Abstract: In practical engineering applications, there is a high demand for inverting parameters for various materials, and obtaining monitoring data can be costly. Traditional inverse methods often involve tedious computational processes, require significant computational effort, and exhibit slow convergence speeds. The recently proposed Physics-Informed Neural Network (PINN) has shown great potential in solving inverse problems. Therefore, in this paper, we propose a transfer learning-based coupling of the Smoothed Finite Element Method (S-FEM) and PINN methods for the inversion of parameters in elastic-plasticity problems. The aim is to improve the accuracy and efficiency of parameter inversion for different elastic-plastic materials with limited data. High-quality small datasets were synthesized using S-FEM and subsequently combined with PINN for pre-training purposes. The parameters of the pre-trained model were saved and used as the initial state for the PINN model in the inversion of new material parameters. The inversion performance of the coupling of S-FEM and PINN is compared with the coupling of the conventional Finite Element Method (FEM) and PINN on a small data set. Additionally, we compared the efficiency and accuracy of both the transfer learning-based and non-transfer learning-based methods of the coupling of S-FEM and PINN in the inversion of different material parameters. The results show that: (1) our method performs well on small datasets, with an inversion error of essentially less than 2%; (2) our approach outperforms the coupling of conventional FEM and PINN in terms of both computational accuracy and computational efficiency; and (3) our approach is at least twice as efficient as the coupling of S-FEM and PINN without transfer learning, while still maintaining accuracy. Our method is well-suited for the inversion of different material parameters using only small datasets. The use of transfer learning greatly improves computational efficiency, making our method an efficient and accurate solution for reducing computational cost and complexity in practical engineering applications.

Keywords: Physics-Informed Neural Network (PINN); Smoothed Finite Element Method (S-FEM); transfer learning; inverse problems; elastic-plastic

1. Introduction

The inverse problem of determining the cause, condition, or input from the effect, performance, or output has been proposed in many fields of scientific research and is commonly referred to as a “mathematical-physical inverse problem” [1]. The importance and necessity of studying inverse problems have been recognized, driven by the needs of natural science and engineering applications. In recent decades, inverse problems have found extensive applications in fields such as geophysics [2–4], resource exploration [5,6], ocean engineering [7], mechanics [8–10], acoustics [11,12], and others.
Traditional mechanical inverse analysis methods usually employ a combination of numerical methods and optimization algorithms. An optimization algorithm is a mathematical method that minimizes or maximizes an objective function by adjusting model parameters or design variables [13]. In mechanical inverse analysis, optimization algorithms are used to adjust model parameters or design variables to minimize the error between simulation results and actual observed or desired results. Traditional mechanical inverse analysis methods typically employ iterative optimization algorithms to incrementally enhance the model, with commonly used algorithms including gradient descent [14], genetic algorithms [15], and particle swarm optimization [16]. However, these conventional optimization algorithms have limitations such as the tendency to converge to local optima and slow convergence speeds. Consequently, numerous novel optimization algorithms have emerged to address these drawbacks. Examples include the Arithmetic Optimization Algorithm [17], the Ibi Logic Algorithm [18], and the modified Sooty Tern Optimization Algorithm [13].

Many researchers have effectively utilized numerical methods and optimization algorithms for the identification of mechanical parameters in elastoplastic materials. For example, De-Carvalho et al. [14] employed two optimization algorithms, namely, the gradient-based Levenberg–Marquardt algorithm and the real search space evolution algorithm, to perform inverse analysis on the constitutive parameters of three different models: the elastic-plastic hardening model, the hyperelastic model, and the elastic-viscoplastic model. The objective was to minimize the disparity between the physical experimental results and the numerical simulation results [14]. Furthermore, a significant number of studies have utilized probability-based parameter estimation methods for mechanical parameter identification. For instance, Khodadadian et al. [19] developed a Bayesian parameter estimation framework to identify the uncertain parameters associated with the phase field fracture problem. Reliable results are obtained by their proposed Bayesian inversion method, even when relatively coarse grids are employed instead of real data [19]. Similarly, Noii et al. [20] applied Bayesian inversion techniques to identify mechanical parameters in various scenarios, including linear elasticity, elastoplasticity, and fracture problems of different materials. They extensively investigated the complex coupled multi-field marginal value problem and shared the open source code of their Bayesian inversion method [20].

However, traditional mechanical inverse analysis methods suffer from cumbersome inversion processes, large computations, and slow convergence efficiencies. The rapid development of artificial intelligence techniques, particularly the impressive capabilities of deep learning in handling high-dimensional complex structural data, have made them powerful tools for solving both forward and inverse problems in mechanics [9,21–24]. For instance, Liu et al. [21,22] proposed a two-way neural network for solving both forward and inverse problems in mechanics, which they combined with nonlinear finite elements to accurately identify the constitutive parameters of hyperelastic materials. Potrzeszcz-Sut et al. [9] utilized a backpropagation neural network in conjunction with indentation test data to efficiently determine the elastic-plastic material parameters described by Ramberg-Osgood’s law. Pichler et al. [24] employed artificial neural networks to approximate finite element analysis and combined them with genetic algorithms to determine optimal parameter estimates for unknown parameter identification. In a similar vein, Liu et al. [25] introduced a surrogate model based on orthogonal decomposition and artificial neural networks. They further integrated this surrogate model with Bayesian theory to achieve the efficient inversion of geotechnical parameters.

