International Journal of Mechanical Sciences 314 (2026) 111414

Contents lists available at ScienceDirect

International Journal of Mechanical Sciences

journal homepage: www.elsevier.com/locate/ijmecsci

# Physics-encoded recurrent graph neural network for long-term consolidation

Xiao-Xuan Chen [a], Bocheng Zeng [b], Xi Wang [a,*], Hao Sun [b], Zhen-Yu Yin [a,c,*]

[a] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[b] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[c] State Key Laboratory of Climate Resilience for Coastal Cities, The Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

*Keywords:*
Physics-encoded recurrent graph neural network
Physics-informed neural network
Consolidation around tunnels
Forward and inverse analysis
Sparse noisy data

ABSTRACT

Although physics-informed neural networks (PINNs) have become a leading paradigm for solving both forward and inverse partial differential equations (PDEs) in a unified framework, their efficacy is often hindered by long-term prediction instability in black-box time marching, training imbalances stemming from multiple physical loss functions, and the inherent difficulty of representing complex geometries within neural architectures. This study develops the first-ever physics-encoded recurrent graph neural network (PeRGNN), a novel graph-based architecture that inherently handles irregular geometries to solve the consolidation problem that is challenging in geomechanics. Continuous PINN incorporates residuals of PDEs, boundary conditions, initial conditions, and data discrepancies, without respecting causal evolution between time snapshots. In comparison, PeRGNN hard-encodes the temporal evolution laws, initial and boundary conditions into a recurrent unit. By treating spatial operators as learnable graph-based message-passing functions while strictly enforcing the time-marching structure, the framework establishes a strong inductive bias that ensures physical consistency. The efficacy of PeRGNN is validated through six geomechanical case studies involving different tunnel geometries, anisotropic material behaviours, forward and inverse analysis. PeRGNN achieves (1) zero-shot prediction for extended time spans, generalizing across initial conditions and consolidation parameters, (2) remarkable data efficiency while maintaining high accuracy on decimated meshes with node counts nearly two orders of magnitude lower than that traditionally required for multi-dimensional consolidation analysis, and (3) robust inverse modelling with sparse and noisy data. The proposed PeRGNN, thus, proves to be a promising method for real-world geotechnical analysis and deserves further development.

## 1. Introduction

The spatiotemporal evolution of physical systems governed by partial differential equations (PDEs) remains a cornerstone of mechanical and geotechnical sciences. In geomechanics, soil consolidation serves as a canonical benchmark, focusing on long-term mechanical responses of soils driven by the dissipation of excess pore water pressure and redistribution of effective stress. The classical consolidation theory was established by Terzaghi [1] and Biot [2]. While various analytical, semi-analytical, and numerical methods, such as finite element methods (FEM), are standard for solving these PDEs [3–5], they face bottlenecks in incorporating real-world monitoring data and are often ill-posed for inverse parameter estimation. Recently, data-driven models have emerged with strong capabilities to model complex geotechnical behaviours, including consolidation analysis [6–10]; however, they are often criticised for their "black-box" nature, dependence on large datasets, and limited generalizability [11–13].

Physics-informed neural networks (PINNs) have recently emerged as a powerful alternative for solving PDEs by combining data and physics [14–18]. Specifically, PINNs leverage both labelled data and physics laws by incorporating residuals of governing PDEs into the loss function of the neural network. Through iteratively minimizing this hybrid loss function, PINNs generate predictions that align with both the observed data and underlying physics. PINNs offer a unified framework for forward and inverse analysis, where unknown parameters can be identified as trainable variables during the optimization process [14,16,19–24]. This eliminates the need for new codes or intensive trial-and-error procedures common in traditional numerical schemes. Until now,

* Corresponding authors.
*E-mail addresses:* xiwang.wang@polyu.edu.hk (X. Wang), zhenyu.yin@polyu.edu.hk (Z.-Y. Yin).

https://doi.org/10.1016/j.ijmecsci.2026.111414
Received 13 October 2025; Received in revised form 7 January 2026; Accepted 14 February 2026
Available online 15 February 2026

PINNs have been applied in various scientific domains such as solid mechanics [19,25–28], fluid mechanics [29–31], geoengineering [21, 32–34], constitutive modelling [35–39], and so on.

Recent advancements in solving spatiotemporal PDEs through PINNs can be categorized into three methodological genres based on the treatment of the space-time domain. The first genre adopts a fully continuous formulation, employing automatic differentiation (AD) to compute derivatives over a collocation set. While it remains the dominant paradigm due to its straightforward manner and easy implementation [29,40–42], it suffers from significant optimization pathologies. Wang et al. [43] demonstrated that continuous-time PINNs often violate physical causality by biasing the model toward minimizing residuals as later times before resolving initial conditions. To mitigate these optimization pathologies while preserving the advantages of a continuous formulation, recent approaches such as DG-PINN employ temporal domain discretization to enforce physical causality and successfully simulated 3D turbulent flow [44]. Besides, substantial PINN variants have been introduced to enhance the performance of original continuous PINNs, such as XPINN [45] and PPINN [46]. The second genre introduces temporal discretization while retaining spatial continuity. These methods decouple temporal evolution from the spatial domain by adopting discrete time integration schemes, such as backward Euler or Runge-Kutta (RK) methods, while preserving a continuous spatial representation within each time step [16,47]. This temporal discretization effectively reduces the need for dense collocation sampling over time. However, they still rely on AD for spatial derivatives, which can become computationally prohibitive for high-dimensional problems or those involving complex geometries where dense spatial sampling is unavoidable [48].

The third category discretizes both spatial and temporal domains. These approaches typically leverage graph-based or finite-difference schemes for approximating derivatives in both space and time to circumvent the computationally expensive process of AD. While convolutional neural network (CNN) architectures have shown promise in this category [49–53], their inherent requirement for regular grids limit their flexibility in handling irregular geometries common in geotechnical problems. Some novel methodologies have emerged to enhance PINN performance for complex geometries, such as domain decomposition [23,54], incorporation of finite element-based discretization [55, 56], and hard-encoding of boundary conditions [57,58]. Besides, a new Monte Carlo PINN model has successfully extended derivative approximations to solve fractional PDEs within irregular boundaries [59]. Parallel to these efforts, attention has shifted toward graph neural networks (GNNs), which enable discrete PINNs to operate on unstructured meshes via graph-based discretization techniques [60–63]. For example, the finite volume method is integrated with GNN to model incompressible flow on unstructured grids, yet this computational paradigm is constrained to steady-state PDEs [64]. Another state-of-the-art model is the PhyMPGN [65], a novel graph-learning architecture designed to simulate cylinder flow at high Reynolds number. However, this research primarily focuses on forward analysis and does not explore the framework's applicability or performance for inverse problems, such as parameter identification from sparse and noisy data. Despite these advancements, current GNN-based discrete PINNs have not yet addressed several critical challenges in geotechnical applications. Most notably, the incorporation of anisotropic material behaviours and the solution of inverse problems. As such, the integration of graph learning with PINNs in geomechanics still remains in its infancy.

In geoengineering practice, the application of PINNs to consolidation analysis has transitioned from one-dimensional (1D) Terzaghi's classical consolidation theory [66,67] to more complex 1D consolidation cases [68]. Technical refinements are proposed to improve convergence for 1D consolidation, such as affine transformations [68], adaptive loss weighting strategies [69], and hard-encoded boundary constraints [70]. The application of PINNs to 2D consolidation has received relatively limited attention. While early efforts applied standard PINN architectures to 2D consolidation domains [71–73], a more recent work done by Guo and Yin [74] introduced a high-order RK-based PINN through temporal discretization. This approach improved training efficiency and stability for modelling multi-dimensional consolidation in regular domains. However, these pioneering studies were restricted to regular geometries. A notable exception is the work done by Xie et al. [75], which investigated 2D rheological consolidation around twin tunnels using a traditional PINN approach. Despite this advancement, the study relied on >60,000 training epochs and over 40,000 collocation points, revealing the inefficiency of traditional PINNs when dealing with irregular geometries and long-term simulations. This is because this study incorporated physics constraints in a soft manner, which often led to complex loss landscapes and increasing training difficulty [76]. Besides, the reliance on spatiotemporally continuous formulations often requires a large number of collocation points and extensive training epochs to achieve acceptable accuracy. Furthermore, inverse analysis is not included in this study.

In summary, most existing PINN-based consolidation models are restricted to simple geometries and belong to either fully continuous space-time formulations or those with continuous space and discretized time. Besides, these models lacked generalizability, being highly case-specific and requiring re-training when applied to new engineering scenarios. To date, no study has explored the fully discretized PINNs with GNN architectures and hard-encoded physics constraints for consolidation problems involving irregular geometries, anisotropic behaviours and inverse analysis. These methodological gaps have led to several critical challenges in practical applications: (1) multiple loss terms lead to unstable training and slow convergence [76,77], (2) a large number of collocation points in both space and time is required [75], imposing a significant computational burden, and (3) previous models are typically tailored to specific loading conditions or soil parameters, while their potential for generalization to new scenarios has received little exploration.

To this end, this study proposes a physics-encoded recurrent graph neural network (PeRGNN), a spatiotemporally discrete framework designed to approximate the evolution of geotechnical systems over space and time. In particular, this model offers four key contributions: (1) PeRGNN adopts a discretized spatiotemporal mesh and hard-encodes the discrete time-marching structure and boundary conditions into the network architecture, thereby simplifying the loss functions and reducing the need for dense training data; (2) the graph-based message passing strategy in PeRGNN naturally accommodates irregular spatial domains without requiring coordinate transformations or mesh regularization; (3) unlike existing GNN-based discrete PINNs, PeRGNN is capable of handling anisotropic material behaviours and performing inverse analysis to identify unknown material parameters, even under sparse and noisy data; (4) once trained, the model can predict the spatiotemporal dynamics by performing the recurrent unit merely based on initial conditions, and further generalize on unseen scenarios (variations in material properties and initial states) without re-training.

The remainder of this study is structured as follows. Section 2 presents the benchmark 2D consolidation problems and provides a detailed architectural overview of the proposed PeRGNN framework. Section 3 evaluates the model's performance across six numerical case studies, focusing on long-term extrapolation, generalizability to unseen physical parameters, and robust parameter identification. Section 4 provides a comparative analysis with traditional numerical solvers and demonstrates its predictive stability when trained on extremely sparse datasets. Section 5 summarizes the key findings, potential engineering applications, current limitations, and future work.

## 2. Methodology

This section presents the governing PDEs for 2D consolidation and computational architecture of the proposed PeRGNN framework, including the recurrent unit, integration of graph-based message passing

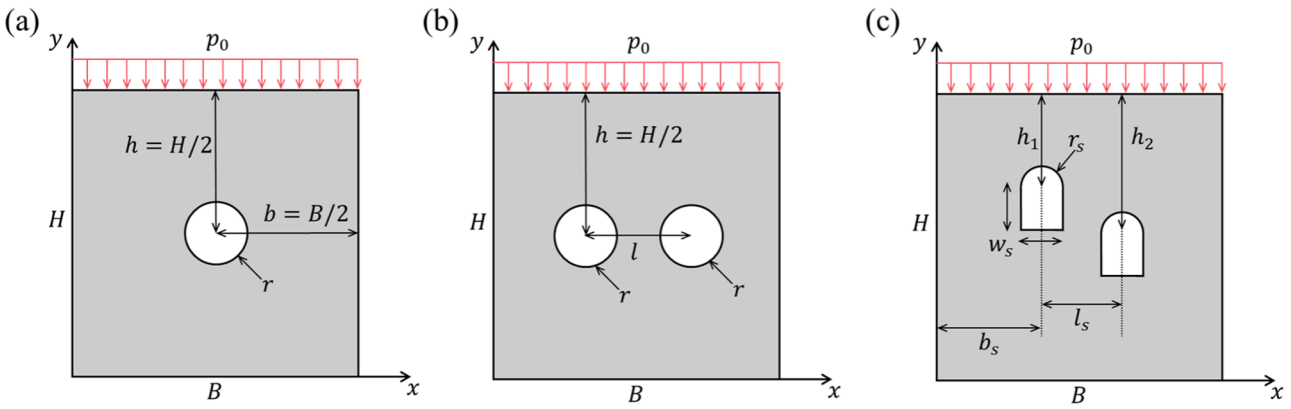


**Fig. 1.** Problem setup of three complex tunnel geometries: (a) A single tunnel in isotropic soil; (b) Twin circular tunnels aligned horizontally in anisotropic soil; (c) Vertically offset straight-wall arched twin tunnels in anisotropic soil.

**Table 1**
Geometric parameters for all computational tunnel configurations (unit: meters).

| $B$ | $H$ | $b$ | $h$ | $r$ | $l$ | $r_s$ | $h_1$ | $h_2$ | $l_s$ | $b_s$ | $w_s$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 40 | 20 | 20 | 4 | 20 | 2.5 | 15 | 22.5 | 10 | 15 | 5 |

with hard-encoded physical constraints, network training schemes, and data acquisition.

### 2.1. Problem statement

The theory of consolidation describes the dissipation of excess pore water pressure $u$ in a porous medium subjected to an instantaneous, uniformly distributed load $p_0$, as illustrated in Fig. 1. Based on Terzaghi's consolidation theory, the general PDE governing 2D consolidation in a porous medium can be given by:

$$\frac{\partial u}{\partial t} = C_v \frac{\partial^2 u}{\partial x^2} + C_h \frac{\partial^2 u}{\partial y^2} \tag{1}$$

where $C_v$ and $C_h$ represent the soil's consolidation coefficient in $x$ and $y$ directions, respectively. For isotropic soil, $C_v = C_h$.

