
Hybrid Physics-Informed Neural Networks Integrating Multi-Relaxation-Time Lattice Boltzmann Method for Forward and Inverse Flow Problems

[Mengyu Feng](#), [Minglei Shan](#)^{*}, [Ling Kuai](#), Chenghui Yang, Yu Yang, Cheng Yin, [Qingbang Han](#)

Posted Date: 16 October 2025

doi: 10.20944/preprints202510.1309.v1

Keywords: physics-informed neural networks; multi-relaxation-time lattice Boltzmann method; computational fluid dynamics; deep learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

Hybrid Physics-Informed Neural Networks Integrating Multi-Relaxation-Time Lattice Boltzmann Method for Forward and Inverse Flow Problems

Mengyu Feng¹, Minglei Shan^{1,*} , Ling Kuai¹, Chenghui Yang¹, Yu Yang², Cheng Yin¹ and Qingbang Han¹

¹ College of Information Science and Engineering, Hohai University, Changzhou 213200, China

² College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China

* Correspondence: shanml@hhu.edu.cn

Abstract

The development of stable, accurate, and generalizable numerical simulation methods remains a critical challenge in computational fluid dynamics (CFD). Recently, physics-informed neural networks (PINNs) have emerged as a promising approach by embedding physical laws directly into the loss function of neural networks, enabling mesh-free solutions of governing equations. PINNs offer a new perspective on CFD and open a significant pathway for AI for science. However, existing PINN models often face trade-offs between generality, stability, and accuracy. To address these trade-offs, this paper proposes a novel hybrid architecture, PINN-MRT, which integrates the multi-relaxation-time lattice Boltzmann method (MRT-LBM) with PINNs. For the first time, the MRT-LBM evolution equation is embedded as the physics-informed residual within the loss function. Due to the mesoscopic kinetic nature of the MRT-LBM equations, the proposed PINN-MRT inherently possesses the potential for fluid solution generality. The PINN-MRT adopts a dual-network architecture, which separately predicts macroscopic conserved variables and non-equilibrium distribution functions. A composite loss function is then constructed to incorporate physical residuals, boundary conditions, and data-driven terms, enabling the PINN-MRT architecture to simultaneously address both forward and inverse problems. Numerical validation on both forward and inverse problems confirms the superior stability and predictive accuracy of the proposed PINN-MRT model, demonstrating significant improvements over the standard PINN and existing PINN-LBM hybrid architectures. This study provides a novel stable, accurate, and generalizable PINN architecture for CFD research.

Keywords: physics-informed neural networks; multi-relaxation-time lattice Boltzmann method; computational fluid dynamics; deep learning

1. Introduction

Developing stable, accurate, and generalizable numerical simulation methods has long been one of the core challenges in computational fluid dynamics (CFD) [1–3]. Traditional numerical methods, such as the finite difference method (FDM), finite volume method (FVM), and spectral method (SM), have established the basic framework of modern CFD through discretization theory, mesh partitioning, and algebraic equation solving [1,4]. These methods have become benchmark tools for simulating physical phenomena such as fluid dynamics and heat transfer. However, when dealing with strong nonlinearity, discontinuous solutions, and complicated geometries, these traditional methods face significant limitations, including heavy reliance on refined meshes, numerical instability, and severe accuracy degradation or even failure [5–7].

In recent years, physics-informed neural networks (PINNs) have emerged as a novel computational method that combines the advantages of physical laws and data-driven learning [8]. Leveraging the mesh-free computation, PINNs demonstrate great potential in addressing diverse fluid problems,

gradually becoming a key new paradigm to break through the bottlenecks of traditional numerical methods [9]. Specifically, PINNs embed physical laws directly into the loss function of neural networks and utilize parameter optimization to implicitly solve the governing equations, thus largely eliminating mesh dependence and enhancing adaptability to complex physical problems [10–12]. For instance, this approach has demonstrated excellent performance in shock capturing, turbulence modeling, and unsteady flow simulations [13]. Nonetheless, PINNs also suffer from challenges including the emergence of spurious solutions, convergence difficulties, and limited capability for dynamic system modeling [10,14,15]. As a result, introducing hybrid model architectures to improve robustness has become a promising research direction. Among these approaches, the integration of PINNs with the lattice Boltzmann method (LBM) represents a recently developed research direction.

The LBM, owing to its clear evolution mechanism, highly parallel computational structure, and flexibility in handling intricate boundaries and multi-physics coupling problems, has been widely adopted for numerical simulation in CFD [16]. Its fundamental governing equation, the lattice Boltzmann equation (LBE), originates from the underlying kinetic equation, i.e., the continuous Boltzmann equation. This intrinsic physical nature rooted in kinetic theory enables the LBE to systematically preserve the essential coupling between microscopic particle dynamics and macroscopic conservation law evolution in physical modeling, thus endowing it with inherent physical consistency, theoretical closure, and high generality for fluid flow solutions [17,18]. The LBE models the mesoscopic evolution of particle distributions through collision and streaming processes, and offers a solid theoretical basis for complex flow simulations.

Based on this, some researchers have proposed combining LBM with PINNs, replacing conventional partial differential equations, such as Navier-Stokes (N-S) equations, with the LBE as the underlying governing equations, thereby constructing hybrid models that integrate mesoscopic physical constraints [19]. For example, Lou et al. proposed a PINN architecture incorporating the Boltzmann-Bhatnagar-Gross-Krook (BGK) equation, substantially enhancing the model physical accuracy, and alleviating issues such as slow convergence and difficulties in modeling highly nonlinear scenarios [20]. However, due to coupling the relaxation of all hydrodynamic and non-hydrodynamic moments to a single parameter, the BGK model exhibits severe parameter coupling and numerical instability in flow simulations, thus limiting its applicability in complex scenarios [16,17].

To address these challenges, this paper proposes a PINN architecture incorporating the multi-relaxation-time lattice Boltzmann method (MRT-LBM). The MRT collision mechanism effectively decouples the relaxation processes of different order momentum components by assigning independent relaxation time to each physical mode [21,22]. This approach retains the advantages of mesoscale modeling while markedly improving the accuracy, numerical stability, and generalizability for diverse fluid structures. The remainder of this paper is organized as follows: Section 2 presents the theoretical foundation, along with the design of the proposed PINN-MRT architecture that integrates dual-network and physics-informed loss functions. Section 3 introduces the benchmark problem setup and model configurations. Section 4 demonstrates the applications of PINN-MRT to forward and inverse problems, including predictive accuracy and parameter identification at different Reynolds numbers (Re). Finally, Section 5 summarizes the main findings and outlines future research directions.

