

Article

Comparison of a Parametric Physics-Informed Neural Network and a Tensorial Reduced-Order Model for the Shallow-Water Dam-Break Problem

Anton Myshak , Md Rezwan Bin Mizan  and Ilya Timofeyev * 

Department of Mathematics, University of Houston, Houston, TX 77004, USA; amyshak@uh.edu (A.M.); mbinmizan@uh.edu (M.R.B.M.)

* Correspondence: itimofey@cougarnet.uh.edu

Abstract

We develop two parametric data-driven reduced models: a physics-informed neural network (PINN) and a non-intrusive tensorial reduced-order model (TROM), and apply both approaches to the parametrized one-dimensional shallow-water dam-break problem. Both reduced models do not require time integration and learn a direct solution map from space, time, and dam-break parameters to the physical state. We present a detailed comparison for out-of-sample and extrapolated parameter values. In addition, we demonstrate that it is essential to introduce shock-aware collocation to improve the robustness of the PINN model.

Keywords: parametric PINN; parametric TROM; Reduced-Order Model; shallow-water equations; dam-break problem

MSC: 65M22; 65M60; 68T07; 35L65

1. Introduction

Hyperbolic balance laws are ubiquitous in many areas of applied mathematics, including fluid flows. One of the main characteristics of hyperbolic problems is the appearance of sharp transients and discontinuities, which are intrinsic to the dynamics. One of the canonical benchmarks is the one-dimensional (1D) shallow-water dam-break problem (e.g., [1–3]). Despite its conceptual simplicity, the solution typically develops nonlinear wave interactions, moving fronts, and (depending on wet/dry configurations) sharp transitions (shocks) that are challenging numerically for both classical discretizations and surrogate modeling. Reliable and efficient prediction of such regimes is central to many practical problems, including uncertainty quantification, design and control, and real-time forecasting, motivating the development of reduced models that remain accurate in the presence of transport and shocks [4–9].

Projection-based reduced-order models (ROMs) (see survey [10] and references therein) constitute a well-established approach to accelerating numerical simulations by projecting the state onto a low-dimensional approximation space constructed from high-fidelity snapshots. However, it is known that for transport-dominated problems that contain shocks, the solution manifold often exhibits slow decay of the Kolmogorov n -width [11,12] and its approximations [13–17] computed by greedy POD [18], limiting the efficiency of linear-subspace compression and leading to a loss of fidelity near discontinuities. Consequently, substantial research has focused on ROM variants that mitigate these issues, for instance,



Academic Editors: Juha Videman and Jaime Klapp

Received: 29 July 2026

Revised: 22 September 2026

Accepted: 2 October 2026

Published: 10 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

through localization, nonlinear manifolds, or feature alignment (e.g., [19–23]). Recently, a parametric tensor Reduced-Order Model (TROM) was proposed to address some of these issues [24–26]. This approach combines a low-rank tensor representation of the parameter-to-solution map with localized reduced bases to improve accuracy across regimes. The TROM is particularly suitable for the development of reduced models for parametric problems. In [27] and [28], this approach was successfully applied to the dam-break problem for the shallow-water equations for Newtonian and visco-plastic fluids, respectively. In this paper, we consider the non-intrusive TROM, where solutions for new parameter values are computed directly from the low-rank compressed representation without time integration of the reduced equations. Details are provided in Section 3.

In parallel, physics-informed neural networks (PINNs) [29] have emerged as a promising paradigm for constructing differentiable surrogate models by embedding governing equations directly into the learning objective. In their classical form, PINNs optimize a neural network representing the solution fields using a composite loss that penalizes violations of the partial differential equations (PDEs) as well as mismatches to initial and boundary data [29]. This hybridization of data and physics has shown a strong potential for learning PDE solution operators from relatively sparse supervision, offering a flexible route to surrogate modeling that is mesh-free in space and time. Nevertheless, hyperbolic conservation laws pose additional difficulties: solutions may not be smooth, and, thus, the differential form of the PDE is not valid at discontinuities, which can degrade training stability and shock localization. This has been addressed by introducing various modifications such as conservative PINNs (cPINNs) [30], weak PINNs (wPINNs) and Integral PINNs (IPINNs) [31–33] that utilize weak formulation of PDEs, variational PINNs (VPINNs) [34,35] that adopt a Petrov–Galerkin viewpoint by pairing neural-network trial functions with polynomial test spaces. Architectural choices have also been explored; for example, physics-informed attention-based models have demonstrated improved shock-front representation and generalization behavior in hyperbolic settings [36].

Recently, PINNs have been applied to free-surface flows governed by the SWEs, motivated by the need for fast surrogates in forecasting and scenario exploration. Recent studies developed PINNs for SWEs in one and two spatial dimensions, including dam-break scenarios [37–39]. These studies also discuss practical issues such as stabilization, loss weighting, and the trade-off between smoothness and accuracy relative to shock-capturing numerical schemes. However, these studies primarily address fixed PDE instances or individual forward scenarios rather than learning a single parametric surrogate over a family of dam-break initial conditions. Thus, while they support the feasibility of physics-informed learning for SWE dynamics, they do not resolve the parametric generalization problem studied here.

This work constructs and compares two parametric reduced-order models for the one-dimensional shallow-water dam-break problem: a physics-informed neural network (PINN) and a non-intrusive tensorial reduced-order model (TROM). Both models are constructed using the same Full-Order Model (FOM) dataset and evaluated using consistent metric across dry-bed, wet-bed, and near-dry-bed regimes, following benchmark settings used in previous ROM studies [40–45] and recent TROM work [27]. We compare their predictive accuracy for the water depth and the discharge, their performance in parameter interpolation and extrapolation, and their respective offline and online computational costs. It is important to emphasize that both surrogate models are evaluated against the same FOM reference solution, rather than against the exact solution of the SWE, so that the surrogate approximation error is not conflated with the discretization error of the underlying numerical FOM solver. An important component of the study is the construction of a robust parametric PINN for this shock-dominated problem. Through

ablation experiments, we demonstrate that shock-aware collocation, residual scaling, and PDE gating substantially improve the accuracy and robustness of the PINN. Overall, the comparison identifies the complementary strengths and limitations of the two approaches for parametric hyperbolic problems.

The rest of the paper is organized as follows. Section 2 introduces the shallow-water model and the parametrized dam-break setup. Sections 3 and 4 give an overview of the tensorial Reduced-Order Model (TROM) and parametric Physics-Informed Neural Networks (PINNs). The numerical results for the interpolation and extrapolation in the parameter space are presented in Sections 5 and 6, respectively. Ablation studies for the PINN model are presented in Section 7. The discussion and conclusions are presented in Section 8.

2. Methods and Setup

In this section, we briefly outline the dam-break problem for the shallow-water equations and the numerical method.

2.1. Governing Equations

We consider the one-dimensional shallow water equations (SWEs), posed on a spatial interval $\Omega := [0, L]$ and time interval $[0, T]$. The equations can be written as a system of conservation laws

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = \mathbf{0}, \quad (x, t) \in \Omega \times (0, T], \quad (1)$$

with

$$\mathbf{U}(x, t; \boldsymbol{\mu}) := \begin{bmatrix} h(x, t; \boldsymbol{\mu}) \\ q(x, t; \boldsymbol{\mu}) \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} q \\ \frac{q^2}{h} + \frac{1}{2}gh^2 \end{bmatrix},$$

where $h(x, t; \boldsymbol{\mu}) \geq 0$ denotes the water depth, $q(x, t; \boldsymbol{\mu}) = h(x, t; \boldsymbol{\mu})u(x, t; \boldsymbol{\mu})$ is the discharge (momentum), u is the depth-averaged velocity, and $g > 0$ is the gravitational acceleration. Here $\boldsymbol{\mu}$ is a vector of parameters describing the initial conditions and $\mathbf{F}(\mathbf{U})$ is the flux function. The system in (1) is a strictly hyperbolic system for $h > 0$; solutions may develop discontinuities (shocks) even from smooth initial data. Consequently, the relevant notion of solution is that of a (weak) entropy solution; see, e.g., [2] for a detailed discussion.

2.1.1. Parametrized Dam-Break Initial Data

We study a parametrized dam-break configuration with a depth discontinuity located at $x_{\text{dam}} = L/2$. The parameter vector is $\boldsymbol{\mu} := (h_L, h_R) \in \mathcal{P} \subset \mathbb{R}^2$, where h_L and h_R denote the upstream and downstream depths, respectively, and satisfy $h_L > h_R \geq 0$. Thus, the initial dam condition is

$$h(x, 0; \boldsymbol{\mu}) = \begin{cases} h_L, & x < x_{\text{dam}}, \\ h_R, & x \geq x_{\text{dam}}, \end{cases} \quad u(x, 0; \boldsymbol{\mu}) = 0, \quad (2)$$

which implies $q(x, 0; \boldsymbol{\mu}) = 0$.

The case $h_R = 0$ is commonly referred to as the *dry-bed* dam-break problem and yields a solution dominated by a rarefaction wave, while the case $h_R > 0$ is the *wet-bed* dam-break problem and is more challenging due to the presence of a downstream shock for all $h_R > 0$ and for all $t > 0$. Although the solution of the wet-bed case converges uniformly to the solution of the dry-bed case on finite time intervals $[0, T]$ such that $x_{\text{dam}} + 2cT < L$ with $c = \sqrt{gh_L}$, $\partial h / \partial h_R \sim O(h_R^{-1/2})$ in a small interval near the shock for small h_R (see discussion in Appendix A of [27]). This complicates the use of projection-based

reduction techniques whose performance depends on the smoothness of the parametric solution manifold.

We consider a standard benchmark setup with $L = 100$, boundary conditions

$$\partial_x h(0, t; \boldsymbol{\mu}) = \partial_x u(0, t; \boldsymbol{\mu}) = \partial_x h(L, t; \boldsymbol{\mu}) = \partial_x u(L, t; \boldsymbol{\mu}) = 0, \quad t \in (0, T], \quad (3)$$

and parameter domain

$$[h_L^{\min}, h_L^{\max}] \times [h_R^{\min}, h_R^{\max}] = [10, 28] \times [0, 8]. \quad (4)$$

2.1.2. Parametric Solution Operator and Learning Objective

Let $\mathbf{U}^{\text{FOM}}(x, t; \boldsymbol{\mu})$ denote the FOM numerical reference solution of (1). The main goal of this paper is to examine two types of surrogate data-driven models for the *parameter-to-solution map*

$$\mathcal{S}: (x, t, \boldsymbol{\mu}) \mapsto \mathbf{U}^{\text{FOM}}(x, t; \boldsymbol{\mu}), \quad (x, t) \in \Omega \times [0, T], \boldsymbol{\mu} \in \mathcal{P}.$$

We emphasize that the parameter-to-solution map $\mathcal{S}$ is defined by the FOM solver and therefore inherits the discretization errors associated with the underlying numerical method.

The first model is a parametrized physics-informed neural network $\mathcal{N}_\theta := (h_\theta(x, t; \boldsymbol{\mu}), u_\theta(x, t; \boldsymbol{\mu}))$ where θ are trainable weights. This model should be consistent with the induced conservative quantity $q_\theta := h_\theta u_\theta$. The surrogate should provide accurate predictions for previously unseen parameter values $\boldsymbol{\mu}$ and times t (with particular interest in generalization within the training parameter domain). The learning problem is therefore to approximate $\mathcal{S}$ on $\Omega \times [0, T] \times \mathcal{P}$ while respecting the governing PDE (1), initial condition (2), boundary condition (3), and admissibility constraint $h(x, t) \geq 0$.

The second model is a non-intrusive tensorial reduced-order model (TROM). Unlike neural networks, this model does not require any iterative training, but it is still a data-driven model since it requires a dataset generated by numerical simulations of the FOM solver. The TROM uses a reduced representation which is obtained by performing a higher-order singular value decomposition (HOSVD) of the full snapshot matrix, nonlinear interpolation in parameter space, and reconstruction of the solution matrix from the reduced tensor representation. Essentially, this model learns the parameter-to-solution map $\mathcal{S}$ by computing the Tucker reduced tensor representation during the offline phase.

2.2. Numerical Method for the Full-Order Model

We use a Local Lax-Friedrichs finite-volume discretization to discretize the SWE in space. The semi-discrete scheme becomes

$$\frac{d}{dt} \mathbf{U}_i(t) = -\frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{\Delta x}, \quad i = 1, \dots, N$$

with

$$\mathbf{F}_{i+1/2} = \frac{\mathbf{F}(\mathbf{U}_i) + \mathbf{F}(\mathbf{U}_{i+1})}{2} - \frac{\lambda_{i+1/2}}{2} (\mathbf{U}_{i+1} - \mathbf{U}_i),$$

with $\lambda_{i+1/2} = \max(|u_i| + \sqrt{gh_i}, |u_{i+1}| + \sqrt{gh_{i+1}})$ and $\Delta x = L/N$, where N is the number of discrete points within the domain. For time-stepping, we use the Euler discretization with the time-step Δt satisfying the CFL condition $\max_i \lambda_{i+1/2} \Delta t / \Delta x < 1$. We use two ghost cells $\mathbf{U}_0 = \mathbf{U}_1$ and $\mathbf{U}_{N+1} = \mathbf{U}_N$ to satisfy the boundary conditions. It is known that the above discretization belongs to the class of strong stability preserving methods [46–48]. The scheme is also height-positivity-preserving. Thus, the non-negativity of the water depth h_i is preserved at the discrete level for all times.

To assess the influence of the FOM discretization, we also implemented a Lax-Wendroff finite-volume scheme equipped with the van Leer limiter (LW-vL) and compared both numerical solutions with the exact Riemann solution. As expected, the LW-vL scheme is more accurate, with errors in the water depth h and discharge q typically smaller by a factor of approximately 2–3 for the representative regimes considered. Nevertheless, the errors obtained with the two schemes remain of the same order of magnitude, and both LLF and LW-vL reproduce the same overall dam-break solution structure, including the rarefaction and shock propagation. The main differences are concentrated near discontinuities, where the LLF solution is more diffusive. In the present study, the exact solution is used only to assess the discretization error of the FOM, whereas the accuracy of the PINN and TROM is evaluated relative to the common LLF-generated FOM reference.