The initial condition, reflecting the drainage state, is defined as:

$$u|_{t=0} = p_0 \tag{2}$$

To accommodate varying transportation requirements and geological conditions, tunnel cross-sectional geometries and surrounding soil properties can differ significantly. This research systematically investigates consolidation behaviours across three tunnel configurations of increasing complexity: (1) a single tunnel embedded in isotropic soil (Fig. 1a), (2) horizontally aligned twin circular tunnels in anisotropic soil (Fig. 1b), and (3) vertically offset straight-wall arched twin tunnels in anisoropic soil (Fig. 1c). The computational domain for all configurations has a width $B = 40$ m and a depth $H = 40$ m. Both circular single and twin tunnels have a radius $r = 4$ m, and their centres are located at a depth $h = H/2$ below the ground surface. For the single tunnel in isotropic soil, the horizontal distance from the tunnel centre to the left/right boundaries is $b = B/2$. For the circular twin tunnels in anisotropic soil, the distance between the two tunnel centres is denoted as $l$. The straight-wall arched twin tunnels characterise a more complex geometry. Their top semi-circular sections have a radius $r_s$, while the bottom square sections have a width $w_s$. The centres of the upper arches are vertically offset at depths $h_1$ and $h_2$, respectively. The horizontal centre-to-centre distance between tunnels is defined as $l_s$, and the distance from the left tunnel's centerline to the left domain boundary is specified as $b_s$. The values of all geometric parameters for computational models are summarized in Table 1.

All boundaries of the computational domain and the tunnel lining are defined as drained boundaries, employing a Dirichlet boundary condition:

$$u|_{(x,y)\in\Gamma} = 0 \tag{3}$$

where $\Gamma$ represents the boundary of the computational domain.

The analytical framework of this research is based on the following assumptions: (1) the soil is assumed to be a homogeneous isotropic/anisotropic material; (2) the tunnel is assumed to be infinitely long in the longitudinal direction, reducing the problem to a plane-strain formulation; (3) both the soil skeleton and pore water are incompressible, and pore water flow follows Darcy's law; (4) small strain kinematics is adopted.

### 2.2. Overview of the pergnn framework

In this section, we introduce a unified PeRGNN framework (Fig. 2) for solving both forward and inverse spatiotemporal problems. To reflect realistic engineering scenarios where high-resolution spatiotemporal data are difficult and expensive to obtain, the input data on unstructured grids are first downsampled to a lower resolution, serving as the training data for PeRGNN. These downsampled inputs on coarse grids are then processed by a recurrent unit, which learns to iteratively generate predictions over time. A more detailed breakdown of the PeRGNN computational paradigm is presented in the following subsections.

#### 2.2.1. Recurrent unit

The computational paradigm of the recurrent unit follows an Encoder-Processor-Decoder framework. First, there are two encoder networks, i.e., node encoder and edge encoder, applied to embed node and edge features using multi-layer processors (MLPs). For each node $i$ in the graph, the node encoder takes the state variable. $u_i^t$, spatial coordinate $\mathbf{x}_i$ and one-hot code of node type $\mathbf{C}_i$ as input features, and output node embedding $\mathbf{h}_i^0$. For each edge $(i, j)$, the edge encoder takes the relative offset $u_{ij}^t = u_j^t - u_i^t$, displacement $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ and coordination distance $d_{ij} = \|\mathbf{x}_{ij}\|_2$ as input features and output edge embedding $\mathbf{e}_{ij}$.

The processor consists of $L$ message passing neural network (MPNN) layers aiming to update node embeddings. Each MPNN layer consists of two key components: (1) a message function $\phi$ that computes edge-wise interactions, and (2) an update function $\gamma$ that aggregates messages to evolve node states, where both $\phi$ and $\gamma$ are MLPs. Specifically, the edge message in the $l^{\text{th}}$ MPNN layer $m_{ij}^l$ for the edge $(i, j)$ is constructed as:

$$m_{ij}^l = \phi\left(\mathbf{h}_i^l,\ \mathbf{h}_j^l - \mathbf{h}_i^l, \mathbf{e}_{ij}\right) \tag{4}$$

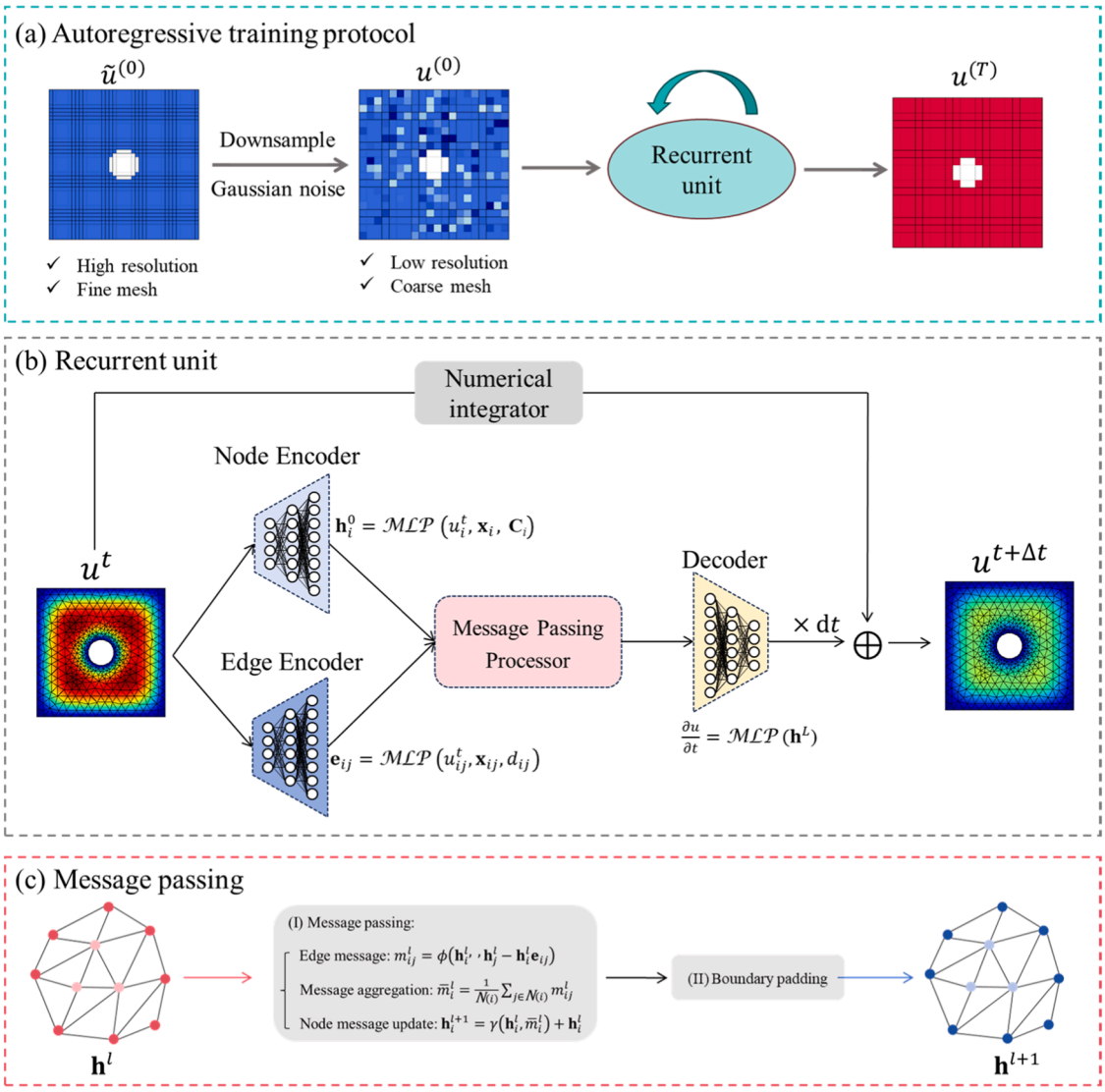


**Fig. 2.** Overview of the PeRGNN framework. (a) The autoregressive training protocol; (b) The internal architecture of the recurrent unit at the time step $t$; (c) One layer of the graph message passing after the encoder between layer $l$ and $l + 1$.

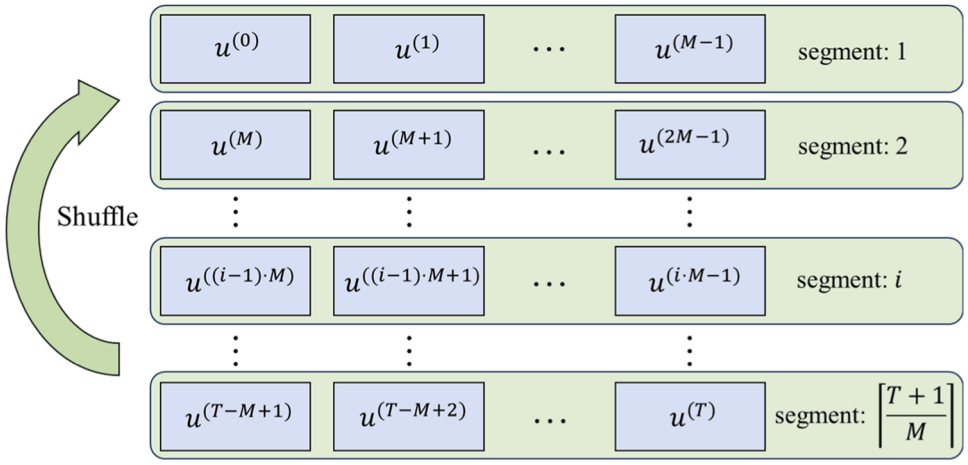


**Fig. 3.** The short time segments for the entire time series of the training samples.

Then, messages from all neighbours $N(i)$ are aggregated via mean pooling:

**Table 2**
The details of spatiotemporal discretization parameters for single and twin tunnel configurations.

| Discretization | Tunnel configurations | High-fidelity resolution | Downsampled resolution |
|---|---|---|---|
| Spatial | Single tunnel | 6724 meshes (3478 nodes) | 858 meshes (471 nodes) |
| | Circular twin tunnels | 6818 meshes (3540 nodes) | 1284 meshes (703 nodes) |
| | Straight-wall arched twin tunnels | 6664 meshes (3466 nodes) | 1336 meshes (728 nodes) |
| Temporal | Single tunnel<br>Circular twin tunnels | 4000 steps (d$t$ = 0.01 year) | 1000 steps (d$t$ = 0.04 year) |
| | Straight-wall arched twin tunnels | | |

$$\overline{m}_i^l = \frac{1}{N(i)} \sum_{j \in \mathrm{N}(i)} m_{ij}^l \tag{5}$$

Finally, the node state is updated through:

$$\mathbf{h}_i^{l+1} = \gamma\left(\mathbf{h}_i^l, \overline{m}_i^l\right) + \mathbf{h}_i^l, \ 0 \le l < L-1 \tag{6}$$

$$\mathbf{h}_i^L = \gamma\left(\mathbf{h}_i^{L-1}, \overline{m}_i^{L-1}\right)$$

After $L$ MPNN layers, the decoder network would take the output of the last MPNN as inputs, and transform the node embedding $\mathbf{h}^L$ in the high-dimensional latent space to the variables in physical dimensions. In this research, we design the decoder to approximate the second-order spatial derivatives in the governing equation, i.e., $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$, to bypass the need for numerical differentiation. While the spatial operator is learned, the temporal evolution is strictly enforced by the hard-coded time-stepping scheme. The residual connections [78] are applied to all intermediate MPNN layers except the final one, which can provide a direct path for gradient propagation and mitigate gradient issues during training.

The temporal derivative term in Eq. (1) is approximated by PeRGNN: $\frac{\partial u}{\partial t} = G(u;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denoets the trainable network parameters. Specifically, the PeRGNN model illustrated in Fig. 2 serves as a universal approximator for the time-evolution operator. The discrete form of this evolution, based on the forward Euler scheme, can be naturally embedded within the model as a specific case:

$$\frac{\partial u}{\partial t} = \frac{u^{t+\Delta t} - u^t}{\Delta t} \tag{7}$$

Accordingly, the solution $u$ at the next time step $t + \Delta t$ can be updated based on the current time step $t$:

$$u^{t+\Delta t} = u^t + \Delta t \cdot G(u^t;\boldsymbol{\theta}) \tag{8}$$

To achieve higher-order accuracy, explicit RK methods are alternatively adaptable. For instance, the second-order RK:

$$k_1 = G(u^t;\boldsymbol{\theta})$$

$$k_2 = G(u^t + \Delta t \cdot k_1;\boldsymbol{\theta}) \tag{9}$$

$$u^{t+\Delta t} = u^t + \frac{\Delta t}{2}(k_1 + k_2)$$

In this way, each time step updates the state based on the current value based on an explicit mathematical expression, preventing gradient

**Table 4**
Performance comparison of activation functions (Average over 20-year extrapolation in Case 1).

| Activation function | Avg. $R^2$ | Avg. MSE |
|---|---|---|
| ELU (selected) | 0.994 | 4.27e-04 |
| Sin | 0.991 | 4.44e-04 |
| Tanh | 0.985 | 4.73e-04 |
| Swish | 0.993 | 4.35e-04 |

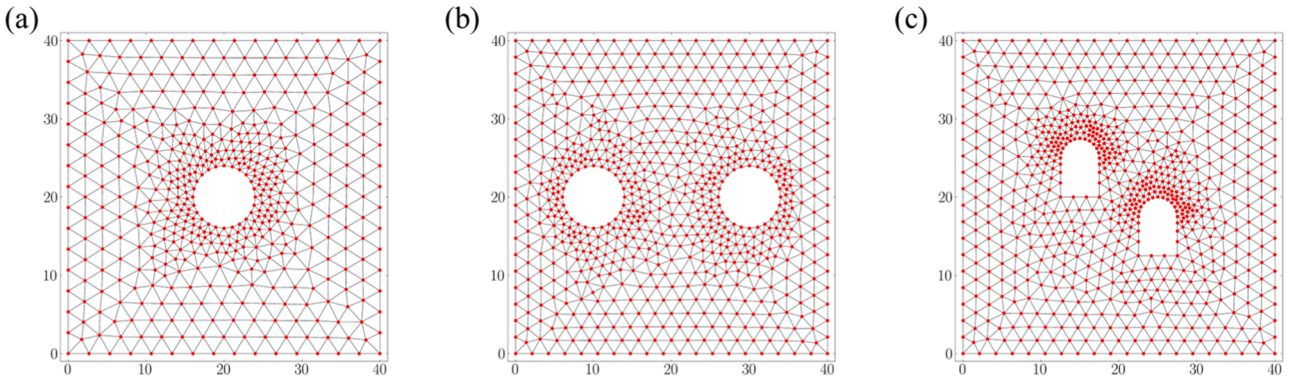


**Fig. 4.** The illustration of downsampled spatial mesh discretization for three tunnel configurations. (a) Single tunnel; (b) Circular twin tunnels; (c) Straight-wall arched twin tunnels.

