

Original Paper

Physics-informed graph neural network for predicting fluid flow in porous media

Hai-Yang Chen^{a,b}, Liang Xue^{a,b,*}, Li Liu^c, Gao-Feng Zou^{b,c}, Jiang-Xia Han^b, Yu-Bin Dong^b, Meng-Ze Cong^b, Yue-Tian Liu^{a,b}, Seyed Mojtaba Hosseini-Nasab^{d,e}

^a State Key Laboratory of Petroleum Resources and Engineering, China University of Petroleum, Beijing, 102249, China

^b Department of Oil-Gas Field Development Engineering, College of Petroleum Engineering, China University of Petroleum, Beijing, 102249, China

^c Research Institute of Exploration and Development, Sinopec Jiangnan Oilfield Branch Company, Wuhan, 430223, Hubei, China

^d School of Chemical Engineering, Petroleum and Gas Engineering Iran University of Science and Technology, Tehran, 1684613114, Iran

^e Department of Geoscience & Engineering, Petroleum Engineering Group, Delft University of Technology, Netherlands

ARTICLE INFO

Article history:

Received 28 November 2024

Received in revised form

12 June 2025

Accepted 17 June 2025

Available online 21 June 2025

Edited by Yan-Hua Sun

Keywords:

Graph neural network (GNN)

Deep-learning

Physical-informed neural network (PINN)

Physics-informed graph neural network (PIGNN)

Flow in porous media

Perpendicular bisectonal grid (PEBI)

Unstructured mesh

ABSTRACT

With the rapid development of deep learning neural networks, new solutions have emerged for addressing fluid flow problems in porous media. Combining data-driven approaches with physical constraints has become a hot research direction, with physics-informed neural networks (PINNs) being the most popular hybrid model. PINNs have gained widespread attention in subsurface fluid flow simulations due to their low computational resource requirements, fast training speeds, strong generalization capabilities, and broad applicability. Despite success in homogeneous settings, standard PINNs face challenges in accurately calculating flux between irregular Eulerian cells with disparate properties and capturing global field influences on local cells. This limits their suitability for heterogeneous reservoirs and the irregular Eulerian grids frequently used in reservoir. To address these challenges, this study proposes a physics-informed graph neural network (PIGNN) model. The PIGNN model treats the entire field as a whole, integrating information from neighboring grids and physical laws into the solution for the target grid, thereby improving the accuracy of solving partial differential equations in heterogeneous and Eulerian irregular grids. The optimized model was applied to pressure field prediction in a spatially heterogeneous reservoir, achieving an average L_2 error and R^2 score of 6.710×10^{-4} and 0.998, respectively, which confirms the effectiveness of model. Compared to the conventional PINN model, the average L_2 error was reduced by 76.93%, the average R^2 score increased by 3.56%. Moreover, evaluating robustness, training the PIGNN model using only 54% and 76% of the original data yielded average relative L_2 error reductions of 58.63% and 56.22%, respectively, compared to the PINN model. These results confirm the superior performance of this approach compared to PINN.

© 2025 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In the process of oil and gas reservoir development, accurately calculating the fluid flow state within the pores is crucial for the effective development and management of underground oil and gas resources. Traditional numerical simulation methods often simulate the movement of fluids and pressure changes in

reservoirs by discretizing and solving differential equations. However, these traditional methods often face a range of challenges and limitations. Numerical simulation techniques, typically based on finite difference, finite element, or finite volume methods, often encounter restrictions when dealing with nonlinearity, heterogeneity, and multiscale problems. Numerical simulation methods generally require iterative approximation of the correct solution to the equations. Therefore, if the discrete matrix of the model is in a large-scale ill-conditioned state, the computational cost can be extremely high (Li et al., 2019). Additionally, when dealing with large-scale oil and gas reservoir simulations, as well as reservoir parameter inversion and optimization, the

* Corresponding author.

E-mail address: xueliang@cup.edu.cn (L. Xue).

Peer review under the responsibility of China University of Petroleum (Beijing).

computational cost and speed can become unacceptable. These drawbacks severely limit the real-time capability of simulations and the efficiency of inversion and optimization calculations. Given the limitations of traditional numerical simulation models, several methods have emerged in recent years to improve computational efficiency, such as upscaling methods (Li et al., 2021; Syed et al., 2020), mixed finite volume method (FVM)/finite element method (FEM) (Ghoreishian Amiri et al., 2017; Shao et al., 2021), polynomial chaos expansion (PCE) (Jain et al., 2017; Tarakanov and Elsheikh, 2019), Gaussian process (Conrad et al., 2016; Hamdi et al., 2017), deep-learning-based methods (Atadeger et al., 2022; Kingma and Dhariwal, 2018), etc.

The rapid development of artificial intelligence and the successful application of neural networks in high-dimensional data regression in computer vision and natural language processing have stimulated research into building surrogate models for high-dimensional nonlinear systems based on deep learning methods (Lazzara et al., 2023; Zhao et al., 2023). Although neural network approaches have achieved widespread application and great success in engineering, most models rely on data-driven methods to extract target features (Han and Xue, 2023; Zhu et al., 2020). This approach, however, presents some unresolved issues (Oishi and Yagawa, 2017; Yuan et al., 2022). First, data-driven methods require a large amount of high-quality data, and errors in the data can significantly affect the model (Xue et al., 2023). Second, during training, data-driven methods are prone to overfitting, leading to weak generalization ability. Additionally, due to the lack of guidance from physical theories, the model's predictions may sometimes violate scientific principles. To address these issues, a model that integrates data-driven methods with physical knowledge is needed to enhance its consistency with predictions from traditional physics-driven modeling. Researchers have explored various strategies to embed physics-based priors into machine learning, influencing aspects like network structure, loss functions, and optimization algorithms. Kang et al. (2025), for instance, introduced a methodology employing strong physical constraints by integrating mechanistic insights with data-driven models; leveraging a physics-informed autoencoder, they incorporated prior knowledge from well log interpretation into their network model. In related work, Qu et al. (2023) proposed a deep neural network for forecasting fracturing parameters. Addressing the challenge of limited data availability, their model incorporated physical constraints based on field expertise and the principle of fracture volume conservation to enhance predictive accuracy. To incorporate physical partial differential equation (PDE) constraints into neural network models, Raissi et al. (2019) proposed physics-informed neural networks (PINNs), which incorporate physical knowledge into network training. Using the residuals of nonlinear partial differential equations as constraints, the network is trained with automatic differentiation on grid data. With this method, the network can learn any underlying physical equations. Based on the PINN model framework, researchers have proposed numerous network models with various strengths to address different issues (Chiu et al., 2022; Hu et al., 2024; Meng et al., 2023). To address the challenge of uncertainty propagation in high-dimensional elliptic stochastic partial differential equations (SPDEs), Karumuri et al. (2020) developed a new method. By introducing a physics-informed loss function derived from the variational principle, they trained a deep residual neural network, effectively addressing the challenges of high-dimensional uncertainty propagation and its inverse problems. To solve fluid flow problems in geological structures, many PINN-based models have been proposed (Han et al., 2023). Shan et al. (2023) introduced a PINN incorporating long short-term memory (LSTM) and attention mechanisms to solve the Buckley–Leverett partial differential equation governing

two-phase flow in porous media. Wang et al. (2020) developed the theory-guided neural network (TgNN). This model is trained based on observed or simulated data, with guidance from physical laws, engineering controls, and expert knowledge, achieving higher prediction accuracy compared to traditional artificial neural networks. To address the challenge of limited access to labeled data, Zhu et al. (2019) developed a model that incorporates the governing equations of the physical model into the loss function. This model can be trained without any labeled data and generates prediction results comparable to those of data-driven supervised learning models. To tackle the challenges posed by anisotropy and source/sink terms when simulating flow on a Eulerian grid. A deep learning framework named the theory-guided convolutional neural network (TgCNN) was developed by Wang et al. (2021) for the purpose of effective uncertainty quantification and data assimilation in reservoir flow scenarios involving uncertain model parameters. Han et al. (2024) proposed criss-cross physics-informed convolutional neural networks (CC-PINN), which use predefined two-dimensional convolutional layers to represent the spatial correlation between neighboring locations. This approach learns the solutions to partial differential equations with spatial heterogeneity parameters that possess physical properties. By improving the PINN model, these researchers have cleverly integrated physical constraints into neural networks to solve partial differential equations in physical fields. However, these advancements are still insufficient for accurately solving the fluid flow states in Eulerian unstructured grids with spatial heterogeneity.

In the process of oil and gas reservoir development, grid refinement is necessary to improve the accuracy of simulations. However, this leads to a decrease in computational efficiency for reservoir numerical simulations. On the other hand, increasing grid size to improve computational efficiency results in the loss of some geological information, thereby reducing the accuracy of the simulations. Since Eulerian irregular grids can adaptively adjust their sizes based on specific needs—dividing into smaller grids in areas with large pressure drops and larger grids in areas with smaller pressure drops—they can achieve a balance between high accuracy and computational efficiency. As a result, they are widely used in the simulation of fluid flow in porous media (Mlacnik et al., 2006). Previous deep neural network models were limited to convolutional architectures on regular grids and could not be extended to unstructured grid reservoir data. The PINN model, which relies on automatic differentiation, exhibits good adaptability in handling isotropic problems. However, due to the inability of automatic differentiation to enforce flux continuity across heterogeneous grids, the PINN framework faces challenges in accurately capturing the interconnectivity of heterogeneous, irregular grids with differing physical properties in anisotropic scenarios (Zhang et al., 2023). To address these issues, leveraging the inherent advantages of graph structures in representing unstructured data, and building on previous research (Pfaff et al., 2021; Shao et al., 2023), we propose PIGNN, a novel physics-informed graph neural network. By incorporating graph convolutional layers with customized adjacency-based convolution kernels, the model strictly enforces flux continuity between grid cells and accurately represents the discretized finite volume method (FVM) control equations, harmonic averaging of permeability, and the upwind-weighted differencing scheme. Building on these design principles, the model not only approximates the training data generated by numerical simulations but also employs the finite volume method to approximate the residuals of the governing partial differential equations. It implicitly handles the reservoir grid pressure and conductivity across cells, enabling the strict enforcement of flux continuity between adjacent elements. In addition, PIGNN leverages local message passing between

adjacent nodes and the underlying data structure for training and inference, which helps alleviate the challenges associated with training deep neural networks. This model enables fast and accurate prediction of fluid flow patterns in reservoirs and achieves higher computational precision compared to conventional PINN models.