Despite the great strides made in deep learning, it cannot be overlooked that its current achievements rely heavily on large, high-quality datasets. However, in practical engineering problems, obtaining a significant amount of high-quality data is often challenging due to measurement difficulties, high costs, and measurement errors. As a result, the accuracy of the computational results may be low or difficult to solve directly using purely data-driven deep learning methods.
In the latest research work, the physics-informed deep learning method proposed by Professor Karniadakis constructs interpretable neural network models by incorporating relevant physical laws or constraints [26]. Physics-informed deep learning guarantees the generalization and validity of the model, even when there is a small amount of data or when noise observation bias, or other such factors are present. This approach enables the model predictions to comply with objective physical constraints [26], ensuring their accuracy and reliability. A typical example of this is the Physics-Informed Neural Network (PINN), which shows great promise in solving both the forward and inverse problems of partial differential equations (PDEs). This is due to its ability to integrate data and PDEs [27–29], making it a powerful tool for researchers in the field. PINNs have proven to be effective in solving many inverse problems of PDEs due to their ease of implementation. This is because the code used for solving the forward problem can be applied to the solution of the inverse problem with minimal modifications [8,30–32].

For example, Fallah et al. [30] utilized PINN to invert the natural frequency of TDFG porous beams on an elastic foundation. They then used this information to solve for the bending and free vibrations of TDFG porous beams using PINN. Depina et al. [31] applied PINNs to the inverse problem of unsaturated groundwater flow. Their research demonstrated that PINNs are capable of accurately identifying the van Genuchten constitutive parameters. Xu et al. [32] improved the training efficiency and accuracy of neural networks for linear elastic and hyperelastic inverse problems by incorporating uncertainty-weighted multi-task learning methods and transfer learning strategies based on PINN. Lu et al. [8] employed physics-informed deep learning to improve the accuracy and efficiency of the inversion process for the mechanical properties of elastic-plastic materials.

Due to the challenges associated with acquiring real measurement data, researchers often resort to numerical methods to synthesize data used to train PINNs when tackling inverse problems involving PDEs. These trained PINNs are subsequently applied to real-world engineering problems where only limited measurement data are available [8,32]. The computational errors arising from numerical methods can be regarded as measurement errors, and PINNs demonstrate robustness in handling such errors. The Finite Element Method (FEM) is commonly employed as the primary numerical method. However, it has certain limitations, such as its high dependency on mesh quality, its inability to handle mesh distortion, and its susceptibility to volume locking issues [33]. To address these problems, the Smoothed Finite Element Method (S-FEM) has been developed. S-FEM is an extension of the FEM that incorporates the smoothing strain technique, enabling it to handle mesh distortion effectively. Moreover, S-FEM offers improved computational accuracy compared to FEM [33–35]. Due to the geometric complexity and irregularity of practical engineering problems, achieving high computational accuracy with quadrilateral cells in the discretization process becomes challenging. In contrast, triangular cells in the S-FEM offer better accuracy when properly adapted to the geometry. Therefore, the use of triangular cells reduces computational complexity while ensuring a better adaptation to real engineering problems [36].

In practical engineering applications, there is a growing need to invert various material parameters. However, traditional machine learning is typically designed for specific tasks, requiring reconstruction of the model after changing the dataset. This can result in a significant increase in computational cost. Transfer learning involves utilizing knowledge from a previously trained model for a new task [37]. For instance, the weights and biases from a pre-trained neural network can be used as initial values for a neural network trained on a similar problem. By avoiding random initialization of the neural network, the training process can converge more quickly.

The combination of transfer learning with PINN for parameter inversion has received relatively little attention in research. Haghighat et al. [38] proposed a PINN framework for solving inverse problems in solid mechanics, but only briefly mentioned its potential for transfer learning without conducting significant analysis or research on the topic. Xu et al. [32] proposed a transfer learning-based PINN method for inverting different load-
ing systems of linear-elastic and hyperelastic structures. However, due to the characteristics of transfer learning, the method presented in [32] still requires a significant amount of data in the pre-training phase.

In this paper, we propose a transfer learning-based coupling of S-FEM and PINN approach to invert different material parameters, with the goal of improving the accuracy and efficiency of the inversion process. The proposed approach involves utilizing S-FEM to generate a small dataset of high quality, which is then combined with PINN for pre-training purposes. The resulting pre-trained model parameters are saved and utilized as the initial state for the PINN model when inverting the material parameters of a new dataset. To assess the accuracy and efficiency of the proposed method during the pre-training phase, validation will be conducted using an elastic-plastic material parameter inversion problem. Subsequently, the inversion of various material parameters, encompassing linear elasticity and elastoplasticity, will be performed by employing the coupling of S-FEM and PINN, both with and without transfer learning.