**Table 3**
Details of PeRGNN configurations for six study cases.

| Study cases | Optimizer<br>Learning rate<br>Activation function<br>Integration scheme | Node encoder<br>Edge encoder | Decoder | Number of MPNN layers | Parameter counts | Time segment size |
|---|---|---|---|---|---|---|
| Case 1 | Adam | 6–32–32–64 | 64–32–32–1 | 3 | 146,593 | 15 |
| Case 2 | 0.0005 | 4–32–32–64 | 64–32–32–1 | | 146,593 | 10 |
| Case 3 | ELU($x$) | | 64–32–32–1 | | 146,593 | 10 |
| Case 4 | second-order RK | | 64–32–32–1 | | 146,594 | 10 |
| Case 5 | | | / | | 149,762 | 10 |
| Case 6 | | | / | | 149,762 | 10 |

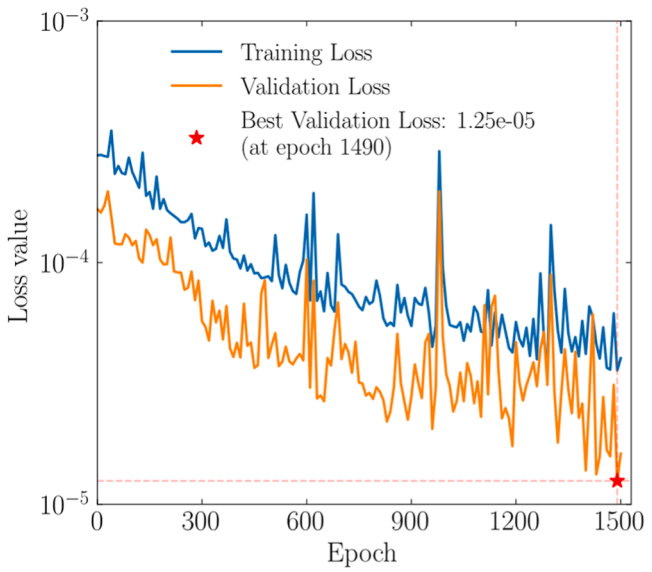


**Fig. 5.** The evolution of the training and validation loss values over 1500 epochs in Case 1.

instability that is comment in traditional recurrent neural networks. By embedding the discrete time-stepping scheme directly into the network architecture, the recurrent unit strictly enforces the governing temporal evolution law, ensuring physical consistency without the need for auxiliary loss constraints. Additionally, this integration strategy enables a trade-off between accuracy and computational cost by allowing the incorporation of various explicit time integration schemes, such as the Euler or RK methods. At each time step, spatial interactions are captured on unstructured meshes, enabling the PeRGNN to simulate coupled spatiotemporal dynamics without relying on computationally expensive implicit solvers.

#### 2.2.2. Encoding boundary conditions

Accurate enforcement of boundary conditions fundamentally guarantees the uniqueness and physical consistency of solutions to PDEs. In vanilla PINNs, boundary conditions are typically imposed through soft constraints, where collocation points are sampled along the domain boundaries, and the residuals at these locations are incorporated into the loss function. While this method is simple for implementation, it can not ensure the exact satisfaction of boundary conditions and can lead to slower convergence [58].

To overcome these limitations, we adopt a padding strategy for boundary conditions which has been theoretically justified and widely used in many neural PDE solvers, including PEFEN [33], PECAN [79], PeRCNN [51], PhyGeoNet [80], and PhyMPGN [65]. These studies demonstrate that the padding strategy preserves numerical stability and avoids boundary-induced errors during long-term rollout. Similar strategies have also shown robustness in practical engineering applications [32,49,81,82]. In our framework, Dirichlet boundary conditions are directly encoded in both the physical and latent spaces to ensure strict adherence throughout the solution process.

In the physical space, hard enforcement is achieved by explicitly assigning the known Dirichlet boundary values to the corresponding boundary nodes in the graph. At each time step, the prediction $u^t$ from the PeRGNN model is subjected to a boundary padding operation, which overwrites the values at boundary nodes with the prescribed boundary conditions. This ensures exact constraints on boundary conditions before the prediction is used in the subsequent time step. As for the latent space, a similar padding strategy is also applied to hidden node features, e.g., $\mathbf{h}^l$, after each MPNN layer except the last layer. Specifically, we reassign the latent features at boundary nodes after passing each MPNN layer to fixed values, preserving the influence of Dirichlet boundary conditions throughout the information propagation process.

During training and optimization, boundary node values are overwritten after each forward update and treated as constants. Thus, their gradients are zero and do not participate in backpropagation. Interior node gradients are computed normally via message passing from neighbouring nodes, ensuring smooth gradient propagation within the computational domain.

Additionally, as introduced in the previous section, we incorporate a one-hot encoding vector to indicate the type of each node (e.g., interior, Dirichlet boundary). This encoding serves as an informative input feature, enabling the model to distinguish node roles during both forward propagation and boundary padding. Although not strictly part of the padding mechanism, this auxiliary feature enhances the model's awareness of boundaries and improves learning efficiency.

#### 2.2.3. Network training

We partition the complete time series ($T+1$ steps) into shorter segments of $M$ consecutive steps ($\ll T+1$). This approach reduces the depth of the computational graph at each training step, ensuring stable gradients and preventing issues like gradient explosion. As illustrated in Fig. 3, each segment spans from $u^{((i-1)\cdot M)}$ to $u^{(i\cdot M-1)}$, where $i \in \{1,2,\ldots,\lceil (T+1)/M \rceil\}$. Following the model architecture depicted in Fig. 2, PeRGNN accepts the initial frame $u^{(i\cdot M)}$ with artificially added Gaussian noise (3% for forward analysis in this research) as input and autoregressively predicts the subsequent $M-1$ steps within each segment. To balance accuracy and computational costs, we design a parsimonious loss function, evaluated only at the first and last predicted steps of each segment:

$$L = \frac{1}{N_d}\left( \| u^{((i-1)\cdot M+1)} - \widehat{u}^{((i-1)\cdot M+1)} \|^2 + \| u^{(i\cdot M-1)} - \widehat{u}^{(i\cdot M-1)} \|^2 \right) \tag{10}$$

where $N_d$ denotes the number of nodes in the graph, $u$ and $\widehat{u}$ denotes the prediction from our model and the ground truth, respectively, and $\frac{1}{N_d}\| * \|^2$ aims to calculate the mean square error (MSE) of the predictions. In this way, a highly sparse temporal dataset is used for supervision. Further, we employ segment shuffling during training to break temporal dependencies and improve generalization.

The validation datasets comprise clean data without artificial Gaussian noise contamination in the initial frame. During testing, the model solely receives the input of the initial variable states $u^{(0)}$ and autonomously predicts all subsequent time steps. Model performance is rigorously quantified on testing sets using both MSE and the R-square ($R^2$) value.

### 2.3. Data acquisition and preprocessing

High-fidelity ground truth solutions are generated using the finite element software COMSOL Multiphysics 6.2 with fine spatiotemporal discretization and then downsampled to a lower resolution. The details of spatiotemporal discretization parameters for different tunnel configurations (Fig. 1) with different resolutions have been summarised in Table 2. The downsampled spatial mesh discretization for each tunnel configuration is illustrated in Fig. 4. PeRGNN operates entirely on the downsampled coarse mesh, where each graph node corresponds directly to a node in that mesh. As a result, the model's inputs and predictions share the same resolution and are evaluated against downsampled ground truth. Super-resolution inference is not considered in this study. Notably, compared to existing physics-informed learning approaches for solving 2D consolidation problems, which require dense training data (as reviewed in the introduction), this study operated with reduced training points by several orders of magnitude.

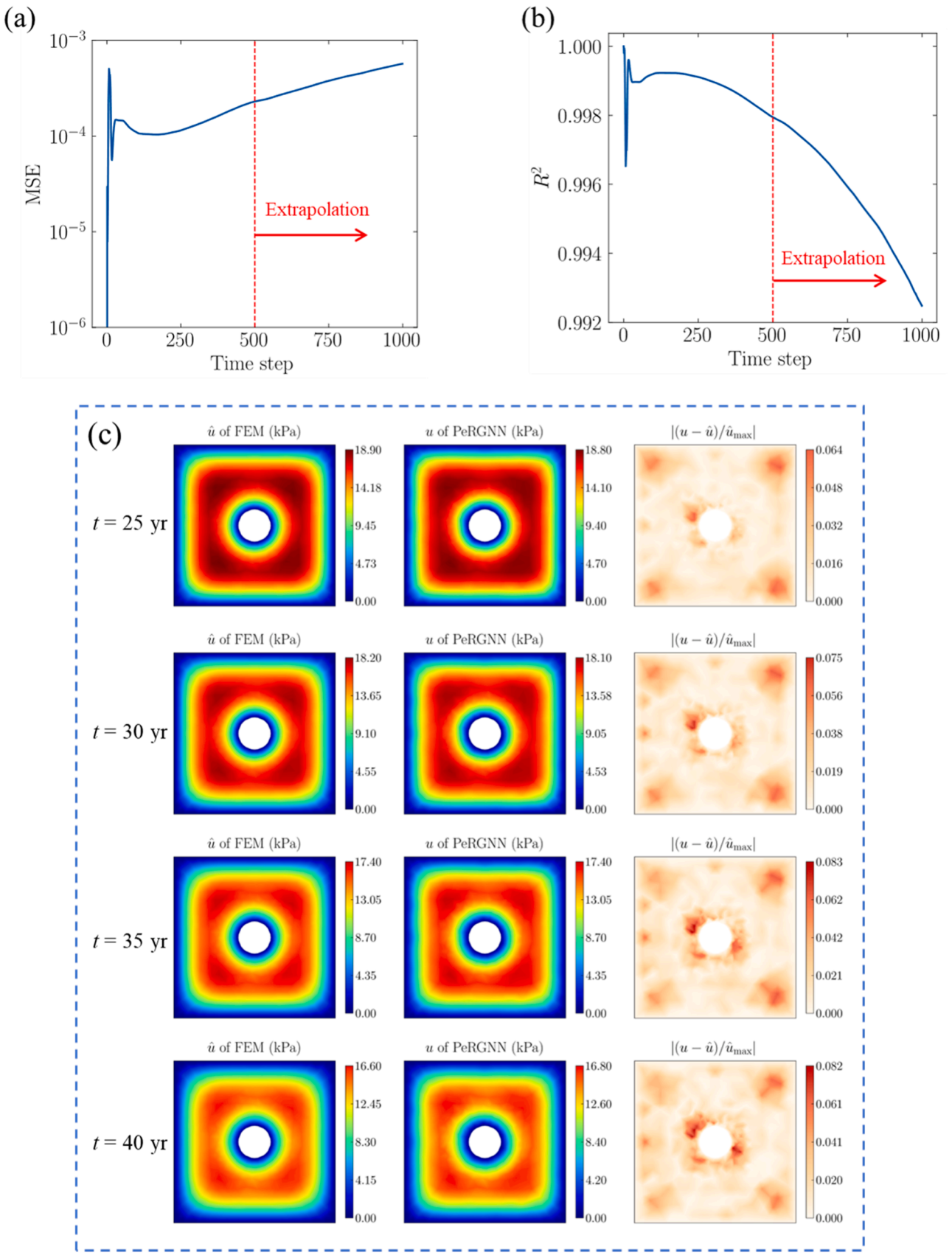


**Fig. 6.** The predictive performance of the PeRGNN model in Case 1. (a-b) The MSE and $R^2$ values of the pore water pressure predicted by the PeRGNN model trained on the 500 time steps but tested on the 1000 time steps; (c) Comparison of pore water pressure distributions (FEM ground truth and PeRGNN predictions) and point-wise relative errors at $t$ = 25, 30, 35, and 40 years.