The structure of this paper is as follows: Section 2 introduces the governing equations for single-phase Darcy flow in reservoirs and their discretized form based on the finite volume method. Next, the network architecture of the PIGNN model and the physics-informed training scheme are presented. Section 3 optimizes the network's hyperparameters using single-phase reservoir simulation data and verifies the superiority of the optimized model in heterogeneous single-phase reservoirs. Additionally, the PIGNN and PINN models are trained using training sets of varying sizes to test the predictive performance of the PIGNN model under different amounts of training data. Finally, Section 4 provides a summary and discussion of the findings.

2. Methodology

2.1. Single-phase flow in subsurface

The fluid flow in reservoir formations is a typical problem of porous media flow. In this study, we consider the general single-phase Darcy governing equation for porous media in reservoir formations. Additionally, the fluid in the reservoir is considered slightly compressible in this study. The macroscopic behavior of single-phase fluid must satisfy the continuity equation (Eq. (1)) and Darcy's law (Eq. (2)).

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho \vec{v}) = \rho q, \quad (1)$$

$$\vec{v} = -\frac{K}{\mu}(\nabla p - g\rho\nabla z), \quad (2)$$

where ϕ represents the porosity of the reservoir; ρ is the density of the reservoir fluid; t denotes time; $\vec{v}$ represents the Darcy velocity of the fluid; q is the source or sink term in the reservoir; K denotes the absolute permeability of the reservoir; μ is the viscosity of the reservoir fluid; p represents the reservoir formation pressure; g is the gravitational acceleration; and z represents the vertical coordinate.

In this work, the compressibility of the slightly compressible fluid and the rock can be expressed as follows:

$$c_f = \frac{1}{\rho} \frac{d\rho}{dp}, \quad (3)$$

$$c_r = \frac{1}{\phi} \frac{d\phi}{dp}, \quad (4)$$

where c_f represents the fluid compressibility coefficient; and c_r represents the rock compressibility coefficient.

By substituting Darcy's law (Eq. (2)), the fluid compressibility equation (Eq. (3)), and the rock compressibility equation (Eq. (4)) into the continuity equation (Eq. (1)), and neglecting vertical effects to consider only horizontal flow, we obtain the parabolic equation for fluid pressure.

$$c_t \phi \rho \frac{\partial p}{\partial t} - \nabla \cdot \left[\frac{\rho K}{\mu} \nabla p \right] = \rho q, \quad (5)$$

where c_t represents the sum of the compressibility coefficients of the rock and fluid, i.e., $c_t = c_f + c_r$.

The governing equation can be rewritten using the volume factor and compressibility coefficient as

$$\frac{c_t \phi}{B_o} \frac{\partial p}{\partial t} - \nabla \cdot \left[\frac{\rho K}{B_o \mu} \nabla p \right] = q, \quad (6)$$

where B_o represents the volume factor of oil in the reservoir.

2.2. Physics-informed graph neural network

Traditional neural network models, such as convolutional neural networks (CNNs), deep neural networks (DNNs), and recurrent neural networks (RNNs), have achieved significant success in processing Euclidean structured data, such as images, text, and audio. However, in many scientific fields, numerous real-world research subjects and problems require complex non-Euclidean graph representations, such as protein molecular structures (Réau et al., 2023), social relationships (Hawthorne et al., 2023), and transportation networks (Sant'Ana da Silva et al., 2023). A graph is a special type of data structure used to describe the attributes of natural entities and the complex relationships between them. It is typically represented as $G = (V, E)$, where V represents the nodes of the network, and E represents the edges connecting the nodes. In this study, V represents the center of each reservoir grid, and E represents the contact surface between grids.

Graph neural networks (GNNs), with their powerful ability to process structured data and extract high-order information, have become an emerging technology in many recommendation problems. In this study, we designed a neural network model with an encode-process-decode structure based on MeshGraphNets (Shao et al., 2023), aimed at learning the numerical simulation patterns of Eulerian irregular grid structures. This model serves as an efficient and accurate alternative to traditional numerical simulation models. The model predicts the graph data G^{t+1} for the next time step based on the graph data G^t at the current time step.

In the encoder part, the Eulerian irregular data of the reservoir is encoded into graph data $G = (V, E)$. In the graph, the nodes V represent the centers of the grids, with the grid pressure used as the feature of each node. Since fluid flows bidirectionally across the contact surfaces between grids in the reservoir, the edges in the graph are encoded as bidirectional. The relative displacement of the position coordinates on both sides of the contact surface, the norm of the relative displacement, and the area of the contact surface between the grids are used as edge features in the graph. The features of G^{t+1} are encoded into multi-dimensional vectors using a multilayer perceptron (MLP) network. The dimensionality D of the target encoded vectors was treated as a hyperparameter and optimized in subsequent research.

The processor part consists of multiple node and edge information exchange blocks, where information is continuously passed between nodes and edges within the graph neural network. Each information exchange block has independent network parameters, and these blocks are connected in series, taking the output of the previous block as the input for the next block, thereby updating the information in the graph. The information update process for nodes and edges in each exchange block can be represented by the following equations:

$$e'_{mn} \leftarrow f^E(e_{mn}, v_m, v_n), v'_m \leftarrow f^V\left(v_m, \sum_n e'_{mn}\right), \quad (7)$$

where e_{mn} represents the feature of the edge between node m and node n ; e'_{mn} represents the updated feature of the edge between node m and node n ; v_m represents the feature of node m ; v_n

represents the feature of node n ; v'_m represents the updated feature of node m ; f^E and f^V represent MLP networks with residual connections.

In the decoder part, the decoder uses a MLP network to convert the multi-dimensional node features v'_m from the processor into output features p_m^{t+1} , such as pressure and other quantities. The set of output features for all nodes P^{t+1} is then used to update the network nodes V , generating G^{t+1} , thereby achieving the prediction of the graph state at time step $t + 1$ based on previous time step t .

2.3. Loss function

Although traditional data-driven methods have developed rapidly, purely data-driven approaches are often considered black-box systems with no physical interpretability. They fail to fully leverage the implicit physical information contained in the collected data, making the predicted outputs physically inconsistent. In this study, we train the proposed PIGNN model using a reservoir numerical simulation dataset based on the governing equations of fluid flow. The model not only needs to approximate the labeled data but also must satisfy the fluid flow governing equations between adjacent time steps. Therefore, the model's loss function must include both a data-driven loss and a physics-driven loss. We define the loss function as

$$\text{Loss} = \text{Loss}_{\text{data}} + \text{Loss}_{\text{physics}}, \quad (8)$$

where $\text{Loss}_{\text{data}}$ represents the data-driven loss; and $\text{Loss}_{\text{physics}}$ represents the physics-driven loss.

The data-driven loss function is defined as

$$\text{Loss}_{\text{data}} = \sum_t \sum_m (q_m^t - \hat{q}_m^t), \quad (9)$$

where t represents the time step; m represents the grid node; p_m^t denotes the pressure value at node m at time step t ; $\hat{p}_m^t$ represents the model-predicted pressure value at node m at time step t .

The flow governing equation (Eq. (6)) can be discretized using the finite volume method. The discretized governing equation is

$$\sum_j T_{ij}^{n+1} (p_j - p_i)^{n+1} - \frac{V_i}{\Delta t} \left(\frac{\phi C_r}{B_o} - \frac{c_o}{B_o} \right) (p_i^{n+1} - p_i^n) = q, \quad (10)$$

where i represents any grid block generated after grid discretization; j represents the adjacent grids around grid i , as shown in Fig. 1; V_i denotes the volume of grid; Δt is the time step size; superscripts n and $n + 1$ represent the previous and next time steps, respectively; the volume factor B_o is taken as the value of the grid with higher pressure, i.e., using the upstream weighting value (Eq. (11)); T_{ij}^{n+1} represents the transmissibility between the two adjacent grids i and j .

The irregular grid used in this study is the perpendicular bisectional grid (PEBI), and the transmissibility between grids can be expressed as Eq. (12).

$$B_o = \begin{cases} B_{o,i} & \text{if } p_i \geq p_j \\ B_{o,j} & \text{if } p_i \leq p_j \end{cases}, \quad (11)$$

$$T_{i,j} = \frac{K_{ij} \omega_{ij}}{\mu_o d_{ij}}, \quad (12)$$

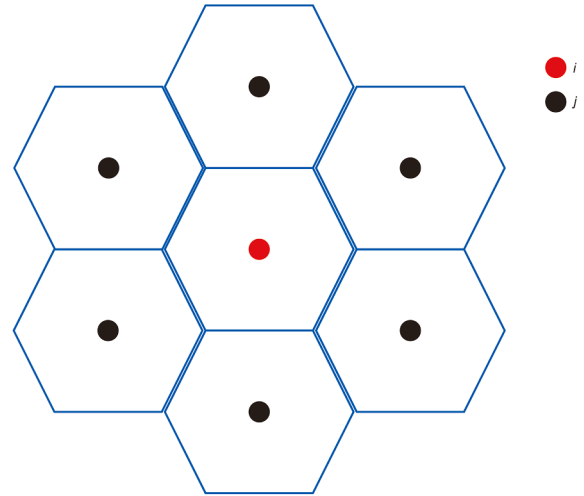


Fig. 1. Reservoir grid element structure.

where K_{ij} represents the harmonic mean of the permeability between grid i and grid j ; the fluid viscosity μ_o is taken as the viscosity at the grid with the higher pressure, i.e., using the upstream weighting (Eq. (13)); ω_{ij} represents the cross-sectional area between grid i and grid j ; and d_{ij} represents the distance between the center of grid i and the center of grid j .

$$\mu_o = \begin{cases} \mu_{o,i} & \text{if } p_i \geq p_j \\ \mu_{o,j} & \text{if } p_i \leq p_j \end{cases}, \quad (13)$$

The well model can be defined by the Peacemen equation (Eq. (14)).

$$q = \frac{2\pi kh}{\mu_o (\ln(r_e/r_w) + S)} (p_w - p_i), \quad (14)$$

where k represents the permeability of the grid at the well point; h denotes the thickness of the grid at the well point; p_i is the pressure in the grid at the well point; p_w represents the bottomhole pressure; r_w is the wellbore radius; and S denotes the skin factor of well.

To ensure the continuity of flux between heterogeneous grids and incorporate physical constraints into the graph neural network, we propose a predefined physical graph neural network convolution kernel, as illustrated in Fig. 2.

Thus, the residual of the discrete governing equation (Eq. (10)) can be represented using the physical graph neural network convolution kernel as follows:

$$\text{MSE}_{\text{physics}} = \frac{1}{N} \sum_{i=1}^N \left[A - \frac{V_i}{\Delta t} \left(\frac{\phi C_r}{B_o} - \frac{c_o}{B_o} \right) (p_i^{n+1} - p_i^n) - q_i \right]^2, \quad (15)$$

where $\text{MSE}_{\text{physics}}$ represents the residual loss of the physical governing equation; $A = \text{physics} - \text{gnn} - \text{kernel} \cdot (p_{\text{surround}}^{n+1} - p_i^{n+1})$; physics - gnn - kernel denotes the physical graph convolution kernel; N is the total number of grids; p_i^{n+1} represents a vector of length equal to the number of grids surrounding grid i , containing the pressures p_i^{n+1} at time $n + 1$; $p_{\text{surround}}^{n+1}$ is a vector consisting of the pressures of the grids surrounding grid i ; and q_i denotes the source or sink term at grid i .