2. Method

2.1. MRT-LBM

The LBM is an efficient numerical simulation approach at the mesoscopic scale, describing the statistical evolution of particle distribution functions through discretization in both space and time, as well as along specific discrete velocity directions. The fundamental dynamical model of LBM originates from the continuous Boltzmann equation [23]

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = \Omega(f), \quad (1)$$

where f denotes the particle distribution function in continuous velocity space, ξ represents the microscopic particle velocity, and $\Omega(f)$ is the collision operator, characterizing the relaxation process toward local equilibrium. A classical simplification is the single-relaxation-time (SRT) or BGK approximation, which linearizes the collision operator to model the relaxation process toward a local equilibrium distribution function [24]

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = -\frac{1}{\tau}(f - f^{eq}), \quad (2)$$

$$f^{eq} = \frac{\rho}{(2\pi RT)^{D/2}} \exp\left[-\frac{|\xi - \mathbf{u}|^2}{2RT}\right]. \quad (3)$$

In Eq.2, τ denotes the single relaxation factor, which directly determines the fluid kinematic viscosity, and f^{eq} represents the equilibrium distribution function. However, the SRT model employs a unified relaxation parameter across all dynamical moments, which inherently leads to strong coupling among different physical processes and restricts the flexibility in independently controlling individual transport phenomena [24]. Fortunately, the MRT model introduces a transformation into moment space assigning independent relaxation parameters to distinct physical modes (mass, momentum, stress, etc.). The corresponding equation is shown as [25,26]

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = -(\mathbf{M}^{-1}\mathbf{\Lambda}\mathbf{M})(f - f^{eq}), \quad (4)$$

where \mathbf{M} denotes an orthogonal transformation matrix that maps f from velocity space to moment space, and $\mathbf{\Lambda}$ is a diagonal matrix whose elements correspond to the relaxation times associated with individual moments. The D2Q9 discrete velocity model is adopted, and the resulting discretized evolution equation is given by

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -(\mathbf{M}^{-1}\mathbf{\Lambda}\mathbf{M})_{ij}(f_j - f_j^{eq}), \quad (5)$$

where f_i is the particle distribution function. f_j^{eq} is the equilibrium discrete distribution function, and for the D2Q9 discrete velocity model can be expressed as

$$f_j^{eq}(\mathbf{x}, t) = w_j \rho \left[1 + \frac{\mathbf{e}_j \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_j \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right], \quad (6)$$

where w_i is weight coefficients in the i th direction, which is given by

$$w_i = \begin{cases} \frac{4}{9}, & i = 0 \\ \frac{1}{9}, & i = 1, 2, 3, 4 \\ \frac{1}{36}, & i = 5, 6, 7, 8 \end{cases} \quad (7)$$

The macroscopic fluid density ρ and macroscopic fluid velocity \mathbf{u} are shown as [27]

$$\rho = \sum_i f_i, \quad \rho \mathbf{u} = \sum_i \mathbf{e}_i f_i, \quad (8)$$

where \mathbf{e}_i discrete velocity along the i th direction. In this study, \mathbf{e}_i is given by

$$\mathbf{e}_i = \begin{cases} (0, 0), & i = 0 \\ (\cos[(i-1)\frac{\pi}{2}], \sin[(i-1)\frac{\pi}{2}]), & i = 1, 2, 3, 4 \\ \sqrt{2}(\cos[(2i-1)\frac{\pi}{4}], \sin[(2i-1)\frac{\pi}{4}]). & i = 5, 6, 7, 8 \end{cases} \quad (9)$$

The orthogonal transformation matrix is expressed as [27]

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}. \quad (10)$$

The diagonal relaxation matrix Λ is shown as

$$\Lambda = \text{diag}(\tau_\rho^{-1}, \tau_e^{-1}, \tau_f^{-1}, \tau_j^{-1}, \tau_q^{-1}, \tau_j^{-1}, \tau_q^{-1}, \tau_v^{-1}), \quad (11)$$

where τ_ρ^{-1} , τ_e^{-1} , τ_f^{-1} , τ_j^{-1} , τ_q^{-1} , and τ_v^{-1} represent the relaxation factors associated with density, energy, square root of energy, momentum flux component, energy flux component, and pressure tensor, respectively. In this study, the relaxation factors are set as follows, $\tau_\rho^{-1} = \tau_j^{-1} = 1.0$, $\tau_e^{-1} = \tau_f^{-1} = 0.8$, and $\tau_q^{-1} = 1.1$. In the MRT model, the kinematic viscosity ν is related to the relaxation factor τ_v by

$$\nu = c_s^2 \left(\tau_v - \frac{1}{2} \right) \delta t, \quad (12)$$

where c_s denotes the lattice sound speed, with $c_s = \sqrt{1/3}$ in the D2Q9 model. δt is the time step.

2.2. PINN-MRT

A novel architecture (PINN-MRT) is developed in this study by integrating MRT-LBM into the PINNs architecture. As shown in Figure 1, the PINN-MRT architecture comprises two main components, i.e. the neural network architecture and the loss function design.

2.2.1. Neural Network Architecture

The neural network architecture has two sub-networks. The first sub-network, denoted as NN_{eq} , takes the spatial coordinates (x, t) as input and outputs the macroscopic density ρ and velocity components (u, v) , then calculates the equilibrium distribution function f_i^{eq} using Eq.6. The second sub-network, NN_{neq} , shares the same input as NN_{eq} but predicts the nine components of the non-equilibrium distribution function f_i^{neq} . This physically motivated separation leverages the smoothness of f_i^{eq} and the complexity of f_i^{neq} , allowing each network to focus on distinct features and thereby improving stability and accuracy [28].

Based on the predicted outputs, the total distribution function at each spatial location is composed as

$$f_i = f_i^{eq} + f_i^{neq}. \quad (13)$$

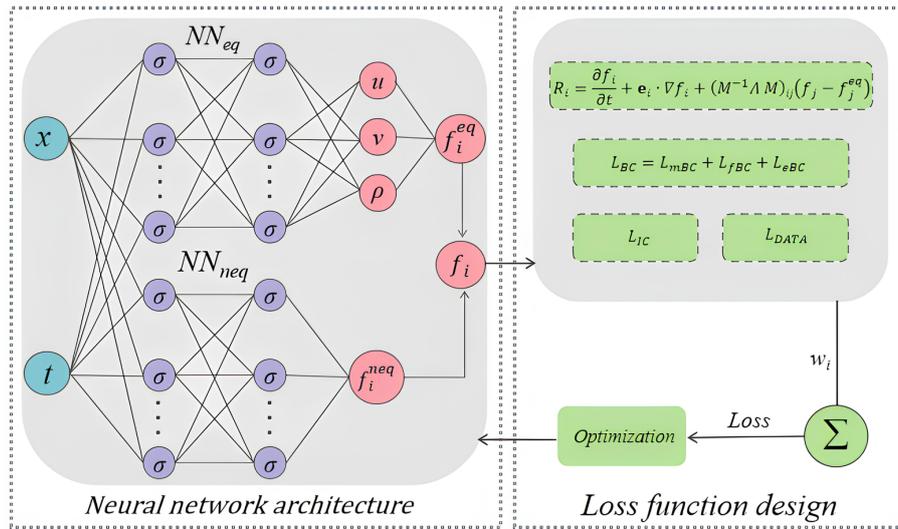


Figure 1. The presented PINN-MRT architecture with MRT-LBM physics constraints.

2.2.2. Loss Function Design

For effective training of the PINN-MRT model, a loss function is formulated incorporating the physical residuals, boundary conditions, initial conditions, and data-driven terms, as shown in Eq.14 [29].

$$\mathcal{L} = \beta_1 L_{\text{PDE}} + \beta_2 L_{\text{BC}} + \beta_3 L_{\text{IC}} + \beta_4 L_{\text{DATA}}, \quad (14)$$

where $\beta_i (i = 1, 2, 3, 4)$ denote the weight coefficients corresponding to the respective loss components. To ensure consistency in comparisons with other models, this study does not implement weight optimization and instead adopts unit weight coefficients (i.e., $\beta_i = 1$) for active loss terms, while setting $\beta_i = 0$ for unused components.

