

ELEMENT-WISE PHYSICS-INFORMED NEURAL NETWORK APPROACH FOR 2D LINEAR ELASTICITY PROBLEMS

Si T. Do^{1,2}, Dai D. Mai^{1,*}

¹*Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City, Vietnam*

²*Faculty of Electrical-Mechanical, FPT Polytechnic, FPT University, Ho Chi Minh City, Vietnam*
E-mail: daimd@hcmute.edu.vn

Received: 12 November 2025 / Revised: 6 February 2026 / Accepted: 24 February 2026
Published online: 29 May 2026

Abstract. Physics-Informed Neural Networks (PINNs) have emerged as a promising framework for solving solid mechanics problems. Most existing PINN frameworks approximate continuous displacement fields and are trained using collocation points. However, several challenges remain when addressing problems involving complex geometries, concentrated forces, and material heterogeneity. This study proposes an Element-Wise Physics-Informed Neural Network (EWPINN) framework for two-dimensional linear elasticity, in which the Neural Network (NN) operates directly on the discrete nodal degrees of freedom defined on a predefined finite element mesh. The structural response is obtained by minimizing the total potential energy, evaluated element-wise over a triangular mesh, thereby eliminating the need for collocation points and derivative-based loss terms. As a result, the proposed model requires significantly fewer training data points and exhibits improved training efficiency compared to conventional PINN approaches. By operating on a mesh-based formulation, the EWPINN can effectively handle complex geometries, concentrated forces, and piecewise heterogeneous materials. The effectiveness and robustness of the proposed approach are demonstrated through several numerical examples. The obtained structural responses exhibit excellent agreement with reference FEM solutions. The proposed framework provides a differentiable energy-based alternative that can be readily integrated into optimization pipelines and extended to nonlinear problems.

Keywords: physics informed neural network, deep neural network, total potential energy, plane stress, plane strain.

1. INTRODUCTION

As an emerging research field, scientific machine learning integrates advances in machine learning with scientific computing. Machine learning algorithms are designed to learn the input-output relationships of complex systems from training data and have been widely and successfully applied in complex domains where closed-form solutions are unavailable, such as image recognition, autonomous driving, and natural language processing. In recent years, these techniques have also attracted significant attention within the computational solid and structural mechanics community (Liang et al., 2018; Raissi et al., 2019). The application of neural networks to structural analysis dates back to the early 1990s. For example, Berke and Hajela (1992) and Hajela and Berke (1991) were among the earliest researchers to employ feedforward neural networks as surrogates for structural analysis to predict force-displacement relationships, thereby enabling rapid approximations of Finite Element Method (FEM) results. Subsequently,

Yagawa and Okuda (1996) presented a comprehensive survey of early neural network applications in computational mechanics, highlighting their potential for structural identification, material modeling, optimization, and inverse problems. Ramasamy and Rajasekaran (1996) applied artificial neural networks to optimize the design of industrial roof structures, while Lee et al. (2017) provided a systematic evaluation of Deep Neural Networks (DNNs) for truss structural analysis. To reduce the computational cost, H. T. Mai et al. (2021) employed NNs as surrogates for the FEM in the optimization of truss structures exhibiting geometrically nonlinear behavior. In another study, a convolutional neural network was proposed by Kollmann et al. (2020) to identify optimal metamaterial designs. Similarly, Liang et al. (2018) introduced a deep learning-based surrogate model to rapidly and accurately estimate stress distributions in the aorta. Moreover, DNN-based surrogates were introduced by Feng et al. (2022) for nonlinear homogenization problems. Despite their demonstrated success, the aforementioned machine learning models are purely data-driven. Consequently, reliable predictions generally require large, high-quality training datasets. More importantly, a fundamental limitation of data-driven approaches is that their predictions are not inherently constrained by the underlying physical laws, which may lead to physically inconsistent results when extrapolating beyond the training data.

To overcome the limitations of purely data-driven models, an alternative paradigm based on physical laws and governing equations, also known as physics-informed neural networks, was first introduced by Lagaris et al. (1998) and later popularized by Raissi et al. (2019) for solving Partial Differential Equations (PDEs). The core idea of PINNs is to enforce the governing equations by minimizing PDE residuals evaluated at collocation points. An important advantage of this framework is that the training data require no labeling and can be generated without reliance on numerical simulations. Accordingly, many variants of PINNs have been developed to address a wide range of problems. Based on the mathematical characteristics of the loss function, PINNs can generally be classified into four categories: residual-based, energy-based, variational, and hybrid. Among these, residual-based PINNs have received considerable attention due to their broad applicability. For instance, Sirignano and Spiliopoulos (2018) developed a Deep Galerkin method for solving high-dimensional PDEs, while Bai et al. (2022) introduced a least-squares weighted residual formulation for two- and three-dimensional solid mechanics. Luo et al. (2023) developed a stepwise PINN framework for large-deformation analysis of hypoelastic materials, and Bai et al. (2022) further proposed a physics-informed radial basis network for nonlinear PDEs. In addition, Hau et al. (2022) developed a novel PINN for predicting structural instability. Meanwhile, energy-based PINNs are particularly well suited for problems characterized by an underlying energy functional and have demonstrated strong performance in computational mechanics applications, including linear elasticity (Li et al., 2021; H. T. Mai, Mai, et al., 2024; Wang et al., 2024), nonlinear elasticity (Abueidda et al., 2022; D. D. Mai et al., 2024; H. T. Mai et al., 2022; Nguyen-Thanh et al., 2020), elastodynamics (Bai et al., 2025), heat transfer (Cai et al., 2021), and structural optimization (He et al., 2023; Jeong et al., 2023). The variational PINNs are formulated based on the weak form of the governing equations, where integral residuals are minimized using suitable test functions. Several studies have reported the application of this approach in solving PDEs (Kharazmi et al., 2019), solid mechanics analysis (D. D. Mai et al., 2025; H. T. Mai et al., 2023; Radin et al., 2023), forward and inverse problems (Ghose et al., 2024; H. T. Mai, Lee, et al., 2024), and material modeling (Liu & Wu, 2023). Finally, hybrid PINNs integrate the three preceding variants to leverage the advantages of the individual models (Elfetni & Darvishi Kamachali, 2025; Hou et al., 2023). Despite their computational success, most existing PINN frameworks rely on NNs to approximate continuous displacement fields defined over the spatial domain. As a result, many reported PINN applications in structural analysis focus on problems involving distributed loads rather than concentrated forces. In addition, accurately enforcing boundary conditions and capturing boundary-induced effects remain challenging for

domains with complex or non-smooth boundaries. Furthermore, for piecewise heterogeneous materials with abrupt property jumps across interfaces, PINNs may have difficulty accurately capturing abrupt changes in strain or stress gradients.

Based on the aforementioned issues, this study proposes an element-wise physics-informed neural network for 2D linear elasticity problems. Instead of approximating a continuous displacement field over the spatial domain, the proposed framework operates directly on discrete nodal degrees of freedom defined on a predefined finite element mesh. Accordingly, the neural network is trained to predict structural responses by minimizing the TPE evaluated element-wise. The nodal coordinates are used as the training inputs, resulting in a significantly reduced number of training data points compared to conventional PINNs. Furthermore, by conforming to a triangular mesh, the proposed framework naturally accommodates complex geometries and enforces boundary conditions in a manner analogous to the classical FEM. The unknown nodal displacements, serving as the network outputs, are then used to construct a derivative-free loss function, which improves training efficiency and reduces computational cost. Once the training process is completed, the structural responses are obtained directly without the need for additional solvers. The effectiveness of the proposed EWPINN is demonstrated through three numerical examples: the first example highlights its capability in handling concentrated forces, the second evaluates its computational efficiency and accuracy in comparison with conventional PINN approaches, and the third illustrates its applicability to problems involving piecewise heterogeneous materials.

The remainder of this paper is organized as follows. Section 2 presents the proposed EWPINN framework for plane elasticity problems. Subsequently, several benchmark examples are presented to evaluate the accuracy, effectiveness, and robustness of the proposed model in Section 3. Finally, Section 4 summarizes the conclusions and suggests directions for future research.