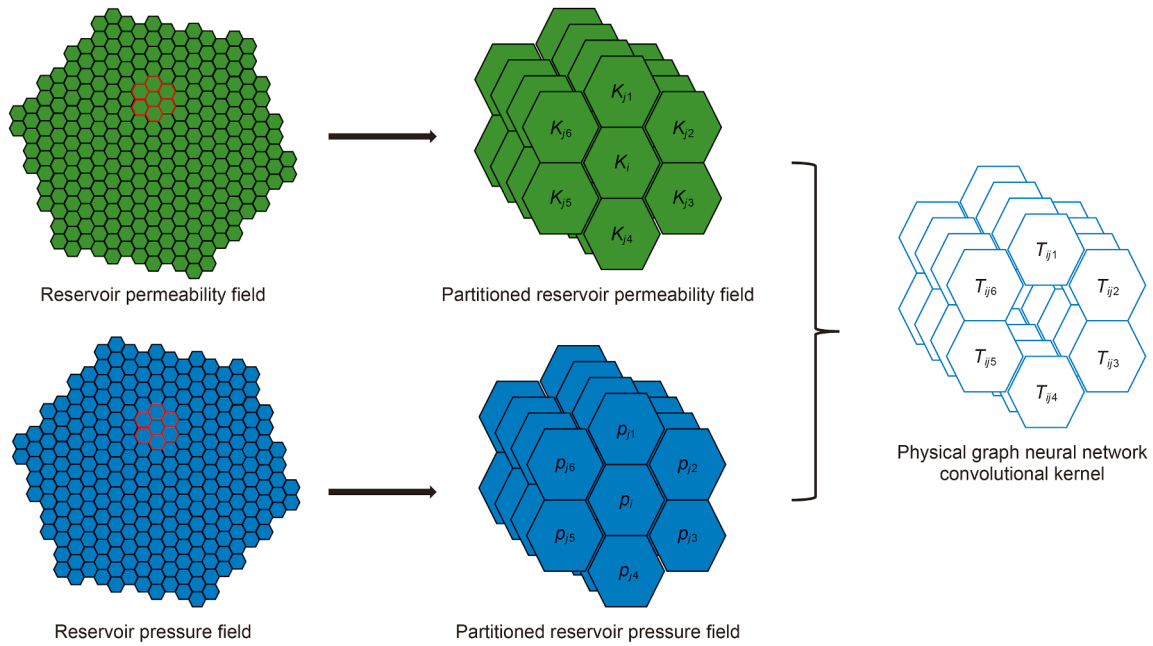


Fig. 2. Physical graph convolution kernel.

For boundary condition constraints, this can be implemented by simulating grid padding. If a closed boundary condition is used, a grid with the same pressure as the boundary grid is added outside the boundary to simulate a no-fluid-flow state. If a constant pressure boundary is applied, a grid with a fixed pressure is added outside the boundary to simulate fluid flow.

How to set the weight coefficients for the data matching loss term and the physics constraint loss term in the loss function is a problem without a definitive answer. Previous researchers typically set the weight coefficients manually based on experience or optimized them through trial and error (Xu et al., 2021). In this work, we adopt the training technique proposed by Han et al. (2024). Since the physics constraint loss is built on top of the data-driven loss, the reduction of the physics loss requires the reduction of the data matching loss. Therefore, we first train the neural network using only the data-driven approach. Once the data-driven loss has been reduced to a certain level, we then add the physics constraint loss to the loss function and continue training the network weights using both data-driven and physics-driven approaches to minimize the overall loss. This method not only partially addresses the issue of setting the weight coefficients for the loss terms, but also avoids the problem of the network's output exceeding the domain of the physics constraint equations due to random initialization. Additionally, it accelerates the network's training process.

The model architecture and network hyperparameters of the physics-constrained graph neural network are shown in Fig. 3 and Table 1.

The input to the PIGNN network consists of node data and edge data. The node data is the pressure data at the center of the irregular grid from the previous time step, represented as a one-dimensional vector. The edge data includes spatial information between adjacent grids from the previous time step, such as grid contact area, distance between grid centers, etc., represented as a multi-dimensional vector. The output is the pressure data at the grid centers for the next time step. The network architecture is shown in Fig. 3, and the network uses the ReLU activation function. The Adam optimizer is used to optimize network parameters, with

a learning rate decay strategy to accelerate training. The hyperparameters such as learning rate, learning rate decay rate, and number of training iterations are detailed in Table 1.

In this study, the coefficient of determination and the relative L_2 error are used as two metrics to evaluate the predictive performance of the PIGNN model, as shown in Eqs. (16) and (17). Additionally, we used root mean squared error (RMSE) and mean absolute percentage error (MAPE) to calculate the error between the model-predicted bottomhole pressure and the reference bottomhole pressure, thereby assessing the prediction accuracy of the model. RMSE and MAPE can be calculated using Eqs. (18) and (19)

$$R^2 = 1 - \frac{\sum_{i=1}^N (\hat{u}_i - u_i)^2}{\sum_{i=1}^N (u_i - \bar{u})^2}, \quad (16)$$

$$L_2 = \frac{\|\hat{u} - u\|_2}{\|u\|_2}, \quad (17)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (u_i - \hat{u}_i)^2}, \quad (18)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{u_i - \hat{u}_i}{u_i} \right|, \quad (19)$$

where $\hat{u}_i$ represents the true value; u_i represents the predicted value; $\bar{u}_i$ represents the average of the true values; N represents the number of evaluated values; and $\|\cdot\|_2$ denotes the standard Euclidean norm; $|\cdot|$ represents the absolute value.

3. Result and discussion

In this section, we optimized some of the model's hyperparameters and evaluated the performance of the optimized model in heterogeneous reservoir problems. Additionally, we

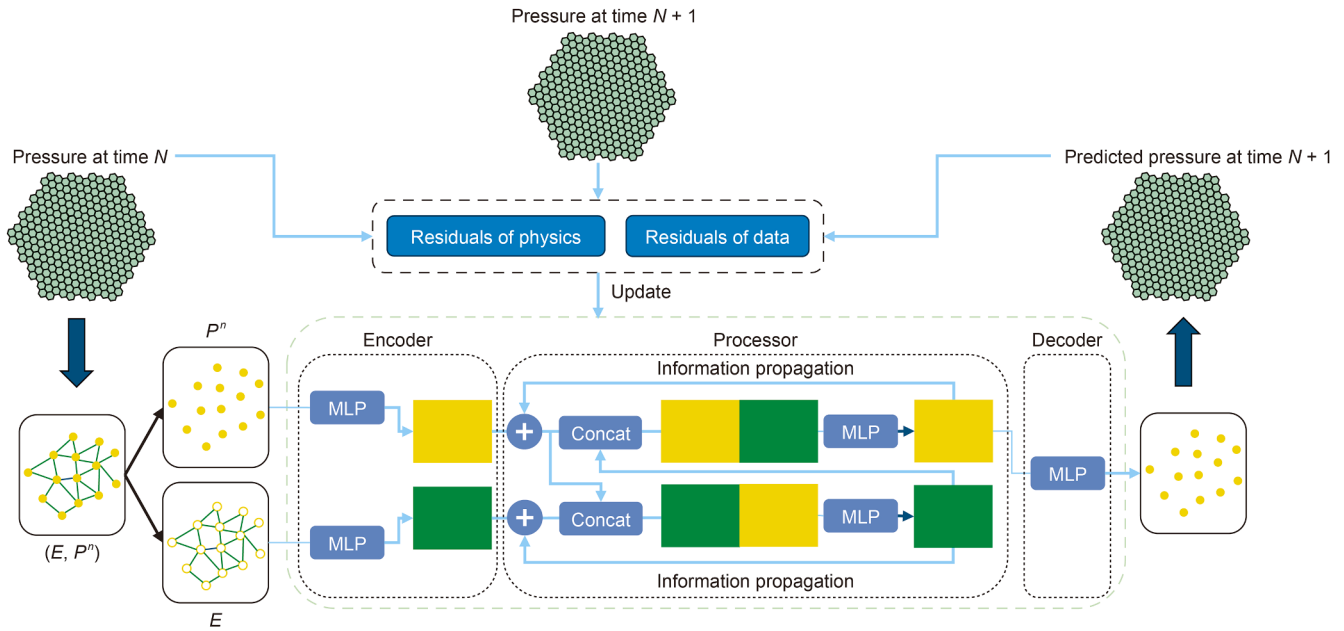


Fig. 3. Model structure of the physics-informed graph neural network.

Table 1
The hyperparameters of the PIGNN.

Hyperparameter	Range of value
Learning rate	0.001
Learning rate decay rate	0.1
Batch size	1
Input node dimension	1
Input edge dimension	4
Output dimension	1
Hidden layer dimension	32, 64, 128, 256
Number of layers	1, 2, 4, 8
Information propagation steps	4, 8, 16, 32

assessed the accuracy and robustness of the PIGNN model compared to the PINN model.

First, we validate the model's performance using a two-dimensional heterogeneous reservoir model. This model is derived from an actual geological reservoir model with four closed boundaries, and the label data was generated using commercial numerical simulation software. The initial fluid pressure in the reservoir is 300 bar. The reservoir model has dimensions of $1020 \text{ m} \times 1020 \text{ m} \times 10 \text{ m}$, and the PEBI (perpendicular bisection) grid method is used to divide the reservoir into grid blocks of varying shapes and sizes. Finer grids are applied in areas with complex near-wellbore flow to improve calculation accuracy, while coarser grids are used in regions with simpler far-wellbore flow to enhance computational efficiency. The permeability of the reservoir is heterogeneous across the plane, as shown in Fig. 4, while the porosity of the reservoir is uniform at 0.3. The reservoir rock and formation water are incompressible, and the volume factor of the reservoir oil at reference pressure is 1.12, with a compressibility factor of 0.0045 bar^{-1} . The viscosities of the reservoir water and oil are 0.3 and 3 cP, respectively, and remain constant regardless of pressure changes. The reservoir model contains three oil production wells, each producing at a fixed oil rate of 70, 90 and $50 \text{ m}^3/\text{d}$, respectively. The well coordinates are (280, 280), (510, 510), and (780, 780). The reservoir simulation runs for 100 time steps, with each time step representing a duration of one month.