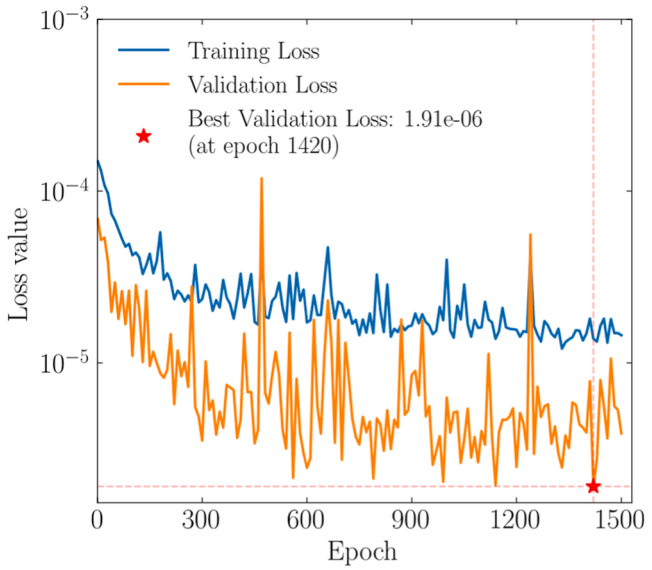


**Fig. 7.** The evolution of the training and validation loss values over 1500 epochs in Case 2.

## 3. Numerical experiments

This section systematically evaluates the performance of the proposed PeRGNN model through six numerical case studies, examining its temporal extrapolation capability for long-term predictions, its generalization performance across varying consolidation parameters and initial conditions, and its inverse analysis capability. Finally, the model's engineering applicability is demonstrated through anisotropic consolidation predictions for tunnels with complex geometries. All cases are considered over a 40-year consolidation period.

### 3.1. Temporal extrapolation in long-term consolidation evolution

The first study case (Case 1) introduced in this section evaluates the PeRGNN model's temporal extrapolation performance. The isotropic consolidation analysis around a single tunnel (Fig. 1a) is selected as the benchmark problem for this case. The soil's coefficient of consolidation $C_v$ is selected to be 0.4 $m^2$/year, and the instantaneous surface load $p_0$ is set as 20 kPa. All details of configurations for the PeRGNN model for Case 1 are shown in Table 3. The time segmentation size for each case is selected to balance predictive stability and computational cost. Very fine segmentation degrades long-term stability due to error accumulation, whereas excessively coarse segmentation offers limited accuracy improvements at significantly higher computational costs. The ELU was selected as the activation function for all cases. While the PeRGNN framework is robust across various activations (including Sin , Tanh, and Swish), ELU offered superior stability for the diffusive dynamics of the consolidation benchmarks (see Table 4). This choice is consistent with the need for compatibility between neural configurations and specific PDE characteristics [83,84]. Given the versatile nature of the PeRGNN framework, alternative activation functions can be seamlessly integrated to suit different spatiotemporal dynamics in future applications.

The model was trained on 500 time steps for up to 1500 epochs, with early stopping implemented by selecting the checkpoint with the minimal validation loss. The evolution of training and validation loss values over 1500 epochs is shown in Fig. 5. Both losses exhibit a general decreasing trend, with the validation loss reaching its minimum value ($1.25 \times 10^{-5}$) at epoch 1490, suggesting that the model achieves its optimal generalization performance.

For the extrapolation test, PeRGNN is given only the initial state $u^{(0)}$ as input and then tasked with predicting the system's spatiotemporal evolution over the next 999 time steps without resorting to intermediate ground-truth data. The well-trained PeRGNN successfully predicts the consolidation process for the subsequent future time steps (40-year simulation), maintaining MSE $< 5 \times 10^{-4}$ and $R^2 > 0.99$, as shown in Fig. 6a. It can be observed that the prediction accuracy of excess pore water pressure remains high in the whole time series. Even beyond the training period (unseen time steps 501–1000), model performance remains strong with minimal error accumulation.

Fig. 6b compares the predicted pore water pressure distributions with ground truth solutions at four representative time steps ($t = 25$, 30, 35, and 40 years), accompanied by their corresponding point-wise relative error distribution. The results demonstrate PeRGNN's remarkable long-term prediction capability, maintaining consistently small errors even at extrapolated time steps beyond the training period.

### 3.2. Generalization of consolidation parameters

This section introduces Case 2 to demonstrate the PeRGNN's generalization ability on the coefficient of consolidation $C_v$. The benchmark problem for Case 2 is depicted in Fig. 1a, and the detailed model configurations of PeRGNN are provided in Table 3. The initial surface load $p_0$ is set as 20 kPa for this case. We generate 3 trajectories with $C_v = 0.5$, 0.7, and 0.9 $m^2$/year for training, and 7 trajectories with $C_v = 0.1$, 0.2, 0.3, 0.4, 0.6, 0.8, and 1.0 $m^2$/year for testing.

The evolution of training and validation loss values over 1500 epochs has been shown in Fig. 7. Both loss values generally decrease, with the validation loss achieving its minimum value of $1.91 \times 10^{-6}$ at epoch 1420. The MSE and $R^2$ values for each test dataset at every time step are depicted in Fig. 8a. As shown, PeRGNN effectively infers the consolidation process for unseen $C_v$ values despite being trained on data generated from only three $C_v$ trajectories. All testing sets exhibit low MSE ($< 8 \times 10^{-4}$) and high $R^2$ values ($> 0.996$), with minimal error accumulation over time. Fig. 8b compares the predicted pore water pressure distributions with ground truth solutions at the last time step (the $40^{th}$ year) for testing sets with $C_v = 0.1$, 0.3, 0.6, and 1.0 $m^2$/year, along with corresponding point-wise relative errors. As shown, the predicted pore water pressure exhibits excellent agreement with the ground truth. While predicted errors gradually increase for larger $C_v$, reflecting increased nonlinearity at higher diffusion rates, they remain within acceptable limits. In this case study, the model demonstrates strong generalization by accurately predicting the consolidation behaviour across seven different $C_v$ values.

### 3.3. Generalization of initial conditions

This section introduces Case 3 to demonstrate the PeRGNN's generalization ability over different initial conditions. The benchmark problem for Case 3 is depicted in Fig. 1a, and the detailed model configurations of PeRGNN are provided in Table 3. The coefficient of consolidation parameter is set as 0.5 $m^2$/year in Case 3. We train PeRGNN with three different initial surface loads $p_0 = 5$, 25, and 45 kPa, and then test the model's generalizability over six different initial surface loads $p_0 = 10$, 15, 20, 30, 35, and 40 kPa.

The evolution of training and validation losses over 2000 epochs has been shown in Fig. 9. Both losses exhibit a decreasing trend during training, with the validation loss achieving its minimum value of $8.0 \times 10^{-5}$ at epoch 1660. The MSE and $R^2$ values for each test dataset at every time step are depicted in Fig. 10a. As shown, PeRGNN effectively infers the consolidation process for unseen initial conditions. All testing sets exhibit low MSE ($< 9 \times 10^{-4}$) and high $R^2$ values ($> 0.985$). Fig. 10b compares the predicted pore water pressure distributions with ground truth solutions at the last time step (the 40th year) for testing sets with $p_0 = 10$, 20, 30, 40 kPa, along with corresponding point-wise relative errors. The results demonstrate that PeRGNN achieves excellent agreement with the ground truth, even under substantially different initial states. This highlights the model's strong generalization capability to

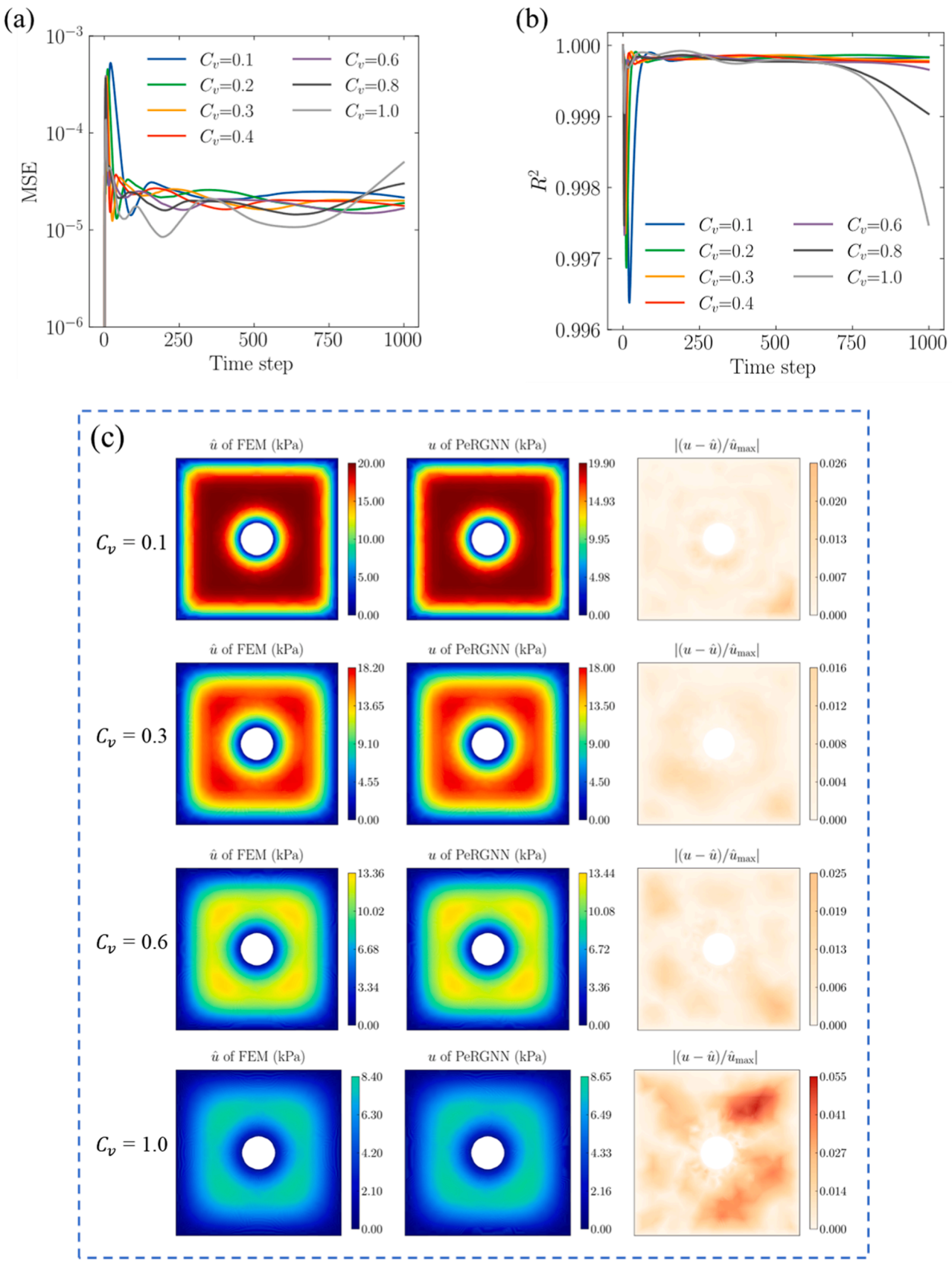


**Fig. 8.** The predictive performance of the PeRGNN model in Case 2. (a-b) The MSE and $R^2$ values of the pore water pressure predicted by the PeRGNN model at every time step trained on 3 $C_v$ values but tested on 7 unseen $C_v$ values; (c) Comparison of pore water pressure distributions (FEM ground truth and PeRGNN predictions) and point-wise relative errors at $t = 40$ years with $C_v = 0.1$, 0.3, 0.6, and 1.0 $m^2$/year.

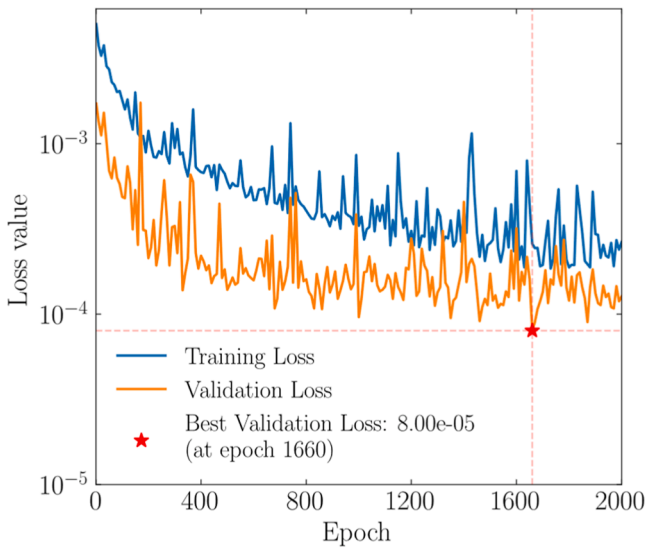


**Fig. 9.** The evolution of the training and validation loss values over 2000 epochs for Case 3.

unseen initial conditions.

### 3.4. Inverse analysis of consolidation parameters from sparse, noisy data

While the previous study cases all focused on forward analysis of consolidation problems, real-world engineering scenarios often require estimating unknown soil parameters. Considering this challenge, as well as the capability of the proposed PeRGNN model to handle both forward and inverse problems in a unified framework, Case 5 introduced in this section investigates PeRGNN's performance in inverse analysis for identifying unknown consolidation parameters under sparse and noisy data. Unlike vanilla PINNs, PeRGNN identifies $C_v$ solely through the observational loss (Eq. (10)) rather than a PDE residual term, as the governing PDE structure is already embedded within the architecture. Thus, $C_v$ is integrated into the trainable parameter set $\boldsymbol{\theta}$ in Eq. (8). The operator $G(u^t; \boldsymbol{\theta})$ necessarily incorporates $C_v$ to transform spatial pressure gradients into temporal increments, thereby ensuring dimensional consistency between $G(u^t; \boldsymbol{\theta})$ and $\partial u/\partial t$. Additional unknown parameters, if present, can likewise be incorporated into $\boldsymbol{\theta}$ and identified during the optimization process.

