

Improved physics-informed neural networks incorporating lattice Boltzmann method optimized by tanh robust weight initialization

Chenghui Yang(杨程晖), Minglei Shan(单鸣雷), Mengyu Feng(冯梦宇), Ling Kuai(蒯玲), Yu Yang(杨雨), Cheng Yin(殷澄), and Qingbang Han(韩庆邦)

Citation: Chin. Phys. B, 2025, 34 (**11**): 110701. DOI: 10.1088/1674-1056/adfc43
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Inverse design of directional hybrid nanoantennas using neural networks

Ru-Lin Guan(管如林), Deng-Chao Huang(黄登朝), Ya-Qiong Li(李雅琼), Chen Wang(王晨), and Bin-Zi Xu(徐彬梓)
Chin. Phys. B, 2025, 34 (**9**): 090702. DOI: 10.1088/1674-1056/adeb5c

Graph neural networks unveil universal dynamics in directed percolation

Ji-Hui Han(韩继辉), Cheng-Yi Zhang(张程义), Gao-Gao Dong(董高高), Yue-Feng Shi(石月凤), Long-Feng Zhao(赵龙峰), and Yi-Jiang Zou(邹以江)
Chin. Phys. B, 2025, 34 (**8**): 080702. DOI: 10.1088/1674-1056/add505

Analysis and design of multivalued many-to-one associative memory driven by external inputs

Qiang Fang(方强) and Hao Zhang(张浩)
Chin. Phys. B, 2025, 34 (**8**): 080701. DOI: 10.1088/1674-1056/add24b

Deep learning-enabled inverse design of polarization-selective structural color based on coding metasurface

Haolin Yang(杨昊霖), Bo Ni(倪波), Junhong Guo(郭俊宏), Hua Zhou(周华), and Jianhua Chang(常建华)
Chin. Phys. B, 2025, 34 (**5**): 050702. DOI: 10.1088/1674-1056/adb94a

Text-guided diverse-expression diffusion model for molecule generation

Wenchao Weng(翁文超), Hanyu Jiang(蒋涵羽), Xiangjie Kong(孔祥杰), and Giovanni Pau
Chin. Phys. B, 2025, 34 (**5**): 050701. DOI: 10.1088/1674-1056/adbedd

---

COMPUTATIONAL PROGRAMS FOR PHYSICS

Improved physics-informed neural networks incorporating lattice Boltzmann method optimized by tanh robust weight initialization

Chenghui Yang(杨程晖)[1], Minglei Shan(单鸣雷)[1,†], Mengyu Feng(冯梦宇)[1], Ling Kuai(蒯玲)[1], Yu Yang(杨雨)[2], Cheng Yin(殷澄)[1], and Qingbang Han(韩庆邦)[1]

[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*

(Received 18 June 2025; revised manuscript received 22 July 2025; accepted manuscript online 18 August 2025)

Physics-informed neural networks (PINNs) have shown considerable promise for performing numerical simulations in fluid mechanics. They provide mesh-free, end-to-end approaches by embedding physical laws into their loss functions. However, when addressing complex flow problems, PINNs still face some challenges such as activation saturation and vanishing gradients in deep network training, leading to slow convergence and insufficient prediction accuracy. We present physics-informed neural networks incorporating lattice Boltzmann method optimized by tanh robust weight initialization (T-PINN-LBM) to address these challenges. This approach fuses the mesoscopic lattice Boltzmann model with the automatic differentiation framework of PINNs. It also implements a tanh robust weight initialization method derived from fixed point analysis. This model effectively mitigates activation and gradient decay in deep networks, improving convergence speed and data efficiency in multiscale flow simulations. We validate the effectiveness of the model on the classical arithmetic example of lid-driven cavity flow. Compared to the traditional Xavier initialized PINN and PINN-LBM, T-PINN-LBM reduces the mean absolute error (MAE) by one order of magnitude at the same network depth and maintains stable convergence in deeper networks. The results demonstrate that this model can accurately capture complex flow structures without prior data, providing a new feasible pathway for data-free driven fluid simulation.

Keywords: lattice Boltzmann method, physical-informed neural networks, fluid mechanics, tanh robust weight initialization

PACS: 07.05.Mh, 02.60.Cb, 02.30.Jr, 84.35.+i

DOI: 10.1088/1674-1056/adfc43 **CSTR:** 32038.14.CPB.adfc43

1. Introduction

Numerical simulation in fluid mechanics fulfills an indispensable role in engineering design, environmental science, and fundamental research. Traditional numerical methods include the finite element method (FEM),[1] finite difference method (FDM),[2] and finite volume method (FVM).[3] These approaches consistently demonstrate robust performance across diverse applications. However, they also face challenges including complex mesh generation, high computational costs, and difficulties in handling multiphysics coupling. In recent years, physics-informed neural networks (PINNs) have emerged as an innovative computational paradigm.[4] It integrates physical laws into the loss functions, enabling mesh-free, end-to-end simulations. This approach not only handles complex boundary conditions with greater flexibility but also seamlessly integrates multiphysics problems, offering new possibilities for research in fluid mechanics.[5–16]

However, PINNs still face significant limitations in practical applications. On one hand, deep neural networks are prone to issues such as activation function saturation and vanishing gradients during training, resulting in slow convergence and insufficient prediction accuracy. On the other hand, under challenging conditions such as high Reynolds number flows, PINNs often suffer from stability issues, making it difficult to accurately capture fine-scale flow features.

Meanwhile, the lattice Boltzmann method (LBM) is a mesoscopic dynamics-based numerical simulation method that recovers the macroscopic fluid behavior by solving the evolution equations of discrete distribution functions.[17] LBM is characterized by simple algorithms, high parallelization efficiency, and the straightforward handling of complex boundary conditions. It has been widely adopted in fluid mechanics applications, including turbulence simulation, multiphase flow, and microfluidics. Incorporating PINNs and LBM leverages the strengths of both methods. On the one hand, PINNs can directly model the residuals of the physical equations. On the other hand, LBM provides mesoscopic flow field information to achieve multi-scale supervision. This kind of PINN-LBM framework combines physical consistency and computational efficiency, and is expected to accurately reconstruct complex flow fields without the need for fine meshing.

Although both PINNs and LBM have witnessed rapid development, research on integrating the two is still in its in-

[†]Corresponding author. E-mail: shanml@hhu.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

fancy. Lou *et al.*[18] applied PINNs to solve the Boltzmann-BGK equations, which reduced mesh dependence and lowered computational cost. Liu *et al.*[19] incorporated the single-relaxation-time lattice Boltzmann method (SRT-LBM) with PINNs to address inverse problems in fluid dynamics, which improved the accuracy and computational efficiency and demonstrated the potential for practical applications. Fang *et al.*[20] proposed the coupling framework of immersed boundary-LBM (IB-LBM) and PINNs for fluid-solid coupling simulation. Currently, most implementations of PINN-LBM models rely on simulated or experimental data to assist training in order to mitigate the training instability caused by insufficient physical residual information. In addition, there are still challenges in terms of stability and initialization strategies for network training. In particular, when the network employs a hyperbolic tangent (tanh) activation function, the common Xavier initialization often leads to deep network activation saturation, resulting in gradient vanishing, which reduces the convergence speed and prediction accuracy.