2. ELEMENT-WISE PHYSICS-INFORMED NEURAL NETWORK

In this section, an element-wise physics-informed neural network is introduced for solving planar elasticity problems. In this framework, the NN is employed to estimate the structural responses by minimizing the TPE, and the overall workflow is depicted in Fig. 1. According to this scheme, the network parameters θ , including weights and biases, are treated as design variables, rather than the displacement field as in conventional methods. The computational domain is first discretized into triangular plane elements. The spatial coordinates of the nodes, $\mathbf{C} \in \mathbb{R}^{n \times 2}$, are then collected as the training input data, while the corresponding unknown nodal displacements, $\hat{\mathbf{u}}' \in \mathbb{R}^{2n \times 1}$, are taken as the network outputs. Based on the predicted displacements obtained from the feedforward propagation, the TPE is formulated as the loss function, incorporating external forces, boundary conditions, and material properties. During training, the gradients of the network parameters are automatically calculated through backpropagation, and an optimizer is used to adjust the weights and biases iteratively until

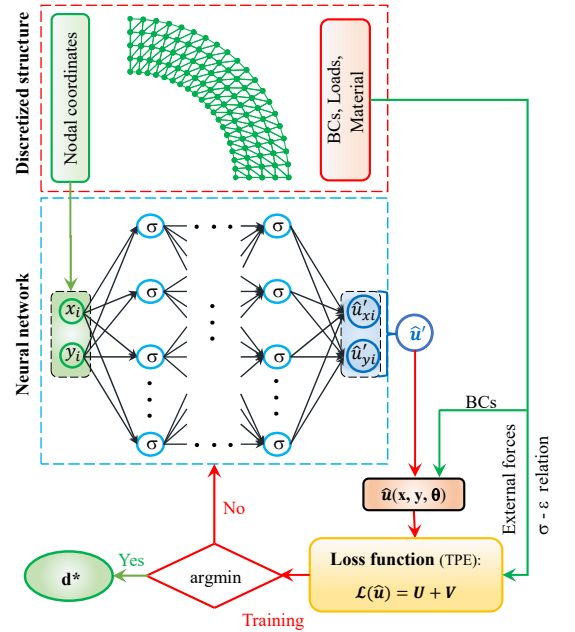


Fig. 1. Schematic of the element-wise physics-informed neural network framework

convergence. Upon completion of the training process, the structural responses corresponding to the minimum total potential energy are obtained directly. To fully understand this paradigm, the following subsections provide detailed descriptions of the training data, network architecture, and loss function formulation.

2.1. Training data

Unlike the conventional PINNs that rely on collocation points, the EWPINN directly uses the coordinates of finite element mesh nodes $\mathbf{C} = [\mathbf{x}, \mathbf{y}]$ as training data. As a result, the size of the training dataset is significantly reduced to $(n \times 2)$, where n denotes the number of mesh nodes. This mesh-based strategy also enables simple and efficient data generation for complex geometries using standard mesh generators. To accelerate the convergence speed, all training data are normalized to the interval $[-1, 1]$. The training dataset consists solely of nodal coordinates and does not include displacement, stress, or strain information. Instead, the structural responses are treated as unknown quantities to be determined during training. External loads, material properties, and boundary conditions are incorporated implicitly through the loss function formulation.

2.2. Deep neural network

A fully connected neural network architecture, illustrated in Fig. 2, is employed to predict the structural responses from the spatial coordinates of the mesh nodes. The network is composed of an input layer (0th), $(\ell - 1)$ hidden layers, and an output layer (ℓ th). In this configuration, the first layer contains two neurons corresponding to the spatial coordinates, while the output layer comprises two neurons representing the predicted nodal displacements $[\hat{u}'_x, \hat{u}'_y]$. All intermediate layers are fully connected hidden layers with the same number of neurons. The trainable parameters of the network, including weights and biases, are defined as follows:

$$\theta = \left\{ w_{ij}^{(k)}, b_i^{(k)} \mid i = 1, 2, \dots, n_\ell; j = 1, 2, \dots, n_{\ell-1}; k = 1, 2, \dots, \ell \right\}, \quad (1)$$

where $w_{ij}^{(k)}$ is the weight connecting neuron j in the $(k - 1)$ th layer to neuron i in the (k) th layer, and $b_i^{(k)}$ denotes the bias associated with neuron i in the (k) th layer; n_0 , n_k , and n_ℓ refer to the numbers of neurons in the input, hidden, and output layers, respectively. The initial parameter values are assumed to be uniformly distributed in the interval $[-1, 1]$.

In general, the predicted displacement field is found by mapping the nodal coordinates $\mathbf{C}$ through the feedforward propagation of the neural network, resulting in: $\mathbb{R}^2 \rightarrow \mathbb{R}^2/\mathbf{C} \Rightarrow \hat{\mathbf{u}}'(\mathbf{C}, \theta)$. The outputs of each network layer are expressed as follows:

$$\begin{aligned} \text{input layer:} \quad & \mathbf{a}^0 = \mathbf{C}(i, :) = [x_i, y_i]^T \in \mathbb{R}^2, \\ \text{hidden layers:} \quad & \mathbf{a}^k = \sigma_h \left(\mathbf{W}^k \mathbf{a}^{(k-1)} + \mathbf{b}^k \right) \in \mathbb{R}^{n_\ell}, \quad \text{for } 1 \leq k \leq (\ell - 1), \\ \text{output layer:} \quad & \mathbf{a}^\ell = \sigma_o \left(\mathbf{W}^\ell \mathbf{a}^{(\ell-1)} + \mathbf{b}^\ell \right) = [\hat{u}'_{xi}, \hat{u}'_{yi}]^T = \hat{\mathbf{u}}'_i \in \mathbb{R}^2, \end{aligned} \quad (2)$$

where n_ℓ denotes the number of neurons in the ℓ th hidden layer; $\hat{\mathbf{u}}'_i$ represents the network's predicted displacement vector at the i th node; $\sigma(\cdot)$ and σ_o are the nonlinear activation functions

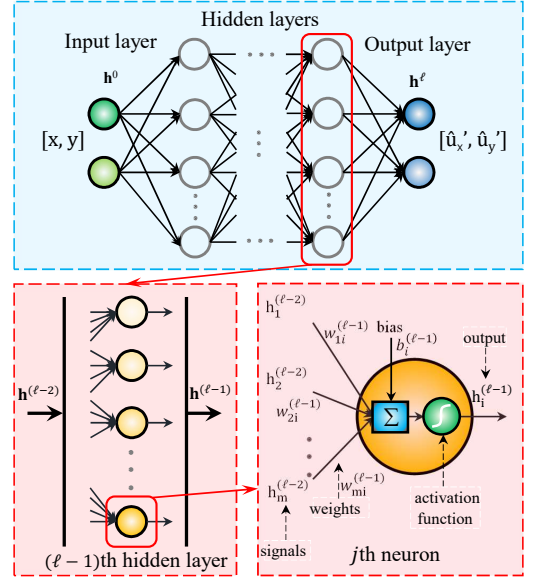


Fig. 2. Fully connected neural network architecture

used in the hidden and output layers, respectively. These activation functions enable the network to capture complex nonlinear relationships between the inputs and outputs. Several activation functions are commonly used, such as Sigmoid, Tanh, Softmax, and LeakyReLU. In this study, ReLU and linear activation functions are employed for the hidden and output layers, respectively.

It is worth noting that the predicted displacements obtained from the network do not strictly satisfy the prescribed boundary conditions. In previous studies, a penalty term is added to the loss function to softly enforce the boundary conditions. Nevertheless, this leads to higher computational costs and slower convergence. To address this issue, a trial function-based hard enforcement strategy is adopted for Dirichlet boundary conditions. Specifically, a boolean mask vector μ is introduced to distinguish constrained and unconstrained degrees of freedom, and the predicted displacement field that exactly satisfies the Dirichlet boundary conditions is constructed as

$$\hat{\mathbf{u}} = \bar{\mathbf{u}} + \mu \odot \hat{\mathbf{u}}', \quad (3)$$

where $\hat{\mathbf{u}}'$ denotes the raw NN output; $\bar{\mathbf{u}}$ represents the prescribed displacement vector associated with non-homogeneous Dirichlet boundary conditions; and the mask vector μ is defined component-wise as

$$\mu_i = \begin{cases} 0 & \text{if degree of freedom } i\text{th is constrained,} \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

By construction, the trial function in Eq. (3) enforces $\hat{u}_i = \bar{u}_i$ at all constrained degrees of freedom, while the NN only learns the displacement field on the unconstrained degrees of freedom.