The isotropic consolidation around the single tunnel is selected as the benchmark problem for Case 5 (Fig. 1a). The initial surface load $p_0$ is set as 20 kPa, and the $C_v$ value is 0.4 $m^2$/year, which needs to be identified by PeRGNN. Artificial Gaussian noise with varying levels, i.e., 3%, 5%, 7%, 9%, and 11 %, is added to the first frame of each short time segment (Fig. 3) during training. The evolution of the validation loss and the learning process of unknown $C_v$ values over 1000 epochs are shown in Fig. 11a and b, respectively. Notably, in Fig. 11a, validation losses for higher noise levels converge to values near the initial losses. This is because the validation process utilizes noisy initial inputs while comparing predictions against clean ground truth. The resulting MSE is therefore dominated by the irreducible discrepancy between the noisy start point and the clean target trajectory. Despite this high MSE, the model effectively captures the underlying physical dynamics, as evidenced by the high accuracy of the identified $C_v$ values.

$C_v$ values are identified at the epoch of minimal validation loss, with exact values and corresponding relative errors summarized in Table 5. The results demonstrate that the identification accuracy of $C_v$ values remains consistently high across different noise levels, with relative errors remaining below $1.33 \times 10^{-3}$ even with 11% noise magnitude, showcasing the robustness of the proposed model. It is also noteworthy that the identification of $C_v$ achieves efficient convergence, approaching the ground truth value within approximately 200 epochs. This indicates that the unknown parameter identification converges faster than the full solution field, a phenomenon consistent with inverse analysis findings in physics-informed learning reported in previous work [21].

The inverse PeRGNN's prediction accuracy of pore water pressure across the whole time series is evaluated based on MSE and $R^2$ values, as illustrated in Fig. 11c and d As shown, the mean value of MSE increases with rising noise levels, indicating a gradual decline in prediction precision of PeRGNN. However, the mean value of MSE is still below 0.003 when the noise level reaches 11%. Similarly, the mean $R^2$ values decrease as noise increases, but remain above 0.96 even at the highest noise level (11%). Besides, the standard deviations reflect the variability in performance across time steps, which becomes more prominent at higher noise levels but is still in an acceptable range. These findings demonstrate the PeRGNN's robustness for inverse analysis with noisy and sparse data, revealing the model's potential for application in practical geoengineering problems.

To further demonstrate PeRGNN's efficiency and robustness in parameter inversion for long-term engineering problems, a vanilla PINN model was trained as a baseline. This PINN employs a 3D input ($x$, $y$, $t$), a single output ($u$), and four hidden layers with 64 neurons for each. All other hyperparameters are kept consistent with PeRGNN. Its loss function integrates data and physics components:

$$L = \lambda_1 L_{data} + \lambda_2 L_{PDE} \tag{11}$$

where $\lambda_1$ and $\lambda_2$ are weights for the data-informed loss $L_{data}$ and physics-informed loss $L_{PDE}$, respectively. In this study, we set$\lambda_1 = 0.01$, $\lambda_2 = 1$. The training results are presented in Fig. 12. The vanilla PINN was trained for 40,000 epochs until convergence over a duration of 16.8 min , which is approximately 70% longer than 9.8 min required by PeRGNN (see Table 7). Although the loss and predicted $C_v$ values converged, the identified $C_v$ is 0.31, representing a significant prediction error of 22.5%. Furthermore, the MSE and $R^2$ for the predicted pore water pressure are 0.217 and 0.955, respectively. In contrast, the prediction accuracy of PeRGNN remains robust even with highly noisy data (Fig. 11).

The vanilla PINN's poor performance in this inverse analysis can be attributed to three aspects: (1) the "curse of dimensionality" regarding collocation points in the whole spatiotemporal domains, (2) increased complexity from the irregular computational domain, and (3) the difficulty of fine-tuning weighting coefficients $\lambda_1$ and $\lambda_2$ to balance competing loss terms. While advanced PINN variants have been proposed to address these issues [74,85], they are beyond this study's scope.

### 3.5. Anisotropic consolidation around circular twin tunnels leveraging different decoding strategies

While Cases 1–4 address isotropic consolidation around a single tunnel, Case 5, introduced in this section, is designed to evaluate PeRGNN's performance in anisotropic consolidation scenarios involving twin tunnels, demonstrating its generalizability across anisotropic $C_v$ values. The initial surface load $p_0$ is set as 20 kPa for this case. The benchmark problem for Case 5 is depicted in Fig. 1b, and the detailed model configurations of PeRGNN are provided in Table 3. We generate 6 trajectories with ($C_v$, $C_h$) = (0.1, 0.5), (0.1, 1.0), (0.5, 1.0), (0.5, 0.1), (1.0, 0.1), (1.0, 0.5) as training datasets and 12 trajectories with ($C_v$, $C_h$) = (0.2, 0.4), (0.2, 0.9), (0.3, 0.6), (0.3, 0.8), (0.3, 0.9), (0.4, 0.2), (0.4, 0.7), (0.6, 0.3), (0.7, 0.4), (0.8, 0.3), (0.9, 0.2), (0.9, 0.3) as testing datasets. The distribution of ($C_v$, $C_h$) values for training and testing is shown in Fig. 13.

In contrast to isotropic cases, where the decoder directly outputs the operator $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$, Case 5 uses a separate decoder for $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$ to account for anisotropy. We evaluate five different decoding configurations: a shared decoder for both operators and twin decoders, each

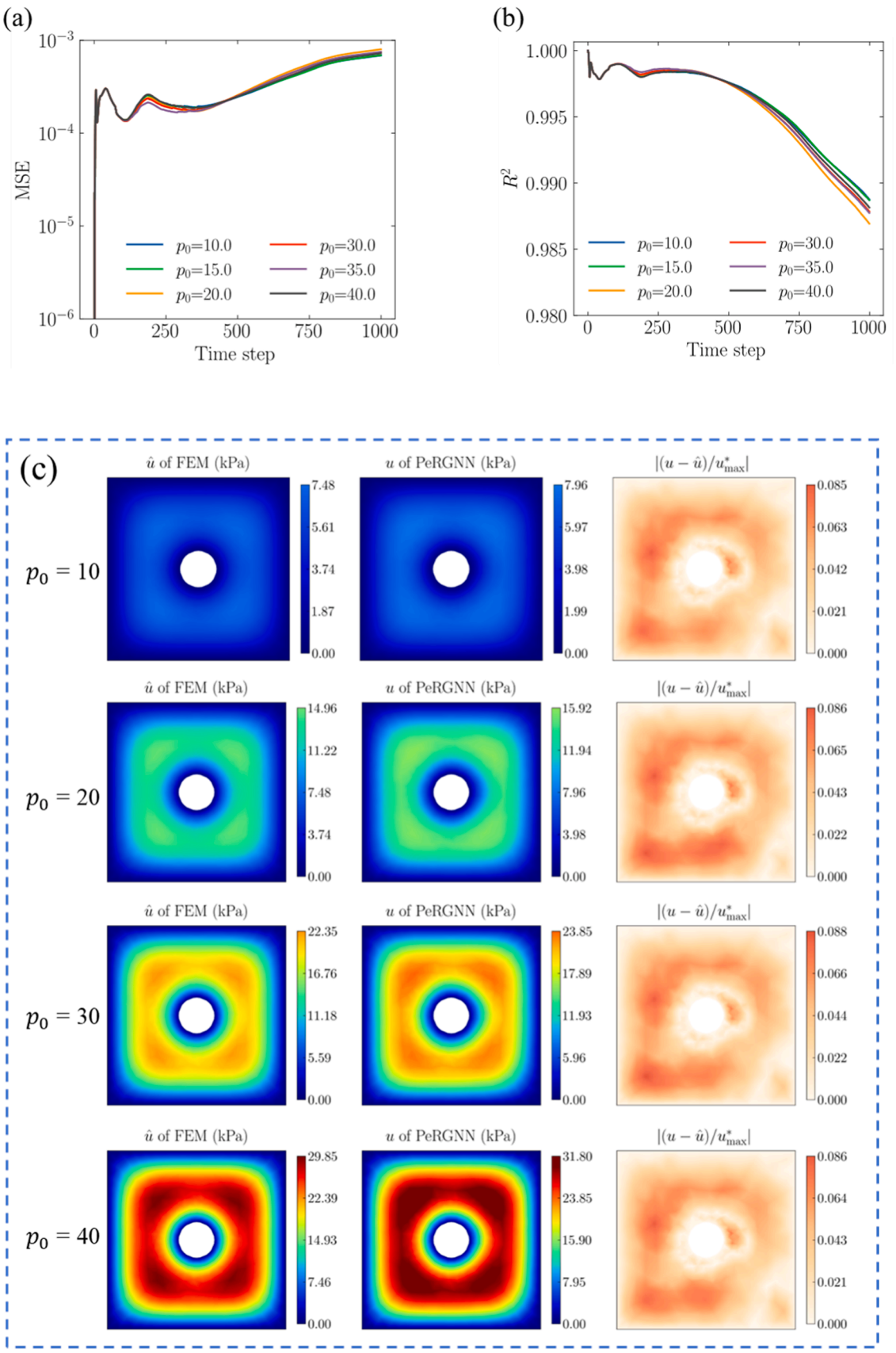


**Fig. 10.** The predictive performance of the PeRGNN model in Case 3. (a-b) The MSE and $R^2$ values of the pore water pressure predicted by the PeRGNN model at every time step trained on 3 initial conditions but tested on 6 unseen initial conditions; (c) Comparison of pore water pressure distributions (FEM ground truth and PeRGNN predictions) and point-wise relative errors at $t = 40$ years with initial surface loading $p_0 = 10$, 20, 30, 40 kPa.

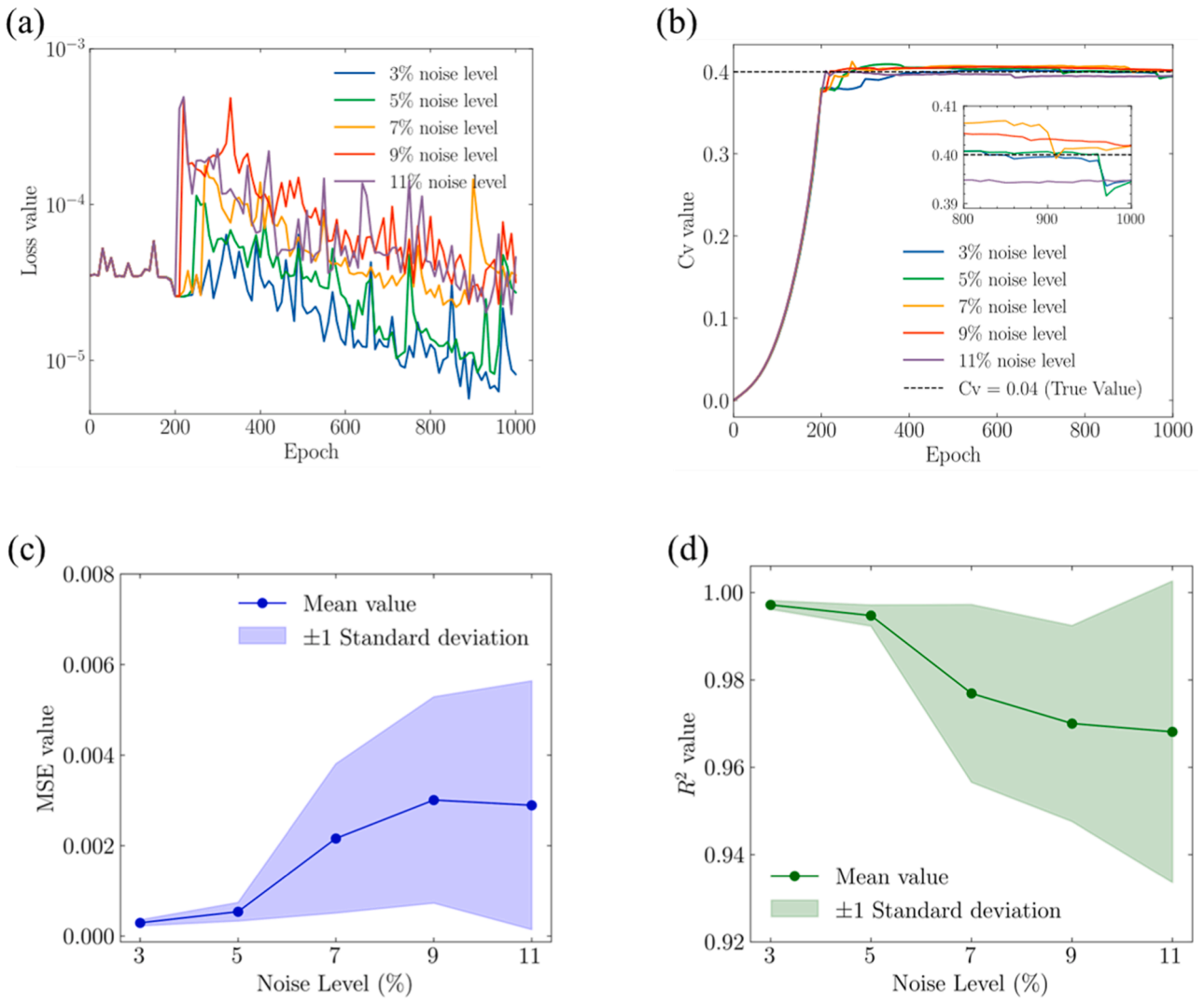


**Fig. 11.** Inverse analysis of the PeRGNN model in Case 4. (a) The evolution of validation loss values over 1000 epochs; (b) The learning process of unknown $C_v$ value over 1000 epochs; (c-d) The mean values and standard deviations of the MSE and $R^2$ for pore water pressure predictions across the entire time series.