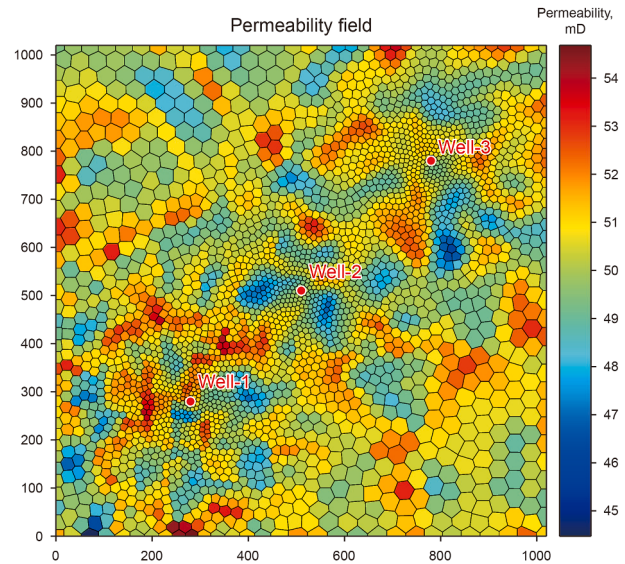


Fig. 4. Permeability field.

3.1. Hyperparameter optimization of the model

In this study, the data from the first 65 time steps were used as the training set, and the data from the remaining 35 time steps were used as the test set to evaluate the model's performance. This section investigated three parameters of the physics-constrained graph neural network model that required sensitivity analysis: the number of neurons in the MLP's hidden layers, the number of hidden layers of the MLP, and the number of message-passing steps in the neural network. By comparing the model's prediction performance on the test set, the optimal values for these parameters were determined, with evaluation metrics including the R^2 score and the relative L_2 error.

Empirically, the number of hidden layers in the model was pre-set to 2, and the number of message-passing steps was set to 16.

The impact of different numbers of neurons in the MLP's hidden layers on the prediction results is shown in Fig. 5(a). With the number of neurons in the MLP's hidden layers set to 256 and the number of message-passing steps set to 16, sensitivity analysis was conducted on different numbers of hidden layers of the MLP. The impact of varying the number of hidden layers on the prediction results is shown in Fig. 5(b). When the number of neurons in the hidden layers of the MLP was 256 and the number of hidden layers was 2, the model achieved the highest R^2 and the lowest relative L_2 error on the test set, indicating optimal prediction performance. As the number of neurons in the hidden layers and the number of hidden layers increased, the model's prediction accuracy first improved and then decreased. Initially, increasing the number of neurons or hidden layers effectively enhanced the model's ability to extract latent features from the data. However, as the number of temporal-spatial convolution blocks increased, the number of parameter matrices in the network grew larger. Despite the physical information constraints, the model inevitably experienced overfitting, leading to reduced prediction accuracy.

Through these experiments, the optimal values for the number of neurons in the MLP's hidden layers and the number of MLP's hidden layers were determined. The number of message-passing steps in the model, being a major factor affecting the capability and efficiency of node and edge information exchange, often had a significant impact on the prediction results. For the number of message-passing steps, the model was trained with settings of 4, 8, 16, and 32 after optimizing other hyperparameters, and the prediction performance was analyzed. The impact of different numbers of message-passing steps on the prediction results is shown in Fig. 5(c). When the number of message-passing steps was set to 4 or 8, insufficient transmission of node and edge information resulted in poorer prediction performance, with a lower coefficient of determination and a higher relative L_2 error. When the number of message-passing steps was set to 32, excessive transmission of node and edge information led to the amplification of irrelevant information, resulting in decreased prediction accuracy and poorer performance. When the number of message-passing steps was 16, the model achieved a balance between effective feature extraction, elimination of irrelevant information, and computational efficiency, resulting in the highest coefficient of determination and the lowest relative L_2 error, indicating the best prediction accuracy.

3.2. Model comparison

In this section, we continue using the numerical simulation data from the previous section as the dataset. The data from the first 65 time steps are extracted as the training dataset for the neural network model, while the data from the last 35 time steps serve as the test set. We establish the optimized PIGNN and PINN models, iteratively adjusting the network parameters using the

training data, and predicting the future pressure field in the test set to compare the predictive accuracy of the PIGNN and PINN models. Additionally, to facilitate a multi-angle visual comparison of the model differences, we compute the fluid flow behavior within the reservoir using Eq. (2) and the predict pressure fields. At time step 75, the predicted pressures and velocity from the PIGNN and PINN models, compared to the reference pressure, are shown in Fig. 6. Similarly, at time step 100, the predicted pressures and velocity are shown in Fig. 7. The experimental results indicate that, compared to the PINN model, the PIGNN model produces pressure and velocity fields more closely aligned with the reference pressure and velocity field, making it better suited for handling fluid flow and pressure distribution in heterogeneous reservoir models.

As shown in Fig. 8, we present a quantitative comparison between the PIGNN and PINN models. The figure illustrates the correlation between the predicted values and the reference values at time steps 75 and 100 for both the PIGNN and PINN models. From the figure, it is evident that the PIGNN model exhibits a better fit between the predicted values and the reference values, resulting in higher prediction accuracy. Additionally, as the time steps increase, the correlation between the predicted values and reference values for the PINN model weakens rapidly, leading to a sharp decline in accuracy. In contrast, the PIGNN model shows a slower decline in prediction accuracy over time.

Fig. 9 compares the PIGNN and PINN models by illustrating their relative L_2 errors and R^2 scores of pressures at each time step on the test set. Averaged over the time steps, the PINN model shows a relative L_2 error of 2.908×10^{-3} and an R^2 score of 0.964. The PIGNN model demonstrates superior performance, with its average relative L_2 error reduced to 6.710×10^{-4} (a 76.93% decrease from PINN) and its average R^2 score increased to 0.998 (a 3.56% improvement over PINN). When the time steps are small, the prediction errors of the PIGNN and PINN models are quite similar. However, as the prediction time steps increase, the error in the PINN model grows exponentially, while the PIGNN model shows a slower increase in prediction error, demonstrating better stability. This is because the graph convolution kernel effectively captures the planar heterogeneity of permeability, resulting in slower accumulation of prediction errors. In contrast, the PINN model struggles to learn the heterogeneous characteristics of the reservoir, leading to an exponential increase in prediction error in the later time steps of the test set.

To further compare the performance of the PIGNN and PINN models, the prediction results of bottomhole pressure in the test set for both models are plotted Fig. 10 and compared with the reference values obtained from numerical simulation. From the figure, it is clear that the PINN model has larger prediction errors, and these errors increase over time. In contrast, the values predicted by the PIGNN model almost overlap with the reference pressure, showing much higher accuracy than the PINN model. This may be due to the difficulty of the PINN model in learning the

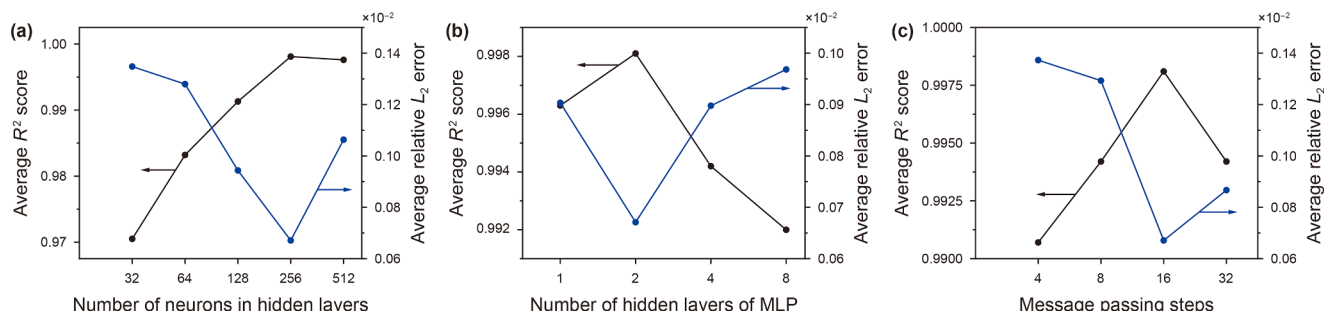


Fig. 5. Results for different hyperparameters.

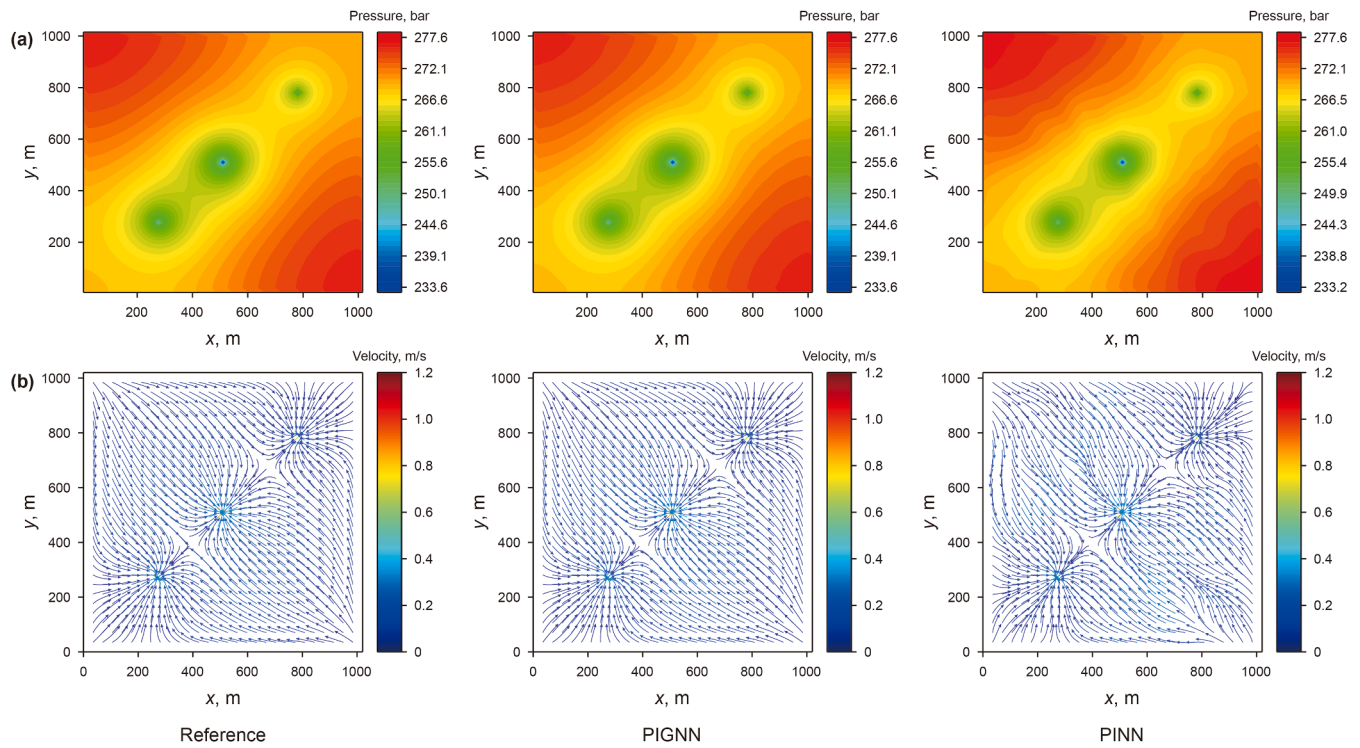


Fig. 6. Pressure (a) and velocity (b) fields obtained by numerical simulation (left), PIGNN model (middle), and PINN model (right) at time step 75.