2.3. Dataset Construction

In the present study, we use a reduced parameter grid consisting of $(N_L \times N_R) = (5 \times 13)$ parameter instances, with sampled parameter values

$$\begin{aligned}\mathcal{P}_1 &= \{10, 14.5, 19, 23.5, 28\}, \\ \mathcal{P}_2 &= \{0, 0.1363, 0.5359, 1.1716, 2, 2.9647, 4, 5.0353, 6, 6.8284, 7.4641, 7.8637, 8\}.\end{aligned}\quad (5)$$

Therefore, the total sampled parameter set $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2$ contains $N_L N_R = 65$ sampled parameter values.

We integrate the full-order model (FOM) with the time-step $\Delta t = 10^{-4}$ to a final time $T = 2.5$ and store snapshots at times $t_k = k\Delta t_s$, $k = 1, \dots, N_T$ with $\Delta t_s = 0.1$. We use $N_x = 402$, which includes two ghost points (one on each boundary) for implementing the boundary conditions. Note that the snapshot times t_k do not include $t = 0$. The initial state is given analytically by (2) for every parameter value, so no surrogate is required at $t = 0$, and all comparisons reported below are made at $t > 0$. Should reconstruction at $t = 0$ be desired, the analytically known initial state can be appended to the snapshot data as an additional temporal slice without any further full-order computation. For each parameter value $\mu^{(m)}$, $m = 1, \dots, N_L N_R$, we store fields $h^{(m)}(x_i, t_k)$ and $u^{(m)}(x_i, t_k)$ with $i = 1, \dots, N_x$ and $k = 1, \dots, N_T$. These snapshots can be combined into a fourth-order tensor, as discussed in Section 3.

Figure 1 illustrates representative space–time structure in the generated FOM trajectories for a dry-bed case $(h_L, h_R) = (19, 0)$ and a wet-bed case $(h_L, h_R) = (19, 2)$. The depth profiles $h(t, x)$ (top row) for $t = 0, 0.5, 1, 2.5$ highlight the qualitative difference between regimes: the dry-bed solution is dominated by a rarefaction wave that spreads from the dam location, whereas the wet-bed solution exhibits a persistent downstream shock for all $t > 0$. The corresponding discharge profiles $q(t, x) = h(t, x)u(t, x)$ (bottom row) depict the local flow rate (volume flux per unit width) and highlight where the solution carries significant momentum.

Train/Validation/Test Splits

We design the evaluation protocol to probe generalization in parameter space, covering both interpolation within the sampled parameter domain (Section 5) and extrapolation beyond it (Section 6); in either case, predictions are compared with the FOM at the stored snapshot times t_k . The split used for training the PINN is specified in Section 4. The TROM requires no such split, as discussed at the beginning of Section 3.

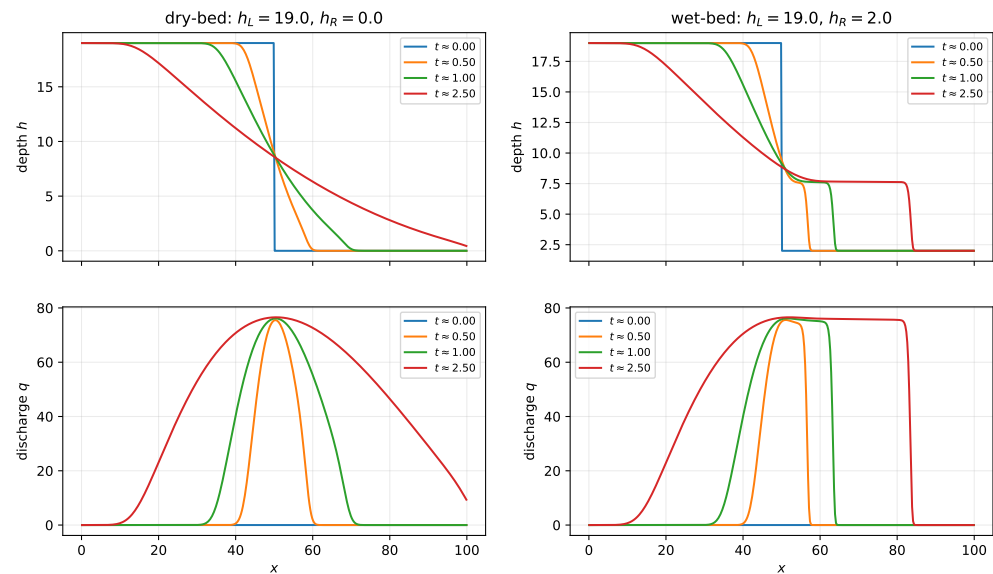


Figure 1. Representative FOM snapshots (depth $h(t, x)$ and discharge $q(t, x)$) for a dry-bed and a wet-bed choice of parameters $(h_L, h_R) = (19, 0)$ (left) and $(h_L, h_R) = (19, 2)$ (right), at times $t = 0, 0.5, 1.0, 2.5$. Top part—water depth $h(t, x)$, Bottom part—discharge $q(t, x)$.

3. Tensorial Reduced-Order Model

In this section, we present the non-intrusive Tensorial Reduced-Order Model (TROM); additional details can be found in [49]. As we discussed in Section 2.3, the solution of the full-order model depends on parameters $\mu^{(m)} \in \mathbb{R}^2$ and we consider the parameter grid $\mathcal{P}$ defined in (5). Using the full-order model, we generate solution snapshots $\{h^{(m)}(x_i, t_k), u^{(m)}(x_i, t_k)\}$, where $\mu^{(m)} \in \mathbb{R}^2$ is a particular parameter value from (5). For each field variable $\phi \in \{h, u\}$, these snapshots can be organized into a fourth-order tensor

$$\mathcal{Q}_\phi \in \mathbb{R}^{N_x \times N_L \times N_R \times N_T},$$

where the first mode corresponds to the spatial discretization ($N_x = 402$), modes 2 and 3 correspond to the parameter dimensions ($N_L = 5$ and $N_R = 13$), and the last mode corresponds to time ($N_T = 25$). We emphasize that, unlike the parametric PINN of Section 4, the TROM requires no iterative training. The low-rank representation is computed once, from the snapshots at all $N_L N_R = 65$ parameter values of the grid $\mathcal{P}$ in (5), all $N_T = 25$ snapshot times, and all $N_x = 402$ spatial grid points, without subsampling in space, time, or parameter space. The threshold ϵ in (7) below is the only tunable parameter of the construction. All parameter values at which the TROM is evaluated in Sections 5 and 6 lie outside the grid (5), so every TROM result reported in this paper is out of sample with respect to the parameter grid. Next, the Tucker decomposition [50] can be used to develop a compressed tensor format. In particular, the tensor $\mathcal{Q}_\phi$ can be represented as

$$\mathcal{Q}_\phi \approx \widetilde{\mathcal{Q}}_\phi = \mathcal{G}_\phi \times_1 \mathbf{V}_\phi^{(x)} \times_2 \mathbf{V}_\phi^{(L)} \times_3 \mathbf{V}_\phi^{(R)} \times_4 \mathbf{V}_\phi^{(T)}, \quad (6)$$

where $\mathcal{G}_\phi \in \mathbb{R}^{\tilde{N}_x \times \tilde{N}_L \times \tilde{N}_R \times \tilde{N}_T}$ is the *core tensor* and $\mathbf{V}_\phi^{(k)} \in \mathbb{R}^{N_k \times \tilde{N}_k}$ are factor matrices with $k \in \{x, L, R, T\}$. Here “ $\cdot \times_n \cdot$ ” denotes the *mode- n product* of a tensor with a matrix (see, e.g., ref. [51] for a review of tensor algebra). The tuple $(\tilde{N}_x, \tilde{N}_L, \tilde{N}_R, \tilde{N}_T)$ represents the multilinear rank of the tensor $\widetilde{\mathcal{Q}}_\phi$. The Tucker format allows independent compression ratios for each mode, and orthonormality of the factor matrices ensures numerical stability and direct error control. The Higher-Order SVD (HOSVD) [52] can be used to compute the Tucker decomposition by performing the SVD of each mode- n unfolding independently and

selecting the truncation rank for each mode. The multilinear ranks $\tilde{N}_k$ with $k \in \{x, L, R, T\}$ are selected using a criterion based on the singular value decay:

$$\tilde{N}_k = \min \left\{ r : \frac{\sum_{i=1}^r \sigma_{i,k}}{\sum_{i=1}^{\tilde{N}_k} \sigma_{i,k}} \geq 1 - \epsilon \right\}, \quad (7)$$

where $\sigma_{i,k}$ are the singular values of the k -mode unfolding of $\mathcal{Q}_\phi$. We use $\epsilon = 10^{-3}$ as a threshold for all modes. The criterion (7) is based on the normalized cumulative sum of the singular values rather than of their squares. The two variants are related: for a non-increasing sequence $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_N \geq 0$ and any r ,

$$\left(\sum_{i \leq r} \sigma_i^2 \right) \left(\sum_{i=1}^N \sigma_i \right) - \left(\sum_{i \leq r} \sigma_i \right) \left(\sum_{i=1}^N \sigma_i^2 \right) = \sum_{i \leq r} \sum_{j > r} \sigma_i \sigma_j (\sigma_i - \sigma_j) \geq 0,$$

since $\sigma_i \geq \sigma_j$ whenever $i \leq r < j$, so the normalized partial sums of the squared singular values dominate those of the singular values themselves. Consequently, for a fixed ϵ , the criterion (7) selects multilinear ranks no smaller than those produced by the standard squared-singular-value (energy) criterion, i.e., it is the more conservative of the two. The multilinear ranks obtained from the truncated HOSVD using the criterion above are $(\tilde{N}_x, \tilde{N}_L, \tilde{N}_R, \tilde{N}_T) = (94, 4, 5, 21)$.

We next quantify the storage reduction achieved by the Tucker representation at these multilinear ranks. For each field, the snapshot tensor $\mathcal{Q}_\phi$ contains $N_x N_L N_R N_T = 402 \times 5 \times 13 \times 25 = 653,250$ entries, whereas the Tucker representation (6) requires $\tilde{N}_x \tilde{N}_L \tilde{N}_R \tilde{N}_T = 39,480$ entries for the core tensor and $402 \times 94 + 5 \times 4 + 13 \times 5 + 25 \times 21 = 38,398$ entries for the four factor matrices, i.e., 77,878 entries in total. This corresponds to a storage reduction by a factor of approximately 8.4, or 11.9% of the uncompressed snapshot data. Note that the ranks in the spatial and temporal modes are limited by the size of those modes themselves: with only $N_T = 25$ stored snapshots the temporal mode is intrinsically low-dimensional and close to full rank by construction, so for a parametric surrogate the essential compression takes place in the parameter modes and in the core tensor. We also emphasize that the non-intrusive TROM is not intended to circumvent the slow decay of the Kolmogorov n -width that is characteristic of transport-dominated problems and was discussed in Section 1; the ranks above are those returned by the conservative criterion (7) at $\epsilon = 10^{-3}$ and are not claimed to be minimal. The accuracy and the online cost of the resulting representation are reported in Sections 5 and 6, and its limitations near the moving front and outside the sampled parameter range are discussed in Section 8. For the non-intrusive TROM, solution snapshots for a new parameter value $\mu^* \in \mathbb{R}^2$ are constructed directly from the low-order representation (6) without any time integration (see also [28]). First, interpolation coefficients for the new parameter values are computed using cubic spline interpolation on the grid $\mathcal{P}_k$

$$\chi_k(\mu_k^*) = \text{interp. coefficients}(\mathcal{P}_k, \mu_k^*), \quad k = 1, 2.$$

The vector $\chi_k \in \mathbb{R}^{N_k}$ contains the interpolation coefficients corresponding to the parameter value μ_k^* on the grid $\mathcal{P}_k$. Next, the Tucker tensor from (6) is contracted along all parameter modes with the interpolation vectors:

$$\mathcal{C}_\phi(\mu^*) = \mathcal{G}_\phi \times_2 \left(\mathbf{V}_\phi^{(L)\top} \chi_1(\mu_1^*) \right) \times_3 \left(\mathbf{V}_\phi^{(R)\top} \chi_2(\mu_2^*) \right). \quad (8)$$

This yields the low-dimensional matrix $\mathcal{C}_\phi(\boldsymbol{\mu}^*) \in \mathbb{R}^{\tilde{N}_x \times \tilde{N}_T}$. The solution is then reconstructed via

$$\mathbf{X}_\phi(\boldsymbol{\mu}^*) = \mathbf{V}_\phi^{(x)} \mathcal{C}_\phi(\boldsymbol{\mu}^*) \left(\mathbf{V}_\phi^{(T)} \right)^\top \in \mathbb{R}^{N_x \times N_T},$$

where the k -th column of $\mathbf{X}_\phi(\boldsymbol{\mu}^*)$ approximates $\phi(\mathbf{x}, t_k; \boldsymbol{\mu}^*)$. The reconstruction above returns the solution at the stored snapshot times $t_k, k = 1, \dots, N_T$, which are the columns of $\mathbf{X}_\phi(\boldsymbol{\mu}^*)$; the non-intrusive TROM as formulated here performs interpolation in parameter space only. Predictions at times between the stored snapshots can be obtained within the same framework by contracting the temporal mode of (6) with spline interpolation coefficients $\chi_T(t^*)$ computed on the grid $\{t_k\}$, exactly as is done above for the parameter modes. We do not pursue this extension here, and all TROM results reported below are evaluated at the snapshot times t_k .

4. Parametric Physics-Informed Neural Network

Recall from Section 2 that the parametric solution operator maps $(x, t, \boldsymbol{\mu}) \in \Omega \times [0, T] \times \mathcal{P}$ to the state of the flow. Here, $\Omega = [0, L]$ and $\mathcal{P} \subset \mathbb{R}^2$ is the parameter domain. Accordingly, we seek a neural network approximation $\mathcal{N}_\theta$ of the form

$$\mathcal{N}_\theta : \mathbb{R}^4 \rightarrow \mathbb{R}^2, \quad \mathcal{N}_\theta : (x, t, \boldsymbol{\mu}) \mapsto (h_\theta(x, t; \boldsymbol{\mu}), u_\theta(x, t; \boldsymbol{\mu})),$$

where θ denotes the trainable network parameters. Additional details about the PINN presented in this Section can be found in [53].

