Journal of Information and Computing Science
J. Info. Comput. Sci., Vol. 20, No. 1, pp. 37-60

DOI: 10.4208/JICS-2025-003
June 2025

Physical Informed Neural Network for Solving Conservation Laws Based on Relaxation Systems

Muyu Zuo[1,*]

[1] *School of Mathematics, Nanjing University of Aeronautics and Astronautics, 29 Jiangjun Avenue, Nanjing 211106, China.*

Abstract. Solving partial differential equations (PDEs) with discontinuous solutions, such as shock waves in multiphase viscous flows through porous media, is critical for a wide range of scientific and engineering applications. These discontinuities represent sudden changes in physical quantities. In recent years, physics-informed neural networks (PINNs) have emerged as a promising method for solving PDEs but face significant challenges when modeling such problems. Specifically, neural networks struggle to compute gradients accurately near shock waves, leading to solutions that deviate from true physical phenomena. To address this issue, we propose a novel relaxation neural network method based on the conservation law relaxation model and its improved version, the relaxation limit neural network method. These two approaches employ auxiliary neural networks to approximate flux functions, enhancing the vanilla PINN framework's capability to simulate shock waves. The proposed methods retain the simplicity and extensibility of vanilla PINNs while avoiding the need for spatiotemporal discretization. Numerical experiments for one-dimensional and two-dimensional problems demonstrate the effectiveness of our approaches. The results show that the improved methods significantly outperform vanilla PINN in capturing shock wave dynamics.

AMS subject classifications: 35L65, 65M99

Key words: Physics-informed neural networks, Partial differential equations, Conservation laws, Relaxation systems.

1 Introduction

Discontinuities in solutions to partial differential equations (PDEs) are ubiquitous in scientific and engineering applications. They arise from sudden changes in system properties, such as velocity, pressure, or phase volume fraction, and typically manifest as shock

*Corresponding author. *Email addresses:* zuomuyu@nuaa.edu.cn (M. Zuo)

waves, contact discontinuities, or spikes. These phenomena occur in diverse contexts, including high-speed aerodynamics [1], astrophysical explosions [2], molecular transport in materials [3, 4], and multiphase flows through porous media [5, 6]. In porous media, discontinuities result from complex interactions among viscous, capillary, and inertial forces, abruptly altering fluid properties. Such behavior challenges traditional modeling and simulation approaches [7].

Mathematically, discontinuities in the solutions of PDEs typically arise from nonlinearities in the governing equations. For instance, in multiphase flow through porous media, the governing equations are derived from mass conservation laws and Darcy's law [8], which relates the volumetric flow rate to fluid viscosity, permeability, and pressure gradients. When viscous forces dominate such as under high-pressure gradients or in coarse grained porous materials—shock waves may develop. Due to the nonsmooth nature of these shocks, traditional discretization methods often introduce nonphysical oscillations or numerical diffusion, making such problems challenging to solve numerically. Conventional numerical techniques for PDEs, including finite difference (FD), finite element (FE), and finite volume (FV) methods, have been widely employed to model these systems [9]. However, accurately capturing discontinuities requires specialized shock-capturing schemes, such as total variation diminishing (TVD) methods, weighted essentially non-oscillatory (WENO) schemes, or adaptive mesh refinement (AMR). These approaches stabilize numerical solutions near discontinuities while preserving high-order accuracy in smooth regions. Despite decades of progress, balancing computational efficiency, accuracy, and robustness remains challenging, especially for complex multiphysics problems.

PINNs have recently emerged as a promising alternative for solving PDEs, utilizing the universal approximation capability of neural networks to directly encode physical laws into loss functions [10]. PINNs offer several advantages, including mesh-free discretization, seamless handling of high-dimensional problems, and the ability to incorporate experimental or observational data. However, vanilla PINNs struggle to resolve discontinuities effectively, often leading to inaccurate or oscillatory solutions near shocks [11, 12]. This limitation stems from the smoothness of neural network representations and the difficulty of optimizing the loss landscape in the presence of sharp gradients. De Ryck and Mishra [13] demonstrated that the estimation error of physics-informed machine learning methods can be significantly affected by the stability of the underlying PDEs (their sensitivity to perturbations). To address these challenges, several strategies have been proposed, including adaptive weighting of loss terms [14], domain decomposition [15], and artificial viscosity [16]. While each method provides unique advantages for addressing discontinuities, they still prove inadequate for complex problems like multiphase flow in porous media. Recent developments include the control volume PINN proposed by Patel et al. [17], which incorporates TVD conditions, and the weak PINN introduced by De Ryck et al. [18] for shock wave capture. Chaumet and Giesselmann [19] further developed a more efficient weak PINN scheme with entropy admissibility conditions to ensure solution uniqueness. However, these methods typi-

cally require either numerical derivatives or prior knowledge of shock positions, effectively reintroducing grid dependence to the inherently mesh-free PINN framework and substantially increasing computational costs.

To accurately capture shock waves while preserving the simplicity of PINNs without requiring spatio-temporal discretization, we propose two novel approaches based on Jin and Xin's relaxation system [20]: the Relaxation physics-informed neural network (RPINN) incorporating a small parameter ε and its simplified version, the Relaxation Limit physics-informed neural network (RLPINN) without ε for multi-dimensional conservation laws. We evaluate these methods on one-dimensional and two-dimensional problems involving three common conservation law equations. Through comparison with high-precision reference solutions from clawpack [21] and vanilla PINN results, our experiments demonstrate that both RPINN and RLPINN maintain strong generalization capabilities while achieving high accuracy for conservation laws.

The paper is organized as follows. Section 2 introduces the weak solutions of multi-dimensional conservation laws and the fundamentals of PINNs. Section 3 presents the relaxation system and relaxation limit system for conservation laws, along with our corresponding neural network improvements. Section 4 details our numerical results for both one-dimensional and two-dimensional conservation law problems, including the neural network architectures and implementation specifics. Finally, Section 5 summarizes our methodology and key findings.

2 Physics-Informed Neural Networks for Multi-Dimensional Systems of Conservation Laws

This section provides a brief introduction to multi-dimensional systems of conservation laws and the definition of their weak solutions. Following this, we present a concise review of how PINNs approach the solution of conservation laws.

2.1 Multi-Dimensional Systems of Conservation Laws and Weak Solutions

Multi-dimensional conservation laws are systems of PDEs that govern the conservation of physical quantities across multiple spatial dimensions. These equations find widespread application in fluid mechanics, gas dynamics, electromagnetics, and related fields. Their general form can be expressed as:

$$\partial_t u+\sum_{i=1}^{m}\partial_{x_i}F_i(u)=0,(t,x_i)\in\mathbb{R}_+\times\mathbb{R},u\in\mathbb{R}^n, \tag{2.1}$$

where each $F_i:\mathbb{R}^n\to\mathbb{R}^n$ is assumed to be a smooth function$(i=1,2,\cdots,m)$, usually called a flux function. For linear hyperbolic systems, the solution is easy to obtain, because it can be seen as a linear combination of the right vectors of each point in spacetime. Physically,

it is a superposition of waves propagating at different speeds. The situation becomes considerably more complex for nonlinear hyperbolic systems, where shock waves develop at characteristic line intersections. At these singularities, the conservation law (2.1) fails to hold in the classical sense, necessitating recourse to the fundamental integral form of the conservation law

$$\frac{d}{dt}\int_{\Omega} u(t,\mathbf{x})d\mathbf{x} = \oint_{\partial\Omega}(F_1(u),\cdots,F_m(u))\cdot\mathfrak{n}dl \tag{2.2}$$

for any region Ω in $\mathbb{R}^m$. Here, $\mathfrak{n}$ is the out of unit normal vector of region Ω. Furthermore, for the shock wave problems, if the solution u satisfies

$$\int_0^{\infty}\int_{\Omega}[u\Phi_t+\sum_{i=1}^{m}F_i(u)\Phi_{x_i}]d\mathbf{x}dt=0 \tag{2.3}$$

for all functions $\Phi\in C_0^1$, We call u the weak solution of the conservation law (2.1).