**Table 5**
The identified $C_v$ value and the corresponding relative errors under different levels of Gaussian noise in the training data.

| Noise level | 3% | 5% | 7% | 9% | 11% |
|---|---|---|---|---|---|
| Identified $C_v$ value | 0.3995 | 0.4000 | 0.4030 | 0.4032 | 0.3947 |
| Relative errors | 1.25e-4 | 9.50e-5 | 7.50e-4 | 8.00e-4 | 1.33e-3 |

outputting one operator. The twin-decoder configurations explored both identical (e.g., ELU-ELU, Tanh-Tanh) and mixed activation functions (e.g., Tanh-Sigmoid, ELU-Tanh).

The rationale for using mixed activation is grounded in existing research. For instance, Qian et al. [86] demonstrated that mixed activation functions improve the model's ability to capture nonlinearity. Jagtap and Karniadakis [45] utilized different activation functions for subdomains in PINNs, and Sarma et al. [87] optimized local solution behaviours by assigning subdomain-specific activations. Although these strategies have been explored for single network or domain decomposition in PINNs, applying mixed activations to independent operator decoders is still novel. Furthermore, as the two decoders both receive the input information from the same encoder processor and MPNN layers, assigning different activation functions prevents the two decoders from collapsing into similar representations, allowing for more flexible and accurate modelling of anisotropic behaviours.

The performance of PeRGNN with different decoding configurations is evaluated based on the predicted pore water pressure at $t = 4$, 20, and 40 years (Fig. 14). Table 6 presents MSE and $R^2$ at $t = 40$ years for each decoding configuration. The results show that the single decoder approach exhibits significantly higher MSE values and lower $R^2$ values, indicating its poor performance. Among the twin-decoder configurations, the models employing identical activation functions (ELU-ELU and Tanh-Tanh) achieve competitive accuracy, but the twin decoder with mixed activation functions (Tanh-Sigmoid and ELU-Tanh) further improves prediction accuracy, highlighting its superior flexibility in handling anisotropic behaviours. These findings demonstrate the effectiveness of using independent decoders with mixed activation functions for modelling anisotropic systems.

### 3.6. *Application to anisotropic consolidation around straight-wall arched twin tunnels*

Case 6 introduced in this section aims to further extend the application of PeRGNN with twin decoders to more complex anisotropic tunnel consolidation scenarios. In this case, a more challenging tunnel configuration is considered, i.e., vertically offset straight-wall arched twin tunnels, as depicted in Fig. 1c. Building on the findings from Case 5,

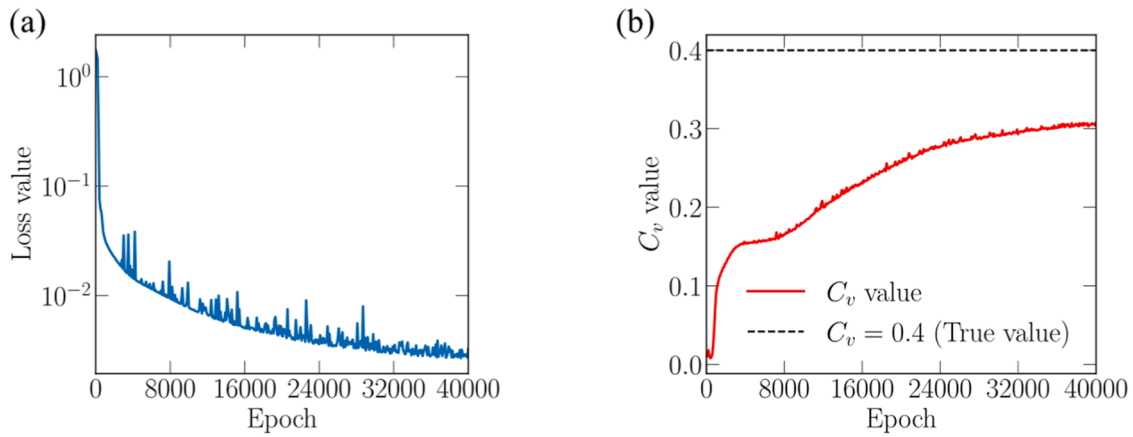


**Fig. 12.** Results of solving inverse consolidation problems using the vanilla PINN model. (a) Evolution of loss values over 40,000 epochs; (b) Learning process of $C_v$ over 40,000 epochs.

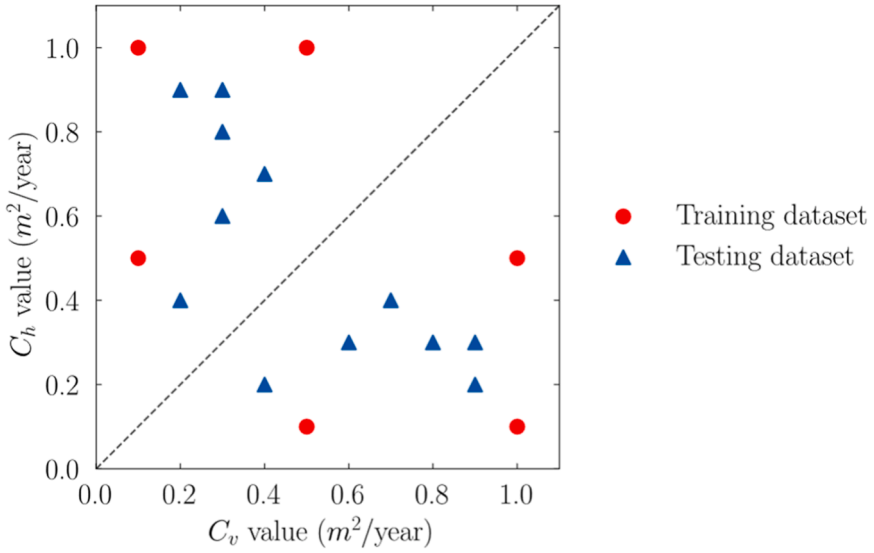


**Fig. 13.** The distribution of ($C_v$, $C_h$) values for the training and testing datasets.

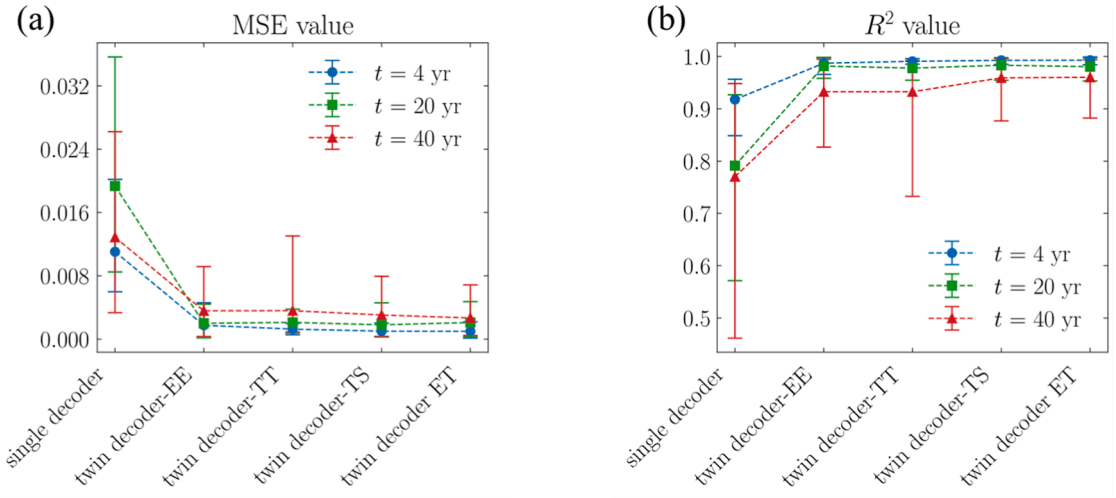


**Fig. 14.** The performance comparison among five different decoding configurations for PeRGNN. (a) The MSE values of predicted pore water pressure at $t$ = 4, 20, and 40 years; (b) The $R^2$ values of predicted pore water pressure at $t$ = 4, 20, and 40 years.

the twin decoder architecture with ELU and Tanh activation functions (twin decoder-ET) is identified as the most accurate strategy and

**Table 6**
MSE and $R^2$ values of predicted pore water pressure at $t = 40$ years using different decoding strategies.

| Decoder configuration | Activation function | MSE | $R^2$ |
|---|---|---|---|
| Single decoder | ELU | 1.29e-2 | 0.770 |
| Twin decoder (EE) | ELU - ELU | 3.57e-3 | 0.933 |
| Twin decoder (TT) | Tanh - Tanh | 3.59e-3 | 0.933 |
| Twin decoder (TS) | Tanh - Sigmoid | 3.05e-3 | 0.959 |
| Twin decoder (ET) | ELU - Tanh | 2.67e-3 | 0.960 |

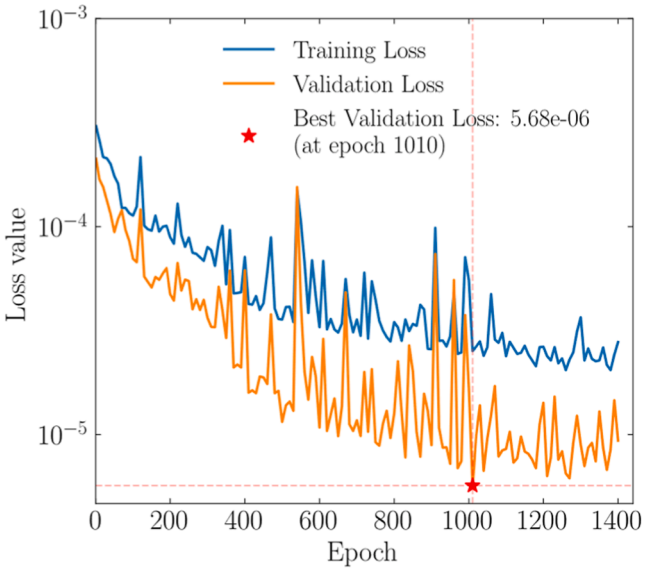


**Fig. 15.** The evolution of training and validation loss values over 1400 epochs in Case 6.

therefore is employed here. The initial surface load $p_0$ is set to 20 kPa, consistent with previous cases. The training and testing datasets used in this case are the same as those in Case 5, comprising 6 training sets and 12 testing sets, as shown in Fig. 13.

The evolution of training and validation losses over 1400 epochs is shown in Fig. 15. Both losses exhibit a generally decreasing trend, indicating effective convergence of the model. The best validation loss of $5.68 \times 10^{-6}$ is achieved at epoch 1010, as highlighted by the red star marker. After this point, no significant overfitting is observed, suggesting that the model maintains strong generalization capability throughout training.

To evaluate the model's performance across 12 testing scenarios, Fig. 16a and b illustrates the MSE and $R^2$ values of predicted excess pore water pressure at $t = 10$, 20, 30, and 40 years for each ($C_v$, $C_h$) parameter combination. While MSE values show a gradual increase over time (from $10^{-5}$ to $10^{-3}$ order of magnitude) and $R^2$ values exhibit a slight decrease (maintaining >0.95 in all cases), both metrics remain within acceptable thresholds throughout the 40-year simulation period. This demonstrates the model's stable generalization and long-term predictive capability across varying soil parameter combinations.

To further visualize the generalization and long-term predictive ability of PeRGNN on challenging vertically offset straigh-wall arched twin tunnels, Fig. 16c presents the spatiotemporal evolution of ground truth of excess pore water pressure, predictions, and the corresponding point-wise relative errors at time snapshots of 10, 20, 30, and 40 years, under the representative test case with consolidation coefficients ($C_v$, $C_h$) = (0.3, 0.6). The predictions made by PeRGNN show excellent agreement with the reference solution across all time steps. Although the point-wise relative error increases slightly over time, it remains within acceptable bounds for engineering practice. These results demonstrate the model's strong ability to generalize to previously unseen parameter regimes in complex, real-world tunnel configurations.

## 4. Discussions

The preceding numerical experiments have validated the effectiveness of PeRGNN in addressing forward and inverse problems in various tunnel consolidation scenarios. This section evaluates the computational performance of the PeRGNN framework relative to traditional numerical solvers and assesses its robustness under varying degrees of data sparsity.

### 4.1. Computational comparison with a traditional FEM solver (Comsol)

The computational costs and hardware specifications for the traditional FEM solver (COMSOL) and PeRGNN are summarized in Table 7. For each case, COMSOL simulations were performed on an Intel Core i7–13700KF CPU, and PeRGNN modelling was conducted on an Nvidia RTX 4090. While PeRGNN does not offer an advantage in training time, it exhibits superior efficiency in offline inference, with an average forward simulation of 4.56 s. Given its slight advantage over FEM for the simple 2D diffusion physics, PeRGNN's primary benefits lie in parallel computation and inverse analysis. Leveraging GPU-accelerated tensor operations, the framework enables simultaneous batch processing of massive parameter trajectories. This represents a significant advancement over traditional solvers that typically scale linearly with the number of trajectories. Furthermore, PeRGNN provides a unified and robust framework for inverse analysis that circumvents the computational burden and potential instabilities of iterative optimization loops required by traditional solver-based inverse methods.

### 4.2. Algorithmic data efficiency

To evaluate the framework's robustness under extreme data limitations, we investigated the PeRGNN's performance in Case 1 using a series of decimated spatial meshes containing 225, 153, 72, and 45 nodes. As illustrated in Fig. 17, the model maintains stable predictive accuracy even as the spatial resolution is reduced by nearly two orders of magnitude from the standard high-resolution mesh (3478 nodes). This data efficiency stems from the hard-encoding architecture of PeRGNN. By strictly enforcing the temporal evolution laws while learning spatial operators through graph-based message passing, PeRGNN significantly reduces the volume of observational data required for the model to adhere to the governing physics.