The contributions of this paper can be summarized as follows:

1. A transfer learning-based coupling of the S-FEM and PINN methods for the inversion of different material parameters is proposed.
2. The proposed approach improves the convergence efficiency of the inversion by at least a factor of two over the method without transfer learning and also provides a degree of improvement in the accuracy of the inversion.
3. The proposed method requires only a small amount of data in the pre-training phase of the model to achieve high accuracy.

This paper is organized as follows. The backgrounds of the S-FEM, PINN, and elastic-plastic problems are presented in Section 2. The PDEs involved in the elastoplastic inversion problem and the specific implementation of our proposed approach are illustrated in Section 3. The results of our proposed method to invert different material parameters are presented in Section 4. The advantages and disadvantages of our proposed method and the future research work are discussed in Section 5. Finally, a summary of the paper is presented.

2. Background

In this section, the basic idea of S-FEM is introduced, highlighting the distinctions between S-FEM and FEM. Additionally, a concise overview of the research progress and fundamental principles of PINN is provided. Finally, a brief introduction is given to the elastic-plasticity problem along with its engineering application background.

2.1. Smoothed Finite Element Method (S-FEM)

The FEM is one of the most widely used numerical methods in solid mechanics. It is a very effective tool for solving complex differential equations or PDEs, especially nonlinear systems of PDEs [39]. Although the FEM is used extensively in engineering science, the demand for greater accuracy and stability in its results has led to a greater focus on the limitations of the traditional FEM. The displacement finite element theory is widely used in commercial finite element software. However, the stiffness matrix of the system obtained based on the displacement finite element method is stiff, which can result in a small displacement solution and a large intrinsic frequency solution [33]. During the solution process, conversions and mappings of local and global coordinate systems occur. Therefore, the problem domain needs to be discretized with only regular grids and not distorted grids to ensure high solution accuracy, as distorted grids can significantly impact the accuracy of the solution [35].

To address the limitations of the FEM, several numerical methods have been proposed, including the S-FEM [40–43]. The S-FEM, proposed by Liu G.R., combines the benefits of the FEM and the mesh-free method. This method can address issues such as mesh distortion and volume locking problems while also improving computational accuracy [35]. The fundamental principle of S-FEM is to divide the integration region into smoothing
subdomains based on the mesh of finite elements. These subdomains are required to follow the rules of being interconnected and not overlapping [35]. S-FEM can be categorized into several types depending on the method of division. These include cell-based S-FEM (CS-FEM) [44,45], node-based S-FEM (NS-FEM) [46], edge-based S-FEM (ES-FEM) [36], face-based S-FEM (FS-FEM) [47], and hybrid S-FEM (HS-FEM) [48]. S-FEM utilizes the finite element background mesh to construct the shape function and perform smoothing strain computation in the smoothing domain constructed on the background mesh. It also uses a gradient-smoothing technique that transforms the area integral into a boundary integral, thereby significantly improving computational accuracy, enhancing the adaptability of low-order cells to mesh distortions, and effectively softening the system stiffness [35]. In contrast to FEM, S-FEM uses linear point interpolation to represent the shape function. Therefore, coordinate transformation is not required, and there is no need to calculate the derivatives of the shape function [35]. Due to the absence of the mapping of shape functions, S-FEM is stable when dealing with irregular meshes. Moreover, S-FEM has high convergence and accuracy because of the softened stiffness matrix.

2.2. Physics-Informed Neural Network (PINN)

PINNs leverage prior knowledge by integrating observational data and mathematical models to efficiently solve both forward and inverse problems of PDEs. Currently, PINNs have demonstrated remarkable success in various mechanics fields, including material mechanics [38], fluid mechanics [49], fracture mechanics [50], and thermodynamics [51]. Several PINN variants have emerged to address different problems, including conservation PINNs (cPINNs) [52], variational PINNs (vPINNs) [53], fractional-order PINNs (fPINNs) [54], and others. Additionally, many researchers have optimized and improved the training process, activation functions, and generalization error of PINNs [55,56]. Furthermore, open-source PINN library packages, such as SciANN [57], DeepXDE [58], and SimNet [59], make it easier to apply PINNs to solve specific problems.

The basic framework of PINN consists of two components: a neural network approximation function and physical information constraints, as illustrated in Figure 1. The input data are approximated using a fully connected neural network to generate predicted values. An automatic differentiation algorithm is used to obtain the residuals of the physical information from the predicted values. These residuals are then incorporated into the loss function as a regular term constraint. The neural network’s weight parameters and deviation vectors are connected and trained using the gradient descent algorithm until the residuals reach the convergence condition, at which point training stops. This process ultimately leads to the solution of the model parameters and the prediction of the results.

Figure 1. Illustration of the principle of PINN.
As shown in Figure 1, \( x_1 \) and \( x_n \) are the inputs of the neural network, \( \sigma \) is the activation function, and \( u_1 \) and \( u_n \) are the outputs of the neural network. The results of the automatic differentiation computation using the neural network are denoted as \( \frac{\partial}{\partial x} \) and \( \frac{\partial}{\partial x} \). \( \text{Loss}_{\text{Data}} \) represents the data-driven partial residuals, and \( \text{Loss}_{\text{PDE}} \) represents the physically constrained partial residuals. \( \varepsilon \) is the threshold of the loss function, \( \text{maxit} \) is the maximum number of training steps, and \( w \) and \( b \) are the weights and biases of the neural network, respectively.