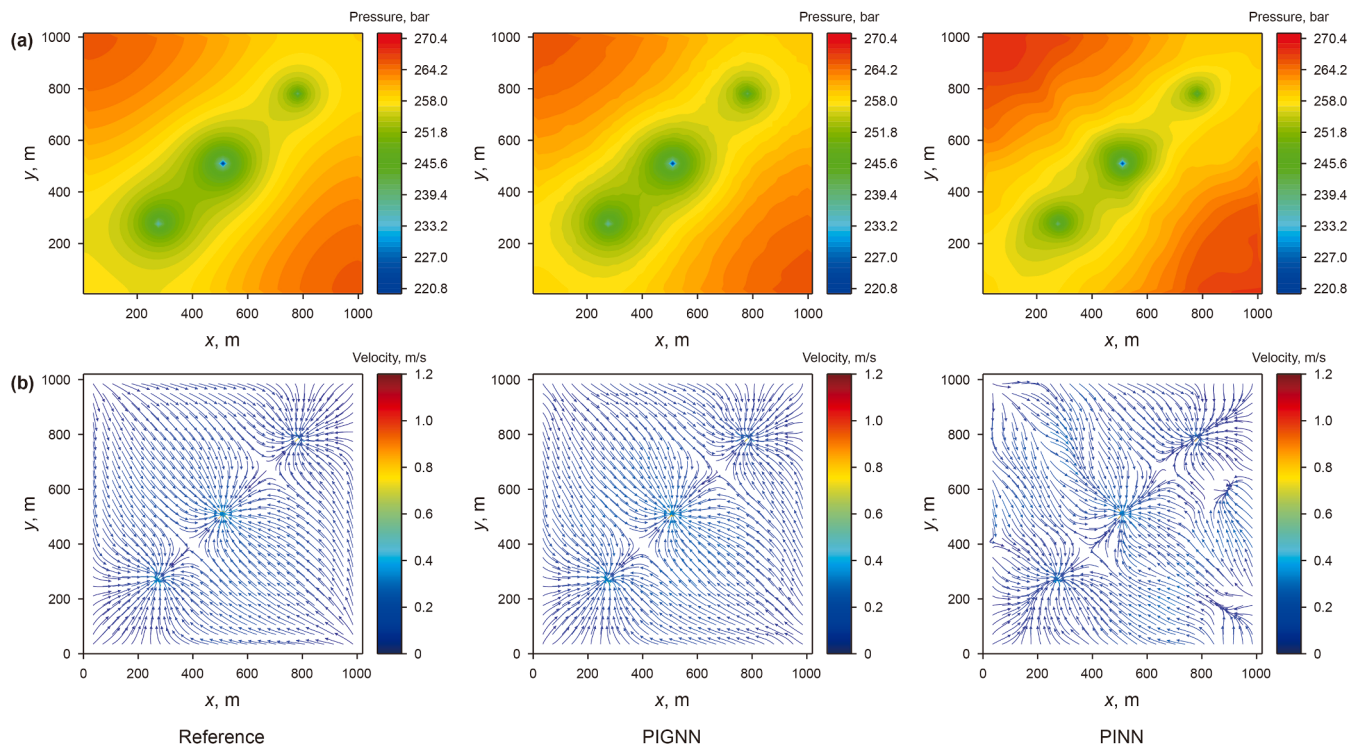


Fig. 7. Pressure (a) and velocity (b) fields obtained by numerical simulation (left), PIGNN model (middle), and PINN model (right) at time step 100.

source and sink characteristics at the well points, leading to poor convergence and inaccurate predictions in regions with large pressure gradients. To quantify the prediction performance of both models for bottomhole pressure, the RMSE and MAPE in the

predicted bottomhole pressure for the three wells was calculated, as shown in Table 2.

After multiple comparisons, it is evident that, under the same number of iterations, the prediction accuracy of the

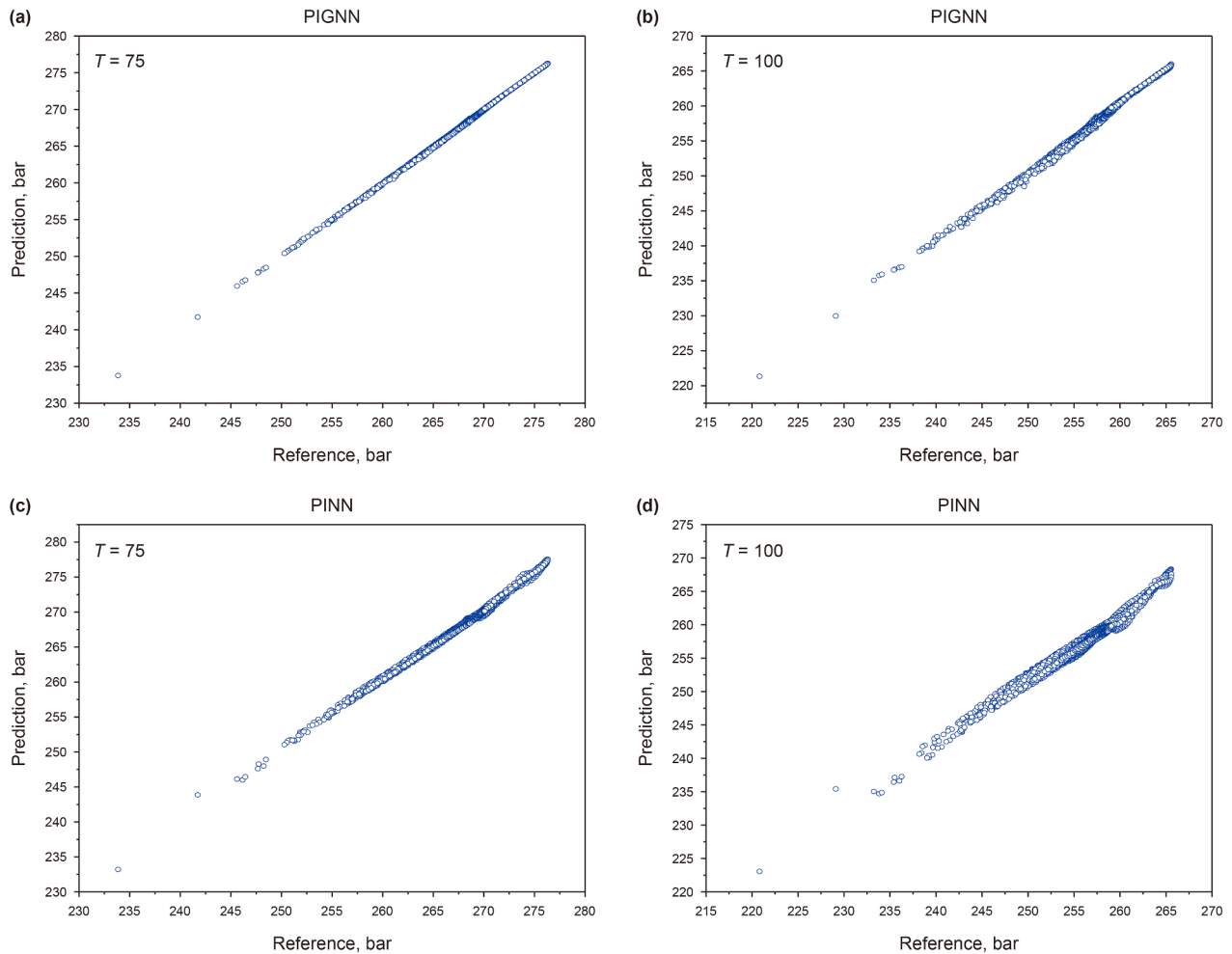


Fig. 8. Correlation between the reference and predicted pressures at time steps 75 (a, c) and 100 (b, d).

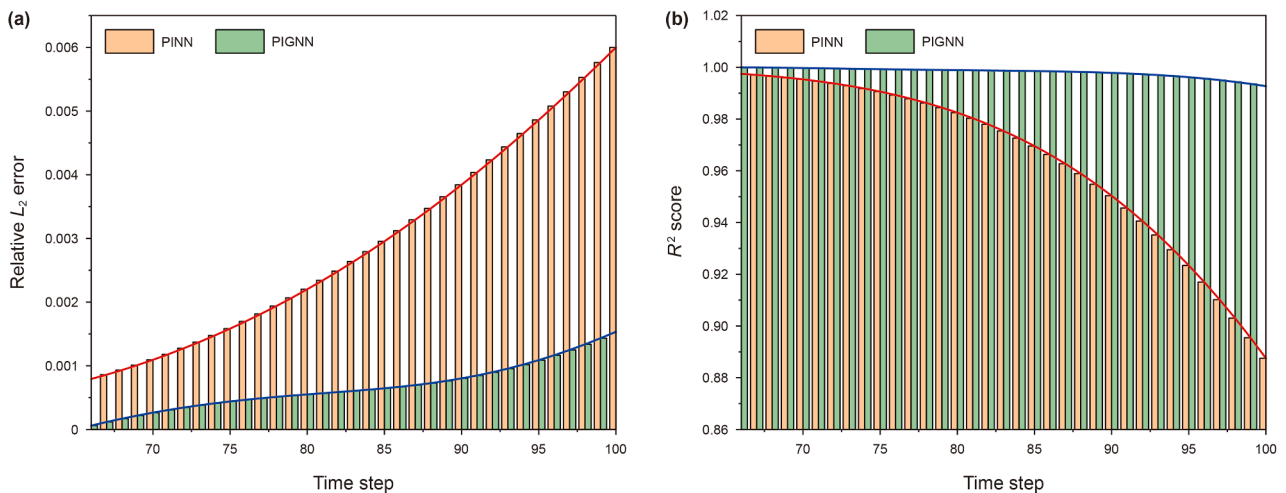


Fig. 9. The relative L_2 error (a) and R^2 score (b) obtained by PIGNN and PINN models on the test dataset.