2.2 PINNs in Solving Conservation Laws

We consider the conservation law problem (2.1) with prescribed initial and boundary conditions

$$u(0,\mathbf{x})=g(\mathbf{x}),\mathbf{x}\in\Omega, \tag{2.4}$$

$$\mathcal{B}(u,t,\mathbf{x})=0,(t,\mathbf{x})\in[0,T]\times\Omega. \tag{2.5}$$

PINNs employ a neural network architecture denoted by $\hat{u}(\theta;t,\mathbf{x})$ to approximate the solution u of the conservation law system (2.1). The network parameters θ are optimized to minimize the following loss function:

$$\mathcal{L}(\theta;\mathcal{T})=w_f\mathcal{L}_f(\theta;\mathcal{T}_f)+w_{ic}\mathcal{L}_{ic}(\theta;\mathcal{T}_{ic})+w_{bc}\mathcal{L}_{bc}(\theta;\mathcal{T}_{bc}), \tag{2.6}$$

where

$$\mathcal{L}_f(\theta;\mathcal{T}_f)=\frac{1}{|\mathcal{T}_f|}\sum_{(t,\mathbf{x})\in\mathcal{T}_f}\left|\partial_t\hat{u}+\sum_{i=1}^{m}\partial_{x_i}F_i(\hat{u})\right|^2,$$

$$\mathcal{L}_{ic}(\theta;\mathcal{T}_{ic})=\frac{1}{|\mathcal{T}_{ic}|}\sum_{\mathbf{x}\in\Omega}|\hat{u}(\theta;0,\mathbf{x})-g(\mathbf{x})|^2,$$

$$\mathcal{L}_{bc}(\theta;\mathcal{T}_{bc})=\frac{1}{|\mathcal{T}_{bc}|}\sum_{(t,\mathbf{x})\in[0,T]\times\partial\Omega}|\mathcal{B}(\hat{u},t,\mathbf{x})|^2,$$

$\mathbf{x}=(x_1,x_2,\cdots,x_m)$, and w_f,w_{ic},w_{bc} are the weights. The three sets of points are samples inside the domain($\mathcal{T}_f$) and on the initial domain($\mathcal{T}_{ic}$) and on the boundary domain($\mathcal{T}_{bc}$). Here, $\mathcal{T}_f$, $\mathcal{T}_{ic}$ and $\mathcal{T}_{bc}$ are referred to as the sets of "residual points", and $\mathcal{T}=\mathcal{T}_f\cup\mathcal{T}_{ic}\cup\mathcal{T}_{bc}$.

3 The RPINN Method and RLPINN Method

In this section, we introduce the relaxation system of conservation law and its simplified version of relaxation limit system, and introduced the corresponding RPINN and RLPINN.

3.1 Relaxation System of Conservation Laws

This section primarily presents the relaxation system formulation for multi-dimensional conservation laws. The one-dimensional case can be directly derived from the multidimensional theory by setting the spatial dimension to 1.

For the conservation law system (2.1), we introduce an associated linear system with a stiff lower-order term (henceforth referred to as the relaxation system):

$$\begin{cases} \partial_t u + \sum_{i=1}^{m} \partial_{x_i} v_i = 0, \quad v_i \in \mathbb{R}_n, \\ \partial_t v_i + A_i \partial_{x_i} u = -\frac{1}{\varepsilon}(v_i - F_i(u)), i = 1,2,\cdots,m. \end{cases}, \tag{3.1}$$

where

$$A_i = a_i I, \quad a_i > 0, \quad i = 1,2,\cdots,m, \tag{3.2}$$

is a positive diagonal matrix to be chosen. In this way the relaxation system (3.1) is non-degenerate. For small ε, applying the ChapmanEnskog expansion in the relaxation system (3.1), one can derive the following approximation for u as [22,23]

$$\partial_t u + \sum_{i=1}^{m} \partial_{x_i} F_i(u) = \varepsilon \sum_{i,j=1}^{m} \partial_{x_i} \left(\left[\delta_{ij} A_i - F_i'(u) F_j'(u) \right] \partial_{x_j} u \right), \tag{3.3}$$

where δ_{ij} is the Kronecker delta. It is clear that one form of the dissipative condition for (3.3) is

$$\delta_{ij} A_i - F_i'(u) F_j'(u) \geqq 0 \quad on \quad \mathbb{R}^{mn}, \tag{3.4}$$

note that the left-hand side of (3.4) is an $mn \times mn$ matrix. The condition (3.4) is satisfied for u in a bounded domain by choosing a sufficiently large A_i. In numerical computation, because of the courant-friedrichs-levy (CFL) condition [24] on numerical stability, it is desirable to obtain the smallest A_i meeting the criterion (3.4). However, when employing continuous-time PINNs to solve conservation laws, the absence of spatiotemporal discretization eliminates the need to consider CFL condition. Consequently, We only need to select A_i that satisfies (3.4). Furthermore, we reformulate (3.1) as follows:

$$\begin{cases} \partial_t u + \sum_{i=1}^{m} \partial_{x_i} v_i = 0, \quad v_i \in \mathbb{R}_n \\ \varepsilon(\partial_t v_i + A_i \partial_{x_i} u) + (v_i - F_i(u)) = 0, i = 1,2,\cdots,m. \end{cases}, \tag{3.5}$$

let $\varepsilon=0$, we can obtain the following simplified system (hereafter called the relaxation limit system)

$$\begin{cases} \partial_t u+\sum_{i=1}^{m}\partial_{x_i}v_i=0, \quad v_i\in\mathbb{R}_n \\ v_i=F_i(u), i=1,2,\cdots,m. \end{cases}, \tag{3.6}$$

this simplified system is more concise and avoids the need to select the small parameter ε or the matrix A_i.

3.2 The RPINN Method and The RLPINN Method

For the conservation law system (3.1) with initial condition (2.4) and boundary condition (2.5), we implement the RPINN method through the relaxation system (3.5) using $m+1$ fully connected neural networks. The first network $\hat{u}_1(\theta_1;t,\mathbf{x})$, approximates the solution u of the conservation law, while the subsequent networks $\hat{u}_{i+1}(\theta_{i+1};t,\mathbf{x})$ approximate the flux functions $F_i(u)(i=1,2,\cdots,m)$. The network parameters $\theta_1,\theta_2,\cdots,\theta_m,\theta_{m+1}$ are trained to approximate the solution u and flux functions $F_1(u,F_2(u),\cdots,F_m(u))$, such that the loss function is minimized:

$$\begin{aligned}\mathcal{L}(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T})=&w_f\mathcal{L}_f(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)+w_r\mathcal{L}_r(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)\\&+w_{ic}\mathcal{L}_{ic}(\theta_1;\mathcal{T}_{ic})+w_{bc}\mathcal{L}_{bc}(\theta_1;\mathcal{T}_{bc}),\end{aligned} \tag{3.7}$$

where

$$\mathcal{L}_f(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)=\frac{1}{|\mathcal{T}_f|}\sum_{(t,\mathbf{x})\in\mathcal{T}_f}\left|\partial_t\hat{u}_1+\sum_{i=1}^{m}\partial_{x_i}\hat{u}_{i+1}\right|^2,$$

$$\mathcal{L}_r(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)=\frac{1}{|\mathcal{T}_f|}\sum_{(t,\mathbf{x})\in\mathcal{T}_f}\sum_{i=1}^{m}\left|\varepsilon(\partial_t\hat{u}_{i+1}+A_i\partial_{x_i}\hat{u}_1)+(\hat{u}_{i+1}-F_i(\hat{u}_1))\right|^2,$$

$$\mathcal{L}_{ic}(\theta_1;\mathcal{T}_{ic})=\frac{1}{|\mathcal{T}_{ic}|}\sum_{\mathbf{x}\in\Omega}|\hat{u}_1(\theta_1;0,\mathbf{x})-g(\mathbf{x})|^2,$$

$$\mathcal{L}_{bc}(\theta_1;\mathcal{T}_{bc})=\frac{1}{|\mathcal{T}_{bc}|}\sum_{(t,\mathbf{x})\in[0,T]\times\partial\Omega}|\mathcal{B}(\hat{u}_1,t,\mathbf{x})|^2,$$

and w_f,w_r,w_{ic} and w_{bc} are the weights. The three sets of points are samples inside the domain($\mathcal{T}_f$) and on the initial domain($\mathcal{T}_{ic}$) and on the boundary domain($\mathcal{T}_{bc}$). Here, $\mathcal{T}_f$, $\mathcal{T}_{ic}$ and $\mathcal{T}_{bc}$ are referred to as the sets of "residual points", and $\mathcal{T}=\mathcal{T}_f\cup\mathcal{T}_{ic}\cup\mathcal{T}_{bc}$.

The RLPINN method is derived from the relaxation limit system (3.6) with ε=0. Compared to RPINN, on the one hand, the system is much simplified, and on the other hand, we do not need to consider the selection of ε and A_i parameters during network initialization. The difference between its loss function and RPINN lies in $\mathcal{L}_r(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)$.