Traditional physical models are used to solve PDEs by giving the initial state, boundary states, and physical parameters at any point. When analytical solutions are not available, numerical methods such as FEM and Finite Difference Method (FDM) are often employed to solve the PDEs. The PINN is a deep neural network-based approach that incorporates physical information into neural networks to approximate the solutions of PDEs. In this method, the model residuals comprise two components: the residuals of the data and the residuals of the physical information, i.e., \( \text{Loss} = \text{Loss}_{\text{Data}} + \text{Loss}_{\text{PDE}} \).

\[
\text{Loss}_{\text{Data}} = \frac{1}{N_{\text{Data}}} \sum_{i=1}^{N_{\text{Data}}} \left| u^N(x_i, y_i) - u_i \right|^2 \quad (1)
\]

Equation (1) represents the data-driven partial residuals, where \( \{u^N(x_i, y_i)\}_{i=1}^{N_{\text{Data}}} \) denotes the result obtained from the neural network prediction; \( \{u_i\}_{i=1}^{N_{\text{Data}}} \) is the known data, including initial and boundary conditions as well as the measured and synthetic data; and \( N_{\text{Data}} \) is the number of known data points.

\[
\text{Loss}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left| r(x_i, y_i) \right|^2 \quad (2)
\]

Equation (2) represents the physically constrained partial residuals, where \( \{r(x_i, y_i)\}_{i=1}^{N_{\text{PDE}}} \) is the PDE’s residual and \( N_{\text{PDE}} \) is the number of configuration points of the PINN.

2.3. Elastoplastic Problems

Elastoplastic mechanics is an important branch of deformable solid mechanics, which is the study of the stress, strain, and displacement of deformable materials and their distribution laws when subjected to external loads, temperature changes, and other factors [60,61]. Most objects generally go through three stages from stress to destruction: elasticity, plasticity, and destruction. The solution of elastic-plastic mechanics problems is crucial in many fields of research, such as civil engineering [62], mechanical engineering [63], aerospace engineering [64], and materials engineering [65].

The basic equations of elastoplastic mechanics need to be established in terms of their geometry, kinematics, and physics [60,61]. Firstly, since elastoplastic mechanics assumes that the object is continuous, all adjacent small units are interconnected during deformation, and the coordination conditions of deformation can be obtained by studying the relationship between displacement and strain [60,61]. The mathematical expressions reflecting the continuous law of deformation are the geometric equations and displacement boundary conditions. Secondly, in the elastic-plastic problem, the object should not only be in equilibrium as a whole but also locally in equilibrium, and the mathematical equations reflecting this law are equilibrium differential equations and load boundary conditions [60,61]. These two types of equations are independent of the mechanical properties of the material and are universal. In physics, it is necessary to establish the relationship between stress and strain or stress and strain increments, and this relationship is called the constitutive relationship. The constitutive relationship describes the mechanical properties of a material in different environments, and the study of the constitutive relationship is crucial in elastoplastic mechanics [60,61].

When tackling an elastoplastic statics problem, several essential elements need to be provided to determine the stresses, strains, and displacements of the object. These include the shape of the object, constitutive relationships, and the physical-mechanical
parameters for each component of the object’s material, as well as the load and displacement boundary conditions to which the object is subjected [60,61]. In engineering problems, determining the constitutive parameters, physical-mechanical parameters, and boundary conditions of materials can often prove challenging [66,67]. The concept of inverse problem solving offers a means to determine these parameters. Through the observation of strain or displacement data from the structure, the inverse analysis method in elastic-plastic mechanics can be utilized to infer crucial information such as the constitutive parameters, physical-mechanical parameters, and boundary conditions of the material [66,67].

3. Methods

In this section, the relevant PDEs associated with the elastoplastic mechanic problem are initially presented, highlighting the parameters that necessitate inversion. Subsequently, the implementation process of our proposed approach, which couples S-FEM and PINN based on transfer learning, is described.

3.1. Governing Equations and Parameters

The equations involved in the elastic-plastic problem mainly include equilibrium differential equations, geometric equations, and physical equations. Since the equilibrium differential equations and geometric equations are independent of the material properties, these two types of equations for the linear elastic and elastoplastic problems are consistent, see Equations (3)–(11). Equations (3)–(5) are equilibrium differential equations, and Equations (6)–(11) are geometric equations.

$$
\sigma_{xx,x} + \sigma_{yx,y} + \sigma_{zx,z} + f_x = 0 \quad (3)
$$

$$
\sigma_{xy,x} + \sigma_{yy,y} + \sigma_{zy,z} + f_y = 0 \quad (4)
$$

$$
\sigma_{xz,x} + \sigma_{yz,y} + \sigma_{zz,z} + f_z = 0 \quad (5)
$$

where \( \sigma_{ij} \) \( (i, j = x, y, z) \) represents the partial derivative of the stress tensor and \( f_i \) \( (i = x, y, z) \) represents the volume force.