2.3. Loss function

The predicted displacement field $\hat{\mathbf{u}}$, which satisfies the essential boundary conditions, is then used to construct the loss function of the network. Consider a design domain discretized using a triangular finite element mesh with m elements and n nodes. The TPE of the structure Π_p can be expressed as follows:

$$\Pi_p(\hat{\mathbf{u}}) = \sum_{e=1}^m U_e(\hat{\mathbf{u}}) + V(\hat{\mathbf{u}}), \quad (5)$$

where

$$V(\hat{\mathbf{u}}) = -\mathbf{f}^T \hat{\mathbf{u}}, \quad (6)$$

$$U_e(\hat{\mathbf{u}}) = \frac{1}{2} A_e \boldsymbol{\varepsilon}(\hat{\mathbf{u}}_e)^T \mathbf{D} \boldsymbol{\varepsilon}(\hat{\mathbf{u}}_e), \quad (7)$$

in which V denotes the external work; $\mathbf{f}$ is the external force vector; U_e , $\boldsymbol{\varepsilon}$, A_e , and $\hat{\mathbf{u}}_e$ refer to the strain energy, strain vector, element area, and predicted displacement vector of the e th element, respectively; and $\mathbf{D}$ is the constitutive matrix. For the plane stress problem, it is given by:

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{pmatrix}. \quad (8)$$

For plane strain conditions, the constitutive matrix is expressed as follows:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} (1-\nu) & \nu & 0 \\ \nu & (1-\nu) & 0 \\ 0 & 0 & (1-2\nu)/2 \end{pmatrix}, \quad (9)$$

where E and ν are the Young's modulus and the Poisson's ratio, respectively.

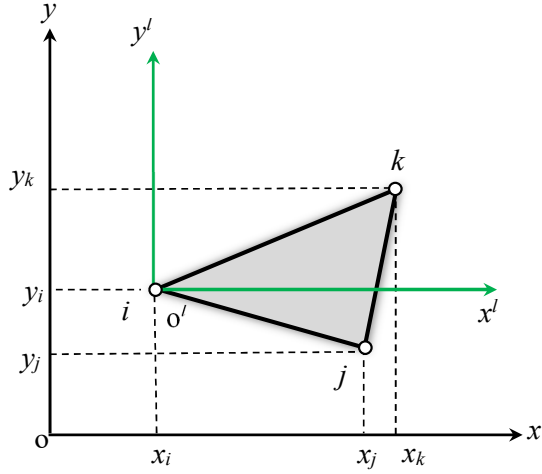


Fig. 3. Three-node triangular element with the local and global coordinates

The strain vector ε is fully determined by the predicted displacement field that satisfies the prescribed boundary conditions. Consider a triangular element e with vertices labeled i, j, k , as shown in Fig. 3. The global coordinates of the p th node are denoted by $\mathbf{X}_p = [x_p, y_p]^T$, where $p = i, j, k$. The i th node is selected as the local origin, such that $\mathbf{X}_i^l = \mathbf{0}$. The relationship between the local and global coordinates of nodes j and k is given by:

$$\begin{bmatrix} \mathbf{X}_j^l & \mathbf{X}_k^l \end{bmatrix} = \begin{bmatrix} x_j^l & x_k^l \\ y_j^l & y_k^l \end{bmatrix} = \begin{bmatrix} (x_j - x_i) & (x_k - x_i) \\ (y_j - y_i) & (y_k - y_i) \end{bmatrix}. \quad (10)$$

In this study, the material is assumed to be linear elastic, isotropic, homogeneous, and continuously distributed throughout its volume. The element is further assumed to deform within the x - y plane. Under these assumptions, the predicted displacement field $[\hat{u}_x, \hat{u}_y]^T$ at any point within the element can be expressed as:

$$\begin{bmatrix} \hat{u}_x \\ \hat{u}_y \end{bmatrix} = \begin{bmatrix} \hat{u}_{xi} \\ \hat{u}_{yi} \end{bmatrix} + \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (11)$$

where $[\hat{u}_{xi}, \hat{u}_{yi}]^T$ is the predicted displacement vector at the i th node; m_1, m_2, m_3 , and m_4 denote the linear interpolation constants, which are determined by substituting the predicted nodal displacements at the j th and k th nodes into Eq. (11). These coefficients can be expressed as:

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \frac{1}{(x_j^l y_k^l - x_k^l y_j^l)} \begin{bmatrix} y_k^l & -y_j^l & 0 & 0 \\ -x_k^l & x_j^l & 0 & 0 \\ 0 & 0 & y_k^l & -y_j^l \\ 0 & 0 & -x_k^l & x_j^l \end{bmatrix} \begin{bmatrix} \hat{u}_{xj} - \hat{u}_{xi} \\ \hat{u}_{xk} - \hat{u}_{xi} \\ \hat{u}_{yj} - \hat{u}_{yi} \\ \hat{u}_{yk} - \hat{u}_{yi} \end{bmatrix}. \quad (12)$$

In classical elasticity, the element strains are obtained from the spatial derivatives of the linear displacement field. Accordingly, the strain vector $\varepsilon = [\varepsilon_x, \varepsilon_y, \gamma_{xy}]^T$ is given by:

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{u}_x}{\partial x} \\ \frac{\partial \hat{u}_y}{\partial y} \\ \frac{\partial \hat{u}_x}{\partial y} + \frac{\partial \hat{u}_y}{\partial x} \end{bmatrix} = \begin{bmatrix} m_1 \\ m_4 \\ m_2 + m_3 \end{bmatrix}. \quad (13)$$

The TPE is therefore fully determined by the predicted displacements, external loads, and boundary conditions, with the predicted displacement field depending on the network training parameters. In the proposed framework, the TPE is adopted as the loss function and minimized during training. Its mathematical expression is given by:

$$\mathcal{L}(\hat{\mathbf{u}}(\boldsymbol{\theta})) = \sum_{e=1}^m U_e(\hat{\mathbf{u}}(\boldsymbol{\theta})) + V(\hat{\mathbf{u}}(\boldsymbol{\theta})). \quad (14)$$

In the proposed framework, unlike the data-driven approaches, the loss function is constructed solely from the network outputs and the governing physical laws, without requiring any additional labeled data. As a result, the training outcomes inherently satisfy the underlying physical constraints. Moreover, the loss function does not involve spatial derivative terms, which reduces computational overhead and improves training efficiency. The structural responses are obtained by minimizing the loss function during the training process to identify the optimal network parameters $\boldsymbol{\theta}^*$.

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}). \quad (15)$$

In this study, the Adam optimizer is employed to update the trainable network parameters (Kingma & Ba, 2014). After training, the network directly provides the displacement field and the corresponding structural responses.

3. NUMERICAL EXAMPLES

In this section, three numerical examples are presented to validate the effectiveness of the EWPINN for solving plane elasticity problems. The obtained results are compared with those from the FEM, the Parametric Deep Energy Method (P-DEM), and available analytical solutions. In all examples, both the input and output layers consist of two neurons, while the number of hidden neurons is determined through trial-and-error and grid search procedures. The ReLU and linear activation functions are employed in the hidden and output layers, respectively. The training process is terminated once the prescribed maximum number of epochs is reached. All examples are implemented using Python on a personal computer equipped with a 3.0 GHz Intel Core i5-8500 CPU with 16 GB of RAM to ensure a fair comparison among the considered methods. Several evaluation metrics are used to assess the performance of the EWPINN, including the maximum error (MAX), mean absolute error (MAE), mean absolute percentage error (MAPE), and the coefficient of determination (R^2) of the predicted structural responses. These metrics are defined as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{s}_i - s_i|, \quad (16)$$

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \frac{|\hat{s}_i - s_i|}{\max(|s_i|, \gamma)}, \quad (17)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (s_i - \hat{s}_i)^2}{\sum_{i=1}^N (s_i - \bar{s})^2}, \quad (18)$$

where N represents the total number of degrees of freedom; $\hat{s}$ denotes the predicted displacement or stress; s is the reference solution, which can be obtained from an exact solution or from the FEM; $\gamma > 0$ is a small positive constant introduced to improve numerical stability for very small values of s_i , and is set to 10^{-12} ; and $\bar{s}$ denotes the average value of the reference solution s .