In order to solve the above problems, this study proposes the T-PINN-LBM model, which integrates tanh robust weight initialization based on fixed point analysis (TRWI) into the PINN-LBM model. The main innovations of this study are that the limitations of traditional methods are overcome by combining the advantages of PINNs and LBM, and the activation function saturation and gradient decay problems can be effectively alleviated by using the TRWI method. Specifically, the T-PINN-LBM model uses the physical constraints of PINNs during the training process to ensure the physical consistency of the hydrodynamic equations, while the mesoscopic formulation of LBM is utilized to enhance the model's generalization ability, thus enabling effective solution of hydrodynamic problems without relying on a large amount of experimental data. Furthermore, TRWI is implemented across all hidden layers to ensure that the tanh activation function oscillates around nonzero fixed points and avoids saturation regions, thereby substantially alleviating activation and gradient issues in deep networks and achieving faster convergence and lower generalization errors, particularly in deep architectures with limited data. Taking the classical lid-driven cavity flow as a benchmark case, the T-PINN-LBM model demonstrates superior accuracy and stability compared to PINNs and PINN-LBM models with Xavier initialization, especially under high Reynolds number conditions.

The structure of this paper is as follows. Section 2 elucidates the principles of PINNs and LBM, detailing their integration within the proposed framework. Section 3 details the experimental setup and results. T-PINN-LBM is applied to the lid-driven cavity flow without any simulation data. We evaluate the robustness of the TRWI method, compare the accuracy of T-PINN-LBM with baseline methods, and assess its stability at high Reynolds numbers. Section 4 concludes with a summary and future perspectives.

2. Method

2.1. Lattice Boltzmann method

The Boltzmann equation is the fundamental equation of kinetics. Since the Boltzmann equation is a complex differential integral equation, its direct solution is impossible, so many simplified equations have been proposed, and Boltzmann–Bhatnagar–Gross–Krook (BGK) is one of the most commonly used simplified forms of the Boltzmann equation, which is as follows:[21]

$$\begin{aligned} &\frac{\partial f}{\partial t}+\xi\cdot\nabla f=-\frac{(f-f^{\mathrm{eq}})}{\tau_f},\\ &f^{\mathrm{neq}}=f-f^{\mathrm{eq}}. \end{aligned} \tag{1}$$

The particle distribution function is denoted as $f=f(x,\xi,t)$, where ξ represents the microscopic velocity and x indicates the spatial position. The right-hand side is the simplified BGK collision operator, with τ_f representing the relaxation time, f^{neq} denoting the nonequilibrium distribution function, and f^{eq} signifying the equilibrium distribution function, typically characterized by a continuous Maxwell–Boltzmann distribution function as[22]

$$f^{\mathrm{eq}}=\frac{\rho}{(2\pi RT)^{D/2}}\exp\left[-\frac{|\boldsymbol{\xi}-\boldsymbol{u}|^2}{2RT}\right], \tag{2}$$

where R is the gas constant, D is the spatial dimension, ρ is the density, T is the temperature, and $\boldsymbol{u}$ is the macroscopic velocity. He and Luo[23] published a series of articles in 1997, examining the intrinsic relationship between the lattice Boltzmann equation (LBE) and the Boltzmann equation, and demonstrating that the LBE can be considered a specialized discrete formulation of the Boltzmann-BGK model. By discretizing the microscopic velocity ξ into a set of lattice velocities e_α $(\alpha=0,1,\ldots,n-1)$, the discrete Boltzmann-BGK equation is derived as[24]

$$\frac{\partial f_\alpha}{\partial t}+e_\alpha\cdot\nabla f=-\frac{1}{\tau_f}\left(f_\alpha-f_\alpha^{\mathrm{eq}}\right), \tag{3}$$

where f_α is a discrete distribution function and f_α^{eq} is a discrete equilibrium distribution function. Integrating both sides of Eq. (3) with respect to time yields the LBE, also known as the standard LBE, and it is represented as[23]

$$f_\alpha\left(x+e_\alpha\delta_t,t+\delta_t\right)-f_\alpha(x,t)=-\frac{1}{\tau}\left(f_\alpha-f_\alpha^{\mathrm{eq}}\right), \tag{4}$$

where $\tau=\tau_f/\delta_t$ is the dimensionless relaxation time. Because there is only one relaxation time τ in Eq. (4), Eq. (4) is also known as the single-relaxation-time (SRT) lattice Boltzmann equation.

In this study, the D2Q9 discrete velocity model[21] is used, which is a two-dimensional nine-speed lattice, $\boldsymbol{e_\alpha}$ can be expressed as

$$\boldsymbol{e_\alpha} = \begin{cases} (0,0) & \alpha = 0, \\ c\left(\cos\left[(\alpha-1)\frac{\pi}{2}\right], \sin\left[(\alpha-1)\frac{\pi}{2}\right]\right) & \alpha = 1,2,3,4, \\ \sqrt{2}c\left(\cos\left[(2\alpha-1)\frac{\pi}{4}\right], \sin\left[(2\alpha-1)\frac{\pi}{4}\right]\right) & \alpha = 5,6,7,8, \end{cases} \quad (5)$$

where $\boldsymbol{e_\alpha}$ represents the direction of velocity discretization, and $c = \delta x/\delta t$ denotes the lattice velocity, with δx and δt being grid spacing and time step, respectively.

The equilibrium distribution function can be expressed as

$$f_\alpha^{\mathrm{eq}} = \omega_\alpha \rho \left[1 + \frac{3}{c^2}(\boldsymbol{e_\alpha}\cdot\boldsymbol{u}) + \frac{9}{2c^4}(\boldsymbol{e_\alpha}\cdot\boldsymbol{u})^2 - \frac{3}{2c^2}\boldsymbol{u}^2\right], \quad (6)$$

where $\boldsymbol{u}$ refers to the fluid velocity, c stands for the lattice speed, and the weighting coefficients ω_α are $\omega_0 = 4/9$, $\omega_{1-4} = 1/9$, $\omega_{5-8} = 1/36$. The local macroscopic density ρ and the macroscopic fluid velocity $\boldsymbol{u}$ can be expressed as

$$\rho = \sum_\alpha f_\alpha(x,t),$$

$$\rho \boldsymbol{u} = \sum_\alpha \boldsymbol{e_\alpha} f_\alpha(x,t). \quad (7)$$

2.2. Physical-informed neural networks combined with LBM

This section outlines the integration of LBM into a PINNs framework to address fluid problems within a mesoscopic context, resulting in the proposed T-PINN-LBM approach. It then details the boundary conditions, loss function, and optimization method employed in the study.

2.2.1. T-PINN-LBM

T-PINN-LBM constructs a mesh-free flow field simulation framework by deeply integrating the physical constraints of PINNs with the mesoscopic characteristics of LBM. Unlike standard PINN-LBM architectures, T-PINN-LBM incorporates the TRWI method, which significantly enhances the training stability and convergence efficiency of deep neural networks.