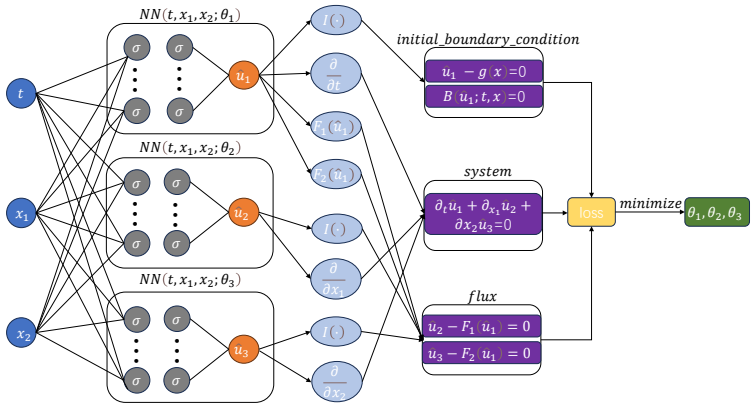

Figure 1: The framework of our proposed RLPINN model

The $\mathcal{L}_r(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)$ of RLPINN is represented as follows:

$$\mathcal{L}_r(\theta_1,\theta_2,\cdots,\theta_{m+1};\mathcal{T}_f)=\frac{1}{|\mathcal{T}_f|}\sum_{(t,\mathbf{x})\in\mathcal{T}_f}\sum_{i=1}^{m}|\hat{u}_{i+1}-F_i(\hat{u}_1)|^2. \tag{3.8}$$

To better illustrate the fundamental solving principle of RLPINN, we present its corresponding two-dimensional network architecture in Figure 1. The architecture of RPINN follows an identical structure and therefore is not separately depicted. The only distinction between the two methods resides in the flux component representation within Figure 1.

However, the design of this unique architecture is not conjured up out of thin air. Its concept is rooted in a profound understanding of the mathematical structure of hyperbolic conservation law systems. The core lies in systematically reconstructing the topological connection of the computational graph through a physically guided dimensional decoupling strategy. The innovativeness of this architecture is reflected in two complementary levels: the analytical decoupling of computational paths and the specialized division of network functions, which together constitute a complete technical system for solving the problems of gradient explosion and shock wave capture.

At the computational graph topology level, traditional PINNs employ a densely connected architecture with a single network, where gradient backpropagation must traverse a highly nonlinear function composition path. Specifically, when computing the conservation law residual $\partial_t u+\sum_{i=1}^{m}\partial_{x_i}F_i(u)$, gradients need to propagate through the entire path $u\to F_i(u)\to\partial_{x_i}F_i(u)$ via the chain rule. In this process, the flux Jacobian matrix [25] $J_{F_i}=\partial F_i/\partial u$ and the spatial gradient of the solution $\partial_{x_i}u$ form a hazardous product term $J_{F_i}\cdot\partial_{x_i}u$ in shock regions. Mathematically, this term corresponds to the derivative amplification effect along characteristic directions, and numerically, it manifests as a squared

growth in the condition number $\kappa \sim O(h^{-2})$, directly leading to an exponential explosion in gradient magnitude.

The multi-network architecture achieves a strategic deconstruction of the computational graph by introducing auxiliary flux variables v_i (where $i=1,\ldots,m$). Within this new architecture, the conservation laws are recast into the form $\partial_t u+\sum_{i=1}^{m}\partial_{x_i} v_i=0$, while physical constraints are established via the relaxation relations $v_i = F_i(u)$ or $\varepsilon(\partial_t v_i + A_i \partial_{x_i} u) + v_i = F_i(u)$. This reformulation decomposes the original monolithic computational graph into $m+1$ parallel sub-paths: one primary variable path $u \to \partial_t u$ and d distinct flux paths $v_i \to \partial_{x_i} v_i$. Backpropagation through each sub-path now involves only first-order derivative calculations, completely circumventing the hazardous product term $J_{F_i} \cdot \partial_{x_i} u$. From a numerical linear algebra perspective, this decomposition reduces the problem's condition number from $O(h^{-2})$ to $O(h^{-1})$, achieving a fundamental improvement in numerical stability.

At the level of network functional specialization, the multi-network architecture assigns each sub-network a distinct physical role and an optimized function space. The primary network $\hat{u}_1(\theta_1;t,\mathbf{x})$ is dedicated to approximating the distribution of the conserved variable u, and its design can be specifically optimized for the characteristics of discontinuous solutions: the configuration of its depth and width can focus on capturing the steep gradients of shock fronts, while the selection of activation functions can prioritize their numerical behavior near discontinuities (for instance, the saturation property of the Tanh function helps form stable shock profiles). The flux networks $\hat{u}_{i+1}(\theta_{i+1};t,\mathbf{x})$ specialize in learning the mapping relationships of the flux components $F_i(u)$, which are often physically endowed with better regularity than the original solution, even if u itself contains discontinuities, certain flux components may remain continuous or Lipschitz continuous. This functional separation allows each network to perform approximations within a specific function space, thereby avoiding the inherent conflict in a single network attempting to simultaneously fit both smooth regions and discontinuous regions.

In the specific implementation of shock capturing, the multi-network architecture exhibits a synergistic enhancement effect when combined with adaptive sampling strategies [26]. Based on the gradient information $\partial_{x_i}\hat{u}_1(\theta_1;t,\mathbf{x})$ from the primary network output, shock regions can be dynamically identified, and the density of training points in these areas can be increased. Thanks to the decoupling of the computational graph, such resampling only needs to target the physical quantities relevant to the primary network, without affecting the training stability of the flux networks, thereby achieving precise allocation of computational resources.

The mathematical essence of this architecture lies in transforming the PDE solving problem into a multi-objective optimization problem, where each sub-objective corresponds to a specific physical constraint, and each sub-network specializes in satisfying a particular type of constraint. Through this cooperative division-of-labor mechanism, both RPINN and RLPINN not only achieve orders-of-magnitude improvement in numerical stability but also reach a level of accuracy and resolution in shock capturing that is difficult to attain with traditional PINNs. The comparison of simplified processes be-

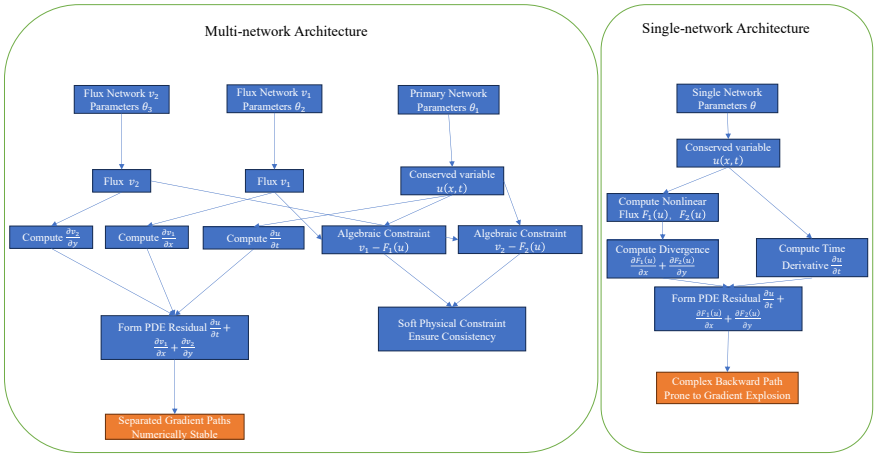

Figure 2: Comparison of Simple Processes between Multi-network Architecture and Single-network Architecture for Solving Two-dimensional Problems.

tween multi network architecture and single network architecture is placed in Figure2.

4 Numerical Results

To demonstrate the effectiveness of our proposed method, we will use RPINN, RLPINN, and PINN to solve one-dimensional and two-dimensional problems related to three common hyperbolic systems: the Burgers' system [27], the Shallow Water system [28], and the Euler system [29]. We will compare the results with the high-precision solutions provided by clawpack. For all examples, the hyperbolic tangent [30] (tanh) is selected as the activation function. Table 1 shows the network architectures used by RPINN and RLPINN for all examples. Table 2 presents the weight settings of RPINN and RLPINN for the loss functions corresponding to these examples. Table 3 outlines the sampling point settings for the relevant regions of all examples. To evaluate the accuracy of the solution $\hat{u}_1$, we use the L2 relative error [31]:

$$\frac{\|\hat{u}_1 - u\|_2}{\|u\|_2},$$

the L2 errors of the numerical solutions obtained by the three methods are presented in Table4 and Table 5.

4.1 One-dimensional Conservation Laws Problems

4.1.1 1-D Burgers' System

The inviscid Burger' system, a fundamental model in fluid dynamics and nonlinear wave theory, holds significant mathematical and physical importance. We consider the one-

Table 1: The network parameter setting for our RPINN and RLPINN on different systems. Here, d_{in},d_{out},N_{depth},N_{width}represents the input dimension, output dimension, depth, and width of the specified neural network's hidden layers. $\backslash$ means it does not exist.

Examples	$\hat{u}_1$				$\hat{u}_2$				$\hat{u}_3$			
	d_{in}	N_{depth}	N_{width}	d_{out}	d_{in}	N_{depth}	N_{width}	d_{out}	d_{in}	N_{depth}	N_{width}	d_{out}
1-D Burgers' System	2	4	20	1	2	4	20	1	\	\	\	\
1-D Shallow Water System	2	6	20	2	2	6	20	2	\	\	\	\
1-D Euler System	2	6	20	3	2	6	20	3	\	\	\	\
2-D Burgers' System	3	4	40	1	3	4	40	1	3	4	40	1
2-D Shallow Water System	3	4	40	3	3	4	40	3	3	4	40	3
2-D Euler System	3	4	40	4	3	4	40	4	3	4	40	4

Table 2: The hyperparameters of our RPINN and RLPINN on different systems.