Rather than feeding $\boldsymbol{\mu} = (h_L, h_R)$ directly into the network, we use the equivalent parameterization

$$\bar{h} := \frac{h_L + h_R}{2}, \quad \gamma := \frac{h_L - h_R}{h_L + h_R}, \quad \bar{\boldsymbol{\mu}} := (\bar{h}, \gamma) \quad (9)$$

Thus, the actual network input and output are

$$\mathbf{z} := (x, t, \bar{\boldsymbol{\mu}}) = (x, t, \bar{h}, \gamma) \in \mathbb{R}^4, \quad \mathcal{N}_\theta(\mathbf{z}) = \begin{bmatrix} h_\theta(\mathbf{z}) \\ u_\theta(\mathbf{z}) \end{bmatrix} \in \mathbb{R}^2.$$

This reparameterization separates the mean water-depth scale $\bar{h}$ from the relative jump magnitude γ . Empirically, this provides a more balanced description of the regime space and is natural for dam-break problems.

Network architecture

We approximate the parametric solution map by a fully connected feed-forward neural network without skip connections or residual blocks. The network with K hidden layers is defined recursively by

$$\mathbf{a}^{(0)} = \mathbf{z}, \quad (10)$$

$$\mathbf{a}^{(k+1)} = \sigma(W^{(k)}\mathbf{a}^{(k)} + \mathbf{b}^{(k)}), \quad k = 0, \dots, K-2, \quad (11)$$

$$\mathcal{N}_\theta(\mathbf{z}) = W^{(K-1)}\mathbf{a}^{(K-1)} + \mathbf{b}^{(K-1)}, \quad (12)$$

where $W^{(k)}$ and $\mathbf{b}^{(k)}$ are the trainable weights and biases of layer k , and σ is the hidden-layer activation function. Throughout this work, we use the *sigmoid linear unit* (SiLU) activation function $\sigma(x) = x \operatorname{sigmoid}(x) = x/(1 + e^{-x})$. The SiLU activation is smooth and non-polynomial, which is advantageous in PINNs because the PDE residual requires repeated differentiation of the network output with respect to the input coordinates. Compared with

piecewise-linear activations, smooth activations produce smooth derivatives and therefore more stable automatic-differentiation-based residuals.

All linear layers are initialized with *Xavier normal initialization* [54]. If a fully connected layer has weight matrix W , then each entry is initialized as $W_{ij} \sim \mathcal{N}(0, 2/(m+n))$. This initialization is designed to keep activation magnitudes at a reasonable scale across layers and to improve optimization stability in deep feed-forward networks. Biases are initialized to zero.

Physics-informed residual

Although the network predicts the primitive variables (h_θ, u_θ) , the governing equations from Section 2 are enforced through the conservative form of the PDE, using the induced discharge $q_\theta = h_\theta u_\theta$. Then the two strong-form residuals are

$$\mathcal{R}_1(x, t; \mu) := \partial_t h_\theta + \partial_x (h_\theta u_\theta), \quad (13)$$

$$\mathcal{R}_2(x, t; \mu) := \partial_t (h_\theta u_\theta) + \partial_x \left(h_\theta u_\theta^2 + \frac{1}{2} g h_\theta^2 \right). \quad (14)$$

These residuals are exactly the mass and momentum residuals corresponding to the conservative shallow-water system when expressed through the primitive variables. In practice, the derivatives in (13) and (14) are computed by automatic differentiation in PyTorch, by differentiating the network outputs with respect to the input coordinates x and t using autograd.

This formulation keeps the exposition consistent with the rest of the paper: the network predicts (h, u) , training data are stored in (h, u) , and only the induced conservative quantity $q = hu$ is formed internally when the governing equations require it.

4.1. Training Data and Collocation Sets

Training uses four types of information: (1) supervised solution snapshots, (2) collocation points for PDE residual enforcement, (3) boundary points for enforcing the boundary conditions, and (4) points at $t = 0$ for enforcing the initial condition. First, let

$$\mathcal{D}_{\text{data}} = \{(x_i, t_i, \mu_i, h_i, u_i)\}_{i=1}^{N_d}$$

denote supervised solution snapshots generated by the reference finite-volume solver. These points are sampled from multiple dam-break regimes in the training set. The spatial and parametric coordinates of boundary points extracted from these snapshots form the set

$$\mathcal{D}_{\text{bc}} = \{(x_k^{\text{bc}}, t_k^{\text{bc}}, \mu_k^{\text{bc}})\}_{k=1}^{N_{\text{bc}}},$$

where $x_k^{\text{bc}} \in [0, L]$. These points are used to enforce the zero-gradient boundary condition.

In addition, we construct the initial-condition set

$$\mathcal{D}_{\text{ic}} = \{(x_m^{\text{ic}}, 0, \mu_m^{\text{ic}}, h_m^{\text{ic}}, u_m^{\text{ic}})\}_{m=1}^{N_{\text{ic}}},$$

where the targets are defined analytically by

$$h_m^{\text{ic}} = \begin{cases} h_L, & x_m^{\text{ic}} < L/2, \\ h_R, & x_m^{\text{ic}} \geq L/2, \end{cases} \quad u_m^{\text{ic}} = 0.$$

For the final model, the supervised training set uses all $N_x = 400$ spatial grid points (excluding the two ghost cells) from each snapshot at $t_k = k\Delta t_s$, $k = 1, \dots, 25$, for every $\mu \in P$, without spatial subsampling. Thus, $N_d = 65 \times 25 \times 400 = 650,000$. The initial-

condition set contains all spatial points for every training parameter instance, giving $N_{ic} = 65 \times 400 = 26,000$, while the boundary set contains both boundary points at every stored positive-time snapshot, giving $N_{bc} = 65 \times 25 \times 2 = 3250$.

Second, let

$$\mathcal{D}_{col} = \{(x_j^{(f)}, t_j^{(f)}, \mu_j^{(f)})\}_{j=1}^{N_f}$$

denote collocation points for enforcing the PDE residual. These points are sampled in the interior of the space-time-parameter domain. Depending on the experiment, collocation may be sampled uniformly or with a shock-aware strategy. In all cases, the method itself is unchanged: the PDE residual is penalized at collocation points, while data and boundary mismatches are penalized at supervised points.

Shock-Aware Collocation Strategy

For wet-bed regimes, strong-form PDE residuals are most difficult to enforce in a narrow neighborhood of the moving shock, where the entropy solution is non-smooth and not classically differentiable. Because strong-form PINNs rely on pointwise spatial and temporal derivatives, PDE enforcement in this region becomes particularly difficult due to high gradients. To avoid over-penalizing the network in this region, we optionally replace uniform collocation sampling in x by a shock-aware strategy that excludes a small band around the shock. Thus, we first estimate the location of the shock, $x_{sh}(t; \mu)$, from the reference FOM dataset (see Appendix A). Given the estimated shock position, we then define a spatial exclusion region for collocation points

$$B_{excl}(t; \mu) := \{x \in \Omega : |x - x_{sh}(t; \mu)| < w_{excl}\}, \quad (15)$$

where $w_{excl} > 0$ is a parameter that defines the exclusion half-width. The spatial collocation points for the PDE residual are then sampled from $x \sim \text{Unif}(\Omega \setminus B_{excl}(t; \mu))$. Consequently, no collocation points are placed inside the excluded shock core B_{excl} .

For dry-bed or near-dry regimes, shock exclusion is disabled. Specifically, if $h_R < \varepsilon_{dry}$ for a prescribed threshold $\varepsilon_{dry} > 0$, then x is sampled uniformly on the full spatial domain Ω . This design reflects the fact that the dry-bed problem is dominated by a rarefaction structure rather than a persistent downstream shock of the wet-bed type. Values of the parameters are reported at the end of the next subsection.

4.2. Loss Function

The loss function is a weighted sum

$$\mathcal{L}(\theta) = \lambda_{data} \mathcal{L}_{data} + \lambda_{pde} \mathcal{L}_{pde} + \lambda_{bc} \mathcal{L}_{bc} + \lambda_{ic} \mathcal{L}_{ic} + \lambda_{nn} \mathcal{L}_{nn}, \quad (16)$$

where $\mathcal{L}_{data}$ is the supervised data loss, $\mathcal{L}_{pde}$ is the PDE residual loss, $\mathcal{L}_{bc}$ is the boundary-condition loss, $\mathcal{L}_{ic}$ is the initial-condition loss, and $\mathcal{L}_{nn}$ is an optional non-negativity penalty for the depth.

1. Data loss.

The data loss is computed using

$$\mathcal{L}_{data} = \frac{1}{N_d} \sum_{i=1}^{N_d} [\ell_h(e_i^{(h)}) + \lambda_u \ell_u(e_i^{(u)})], \quad e_i^{(h)} = \frac{h_{\theta,i} - h_i}{\sigma_h}, \quad e_i^{(u)} = \frac{u_{\theta,i} - u_i}{\sigma_u}, \quad (17)$$

where $\sigma_h = \text{StdDev}(h_i)$, $\sigma_u = \text{StdDev}(u_i)$ computed over the entire training dataset $\{(h_i, u_i)\}$, and the scalar $\lambda_u > 0$ balances the velocity contribution relative to the depth contribution. Here, ℓ_h, ℓ_u denote either $|\cdot|$ or $(\cdot)^2$, depending on the experiment.

2. PDE residual loss.

The PDE residual (13) and (14) is penalized at chosen collocation points. Since the two residual components may differ substantially in magnitude, especially during early training, we rescale them component-wise before aggregation:

$$\tilde{\mathcal{R}}_i = \mathcal{R}_i / s_i, \quad (18)$$

where s_i are exponential-moving-average root-mean-square (EMA-RMS) scales (see Appendix A). Hence, the strong-form PDE loss is

$$\mathcal{L}_{\text{pde}} = \frac{1}{N_f} \sum_{j=1}^{N_f} w_j \left[\psi(\tilde{\mathcal{R}}_1(z_j^{(f)}; \theta)) + \psi(\tilde{\mathcal{R}}_2(z_j^{(f)}; \theta)) \right], \quad (19)$$

where $w_j \in (0, 1]$ is a point-dependent gating weight used to downweight strong-form residual enforcement in regions where the solution is less regular, such as near steep gradients or very small predicted depths. Details are presented in Appendix A.

The penalty function ψ in (19) is either the quadratic penalty $\psi(r) = r^2$ or the Charbonnier penalty $\psi(r) = \sqrt{r^2 + \varepsilon^2}$, with $\varepsilon = 10^{-5}$, which behaves like $|r|$ for large residuals but remains smooth near zero. This makes it a differentiable approximation to the absolute value and can improve optimization stability when residual outliers occur.

3. Boundary loss.

At the left and right boundaries, we impose a zero-gradient Neumann condition on the primitive variables,

$$\partial_x h_\theta = 0, \quad \partial_x u_\theta = 0, \quad x \in \{0, L\}.$$

The corresponding boundary loss is

$$\mathcal{L}_{\text{bc}} = \frac{1}{N_{\text{bc}}} \sum_{k=1}^{N_{\text{bc}}} \left[\left(\frac{\partial_x h_\theta(z_k^{\text{bc}})}{\sigma_h} \right)^2 + \lambda_u \left(\frac{\partial_x u_\theta(z_k^{\text{bc}})}{\sigma_u} \right)^2 \right]. \quad (20)$$

The spatial derivatives are evaluated using automatic differentiation. This term stabilizes the learned solution near the edges of the spatial domain and ensures consistency with the reference trajectories used in training.

4. Initial-condition loss.

The initial dam-break state is imposed explicitly using the analytically constructed dataset $\mathcal{D}_{\text{ic}}$. The corresponding loss is

$$\mathcal{L}_{\text{ic}} = \frac{1}{N_{\text{ic}}} \sum_{m=1}^{N_{\text{ic}}} \left[\ell_h \left(\frac{h_\theta(z_m^{\text{ic}}) - h_m^{\text{ic}}}{\sigma_h} \right) + \lambda_u \ell_u \left(\frac{u_\theta(z_m^{\text{ic}}) - u_m^{\text{ic}}}{\sigma_u} \right) \right],$$

where the penalty functions ℓ_h and ℓ_u are chosen consistently with the supervised data loss. This term enforces the discontinuous initial depth profile and the initially stationary velocity field directly at $(t = 0)$.

5. Non-negativity penalty.

Physical admissibility requires $h \geq 0$. The implementation therefore supports an additional penalty term that penalizes negative depth predictions on supervised points

$$\mathcal{L}_{\text{nn}} = \frac{1}{N_d} \sum_{i=1}^{N_d} \max(0, -h_{\theta,i}).$$

6. Optimization.

The network is trained by minimizing (16) with Adam optimizer [55]. We also implemented a cosine-annealing learning-rate schedule and gradient clipping to control occasional large updates. In addition, the PDE, boundary-condition, and initial-condition weights are ramped up during the early training epochs. This warm-up strategy is designed to avoid forcing the network too aggressively toward the physics term before it has learned a reasonable coarse representation of the data. Such staged weighting can improve optimization stability for parametric hyperbolic problems, where the residual landscape may be highly nonuniform across regimes and times.

Final Training Configuration

The final model is a fully connected parametric PINN with four-dimensional input $(x, t, \bar{h}, \gamma)$, where $(\bar{h}, \gamma)$ is a transformation of μ in (9). The network has five hidden layers of width 120, SiLU activation, and a two-dimensional output for (h, u) . The network is trained for 20,000 epochs with Adam using learning rate 10^{-3} . A cosine scheduler is used with $T_{\text{max}} = 20,000$ and $\eta_{\text{min}} = 10^{-5}$, and gradients are clipped at 1.0. The final loss weights are $\lambda_{\text{data}} = 1.0$, $\lambda_{\text{pde}} = 0.3$, $\lambda_{\text{bc}} = 0.6$, $\lambda_{\text{ic}} = 1.0$, $\lambda_{\text{nn}} = 0.01$, with an additional velocity weight $\lambda_u = 5.0$. The supervised data and initial-condition terms use L^1 -type penalties, while the PDE residual and boundary-condition terms use L^2 -type penalties. The PDE, boundary-condition, and initial-condition weights are initialized at 10% of their final values and increased linearly to their target values over the first quarter of training. For strong-form residual enforcement, we use $N_{\text{colloc}}^{\text{train}} = 80,000$, $N_{\text{colloc}}^{\text{val}} = 20,000$ collocation points. Shock-aware collocation is enabled with exclusion width $w_{\text{excl}} = 2.5$ and dry-bed threshold $\epsilon_{\text{dry}} = 0.5$. EMA-RMS residual scaling is used with $\beta = 0.99$ and $\epsilon_{\text{scale}} = 10^{-8}$. The PDE gating parameters are set to $\alpha = 8.0$, $h_{\text{min}} = 0.05$, and $w_{\text{min}} = 0.05$. The model contains 58,922 trainable parameters.