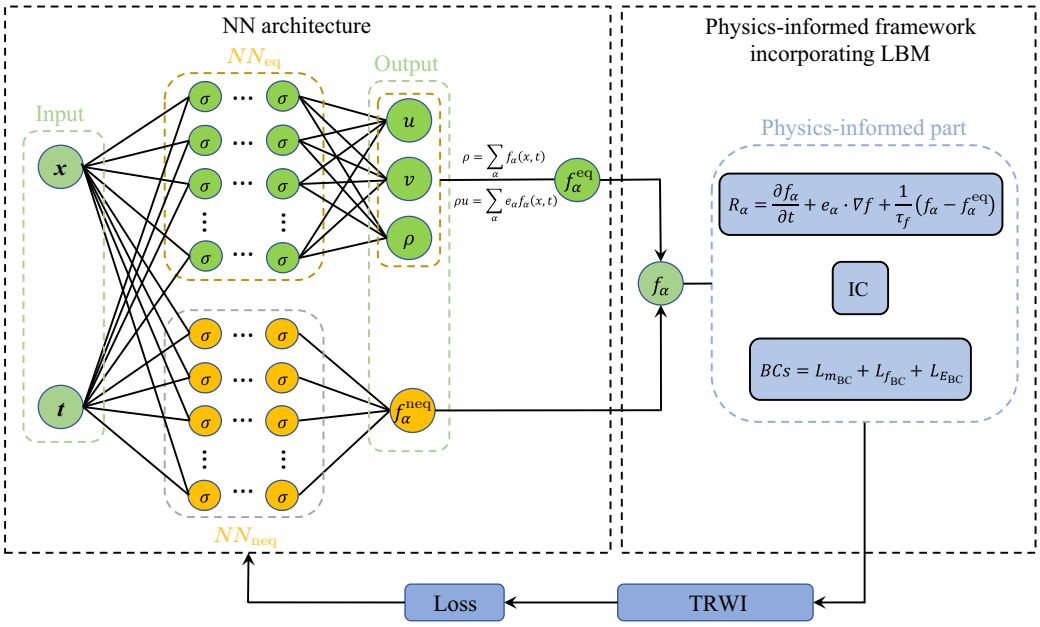


Fig. 1. The framework of T-PINN-LBM.

As illustrated in Fig. 1, the T-PINN-LBM model consists of two main components: (1) a neural network component, and (2) a physics-informed component incorporating LBM. The neural network structure of the T-PINN-LBM model takes spatial coordinates x and time t as inputs, and employs two deep neural networks to output approximate equilibrium distribution function f_α^{eq} and approximate nonequilibrium distribution function f_α^{neq}, respectively. Both networks use the hyperbolic tangent (tanh) as the activation function and are parameterized with adjustable depth and width. Their outputs provide the predicted distribution functions at each spatiotemporal input, serving as inputs for the physics-informed residual computation. To mitigate the vanishing gradient problem commonly observed in deep tanh networks, the TRWI method is integrated into the PINN-LBM framework by initializing the weights of the neural networks predicting the equilibrium and nonequilibrium distribution functions, which ensures that the activation values remain outside the saturation regions of the tanh function. In contrast, standard PINN-LBM models typically rely on Xavier initialization, which lacks optimization for deep tanh networks and often leads to poor convergence performance when simulating complex flow dynamics.

The physics-informed LBM component of T-PINN-LBM uses a discrete velocity model (DVM) to incorporate the LBM physics. It takes the neural network outputs and, according to Eq. (7), converts the predicted equilibrium and nonequilib-

rium distribution functions into the macroscopic quantities u, v, and ρ. These macroscopic fields are then used to compute the residuals of the governing equations via automatic differentiation (Autograd), forming the physics-based loss terms. LBM plays a critical role by linking the neural network's output to macroscopic quantities via the discrete velocity model, as shown in Eq. (7). This mesoscopic information introduces physical consistency, improving the model's stability, accuracy and universality, especially for complex flows. By integrating LBM, T-PINN-LBM leverages multi-scale supervision, enabling robust reconstruction of flow fields without requiring extensive training data. The training process iteratively updates the network weights to minimize these residuals, and continues until a maximum number of iterations is reached. The PDE residual is defined using the Eq. (3). It is worth noting that the output dimensionality of the neural networks must be configured according to the chosen DVM. For example, if a D2Q9 DVM is used, then the equilibrium and nonequilibrium distribution outputs should each have 9 components corresponding to the 9 discrete velocity directions. In addition, if other DVMs are desired, it is sufficient to change the corresponding variables in the residual equation.

2.2.2. Tanh robust weight initialization method

To alleviate the activation/gradient decay problem prevalent in deep tanh networks, the TRWI method proposed by Lee *et al.*[25] is employed. This method improves the efficiency of signal propagation by controlling the distribution of weights to ensure that the activation values do not enter the saturation region. Specifically, the initialization weights follow a Gaussian distribution as

$$\boldsymbol{W}_\ell = \boldsymbol{D}_\ell + \boldsymbol{Z}_\ell, \tag{8}$$

where $\boldsymbol{D}_\ell$ is a diagonal matrix with ones on the diagonal and zeros in the off-diagonal positions, ensuring a uniform distribution of the weights. $\boldsymbol{Z}_\ell$ represents a Gaussian noise matrix, with its elements following the $N\left(0,\sigma_z^2\right)$ distribution, where σ_z denotes the standard deviation of the noise. The standard deviation is related to the number of layers and nodes in the network as follows:

$$\sigma_z = \frac{\alpha}{\sqrt{N_{\ell-1}}}, \tag{9}$$

where $N_{\ell-1}$ denotes the number of neurons in the preceding layer, and α is a constant typically set to 0.085. With this approach, the network can achieve faster convergence and higher stability without additional normalization, especially in the case of deeper networks and limited data.

2.2.3. Loss functions and optimization methods

In addressing hydrodynamic problems, assuming known initial and boundary conditions as well as governing equations, the physical constraints of LBM are enforced during training via minimization of the following loss function:

$$\begin{aligned}
L &= L_{\mathrm{Eq}} + L_{\mathrm{IC}} + L_{\mathrm{BC}},\\
L_{\mathrm{Eq}} &= \frac{1}{N_{\mathrm{e}}}\sum_{\alpha=0}^{Q-1}\sum_{j=0}^{N_{\mathrm{e}}-1}\left|R_\alpha\left(x_j,t_j\right)\right|^2,\\
L_{\mathrm{IC}} &= \frac{1}{N_{\mathrm{i}}}\sum_{j=0}^{N_{\mathrm{i}}-1}\left|\phi_{NN_{\mathrm{eq}}}\left(x_j,0\right)-\phi^*\left(x_j,0\right)\right|^2,\\
L_{\mathrm{BC}} &= \frac{1}{N_{\mathrm{b}}}\sum_{j=0}^{N_{\mathrm{b}}-1}\left|\phi_{NN_{\mathrm{eq}}}\left(x_{\mathrm{b}_j},t_j\right)-\phi^*\left(x_{\mathrm{b}_j},t_j\right)\right|^2,
\end{aligned} \tag{10}$$

where $R_\alpha(x,t)$ is the residual corresponding to the control equation, which can be calculated by Eq. (3). The constructed residuals are calculated as

$$R_\alpha = \frac{\partial f_\alpha}{\partial t} + e_\alpha\cdot\nabla f + \frac{1}{\tau_f}\left(f_\alpha - f_\alpha^{\mathrm{eq}}\right),\quad \alpha = 0,1,\ldots,8. \tag{11}$$

In addition, L_{Eq}, L_{IC}, and L_{BC} denote the loss functions of the residuals, initial conditions, and boundary conditions of T-PINN-LBM. Specifically, ϕ represents the macroscopic variables (u, v, and ρ), while $\phi_{NN_{\mathrm{eq}}}$ and ϕ^* denote the outputs of NN_{eq} and the prescribed initial and boundary conditions, respectively. N_{e}, N_{i} and N_{b} refer to the number of training data points for the control equations, initial conditions, and boundary conditions, respectively. Furthermore, as shown in Fig. 1, L_{BC} in the T-PINN-LBM framework also consists of three terms. $L_{m_{\mathrm{BC}}}$ enforces prescribed macroscopic density and velocity, $L_{f_{\mathrm{BC}}}$ constrains the non-equilibrium distribution function for incoming directions, and $L_{E_{\mathrm{BC}}}$ ensures that outgoing distributions satisfy the steady-state Boltzmann-BGK equation.