Examples	w_f	w_r	w_{ic}	w_{bc}
1-D Burgers' System	2	3	6	1
1-D Shallow Water System	2	3	6	1
1-D Euler System	1	5	5	1
2-D Burgers' System	2	5	10	1
2-D Shallow Water System	2	5	10	1
2-D Euler System	5	5	10	1

Table 3: Number of sampling points in the corresponding domains on different systems.

Examples	$\lvert\mathcal{T}_f\rvert$	$\lvert\mathcal{T}_{ic}\rvert$	$\lvert\mathcal{T}_{bc}\rvert$
1-D Burgers' System	12000	600	800
1-D Shallow Water System	20000	800	800
1-D Euler System	16000	800	400
2-D Burgers' System	40000	10000	32000
2-D Shallow Water System	40000	10000	32000
2-D Euler System	32000	10000	32000

dimensional Burgers' sine problem defined as follows:

$$\begin{cases} \partial_t u + \partial_x(\frac{1}{2}u^2) = 0, (t,x) \in [0,1] \times [-1,1], \\ u(0,x) = -sin(\pi x), \\ u(-1,t) = u(1,t) = 0, \end{cases} \tag{4.1}$$

In this benchmark problem, we define u as the flow velocity and employ a two-phase optimization strategy to train the neural networks. The training process begins with

Table 4: L2 relative error of the solutions (Three methods to solve three one-dimensional conservation law systems), the bold numbers indicate the best results.

Methods	1-D Burgers' System	1-D Shallow Water System		1-D Euler System		
	u	h	u	ρ	u	p
PINN	0.38269	0.02006	0.12660	0.03106	0.18594	0.03648
RPINN	**0.02741**	**0.00508**	**0.03025**	**0.00657**	**0.01775**	**0.00446**
RLPINN	0.03662	0.00530	0.03037	0.00873	0.03299	0.00920

Table 5: L2 relative error of the solutions (Three methods to solve three two-dimensional conservation law systems), the bold numbers indicate the best results.

Methods	2-D Burgers' System	2-D Shallow Water System	2-D Euler System
	u	h	ρ
PINN	0.18097	0.02182	0.31990
RPINN	0.10303	0.01881	0.12254
RLPINN	**0.07673**	**0.01804**	**0.10727**

initial optimization using the Adam optimizer [32] for 5,000 iterations, followed by a refinement phase utilizing the L-BFGS [33] optimizer for an additional 10,000 iterations. Upon completion of the training procedure, we obtain the predicted solution field $\hat{u}_1$. To ensure a rigorous comparison across methodologies, we maintain identical hyperparameter configurations for the RPINN, RLPINN, and PINN implementations throughout all numerical experiments. Figure 3 illustrates the results from the three methods for solving the one-dimensional Burgers' system, including high-precision reference results by clawpack and the corresponding absolute error distribution.

4.1.2 1-D Shallow Water System

The shallow water equation is a classic mathematical model that describes shallow water flow and is widely used in simulating river dynamics, ocean tides, flood routing, and dam break problems. The problem of dam break refers to the rapid flow of water under the action of gravity after the sudden collapse of a dam, involving strong nonlinear wave dynamics and complex boundary conditions. Here we consider a simple one-dimensional dam break problem:

$$\begin{cases} \partial_t h + \partial_{x_1}(hu) = 0, \\ \partial_t(hu) + \partial_{x_1}(hu^2 + \frac{1}{2}gh^2) = 0, \end{cases} \tag{4.2}$$

with initial and boundary conditions:

$$(h(0,x), u(0,x)) = \begin{cases} (1,0)\, if\, x \in [-1.5,0] \\ (0.5,0)\, if\, x \in [-0,1.5] \end{cases},$$

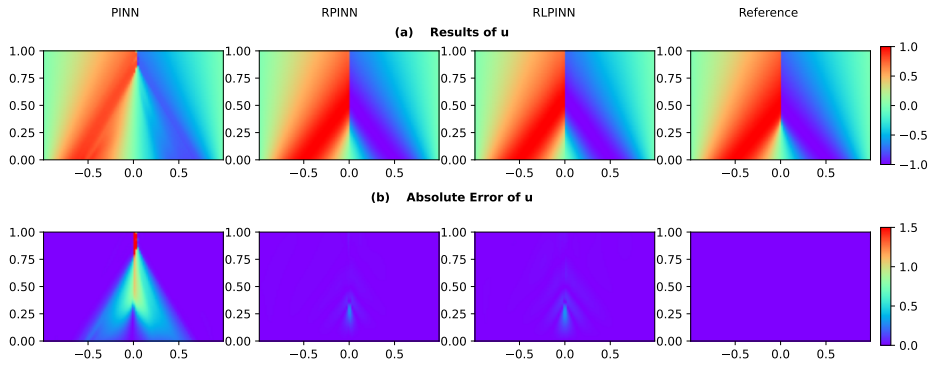

Figure 3: A comparative analysis of the one-dimensional Burgers equation solutions is presented, contrasting the performances of PINN, RPINN, and RLBINN against a high-fidelity reference. The first row displays the solution profiles from each method (successive columns: PINN, RPINN, RLBINN) alongside the reference solution. The second row presents the corresponding absolute error distributions, with the color gradient (purple to red) representing the spectrum from low to high error values.

$$(h(t,x),u(t,x))=\begin{cases}(1,0)\,if\,(t,x)\in[0,1]\times\{x=-1.5\}\\(0.5,0)\,if\,(t,x)\in[0,1]\times\{x=1.5\}\end{cases}.$$

In this formulation, h denotes the fluid depth, u represents the flow velocity, and g is the gravitational acceleration. For consistent comparison with clawpack reference solutions, we normalize the gravitational constant to $g=1$. The neural network training protocol consists of two phases: initial optimization using the Adam optimizer for 5,000 iterations and fine-tuning with L-BFGS optimizer for 15,000 iterations. Figure 4 shows the results of three methods for solving the one-dimensional Shallow Water system, the high-precision reference results provided by clawpack, and the corresponding absolute error distribution.

4.1.3 1-D Euler System

The Euler equations of gas dynamics constitute a system of hyperbolic partial differential equations governing the motion of an inviscid (non-viscous) fluid. These equations are derived from the fundamental principles of conservation of mass, momentum, and energy, and are widely employed in the fields of fluid dynamics and aerodynamics to model the behavior of gases. It is often used to judge the accuracy of numerical methods. Its one-dimensional form is expressed as follows:

$$\begin{cases}\partial_t\rho+\partial_x(\rho u)=0\\\partial_t(\rho u)+\partial_x(\rho u^2+p)=0\\\partial_t E+\partial_x(u(E+p))=0\\E=\frac{P}{\gamma-1}+\frac{1}{2}\rho u^2\end{cases},\tag{4.3}$$

we consider the shock tube problem, with corresponding initial and boundary conditions of

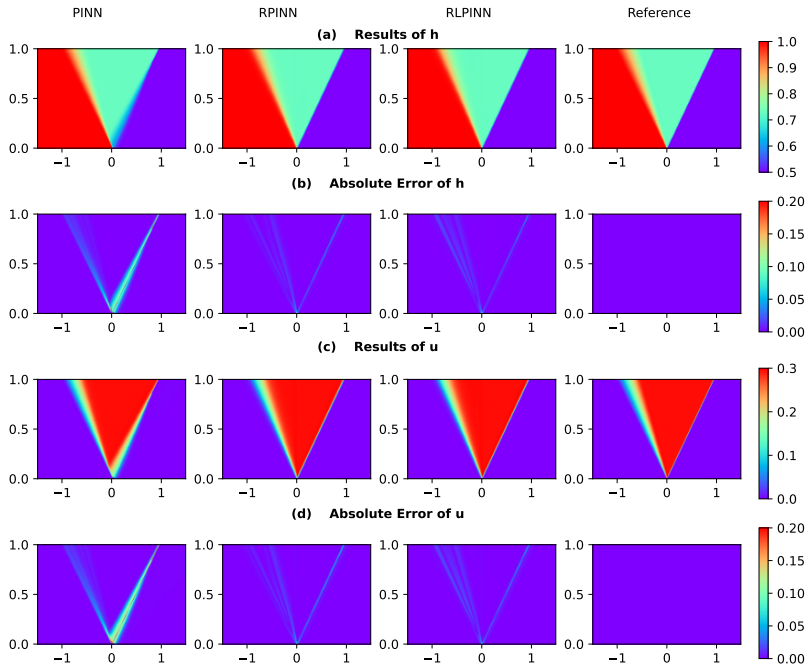

Figure 4: This figure presents a comprehensive comparison of the solutions for the one-dimensional Shallow Water equation, detailing both variables h and u. The first row illustrates the solution profiles for h obtained from the PINN, RPINN, and RLBINN methods alongside the high-fidelity reference. Directly below, the second row displays the corresponding absolute error distributions for h, where the color gradient from purple to red signifies the transition from low to high error values. The third row then presents the solution profiles for the variable u using the same methodological order, followed by the absolute error distributions for u in the fourth row, which employs an identical color mapping to quantify the error magnitude.

$$(\rho(0,x),u(0,x),p(0,x)) = \begin{cases} (3,0,3)\, if\, x \in [-1,0] \\ (1,0,1)\, if\, x \in [0,1] \end{cases},$$

$$(\rho(t,x),u(t,x),p(t,x)) = \begin{cases} (3,0,3)\, if\, (t,x) \in [0,0.4] \times \{x=-1\} \\ (1,0,1)\, if\, (t,x) \in [0,0.4] \times \{x=1\} \end{cases}.$$

Here, ρ represents the density of the fluid, u represents the velocity of the fluid, and p represents the pressure of the fluid. The parameter γ is set to 1.4 to indicate an ideal gas environment. During the training process, we first employ the Adam optimizer for 5,000 iterations, followed by the L-BFGS optimizer for an additional 25,000 iterations. Figure 5 shows the results of three methods for solving the one-dimensional Euler system, the high-precision reference results provided by clawpack, and the corresponding absolute error distribution.