The physics residual loss L_{PDE} ensures that the neural network predictions satisfy the governing dynamics derived from the MRT-LBM equations. The specific equation can be expressed as [30]

$$L_{\text{PDE}} = \frac{1}{N_e} \sum_{i=0}^{Q-1} \sum_{j=0}^{N_t-1} |R_i(\mathbf{x}_j, t_j)|^2, \quad (15)$$

$$R_i = \frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i + (\mathbf{M}^{-1} \Lambda \mathbf{M})_{ij} (f_j - f_j^{eq}), \quad (16)$$

where R_i denotes the residual of the MRT-LBM evolution in the i -th discrete velocity direction, Q refers to the total number of discrete velocity directions, and N_t and N_e specify the numbers of temporal and internal sampling points, respectively. The spatial coordinate for the j -th sampling point is given by \mathbf{x}_j , with t_j representing its corresponding temporal coordinate.

The boundary condition loss L_{BC} is introduced to enforce that the predicted solutions strictly satisfy the prescribed boundary conditions, thereby ensuring physical consistency at the domain boundaries. In the LBM kinetic framework, it is further divided into three components

- Macroscopic boundary loss L_{mBC} : Constrains the predicted macroscopic variables on the boundaries to match the prescribed physical values.
- Distribution function consistency loss L_{fBC} : Constrains the predicted equilibrium and non-equilibrium distribution functions to match the theoretically defined boundary distributions.
- Boundary PDE residual loss L_{eBC} : Imposes constraints on the neural network outputs by penalizing the residuals of the governing equations at the boundaries.

The equations for the three components are given by

$$L_{m_{BC}} = \frac{1}{N_b} \sum_{j=0}^{N_b-1} \left| \phi_{NN_{neq}}(\mathbf{x}_{b,j}, t_j) - \phi^*(\mathbf{x}_{b,j}, t_j) \right|^2, \quad (17)$$

$$L_{f_{BC}} = \frac{1}{N_b} \sum_{j=0}^{N_b-1} \sum_{\xi_i \cdot \mathbf{n} > 0} \left| f_{i,NN_{neq}}(\mathbf{x}_{b,j}, t_j) - f_{i,neq}^*(\mathbf{x}_{b,j}, t_j) \right|^2, \quad (18)$$

$$L_{e_{BC}} = \frac{1}{N_b} \sum_{j=0}^{N_b-1} \sum_{\xi_i \cdot \mathbf{n} < 0} \left| R_i(\mathbf{x}_{b,j}, t_j) \right|^2. \quad (19)$$

Thus, the formula for L_{BC} is

$$L_{BC} = L_{m_{BC}} + L_{f_{BC}} + L_{e_{BC}}. \quad (20)$$

In Eq.17, where $\phi_{NN_{neq}}$ denotes the predicted macroscopic physical quantities (e.g., velocity or density), ϕ^* represents the ground truth boundary condition, N_b is the number of boundary sampling points, and $\mathbf{x}_{b,j}$ and t_j denote the spatial and temporal coordinates, respectively, of the j -th boundary point. In Eq.18, $f_{i,NN_{neq}}^{neq}$ represents the non-equilibrium distribution function in the i -th discrete velocity direction predicted by the neural network, $f_i^{neq,*}$ is the theoretical reference value at the boundary, and $\xi_i \cdot \mathbf{n} > 0$ indicates that only outgoing velocity directions are considered (from the interior to the boundary). Eq.19 indicates that the evolution equation in the incident direction remains enforced at the boundary.

The initial condition loss L_{IC} ensures that the predicted initial state strictly conforms to the known reference solution. The equation is given by

$$L_{IC} = \frac{1}{N_i} \sum_{j=0}^{N_i-1} \left| \phi_{NN_{neq}}(\mathbf{x}_j, 0) - \phi^*(\mathbf{x}_j, 0) \right|^2, \quad (21)$$

with N_i representing the number of sampling points.

The data-driven loss L_{DATA} minimizes the discrepancy between the predicted values and the experimentally measured or observed data. It is defined as

$$L_{DATA} = \frac{1}{N_d} \sum_{j=0}^{N_d-1} \left| u(\mathbf{x}_{u_j}, t_{u_j}) - u^*(\mathbf{x}_{u_j}, t_{u_j}) \right|^2, \quad (22)$$

where N_d denotes the total number of data points, u^* represents the observed velocity data, thereby guiding the neural network to better approximate the actual physical data.

3. Benchmark Modeling

The classical lid-driven cavity flow is selected as the benchmark case to evaluate the model performance. As shown in Figure 2, the computational domain is a unit square with spatial coordinates $(x, y) \in [0, 1]^2$. A constant velocity of $U = 1$, $V = 0$ is prescribed on the top boundary, and the remaining walls are subjected to no-slip conditions (i.e., stationary). This setup induces a characteristic steady-state vortex structure that varies with Re , making it a widely adopted benchmark for assessing

the predictive capability of both forward and inverse modeling approaches. The specific governing equations are written as [31]

$$\begin{aligned}\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \\ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right),\end{aligned}\quad (23)$$

where (u, v) denote the velocity components in the x and y directions, respectively, p is the pressure, and ν represents the kinematic viscosity, which can be expressed in terms of the Re as

$$\nu = UL/Re, \quad (24)$$

here, U represents the lid velocity, and L denotes the characteristic length (i.e., the cavity length).

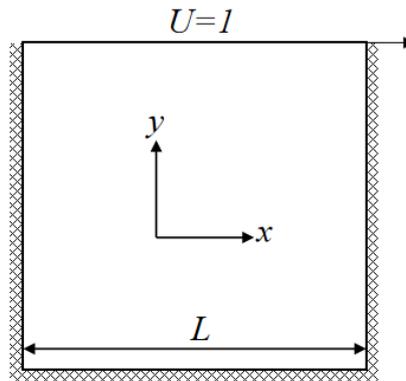


Figure 2. 2D computational domain and boundary conditions for the lid-driven cavity flow.

These governing equations are formulated from the general N-S equations under steady-state, incompressible, and two-dimensional assumptions.

To rigorously evaluate the performance of the proposed PINN-MRT architecture, two benchmark models are established for comparative analysis. All models share the same network architecture and training configuration, differing solely in the formulation of the residual term L_{PDE} , which encodes the respective physical constraints [32]. The standard PINN adopts Eq.23 as its macroscopic residual constraints. Meanwhile, PINN-SRT introduces mesoscopic physical constraints based on SRT equation, which can be regarded as a special case of the MRT model when all the diagonal elements of the relaxation matrix are set to unity [32].

The detailed configurations for each model in inverse problem are listed in Table 1. The training data are obtained by solving Eq.23 using the traditional FDM, which also provides the reference numerical solutions for comparison in Section.4.

Table 2 summarizes the corresponding configurations for forward problem. In this context, all physical parameters are known, and no additional data are introduced. A deeper network architecture (7-layers) is adopted to handle this more challenging data-free prediction task. Meanwhile, the sub-network for the non-equilibrium part uses more neurons (100 vs. 60) to better capture complex local features [10,14,33,34].

Table 1. Configurations and training settings for inverse problem.

Category	Description
Network structure	5-layer fully connected network (1 input, 3 hidden, 1 output)
Hidden neurons	60 neurons per hidden layer
Activation function	Tanh
Trainable parameter	Both ν and U_{lid} are randomly initialized in $(0, 1)$
Loss function	$\beta_1 = \beta_2 = \beta_4 = 1, \beta_3 = 0$
Data source	FDM
Data location	Random sampling
Sampling distribution	20,000 collocation pts; 5,000 boundary pts; 1,000 data pts
Optimizer	Adam (80,000 iterations, initial LR 10^{-3} , exponential decay)

Table 2. Configurations and training settings for forward problem.