During the training process of the neural network model, two distinct optimizers, Adam[26] and L-BFGS,[27] are introduced, both effective in minimizing the loss function and updating model parameters. To accelerate the training process, a combination of Adam and L-BFGS optimizers is utilized in the training phase. During the initial training phase, the Adam optimizer is employed until the loss falls below a predefined threshold; subsequently, optimization proceeds with L-BFGS for up to the specified maximum number of iterations. This strategy leverages the complementary strengths of both optimizers to enhance model convergence.

3. Results and analysis

In this section, the proposed T-PINN-LBM model is evaluated using the classical lid-driven cavity flow benchmark problem in fluid dynamics. We first assess the performance of the proposed tanh-based weight initialization under varying network sizes. Then, using a high-fidelity reference solution as a baseline, we compare the accuracy of T-PINN-LBM against

the vanilla PINN and the baseline PINN-LBM method. Finally, we test the stability of T-PINN-LBM for flows at higher Reynolds numbers, where the vanilla PINN often fails.

3.1. Verification of the robustness of TRWI

As depicted in Fig. 2, the computational domain for the lid-driven cavity flow is defined as a unit square $[0,1]\times[0,1]$. No-slip velocity boundary conditions are imposed on the three stationary walls, while the top lid translates in the x-direction at a constant velocity ($u=1$). The governing equations are shown below:

$$\begin{aligned}
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0,\\
&u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right),\\
&u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right),
\end{aligned}\tag{12}$$

where the boundary condition for the pressure p is a directional derivative of 0 for all 4 walls, the boundary conditions for the vertical velocity v are all 0, and the boundary condition for the horizontal velocity u is 0 for both the left and right lower walls and 1 for the upper wall; the Reynolds number $Re=Lu/\mu=100$ characterizes the flow, where L is the length of the cavity and μ is the shear viscosity of the fluid.

This section presents experimental results demonstrating that the TRWI method provides greater robustness than the Xavier initialization method across various network sizes, while also achieving improved data efficiency. Feed-forward neural networks of different depths and widths were constructed and their training performance evaluated within the PINN-LBM framework. Specifically, networks with 5, 10, 20, 30, 40, 50, 60, and 80 hidden layers were tested, utilizing 4, 8, 16, 32, or 64 neurons per layer for each configuration. In total, this yielded 40 distinct network configurations. For each configuration, the PINN-LBM model was trained to convergence using either the Xavier initialization method or the TRWI method, with the final training loss recorded. All models were trained using the Adam optimizer with a learning rate of 0.001 for 5000 iterations, using a training dataset of 1000 boundary points and 10000 interior collocation points randomly sampled across the domain.

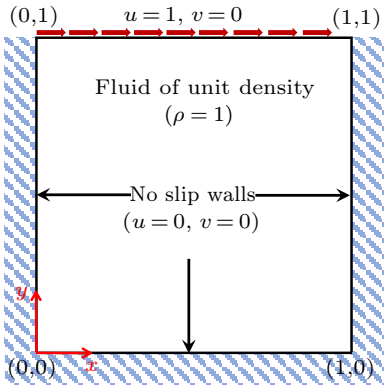


Fig. 2. Schematic diagram of square-cavity flow showing domain and boundary conditions.

Table 1 summarizes the PINN-LBM training loss for different network depths and widths using the Xavier initialization method and TRWI method. Each row corresponds to a different number of hidden layers, and each pair of columns compares Xavier initialization and TWRI for a given number of neurons per layer. As shown in Table 1, the Xavier initialization method tends to produce higher losses as the depth of the network increases. In many cases, it either diverges or yields large errors for very deep networks. In contrast, the TRWI method consistently achieves much lower loss even for very deep networks. In fact, for every tested network configuration, the TRWI method reached the lowest loss, indicating superior robustness. Notably, the proposed method eliminates the need for extensive hyperparameter tuning and additional regularization that the Xavier-initialized networks often require to train deep architectures.

Table 1. PINN-LBM loss for FFNNs with varying neurons and layers using the Xavier initialization method and the TRWI method.

4 neurons	5	10	20	30	40	50	60	80
Xavier	4.15×10^{-1}	6.28×10^{-2}	6.09×10^{-2}	5.53×10^{-2}	4.16×10^{-1}	4.88×10^{-1}	5.00×10^{-1}	5.00×10^{-1}
TRWI	$\mathbf{8.61\times10^{-2}}$	$\mathbf{4.23\times10^{-2}}$	$\mathbf{3.54\times10^{-2}}$	$\mathbf{2.97\times10^{-2}}$	$\mathbf{3.10\times10^{-2}}$	$\mathbf{3.24\times10^{-2}}$	$\mathbf{7.33\times10^{-2}}$	$\mathbf{7.28\times10^{-2}}$
8 neurons	5	10	20	30	40	50	60	80
Xavier	2.30×10^{-1}	8.67×10^{-2}	5.47×10^{-2}	4.48×10^{-1}	5.58×10^{-2}	5.00×10^{-1}	6.91×10^{-2}	5.99×10^{-2}
TRWI	$\mathbf{7.14\times10^{-2}}$	$\mathbf{5.33\times10^{-2}}$	$\mathbf{2.51\times10^{-2}}$	$\mathbf{4.54\times10^{-2}}$	$\mathbf{1.87\times10^{-2}}$	$\mathbf{3.89\times10^{-2}}$	$\mathbf{3.44\times10^{-2}}$	$\mathbf{3.25\times10^{-2}}$
16 neurons	5	10	20	30	40	50	60	80
Xavier	4.44×10^{-2}	3.87×10^{-2}	4.28×10^{-2}	4.83×10^{-2}	4.42×10^{-2}	5.47×10^{-2}	5.00×10^{-1}	5.00×10^{-1}
TRWI	$\mathbf{1.01\times10^{-2}}$	$\mathbf{8.24\times10^{-3}}$	$\mathbf{6.19\times10^{-3}}$	$\mathbf{7.14\times10^{-3}}$	$\mathbf{6.23\times10^{-3}}$	$\mathbf{7.98\times10^{-3}}$	$\mathbf{7.54\times10^{-3}}$	$\mathbf{7.49\times10^{-3}}$
32 neurons	5	10	20	30	40	50	60	80
Xavier	6.14×10^{-2}	5.03×10^{-2}	4.13×10^{-2}	5.59×10^{-2}	4.96×10^{-1}	4.81×10^{-2}	5.00×10^{-1}	5.00×10^{-1}
TRWI	$\mathbf{9.97\times10^{-3}}$	$\mathbf{8.31\times10^{-3}}$	$\mathbf{6.01\times10^{-3}}$	$\mathbf{6.84\times10^{-3}}$	$\mathbf{5.19\times10^{-3}}$	$\mathbf{6.08\times10^{-3}}$	$\mathbf{5.87\times10^{-3}}$	$\mathbf{5.92\times10^{-3}}$
64 neurons	5	10	20	30	40	50	60	80
Xavier	4.96×10^{-2}	4.61×10^{-2}	4.16×10^{-2}	5.65×10^{-2}	4.87×10^{-2}	5.00×10^{-1}	5.00×10^{-1}	5.98×10^{-2}
TRWI	$\mathbf{5.97\times10^{-3}}$	$\mathbf{5.24\times10^{-3}}$	$\mathbf{4.03\times10^{-3}}$	$\mathbf{4.78\times10^{-3}}$	$\mathbf{4.26\times10^{-3}}$	$\mathbf{4.97\times10^{-3}}$	$\mathbf{4.06\times10^{-3}}$	$\mathbf{3.79\times10^{-3}}$

From the results in Table 1, it is evident that PINN-LBM with TRWI method outperforms the Xavier initialization method for all tested network configurations. In many cases, PINN-LBM with Xavier initialization method struggles as the network becomes deeper. The loss increases or even fails to decrease, indicated by the entries around 5.00×10^{-1} which suggest stagnation at a high value. In contrast, the TRWI method achieves consistently low losses for deep networks, indicating stable convergence. This demonstrates the robustness of the TRWI method in enabling training of very deep PINNs models. It effectively removes the additional computational overhead and hyperparameter tuning that would otherwise be required to train deep networks with the Xavier initialization method.