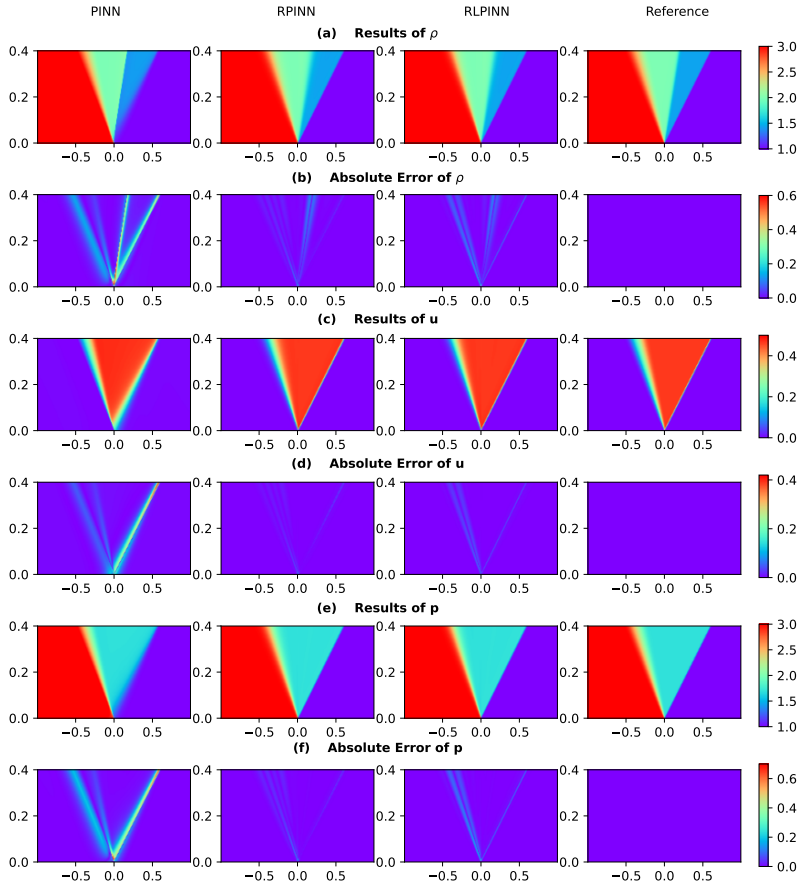

Figure 5: This figure presents a comprehensive comparison of the solutions for the one-dimensional Euler equations, detailing the three primary variables ρ, u, and p. The first row illustrates the solution profiles for ρ obtained from the PINN, RPINN, and RLBINN methods alongside the high-fidelity reference solution. Immediately below, the second row displays the corresponding absolute error distributions for ρ, employing a color gradient from purple to red to represent the spectrum from minimal to substantial error values. The third row proceeds to present the solution profiles for the velocity variable u following the same methodological sequence. Subsequent to this, the fourth row exhibits the absolute error distributions for u, maintaining the identical color convention to quantify discrepancies. Concluding the analysis, the fifth row showcases the solution profiles for the momentum variable p, while the sixth and final row provides the corresponding absolute error distributions for p, consistently applying the established color mapping to visualize error magnitudes across all presented methods.

4.2 Two-dimensional Conservation Laws Problems

The extension from one-dimensional to two-dimensional shock wave solving represents a substantial challenge, with difficulties increasing not merely incrementally, but explosively. A primary source of this increased complexity lies in the drastic change in geometric structure. In one-dimensional problems, the solution varies only along a single line. Shock waves, contact discontinuities, and other such structures manifest merely as points

along this line. In contrast, two-dimensional problems require solutions that vary across an entire plane. Here, a shock wave is no longer a point, but rather a curve. This curve may take on arbitrary shapes—straight lines, curves, or even closed contours—and its geometry and position can evolve in highly complex ways over time, including twisting, merging, and splitting. This article considers three two-dimensional numerical examples from easy to difficult.

4.2.1 2-D Burgers' System

For the two-dimensional inviscid Burgers' system, the computational region we consider is $\{(t,x_1,x_2)\} = [0,0.4]\times[0,1]\times[0,1]$. The Corresponding system expression and initial conditions are as follows:

$$\partial_t u + \partial_{x_1}(\frac{1}{2}u^2) + \partial_{x_2}(\frac{1}{2}u^2) = 0, \tag{4.4}$$

$$u(0,x_1,x_2) = \begin{cases} 1, if\ (x_1,x_2) \in [0,0.5]\times[0,0.5] \\ 0, \quad otherwise \end{cases}.$$

The boundary conditions need only satisfy basic compatibility requirements. However, the computational complexity increases significantly when extending from one-dimensional to two-dimensional problems, necessitating proportional increases in domain sampling points, network neurons, and iteration counts.During the training process, we first employ the Adam optimizer for 5,000 iterations, followed by the L-BFGS optimizer for an additional 15,000 iterations. Figure 6 shows the results of three methods for solving the two-dimensional Burgers' system at t=0.1 and t=0.3, the high-precision reference results provided by clawpack, and the corresponding absolute error distribution.

4.2.2 2-D Shallow Water System

Two-dimensional radial dam break is a classic benchmark test problem in computational fluid dynamics [34–36](CFD), used to simulate the fluid diffusion process after the collapse of a circular or cylindrical symmetric structure. Its core feature is that the fluid propagates radially from the center point to the surrounding areas on a two-dimensional plane, forming symmetrical swells and shock waves. We consider a radial dam break problem with a limited domain $\{(t,x_1,x_2)\} = [0,0.4]\times[-1.5,1.5]\times[-1.5,1.5]$. The specific expression and corresponding initial and boundary conditions are as follows:

$$\begin{cases} \partial_t h + \partial_{x_1}(hu) + \partial_{x_2}(hv) = 0, \\ \partial_t(hu) + \partial_{x_1}(hu^2 + \frac{1}{2}gh^2) + \partial_{x_2}(huv) = 0,, \\ \partial_t(hv) + \partial_{x_1}(huv) + \partial_{x_2}(hv^2 + \frac{1}{2}gh^2) = 0. \end{cases} \tag{4.5}$$

$$(h,u,v) = \begin{cases} (2,0,0), if\ x_1^2 + x_2^2 \le 0.25, \\ (1,0,0), \quad otherwise. \end{cases},$$