For validation, we use the held-out parameter value $\mu_{\text{val}} = (17, 4)$, which does not belong to P , and generate FOM snapshots with sampling interval $\Delta t_s^{\text{val}} = 0.25$. The validation PDE residual is evaluated at 20,000 uniformly sampled collocation points. All training losses are evaluated in full-batch mode. NumPy (v. 2.3.1) and PyTorch (v. 2.5.1) random-number generators are initialized with seed 42 for ablation study and seeds 42–46 for numerical results in Tables 1 and 2. Visual comparison is presented for the network initialized with seed 42. The network at the final training epoch (20,000) is used for all reported test evaluations.

All PINN computations were performed in single precision on the CPU of a MacBook Pro equipped with an Apple M1 Max chip with 10 CPU cores and 64 GB of unified memory, using 10 PyTorch threads.

Table 1. Relative L^2 Errors in (21) for the TROM and the PINN predictions at time $t = 1$. PINN results are reported as mean $\pm$ sample standard deviation over five independent training runs.

μ	ϵ_h^{TROM}	ϵ_h^{PINN}	ϵ_q^{TROM}	ϵ_q^{PINN}
(12, 0)	1.4027×10^{-3}	$(6.30 \pm 1.22) \times 10^{-3}$	1.1859×10^{-2}	$(4.24 \pm 0.66) \times 10^{-2}$
(12, 7)	4.0641×10^{-3}	$(2.66 \pm 0.60) \times 10^{-3}$	4.1335×10^{-2}	$(1.409 \pm 0.270) \times 10^{-2}$
(15, 4)	3.2675×10^{-3}	$(3.18 \pm 1.63) \times 10^{-3}$	1.9264×10^{-2}	$(7.16 \pm 1.15) \times 10^{-3}$
(18, 0)	3.5761×10^{-4}	$(7.16 \pm 1.18) \times 10^{-3}$	2.8654×10^{-3}	$(5.11 \pm 0.85) \times 10^{-2}$
(26, 0.14)	5.6640×10^{-3}	$(1.494 \pm 0.209) \times 10^{-2}$	3.3705×10^{-2}	$(8.11 \pm 1.22) \times 10^{-2}$
(26, 7)	6.0776×10^{-3}	$(2.44 \pm 1.59) \times 10^{-3}$	2.9323×10^{-2}	$(5.88 \pm 2.88) \times 10^{-3}$

Table 2. Relative L^2 Errors in (21) for the TROM and the PINN predictions at time $t = 2$. PINN results are reported as mean $\pm$ sample standard deviation over five independent training runs.

μ	$\varepsilon_h^{\text{TROM}}$	$\varepsilon_h^{\text{PINN}}$	$\varepsilon_q^{\text{TROM}}$	$\varepsilon_q^{\text{PINN}}$
(12, 0)	2.5749×10^{-3}	$(8.55 \pm 1.27) \times 10^{-3}$	1.5302×10^{-2}	$(4.80 \pm 0.71) \times 10^{-2}$
(12, 7)	1.1370×10^{-2}	$(2.59 \pm 0.48) \times 10^{-3}$	8.1167×10^{-2}	$(9.66 \pm 0.78) \times 10^{-3}$
(15, 4)	9.3945×10^{-3}	$(4.64 \pm 2.34) \times 10^{-3}$	3.7016×10^{-2}	$(5.95 \pm 1.93) \times 10^{-3}$
(18, 0)	7.2675×10^{-4}	$(9.56 \pm 1.36) \times 10^{-3}$	4.1002×10^{-3}	$(4.98 \pm 0.71) \times 10^{-2}$
(26, 0.14)	1.5224×10^{-2}	$(2.624 \pm 0.308) \times 10^{-2}$	6.0324×10^{-2}	$(1.008 \pm 0.106) \times 10^{-1}$
(26, 7)	2.0034×10^{-2}	$(3.86 \pm 1.66) \times 10^{-3}$	6.4883×10^{-2}	$(4.66 \pm 2.34) \times 10^{-3}$

5. Numerical Results

To quantify our results, we use relative L^2 errors for $h(x, t)$ and $q(x, t)$. In particular, we define

$$\varepsilon_h^{\text{Pred}}(t; \mu) = \frac{\|h^{\text{Pred}}(t; \mu) - h^{\text{FOM}}(t; \mu)\|_2}{\|h^{\text{FOM}}(t; \mu)\|_2}, \quad \varepsilon_q^{\text{Pred}}(t; \mu) = \frac{\|q^{\text{Pred}}(t; \mu) - q^{\text{FOM}}(t; \mu)\|_2}{\|q^{\text{FOM}}(t; \mu)\|_2}, \quad (21)$$

where h^{FOM} and q^{FOM} stand for numerical solutions of the full-order model, and h^{Pred} and q^{Pred} denote reduced model predictions using TROM or PINN. Prediction errors in (21) measure the fidelity of each surrogate relative to the FOM numerical solution, which defines the parameter-to-solution map $\mathcal{S}$ that both reduced models are constructed to approximate. The discretization error of the full-order scheme itself is therefore not included in (21). Since the same reference data are used for both surrogates, this error is common to both surrogate evaluations and is distinct from the surrogate approximation error measured here. We consider six representative parameter regimes

$$(h_L, h_R) = (12, 0), (12, 7), (15, 4), (18, 0), (26, 0.14), (26, 7) \quad (22)$$

evaluated at $t \in \{0.5, 1, 1.5, 2\}$. These cases include both dry-bed and wet-bed settings and are used to compare qualitative field predictions against the FOM simulations. All parameter regimes in (22) are out-of-sample with respect to the parameter pairs used for training. To assess the sensitivity of the PINN predictions to random initialization and stochastic sampling, the final PINN configuration was trained independently using five random seeds, 42–46. Accordingly, the PINN errors reported in Tables 1 and 2 are given as the mean $\pm$ sample standard deviation over these five runs. The relatively small variation across the independent PINN runs indicates that the principal accuracy trends are stable with respect to the random seed.

We depict a comparison of the water depth $h(x, t; \mu)$ and the discharge $q(x, t; \mu)$ in Figures 2, 3, 4 and 5, respectively.

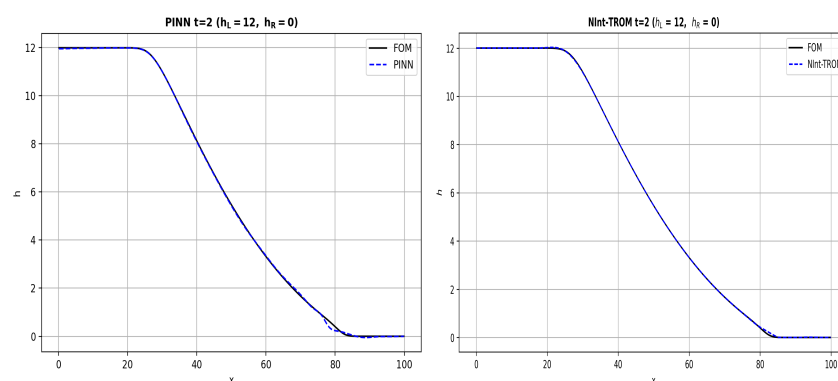


Figure 2. Cont.

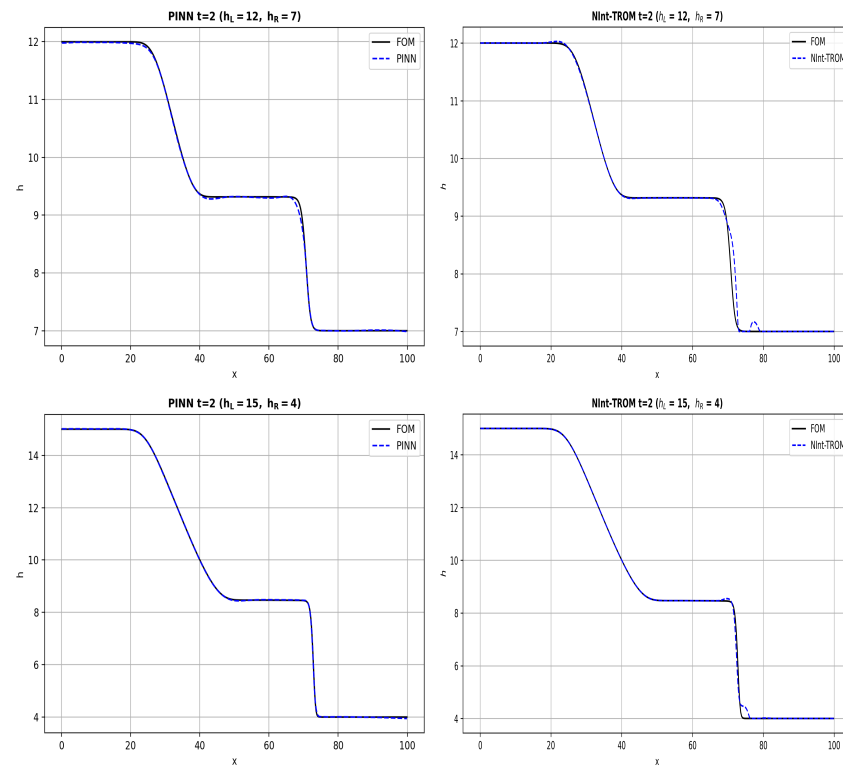


Figure 2. Comparison of $h(x, t; \mu)$ from the FOM simulations and predictions of the PINN (left column) and the TROM (right column) for $\mu = (12, 0), (12, 7), (15, 4)$ (top to bottom) at time $t = 2$.

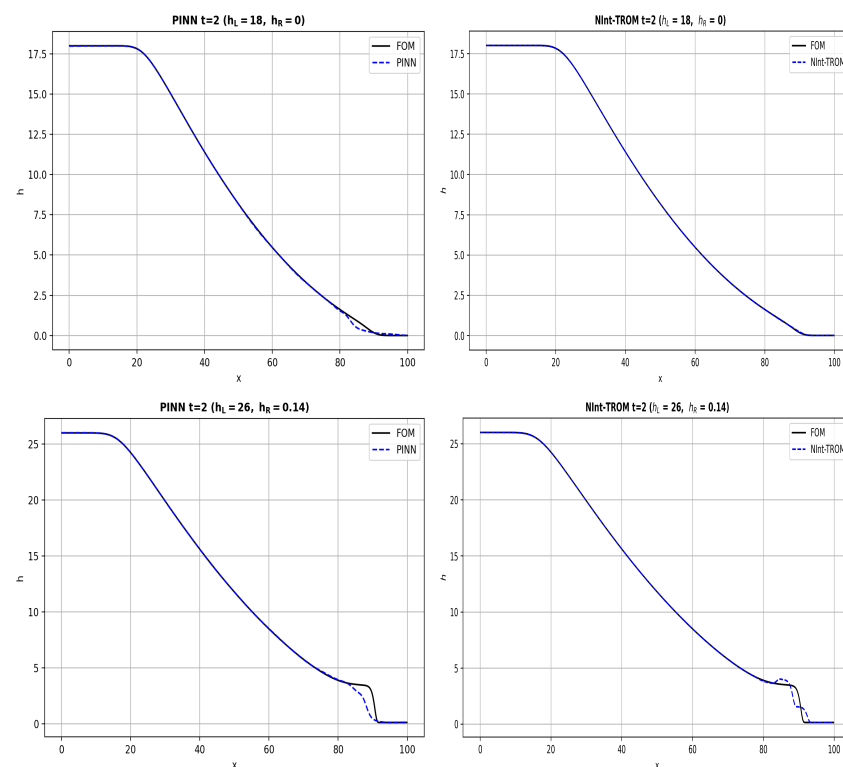


Figure 3. Cont.

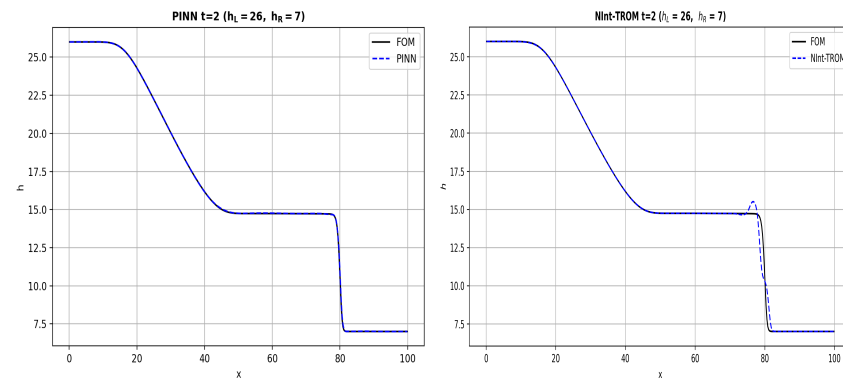


Figure 3. Comparison of $h(x, t; \mu)$ from the FOM simulations and predictions of the PINN (left column) and the TROM (right column) for $\mu = (18, 0), (26, 0.14), (26, 7)$ (top to bottom) at time $t = 2$.