3.1. Disk under diametrical compression

A circular disk with a thickness of 1 mm, subjected to diametrically opposed point loads, was examined as the first benchmark problem in the linear analysis, as illustrated in Fig. 4(a). As the structure is symmetrical, only one-quarter of the domain was modeled and discretized using 60 three-node triangular elements with 42 nodes, as depicted in Fig. 4(b). The material was assumed to be linearly elastic with Young's modulus $E = 100 \text{ N/mm}^2$ and Poisson's ratio $\nu = 0.25$ under plane stress conditions. This benchmark problem has been previously studied by Sadd (2009). The network configuration reported in Table 1 yielded the best performance for this structure.

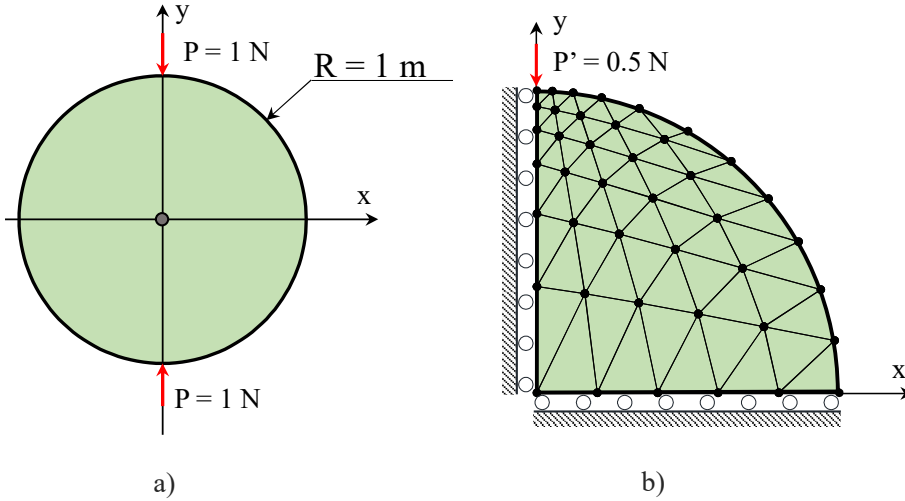


Fig. 4. Disk under diametrical compression

Table 1. Architecture of the neural network for the plane elasticity problems in this study

Problems	Architecture of the network		
	Depth	Width	Epoch _{max}
Disk under diametrical compression	4	100	1,000
Thick-walled cylinder under pressure	4	200	1,000
Soil under buttress dam	4	150	1,000

Table 2. Minimum loss values (N·mm) obtained using various optimizers and activation functions

Activation function	Optimizers					
	Adam	Adamw	Adadelta	Adagrad	RMSprop	SGD
Softmax	−396.2230	−396.2143	−306.4237	−381.2721	−393.3561	−392.7330
Softplus	−392.0778	−382.8911	−245.9307	−342.2526	−322.1573	−339.3639
Tanh	−390.8364	−387.0729	−219.0406	−304.7311	−385.8454	−210.6764
Sigmoid	−389.7309	−332.4575	−167.5762	−271.7305	−383.2635	−172.6379
ReLU	−396.2304	−395.9704	−344.4298	−390.6543	−393.3794	−388.3073
LeakyReLU	−396.2303	−395.5607	−350.4776	−388.6118	−393.2134	−386.5741

Based on the above network architecture, an investigation is performed to evaluate the effects of different optimizer and activation function combinations on the model performance. In this analysis, the TPE is used as the evaluation metric. A summary of the obtained TPE results is reported in Table 2. Among the considered optimizers, Adam achieves the best performance, while Adadelata performs the worst. In particular, the Adam–ReLU combination attains the smallest TPE value of -396.2304 N.mm, and is therefore selected for use in this study. In addition, the influence of network depth and width on prediction accuracy is also examined. As shown by the TPE values listed in Table 3, increasing the number of hidden neurons or hidden layers does not result in further performance improvement. Consequently, the NN architecture (2–100–100–100–2) listed in Table 1 is identified as the most suitable configuration for this problem.

Table 3. Influence of network depth and width on the loss value

Number of hidden layer	Number of hidden neuron					
	50	100	150	200	250	300
1	−290.5074	−297.9996	−317.4125	−330.1326	−353.8959	−318.2918
2	−360.5435	−354.6797	−383.0086	−348.2744	−354.8917	−361.1017
3	−395.2266	−396.1791	−396.1244	−396.2291	−396.1554	−396.0772
4	−396.2232	−396.2304	−396.2265	−396.2304	−396.2303	−396.2301
5	−396.2303	−396.2301	−396.2302	−396.2303	−396.2284	−396.2289

Fig. 5 illustrates the convergence history of the network loss. As can be seen, the TPE value initially decreases rapidly during the early stages of training, stabilizes after approximately 400 epochs, and converges within 1,000 epochs. This behavior is attributed to the gradient-based optimization strategy adopted in the proposed framework. Moreover, the absence of spatial derivative terms in the loss function contributes to faster convergence and reduces the number of training epochs required. The TPE obtained by the proposed model (-396.2294 N mm) is in excellent agreement with the FEM result (-396.2304 N mm), with only a negligible difference. Tables 4, 5 and Fig. 6 present a comparison between the results obtained by the EWPINN and the FEM. The contour plots indicate that the predicted displacement and stress fields closely match the FEM solutions. More specifically, the absolute displacement errors are on the order of 1×10^{-5} mm for a maximum displacement of 1.58 mm, while the stress errors are approximately 1×10^{-2} N/mm² for a maximum stress of 297 N/mm². Furthermore, as reported in Table 4, all coefficients of determination R^2 are very close to unity. The MAEs and MAPEs of the predicted displacements are below 0.000006 and 0.012 %, while those for the predicted stresses are below 0.0015 and 0.018 %. These results demonstrate the accuracy and reliability of the developed framework for plane stress analysis. As summarized in Table 5, the FEM achieves higher computational efficiency than the EWPINN in terms of both CPU time and memory usage. This can be attributed to the fact that the FEM solves a system of linear equations in a single iteration, whereas the EWPINN treats the problem as an optimization task

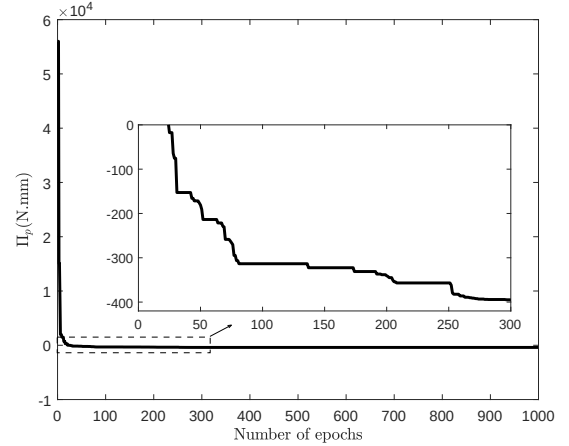


Fig. 5. The loss convergence history of the disk under diametrical compression

by minimizing the total potential energy over the discretized elements. As a result, the EWPINN incurs a higher overall computational cost while achieving a similar level of accuracy. Despite this, the EWPINN exhibits a much lower computational cost per iteration (0.00797 s). This is mainly because the EWPINN does not require the assembly of a global stiffness matrix. This feature suggests that the EWPINN may become advantageous in nonlinear analyses, where conventional FEM approaches usually rely on incremental and iterative solution schemes. In addition, most existing PINN-based structural analysis methods, whether based on strong-form PDE residuals or energy minimization, tend to avoid concentrated loading conditions, as point loads produce highly localized stress fields that are difficult to represent using smooth NN approximations. In contrast, the present model is formulated within a displacement-based energy framework, where the TPE is directly evaluated from the predicted nodal displacement field.