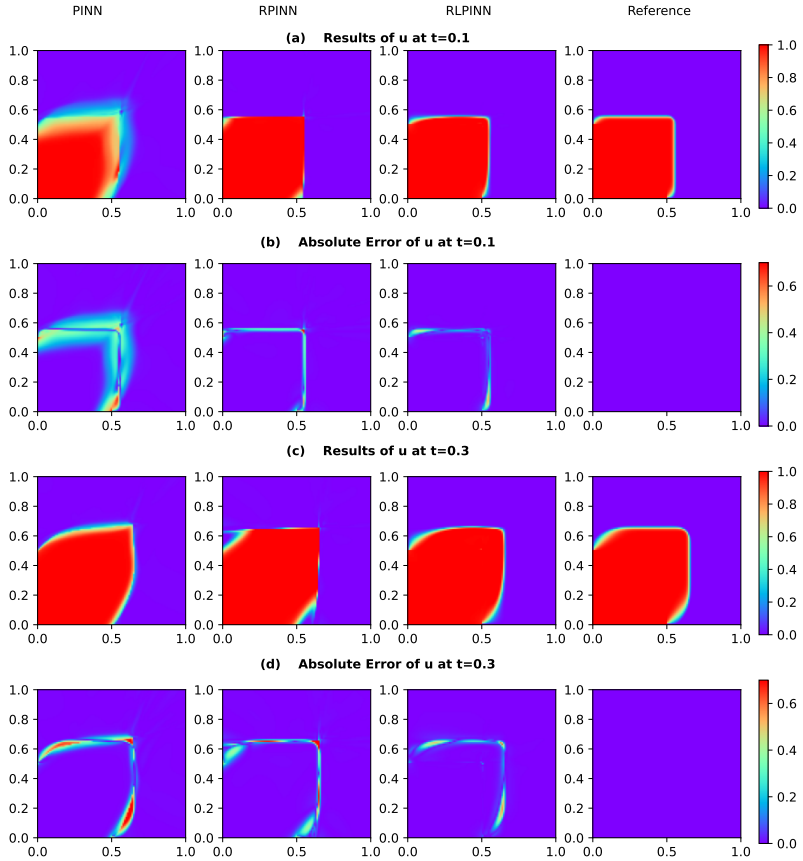

Figure 6: This figure presents a comprehensive comparison of the solutions for the two-dimensional Burgers equation, focusing on the temporal evolution of the variable u. The first row illustrates the solution profiles for u at time $t=0.1$ obtained from the PINN, RPINN, and RLBINN methods alongside the high-fidelity reference. Directly below, the second row displays the corresponding absolute error distributions for u at $t=0.1$, where the color gradient from purple to red signifies the transition from low to high error values. The third row then presents the solution profiles for u at time $t=0.3$ using the same methodological order, followed by the absolute error distributions for u at $t=0.3$ in the fourth row, which employs an identical color mapping to quantify the error magnitude.

$$(h,u,v)=(1,0,0),(t,x_1,x_2)\in[0,0.4]\times\partial\Omega.$$

Here, h represents water depth, u represents velocity in the x_1 axis direction, v represents velocity in the x_2 axis direction, and g is the gravitational constant (set to 1 here). We trained neural networks to solve this problem, with 5000 iterations of the first stage Adam optimizer and 35000 iterations of the second stage L-BFGS optimizer, for a total of 40000 iterations. Due to too many result images, this article only displays the results of variable h. Figure 7 shows the results of three methods for solving the two-dimensional Shallow Water system variable h at t=0.1 and t=0.3, the high-precision reference results provided by clawpack, and the corresponding absolute error distribution.

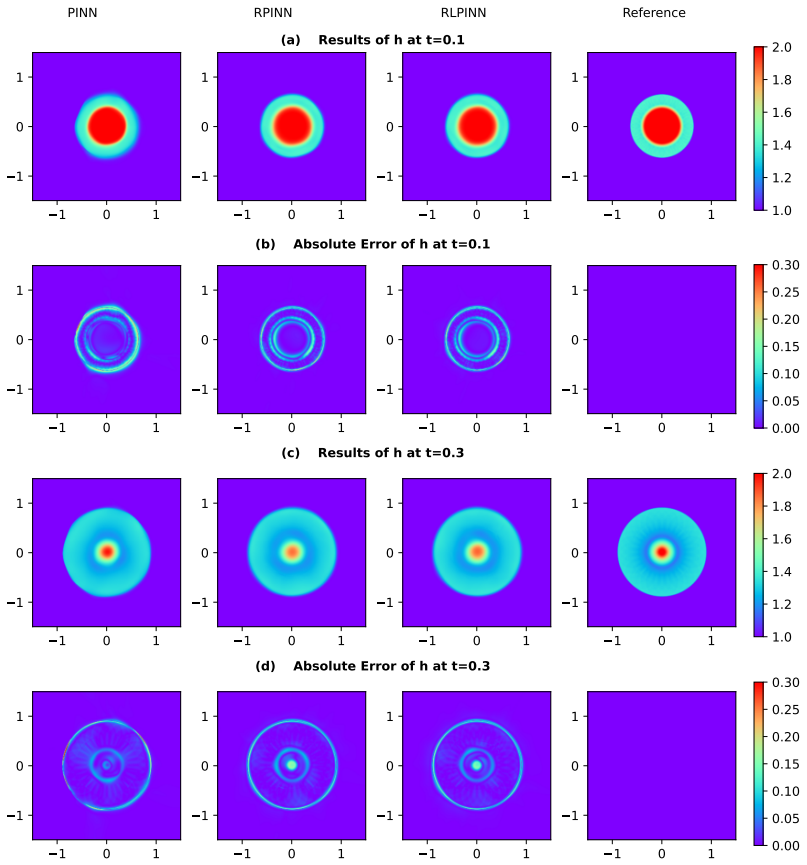

Figure 7: This figure presents a comprehensive comparison of the solutions for the two-dimensional Shallow Water equation, focusing on the temporal evolution of the variable h. The first row illustrates the solution profiles for h at time $t=0.1$ obtained from the PINN, RPINN, and RLBINN methods alongside the high-fidelity reference. Directly below, the second row displays the corresponding absolute error distributions for h at $t=0.1$, where the color gradient from purple to red signifies the transition from low to high error values. The third row then presents the solution profiles for h at time $t=0.3$ using the same methodological order, followed by the absolute error distributions for h at $t=0.3$ in the fourth row, which employs an identical color mapping to quantify the error magnitude.

4.2.3 2-D Euler System

At last, we consider the more complex two-dimensional inviscid euler system corresponding to the spatiotemporal domain of $\{(t,x_1,x_2)\}=[0,0.3]\times[0,1]\times[0,1]$,

$$\begin{cases} \partial_t\rho+\partial_{x_1}(\rho u)+\partial_{x_2}(\rho v)=0, \\ \partial_t(\rho u)+\partial_{x_1}(\rho u^2+p)+\partial_{x_2}(\rho uv)=0, \\ \partial_t(\rho v)+\partial_{x_1}(\rho uv)+\partial_{x_2}(\rho v^2+p)=0, \\ \partial_t(E)+\partial_{x_1}(u(E+p))+\partial_{x_2}(v(E+p))=0,. \\ E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2) \end{cases}, \tag{4.6}$$

here, ρ is the fluid density, u is the fluid velocity in the x_1 axis direction, v is the fluid velocity in the x_2 axis direction, and p is the ideal gas pressure. The corresponding initial and boundary conditions are

$$(\rho,u,v,p)=\begin{cases}(0.532258,1.206045,0,0.3), & if\,(x_1,x_2)\in[0,0.8]\times[0.8,1],\\ (0.137993,1.206045,1.206045,0.029032), & if\,(x_1,x_2)\in[0,0.8]\times[0,0.8],\\ (0.532258,0,1.206045,0.3), & if\,(x_1,x_2)\in[0.8,1]\times[0,0.8],\\ (1.5,0,0,1.5), & if\,(x_1,x_2)\in[0.8,1]\times[0.8,1].\end{cases},$$

$$\left(\frac{\partial\rho}{\partial\mathfrak{n}},\frac{\partial u}{\partial\mathfrak{n}},\frac{\partial v}{\partial\mathfrak{n}},\frac{\partial p}{\partial\mathfrak{n}}\right)=(0,0,0,0),(t,x_1,x_2)\in[0,0.3]\times\partial\Omega,$$

this initial condition is derived from Liska and Wendroff's [37] classic paper, used to test the performance of CFD algorithms in multidimensional shock wave interactions and this boundary condition is zero Neumann boundary condition, $\mathfrak{n}$ is the normal vector outside the unit of the region Ω. During the training process, we first employ the Adam optimizer for 5,000 iterations, followed by the L-BFGS optimizer for an additional 45,000 iterations. As with the previous example, only the results of the two-dimensional Euler system variable ρ are shown here. Figure 8 shows the results of three methods for solving the two-dimensional Euler equation variable ρ at t=0.1 and t=0.2, the high-precision reference results provided by clawpack, and the corresponding absolute error distribution.

The results presented in the figures and tables demonstrate significant improvements in shock wave capture achieved by the proposed methods, namely RPINN and RLPINN. In one-dimensional cases, the absolute error distributions across several examples indicate that both RPINN and RLPINN outperform the vanilla PINN substantially. This conclusion is further supported by the L2 relative error values provided in Table 4, where the proposed methods reduce errors by up to an order of magnitude compared to vanilla PINN. In two-dimensional cases, the absolute error plots reveal considerable improvement over vanilla PINN for both the Burgers and Euler systems. However, for the shallow water system, the gain, though observable, is less pronounced. The L2 error metrics summarized in Table 5 confirm these trends: significant error reduction is achieved for the Burgers and Euler systems, while the improvement for the shallow water system is relatively modest.

Finally, as illustrated in Figure 9, the training loss curves indicate that both RPINN and RLPINN lead to more effective optimization compared to vanilla PINN, confirming the enhanced capability of the proposed methods in handling this class of problems through reformulated optimization.