PIGNN model is superior to that of the PINN model. The PIGNN model is better equipped to handle fluid flow and pressure distribution in reservoirs with planar heterogeneity. This effectiveness may stem from the ability of the PIGNN

model to accurately calculate the residual of the discrete percolation partial differential equation via predefined graph convolutional kernels, while simultaneously predicting the pressure of a target grid cell for the next step using

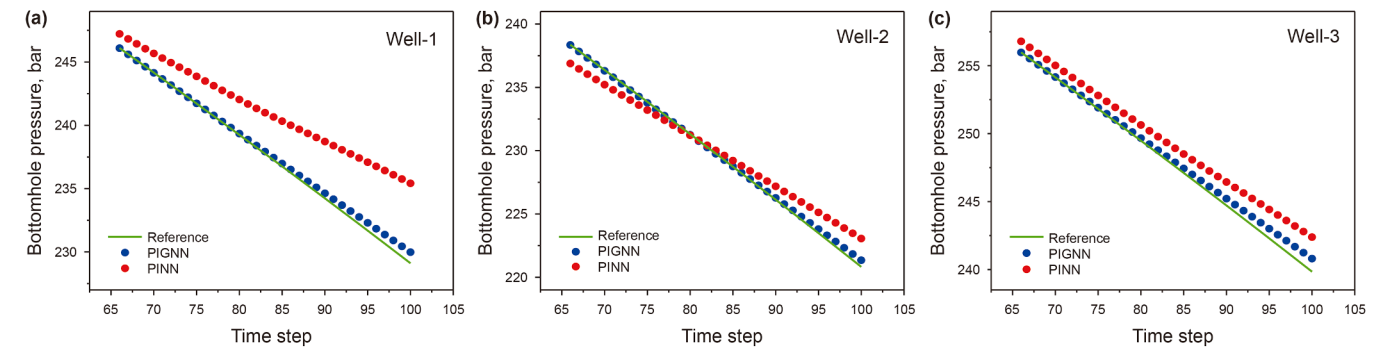


Fig. 10. The bottomhole pressures of well-1 (a), well-2 (b), and well-3 (c) obtained by numerical simulation, PIGNN model, and PINN model.

Table 2
RMSE and MAPE of bottomhole pressures obtained by PIGNN and PINN models.

Model	RMSE			MAPE		
	Well-1	Well-2	Well-3	Well-1	Well-2	Well-3
PIGNN	0.418	0.231	0.485	1.393×10^{-3}	8.590×10^{-4}	1.617×10^{-3}
PINN	4.158	1.349	1.647	1.770×10^{-2}	6.011×10^{-3}	6.718×10^{-3}

information from the target and adjacent cells at the previous time step.

3.3. Effect of different training data volumes

In the previous section, the training data consisted of 65 time steps, while the prediction data covered 35 time steps, with the training data accounting for 65% of the total dataset. However, in some real-world scenarios, the available data might not meet the requirements for training data. Therefore, in this section, we

investigated and compared the prediction performance of the PIGNN and PINN models under different training data volumes, testing the robustness of the PIGNN model when the training data was limited.

When the training dataset consists of 35 time steps, the prediction results of the PIGNN and PINN models at time steps 45 and 75 are shown in Fig. 11. When the training dataset consists of 50 time steps, the prediction results at steps 60 and 85 are shown in Fig. 12. The study reveals that the performance of the PIGNN model is less affected by the size of the training dataset, whereas the

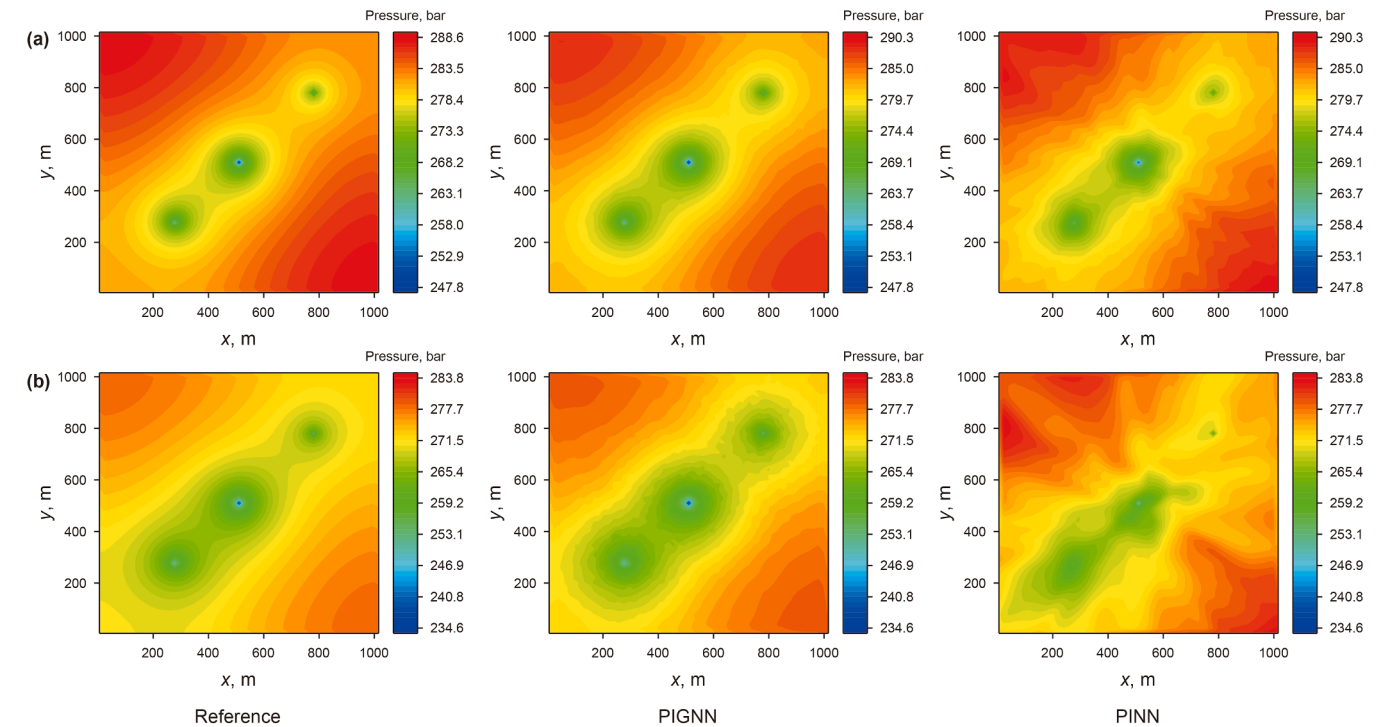


Fig. 11. The pressure fields obtained by numerical simulation (left), PIGNN model (middle), and PINN model (right) at time steps 45 (a) and 75 (b) when the training set is 35 time steps.

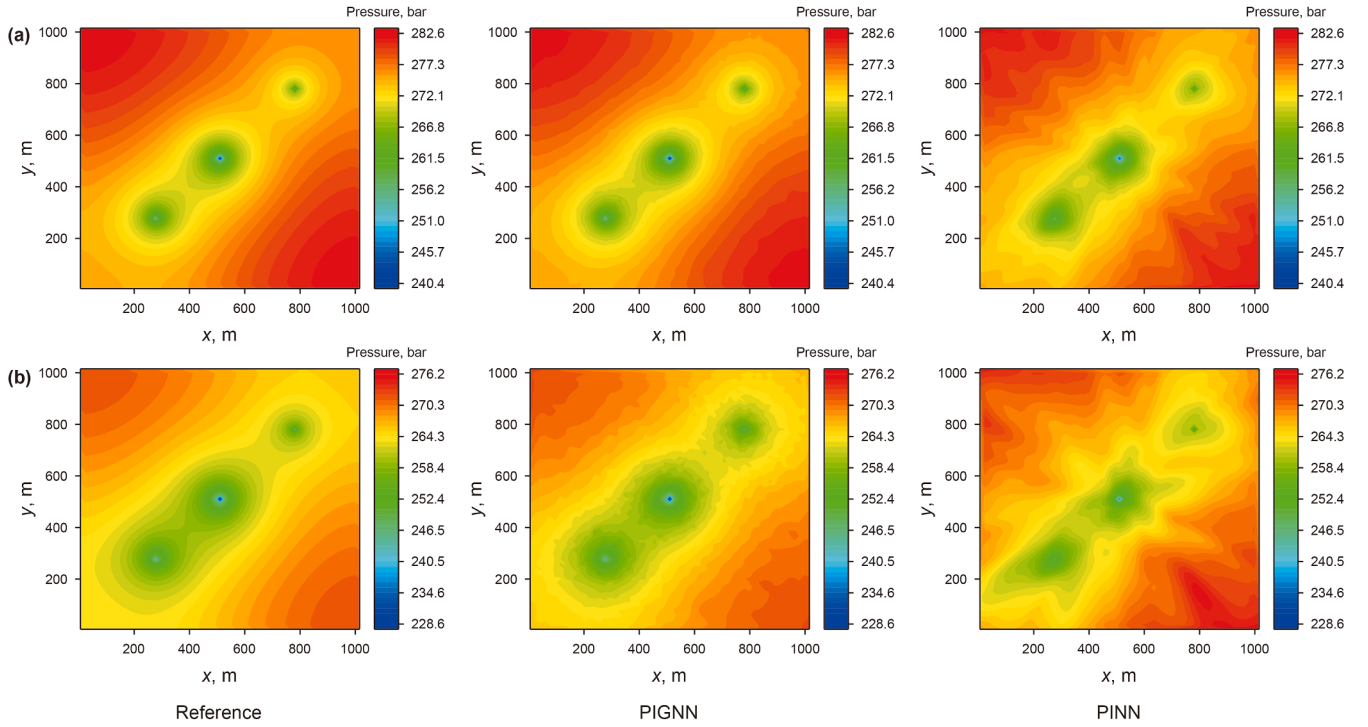


Fig. 12. The pressure fields obtained by numerical simulation (left), PIGNN model (middle), and PINN model (right) at time steps 60 (a) and 85 (b) when the training set is 50 time steps.

prediction accuracy of the PINN model deteriorates significantly as the training data decreases, indicating that the PIGNN model exhibits better robustness than the PINN model. This may be because the PINN model relies solely on the data features of the target grid and the differential terms at the grid points. When the amount of data is limited, the model learns fewer features, leading to poorer performance. In contrast, the PIGNN model considers the influence of surrounding grids on the prediction of the target grid, providing a broader view when predicting the pressure of the target grid. This increases the influence of the physical constraint term in the loss function, allowing the model to learn more features and mitigating the negative impact of insufficient data, thus slowing down the rate of performance degradation.