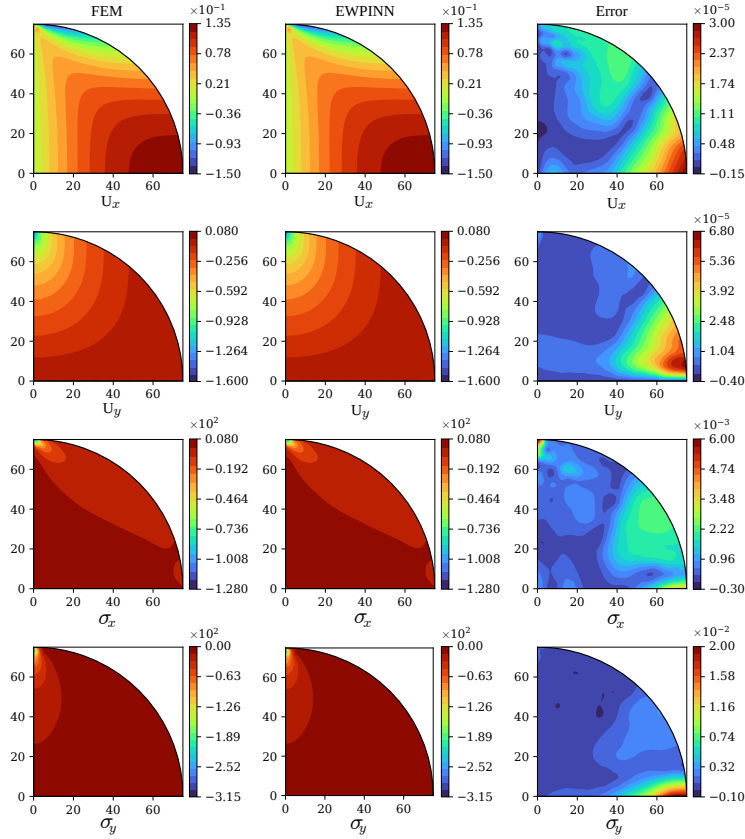


Fig. 6. Analysis results of the disk under diametrical compression by EWPINN and FEM

Table 4. Statistical results of the EWPINN compared with FEM for the circular disk

Structural responses	EWPINN			
	MAX	MAE	MAPE (%)	R ²
u_x	-0.1539	0.0000	0.0112	1.0000
u_y	-1.5849	0.0000	0.0028	1.0000
σ_x	-123.4883	0.0010	0.0171	1.0000
σ_y	-297.0768	0.0013	0.0081	1.0000

Table 5. Comparison of accuracy and computational efficiency for the circular disk

Method	TPE (N.mm)	No. of nodes	No. of dofs	Total iterations	CPU time per iteration (s)	Total CPU time(s)	Memory usage (MB)
FEM	-396.2294	42	84	1	1.45	1.45	11.03
EWPINN	-396.2304	42	84	1,000	0.00797	7.97	22.89

3.2. Thick walled cylinder under pressure

The second example considers a thin annular disk with an internal radius $r_i = 1 \mu\text{m}$ and an outer radius $r_o = 5 \mu\text{m}$. The disk is subjected to an internal pressure of $p_i = 10 \text{ MPa}$ acting radially, as illustrated in Fig. 7(a). Plane stress conditions are assumed. The structure is made of a material with an elastic modulus of $E = 135 \text{ GPa}$ and a Poisson's ratio of 0.3. Due to geometric symmetry, only a quarter of the domain is modeled. To evaluate the solution accuracy, two meshes are considered for both the FEM and EWPINN approaches: a coarse mesh with 121 nodes and a fine mesh with 441 nodes, as illustrated in Fig. 7(b). The network architecture adopted for the EWPINN consists of four hidden layers with 200 neurons per layer and is trained for 1,000 epochs, as summarized in Table 1. In addition, the P-DEM (Nguyen-Thanh et al., 2021), employed as an alternative energy-based PINN, is used to evaluate the computational efficiency of the EWPINN. Unlike the EWPINN, the P-DEM is trained using collocation points instead of nodal points. The numerical results are compared with the analytical solution reported in the literature (Nguyen-Thanh et al., 2021; Timoshenko, 1934).

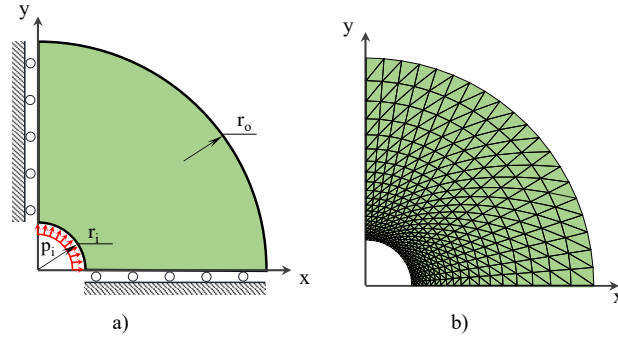


Fig. 7. A quarter of the annulus

The loss convergence history of the EWPINN is shown in Fig. 8. Similar to the first example, the convergence speed increases rapidly in the initial stage, and the solution converges within 1,000 epochs. As in the previous example, the analysis results obtained by the proposed method are summarized in Tables 6 and 7 and compared with those of the FEM and P-DEM. The minimum TPE found by the EWPINN shows excellent agreement with the FEM results for both mesh resolutions. This agreement is further confirmed by the statistical measures, where the coefficients of determination R^2 for the predicted displacement and stress fields are close to 1. In addition, the MAE and MAPE values remain very small for both displacements and stresses. The P-DEM is able to produce accurate results when 6,480 collocation points are used, whereas convergence is not achieved with 2,500 points. Although both the EWPINN and the P-DEM minimize the TPE using neural networks, the P-DEM requires a larger number of collocation points, while the proposed EWPINN attains comparable accuracy using only 441 nodal points. Figs. 9 and 10 provide the numerical solutions and absolute errors with respect to the analytical solution. The results indicate strong agreement among the methods, with displacement and stress errors on the order of 10^{-6} and 10^{-3} , respectively. In terms of computational efficiency,

the FEM remains the most efficient approach for this example. However, its computational cost increases by approximately fourfold as the mesh is refined. In contrast, that of the EWPINN increases only slightly, by about 2 seconds, demonstrating good scalability with respect to mesh refinement. Compared with the P-DEM, the EWPINN achieves similar accuracy with significantly lower computational cost, as reflected by fewer iterations and shorter CPU time. Moreover, the EWPINN exhibits a lower CPU time per iteration (0.013 s) than the P-DEM (0.07035 s). This is mainly because the loss function of the EWPINN does not involve derivative information, leading to reduced computational complexity per iteration.

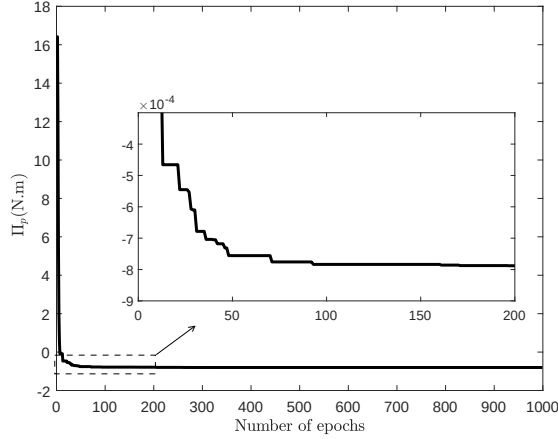


Fig. 8. The loss convergence of the pressurized annulus

Table 6. Comparison of accuracy and computational efficiency for the annulus problem

Method	TPE (N.mm)	No. of nodes	No. of dofs	Total iterations	CPU time per iteration (s)	Total CPU time(s)	Memory usage (MB)
Mesh 1							
FEM	-7.889E-04	121	242	1	1.2	1.2	30
P-DEM	-	2,500	-	-	-	-	-
EWPINN	-7.892E-04	121	242	1,000	0.011	11	68.47
Mesh 2							
FEM	-8.024E-04	441	882	1	5.6	5.6	35.47
P-DEM	-8.047E-04	6,480	12,960	20,000	0.07035	1407.0	95.51
EWPINN	-8.024E-04	441	882	1,000	0.01300	13.0	72.82

Table 7. Statistical results of the EWPINN compared with FEM for the annulus problem

Structural responses	EWPINN			
	Max	MAE	MAPE (%)	R ²
u_x	1.03E-04	0.0000	0.0984	1.0000
u_y	1.01E-04	0.0000	0.1029	1.0000
σ_x	0.0104	0.0000	0.1638	1.0000
σ_y	0.0109	0.0000	0.1261	1.0000