Category	Description
Network structure	7-layer fully connected network (1 input, 5 hidden, 1 output)
Hidden neurons	60 neurons per hidden layer in NN_{eq} ; 100 in NN_{neq}
Activation function	Tanh
Trainable parameter	None
Loss function	$\beta_1 = \beta_2 = 1, \beta_3 = \beta_4 = 0$
Data source	None
Data location	None
Sampling distribution	20,000 collocation pts; 5,000 boundary pts
Optimizer	Adam (80,000 iterations, initial LR 10^{-3} , exponential decay)

4. Results

4.1. Inverse Problem

The subsequent analysis of the inverse problem is divided into two parts: Flow Field Analysis, focusing on the reconstruction of physical fields, and Parameter Inversion, addressing the identification of underlying model parameters.

4.1.1. Flow Field Analysis

The comprehensive performance comparison among the standard PINN, PINN-SRT and PINN-MRT models is conducted up to $Re = 1000$, since the standard PINN and PINN-SRT models become numerically unstable and fail to produce reliable results when the Re exceeds 1000. In this range, all models are systematically analyzed in terms of vortex structure reconstruction, error distribution, and velocity profile consistency. To further verify the robustness of the PINN-MRT model under more challenging flow conditions, additional evaluations at $Re = 2000$ and 5000 are conducted exclusively for the PINN-MRT model.

(A) Vortex Structure Reconstruction and Error Distribution

Figure 3 illustrates the predicted u and v velocity fields at $Re = 100$, showing small differences among three models. The absolute error (Abs) distributions in Figure 4 further confirm this observation. While the errors of all three models remain comparably low, PINN-MRT achieves the smallest and most localized errors. These results motivate further exploration at higher Re .

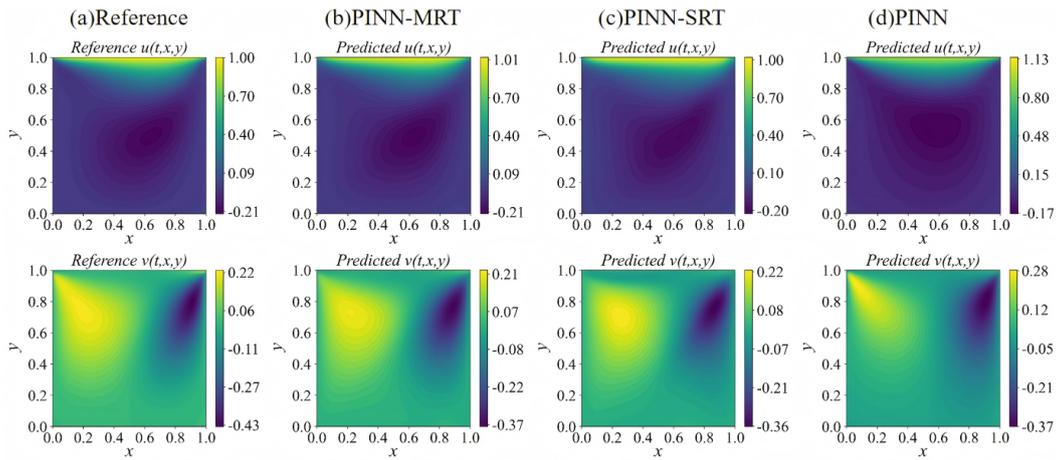


Figure 3. Inverse problem solution comparison of velocity field at $Re = 100$.

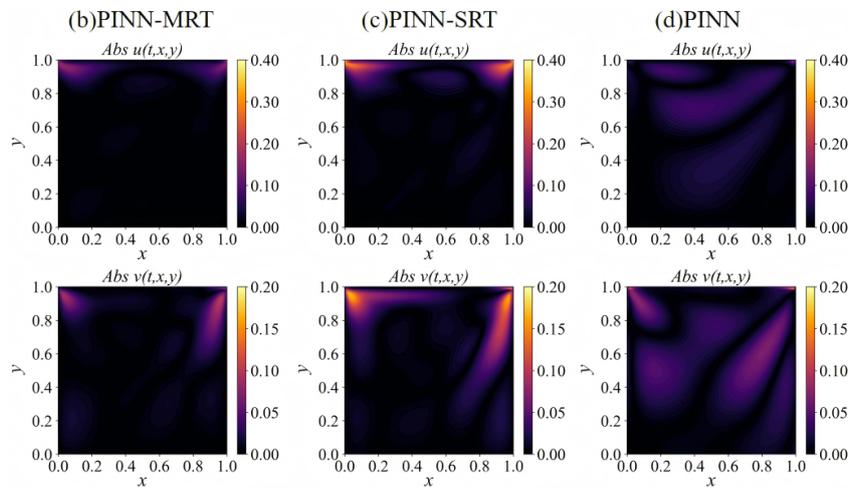


Figure 4. Absolute error distributions of the velocity components u and v at $Re = 100$.

Figure 5 presents the flow field results at $Re = 1000$, where more complex structures appear compared to low Re . The primary vortex is mainly identified in the u velocity field as a large central rotational structure, whereas the secondary vortices are clearly observed in the v velocity field near the bottom corners. Among the models, the PINN-MRT accurately reconstructs both primary and secondary flow structures features, capturing their rotational directions and spatial distributions. By comparison, the PINN-SRT partially recovers the primary vortex but fails to resolve the secondary features. The standard PINN exhibits a complete breakdown in prediction. As shown in Figure 6, the PINN-MRT maintains low error levels, with peak errors below 0.10 for u and 0.15 for v . Meanwhile, the PINN-SRT reaches peak errors of 0.30 with wider distributions, and the standard PINN yields errors exceeding 0.40, indicating clear failure.

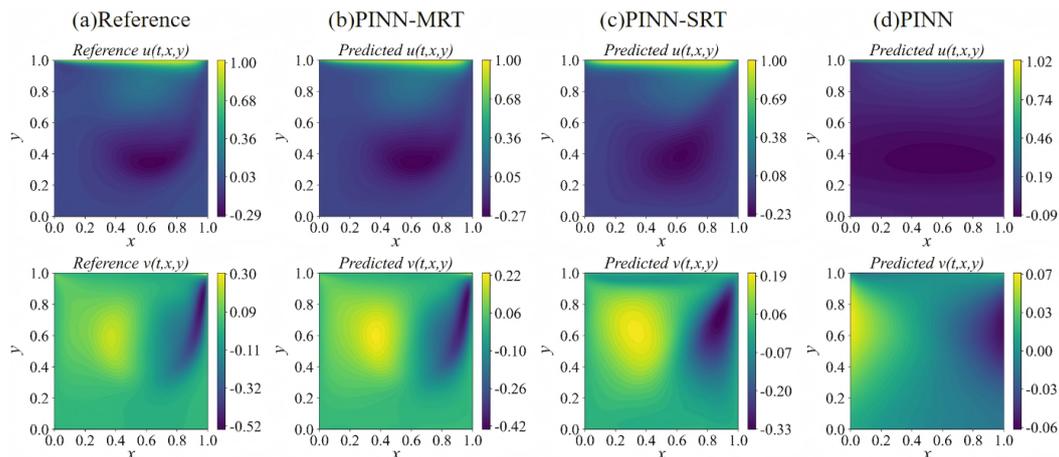


Figure 5. Inverse problem solution comparison of velocity field at $Re = 1000$.