In addition to final losses, the data efficiency of the two initialization methods was evaluated. Specifically, the reduction in mean absolute error (MAE) was analyzed as a function of increasing collocation points. A representative deep neural network, comprising 40 hidden layers with 32 neurons per layer, was employed to model a particularly complex configuration. Figure 3 plots the MAE of the trained models using the Xavier initialization method and the TRWI method against the number of collocation points used in training. Each model was trained with the Adam optimizer for 400 iterations, and the MAE was averaged over 5 independent runs. As shown in Fig. 3, the MAE achieved with the TRWI method is significantly lower than that with the Xavier initialization method for various numbers of collocation points. Notably, as the number of collocation points increases beyond about 5000, the gap in MAE between the two methods becomes even more pronounced. This indicates that the TRWI method not only stabilizes training for deep networks but also makes more efficient use of training data to achieve higher accuracy.

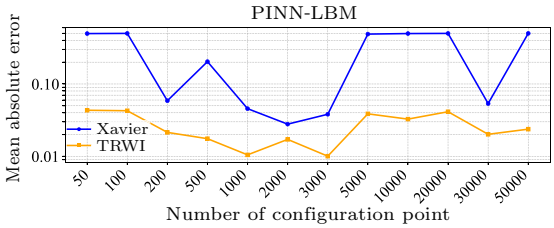


Fig. 3. Mean absolute error for models using Xavier initialization and TRWI with different numbers of configuration points.

3.2. Validation of the accuracy of the T-PINN-LBM model

Next, the accuracy of the T-PINN-LBM model is validated on the lid-driven cavity flow and compared with baseline methods. A network architecture of 6 hidden layers with 60 neurons per layer is selected for all models in this comparison. All models were trained utilizing the Adam optimizer (learning rate 0.001) over 80000 iterations to achieve convergence. The training dataset comprised 1000 boundary points and 20000 interior collocation points. Boundary points were generated through random sampling of 1000 points along each of the four boundaries, while interior points were randomly sampled across the computational domain. No simulation data or reference solution data are used during training.

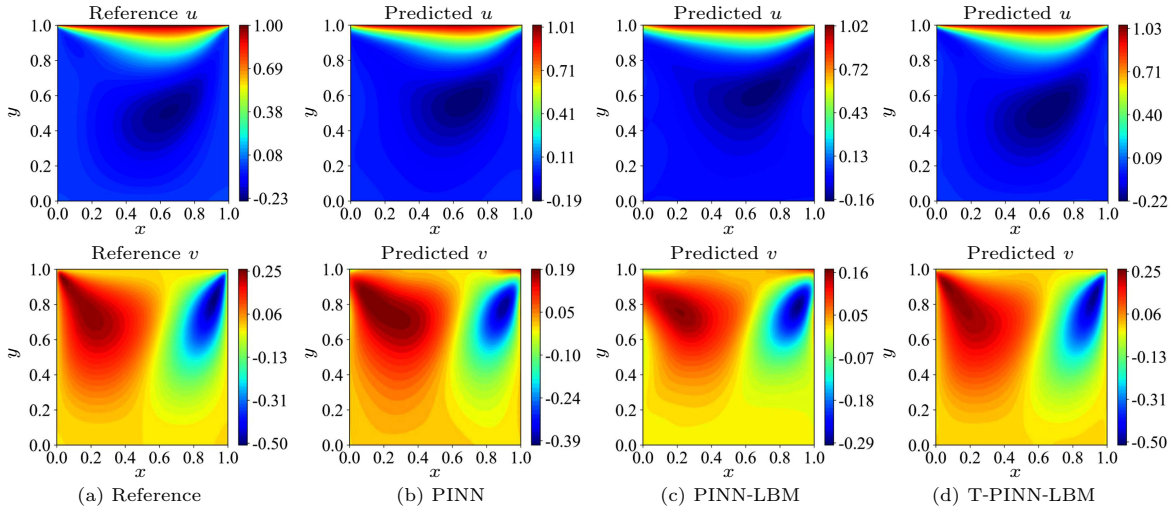


Fig. 4. At $Re = 100$: from left to right, these figures depict (a) the reference solution, the predicted solutions of (b) PINN, (c) PINN-LBM, and (d) T-PINN-LBM of the lid-driven cavity flow.

Figures 4–6 illustrate the comparative results of the PINN, PINN-LBM, and T-PINN-LBM models against the reference solution for the lid-driven cavity flow. These figures present various visualizations of the flow field, such as the reference solution, predicted solutions from different models, pointwise absolute error distributions, and representative velocity profiles. In particular, figure 4 shows the contour plots of the velocity field for the reference solution and the predictions by each model; figure 5 shows the contour plots of the absolute error for each model; and figure 6 compares the horizontal and vertical centerline velocity profiles of each model with the reference solution.

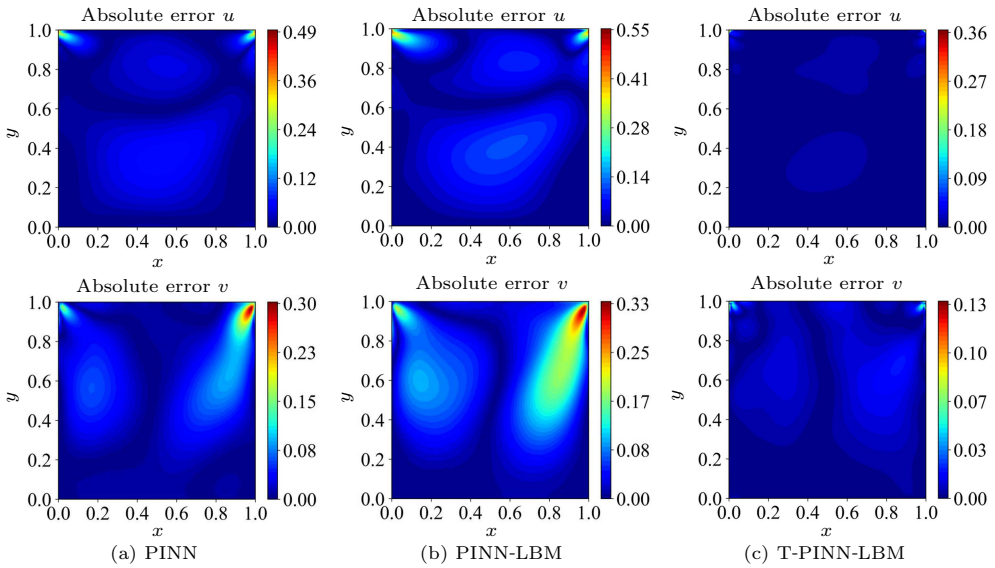


Fig. 5. At $Re = 100$: from left to right, these figures depict the absolute error between the predicted solutions of (a) PINN, (b) PINN-LBM, and (c) T-PINN-LBM and the reference solution of the lid-driven cavity flow.