## 5. Conclusions

This study presents a novel physics-encoded recurrent graph neural network (PeRGNN) learning paradigm designed for the spatiotemporal analysis of 2D consolidation around complex tunnel geometries, considering both isotropic and anisotropic consolidation behaviours. The proposed framework addresses both forward and inverse problems in a unified manner by integrating graph-based message-passing spatial representations with a recurrent time-marching structure. The core strength of PeRGNN lies in its hard-encoded architecture, which explicitly embeds the initial and boundary conditions as well as discrete temporal evolution laws. This design overcomes limitations of traditional physics-informed learning methods in irregular domains and long-term prediction tasks, while ensuring physical consistency throughout the simulation timeline, and therefore providing a robust computational surrogate for complex geomechanical processes.

The numerical case studies involving irregular tunnel geometries and anisotropic material behaviours demonstrate that PeRGNN is highly effective for both forward prediction, extrapolation, generalization, and inverse analysis. The framework exhibits the capability to extrapolate accurately over long-term timelines and generalize to unseen material

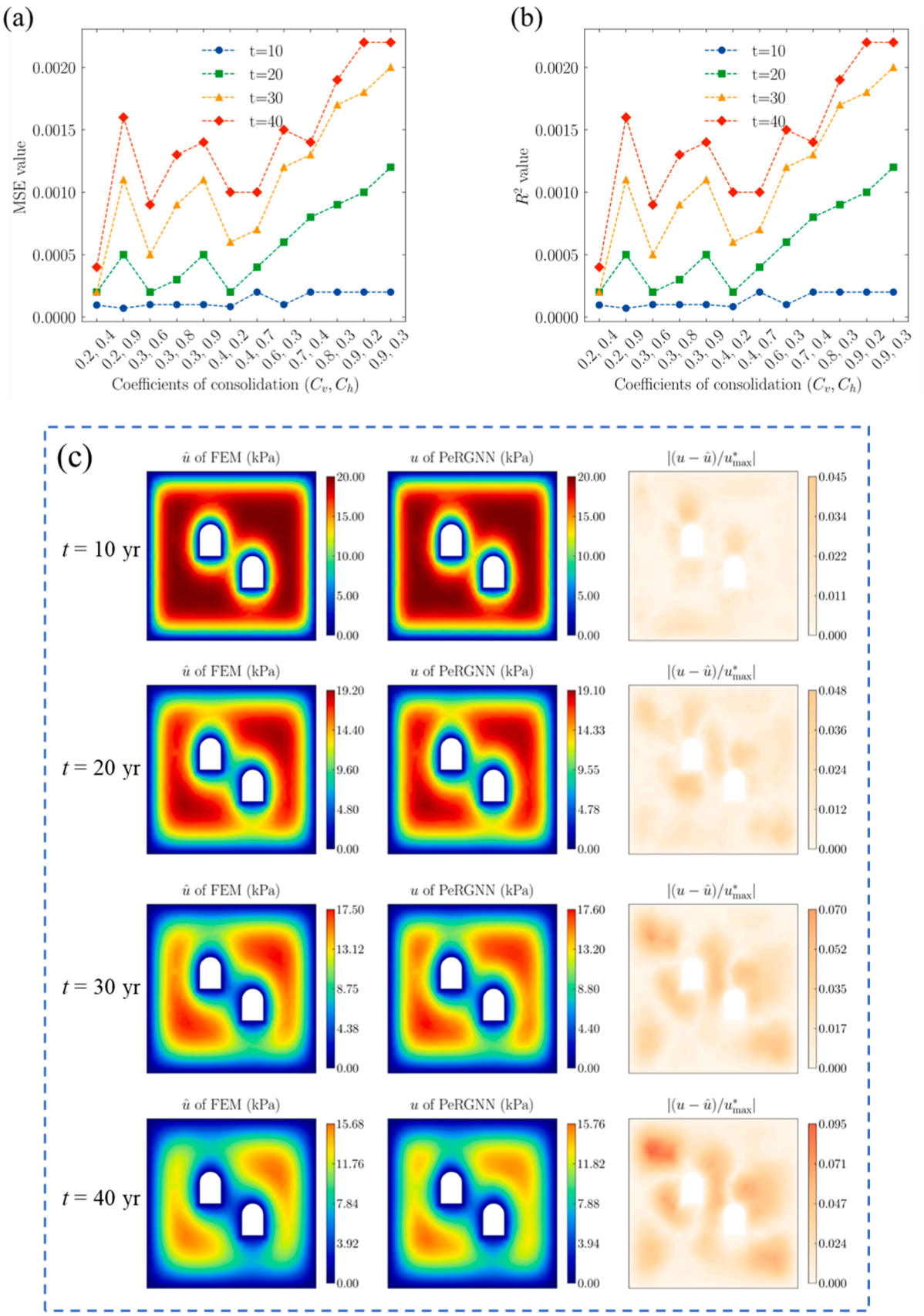


**Fig. 16.** The predictive performance of the PeRGNN model in Case 6. (a-b) The MSE and $R^2$ values at four time snapshots ($t$ = 10, 20, 30, and 40 years) across varying coefficients of consolidation ($C_v$, $C_h$); (c) Comparison of pore water pressure distributions (FEM ground truth and PeRGNN predictions) and point-wise relative errors at $t$ = (a) 10, 20, 30, and 40 years.

**Table 7**
Computational costs and hardware specifications for COMSOL and PeRGNN.

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 |
|---|---|---|---|---|---|---|
| **COMSOL Device** | 13th Gen Intel(R) Core(TM) i7–13700KF | | | | | |
| Computational time per trajectory (s) | 6.5 | 6.5 | 6.5 | / | 7.0 | 7.3 |
| Computational time on all trajectories (s) | / | 45.5 | 39.0 | / | 84.0 | 87.6 |
| **PeRGNN Device** | Nvidia RTX 4090 | | | | | |
| Training time (min) | 9.8 | 54.5 | 71.4 | 4.5 | 102.6 | 103.2 |
| Inference time per trajectory (s) | 4.42 | 4.57 | 4.46 | / | 4.64 | 4.73 |
| Inference time on all trajectories (s) | / | 4.62 | 4.50 | / | 4.69 | 4.81 |

parameters and loading conditions without the need for re-training. Furthermore, the unified inverse analysis framework established in this study allows for the identification of soil consolidation parameters from sparse data containing up to 11% noise, effectively bypassing the trial-and-error fashion of traditional solver-based inverse methods.

A significant finding of this research is the framework's exceptional data efficiency. By leveraging its hard-encoded constraints, the PeRGNN maintains stable predictive performance even when the spatial sampling density is reduced by approximately two orders of magnitude compared to the dense datasets typically required by existing physics-informed methods. This makes the model particularly suitable for practical engineering scenarios where high-resolution monitoring data is limited.

While this study focuses on 2D homogeneous consolidation benchmarks, the underlying graph-recurrent architecture is inherently versatile and capable of solving other complex spatiotemporal PDEs within a unified framework. Future work will focus on extending PeRGNN to more complex spatiotemporal PDEs and 3D heterogeneous scenarios to further enhance its practical utility. To summarize, PeRGNN provides a transformative approach for real-time monitoring and digital twinning in geomechanics by bridging the predictive efficiency of neural networks with the physical rigour of numerical discretization.

## Data availability

The PeRGNN model code and datasets will be publicly available in the following GitHub repository: [https://github.com/XiaoXuan0418/Physic-Encoded-Recurrent-Graph-Neural-Network-PeRGNN] after publishing some ongoing works.

## CRediT authorship contribution statement

**Xiao-Xuan Chen:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Bocheng Zeng:** Visualization, Software, Methodology, Investigation, Conceptualization. **Xi Wang:** Writing – review & editing, Validation, Methodology, Investigation, Conceptualization. **Hao Sun:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Zhen-Yu Yin:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

## 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.

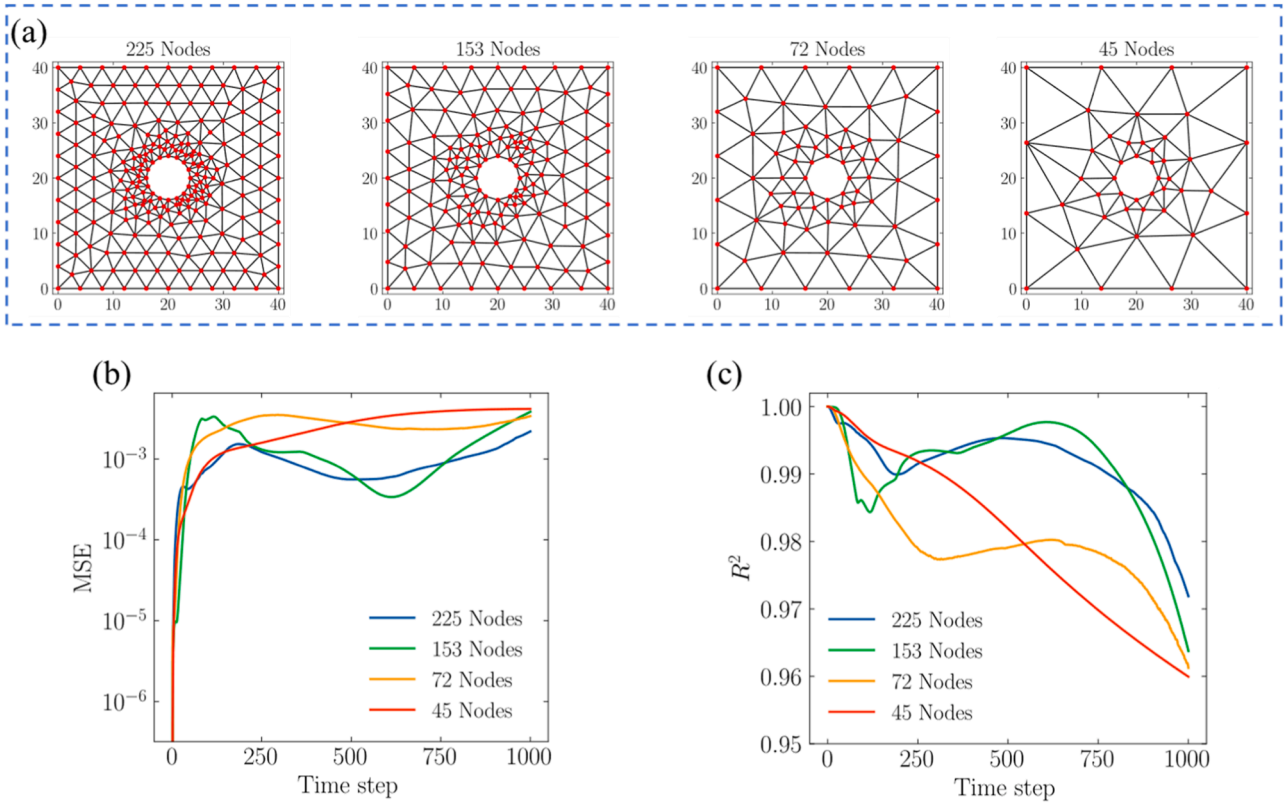


**Fig. 17.** The performance of PeRGNN using an extremely sparse dataset. (a) The distribution of mesh discretization and spatial nodes with 225, 153, 72, and 45 spatial nodes; (b-c) The MSE and $R^2$ values of the pore water pressure predicted by the PeRGNN model trained on the 500 time steps but tested on the 1 000 time steps.

## Acknowledgements

This work was supported by the general research fund of the Research Grants Council (RGC) of the Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No E-PolyU501/24, T22-607/24-N, 15220423, 15227923), the State Key Laboratory of Climate Resilience for Coastal Cities at the Hong Kong Polytechnic University, the National Natural Science Foundation of China (No. 62276269, No 92270118), and the Beijing Natural Science Foundation (No. 1232009).