$$
\varepsilon_{xx} = u_{x,x} \quad (6)
$$

$$
\varepsilon_{yy} = u_{y,y} \quad (7)
$$

$$
\varepsilon_{zz} = u_{z,z} \quad (8)
$$

$$
\varepsilon_{xy} = \frac{1}{2}(u_{x,y} + u_{y,x}) \quad (9)
$$

$$
\varepsilon_{yz} = \frac{1}{2}(u_{y,z} + u_{z,y}) \quad (10)
$$

$$
\varepsilon_{zx} = \frac{1}{2}(u_{z,x} + u_{x,z}) \quad (11)
$$

where \( \varepsilon_{ij} \) \( (i, j = x, y, z) \) represents the strain tensor and \( u_{i,j} \) \( (i, j = x, y, z) \) represents the partial derivative of the displacement. \( \sigma_m = (\sigma_{xx} + \sigma_{yy} + \sigma_{zz})/3 \) is the mean stress and \( \varepsilon_m = (\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz})/3 \) is the mean strain.

For the linear elastic plane strain problem, where \( \varepsilon_{zz} = \varepsilon_{yz} = \varepsilon_{xz} = 0 \), the physical equations are Equations (12)–(14).

$$
(\lambda + 2\mu)\varepsilon_{xx} + \lambda\varepsilon_{yy} - \sigma_{xx} = 0 \quad (12)
$$
When \( \sigma \) is the equivalent stress, \( \sigma_{ij} \) \((i, j = x, y, z)\) represents the stress tensor, and \( \sigma_{s} \) is the yield stress of the material, which is the first material parameter to be inverted in an inversion process.

In the linear elastic plane problem, the PDE residuals and the data residuals can be expressed as Equations (15) and (16).

\[
\text{Loss}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left( (\sigma_{xx}^i + \sigma_{yy}^i + f_x^i)^2 + (\sigma_{xy}^i + \sigma_{yx}^i + f_y^i)^2 \right)
\]

\[
\text{Loss}_{\text{Data}} = \frac{1}{N_{\text{Data}}} \sum_{i=1}^{N_{\text{Data}}} \left( |u_x^i - u_x^i|^2 + |u_y^i - u_y^i|^2 + |\sigma_{xx}^i - \sigma_{xx}^i|^2 \right)
\]

where the physical quantities without an asterisk superscript represent the predicted results obtained from the PINN or are computed based on these predicted results. Moreover, the physical quantities with an asterisk superscript denote the labeled data.

Before establishing the physical equations of the elastic-plastic problem, it is necessary to determine the yield criterion. The yield criterion serves as a condition to discern whether the material is in an elastic or plastic state. In this paper, the von Mises criterion, which is widely applicable to metallic materials, is selected as the yield criterion for the elastoplastic problem (see Equation (17)).

\[
\sigma = \frac{1}{\sqrt{2}} \sqrt{(|\sigma_{xx} - \sigma_{yy}|^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + 6(\sigma_{xy}^2 + \sigma_{yx}^2 + \sigma_{zz}^2)} = \sigma_s
\]

where \( \sigma \) is the equivalent stress, \( \sigma_{ij} \) \((i, j = x, y, z)\) represents the stress tensor, and \( \sigma_{s} \) is the yield stress of the material, which is the first material parameter to be inverted in an elastoplastic problem. The material is in the elastic state when \( \sigma < \sigma_s \) and in the plastic state when \( \sigma \geq \sigma_s \).

For small elastic-plastic deformation problems, the stress–strain relationship is described by the following equations:

\[
\varepsilon_{xx} - \varepsilon_m = \frac{3 \varepsilon}{2 \sigma_s} (\sigma_{xx} - \sigma_m)
\]

\[
\varepsilon_{yy} - \varepsilon_m = \frac{3 \varepsilon}{2 \sigma_s} (\sigma_{yy} - \sigma_m)
\]

\[
\varepsilon_{zz} - \varepsilon_m = \frac{3 \varepsilon}{2 \sigma_s} (\sigma_{zz} - \sigma_m)
\]
\[
\varepsilon_{xy} = \frac{3}{2} \bar{\sigma} \varepsilon_{xy}
\]
(21)
\[
\varepsilon_{yz} = \frac{3}{2} \bar{\sigma} \varepsilon_{yz}
\]
(22)
\[
\varepsilon_{zx} = \frac{3}{2} \bar{\sigma} \varepsilon_{zx}
\]
(23)

where \( \bar{\sigma} \) is the equivalent strain. According to the power-hardening constitutive relationship, \( \bar{\sigma} \) and \( \varepsilon \) can be expressed as Equation (24) in this paper. In the elastic phase, the stress–strain relationship follows Hooke’s law and is linear. However, when the material transitions into the plastic phase, the stress–strain relationship becomes nonlinear, characterized by a power exponent relationship between stress and strain.

\[
\bar{\sigma} = \begin{cases} 
E\varepsilon(0 \leq \varepsilon \leq \varepsilon_s) \\
B(\varepsilon - \varepsilon_0)^m(\varepsilon \geq \varepsilon_s) 
\end{cases}
\]
(24)

where \( E \) is Young’s modulus, the second material parameter to be inverted in the elastoplastic problem. \( \varepsilon_0 = \varepsilon_s(1 - m) \), \( B = \frac{E\varepsilon_0}{(\varepsilon_0 - \varepsilon_s)} \), \( m \) is the power hardening index, and \( \varepsilon_s \) is the equivalent strain corresponding to the yield stress \( \sigma_y \) of the material.