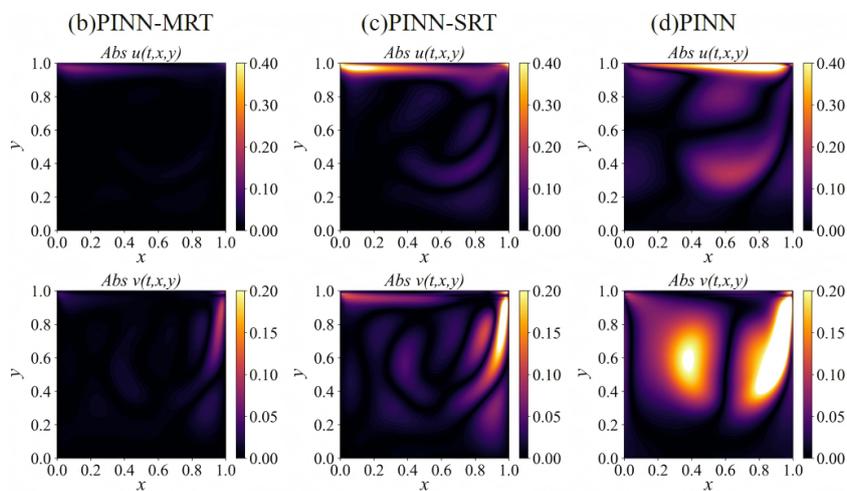


Figure 6. Absolute error distributions of the velocity components u and v at $Re = 1000$.

(B) Velocity Profile

Centerline velocity profiles are analyzed to evaluate the accuracy of flow reconstruction along key directions. Table 3 summarizes the absolute error ranges of the horizontal and vertical velocity components for $Re = 100$ and 1000 . The PINN-MRT model achieves the narrowest error bounds in both directions, confirming its superior predictive accuracy.

Table 3. Absolute error ranges of horizontal (Δu) and vertical (Δv) velocity components for all three models in inverse problem at $Re = 100$ and 1000 .

Model	Δu	Δv
Re = 100		
PINN	[0.0005, 0.0763]	[0.0005, 0.0654]
PINN-SRT	[0.0002, 0.0531]	[0.0003, 0.0526]
PINN-MRT	[0.0003, 0.0447]	[0.0004, 0.0310]
Re = 1000		
PINN	–	–
PINN-SRT	[0.0005, 0.2195]	[0.0001, 0.1318]
PINN-MRT	[0.0005, 0.0573]	[0.0001, 0.0635]

(C) High-Re Error Evaluation of PINN-MRT

The summarized relative L^2 errors of the PINN-MRT model across all tested Re are presented in Table 4. Here, err_u and err_v represent the horizontal and vertical velocity component errors,

respectively, and $\text{err}_{\text{total}}$ denotes the total velocity magnitude error. As the Re increases from 100 to 5000, the errors naturally rise, with err_u increasing from 9.067% to 15.281%, err_v from 11.942% to 21.262%, and $\text{err}_{\text{total}}$ from 4.549% to 14.121%. Nevertheless, the error levels remain well controlled, demonstrating the robustness and practical applicability of the PINN-MRT model.

Table 4. Relative L^2 errors (%) of PINN-MRT in inverse problem.

PINN-MRT	err_u (%)	err_v (%)	$\text{err}_{\text{total}}$ (%)
Re = 100	9.067	11.942	4.549
Re = 1000	12.179	17.048	9.894
Re = 2000	14.088	20.369	13.845
Re = 5000	15.281	21.262	14.121

4.1.2. Parameter Inversion

Building on the successful flow field reconstruction, the proposed PINN-MRT model is further assessed for its capability in parameter inversion, focusing on both the kinematic viscosity coefficient ν and the lid velocity U_{lid} . The inversion accuracy is systematically compared among the standard PINN, PINN-SRT, and PINN-MRT models across Reynolds numbers ranging from $\text{Re} = 100$ to 5000, as well as under noise-contaminated conditions.

As shown in Figure 7, the left panels present the relative L^2 errors of the inferred ν and U_{lid} for different models, while the right panels illustrate the corresponding robustness tests with noisy data, where the shaded regions denote a $\pm 5\%$ uncertainty band.

For the noiseless cases (left column), both PINN and PINN-SRT exhibit a rapid degradation of accuracy as Re increases, with relative errors exceeding 80% and 50%, respectively, at $\text{Re} = 5000$. In contrast, the PINN-MRT maintains errors below 20% for both parameters across the entire Reynolds number range, demonstrating remarkable stability and resistance to ill-conditioning at high Reynolds numbers.

Under noisy inputs (right column), the PINN-MRT results remain within the $\pm 5\%$ confidence interval, showing only marginal increases in error. This confirms that the MRT-based regularization effectively suppresses the amplification of noise during inverse inference, thereby ensuring both robustness and consistency of the reconstructed physical parameters.

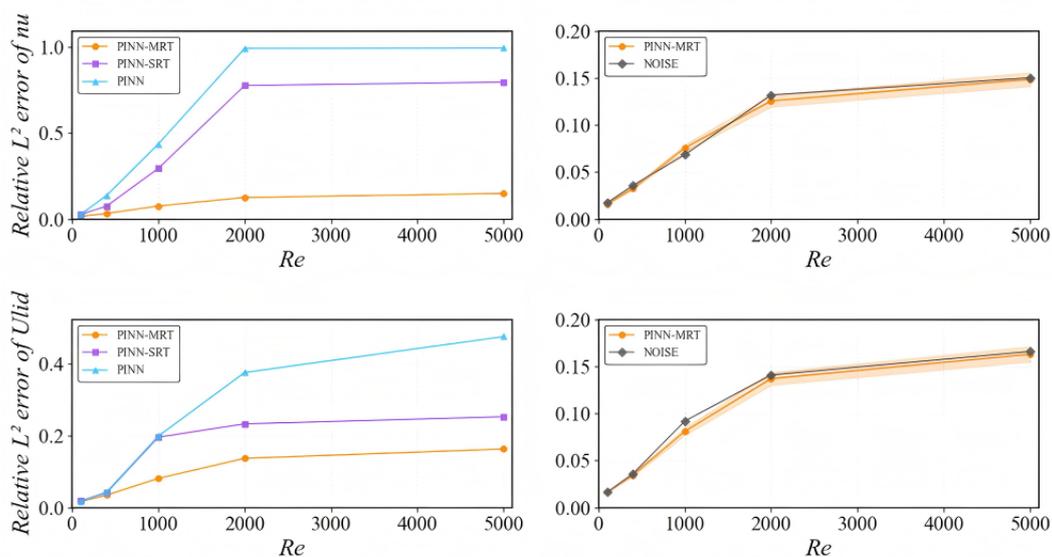


Figure 7. Relative L^2 errors of the inferred parameters ν (top) and U_{lid} (bottom) under different Reynolds numbers. The left column compares the inversion accuracy of PINN, PINN-SRT, and PINN-MRT models, while the right column shows the robustness of PINN-MRT under noisy data. The shaded areas indicate $\pm 5\%$ uncertainty bands.