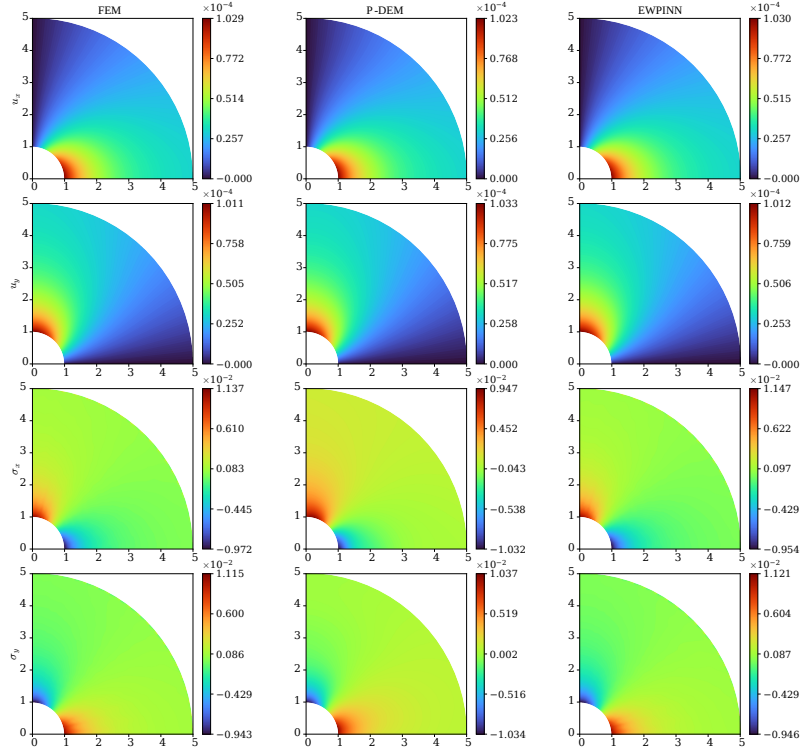


Fig. 9. Analysis results of the pressurized annulus by FEM, P-DEM and EWPINN

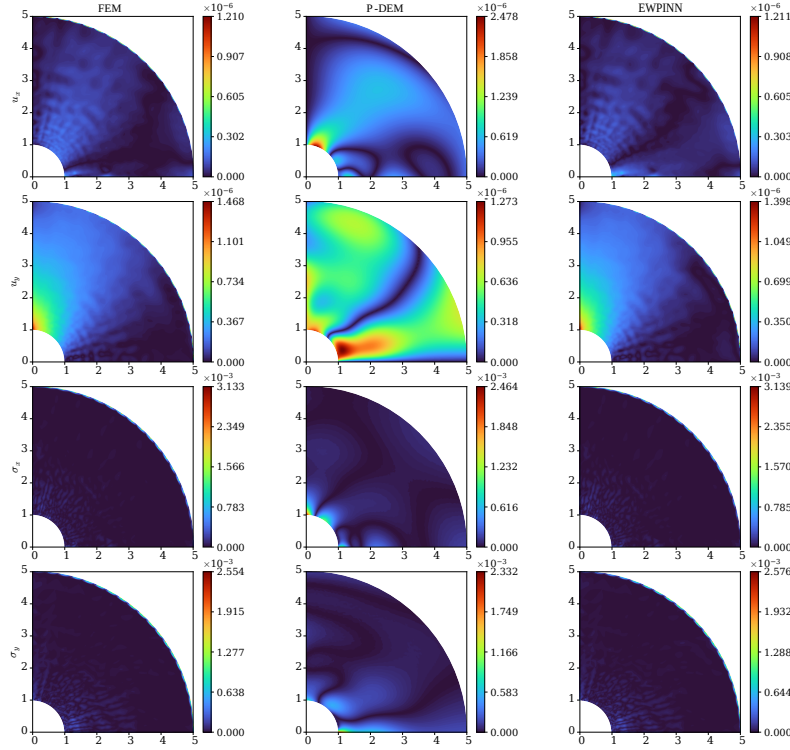


Fig. 10. Error distribution between FEM, P-DEM, and EWPINN for the pressurized annulus with the analytical solution

3.3. Soil under buttress dam

To further evaluate the efficiency and reliability of the EWPINN, the final benchmark problem involves a plane strain analysis of a multilayered soil foundation supporting a buttress dam. The geometry, loading conditions, boundary conditions, and finite element mesh are illustrated in Fig. 11. The soil foundation consists of three distinct layers: mudstone, grit, and altered grit, with elastic moduli of $E_m = 10342 \text{ N/mm}^2$, $E_g = 5171 \text{ N/mm}^2$, and $E_a = 2068 \text{ N/mm}^2$, respectively. Each layer has a thickness of 1 mm. All soil layers are assumed to have a Poisson's ratio of 0.25. The computational domain is discretized into 125 elements and 76 nodes for the training. The self-weight of the buttress dam is applied to the top surface of the soil as a distributed load. This load is approximated by equivalent concentrated forces at six nodes: $P_1 = 104657 \text{ N}$, $P_2 = 627941 \text{ N}$, $P_3 = 1464725 \text{ N}$, $P_4 = 2406008 \text{ N}$, $P_5 = 1255569 \text{ N}$, and $P_6 = 104657 \text{ N}$, as shown in Fig. 11. This structure is subjected to pronounced material heterogeneity and localized loading. As summarized in Table 1, the network architecture consists of four hidden layers with 150 neurons per layer and is trained for 1,000 epochs.

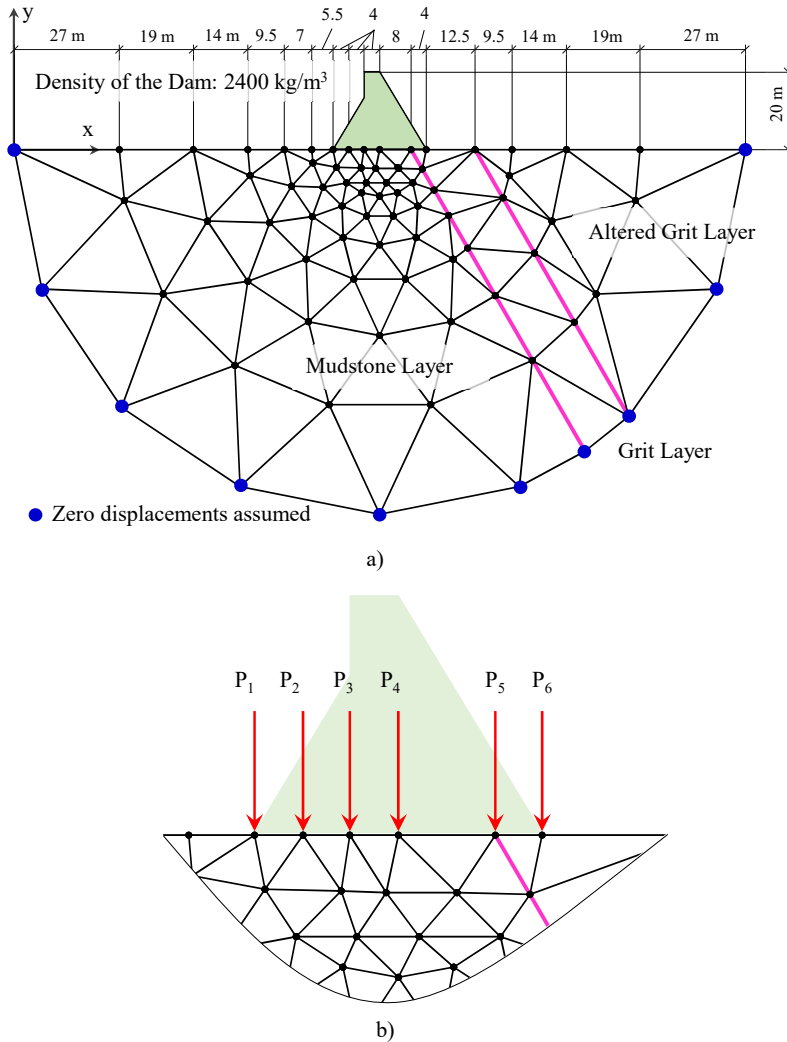


Fig. 11. Soil under a buttress dam

The convergence history of the TPE is illustrated in Fig. 12. Once again, the results show that the model demonstrates fast convergence in the early training epochs. In this case, the

minimum TPE found by the EWPINN (-2784437.5047 kN.mm) is very close to that of the FEM (-2784437.5032 kN.mm). As in the previous examples, Tables 8, 9 and Figs. 12, 13 summarize the comparative results between the proposed framework and the FEM. The results indicate that the differences between the two approaches are very small for all structural responses. In particular, the MAEs of the predicted quantities are below 10^{-4} . Although the structure is affected by material heterogeneity, the statistical results indicate strong agreement between the EWPINN and the FEM solutions. The coefficients of determination R^2 are close to 1, while the MAPE values are less than 0.003 %. These results demonstrate that the proposed method maintains high accuracy not only for homogeneous materials but also for heterogeneous configurations. As observed in the previous examples, the overall computational cost of the EWPINN remains higher than that of the FEM due to the iterative nature of the training process.