## References

[1] Terzaghi K. Erdbaumechanik auf bodenphysikalischer Grundlage, F. Deuticke; 1925.
[2] Biot MA. General theory of three-dimensional consolidation. J Appl Phys 1941;12:155–64.
[3] Schiffman RL, Stein JR. One-dimensional consolidation of layered systems. J Soil Mech Found Div 1970;96:1499–504.
[4] Ng A, Small J. Use of coupled finite element analysis in unsaturated soil problems. Int J Numer Anal Methods Geomech 2000;24:73–94.
[5] Lu X, Li C. An analytical solution for nonlinear consolidation of composite foundations improved by stone columns and vertical drains. Comput Geotech 2023;161:105598.
[6] Chen XX, Yang J, He GF, Huang LC. Development of an LSTM-based model for predicting the long-term settlement of land reclamation and a GUI-based tool. Acta Geotech 2023;18:1–14.
[7] He G-F, Yin Z-Y, Zhang P. Uncertainty quantification in data-driven modelling with application to soil properties prediction. Acta Geotech 2025;20:843–59.
[8] Zhang P, Yin ZY, Jin YF. State-of-the-art review of machine learning applications in constitutive modeling of soils. Arch Comput Methods Eng 2021;28:3661–86.
[9] Shi C, Wang Y. Data-driven spatio-temporal analysis of consolidation for rapid reclamation. Geotechnique 2023;74:676–96.
[10] Lo MK, Loh DR, Chian SC, Ku T. Probabilistic prediction of consolidation settlement and pore water pressure using variational autoencoder neural network. J Geotech Geoenviron Eng 2023;149:04022119.
[11] Krishnan M. Against interpretability: a critical examination of the interpretability problem in machine learning. Philos Technol 2020;33:487–502.
[12] Yan W, Yan Y, Shen P, Zhou W-H. A hybrid physical data informed DNN in axial displacement prediction of immersed tunnel joint. Georisk 2023;17:169–80.
[13] Liu Y, Liao S, Yang Y, Zhang B. Data-driven and physics-informed neural network for predicting tunnelling-induced ground deformation with sparse data of field measurement. Tunn Undergr Sp Tech 2024;152:105951.
[14] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3:422–40.
[15] Lu L, Xuhui M, Zhiping M, Karniadakis GE. DeepXDE: A deep learning library for solving differential equations. Siam Rev 2021;63:208–28.
[16] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[17] Urbán JF, Stefanou P, Pons JA. Unveiling the optimization process of physics informed neural networks: how accurate and competitive can PINNs be? J Comput Phys 2025;523:113656.
[18] Toscano JD, Oommen V, Varghese AJ, Zou Z, Ahmadi Daryakenari N, Wu C, Karniadakis GE. From pinns to pikans: recent advances in physics-informed machine learning. Mach Learn Comput Sci Eng 2025;1:1–43.
[19] Haghighat E, Raissi M, Moure A, Gomez H, Juanes R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput Meth Appl Mech Eng 2021;379:113741.
[20] Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE. Physics-informed neural networks (PINNs) for fluid mechanics: A review. Acta Mech Sinica-Prc 2022;37:1727–38.
[21] Chen X-X, Zhang P, Yin Z-Y. A comprehensive investigation of physics-informed learning in forward and inverse analysis of elastic and elastoplastic footing. Comput Geotech 2025;181:107110.
[22] Haghighat E, Amini D, Juanes R. Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training. Comput Meth Appl Mech Eng 2022;397:115141.
[23] Jagtap AD, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems. Comput Meth Appl Mech Eng 2020;365:113028.
[24] Jagtap AD, Mao Z, Adams N, Karniadakis GE. Physics-informed neural networks for inverse problems in supersonic flows. J Comput Phys 2022;466:111402.
[25] Haghighat E, Bekar AC, Madenci E, Juanes R. A nonlocal physics-informed deep learning framework using the peridynamic differential operator. Comput Meth Appl Mech Eng 2021;385:114012.
[26] Wang L, Liu G, Wang G, Zhang K. M-PINN: A mesh-based physics-informed neural network for linear elastic problems in solid mechanics. Int J Numer Meth Eng 2024;125:e7444.
[27] Hu H, Qi L, Chao X. Physics-informed Neural Networks (PINN) for computational solid mechanics: numerical frameworks and applications. Thin-Walled Struct 2024;205:112495.
[28] Roy AM, Guha S, Sundararaghavan V, Arróyave R. Physics-infused deep neural network for solution of non-associative Drucker–Prager elastoplastic constitutive model. J Mech Phys Solids 2024;185:105570.
[29] Sun L, Gao H, Pan S, Wang JX. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput Meth Appl Mech Eng 2020;361:112732.
[30] Zhao C, Zhang F, Lou W, Wang X, Yang J. A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics. Phys. Fluids 2024;36:101301.
[31] Ren X, Hu P, Su H, Zhang F, Yu H. Physics-informed neural networks for transonic flow around a cylinder with high reynolds number. Phys. Fluids 2024;36:036129.
[32] Wang X, Yin Z-Y, Wu W, Zhu H-H. Neural network-augmented differentiable finite element method for boundary value problems. Int J Mech Sci 2025;285:109783.
[33] Wang X, Yin ZY. Interpretable physics-encoded finite element network to handle concentration features and multi-material heterogeneity in hyperelasticity. Comput Meth Appl Mech Eng 2024;431:117268.
[34] Chen X-X, Zhang P, Yu H-S, Yin Z-Y, Sheil B. Parsimonious universal function approximator for elastic and elastoplastic cavity expansion problems. J Geotech Geoenviron Eng 2025;151:04025093.
[35] Haghighat E, Abouali S, Vaziri R. Constitutive model characterization and discovery using physics-informed deep learning. Eng Appl Artif Intell 2023;120:105828.
[36] Moon H, Park D, Cho H, Noh H-K, Lim JH, Ryu S. Physics-informed neural network-based discovery of hyperelastic constitutive models from extremely scarce data. Comput Meth Appl Mech Eng 2025;446:118258.
[37] Zhongbo Y, Hien PL. Pre-trained transformer model as a surrogate in multiscale computational homogenization framework for elastoplastic composite materials subjected to generic loading paths. Comput Meth Appl Mech Eng 2024;421:116745.
[38] Wang J, Ma G, Qu T, Guan S, Zhou W, Chang X. GPM-PeNN: A generalized plasticity model-based data-driven constitutive modeling framework using physics-encoded neural network. Comput Meth Appl Mech Eng 2025;436:117694.
[39] Weng H, Bamer F, Luo C, Markert B, Yuan H. Physics-informed neural network for constitutive modeling of cyclic crystal plasticity considering deformation mechanism. Int J Mech Sci 2025; 302:110491.
[40] Ding Y, Chen S, Li X, Jin L, Luan S, Sun H. Physics-constrained neural networks for half-space seismic wave modeling. Comput Geosci 2023;181:105477.
[41] Amini D, Haghighat E, Juanes R. Physics-informed neural network solution of thermo-hydro-mechanical processes in porous Media. J Eng Mech 2022;148:04022070.
[42] Chen Z, Lai S-K, Yang Z, Ni Y-Q, Yang Z, Cheung KC. AT-PINN-HC: A refined time-sequential method incorporating hard-constraint strategies for predicting structural behavior under dynamic loads. Comput Meth Appl Mech Eng 2025;436:117691.
[43] Wang S, Sankaran S, Perdikaris P. Respecting causality for training physics-informed neural networks. Comput Meth Appl Mech Eng 2024;421:116813.
[44] Khademi A, Salari E, Dufour S. Simulation of 3D turbulent flows using a discretized generative model physics-informed neural networks. Int J Non Linear Mech 2025;170:104988.
[45] Jagtap AD, Karniadakis GE. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Commun Comput Phys 2020;28:2002–41.
[46] Meng X, Li Z, Zhang D, Karniadakis GE. PPINN: parareal physics-informed neural network for time-dependent PDEs. Comput Meth Appl Mech Eng 2020;370:113250.
[47] Li Y, Zhou Z, Ying S. DeLISA: deep learning based iteration scheme approximation for solving PDEs. J Comput Phys 2022;451:110884.
[48] He J, Li X, Zhu H. An adaptive discrete physics-informed neural network method for solving the Cahn-Hilliard equation. Eng Anal Bound Elem 2023;155:826–38.
[49] Ren P, Rao C, Liu Y, Wang J-X, Sun H. PhyCRNet: physics-informed convolutional-recurrent network for solving spatiotemporal PDEs. Comput Meth Appl Mech Eng 2022;389:114399.
[50] Yuan B, Wang H, Heitor A, Chen X. f-PICNN: A physics-informed convolutional neural network for partial differential equations with space-time domain. J Comput Phys 2024;515:113284.
[51] Rao C, Ren P, Wang Q, Buyukozturk O, Sun H, Liu Y. Encoding physics to learn reaction-diffusion processes. Nat Mach Intell 2023;5:765–79.
[52] Wang Q, Ren P, Zhou H, Liu X-Y, Deng Z, Zhang Y, Chengze R, Liu H, Wang Z, Wang J-X. P $^ 2$ C $^ 2$ net: PDE-preserved coarse correction network for efficient prediction of spatiotemporal dynamics. Adv Neural Inf Process Syst 2024;37:68897–925.
[53] Liu X-Y, Zhu M, Lu L, Sun H, Wang J-X. Multi-resolution partial differential equations preserved learning framework for spatiotemporal dynamics. Commun Phys 2024;7:31.
[54] Wang Y, Sun J, Li W, Lu Z, Liu Y. CENN: conservative energy method based on neural networks with subdomains for solving variational problems involving heterogeneous and complex geometries. Comput Meth Appl Mech Eng 2022;400:115491.
[55] Yao H, Gao Y, Liu Y. FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction. Comput Meth Appl Mech Eng 2020;363:112892.
[56] Wu Z, Zhang H, Ye H, Zhang H, Zheng Y, Guo X. PINN enhanced extended multiscale finite element method for fast mechanical analysis of heterogeneous materials. Acta Mech 2024;235:4895–913.

[57] Liu S, Zhongkai H, Ying C, Su H, Zhu J, Cheng Z. A unified hard-constraint framework for solving geometrically complex pdes. Adv Neural Inf Process Syst 2022;35:20287–99.
[58] Sukumar N, Srivastava A. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. Comput Meth Appl Mech Eng 2022;389:114333.
[59] Wang S, Karniadakis GE. GMC-PINNs: A new general Monte Carlo PINNs method for solving fractional partial differential equations on irregular domains. Comput Meth Appl Mech Eng 2024;429:117189.
[60] Jiang L, Wang L, Chu X, Xiao Y, Zhang H. PhyGNNet: solving spatiotemporal PDEs with physics-informed graph neural network. In: Proceedings of the 2023 2nd Asia Conference on Algorithms, Computing and Machine Learning. Shanghai, China: Association for Computing Machinery; 2023. p. 143–7.
[61] Mi Y, Ren P, Xu H, Liu H, Wang Z, Guo Y, Wen J-R, Sun H, Liu Y. Conservation-informed graph learning for spatiotemporal dynamics prediction. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1. New York, USA: Association for Computing Machinery; 2025. p. 1056–67.
[62] Xiang Z, Peng W, Yao W, Liu X, Zhang X. Solving spatiotemporal partial differential equations with physics-informed graph neural network. Appl Soft Comput 2024;155:111437.
[63] Gao H, Zahr MJ, Wang J-X. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems. Comput Meth Appl Mech Eng 2022;390:114502.
[64] T. Li, Y. Zou, S. Zou, X. Chang, L. Zhang, X. Deng, A fully differentiable gnn-based pde solver: with applications to poisson and navier-stokes equations, arXiv preprint arXiv:2405.04466, (2024).
[65] B. Zeng, Q. Wang, M. Yan, Y. Liu, R. Chengze, Y. Zhang, H. Liu, Z. Wang, H. Sun, Phympgn: physics-encoded message passing graph network for spatiotemporal pde systems, arXiv preprint arXiv:2410.01337, (2024).
[66] Bekele YW. Physics-informed deep learning for one-dimensional consolidation. J Rock Mech Geotech 2021;13:420–30.
[67] Zhang S, Lan P, Li H-C, Tong C-X, Sheng D. Physics-informed neural networks for consolidation of soils. Eng Comput (Swansea) 2022;39:2845–65.
[68] Mandl L, Mielke A, Seyedpour SM, Ricken T. Affine transformations accelerate the training of physics-informed neural networks of a one-dimensional consolidation problem. Sci Rep 2023;13:15566.
[69] Zhou H, Wu H, Sheil B, Wang Z. A self-adaptive physics-informed neural networks method for large strain consolidation analysis. Comput Geotech 2025;181:107131.
[70] Lan P, Su J-j, Ma X-y, Zhang S. Application of improved physics-informed neural networks for nonlinear consolidation problems with continuous drainage boundary conditions. Acta Geotech 2024;19:495–508.
[71] Lu Y, Mei G. A deep learning approach for predicting two-dimensional soil consolidation using physics-informed neural networks (PINN). Mathematics 2022;10:2949.
[72] Wang Y, Shi C, Shi J, Lu H. Data-driven forward and inverse analysis of two-dimensional soil consolidation using physics-informed neural network. Acta Geotech 2024;19:8051–69.
[73] Zhang P, Sheil B, Girolami M, Yaji K, Yin Z-Y. A novel consolidation analysis framework: universal function approximators regularized by physical principles. Can Geotech J 2024;62:1–17.
[74] Guo H, Yin Z-Y. A novel physics-informed deep learning strategy with local time-updating discrete scheme for multi-dimensional forward and inverse consolidation problems. Comput Meth Appl Mech Eng 2024;421:116819.
[75] Xie S, Hu A, Xiao Z, Mariani S, Vecchia GDella. PINN-based approach to the consolidation analysis of visco-elastic soft soil around twin tunnels. Tunn Undergr Sp Tech 2024;153:105981.
[76] Wang S, Yu X, Perdikaris P. When and why PINNs fail to train: A neural tangent kernel perspective. J Comput Phys 2022;449:110768.
[77] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J Sci Comput 2021;43:A3055–81.
[78] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
[79] Wang X, Yin Z-Y. Physics-encoded convolutional attention network for forward and inverse analysis of spatial-temporal parabolic dynamics considering discontinuous heterogeneity. Comput Meth Appl Mech Eng 2025;442:118025.
[80] Gao H, Sun L, Wang JX. PhyGeoNet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. J Comput Phys 2021;428:110079.
[81] Zhao X, Gong Z, Zhang Y, Yao W, Chen X. Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. Eng Appl Artif Intell 2023;117:105516.
[82] Ren P, Song J, Rao C, Wang Q, Guo Y, Sun H, Liu Y. Learning spatiotemporal dynamics from sparse data via a high-order physics-encoded network. Comput Phys Commun 2025;312:109582.
[83] K. Leng, J. Thiyagalingam, On the compatibility between neural networks and partial differential equations for physics-informed learning, arXiv preprint arXiv:2212.00270, (2022).
[84] Jagtap AD, Karniadakis GE. How important are activation functions in regression and classification? A survey, performance comparison, and future directions. J Mach Learn Model Comput 2023;4:21–75.
[85] Bai J, Rabczuk T, Gupta A, Alzubaidi L, Gu Y. A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics. Comput Mech 2022;71:543–62.
[86] Qian S, Liu H, Liu C, Wu S, Wong HSan. Adaptive activation functions in convolutional neural networks. Neurocomputing 2018;272:204–12.
[87] Sarma AK, Roy S, Annavarapu C, Roy P, Jagannathan S. Interface PINNs (I-PINNs): A physics-informed neural networks framework for interface problems. Comput Meth Appl Mech Eng 2024;429:117135.