4.2. Forward Problem

4.2.1. Flow Field Analysis

The forward problem focuses on evaluating the capability of each model to predict steady-state flow fields given known physical parameters. In this scenario, no additional data-driven constraints are introduced. Comparative evaluations are conducted at $Re = 100$ and 400 for all models (PINN, PINN-SRT, and PINN-MRT). For higher Re ($Re > 400$), only the PINN-MRT model is considered, as the other two models fail to produce reliable predictions under these more challenging conditions.

(A) Vortex Structure Reconstruction and Error Distribution

As shown in Figure 9, at $Re = 100$, the absolute error distributions demonstrate that the PINN-MRT model achieves the lowest error levels, highlighting its superior accuracy under low Re conditions.

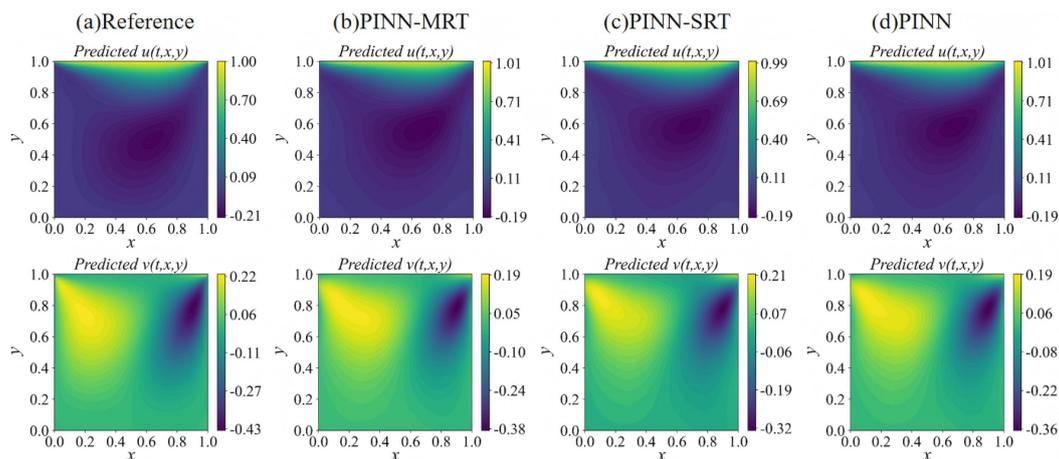


Figure 8. Forward problem solution comparison of velocity field at $Re = 100$.

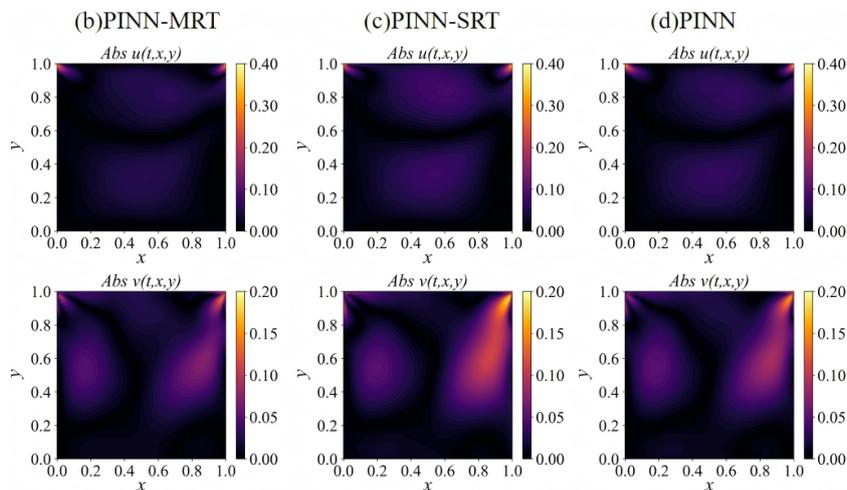


Figure 9. Absolute error distributions of the velocity components u and v at $Re = 100$.

At $Re = 400$ (Figure 11), the absolute error distributions clearly illustrate significant performance differences among the models. Compared to $Re = 100$, the errors of the PINN-MRT model show a slight increase. In contrast, both the standard PINN and PINN-SRT models have completely failed.

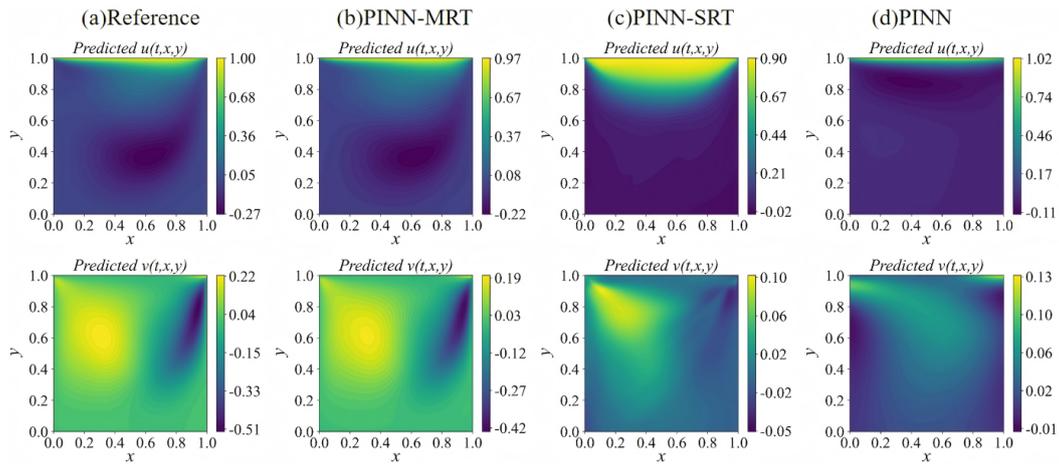


Figure 10. Forward problem solution comparison of velocity field at $Re = 400$.

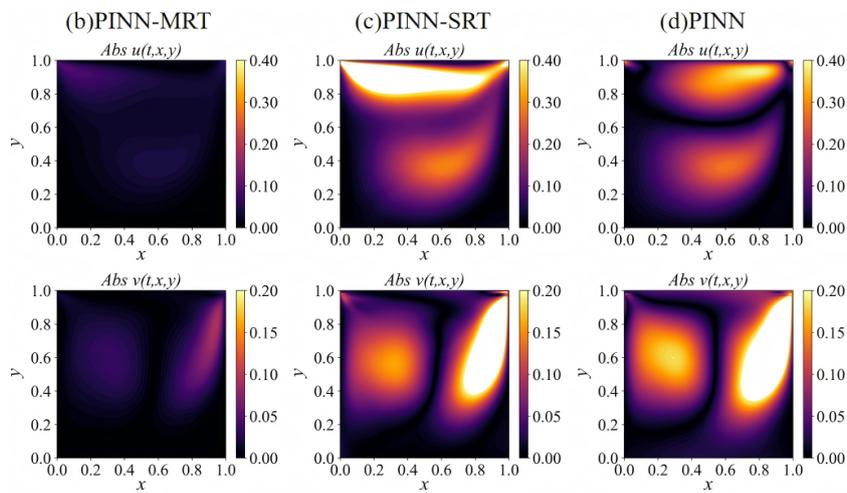


Figure 11. Absolute error distributions of the velocity components u and v at $Re = 400$.