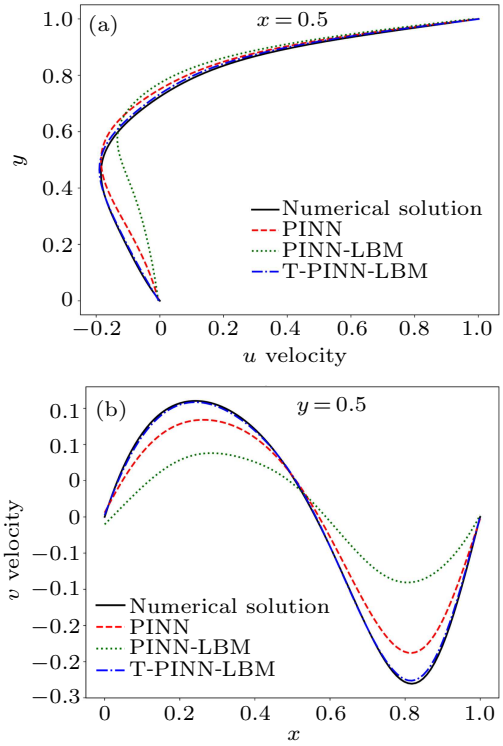


Fig. 6. Comparison of the velocity slices of PINN, PINN-LBM and T-PINN-LBM with the reference solution at position (a) $x = 0.5$ and (b) $y = 0.5$.

After training, the overall relative *L2* error of each model is computed with respect to the reference solution. In this experiment, the resulting relative errors were PINN 1.19×10^{-1}, PINN-LBM 1.01×10^{-1}, T-PINN-LBM 1.45×10^{-2}. Thus, T-PINN-LBM achieves a dramatically lower error compared to PINN and PINN-LBM. This corresponds to roughly an 88% reduction in error relative to PINN-LBM, highlighting the effectiveness of the proposed initialization method and framework. While T-PINN-LBM achieves higher accuracy than PINN, its computational efficiency is impacted by the added complexity of LBM and deeper architectures, potentially increasing training time. However, TRWI improves convergence speed by stabilizing gradient propagation, partially offsetting this cost. Compared to the vanilla PINN, T-PINN-LBM requires more computational resources per iteration but achieves lower errors with fewer iterations in deep networks. Future work will focus on optimizing the model structure and training algorithms to enhance efficiency, potentially through adaptive sampling or parallelization. Furthermore, comparative experiments across various network architectures are conducted to assess the consistency of these results, as detailed in Table 2.

Table 2. Relative error comparison of the lid-driven cavity across various network structures and models.

Network structure	PINN	PINN-LBM	**T-PINN-LBM**
20 units & 2 hidden layers	4.53×10^{-1}	6.47×10^{-1}	$\mathbf{7.42\times10^{-2}}$
40 units & 2 hidden layers	4.46×10^{-1}	5.34×10^{-1}	$\mathbf{6.65\times10^{-2}}$
60 units & 2 hidden layers	4.43×10^{-1}	4.84×10^{-1}	$\mathbf{6.99\times10^{-2}}$
80 units & 2 hidden layers	4.44×10^{-1}	2.53×10^{-1}	$\mathbf{6.17\times10^{-2}}$
20 units & 4 hidden layers	3.02×10^{-1}	5.11×10^{-1}	$\mathbf{6.43\times10^{-2}}$
40 units & 4 hidden layers	2.31×10^{-1}	3.14×10^{-1}	$\mathbf{5.02\times10^{-2}}$
60 units & 4 hidden layers	1.28×10^{-1}	1.73×10^{-1}	$\mathbf{2.06\times10^{-2}}$
80 units & 4 hidden layers	1.19×10^{-1}	9.86×10^{-2}	$\mathbf{2.10\times10^{-2}}$
20 units & 6 hidden layers	3.06×10^{-1}	4.09×10^{-1}	$\mathbf{3.04\times10^{-2}}$
40 units & 6 hidden layers	1.24×10^{-1}	2.57×10^{-1}	$\mathbf{1.88\times10^{-2}}$
60 units & 6 hidden layers	1.19×10^{-1}	1.01×10^{-1}	$\mathbf{1.45\times10^{-2}}$
80 units & 6 hidden layers	1.15×10^{-1}	9.34×10^{-2}	$\mathbf{1.50\times10^{-2}}$
20 units & 8 hidden layers	1.60×10^{-1}	1.76×10^{-1}	$\mathbf{2.95\times10^{-2}}$
40 units & 8 hidden layers	1.28×10^{-1}	9.89×10^{-2}	$\mathbf{2.12\times10^{-2}}$
60 units & 8 hidden layers	1.25×10^{-1}	9.34×10^{-2}	$\mathbf{2.96\times10^{-2}}$
80 units & 8 hidden layers	9.79×10^{-2}	7.54×10^{-2}	$\mathbf{1.72\times10^{-2}}$

To further quantify performance, Table 2 lists the relative error of the lid-driven cavity flow solution for different network sizes and for the three models. Each row in Table 2 corresponds to a specific network architecture. The architecture

is defined by the number of neurons per layer and the number of hidden layers. It can be seen that across all tested configurations, T-PINN-LBM consistently yields the lowest error, often by a wide margin. For smaller networks (e.g., 2 hidden layers), the performance of PINN-LBM may not surpass that of PINNs. This is likely due to the limited representational capacity and the added complexity of the LBM module, which cannot be effectively leveraged with such shallow architectures. However, as network size increases, PINN-LBM generally outperforms PINN, and T-PINN-LBM outperforms both in all cases, demonstrating strong performance and robustness in every scenario. Typically, T-PINN-LBM achieves an order-of-magnitude improvement in accuracy over PINN for comparable network sizes. These results underline that the proposed framework not only improves accuracy for a given model size, but also scales well with larger network architectures.

Figure 7, which presents the data from Table 2, demonstrates that the T-PINN-LBM model consistently outperforms both the PINN and PINN-LBM models across all evaluated network architectures. Typically, T-PINN-LBM achieves about an order of magnitude lower error than PINN for a given network size. Furthermore, figure 7 shows that PINN-LBM can be less effective for relatively small networks, and in some cases it even performs worse than PINN. This suggests that without proper initialization or sufficient capacity, the added complexity of LBM component might not pay off. However, with TRWI method, T-PINN-LBM consistently delivers strong accuracy and robustness in all cases, and the performance gap widens as the network complexity increases.

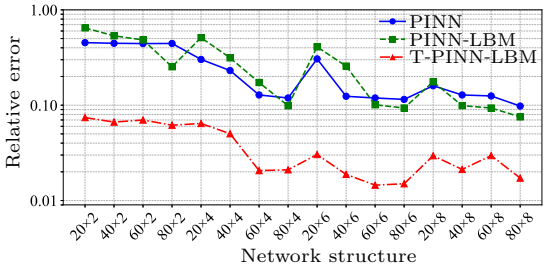


Fig. 7. Visualization of the relative error comparison for the lid-driven cavity across different network structures and models.

3.3. Validation of the stability of the T-PINN-LBM model for high Reynolds number problems

Previous literature indicates that the vanilla PINN framework struggles to address the benchmark problem for Reynolds numbers (Re) exceeding 100, with certain enhanced PINN variants maintaining accuracy and stability only for moderate Reynolds numbers (e.g., $Re < 500$).[28,29] This section evaluates the performance of the T-PINN-LBM model in simulating high Reynolds number cavity flows, comparing its efficacy against that of the vanilla PINN in these regimes.

Experiments are conducted for lid-driven cavity flow at increasing Reynolds numbers. The same neural network configuration as before is used, and the PINN and T-PINN-LBM models are trained using the Adam optimizer. For each case, the governing PDE residual and boundary conditions are enforced at 20000 random interior points and 2000 uniform boundary points. Following training, the relative $L2$ error of the solution was assessed on a refined 500×500 grid spanning the computational domain.