In the elastoplastic plane stress problem, where \( \sigma_{zz} = \sigma_{yz} = \sigma_{xx} = 0 \), the PDE residuals and data residuals can be expressed as Equations (25) and (26), respectively. For a three-dimensional (3D) elastic-plastic problem, the PDE residuals and data residuals can be expressed as Equations (27) and (28), respectively.

\[
\text{Loss}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left( \left| \varepsilon_{xx,x}^i + \sigma_{xy,y}^i + f_{xx}^i \right|^2 + \left| \varepsilon_{xy,y}^i + \sigma_{yy,y}^i + f_{yy}^i \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{xx}^i - \bar{\epsilon}_m^i) - (\varepsilon_{xx}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{yy}^i - \bar{\epsilon}_m^i) - (\varepsilon_{yy}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{zx}^i - \bar{\epsilon}_m^i) - (\varepsilon_{zx}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{zy}^i - \bar{\epsilon}_m^i) - (\varepsilon_{zy}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| (\sigma_{xx}^i/E) - \left( (\sigma_{xx}^i/B^{(1/m)}) + \varepsilon_0 \right) \right|^2 \right)
\]
(25)

\[
\text{Loss}_{\text{Data}} = \frac{1}{N_{\text{Data}}} \sum_{i=1}^{N_{\text{Data}}} \left( \left| u_x^i - u_{xx}^i \right|^2 + \left| u_y^i - u_{yy}^i \right|^2 
\right.
\]
\[
\left. + \left| \sigma_{xx}^i - \bar{\sigma}_{xx}^i \right|^2 + \left| \sigma_{yy}^i - \bar{\sigma}_{yy}^i \right|^2 + \left| \sigma_{xy}^i - \bar{\sigma}_{xy}^i \right|^2 \right)
\]
(26)

\[
\text{Loss}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left( \left| \varepsilon_{xx,x}^i + \sigma_{xy,y}^i + \sigma_{zx,z}^i + f_{xx}^i \right|^2 + \left| \varepsilon_{xy,y}^i + \sigma_{yy,y}^i + \sigma_{yz,z}^i + f_{yy}^i \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{xx}^i - \bar{\epsilon}_m^i) - (\varepsilon_{xx}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{yy}^i - \bar{\epsilon}_m^i) - (\varepsilon_{yy}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{zx}^i - \bar{\epsilon}_m^i) - (\varepsilon_{zx}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| 3\varepsilon/(2\bar{\sigma})(\sigma_{zy}^i - \bar{\epsilon}_m^i) - (\varepsilon_{zy}^i - \bar{\epsilon}_m^i) \right|^2 
\right.
\]
\[
+ \left. \left| (\sigma_{xx}^i/E) - \left( (\sigma_{xx}^i/B^{(1/m)}) + \varepsilon_0 \right) \right|^2 \right)
\]
(27)
\[
\text{Loss}_{\text{Data}} = \frac{1}{N_{\text{Data}}} \sum_{i=1}^{N_{\text{Data}}} \left( |u_i^x - u_i^{*x}|^2 + |u_i^y - u_i^{*y}|^2 + |u_i^z - u_i^{*z}|^2 + \right. \\
\left. |\sigma_i^{xx} - \sigma_i^{*xx}|^2 + |\sigma_i^{yy} - \sigma_i^{*yy}|^2 + |\sigma_i^{zz} - \sigma_i^{*zz}|^2 + \right. \\
\left. |\sigma_i^{xy} - \sigma_i^{*xy}|^2 + |\sigma_i^{yz} - \sigma_i^{*yz}|^2 + |\sigma_i^{zx} - \sigma_i^{*zx}|^2 \right)
\]  

(28)

where physical quantities without asterisk superscripts represent predictions obtained from PINN or calculations based on these predictions. In addition, physical quantities with asterisk superscripts represent labeled data.

3.2. Transfer Learning-Based Coupling of S-FEM and PINN

In our approach, the S-FEM is employed to generate high-quality datasets, while the PINN is used to invert elastoplastic material parameters. Additionally, transfer learning is incorporated to enable the inversion of different elastoplastic material parameters. One advantage of the S-FEM is its ability to synthesize data in low-order linear cells, achieving higher accuracy compared to the FEM in bilinear cells with the same nodes. This is achieved through the strain smoothing technique. Importantly, the S-FEM enables the synthesis of high-quality datasets without incurring any additional computational costs. By coupling these high-quality datasets with PINN, the accuracy of the elastoplastic inversion process is further improved. An S-FEM coupled with PINN is employed to invert a set of elastoplastic material parameters, which are subsequently utilized as a pre-trained model for transfer learning. The neural network parameters (weights, biases, etc.) of this pre-trained model are saved and utilized to initialize the neural network parameters of the model for a new dataset. By initializing the neural network with these pre-trained parameters instead of random initialization, the model can expedite its approach towards the optimal solution and reduce the required number of iterations. This accelerated convergence significantly facilitates the inversion process for different elastoplastic material parameters.