4.3 Results Analysis

It is noteworthy that RPINN demonstrates optimal performance for one-dimensional problems, whereas RLPINN proves more effective in two-dimensional settings. This observed dichotomy stems from fundamental differences in their theoretical frameworks,

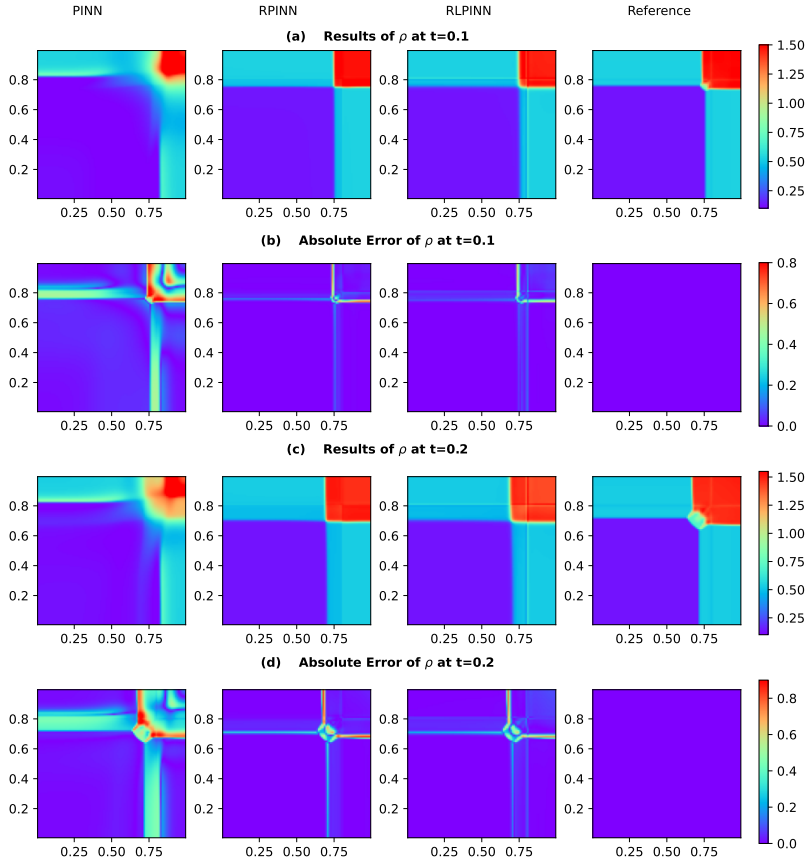

Figure 8: This figure presents a comprehensive comparison of the solutions for the two-dimensional Euler equation, focusing on the temporal evolution of the variable ρ. The first row illustrates the solution profiles for ρ at time $t=0.1$ obtained from the PINN, RPINN, and RLBINN methods alongside the high-fidelity reference. Directly below, the second row displays the corresponding absolute error distributions for ρ at $t=0.1$, where the color gradient from purple to red signifies the transition from low to high error values. The third row then presents the solution profiles for ρ at time $t=0.2$ using the same methodological order, followed by the absolute error distributions for ρ at $t=0.2$ in the fourth row, which employs an identical color mapping to quantify the error magnitude.

specifically in the physical roles and numerical behaviors of their respective relaxation parameters ε and relaxation matrices. These distinctions not only define the theoretical properties of each method but also directly govern their practical efficacy in solving conservation laws across different dimensions. For general systems of conservation laws, RPINN adopts the complete form of the relaxation system. The relaxation parameter $\varepsilon>0$ plays multiple critical roles: from a physical perspective, ε governs the rate at which the system approaches the equilibrium state, which in the context of hyperbolic conservation laws corresponds to the entropy solution of the original system. The relaxation matrix A_i satisfies the condition 3.4, this condition ensures that the eigenvalues of the relaxation

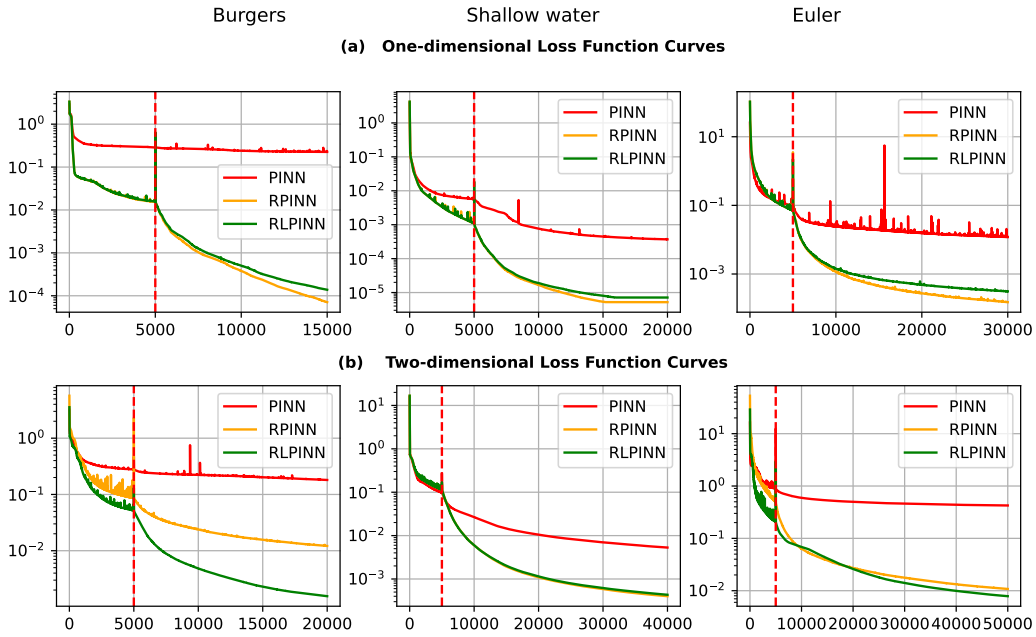

Figure 9: Variation diagram of loss function for six examples:the red solid line represents PINN, the yellow solid line represents RPINN, the green solid line represents RLPINN. The left curve of the red dashed line is iterated using Adam optimizer, while the right curve is iterated using L-BFGS optimizer.

system fully encompass those of the original system, thereby guaranteeing that the relaxed solution converges to the entropy solution of the original system as $\varepsilon \to 0$. At the numerical level, the dissipation term introduced by ε acts as a regularization mechanism with a magnitude of $O(\varepsilon)$. This controlled dissipation exhibits distinct advantages in one-dimensional problems. The structure of one-dimensional shocks is relatively simple, with shock fronts appearing as isolated points or lines, allowing the isotropic dissipation provided by ε to uniformly stabilize the numerical solution both ahead of and behind the shock without introducing significant directional bias. Furthermore, the temporal evolution characteristics of the relaxation process align closely with the physical propagation properties of one-dimensional problems. The propagation of characteristics along a single direction enables the wave structure of the relaxation system to be accurately captured.

At the implementation level, the relaxation term $\varepsilon(\partial_t v_i + A_i \partial_{x_i} u)$(where $i=1,\cdots,m.$) in RPINN introduces numerical viscosity at the discrete level. This viscosity automatically intensifies near shocks and naturally diminishes in smooth regions, achieving adaptive stability control. In one-dimensional problems, this mechanism precisely balances the sharpness of shock capture with numerical stability, avoiding dissipation anisotropy is-

sues that could arise from low-dimensional configurations. In contrast, RLPINN corresponds to the limiting case where $\varepsilon \to 0$, causing the relaxation system to reduce to a differential-algebraic system. This transformation brings fundamental changes in theoretical properties: the system evolves from a strictly hyperbolic PDE system into a differential-algebraic system with index 1. In two-dimensional and higher-dimensional problems, this simplified formulation demonstrates significant advantages, primarily because it avoids the anisotropic dissipation issues induced by the relaxation parameter in high-dimensional spaces. When $\varepsilon = 0$, the system completely eliminates the numerical dissipation introduced by the relaxation term, which holds particular significance in multi-dimensional problems. In two- or higher-dimensional spaces, any non-zero ε introduces numerical dissipation of varying degrees along different coordinate directions. This anisotropic nature can lead to unphysical distortions and deformations of shock fronts along different propagation directions. Particularly in regions of shock interaction or near complex geometric boundaries, anisotropic dissipation may trigger unphysical numerical oscillations or excessive dissipation, severely compromising solution accuracy. This issue becomes especially critical for anisotropic flux functions or in scenarios involving complex wave system interactions.It is worth noting that the anisotropic dissipation problem caused by ε mentioned above precisely highlights the unique advantages of RLPINN in simplifying the architecture. On the practical level, the simplified formulation of RLPINN also brings improvements in training efficiency. Eliminating the ε-dependent terms not only reduces the number of gradient components that need to be computed but also avoids the numerical stiffness and training instability caused by inappropriate ε values. For complex nonlinear flux functions, selecting an appropriate ε value is itself a challenge, which RLPINN completely circumvents. This enables RLPINN to demonstrate superior robustness and convergence characteristics when solving multi-dimensional complex flow problems, with its advantages becoming particularly pronounced in handling problems involving complex geometric boundaries and initial conditions.

