water equations. *International Journal for Numerical Methods in Fluids*, *66*(5), 537–554.

Liang, Q., & Marche, F. (2009). Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in water resources*, *32*(6), 873–884.

Liu, Y., Liu, W., Yan, X., Guo, S., & Zhang, C.-a. (2023). Adaptive transfer learning for PINN. *Journal of Computational Physics*, *490*, 112291.

Lynch, D. R., & Gray, W. G. (1979). A wave equation model for finite element tidal computations. *Computers & fluids*, *7*(3), 207–228.

Mao, Z., Jagtap, A. D., & Karniadakis, G. E. (2020). Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, *360*, 112789.

Mishkin, D., & Matas, J. (2015). All you need is a good init. *arXiv preprint arXiv:1511.06422*.

Molls, T., & Chaudhry, M. H. (1995). Depth-averaged open-channel flow model. *Journal of hydraulic engineering*, *121*(6), 453–465.

Narkhede, M. V., Bartakke, P. P., & Sutaone, M. S. (2022). A review on weight initialization strategies for neural networks. *Artificial intelligence review*, *55*(1), 291–322.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. International conference on machine learning,

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., & Antiga, L. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Qi, X., de Almeida, G. A., & Maldonado, S. (2024). Physics-informed neural networks for solving flow problems modeled by the 2D Shallow Water Equations without labeled data. *Journal of Hydrology*, *636*, 131263.

Qi, X., Zhang, D., Wang, F., Bi, W., & Lu, M. (2025). NDAWL-PINN: a new non-dimensionalization and multi-task learning approach for efficient training of physics-informed neural networks to solve the shallow water equations. *Engineering Applications of Computational Fluid Mechanics*, *19*(1), 2535015.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Sun, L., Gao, H., Pan, S., & Wang, J.-X. (2020). Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, *361*, 112732.

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F., Dutta, D., & Kim, S. (2017). Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental modelling & software*, *90*, 201–216.

Tian, Y., Ding, S., Huang, L., Su, G., & Chen, J. (2025). Physics-informed neural networks for solving the two-dimensional shallow water equations with terrain topography and rainfall source terms. *Water Resources Research*, *61*(9), e2025WR040052.

Toro, E. F., & Garcia-Navarro, P. (2007). Godunov-type methods for free-surface shallow flows: A review. *Journal of Hydraulic Research*, *45*(6), 736–751.

Wang, S., Yu, X., & Perdikaris, P. (2020). When and why PINNs fail to train: A neural tangent kernel perspective. *arXiv preprint arXiv:2007.14527*.

Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, *449*, 110768.

Wenyan, H., Sifang, Z., Zhenghe, X., Weiyan, P., Xin, C., & Yuqi, L. (2025). Hydrodynamic Simulation and Flood Risk Analysis of

Tide-sensitive Section of Xiaoqing River Estuary

Based on MIKE 21 Model. *Journal of University of Jinan (Science and Technology)*, *39*(3), 346–354. https://doi.org/10.13349/j.cnki.jdxbn.20250310.001

Wilson, M., Bates, P., Alsdorf, D., Forsberg, B., Horritt, M., Melack, J., Frappart, F., & Famiglietti, J. (2007). Modeling large-scale inundation of Amazonian seasonally flooded wetlands. *Geophysical Research Letters*, *34*(15).

Xu, C., Cao, B. T., Yuan, Y., & Meschke, G. (2023). Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios. *Computer Methods in Applied Mechanics and Engineering*, *405*, 115852.

Yang, X., Zafar, S., Wang, J.-X., & Xiao, H. (2019). Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, *4*(3), 034602.

Yoon, T. H., & Kang, S.-K. (2004). Finite volume model for two-dimensional shallow water flows on unstructured grids. *Journal of hydraulic engineering*, *130*(7), 678–688.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, *27*.

Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., & Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, *394*, 56–81.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.