The transfer learning-based coupling of S-FEM and PINN for the inversion of elastoplastic material parameters consists of two phases: pre-training and transfer learning. The workflow of this method is illustrated in Figure 2.

**Figure 2.** Flow chart of transfer learning-based coupling of S-FEM and PINN for the inversion of various material parameters.
The implementation process of the pre-training phase of 2D elastoplastic parameters is shown in Figure 3.

1. The problem domain is discretized using triangular elements to obtain the spatial coordinates of the nodes. These nodes are directly used as inputs for the PINN model and as the training configuration points. A smoothing domain is constructed by connecting the nodes of adjacent unit shape centroids using the edges of the triangle as the unit.

2. The elastic-plastic forward problem is solved using S-FEM, which involves the following steps:
   a. The creation of displacement fields through the construction of shape functions;
   b. The construction of smoothing strain fields. In the case of triangular elements, it is only necessary to perform the line integral directly on the boundary of the smoothing domain;
   c. The creation of the system equation set, including the assembly stiffness matrix and the load vector. This process involves a simple summation operation for the parameters associated with the smoothing domain in the S-FEM;
   d. The imposition of boundary conditions and solving of the system equations to obtain the displacement solutions;
   e. The reconstruction of the strain field based on the obtained displacement solutions. The stresses of the equivalent nodes are then obtained in the smoothing domain using the weighted average method. The continuous stress field in the problem domain is obtained using the shape function interpolation method. The equivalent node stresses and displacements are embedded in the loss function of PINN as label data.

3. Construction of the PINN. A fully connected neural network is constructed using DeepXDE, the Python library package developed by PINN. The size of the neural network is predetermined, and its parameters, i.e., weights (w) and bias (b), are initialized using the Glorot Uniform method, which is a technique for uniformly distributing initialization values. The activation function of the neural network is set to the hyperbolic tangent function (tanh).

![S-FEM](image1)

![PINN](image2)

**Figure 3.** Pre-training process of transfer learning-based coupling of S-FEM and PINN for the inversion of 2D elastic-plastic material parameters.
In the S-FEM and PINN coupling method, the neural network takes the normalized values of the spatial coordinates \( x \) and \( y \) of the nodes from the discrete problem domain of the S-FEM as input. The outputs of the neural network are the displacements \( u_x \) and \( u_y \) and stresses \( \sigma_{xx} \), \( \sigma_{yy} \), and \( \sigma_{xy} \) of the nodes in the problem domain. To invert the parameters \( E \) and \( \sigma \) for the elastic-plastic problem, they are assigned an initial value and are used as trainable parameters for the neural network, similar to the weights and biases. The PDEs of the elastic-plastic problem are defined, including the equilibrium differential equations, the geometric equations, and the physical equations. The output of the neural network is processed using the automatic differentiation method, and the resulting values are then substituted into the PDEs to obtain the PDE residuals. The labeled data are then used to calculate the residuals of the data-driven part, which are compared to the displacements and stresses output by the neural network. The sum of the PDE residuals and the data residuals is then used as the loss function of the neural network.

4) Training the PINN. The gradient descent algorithm Adam is used to optimize the neural network parameters and minimize the loss function. The number of training steps or the threshold value of the loss function is set to determine when the training is complete.

5) After training the PINN, the parameters of the pre-trained model are saved, including the weights, biases, and inverse material parameters.

The transfer learning phase consists of the following steps: (1) synthesizing a new dataset using S-FEM, (2) directly loading the PINN model constructed in the pre-training phase, (3) initializing the newly loaded PINN model with the neural network parameters saved in the pre-training phase, (4) training the PINN model until convergence is reached, and (5) obtaining the inversion parameter values. As shown in Figure 4, the principle of the inversion of different elastoplastic material parameters based on transfer learning-coupled S-FEM and PINN is illustrated.

Figure 4. Illustration of the principle of the inversion of different elastoplastic material parameters based on transfer learning-coupled S-FEM and PINN.

4. Results and Analysis

In this section, the effectiveness of coupling S-FEM and PINN without transfer learning is demonstrated through a two-dimensional elastic-plastic problem. The computational results obtained from this method are compared with those obtained from coupling traditional FEM and PINN. Subsequently, the transfer learning-based coupled S-FEM and PINN method is employed to invert different material parameters. The accuracy and efficiency
of coupling S-FEM and PINN without transfer learning when solving the same problem are compared. The hardware and software details utilized in this section are provided in Table 1.

Table 1. Environment configurations.