Fig. 13 presents histograms of the relative L_2 errors on the test dataset for models trained with varying amounts of data (represented by the number of training time steps). A lower relative L_2 error is generally indicative of superior performance in regression prediction. Specifically, with 35 training time steps, the PIGNN model achieved a prediction error of 2.047×10^{-3} , and with 50 time steps, the error was 1.761×10^{-3} . These errors represent reductions of 56.22% and 58.63%, respectively, compared to the prediction errors of the PINN model under the same conditions (4.677×10^{-3} and 4.257×10^{-3}). This comparison suggests that the PIGNN model exhibits enhanced predictive robustness. Lower L_2 errors demonstrate the model's superiority in regression prediction. From the figure, it is evident that the PIGNN model exhibits

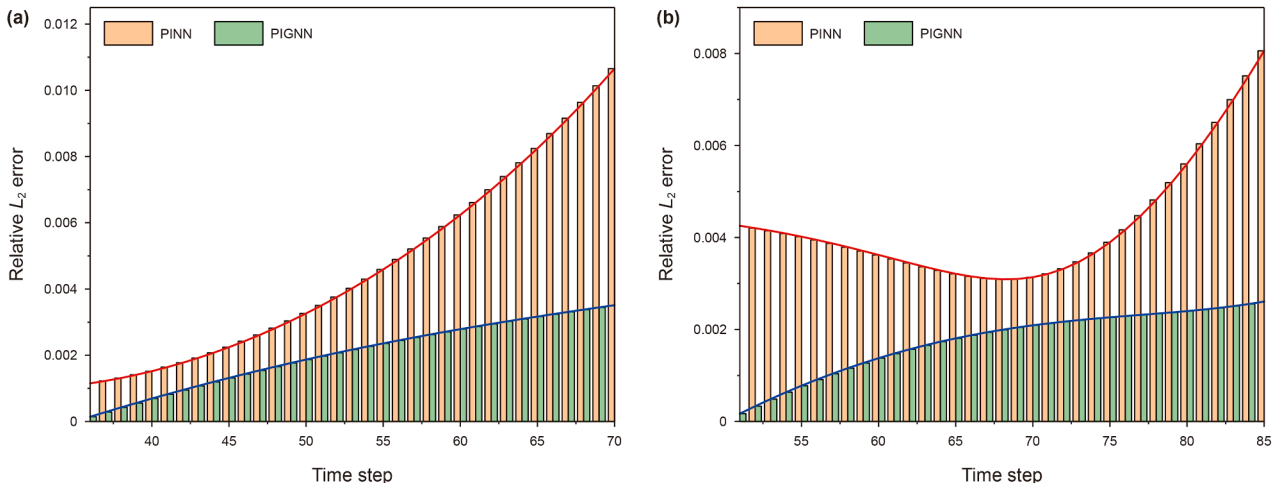


Fig. 13. The relative L_2 errors obtained by PIGNN and PINN models on the test dataset when the training set is 35 (a) and 50 (b) time steps, respectively.

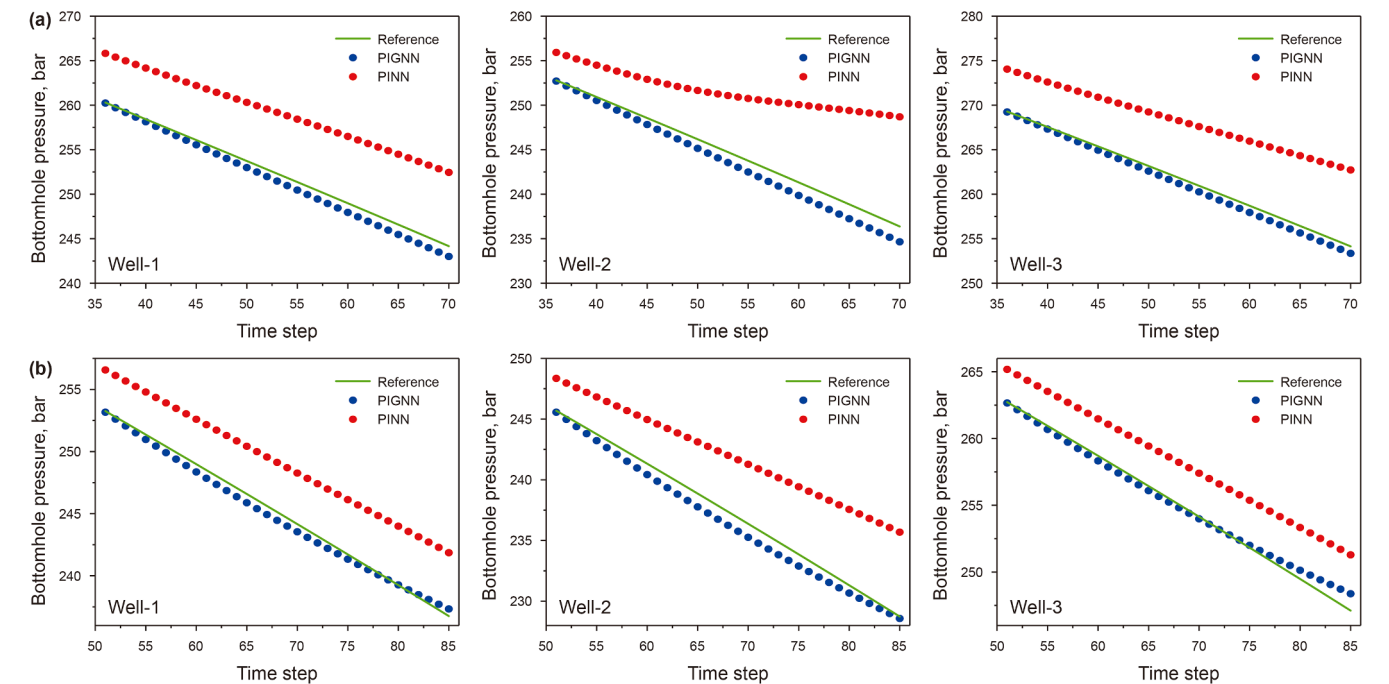


Fig. 14. The bottomhole pressures of well-1 (left), well-2 (middle), and well-3 (right) obtained by numerical simulation, PIGNN model, and PINN model when the training set is 35 (a) and 50 (b) time steps.

Table 3
RMSE and MAPE of bottomhole pressures obtained by PIGNN and PINN models when the training set is 35 time steps.

Model	RMSE			MAPE		
	Well-1	Well-2	Well-3	Well-1	Well-2	Well-3
PIGNN	0.907	1.368	0.653	3.609×10^{-3}	5.947×10^{-3}	2.470×10^{-3}
PINN	7.549	8.565	7.039	0.0324	0.0378	0.0282

Table 4
RMSE and MAPE of bottomhole pressures obtained by PIGNN and PINN models when the training set is 50 time steps.

Model	RMSE			MAPE		
	Well-1	Well-2	Well-3	Well-1	Well-2	Well-3
PIGNN	0.541	0.966	0.517	2.181×10^{-3}	4.418×10^{-3}	1.726×10^{-3}
PINN	4.493	5.615	3.480	0.0199	0.0269	0.0144

better predictive performance. The L_2 error for each time step in the test set increases as the time steps progress, likely due to the cumulative prediction errors of the model. This aligns with the classic theory of error accumulation in neural networks.

Fig. 14 shows the predicted bottomhole pressures for the test set using both models. Comparing the predicted bottomhole pressures of the three production wells with the reference pressures reveals that the PINN model's predictions significantly deviate from the actual bottomhole pressures, while the PIGNN model's predictions align well with the actual pressure curves. One explanation for the PINN model's accuracy limitations, especially near well grids, involves challenges such as insufficient data to learn the governing flow physics and slow convergence due to large pressure drops. The PIGNN model circumvents these issues through its design: its graph neural network architecture, combined with predefined graph convolutional kernels, enables it to effectively process structural information between grid cells and embed physical laws, thus facilitating accurate prediction of grid pressure. These results further demonstrate the robustness and superiority of the PIGNN model. To

quantitatively compare the prediction performance of the models, the RMSE and MAPE for the predicted bottomhole pressures were calculated, as shown in Tables 3 and 4. The results indicate that the prediction accuracy of the PIGNN model is significantly higher than that of the PINN model, which is consistent with the conclusions of the above.

4. Conclusions

In this work, we proposed the PIGNN model, a new framework for graph-based deep learning applied to subsurface fluid flow, aimed at predicting reservoir permeability fields defined on commonly used unstructured grids. Leveraging the characteristics of physical governing equations and graph neural networks, the PIGNN model not only incorporates physical information like the classical PINN model but also considers information from the surroundings of the target, enhancing the model's prediction accuracy and suitability for irregular grid numerical simulations. The graph convolution module proposed in this study seamlessly

integrates physical equations, boundary conditions, and fluxes between grids into the model's residual calculation, improving computational efficiency.

The optimized model was applied to pressure field prediction in spatially heterogeneous reservoirs; the predicted average L_2 error and R^2 score were 6.710×10^{-4} and 0.998, respectively, by which the model's effectiveness was validated. The prediction performance of the PIGNN model was compared with that of the classic PINN model; the PIGNN model's L_2 error was reduced by 76.93% compared to the PINN model, the R^2 score was increased by 3.56% compared to the PINN model, and as the time step increased, the PIGNN model's prediction error grew slower compared to the PINN model. In addition, a better pressure fitting effect at well points was exhibited by the PIGNN model; the average RMSE and MAPE errors were reduced by 84.15% and 87.28%, respectively, compared to the PINN model. This indicates that a better convergence effect is possessed by the PIGNN model in regions where pressure changes rapidly. The prediction accuracy of the two models when the amount of training data is small was discussed, and the robustness of the two models regarding the amount of training data was compared. When the prediction time steps were reduced from 65 to 50 and 35, the PIGNN model's L_2 errors were 2.047×10^{-3} and 1.761×10^{-3} respectively; compared to the PINN model's errors, they were reduced by 56.22% and 58.63%, by which the robustness of the PIGNN model and its superiority over the PINN model were proven. It is proven by these results that the PIGNN model proposed by the research is superior to the existing PINN model in predicting fluid flow in heterogeneous reservoirs.

Although the proposed PIGNN model exhibits strong performance in addressing fluid flow in heterogeneous oil reservoirs, certain aspects remain unexplored, and several technical challenges still exist. For example, the current study is based on a single-phase reservoir flow model and only considers two-dimensional planar flow, without accounting for water flooding scenarios, oil–water two-phase flow, or three-dimensional flow dynamics. Considering these limitations, future work will aim to extend the model to two-phase and three-phase flow, three-dimensional simulations, and further reduce reliance on labeled data.