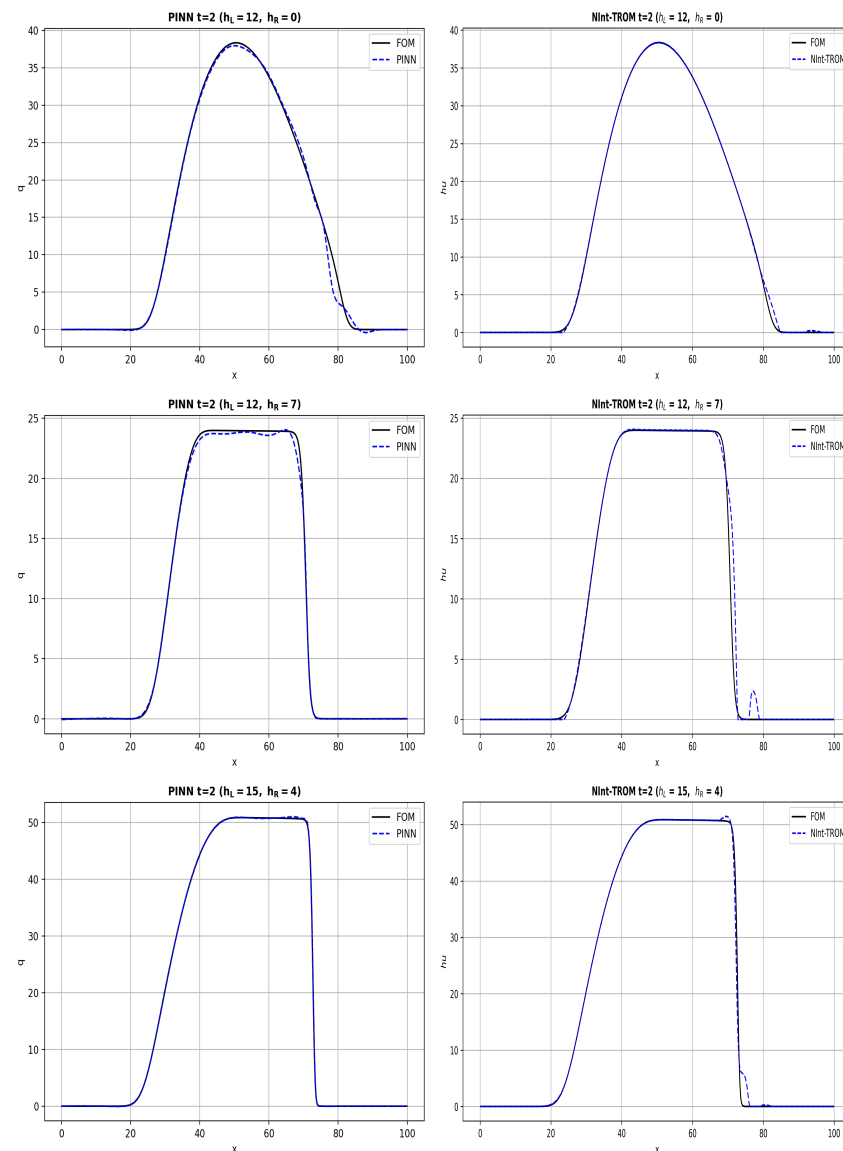


Figure 4. Comparison of $q(x, t; \mu)$ from the FOM simulations and predictions of the PINN (left column) and the TROM (right column) for $\mu = (12, 0), (12, 7), (15, 4)$ (top to bottom) at time $t = 2$.

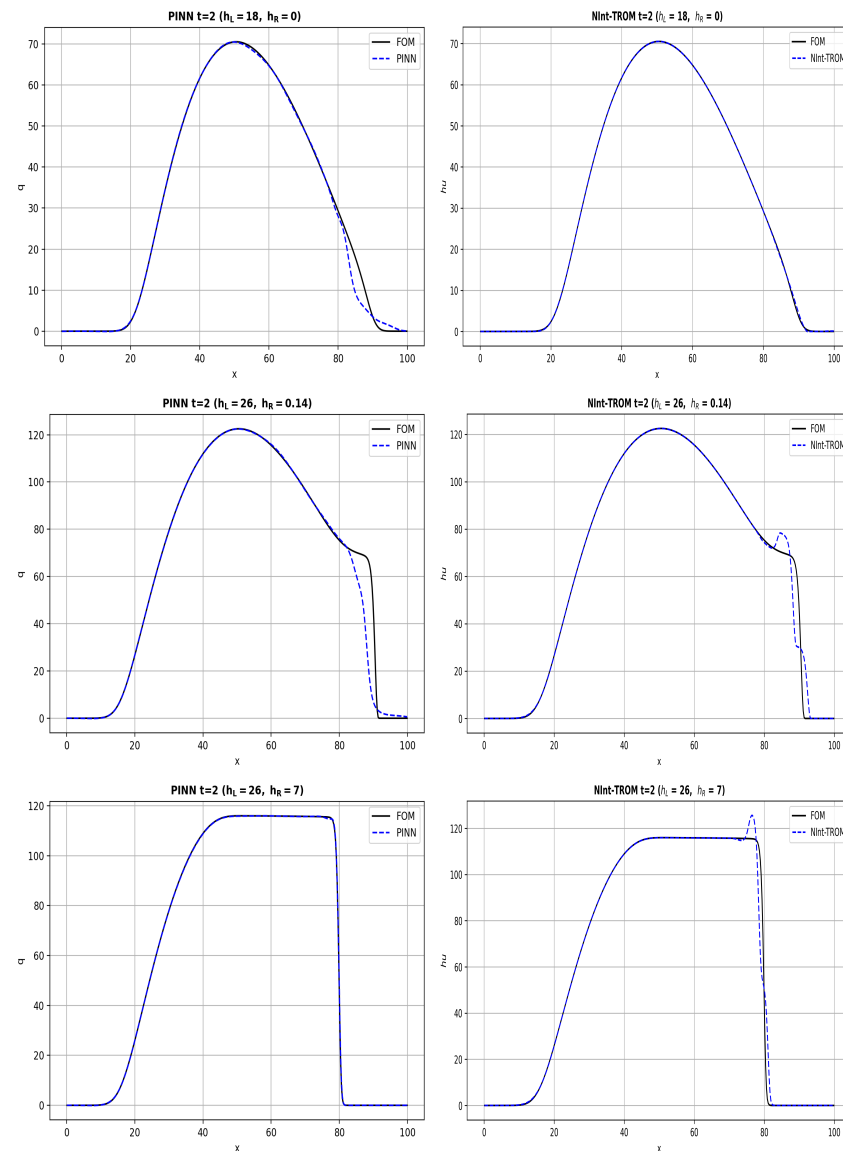


Figure 5. Comparison of $q(x, t; \mu)$ from the FOM simulations and predictions of the PINN (left column) and the TROM (right column) for $\mu = (18, 0)$, $(26, 0.14)$, $(26, 7)$ (top to bottom) at time $t = 2$.

Both reduced models provide accurate predictions across all six parameter regimes and reproduce the shock location with good accuracy. The largest discrepancies usually occur near the moving discontinuities, where small errors in both profiles can produce noticeable pointwise differences. Nevertheless, the relative L^2 errors remain low overall, consistent with the close agreement between the predicted and the FOM profiles of $h(x, t)$ and $q(x, t)$ at $t = 2$ shown in Figures 2–5. The most challenging case for both models is the near-dry-bed regime $\mu = (26, 0.14)$, for which both methods exhibit larger errors, particularly in the vicinity of the shock. A clear regime-dependent trend is also observed: the TROM is generally more accurate for the dry-bed cases, whereas the parametric PINN consistently achieves lower errors in the wet-bed regimes.

A well-known limitation of deep neural networks is their spectral bias (or Frequency Principle) [56–58], when the low-frequency (large-scale) components of a target function (the solution map in our case) are learned earlier and more rapidly than high-frequency components, making sharp gradients and discontinuities more difficult to approximate accurately. To further assess the performance of the parametric PINN model, we depict the magnitude of the spectra of the predicted water depth $h(x, t = 2)$ in Figure 6. To this end,

we first remove the linear trend connecting the boundary values, $(h(0, t), h(L, t)) = (h_L, h_R)$, and then compute the magnitude of the fast Fourier transform. The largest spectral discrepancies occur at high wavenumbers, particularly for the dry-bed and near-dry-bed regimes, which contain the sharpest solution features. Among the tested cases, $\mu = (26, 0.14)$ presents the greatest challenge for reproducing high-frequencies of the solution; here the spectra in the predicted solution decays faster than the spectra in the reference FOM solution. For all other cases, the predicted spectra show good agreement with the reference FOM over a broad range of wavenumbers, while the largest discrepancies remain concentrated at high wavenumbers. Thus, the spectral agreement is stronger for larger values of h_R , whereas the dry-bed and near-dry-bed regimes remain more challenging.

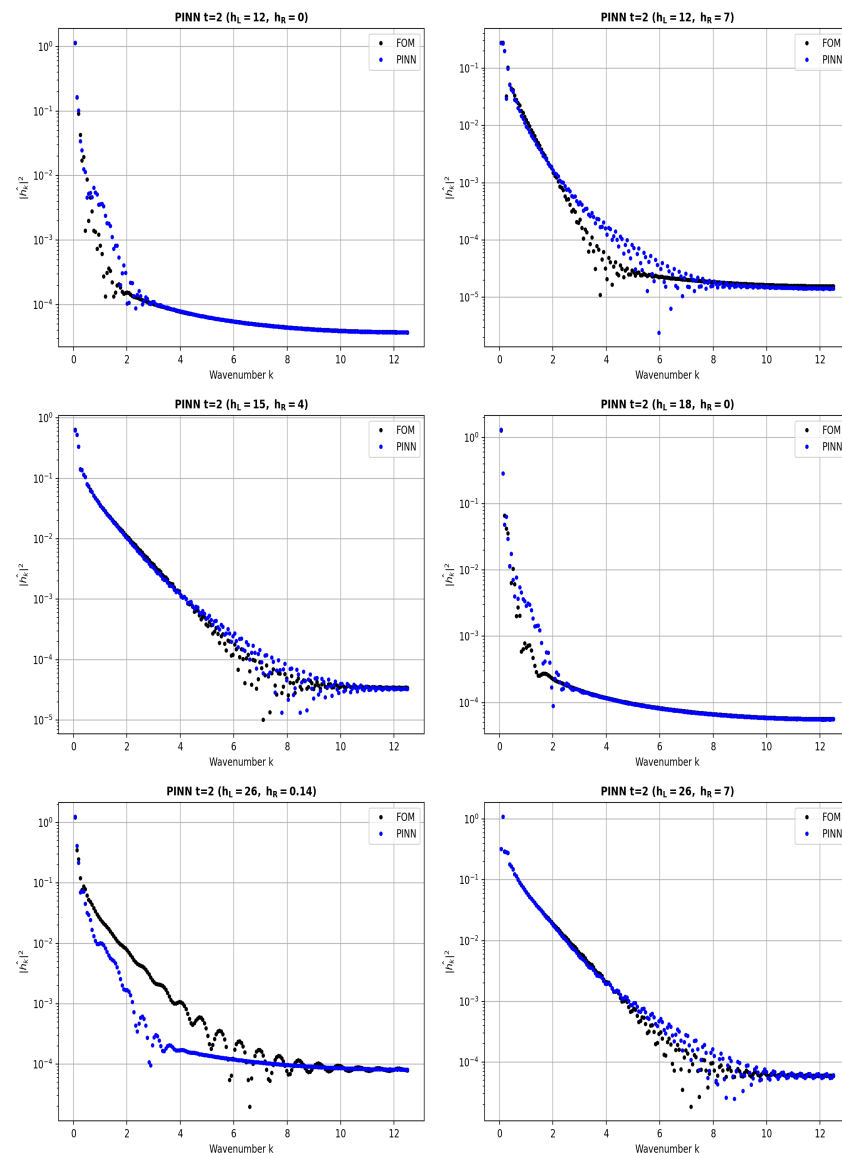


Figure 6. Fourier-amplitude spectra of the water-depth profiles at $t = 2$ for $\mu = (12, 0)$ (top left), $(12, 7)$ (top right), $(15, 4)$ (middle left), $(18, 0)$ (middle right), $(26, 0.14)$ (bottom left), $(26, 7)$ (bottom right).

Tables 1 and 2 indicate that the relative errors for both the water depth and the discharge remain of the same order in most cases, although some increase with time. In addition, the relative errors for the discharge are slightly larger and more regime-dependent than those for the water depth, reflecting the greater sensitivity of $q = hu$ to local mismatches near the shock front. Similar to the water-depth errors, the dry-bed and

near-dry regimes remain the most challenging, while the wet-bed cases achieve the lowest discharge errors.

Overall, these results demonstrate that the proposed parametric PINN is capable of learning the FOM parameter-to-solution map with high accuracy across a diverse range of dam-break configurations. Despite the challenges posed by hyperbolic dynamics and moving discontinuities, it achieves low relative errors and accurate shock localization, making it a robust and competitive surrogate model for parametrized shallow-water flows.

For the dam-break problem, a right-propagating shock forms when $h_R > 0$, whereas the dry-bed case is characterized by a moving rarefaction wave. Since the shock propagation speed increases as h_R decreases, the shock exits the computational domain by $t = 2.5$ for the near-dry-bed regime $\mu = (26, 0.14)$. Figure 7 shows the predictions of the PINN and the TROM for the near-dry-bed regime $\mu = (26, 0.14)$ at time $t = 2.5$. Although the training data cover the time interval up to $t = 2.5$, neither model accurately captures the FOM solution at time $t = 2.5$ near the right boundary in this regime. However, the discrepancies are localized near the right boundary, and both models accurately predict the solution over the rest of the computational domain.

Our results indicate that both models reproduce the FOM simulations with high accuracy for parameter regimes within the training domain. The errors of both the PINN and TROM are primarily localized near moving fronts, while the global wave structure is preserved. For the PINN, the spectral comparisons also show good agreement with the FOM over low and intermediate wavenumbers in most cases. The dry-bed and near-dry-bed configurations are the most challenging for both models, with the discharge being more sensitive than the water depth. In addition, both models face difficulties when predicting FOM solutions at times when the shock exits the computational domain.