Moreover, a natural question that arises concerns how the number of auxiliary networks impacts the performance and scalability of both RPINN and RLPINN in higher-dimensional settings. The fundamental design principle of this architecture is the decoupling of the flux computations $F_i(u)$ from the primary network $\hat{u}_1(\theta_1; t, \mathbf{x})$, assigning them instead to dedicated auxiliary networks $\hat{u}_{i+1}(\theta_{i+1}; t, \mathbf{x})$(where $i = 1, \cdots, m$.). A direct consequence of this approach is that the number of required auxiliary networks scales linearly with the spatial dimension m of the problem—a m-dimensional system necessitates exactly m flux networks.

This linear scaling presents a dual impact. On the positive side, this decoupling serves as a cornerstone for computational stability. It decomposes the complex, ill-conditioned high-order gradient computations into multiple parallel and independent first-order gradient computation paths. Each sub-network (comprising one main network and one flux network) only needs to learn a relatively simple input-output mapping, significantly mitigating gradient conflict and training difficulty. In two-dimensional problems , this strat-

egy, which trades a modest number of additional networks for enhanced training stability and convergence, has proven to be highly effective.

However, from a scalability perspective, the linearly increasing number of auxiliary networks may become a potential bottleneck in very high dimensions (e.g.,$d \geqslant 3$. Firstly, the total number of model parameters and computational overhead increases accordingly, although each flux network can be designed to be relatively compact. Secondly, and more critically, the challenge lies in coordinating the optimization process across these multiple independent networks. As the number of networks grows, ensuring synchronized and stable convergence for all becomes more complex and may introduce new optimization uncertainties.

5 Summary

This paper proposes two novel methods for shock wave capture in systems of conservation laws: the Relaxation Physics-Informed Neural Network (RPINN) and its simplified counterpart, the Reduced Relaxation Physics-Informed Neural Network (RLPINN), both grounded in relaxation system theory for conservation laws. These approaches employ first-order linear approximation systems to resolve shock waves via two principal mechanisms: (1) mitigating gradient explosion near discontinuities by circumventing direct differentiation of nonlinear flux functions, and (2) incorporating auxiliary neural networks to approximate flux terms within the loss formulation.

However, when extended to high-dimensional conservation laws, the proposed methods exhibit certain limitations in sharply resolving shock profiles. Although they successfully capture regular shock structures in both the two-dimensional Burgers and Shallow Water systems, challenges persist in accurately representing irregular shock patterns in the inviscid Euler system. We hypothesize that these difficulties stem primarily from the parametric complexity induced by multiple neural networks, which can introduce optimization obstacles in high-dimensional parameter spaces and hinder consistent convergence to physically admissible shock solutions. Resolving this inherent trade-off between model expressivity—arising from multi-network architecture—and optimizability in multidimensional settings remains a crucial direction for future work.

Acknowledgments

The author would like to thank Prof. Xu Jiang and Assoc. Prof. Yang Yong from the School of Mathematics, Nanjing University of Aeronautics and Astronautics for their valuable discussions. The author also thank all of the editors and reviewers for their very important suggestions.

Conflicts of Interest

The author(s) declare no conflict of interest.

References

[1] D. Gaitonde, Progress in shock wave/boundary layer interactions, *Prog. Aerosp. Sci.*, 2015, 72: 80-99.

[2] M. S. Longair, *High energy astrophysics*, Cambridge university press, 2011.

[3] K. Mattila, T. Puurtinen, J. Hyväluoma, et al, A prospect for computing in porous materials research: Very large fluid flow simulations, *J. Comput. Sci.*, 2016, 12: 62-76.

[4] T. Kodama, A. G. Doukas and M. R. Hamblin, Shock wave-mediated molecular delivery into cells, *Biochim. Biophys. Acta, Mol. Cell Res.*, 2002, 1542(1-3): 186-194.

[5] M. Sahimi, *Flow and transport in porous media and fractured rock: from classical methods to modern approaches*, John Wiley & Sons, 2011.

[6] J. Zhang, U. Braga-Neto and E. Gildin, Physics-informed neural networks for multiphase flow in porous media considering dual shocks and interphase solubility, *Energy Fuels*, 2024, 38(18): 17781-17795.

[7] K. Mitra, T. K"oppl and I. S. Pop, et al, Fronts in two-phase porous media flow problems: The effects of hysteresis and dynamic capillarity, *Stud. Appl. Math.*, 2020, 144(4): 449-492.

[8] J. Bear, *Dynamics of fluids in porous media*, Courier Corporation, 2013.

[9] D. V. Griffiths and I. M. Smith, *Numerical methods for engineers*, Chapman and Hall/CRC, 2006.

[10] M. Raissi, P. Perdikaris and G. E. Karniadakis, 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.

[11] L. Liu, S. Liu and H. Xie, et al. Discontinuity computing using physics-informed neural networks, *J. Sci. Comput.*, 2024, 98(1): 22-59.

[12] T. Wang, B. Zhao and S. Gao, et al, Understanding the difficulty of solving Cauchy problems with PINNs, in: *Proc. 6th Annu. Learn. Dyn. Control Conf.*, 2024: 453-465.

[13] T. De Ryck and S. Mishra, Numerical analysis of physics-informed neural networks and related models in physics-informed machine learning, *Acta Numer.*, 2024, 33: 633-713.

[14] S. Wang, Y. Teng and P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM J. Sci. Comput.*, 2021, 43(5): A3055-A3081.

[15] A. D. Jagtap and G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *Commun. Comput. Phys.*, 2020, 28(5).

[16] O. Fuks and H. A. Tchelepi, Limitations of physics informed machine learning for nonlinear two-phase transport in porous media, *J. Mach. Learn. Model. Comput.*, 2020, 1(1).

[17] R. G. Patel, I. Manickam and N. A. Trask, et al, Thermodynamically consistent physics-informed neural networks for hyperbolic systems, *J. Comput. Phys.*, 2022, 449: 110754.

[18] T. De Ryck, S. Mishra and R. Molinaro, Weak physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws, *SIAM J. Numer. Anal.*, 2024, 62(2): 811-841.

[19] A. Chaumet and J. Giesselmann, Efficient wpinn-approximations to entropy solutions of hyperbolic conservation laws, 2022, *https: //doi.org/10.48550/arXiv.2211.12393*.

[20] S. Jin and Z. Xin, The relaxation schemes for systems of conservation laws in arbitrary space dimensions, *Commun. Pure Appl. Math.*, 1995, 48(3): 235-276.
[21] Clawpack Development Team (2024), Clawpack Version 5.11.0, Available online: https://www.clawpack.org (accessed on 2024).
[22] G. Chen, C. D. Levermore and T. Liu, Hyperbolic conservation laws with stiff relaxation terms and entropy, *Commun. Pure Appl. Math.*, 1994, 47(6): 787-830.
[23] T. Liu, Hyperbolic conservation laws with relaxation, *Commun. Math. Phys.*, 1987, 108: 153-175.
[24] R. Courant, K. Friedrichs and H. Lewy, On the partial difference equations of mathematical physics, *IBM J. Res. Dev.*, 1967, 11(2): 215-234.
[25] B. M. Wilamowski, N. J. Cotton, O. Kaynak, et al, Computing gradient vector and Jacobian matrix in arbitrarily connected neural networks, *IEEE Trans. Ind. Electron.*, 2008, 55(10): 3784-3790.
[26] W. Cai, R. Encarnacion, B. Chern, et al, Adaptive sampling strategies to construct equitable training datasets, in: *Proc. ACM Conf. Fairness Account. Transp.*, 2022: 1467-1478.
[27] J. M. Burgers, A mathematical model illustrating the theory of turbulence, *Adv. Appl. Mech.*, 1948, 1: 171-199.
[28] J. A. Liggett and D. A. Woolhiser, Difference solutions of the shallow-water equation, *J. Eng. Mech. Div.*, 1967, 93(2): 39-72.
[29] A. V. Gavrilov, A steady Euler flow with compact support, *Geom. Funct. Anal.*, 2019, 29(1): 190-197.
[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[31] A. Quarteroni, R. Sacco and F. Saleri, *Numerical Mathematics*, Springer Science & Business Media, 2006.
[32] M. Pérez, An Investigation of ADAM: A Stochastic Optimization Method, in: *Proc. ICML*, 2022: 17-23.
[33] D. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Program.*, 1989, 45(1): 503-528.
[34] N. Ashgriz and J. Mostaghimi, An introduction to computational fluid dynamics, *Fluid Flow Handb.*, 2002, 1: 1-49.
[35] R. Vinuesa and S. L. Brunton, Enhancing computational fluid dynamics with machine learning, *Nat. Comput. Sci.*, 2022, 2(6): 358-366.
[36] F. Wang, I.L. Animasaun and T. Muhammad, et al, Recent advancements in fluid dynamics: drag reduction, lift generation, computational fluid dynamics, turbulence modelling, and multiphase flow, *Arab. J. Sci. Eng.*, 2024, 49(8): 10237-10249.
[37] R. Liska and B. Wendroff, Composite schemes for conservation laws, *SIAM J. Numer. Anal.*, 1998, 35(6): 2250–2271.