(B) Velocity Profile

As summarized in Table 5, at $Re = 100$, both the standard PINN and PINN-SRT models exhibit error ranges in Δu and Δv approximately twice as large as those of PINN-MRT. When Re increases to 400, the Δu of PINN reaches nearly three times, and that of PINN-SRT about five times, compared to PINN-MRT. For Δv , the errors are even more pronounced, with PINN and PINN-SRT reaching approximately seven and six times larger than PINN-MRT, respectively.

Table 5. Absolute error ranges of horizontal (Δu) and vertical (Δv) velocity components for all three models in forward problem at $Re = 100$ and 400.

Model	Δu	Δv
Re = 100		
PINN	[0.0001, 0.0765]	[0.0000, 0.0248]
PINN-SRT	[0.0004, 0.0851]	[0.0005, 0.1050]
PINN-MRT	[0.0003, 0.0447]	[0.0004, 0.0310]
Re = 400		
PINN	[0.0002, 0.3099]	[0.0059, 0.3688]
PINN-SRT	[0.0041, 0.5389]	[0.0003, 0.3197]
PINN-MRT	[0.0003, 0.1114]	[0.0004, 0.0508]

(C) High Re Error Evaluation of PINN-MRT

As presented in Table 6, the PINN-MRT model performs robustly across all tested conditions. Notably, the increase in the vertical velocity component error is minimal, only 1.801% across the entire Re range. The horizontal velocity and total velocity component errors exhibit increases of approximately 5.000%. These restrained error increments underscore the model's robust stability and accuracy, even under high Re conditions.

Table 6. Relative L^2 errors (%) of PINN-MRT in forward problems.

PINN-MRT	err_u (%)	err_v (%)	err_{total} (%)
$Re = 100$	13.436	22.014	12.058
$Re = 400$	14.328	22.669	14.145
$Re = 1000$	17.086	22.834	15.910
$Re = 2000$	17.152	22.962	16.691
$Re = 5000$	18.003	23.815	17.076

4.2.2. Viscosity Sensitivity

To further evaluate the stability of the model with respect to parameter perturbations, a sensitivity study was conducted by introducing controlled variations in the kinematic viscosity coefficient ν . Figure 12 presents the relative L^2 errors of the reconstructed velocity fields (u , v , and total) under viscosity perturbations ranging from 0% to 20% for different Reynolds numbers. The shaded regions represent $\pm 3\%$ uncertainty bands.

Across all cases, the PINN-MRT model maintains stable error levels, with total relative errors remaining within the $\pm 3\%$ range, indicating low sensitivity to variations in ν . The v -component shows relatively higher errors than u , particularly at higher Reynolds numbers. This behavior is attributed to the weak vertical velocity magnitude and the strong secondary vortex effects in the lid-driven cavity flow, which amplify relative deviations when minor physical perturbations occur. Overall, the results confirm that the proposed PINN-MRT framework achieves consistent flow reconstruction accuracy and robustness against viscosity perturbations in the forward modeling task.

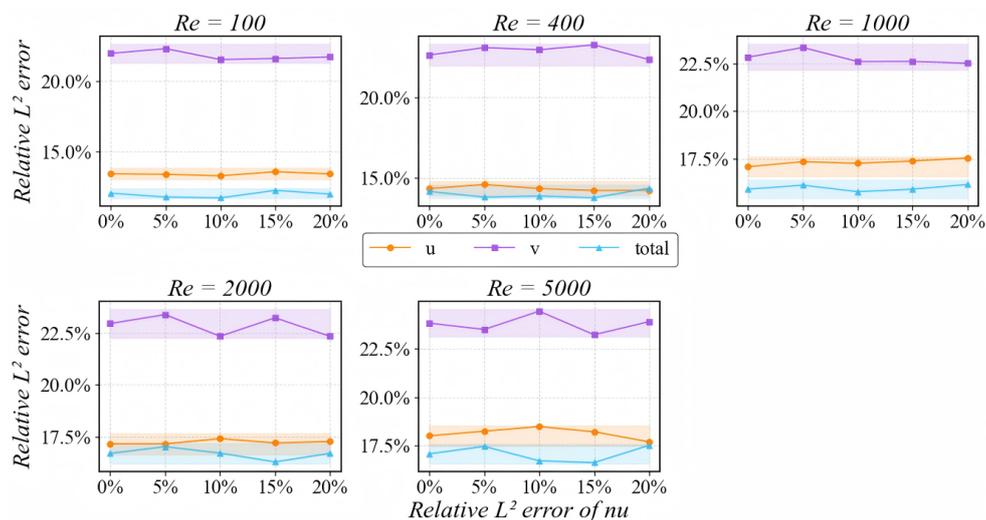


Figure 12. Relative L^2 errors of reconstructed velocity components under kinematic viscosity perturbations from 0% to 20% for different Reynolds numbers. The shaded regions denote $\pm 3\%$ uncertainty bands, representing the error tolerance range of the PINN-MRT model.

5. Conclusions

In this study, the proposed PINN-MRT model is systematically applied to both forward and inverse problems, covering Re ranging from 100 to 5000. For inverse problems, it achieves high-fidelity parameter identification and flow field reconstruction, with remarkable robustness. For forward problems, the architecture exhibits outstanding predictive accuracy, precisely capturing intricate

flow structures and velocity profiles. In addition, a sensitivity analysis on the kinematic viscosity ν demonstrates that the model maintains stable convergence under $\pm 3\%$ perturbations, confirming its insensitivity to small variations in physical parameters. These results highlight the robustness and predictive accuracy of the PINN-MRT framework across a broad range of flow conditions, including challenging high Re scenarios.

The primary contribution of this study is the introduction of the MRT collision mechanism into the PINN architecture. By assigning independent relaxation factors to different moment components, this approach effectively mitigates the critical trade-offs between stability, accuracy, and generalization that often constrains conventional PINN methods. This leads to a significant enhancement in the overall robustness and convergence of the training process.

It is worth noting that the current implementation employs basic fully connected networks, simple optimization strategies, and the fundamental MRT formulation.

In future work, this framework can be further improved by leveraging advanced neural architectures and more efficient training strategies to enhance convergence and scalability. For example, adaptive weighting mechanisms and dynamic optimization schemes will be introduced to better balance multiple loss components and improve numerical stability. Exploring advanced PINN variants, such as XPINNs [35], PDPINNs [36], and GPINNs [37], will also extend the applicability of the framework to high-dimensional and noisy flow systems. Furthermore, the PINN-MRT architecture can be extended through integrating specialized physics modules, such as pseudopotential models for multiphase and multicomponent flows, thermal formulations for heat transfer problems, and advanced boundary schemes for complex geometries. These extensions are expected to improve the capability of the framework to resolve complex interactions such as multiphase interfaces, thermal effects, and multiscale dynamics, while the pursuit of higher generalizability remains a key objective. In summary, the PINN-MRT model can be viewed as a bridge between LBM and deep learning, offering a promising foundation upon which more sophisticated and robust hybrid models can be built in future research.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors are grateful for the valuable comments and suggestions from the respected reviewers, which have greatly enhanced the quality and significance of this work. Financial support from the National Natural Science Foundation of China (Grant Nos. 12474453, 12174085, and 12404530) is also gratefully acknowledged.

Conflicts of Interest: The authors have no conflicts to disclose.