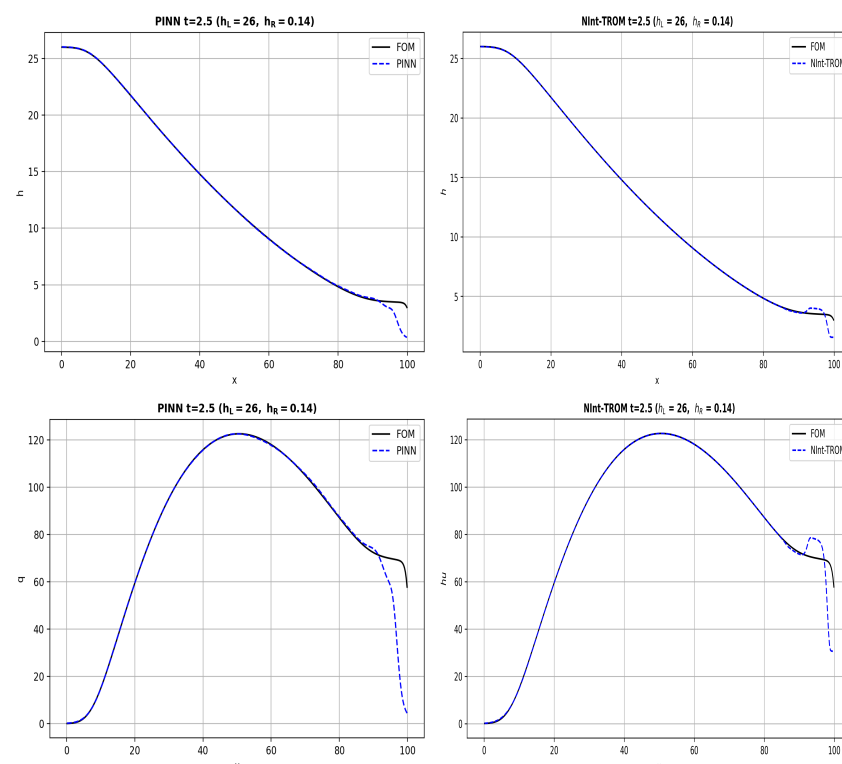


Figure 7. Comparison of the water-depth profiles $h(x, t)$ (top) and discharge profiles $q(x, t)$ (bottom) at $t = 2.5$ predicted by the PINN (left column) and TROM (right column) for $\mu = (26, 0.14)$.

Computational Time Comparison

Generation of the FOM training dataset over the 65 training parameter instances required 42.19 ± 4.3 s, with individual FOM run times ranging from 0.64 to 0.66 s.

For the PINN, shock detection and construction of the shock trajectories for the full training dataset required 0.049 ± 0.018 s. Thus, the shock-aware collocation strategy introduces negligible preprocessing cost and does not require any additional FOM simulations beyond those already used to construct the supervised training dataset.

Training the final PINN configuration for 20,000 epochs required 18.5 ± 3.0 h, including the validation evaluations performed during training. These timing results are reported as the mean and sample standard deviation over five independent training runs. Once trained, evaluation of the PINN for a single parameter–time query over the full spatial grid requires approximately 1.6×10^{-3} s. Hence, the dominant computational expense of the PINN is the offline neural-network training, whereas FOM dataset generation, shock-aware preprocessing, and subsequent online evaluations constitute comparatively small costs.

Construction of the Tucker decomposition of the resulting snapshot tensors required 2.5 s (0.7 s for the h -component, 1.8 s for the hu -component). Thus, FOM data generation accounts for approximately 94% of the offline cost, with the Tucker decomposition contributing the remaining 6%.

Once constructed, the TROM required approximately 6.3×10^{-4} s to reconstruct a solution for a single online query, compared to approximately 0.65 s for a corresponding FOM run, a speedup of roughly 1049 times. Therefore, the dominant computational cost of the TROM lies in the offline construction stage, primarily FOM data generation rather than the decomposition step, while online evaluation is negligible by comparison.

6. Extrapolation Experiments

To further assess the performance of the parametric PINN and the TROM, we consider the extrapolation benchmark, where we evaluate the accuracy of both models for parameter values outside of the training range $(h_L, h_R) \in [10, 28] \times [0, 8]$ for $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2$ with parameters for the training mesh $\mathcal{P}_1, \mathcal{P}_2$ in (5). This test highlights how well each model generalizes beyond the training regime. Specifically, we consider six parameter values $(h_L, h_R) = (9, 4), (20, 8.5), (29, 4), (32, 4), (29, 8.5),$ and $(32, 10)$ and predict the solution at time $t = 2$. Note that the value $h_R = 4$ is inside the training range for h_R , but the corresponding values for h_L are outside of $[10, 28]$.

Figures 8 and 9 compare profiles for $h(x, t)$ and Table 3 presents relative L^2 errors at the final time $t = 2$. Profiles for the discharge $q(x, t)$ follow a similar trend and are not presented here for brevity.

Table 3. Relative L^2 Errors in (21) for the TROM and the PINN parameter extrapolation predictions at time $t = 2$.

μ	ϵ_h^{TROM}	ϵ_h^{PINN}	ϵ_q^{TROM}	ϵ_q^{PINN}
(9,4)	1.28×10^{-2}	6.87×10^{-3}	6.73×10^{-2}	2.3×10^{-2}
(20,8.5)	7.25×10^{-3}	2.88×10^{-3}	3.36×10^{-2}	0.91×10^{-2}
(29,4)	1.97×10^{-2}	2.58×10^{-3}	6.35×10^{-2}	0.35×10^{-2}
(32,4)	6.79×10^{-2}	5.64×10^{-3}	2.06×10^{-1}	0.59×10^{-2}
(29,8.5)	1.37×10^{-2}	3.64×10^{-3}	4.76×10^{-2}	0.71×10^{-2}
(32,10)	4.92×10^{-2}	9.46×10^{-3}	1.65×10^{-1}	1.74×10^{-2}

Overall, the performance of both models in the extrapolation regime is good; both models reproduce the large-scale solution structure and correctly track the propagating shock. However, the parametric PINN provides more accurate predictions in the vicinity of

the shock and consistently yields lower relative errors than the TROM for all extrapolation cases considered here. In particular, for $h_L > 28$, the TROM prediction develops an artificial bump near the shock location, as seen in the middle and bottom rows of Figure 8 and the bottom row of Figure 9. This localized artifact becomes more pronounced as the extrapolation distance in h_L increases, whereas the PINN preserves a more accurate representation of the front.

These results indicate that both models are able to predict solutions for parameter values outside the training range, but the parametric PINN model provides more accurate extrapolations, particularly for larger values of h_L .

The spectral plots for the PINN model (not shown here for brevity) show good agreement with the FOM at low and intermediate wavenumbers, while the remaining mismatch is concentrated primarily in the high-wavenumber tail. Thus, even outside the training domain in parameter space, the learned PINN model continues to capture the dominant scales of the flow with good accuracy.

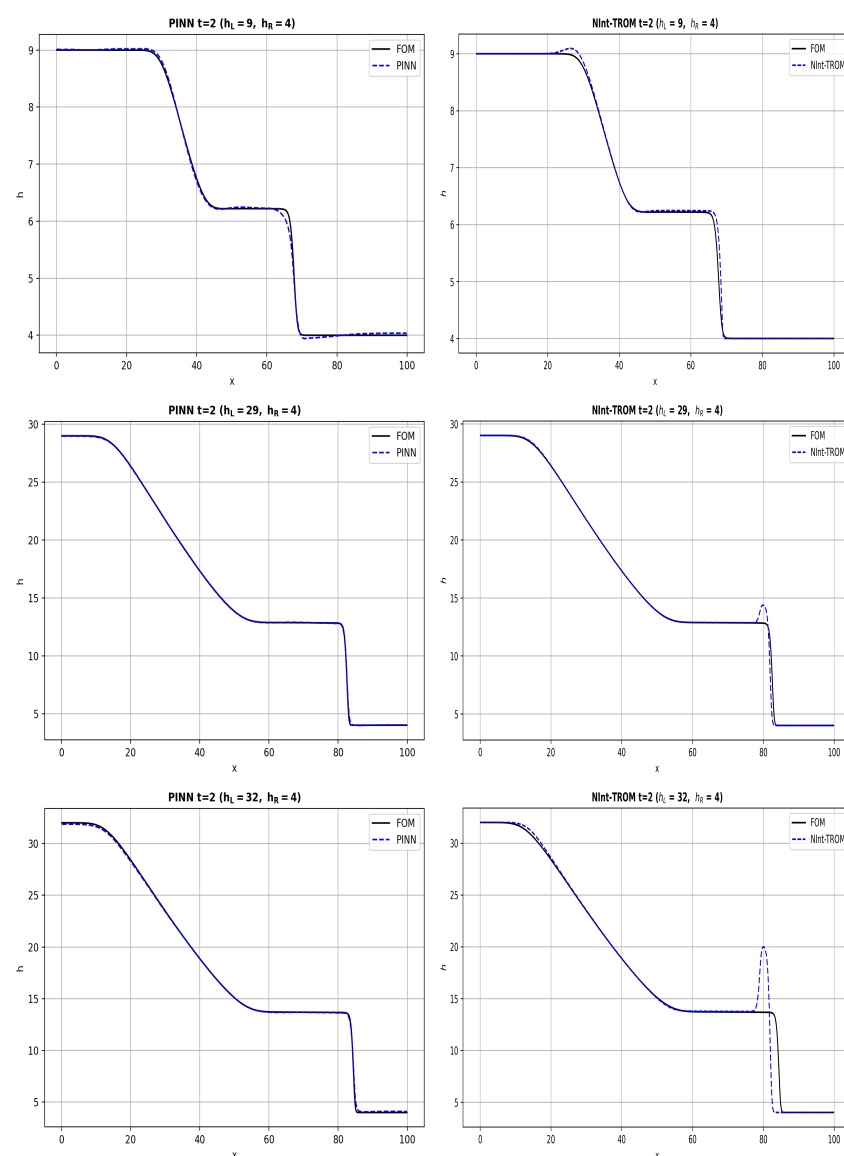


Figure 8. Comparison of $h(x, t; \mu)$ in FOM simulations and predictions of PINN (left column) and TROM (right column) for $\mu = (9, 4)$, $(29, 4)$, $(32, 4)$ (top to bottom) at time $t = 2$.

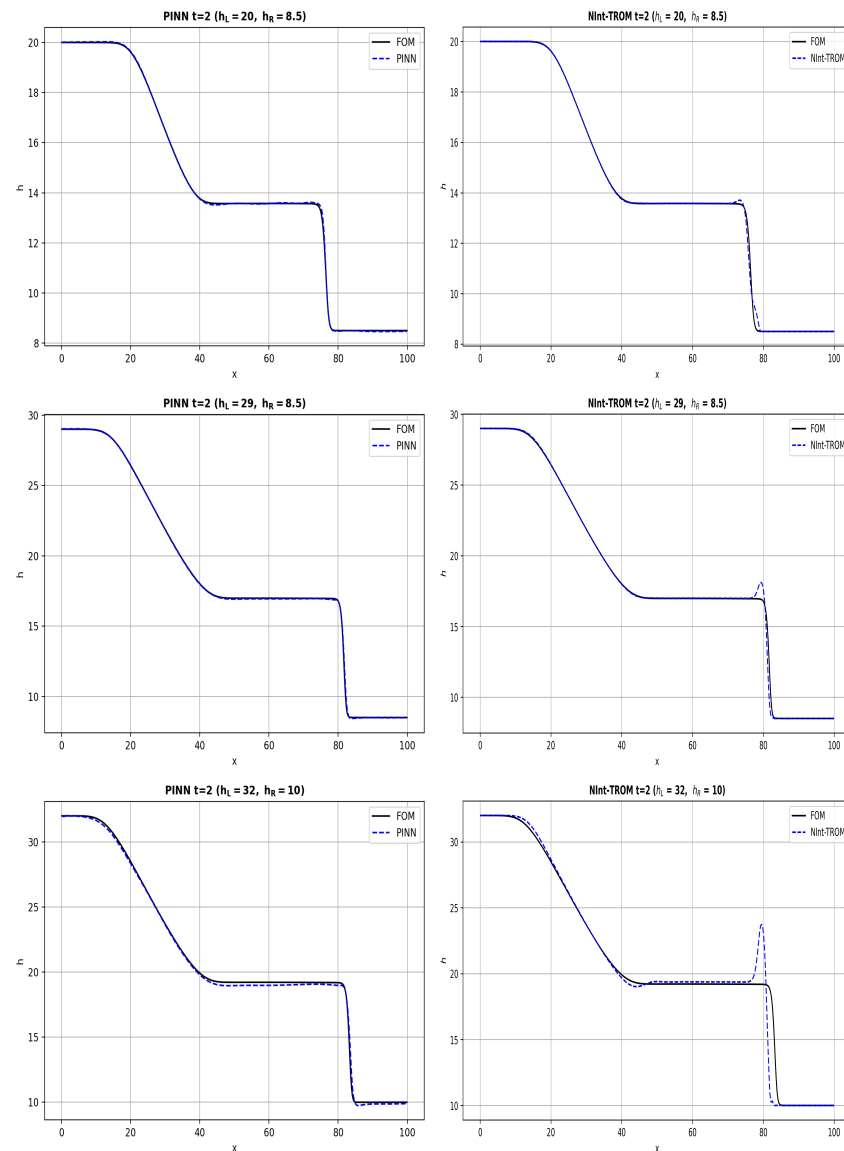


Figure 9. Comparison of $h(x, t; \mu)$ from the FOM simulations and predictions of the PINN (left column) and the TROM (right column) for $\mu = (20, 8.5), (29, 8.5), (32, 10)$ (top to bottom) at time $t = 2$.

7. Ablation Study

As discussed in Section 4, the design of the PINN model involves several nontrivial choices, some of which are specific to hyperbolic conservation laws. In this section, we perform a controlled, one-at-a-time ablation study around a fixed reference configuration to assess the effect of individual design choices. The goal is to identify which design decisions are crucial to the performance of the PINN model under the training and evaluation setup considered in this paper. As the reference configuration, we use the model described in Section 4, with two minor modifications introduced to reduce the training time: 15,000 training epochs and a longer 50% ramp-up period. All ablation networks were initialized with random seed 42.

Each ablation then turns off one parameter relative to this reference configuration, while the rest of the setup is kept unchanged.

Table 4 reports the mean space-time relative L^2 errors for h and q over a Monte Carlo benchmark consisting of $MC = 99$ parameter instances $\mu = (h_L, h_R)$. To construct the benchmark, we first discretize the parameter domain $\mathcal{P}$ defined in (4) using 20 values of h_L

and 25 values of h_R , resulting in a tensor-product set of 500 candidate parameter instances. We then sample MC instances uniformly without replacement from this set. The same sampled benchmark set is used to evaluate all ablation configurations. For a field $y \in \{h, q\}$, the space-time relative L^2 error corresponding to a parameter instance μ is defined as

$$E_{L^2}^y(\mu) = \frac{\left(\int_0^T \int_{\Omega} |y^{\text{pred}}(x, t; \mu) - y^{\text{ref}}(x, t; \mu)|^2 dx dt \right)^{1/2}}{\left(\int_0^T \int_{\Omega} |y^{\text{ref}}(x, t; \mu)|^2 dx dt \right)^{1/2}}.$$

The corresponding Monte Carlo means and the aggregate score are

$$\bar{E}_{L^2}^y = \frac{1}{MC} \sum_{i=1}^{MC} E_{L^2}^y(\mu_i), \quad y \in \{h, q\}, \quad S := \frac{1}{2} \left(\bar{E}_{L^2}^h + \bar{E}_{L^2}^q \right). \quad (23)$$

The aggregate errors above are used as a ranking statistic for the ablation study.