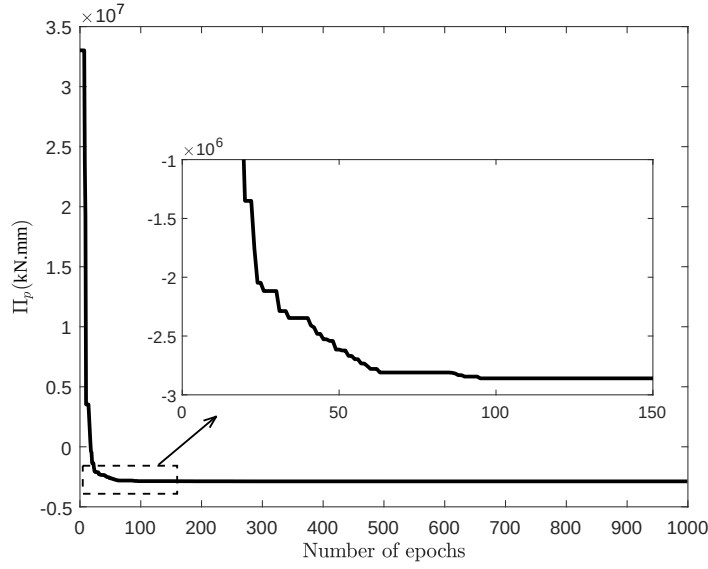


Fig. 12. Loss convergence history of the network for the soil under a buttress dam

Table 8. Comparison of accuracy and computational efficiency for the soil under a buttress dam

Method	TPE (N.mm)	No. of nodes	No. of dofs	Total iterations	CPU time per iteration (s)	Total CPU time(s)	Memory usage (MB)
FEM	$-2.7844\text{E}+06$	76	152	1	1.02	1.02	12.16
EWPINN	$-2.7844\text{E}+06$	76	152	1,000	0.00751	7.51	20.37

Table 9. Statistical results of the EWPINN compared with FEM for the soil under a buttress dam

Structural responses	EWPINN			
	Max	MAE	MAPE (%)	R^2
u_x	0.1993	0.0000	0.0020	1.0000
u_y	-1.0215	0.0000	0.0007	1.0000
σ_x	-0.2102	0.0000	0.0041	1.0000
σ_y	-0.3020	0.0000	0.0029	1.0000

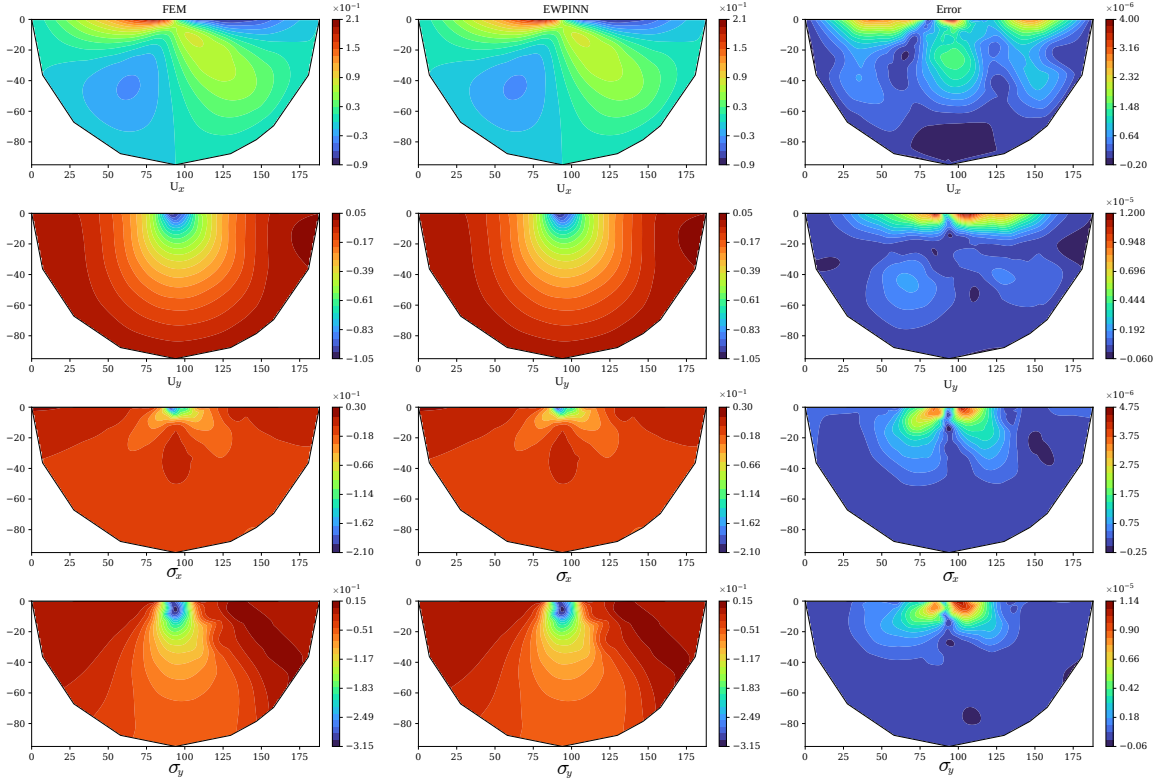


Fig. 13. Analysis results of the soil under a buttress dam by FEM and EWPINN

4. CONCLUSIONS

This study introduced a physics-informed neural finite element approach for solving plane elasticity problems. Unlike conventional finite element methods, which require the assembly and solution of a global system of equations, the proposed framework trains a neural network to minimize a derivative-free loss function comprising the TPE and hard boundary conditions. Consequently, the structural responses are obtained directly upon completion of the training process. The accuracy and reliability of the proposed EWPINN are demonstrated through three numerical examples involving plane stress and plane strain analyses. The numerical results show that the proposed framework can accurately predict structural responses with small relative errors and strong agreement with the FEM. It can directly handle concentrated forces and piecewise heterogeneous materials commonly encountered in practical engineering problems. In addition, the use of nodal spatial coordinates as training inputs significantly reduces the number of training points compared to conventional PINN approaches based on collocation points. The derivative-free loss function also contributes to improved training stability and faster convergence. Although the computational cost of the EWPINN remains higher than that of the FEM due to the iterative training process, the method avoids explicit stiffness matrix assembly and conventional solver routines. Given these favorable characteristics, the EWPINN demonstrates strong potential as an alternative computational framework for structural analysis, with promising applicability to nonlinear problems.

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.