CRedit authorship contribution statement

Hai-Yang Chen: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization. **Liang Xue:** Writing – review & editing, Supervision, Project administration, Data curation, Conceptualization. **Li Liu:** Writing – review & editing, Visualization, Validation, Investigation, Conceptualization. **Gao-Feng Zou:** Writing – review & editing, Visualization, Investigation, Conceptualization. **Jiang-Xia Han:** Writing – review & editing, Validation, Supervision, Data curation. **Yu-Bin Dong:** Visualization, Investigation. **Meng-Ze Cong:** Visualization, Investigation, Conceptualization. **Yue-Tian Liu:** Supervision. **Seyed Mojtaba Hosseini-Nasab:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 52274048) and Beijing Natural Science Foundation (No. 3222037).

References

- Atadeger, A., Sheth, S., Vera, G., et al., 2022. Deep learning-based proxy models to simulate subsurface flow of three-dimensional reservoir systems. In: ECMOR 2022. European Association of Geoscientists & Engineers, pp. 1–32. <https://doi.org/10.3997/j2214-4609.202244049>.
- Chiu, P.-H., Wong, J.C., Ooi, C., et al., 2022. CAN-PINN: A fast physics-informed neural network based on coupled-automatic-numerical differentiation method. *Comput. Methods Appl. Mech. Eng.* 395, 114909. <https://doi.org/10.1016/j.cma.2022.114909>.
- Conrad, P.R., Marzouk, Y.M., Pillai, N.S., et al., 2016. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *J. Am. Stat. Assoc.* 111 (516), 1591–1607. <https://doi.org/10.1080/01621459.2015.1096787>.
- Ghoreishian Amiri, S.A., Sadrnejad, S.A., Ghasemzadeh, H., 2017. A hybrid numerical model for multiphase fluid flow in a deformable porous medium. *Appl. Math. Model.* 45, 881–899. <https://doi.org/10.1016/j.apm.2017.01.042>.
- Hamdi, H., Couckuyt, I., Sousa, M.C., et al., 2017. Gaussian processes for history-matching: Application to an unconventional gas reservoir. *Comput. Geosci.* 21 (2), 267–287. <https://doi.org/10.1007/s10596-016-9611-2>.
- Han, J., Xue, L., 2023. Multiple production time series forecasting using deepar and probabilistic forecasting. In: SPE Annual Technical Conference and Exhibition. <https://doi.org/10.2118/214769-MS>.
- Han, J.-X., Xue, L., Wei, Y.-S., et al., 2023. Physics-informed neural network-based petroleum reservoir simulation with sparse data using domain decomposition. *Pet. Sci.* 20 (6), 3450–3460. <https://doi.org/10.1016/j.petsci.2023.10.019>.
- Han, J., Xue, L., Jia, Y., et al., 2024. Prediction of porous media fluid flow with spatial heterogeneity using criss-cross physics-informed convolutional neural networks. *CMES-Comput. Model. Eng. Sci.* 140 (2), 1323–1340. <https://doi.org/10.32604/cmescs.2023.031093>.
- Hawthorne, H., Mercer, F., Tzeng, D., et al., 2023. Graph neural network enhancing recommendation system based on social relationship and attention mechanism. *Res. Square*. <https://doi.org/10.21203/rs.3.rs-3181316/v1>.
- Hu, H., Qi, L., Chao, X., 2024. Physics-informed Neural Networks (PINN) for computational solid mechanics: Numerical frameworks and applications. *Thin-Walled Struct.* 205, 112495. <https://doi.org/10.1016/j.tws.2024.112495>.
- Jain, T., Patel, R.G., Trivedi, J., 2017. Application of polynomial chaos theory as an accurate and computationally efficient proxy model for heterogeneous steam-assisted gravity drainage reservoirs. *Energy Sci. Eng.* 5 (5), 270–289. <https://doi.org/10.1002/ese3.177>.
- Kang, M.-L., Zhou, J., Zhang, J., et al., 2025. An integrated method of data-driven and mechanism models for formation evaluation with logs. *Pet. Sci.* 22 (3), 1110–1124. <https://doi.org/10.1016/j.petsci.2025.01.004>.
- Karumuri, S., Tripathy, R., Bilonis, I., et al., 2020. Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. *J. Comput. Phys.* 404, 109120. <https://doi.org/10.1016/j.jcp.2019.109120>.
- Kingma, D.P., Dhariwal, P., 2018. Glow: Generative flow with invertible 1×1 convolutions. In: The 32nd International Conference on Neural Information Processing Systems. <https://doi.org/10.48550/arXiv.1807.03039>.
- Lazzara, M., Chevalier, M., Lapeyre, C., et al., 2023. NeuralODE-based latent trajectories into autoencoder architecture for surrogate modelling of parametric high-dimensional dynamical systems. In: Iliadis, L., Papaleonidas, A., Angelov, P., Jayne, C. (Eds.), *Artificial Neural Networks and Machine Learning-ICANN 2023*. Springer Nature Switzerland, Cham, pp. 497–508. https://doi.org/10.1007/978-3-031-44223-0_40.
- Li, D., Ionescu, C.-L., Muftakhidinov, B., et al., 2021. A new simulation layer optimization and permeability upscaling method for preserving critical reservoir heterogeneity. In: SPE Annual Caspian Technical Conference. <https://doi.org/10.2118/207074-MS>.
- Li, J., Zhang, T., Sun, S., et al., 2019. Numerical investigation of the POD reduced-order model for fast predictions of two-phase flows in porous media. *Int. J. Numer. Methods Heat Fluid Flow* 29 (11), 4167–4204. <https://doi.org/10.1108/HFF-02-2019-0129>.
- Meng, Z., Qian, Q., Xu, M., et al., 2023. PINN-FORM: A new physics-informed neural network for reliability analysis with partial differential equation. *Comput. Methods Appl. Mech. Eng.* 414, 116172. <https://doi.org/10.1016/j.cma.2023.116172>.
- Macnik, M.J., Durlafsky, L.J., Heinemann, Z.E., 2006. Sequentially adapted flow-based PEBI grids for reservoir simulation. *SPE J.* 11 (3), 317–327. <https://doi.org/10.2118/90009-PA>.
- Oishi, A., Yagawa, G., 2017. Computational mechanics enhanced by deep learning. *Comput. Methods Appl. Mech. Eng.* 327, 327–351. <https://doi.org/10.1016/j.cma.2017.08.040>.
- Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., et al., 2021. Learning mesh-based simulation with graph networks. *arXiv*. <https://doi.org/10.48550/arXiv.2010.03409>.
- Qu, H.-Y., Zhang, J.-L., Zhou, F.-J., et al., 2023. Evaluation of hydraulic fracturing of horizontal wells in tight reservoirs based on the deep neural network with physical constraints. *Pet. Sci.* 20 (2), 1129–1141. <https://doi.org/10.1016/j.petsci.2023.03.015>.
- Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>.

- Réau, M., Renaud, N., Xue, L.C., et al., 2023. DeepRank-GNN: A graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics* 39 (1), btac759. <https://doi.org/10.1093/bioinformatics/btac759>.
- Sant'Ana da Silva, E., Pedrini, H., Santos, A.L. dos, 2023. Applying graph neural networks to support decision making on collective intelligent transportation systems. *IEEE Transact. Netw. Serv. Manag.* 20 (4), 4085–4096. <https://doi.org/10.1109/TNSM.2023.3257993>.
- Shan, L., Liu, C., Liu, Y., et al., 2023. Physics-informed machine learning for solving partial differential equations in porous media. *Adv. Geo-Energ. Res.* 8 (1), 37–44. <https://doi.org/10.46690/ager.2023.04.04>.
- Shao, Q., Matthai, S., Driesner, T., et al., 2021. Predicting plume spreading during CO₂ geo-sequestration: Benchmarking a new hybrid finite element–finite volume compositional simulator with asynchronous time marching. *Comput. Geosci.* 25 (1), 299–323. <https://doi.org/10.1007/s10596-020-10006-1>.
- Shao, X., Liu, Z., Zhang, S., et al., 2023. PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh. *Build. Environ.* 232, 110056. <https://doi.org/10.1016/j.buildenv.2023.110056>.
- Syed, I., Liu, C.-H., Kelkar, M.G., et al., 2020. Improved distance based upgridding and diffuse source upscaling for high resolution geologic models. In: *SPE Annual Technical Conference and Exhibition*. <https://doi.org/10.2118/201727-MS>.
- Tarakanov, A., Elsheikh, A.H., 2019. Regression-based sparse polynomial chaos for uncertainty quantification of subsurface flow models. *J. Comput. Phys.* 399, 108909. <https://doi.org/10.1016/j.jcp.2019.108909>.
- Wang, N., Zhang, D., Chang, H., et al., 2020. Deep learning of subsurface flow via theory-guided neural network. *J. Hydrol.* 584, 124700. <https://doi.org/10.1016/j.jhydrol.2020.124700>.
- Wang, N., Chang, H., Zhang, D., 2021. Efficient uncertainty quantification and data assimilation via theory-guided convolutional neural network. *SPE J.* 26 (6), 4128–4156. <https://doi.org/10.2118/203904-PA>.
- Xu, R., Zhang, D., Rong, M., et al., 2021. Weak form theory-guided neural network (TgNN-wf) for deep learning of subsurface single- and two-phase flow. *J. Comput. Phys.* 436, 110318. <https://doi.org/10.1016/j.jcp.2021.110318>.
- Xue, L., Li, D., Dou, H., 2023. Artificial intelligence methods for oil and gas reservoir development: Current progresses and perspectives. *Adv. Geo-Energ. Res.* 10 (1), 65–70. <https://doi.org/10.46690/ager.2023.10.07>.
- Yuan, L., Park, H.S., Lejeune, E., 2022. Towards out of distribution generalization for problems in mechanics. *Comput. Methods Appl. Mech. Eng.* 400, 115569. <https://doi.org/10.1016/j.cma.2022.115569>.
- Zhang, Z., Yan, X., Liu, P., et al., 2023. A physics-informed convolutional neural network for the simulation and prediction of two-phase Darcy flows in heterogeneous porous media. *J. Comput. Phys.* 477, 111919. <https://doi.org/10.1016/j.jcp.2023.111919>.
- Zhao, Y., Jiang, C., Vega, M.A., et al., 2023. A comparative study of surrogate modeling of nonlinear dynamic systems. *J. Comput. Inf. Sci. Eng.* 23 (1), 011001. <https://doi.org/10.1115/1.4054039>.
- Zhu, L., Zhang, C., Zhang, Z., et al., 2020. High-precision calculation of gas saturation in organic shale pores using an intelligent fusion algorithm and a multi-mineral model. *Adv. Geo-Energ. Res.* 4 (2), 135–151. <https://doi.org/10.26804/ager.2020.02.03>.
- Zhu, Y., Zabarav, N., Koutsourelakis, P.-S., et al., 2019. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J. Comput. Phys.* 394, 56–81. <https://doi.org/10.1016/j.jcp.2019.05.024>.