Table 4. Controlled ablation study around the reference PINN configuration. Each row modifies one component of the baseline while keeping the remaining setup fixed. Reported values are the aggregate errors defined in (23).

Variant	Change from Baseline	$\bar{E}_{L^2}^h$	$\bar{E}_{L^2}^q$	S
Baseline	reference configuration	1.358×10^{-2}	3.629×10^{-2}	2.494×10^{-2}
No PDE	$\lambda_{\text{pde}} = 0$	1.528×10^{-2}	3.904×10^{-2}	2.716×10^{-2}
Reduced q weight	$\lambda_u = 1$	1.306×10^{-2}	4.138×10^{-2}	2.722×10^{-2}
No ramp-up	$\lambda_{\text{pde}}, \lambda_{\text{bc}},$ and λ_{ic} are constant	1.264×10^{-2}	3.464×10^{-2}	2.364×10^{-2}
L^1 PDE loss	$L^2 \rightarrow L^1$ in PDE term	2.122×10^{-2}	4.052×10^{-2}	3.087×10^{-2}
Shock-aware off	uniform collocation	4.031×10^{-2}	1.165×10^{-1}	7.839×10^{-2}
PDE gate off	no residual gating	2.142×10^{-2}	5.253×10^{-2}	3.697×10^{-2}
No residual scaling	EMA scaling removed	2.603×10^{-2}	6.092×10^{-2}	4.347×10^{-2}

The dominant trend in Table 4 is that the largest degradation occurs when shock-aware collocation (see Equation (15)) is removed. Replacing the shock-aware strategy with uniform collocation increases the aggregate score from 2.494×10^{-2} to 7.839×10^{-2} , more than tripling the baseline value. The degradation is significant for both fields, but is especially pronounced for the discharge q , whose mean relative error rises from 3.629×10^{-2} to 1.165×10^{-1} . It shows that careful treatment of the moving front region is essential for obtaining good accuracy, especially for the discharge q .

The next most important components are PDE gating (see Equation (A5)) and residual scaling (see Equation (18)). Disabling the soft PDE gate increases the aggregate score to 3.697×10^{-2} , while removing EMA residual scaling yields 4.347×10^{-2} . Both changes degrade h and q , with a particularly clear effect on the discharge variable. Taken together, these results indicate that, once the collocation points are placed in a shock-aware manner, the way the strong-form residual is weighted and balanced remains critical. In other words, this ablation study shows that both (i) *where* (in space) the PDE is enforced and (ii) *how* it is enforced are important.

Removing the PDE loss altogether also worsens performance, increasing the aggregate score from 2.494×10^{-2} to 2.716×10^{-2} . This confirms that physics-informed supervision provides measurable benefit even in the presence of substantial paired training data. However, the increase in error is much smaller than for the shock-aware, gating, and residual-scaling ablations. This suggests that the main challenge for this shock-dominated benchmark is not the inclusion of a PDE loss in itself, but rather enforcing that PDE loss remains meaningful near shocks.

The discharge weighting parameter λ_u (see Equations (17) and (20)) also has a visible effect. Reducing it from 5 to 1 leaves the mean depth error nearly unchanged, but increases the mean discharge error from 3.629×10^{-2} to 4.138×10^{-2} . This is consistent with the observation made throughout the paper that predicting q is more difficult than predicting h , and that the discharge component benefits from additional emphasis during optimization.

By contrast, removing ramp-up produces only a small change in the aggregate score. However, it appears to act more as a stabilizing or preventive training choice, which may still be useful in practice even if its direct effect on the aggregate error is limited in this sweep.

Overall, the ablation study shows that the most important components of the present training design are those related to shock-aware collocation, PDE gating, and residual scaling. By comparison, removing the PDE term or changing the discharge weight leads to a more moderate degradation, while removing ramp-up has little effect on the reported aggregate metric. For this benchmark, the main gains therefore come from approaches that improve how the residual is enforced near sharp fronts.

8. Conclusions

Fast and accurate parametric reduced-order models are important for a broad range of practical applications, including inverse problems, shape optimization, and digital twins. Such applications often require efficient evaluation of the underlying physical system across many parameter configurations. Developing reduced models for unified representation of parametric solutions is a challenging task because the behavior of solutions can change drastically for different parameter values, even for parameters in the training dataset.

In this paper, we compared and contrasted two parametric reduced models—the PINN and the TROM. Both models were constructed using the same dataset generated by a FOM solver and evaluated at the same parameter values. In this work, the accuracy of the surrogate models is therefore understood as their fidelity to the common FOM parameter-to-solution map generated by the FOM LLF solver, rather than to the exact solution of the SWE. Consequently, both surrogate models might inherit some of the discretization characteristics of the underlying FOM data, including the numerical diffusion introduced near discontinuities. Neither model requires time integration during online evaluation, allowing solutions to be reconstructed rapidly; the PINN can reconstruct solutions at arbitrary times, whereas the TROM reconstructs solutions at the snapshot times stored in the training data. In our simulations, both models were more than two orders of magnitude (approximately 500 to 700 times) faster than the full-order model at $t = 2$. Moreover, the inference cost of either model is essentially independent of the queried time t , since neither approach uses time stepping to predict the solution. This substantial acceleration potentially allows the efficient treatment of realistic, computationally demanding problems.

Both the TROM and the PINN perform well in the interpolation cases considered here, producing small relative errors overall, with the largest discrepancies concentrated near the moving shock. The TROM is more accurate in the dry-bed regime. Both models perform comparably in the challenging near-dry-bed regime at $h_R = 0.14$, where they exhibit some loss of accuracy near the moving shock. In contrast, the PINN performs better in the wet-bed regimes with larger values of h_R considered here. In addition, the PINN spectra show good agreement with the FOM over a broad range of wavenumbers in most cases, with the largest discrepancies occurring at high wavenumbers in the near-dry-bed regime $h_R = 0.14$. Overall, the near-dry-bed regime with $h_R = 0.14$ remains the most challenging for both models because of the sharp, small-scale features associated with the propagating shock.

In addition, both models encounter difficulties when the shock exits the computational domain $[0, L]$, as occurs in the near-dry-bed regime $\mu = (26, 0.14)$ at $t = 2.5$. Although $t = 2.5$ is included in the training dataset, neither model accurately reproduces the solution profile near the right boundary $x = L$. Away from the right boundary, however, both the water depth and the discharge are predicted accurately. One possible strategy for mitigating this limitation is to initialize a short-time FOM simulation using the reduced-model prediction at an earlier time, thereby resolving the shock as it exits the computational domain.

A more demanding test of a reduced model is its ability to generalize to parameter regimes outside the range represented in the training data. Demonstrating predictive accuracy under such conditions extends the model's applicability for parameter values outside of the training range and increases its value for practical use. The PINN performs better in the extrapolation regimes considered here. This result is consistent with the construction of the TROM, which relies on interpolation (in parameter space) among nearby solutions in the training dataset to approximate the solution at a new parameter value. Because interpolation-based approximations can lose accuracy outside the sampled parameter domain, this limitation naturally carries over to the TROM when it is applied outside of the training range. While the PINN reproduces the solutions with high accuracy in all extrapolation cases, the TROM exhibits larger discrepancies, concentrated primarily near the shock front. If predictions outside the initial training range are required, the extrapolation performance of the TROM can be improved by augmenting the training set with full-order solutions at selected new parameter values and updating the low-rank representation.

To compare the offline stages of the two approaches, we first note that the TROM can efficiently handle large datasets generated on fine parameter grids. In contrast, PINN training is more computationally intensive and requires iterative batch processing of solution samples. Consequently, its training cost can grow substantially with the size of the dataset. In addition, the offline stage of the TROM is considerably faster than PINN training. As quantified in Section 5, the offline construction of the TROM is considerably less expensive than PINN training.

The offline phase of the TROM consists of computing a low-rank representation of the full snapshot tensor, and the singular-value threshold in (7) is the only parameter controlling rank selection. In contrast, the PINN has several tunable hyperparameters and requires shock tracking and shock-aware collocation during training. The complexity of the PINN construction described in this paper reflects the challenges inherent in approximating hyperbolic problems with shocks.

The ablation study further demonstrates that shock-aware collocation, PDE gating, and residual scaling are essential components of the proposed PINN framework. Removing any of these components substantially increases the prediction errors, whereas removing the PDE loss itself produces a smaller degradation. These results indicate that, for shock-dominated problems, where and how the strong-form residual is enforced may be more important than simply including it in the loss function.

One of the major advantages of PINNs is their ability to work with incomplete or irregularly sampled data in parameter space. In contrast, TROMs require a rectangular tensor-product grid in parameter space and therefore do not readily support local parameter grid refinement or irregular parameter domains. This restriction follows directly from the tensor representation of the full snapshot dataset. When data are missing, modern tensor-completion methods can be incorporated into a TROM framework; however, they increase computational complexity and introduce an additional source of approximation error. The requirement that TROMs use rectangular tensor-product grids can lead to very large datasets, particularly for high-resolution or high-dimensional parameter discretizations. PINNs are more flexible in this respect because they can be trained using solution snapshots

sampled nonuniformly or irregularly in parameter space. Incorporating new data into a PINN is relatively straightforward because an existing model can be efficiently updated through transfer learning or fine-tuning. In contrast, TROMs rely on rectangular tensor-product grids, so a single additional FOM solution generally cannot be incorporated directly. Expanding the parameter grid may require additional FOM simulations to complete the new tensor.

Both TROMs and PINNs can be extended to two- and three-dimensional problems. However, extending TROMs is relatively straightforward; see, for example, [27,28]. In contrast, extending PINNs is more challenging because of the higher computational cost of neural-network training and the possible need to track multidimensional shocks. These questions will be investigated in future work.

Both models favor a sufficiently regular dependence of the solution on the parameters. This dependence is explicit in the TROM, which interpolates between neighboring solutions in parameter space, and is also embedded in the PINN through its smooth neural-network representation. Near the dry-bed limit $h_R = 0$, however, the solution depends rapidly and non-smoothly on h_R : the wet-bed regime $h_R > 0$ contains a propagating shock, whereas the dry-bed regime $h_R = 0$ is dominated by a rarefaction wave. This loss of parametric regularity potentially explains why the near-dry-bed regime $h_R = 0.14$ is particularly challenging for both models.

Finally, the accuracy of both surrogate models in this study is assessed relative to the common FOM parameter-to-solution map generated by the Local Lax-Friedrichs solver. Consequently, both models may inherit some of the discretization characteristics of the reference data, including the numerical diffusion near discontinuities. This dependence is direct for the TROM, which is constructed entirely from the FOM snapshots. The PINN also uses these snapshots through its supervised loss, but additionally incorporates the governing SWE, initial conditions, and boundary conditions through the physics-informed loss. Thus, the PINN is not constrained solely by the numerical data. The same TROM and PINN formulations can, in principle, be applied to snapshot data generated by a higher-resolution shock-capturing solver, although the models would need to be reconstructed and, in the case of the PINN, retrained. Therefore, both surrogate models are evaluated against the numerical LLF solution of the FOM, which allows the surrogate approximation error to be assessed separately from the discretization error of the underlying full-order scheme.

Overall, the two reduced-order modeling frameworks considered here should be viewed as complementary rather than competing approaches. This study highlights the strengths and potential limitations of each approach when applied to hyperbolic problems. TROMs offer a fast and relatively straightforward construction when solution data are available on a structured parameter grid, whereas PINNs provide greater flexibility in handling irregularly sampled data and extrapolating beyond the training regime. These advantages come with different costs: TROMs rely on compressed tensor-product representations, while PINNs require substantially more computational effort and careful training to capture shock-dominated dynamics. The choice between the two approaches should therefore depend on the available data, the structure of the parameter domain, and the intended application. Combining elements of both frameworks may provide a promising direction for developing more accurate, flexible, and efficient reduced-order models for complex hyperbolic systems.

Author Contributions: Conceptualization, I.T.; investigation PINN, A.M.; investigation TROM, M.R.B.M.; software A.M. and M.R.B.M.; validation, A.M. and M.R.B.M.; writing—original draft preparation, A.M. and I.T.; writing—review and editing, A.M., M.R.B.M. and I.T.; visualization, A.M. and M.R.B.M.; supervision, I.T.; All authors have read and agreed to the published version of the manuscript.

Funding: M.R.B.M. was partially supported by the National Science Foundation (NSF) under the award DMS-2309197.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The numerical code and data supporting the parametric PINN results are publicly available at <https://github.com/Saizt/parametric-swe-pinn>, accessed in 28 July 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. PINN Design

Estimating shock position.

For each regime $\mu = (h_L, h_R)$ and time t , we estimate a dominant shock location $x_{\text{sh}}(t; \mu)$ from the reference FOM snapshots. Let

$$y(\cdot, t; \mu) \in \{h(\cdot, t; \mu), u(\cdot, t; \mu)\}$$

be the chosen indicator field on the uniform spatial grid $\{x_i\}_{i=1}^{N_x}$. We first apply a moving-average filter,

$$\tilde{y}_i := \frac{1}{m} \sum_{k=-(m-1)/2}^{(m-1)/2} y_{i+k},$$

where $m \geq 3$ is an odd window size. We then approximate the spatial derivative by finite differences:

$$(\partial_x \tilde{y})_i \approx \begin{cases} \frac{\tilde{y}_{i+1} - \tilde{y}_i}{\Delta x}, & i = 1, \\ \frac{\tilde{y}_{i+1} - \tilde{y}_{i-1}}{2\Delta x}, & 2 \leq i \leq N_x - 1, \\ \frac{\tilde{y}_i - \tilde{y}_{i-1}}{\Delta x}, & i = N_x, \end{cases}$$

where Δx is the spatial mesh size. The shock location is then defined as the grid point at which the gradient magnitude is maximal over an interior subset $\Omega_{\text{int}} \subset \Omega$, i.e.,

$$x_{\text{sh}}(t; \mu) \approx x_{i^*}, \quad i^* \in \arg \max_{x_i \in \Omega_{\text{int}}} |(\partial_x \tilde{y})_i|,$$

where Ω_{int} excludes a small boundary buffer near $x = 0$ and $x = L$ in order to avoid boundary artifacts.