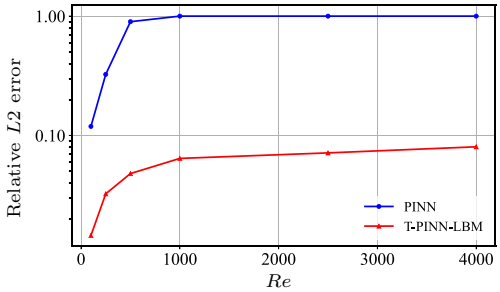


Fig. 8. Solving for prediction errors in the lid-driven cavity flow with different Re.

Figures 8–12 show the results for various Reynolds numbers. Figure 8 summarizes the prediction error of the vanilla PINNs and T-PINN-LBM models at different Re. Figures 9–12 show qualitative comparisons of the flow field predictions with the reference solution at different Reynolds numbers (500, 1000, 2500, and 4000). Each of Figs. 9–12 includes the reference solution and the predictions by PINN and T-PINN-LBM for the given Re, allowing visual assessment of accuracy and stability.

The results demonstrate a stark contrast in performance. For Reynolds numbers beyond a certain threshold ($Re = 250$), the vanilla PINN fails to obtain correct solutions. It either diverges or predicts unphysical flow patterns. In contrast, T-PINN-LBM is able to produce accurate solutions even for Re as high as 4000. This is a significant finding, as it indicates that the proposed PINN-LBM framework, augmented with TRWI, can solve much more challenging fluid dynamics problems than the vanilla PINN. The ability to handle higher Reynolds numbers means that T-PINN-LBM can tackle more complex and realistic engineering flow problems, which are often characterized by high inertia and nonlinearity. In Fig. 8, one can see that the error of the vanilla PINN increases dramatically or the solution fails altogether as Re increases above 250, whereas the error of T-PINN-LBM remains low and the method continues to converge for Re up to 4000. Figures 9–12 further illustrate that the predictions of the vanilla PINN at high Re are incorrect. In many cases, the model fails to capture even the basic flow structures. In contrast, the predictions of T-PINN-LBM closely match the reference solution, even at $Re = 4000$.

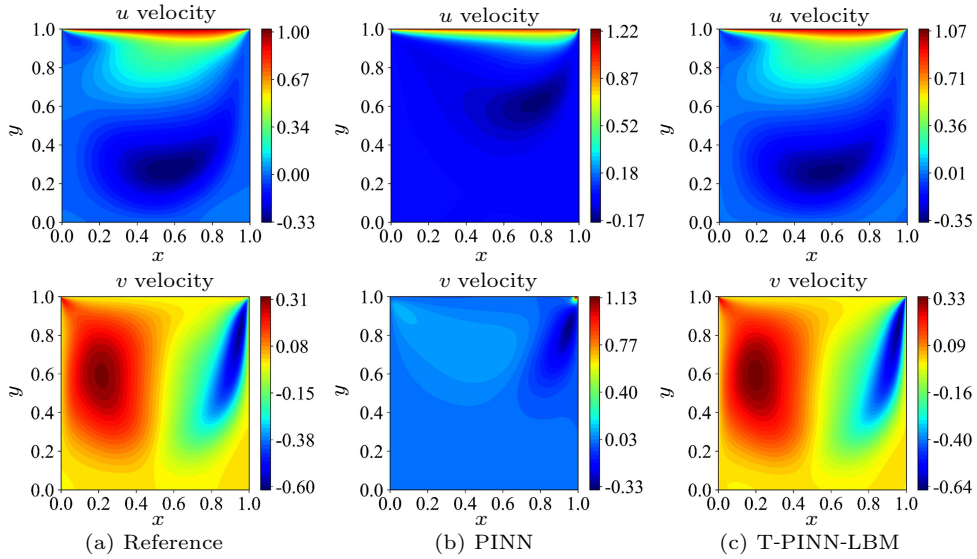


Fig. 9. At $Re = 500$: comparison of predicted values of (a) the reference solution, (b) PINN, and (c) T-PINN-LBM.

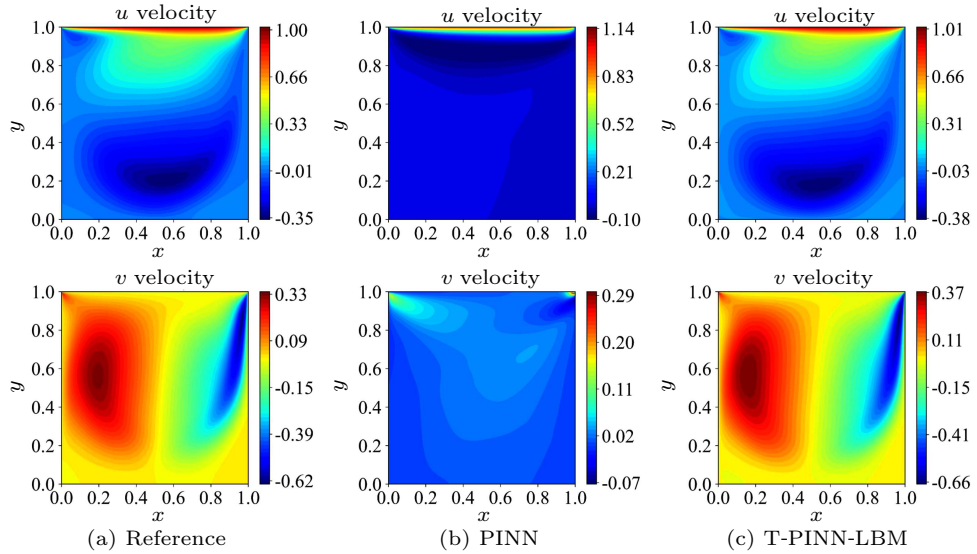


Fig. 10. At $Re = 1000$: comparison of predicted values of (a) the reference solution, (b) PINN, and (c) T-PINN-LBM.

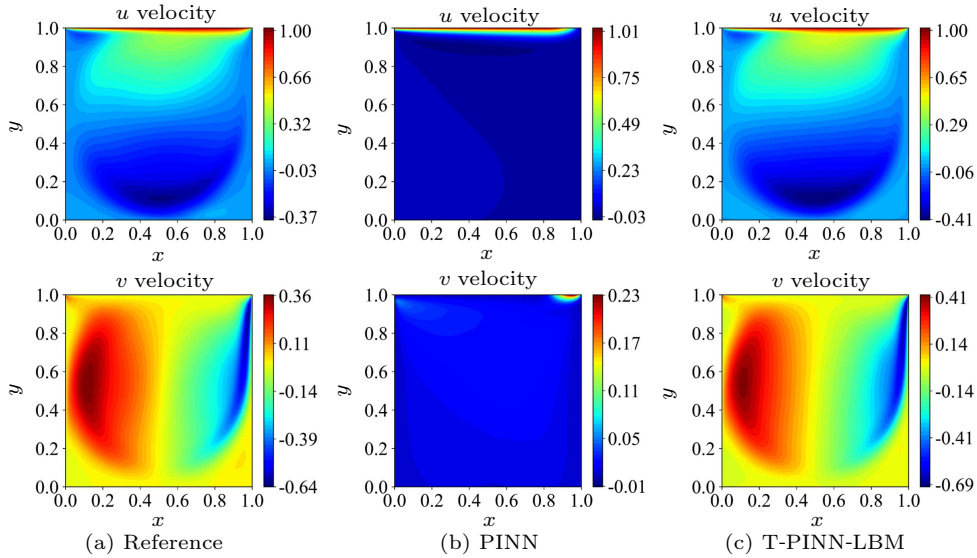


Fig. 11. At $Re = 2500$: comparison of predicted values of (a) the reference solution, (b) PINN, and (c) T-PINN-LBM.