References

1. Runchal, A.K. Evolution of CFD as an engineering science. A personal perspective with emphasis on the finite volume method. *Comptes Rendus. Mécanique* **2022**, *350*, 233–258.
2. Ranganathan, P.; Pandey, A.K.; Sirohi, R.; Hoang, A.T.; Kim, S.H. Recent advances in computational fluid dynamics (CFD) modelling of photobioreactors: Design and applications. *Bioresource technology* **2022**, *350*, 126920.
3. Lee, S.; Zhao, Y.; Luo, J.; Zou, J.; Zhang, J.; Zheng, Y.; Zhang, Y. A review of flow control strategies for supersonic/hypersonic fluid dynamics. *Aerospace Research Communications* **2024**, *2*, 13149.
4. Tu, J.; Yeoh, G.H.; Liu, C.; Tao, Y. *Computational fluid dynamics: a practical approach*; Elsevier, 2023.
5. Hafeez, M.B.; Krawczuk, M. A Review: Applications of the Spectral Finite Element Method: MB Hafeez and M. Krawczuk. *Archives of Computational Methods in Engineering* **2023**, *30*, 3453–3465.
6. Xu, H.; Cantwell, C.D.; Monteserin, C.; Eskilsson, C.; Engsig-Karup, A.P.; Sherwin, S.J. Spectral/hp element methods: Recent developments, applications, and perspectives. *Journal of Hydrodynamics* **2018**, *30*, 1–22.
7. Droniou, J. Finite volume schemes for diffusion equations: introduction to and review of modern methods. *Mathematical Models and Methods in Applied Sciences* **2014**, *24*, 1575–1619.
8. 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. *Journal of Computational physics* **2019**, *378*, 686–707.

9. Rui, E.Z.; Zeng, G.Z.; Ni, Y.Q.; Chen, Z.W.; Hao, S. Time-averaged flow field reconstruction based on a multifidelity model using physics-informed neural network (PINN) and nonlinear information fusion. *International Journal of Numerical Methods for Heat & Fluid Flow* **2024**, *34*, 131–149.
10. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nature Reviews Physics* **2021**, *3*, 422–440.
11. Liu, B.; Wei, J.; Kang, L.; Liu, Y.; Rao, X. Physics-informed neural network (PINNs) for convection equations in polymer flooding reservoirs. *Physics of Fluids* **2025**, *37*.
12. Wong, J.C.; Gupta, A.; Ong, Y.S. Can transfer neuroevolution tractably solve your differential equations? *IEEE Computational Intelligence Magazine* **2021**, *16*, 14–30.
13. Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; Kumar, V. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys* **2022**, *55*, 1–37.
14. Lu, L.; Meng, X.; Mao, Z.; Karniadakis, G.E. DeepXDE: A deep learning library for solving differential equations. *SIAM review* **2021**, *63*, 208–228.
15. Lee, J.; Shin, S.; Kim, T.; Park, B.; Choi, H.; Lee, A.; Choi, M.; Lee, S. Physics informed neural networks for fluid flow analysis with repetitive parameter initialization. *Scientific Reports* **2025**, *15*, 16740.
16. d’Humières, D. Generalized lattice-Boltzmann equations. *Rarefied gas dynamics* **1992**.
17. Guo, Z.; Shi, B.; Wang, N. Lattice BGK model for incompressible Navier–Stokes equation. *Journal of Computational Physics* **2000**, *165*, 288–306.
18. Chen, S.; Doolen, G.D. Lattice Boltzmann method for fluid flows. *Annual review of fluid mechanics* **1998**, *30*, 329–364.
19. Chen, Z.; Liu, Y.; Sun, H. Physics-informed learning of governing equations from scarce data. *Nature communications* **2021**, *12*, 6136.
20. Lou, Q.; Meng, X.; Karniadakis, G.E. Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann-BGK formulation. *Journal of Computational Physics* **2021**, *447*, 110676.
21. Shi, X.; Huang, X.; Zheng, Y.; Ji, T. A hybrid algorithm of lattice Boltzmann method and finite difference-based lattice Boltzmann method for viscous flows. *International Journal for Numerical Methods in Fluids* **2017**, *85*, 641–661.
22. Cheng, X.; Su, R.; Shen, X.; Deng, T.; Zhang, D.; Chang, D.; Zhang, B.; Qiu, S. Modeling of indoor airflow around thermal manikins by multiple-relaxation-time lattice Boltzmann method with LES approaches. *Numerical Heat Transfer, Part A: Applications* **2020**, *77*, 215–231.
23. Zheng, J.; Zha, Y.; Feng, M.; Shan, M.; Yang, Y.; Yin, C.; Han, Q. Numerical investigation on jet-enhancement effect and interaction of out-of-phase cavitation bubbles excited by thermal nucleation. *Ultrasonics Sonochemistry* **2025**, p. 107365.
24. Chai, Z.; Shi, B.; Guo, Z. A multiple-relaxation-time lattice Boltzmann model for general nonlinear anisotropic convection–diffusion equations. *Journal of Scientific Computing* **2016**, *69*, 355–390.
25. Krüger, T.; Kusumaatmaja, H.; Kuzmin, A.; Shardt, O.; Silva, G.; Viggen, E.M. *The lattice Boltzmann method*; Vol. 10, Springer, 2017.
26. Yang, Y.; Tu, J.; Shan, M.; Zhang, Z.; Chen, C.; Li, H. Acoustic cavitation dynamics of bubble clusters near solid wall: A multiphase lattice Boltzmann approach. *Ultrasonics Sonochemistry* **2025**, *114*, 107261.
27. Shan, M.; Zha, Y.; Yang, Y.; Yang, C.; Yin, C.; Han, Q. Morphological characteristics and cleaning effects of collapsing cavitation bubble in fractal cracks. *Physics of Fluids* **2024**, *36*.
28. Zhao, B.; Sun, D.; Wu, H.; Qin, C.; Fei, Q. Physics-informed neural networks for solving inverse problems in phase field models. *Neural Networks* **2025**, p. 107665.
29. Jagtap, A.D.; Mao, Z.; Adams, N.; Karniadakis, G.E. Physics-informed neural networks for inverse problems in supersonic flows. *Journal of Computational Physics* **2022**, *466*, 111402. <https://doi.org/10.1016/j.jcp.2022.111402>.
30. Lu, L.; Pestourie, R.; Yao, W.; Wang, Z.; Verdugo, F.; Johnson, S.G. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing* **2021**, *43*, B1105–B1132.
31. Jin, X.; Cai, S.; Li, H.; Karniadakis, G.E. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics* **2021**, *426*, 109951. <https://doi.org/10.1016/j.jcp.2020.109951>.
32. Lallemand, P.; Luo, L.S. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E* **2000**, *61*, 6546–6562. <https://doi.org/10.1103/PhysRevE.61.6546>.
33. Markidis, S. The Old and the New: Can Physics-Informed Deep-Learning Replace Traditional Linear Solvers?, 2021, [arXiv:math.NA/2103.09655].

34. Hao, Z.; Yao, J.; Su, C.; Su, H.; Wang, Z.; Lu, F.; Xia, Z.; Zhang, Y.; Liu, S.; Lu, L.; et al. PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs, 2023, [[arXiv:cs.LG/2306.08827](https://arxiv.org/abs/cs.LG/2306.08827)].
35. Jagtap, A.D.; Karniadakis, G.E. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics* **2020**, *28*.
36. Peng, W.; Zhou, W.; Zhang, J.; Yao, W. Accelerating physics-informed neural network training with prior dictionaries. *arXiv preprint arXiv:2004.08151* **2020**.
37. Yu, J.; Lu, L.; Meng, X.; Karniadakis, G.E. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering* **2022**, *393*, 114823.

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.