This produces shock-position estimates at the discrete snapshot times stored in the FOM dataset. Since collocation times are sampled continuously in $[0, T]$, the shock location at an arbitrary time t is obtained by interpolating between the neighboring discrete shock-position estimates.

For shock-aware collocation, μ is sampled uniformly from the discrete set of 65 training parameter instances P , while t is sampled continuously from $[0, T]$. Hence, the shock trajectory is interpolated only in time; no interpolation in parameter space is performed. Only FOM snapshots corresponding to $\mu \in P$ are used to construct these trajectories, and no shock-location information from the validation or test parameter values enters the training procedure. For the final model, the moving-average window is $m = 7$, the outer 2% of the spatial domain is excluded from the shock search, and shock exclusion is disabled for $h_R < \epsilon_{\text{dry}} = 0.5$.

The shock trajectories are extracted from the same FOM snapshots used to construct the supervised PINN training data and therefore require no additional simulations; shock detection and temporal interpolation constitute only an additional preprocessing step.

PDE Residual Rescaling.

Recall PDE residual scaling in Equation (18). At training iteration n , we first compute the batch RMS values

$$r_{i,\text{rms}}^{(n)} := \left(\frac{1}{N_f} \sum_{j=1}^{N_f} \mathcal{R}_i(z_j^{(f)}; \theta^{(n)})^2 + \varepsilon_{\text{scale}} \right)^{1/2}, \quad (\text{A1})$$

and then update the running scales according to

$$s_i^{(n)} = \beta s_i^{(n-1)} + (1 - \beta) r_{i,\text{rms}}^{(n)}, \quad s_i^{(1)} := r_{i,\text{rms}}^{(1)}, \quad (\text{A2})$$

where $\beta \in (0, 1)$ is the EMA parameter and $\varepsilon_{\text{scale}} > 0$ is a small constant for numerical stability.

PDE residual gating.

For the j -th collocation point we define

$$w_j^{\text{grad}} = \exp\left(-\alpha |\partial_x h_\theta(x_j^{(f)}, t_j^{(f)}; \mu_j^{(f)})|\right), \quad (\text{A3})$$

and

$$w_j^{(h)} = \min\left\{1, \max\left\{0, \frac{h_\theta(x_j^{(f)}, t_j^{(f)}; \mu_j^{(f)}) - h_{\min}}{h_{\min}}\right\}\right\}, \quad (\text{A4})$$

where $\alpha > 0$ controls sensitivity to steep gradients and $h_{\min} > 0$ defines a small-depth threshold. Then, the final gate is

$$w_j = \max\{w_{\min}, w_j^{\text{grad}} w_j^{(h)}\}, \quad (\text{A5})$$

where $w_{\min} \in (0, 1)$ is a prescribed lower bound. Thus, PDE residual enforcement is weakened near sharp fronts and near-dry regions, while remaining nonzero everywhere else. The gating weights w_j are recomputed from the current network prediction at each loss evaluation, but are detached from the automatic-differentiation graph and therefore treated as fixed pointwise weights when differentiating the PDE loss.

References

1. Stoker, J.J. *Water Waves: The Mathematical Theory with Applications*; Dover Publications: Garden City, NY, USA, 2019.
2. Toro, E.F. *Computational Algorithms for Shallow Water Equations*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2024. [CrossRef]
3. Castro-Orgaz, O.; Chanson, H. Ritter's dry-bed dam-break flows: Positive and negative wave dynamics. *Environ. Fluid Mech.* **2017**, *17*, 665–694. [CrossRef]
4. LeVeque, R.J. *Finite Volume Methods for Hyperbolic Problems*; Cambridge University Press: Cambridge, UK, 2002.
5. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
6. Ohlberger, M.; Rave, S. Reduced Basis Methods: Success, Limitations and Future Challenges. *arXiv* **2015**, arXiv:1511.02021. [CrossRef]
7. Mojamder, M.A.H.; Xu, Z.; Wang, M.; Timofeyev, I. Parametrization of Subgrid Scales in Long-Term Simulations of the Shallow-Water Equations Using Machine Learning and Convex Limiting. *Fluids* **2026**, *11*, 76. [CrossRef]
8. Crisovan, R.; Torlo, D.; Abgrall, R.; Tokareva, S. Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification. *J. Comput. Appl. Math.* **2019**, *348*, 466–489. [CrossRef]
9. Chan, J. Entropy stable reduced order modeling of nonlinear conservation laws. *J. Comput. Phys.* **2020**, *423*, 109789. [CrossRef]
10. Benner, P.; Gugercin, S.; Willcox, K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **2015**, *57*, 483–531. [CrossRef]
11. Melenk, J.M. On n -widths for elliptic problems. *J. Math. Anal. Appl.* **2000**, *247*, 272–289. [CrossRef]

12. Lassila, T.; Manzoni, A.; Quarteroni, A.; Rozza, G. Generalized Reduced Basis Methods and N-Width Estimates for the Approximation of the Solution Manifold of Parametric PDEs. In *Analysis and Numerics of Partial Differential Equations*; Bonito, A., Nochetto, R.H., Quarteroni, A., Saleri, F., Eds.; Springer: Milan, Italy, 2013; pp. 307–329. [\[CrossRef\]](#)
13. Cohen, A.; DeVore, R.; Schwab, C. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.* **2010**, *10*, 615–646. [\[CrossRef\]](#)
14. Buffa, A.; Maday, Y.; Patera, A.T.; Prud'homme, C.; Turinici, G. A priori convergence of the Greedy algorithm for the parametrized reduced basis method. *ESAIM Math. Model. Numer. Anal.* **2012**, *46*, 595–603. [\[CrossRef\]](#)
15. Cohen, A.; DeVore, R. Kolmogorov widths under holomorphic mappings. *IMA J. Numer. Anal.* **2016**, *36*, 1–12. [\[CrossRef\]](#)
16. Hesthaven, J.S.; Rozza, G.; Stamm, B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*; Springer: Berlin/Heidelberg, Germany, 2016. [\[CrossRef\]](#)
17. Bachmayr, M.; Cohen, A. Kolmogorov widths and low-rank approximations of parametric elliptic PDEs. *Math. Comput.* **2017**, *86*, 701–724. [\[CrossRef\]](#)
18. Binev, P.; Cohen, A.; Dahmen, W.; DeVore, R.; Petrova, G.; Wojtaszczyk, P. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.* **2011**, *43*, 1457–1472. [\[CrossRef\]](#)
19. Barnett, J.; Farhat, C. Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction. *J. Comput. Phys.* **2022**, *464*, 111348. [\[CrossRef\]](#)
20. Amsallem, D.; Zahr, M.J.; Farhat, C. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Numer. Methods Eng.* **2012**, *92*, 891–916. [\[CrossRef\]](#)
21. Reiss, J.; Schulze, P.; Sesterhenn, J.; Mehrmann, V. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM J. Sci. Comput.* **2018**, *40*, A1322–A1344. [\[CrossRef\]](#)
22. Taddei, T. A registration method for model order reduction: Data compression and geometry reduction. *SIAM J. Sci. Comput.* **2020**, *42*, A997–A1027. Erratum in *SIAM J. Sci. Comput.* **2026**, *48*, A1839–A1842. [\[CrossRef\]](#)
23. Rim, D.; Peherstorfer, B.; Mandli, K.T. Manifold Approximations via Transported Subspaces: Model Reduction for Transport-Dominated Problems. *SIAM J. Sci. Comput.* **2023**, *45*, A170–A199. [\[CrossRef\]](#)
24. Mamonov, A.V.; Olshanskii, M.A. Interpolatory tensorial reduced order models for parametric dynamical systems. *Comput. Methods Appl. Mech. Eng.* **2022**, *397*, 115122. [\[CrossRef\]](#)
25. Mamonov, A.V.; Olshanskii, M.A. Tensorial parametric model order reduction of nonlinear dynamical systems. *SIAM J. Sci. Comput.* **2024**, *46*, A1850–A1878. [\[CrossRef\]](#)
26. Olshanskii, M.A.; Rebholz, L.G. Approximating a branch of solutions to the Navier–Stokes equations by reduced-order modeling. *J. Comput. Phys.* **2025**, *524*, 113728. [\[CrossRef\]](#)
27. Mizan, M.R.B.; Olshanskii, M.A.; Timofeyev, I. A parametric tensor ROM for the shallow water dam break problem. *Comput. Fluids* **2026**, *309*, 107005. [\[CrossRef\]](#)
28. Mizan, M.R.B.; Timofeyev, I.; Olshanskii, M.A. Reduced-Order Modeling of Parameterized Visco-Plastic Shallow Flows. *arXiv* **2026**, arXiv:2605.06526. [\[CrossRef\]](#)
29. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
30. Jagtap, A.D.; Kharazmi, E.; Karniadakis, G.E. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* **2020**, *365*, 113028. [\[CrossRef\]](#)
31. De Ryck, T.; Mishra, S.; Molinaro, R. wPINNs: Weak Physics Informed Neural Networks for Approximating Entropy Solutions of Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.* **2024**, *62*, 811–841. [\[CrossRef\]](#)
32. Chaumet, A.; Giesselmann, J. Improving Weak PINNs for Hyperbolic Conservation Laws: Dual Norm Computation, Boundary Conditions and Systems. *SMAI J. Comput. Math.* **2024**, *10*, 373–401. [\[CrossRef\]](#)
33. Rajvanshi, M.P.; Ketcheson, D.I. Integral PINNs for Hyperbolic Conservation Laws. In Proceedings of the ICLR 2024 Workshop on AI4DifferentialEquations in Science, Vienna, Austria, 11 May 2024.
34. Kharazmi, E.; Zhang, Z.; Karniadakis, G.E. Variational Physics-Informed Neural Networks For Solving Partial Differential Equations. *arXiv* **2019**, arXiv:1912.00873. [\[CrossRef\]](#)
35. Kharazmi, E.; Zhang, Z.; Karniadakis, G.E. hp-VPINNs: Variational Physics-Informed Neural Networks with Domain Decomposition. *Comput. Methods Appl. Mech. Eng.* **2021**, *374*, 113547. [\[CrossRef\]](#)
36. Rodriguez-Torrado, R.; Ruiz, P.; Cueto-Felgueroso, L.; Green, M.C.; Friesen, T.; Matringe, S.; Togelius, J. Physics-informed attention-based neural network for hyperbolic partial differential equations: Application to the Buckley–Leverett problem. *Sci. Rep.* **2022**, *12*, 7557. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Bihlo, A.; Popovych, R.O. Physics-informed neural networks for the shallow-water equations on the sphere. *J. Comput. Phys.* **2022**, *456*, 111024. [\[CrossRef\]](#)
38. Qi, X.; de Almeida, G.A.M.; Maldonado, S. Physics-informed neural networks for solving flow problems modeled by the 2D Shallow Water Equations without labeled data. *J. Hydrol.* **2024**, *636*, 131263. [\[CrossRef\]](#)

39. Mumtaz, K.; Nadeem, M.W.; Khan, A.; Lakdawala, Z. Investigating the use of physics informed neural networks for dam-break scenarios. *PLoS ONE* **2025**, *20*, e0332694. [[CrossRef](#)] [[PubMed](#)]
40. Chang, T.J.; Kao, H.M.; Chang, K.H.; Hsu, M.H. Numerical simulation of shallow-water dam break flows in open channels using smoothed particle hydrodynamics. *J. Hydrol.* **2011**, *408*, 78–90. [[CrossRef](#)]
41. Magdalena, I.; Pebriansyah, M.F.E. Numerical treatment of finite difference method for solving dam break model on a wet-dry bed with an obstacle. *Results Eng.* **2022**, *14*, 100382. [[CrossRef](#)]
42. Muchiri, D.K.; Hewett, J.N.; Sellier, M.; Moyers-Gonzalez, M.; Monnier, J. Numerical simulations of dam-break flows of viscoplastic fluids via shallow water equations. *Theor. Comput. Fluid Dyn.* **2024**, *38*, 557–581. [[CrossRef](#)]
43. Peng, S.H. 1D and 2D numerical modeling for solving dam-break flow problems using finite volume method. *J. Appl. Math.* **2012**, *2012*, 489269. [[CrossRef](#)]
44. Zokagoa, J.M.; Soulaïmani, A. A POD-based reduced-order model for free surface shallow water flows over real bathymetries for Monte-Carlo-type applications. *Comput. Methods Appl. Mech. Eng.* **2012**, *221–222*, 63–82. [[CrossRef](#)]
45. Zoppou, C.; Roberts, S. Numerical solution of the two-dimensional unsteady dam break. *Appl. Math. Model.* **2000**, *24*, 457–475. [[CrossRef](#)]
46. Shu, C.W.; Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **1988**, *77*, 439–471. [[CrossRef](#)]
47. Gottlieb, S.; Shu, C.W.; Tadmor, E. Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **2001**, *43*, 89–112. [[CrossRef](#)]
48. Gottlieb, S.; Ketcheson, D.; Shu, C.W. *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*; World Scientific: Singapore, 2011.
49. Bin Mizan, M.R. Parametric Low-Rank Model Reduction for Nonlinear Fluid Systems. Ph.D. Dissertation, University of Houston, Houston, TX, USA, 2026. Available online: <https://hdl.handle.net/10657/21905> (accessed on 1 October 2026).
50. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311. [[CrossRef](#)] [[PubMed](#)]
51. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]
52. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1253–1278. [[CrossRef](#)]
53. Myshak, A. Machine Learning for Data-Driven and Simulated Problems of Fluid Dynamics. Ph.D. Dissertation, University of Houston, Houston, TX, USA, 2026. Available online: <https://hdl.handle.net/10657/21501> (accessed on 1 October 2026).
54. Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; Volume 9, pp. 249–256.
55. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980. [[CrossRef](#)]
56. Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the spectral bias of neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5301–5310.
57. Zhi-Qin, J.; Xu, J.; Tao, L.; Yanyang, X.; Zheng, M. Frequency principle: Fourier analysis sheds light on deep neural networks. *Commun. Comput. Phys.* **2020**, *28*, 1746–1767. [[CrossRef](#)]
58. Xu, Z.Q.J.; Zhang, Y.; Luo, T. Overview frequency principle/spectral bias in deep learning. *Commun. Appl. Math. Comput.* **2025**, *7*, 827–864. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.