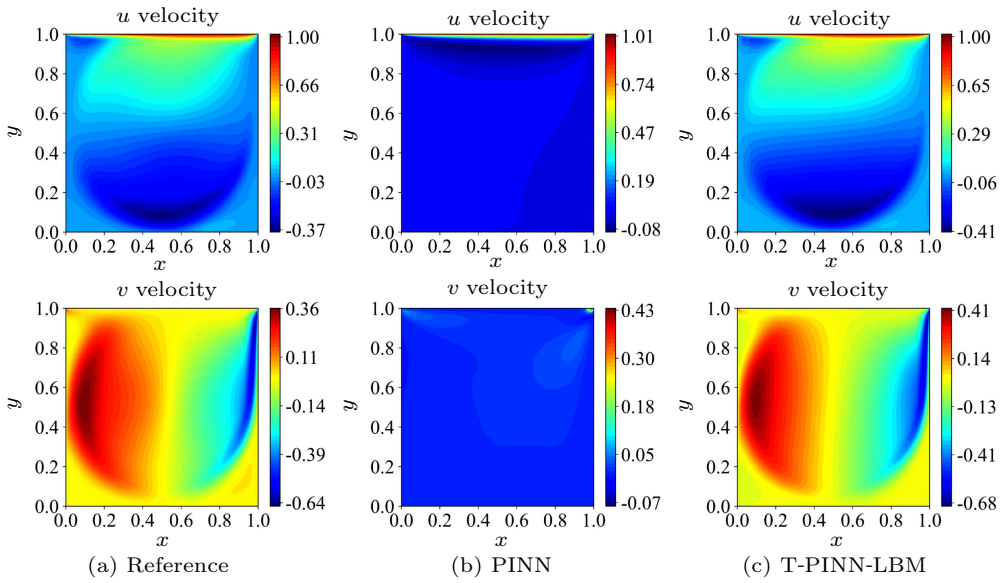


Fig. 12. At $Re = 4000$: comparison of predicted values of (a) the reference solution, (b) PINN, and (c) T-PINN-LBM.

These findings highlight the enhanced stability and accuracy of our method in regimes where the vanilla PINN struggles. The improvements can be attributed to the combination of LBM's strength in handling fluid advection and turbulence and the robust training provided by TRWI. This synergy enables the neural network-based solver to maintain stability and accuracy in solving PDEs at high Reynolds numbers, greatly extending the applicability of PINNs to more complex fluid dynamics problems.

4. Conclusions

In this paper, we presented the T-PINN-LBM model, which achieves superior accuracy and robustness compared to the vanilla PINN and PINN-LBM approaches for steady flow simulations without any data-driven supervision. By introducing TRWI method grounded in fixed-point analysis, we significantly improved gradient propagation and activation distribution in deep and wide neural networks, enabling stable convergence and low error across different network scales. Experiments on the lid-driven cavity flow demonstrated that, relative to Xavier initialization, T-PINN-LBM reduced the relative error by about 88% in this benchmark case. Furthermore, T-PINN-LBM was shown to successfully handle flows at Reynolds numbers up to 4000, a regime where the vanilla PINN failed to produce correct solutions.

The proposed model accurately captures complex flow structures without requiring any prior simulation or experimental data, thus providing a new viable path for data-free fluid simulation. We expect that this approach can be extended to more complex scenarios. To address unsteady flows, particularly at high Reynolds numbers where transient effects are significant, we plan to extend the T-PINN-LBM model in future work by incorporating time-dependent governing equations. This will enable the simulation of transient flows at high Reynolds numbers. Additionally, we aim to explore the application of the TRWI method to three-dimensional and multiphase flow problems, integrating adaptive activation functions and dynamic sampling strategies to further enhance the efficiency and accuracy of physics-informed fluid simulations in the absence of data.

Program availability

The source code is openly available in Science Data Bank at https://doi.org/10.57760/sciencedb.28294.

Acknowledgement

Project supported by the National Natural Science Foundation of China (Grant Nos. 12474453, 12174085, and 12404530).

References

[1] Zienkiewicz O C and Taylor R L 2013 *Finite Element Method* (Oxford: Amsterdam Butterworth-Heinemann Press) pp. 705–714
[2] Liu Y Q, Cheng R J and Ge H X 2013 *Chin. Phys. B* **22** 100204
[3] Cardiff P and Demirdzic I 2021 *Arch. Comput. Meth. Engin.* **28** 3721
[4] Raissi M, Perdikaris P and Karniadakis G E 2018 *J. Comput. Phys.* **378** 686
[5] Mao Z P, Jagtap A D and Karniadakis G E 2020 *Comput. Meth. Appl. Mech. Engin.* **360** 112789
[6] Shukla K, Leoni D and Karniadakis G E 2020 *J. Nondest. Eval.* **39** 61
[7] Lu L, Jin P Z, Pang G F, Zhang Z Q and Karniadakis G E 2021 *Nat. Mach. Intell.* **3** 218
[8] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S F and Yang L 2021 *Nat. Rev. Phys.* **3** 422
[9] Yin M L, Zheng X N, Humphrey J and Karniadakis G E 2021 *Comput. Meth. Appl. Mech. Engin.* **375** 113603
[10] Bararnia H and Esmaeilpour M 2022 *Inter. Com. Heat. Mass Tran.* **132** 105890
[11] Cuomo S, Di C, Vincenzo S, Giampaolo F, Rozza G, Raissi M and Piccialli F 2022 *J. Sci. Comput.* **92** 88
[12] Hanna J M, Aguado J, Askri R and Borzacchiello D 2022 *Comput. Meth. Appl. Mech. Engin.* **396** 115100

[13] Hosseini V R, Mehrizi A, Gungor A and Afrouzi H 2023 *Fuel* **332** 125908
[14] Zhu J A, Xue Y H and Liu Z S 2024 *Appl. Math. Mech.* **45** 1685
[15] Liu J N, Hou Q Z, Wei J G and Sun Z W 2023 *Chin. Phys. B* **32** 070702
[16] Zha W S, Chen D S, Li D L, Shen L H and Chen E Y 2025 *Chin. Phys. B* **34** 040701
[17] Chen S Y and Doolen G D 2003 *Annu. Rev. Flu. Mech.* **30** 329
[18] Lou Q, Meng X H and Karniadakis G E 2021 *J. Comput. Phys.* **447** 110676
[19] Liu Z X, Chen Y, Song G, Song W and Xu J 2023 *Mathematics* **11** 4147
[20] Fang D H, Tan J F and Xu J 2022 *Oce. Engin.* **263** 112360
[21] Qian Y H, Humieres D and Lallemand P 1992 *Europhys. Lett.* **17** 479
[22] He X Y and Luo L S 1997 *Phys. Rev. E* **55** R6333
[23] Luo L S and Girimaji S 2003 *Phys. Rev. E.* **67** 036302
[24] Boon J P 2003 *Euro. J. Mech. Fluids.* **22** 101
[25] Lee H, Choi H and Kim H 2025 arXiv:2410.02242 [cs.LG]
[26] Kingma D and Ba J 2014 arXiv:1412.6980v9 [cs.LG]
[27] Li Y F, Shi Q Z, Li Y, Song X J, Liu C C and Wang W Q 2021 *Chin. Phys. B* **30** 014302
[28] Wang S F, Teng Y J and Perdikaris P 2021 *SIAM J. Sci. Comput.* **43** A3055
[29] Chiu P, Wong J C, Ooi C, Dao M and Ong Y 2022 *Comput. Meth. Appl. Mech. Engin.* **395** 114909