CREDIT AUTHOR STATEMENT

Si T. Do: *Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing.* Dai D. Mai: *Conceptualization, Funding acquisition, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing.*

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

- Abueidda, D. W., Koric, S., Al-Rub, R. A., Parrott, C. M., James, K. A., & Sobh, N. A. (2022). A deep learning energy method for hyperelasticity and viscoelasticity. *European Journal of Mechanics-A/Solids*, 95, 104639. <https://doi.org/10.1016/j.euromechsol.2022.104639>
- Bai, J., Lin, Z., Wang, Y., Wen, J., Liu, Y., Rabczuk, T., Gu, Y., & Feng, X.-Q. (2025). Energy-based physics-informed neural network for frictionless contact problems under large deformation. *Computer Methods in Applied Mechanics and Engineering*, 437, 117787. <https://doi.org/10.1016/j.cma.2025.117787>
- Bai, J., Rabczuk, T., Gupta, A., Alzubaidi, L., & Gu, Y. (2022). A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics. *Computational Mechanics*, 71(3), 543–562. <https://doi.org/10.1007/s00466-022-02252-0>
- Berke, L., & Hajela, P. (1992). Applications of artificial neural nets in structural mechanics. *Structural Optimization*, 4, 90–98. <https://doi.org/10.1007/BF01759922>
- Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6), 060801. <https://doi.org/10.1115/1.4050542>
- Elfetni, S., & Darvishi Kamachali, R. (2025). PINN-phase: A physics-informed neural network hybrid framework for energy-based transfer learning in diffuse interface problems. *Available at SSRN 5207041*. <https://doi.org/10.2139/ssrn.5207041>
- Feng, N., Zhang, G., & Khandelwal, K. (2022). Finite strain FE2 analysis with data-driven homogenization using deep neural networks. *Computers & Structures*, 263, 106742. <https://doi.org/10.1016/j.compstruc.2022.106742>
- Ghose, D., Anandh, T., & Ganesan, S. (2024). FastVPINNs: A fast, versatile and robust variational PINNs framework for forward and inverse problems in science. *ICLR 2024 Workshop on AI4DifferentialEquations In Science*.
- Hajela, P., & Berke, L. (1991). Neurobiological computational models in structural analysis and design. *Computers & Structures*, 41(4), 657–667. [https://doi.org/10.1016/0045-7949\(91\)90178-S](https://doi.org/10.1016/0045-7949(91)90178-S)
- Hau, T. M., Tam, T.-T., Joowon, K., Dai, M., & Jaehong, L. (2022). A robust physics-informed neural network approach for predicting structural instability. *Finite Elements in Analysis and Design*, 216, 103893. <https://doi.org/10.1016/j.finel.2022.103893>
- He, J., Chadha, C., Kushwaha, S., Koric, S., Abueidda, D., & Jasiuk, I. (2023). Deep energy method in topology optimization applications. *Acta Mechanica*, 234(4), 1365–1379. <https://doi.org/10.1007/s00707-022-03449-3>
- Hou, M., Chen, Y., Cao, S., Chen, Y., & Ying, J. (2023). HRW: Hybrid residual and weak form loss for solving elliptic interface problems with neural network. *Numerical Mathematics: Theory, Methods and Applications*, 16(4), 883–913. <https://doi.org/10.4208/nmtma.aa-2023-0097>

- Jeong, H., Batuwatta-Gamage, C., Bai, J., Xie, Y. M., Rathnayaka, C., Zhou, Y., & Gu, Y. (2023). A complete physics-informed neural network-based framework for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 417, 116401. <https://doi.org/10.1016/j.cma.2023.116401>
- Kharazmi, E., Zhang, Z., & Karniadakis, G. E. (2019). Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*. <https://doi.org/10.48550/arXiv.1912.00873>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. <https://doi.org/10.48550/arXiv.1412.6980>
- Kollmann, H. T., Abueidda, D. W., Koric, S., Guleryuz, E., & Sobh, N. A. (2020). Deep learning for topology optimization of 2D metamaterials. *Materials & Design*, 196, 109098. <https://doi.org/10.1016/j.matdes.2020.109098>
- Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5), 987–1000. <https://doi.org/10.1109/72.712178>
- Lee, S., Ha, J., Zokhirova, M., Moon, H., & Lee, J. (2017). Background information of deep learning for structural engineering. *Archives of Computational Methods in Engineering*, 25(1), 121–129. <https://doi.org/10.1007/s11831-017-9237-0>
- Li, W., Bazant, M. Z., & Zhu, J. (2021). A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches. *Computer Methods in Applied Mechanics and Engineering*, 383, 113933. <https://doi.org/10.1016/j.cma.2021.113933>
- Liang, L., Liu, M., Martin, C., & Sun, W. (2018). A deep learning approach to estimate stress distribution: A fast and accurate surrogate of finite-element analysis. *Journal of The Royal Society Interface*, 15(138), 20170844. <https://doi.org/10.1098/rsif.2017.0844>
- Liu, C., & Wu, H. A. (2023). A variational formulation of physics-informed neural network for the applications of homogeneous and heterogeneous material properties identification. *International Journal of Applied Mechanics*, 15(08), 2350065. <https://doi.org/10.1142/s1758825123500655>
- Luo, Z., Wang, L., & Lu, M. (2023). A stepwise physics-informed neural network for solving large deformation problems of hypoelastic materials. *International Journal for Numerical Methods in Engineering*, 124(20), 4453–4472. <https://doi.org/10.1002/nme.7323>
- Mai, D. D., Bao, T. D., Lam, T.-D., & Mai, H. T. (2024). Physics-informed neural network for nonlinear analysis of cable net structures. *Advances in Engineering Software*, 196, 103717. <https://doi.org/10.1016/j.advengsoft.2024.103717>
- Mai, D. D., Do, S. T., Lee, S., & Mai, H. T. (2025). A force neural network framework for structural optimization. *Engineering Applications of Artificial Intelligence*, 143, 109991. <https://doi.org/10.1016/j.engappai.2024.109991>
- Mai, H. T., Kang, J., & Lee, J. (2021). A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior. *Finite Elements in Analysis and Design*, 196, 103572. <https://doi.org/10.1016/j.finel.2021.103572>
- Mai, H. T., Lee, S., Kang, J., & Lee, J. (2024). A damage-informed neural network framework for structural damage identification. *Computers & Structures*, 292, 107232. <https://doi.org/10.1016/j.compstruc.2023.107232>
- Mai, H. T., Lieu, Q. X., Kang, J., & Lee, J. (2022). A robust unsupervised neural network framework for geometrically nonlinear analysis of inelastic truss structures. *Applied Mathematical Modelling*, 107, 332–352. <https://doi.org/10.1016/j.apm.2022.02.036>
- Mai, H. T., Lieu, Q. X., Kang, J., & Lee, J. (2023). A novel deep unsupervised learning-based framework for optimization of truss structures. *Engineering with Computers*, 39(4), 2585–2608. <https://doi.org/10.1007/s00366-022-01636-3>

- Mai, H. T., Mai, D. D., Kang, J., Lee, J., & Lee, J. (2024). Physics-informed neural energy-force network: A unified solver-free numerical simulation for structural optimization. *Engineering with Computers*, 40(1), 147–170. <https://doi.org/10.1007/s00366-022-01760-0>
- Nguyen-Thanh, V. M., Anitescu, C., Alajlan, N., Rabczuk, T., & Zhuang, X. (2021). Parametric deep energy approach for elasticity accounting for strain gradient effects. *Computer Methods in Applied Mechanics and Engineering*, 386, 114096. <https://doi.org/10.1016/j.cma.2021.114096>
- Nguyen-Thanh, V. M., Zhuang, X., & Rabczuk, T. (2020). A deep energy method for finite deformation hyperelasticity. *European Journal of Mechanics-A/Solids*, 80, 103874. <https://doi.org/10.1016/j.euromechsol.2019.103874>
- Radin, N., Klinkel, S., & Altay, O. (2023). Effects of variational formulations on physics-informed neural network performance in solid mechanics. *PAMM*, 23(4), e202300222. <https://doi.org/10.1002/pamm.202300222>
- 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. *Journal of Computational Physics*, 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Ramasamy, J. V., & Rajasekaran, S. (1996). Artificial neural network and genetic algorithm for the design optimization of industrial roofs—a comparison. *Computers & Structures*, 58(4), 747–755. [https://doi.org/10.1016/0045-7949\(95\)00179-K](https://doi.org/10.1016/0045-7949(95)00179-K)
- Sadd, M. H. (2009). *Elasticity: Theory, applications, and numerics*. Academic Press.
- Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375, 1339–1364. <https://doi.org/10.1016/j.jcp.2018.08.029>
- Timoshenko, S. (1934). *Theory of Elasticity*. McGraw-Hill.
- Wang, Y., Sun, J., Rabczuk, T., & Liu, Y. (2024). DCEM: A deep complementary energy method for linear elasticity. *International Journal for Numerical Methods in Engineering*, 125(24), e7585. <https://doi.org/10.1002/nme.7585>
- Yagawa, G., & Okuda, H. (1996). Neural networks in computational mechanics. *Archives of Computational Methods in Engineering*, 3, 435–512. <https://doi.org/10.1007/BF02818935>