<table>
<thead>
<tr>
<th>Environment Configurations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Windows 11 Professional</td>
</tr>
<tr>
<td>Deep learning framework</td>
<td>TensorFlow2.9-GPU</td>
</tr>
<tr>
<td>Dependent library</td>
<td>DeepXDE, Numpy, Pandas, etc.</td>
</tr>
<tr>
<td>CPU</td>
<td>AMD Ryzen 7 6800H with Radeon Graphics</td>
</tr>
<tr>
<td>CPU RAM (GB)</td>
<td>16</td>
</tr>
<tr>
<td>CPU Frequency (GHz)</td>
<td>3.2</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA GeForce RTX3060 Laptop GPU</td>
</tr>
</tbody>
</table>

4.1. Coupling S-FEM and PINN for the Inversion of Elastic-Plastic Material Parameters without Transfer Learning

In this section, the material parameters of an elastoplastic cantilever beam, specifically Young’s modulus $E$ and yield stress $\sigma_s$, are inverted using the coupling of S-FEM and PINN without transfer learning. The obtained results are then compared with those obtained through the conventional coupling of FEM and PINN to evaluate the performance of the coupled S-FEM and PINN in material parameter inversion.

A cantilever beam subjected to a uniform load, as shown in Figure 5a, is analyzed here as an elastoplastic plane stress problem. The material follows the power-hardening stress–strain relationship given by Equation (24), where the power-hardening index is denoted by $m = 0.1$. The left boundary of the cantilever beam is fixed, and the top boundary is subjected to a uniform load $q = -0.005 \text{ N/mm}^2$. The material parameters for the beam are given by Young’s modulus $E = 2.0 \text{ MPa}$ and yield stress $\sigma_s = 0.235 \text{ MPa}$. In this study, we consider the material parameters $E$ and $\sigma_s$ as unknown parameters and employ the coupling of S-FEM and PINN for inversion.

![Figure 5](image_url)

**Figure 5.** Example of an elastic-plastic cantilever beam: (a) geometry and boundary conditions of the elastic-plastic cantilever beam, (b) distribution of sampling points obtained from the S-FEM discrete problem domain.

The problem domain is discretized using triangular elements to obtain the spatial coordinates of 105 nodes, as depicted in Figure 5b. The edge-based smoothing domain is then constructed, and the displacements and stresses of the 105 nodes are obtained using
the S-FEM. The coordinates, displacements, and stresses of these 105 nodes are normalized using the z-score method. The normalized node coordinates are utilized as input to the PINN model, while the normalized node displacements and stresses are incorporated into the loss function of PINN. The PINN model is then trained to obtain the material parameter values. The neural network parameters of the PINN model are specified in Table 2. The initial value of the unknown parameter $E$ is set to 2.0 MPa, while the initial value of the unknown parameter $\sigma_s$ is set to 0.1 MPa.

Table 2. The neural network parameters for the elastoplastic parameter inversion.

<table>
<thead>
<tr>
<th>Neural Network Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layers</td>
<td>[20] × 5</td>
</tr>
<tr>
<td>Activation Functions</td>
<td>tanh</td>
</tr>
<tr>
<td>Initial Learning Rate</td>
<td>0.005</td>
</tr>
<tr>
<td>Learning Rate Decay</td>
<td>0.5/500 epochs</td>
</tr>
<tr>
<td>Epochs</td>
<td>10,000</td>
</tr>
<tr>
<td>Initializer</td>
<td>Glorot uniform</td>
</tr>
</tbody>
</table>

The results of the inversion obtained after 10,000 training steps are presented in Figure 6. It can be observed from the results in Figure 6 that both the values of the parameters to be inverted converge after about 5000 training steps. The inversion results for Young’s modulus $E$ and yield stress $\sigma_s$ are 2.001 MPa and 0.234 MPa, respectively. The relative error of Young’s modulus $E$ from the true value is only 0.05%, while the relative error of yield stress $\sigma_s$ from the true value is 0.638%. The results demonstrate that the coupling of S-FEM and PINN can achieve computational errors of less than 1% using only 105 data nodes. The output displacement and stress of the neural network are visualized in Figure 7, and their $L_2$ relative error is computed. The results reveal that the neural network output fits the labeled data well, with a maximum $L_2$ relative error of 5.2%.

![Figure 6](image_url)

**Figure 6.** Convergence process of coupling S-FEM and PINN for inversion of elastic-plastic material parameters: (a) Young’s modulus $E$, (b) yield stress $\sigma_s$.

Furthermore, a comparison is conducted between the results obtained from the coupling of S-FEM and PINN and those obtained from the coupling of FEM and PINN. In the latter method, similar to the former method, the problem domain is first discretized using triangular elements to determine the spatial coordinates of 105 nodes. Subsequently, the FEM is employed to calculate the displacements and stresses at these 105 nodes. The coordinates, displacements, and stresses of these 105 nodes are normalized using the z-score method.
method. The normalized node coordinates are used as input to the PINN, while the node displacements and stresses are embedded in the PINN’s loss function to obtain the material parameter values. The neural network parameters of PINN are set as shown in Table 2. The initial value of the unknown parameter $E$ is set to 2.0 MPa, and the initial value of the unknown parameter $\sigma_s$ is set to 0.1 MPa.

As depicted in Figure 8, the proposed coupled S-FEM and PINN approach yields material parameter values that are significantly closer to the true values compared to the inversion results obtained using the couple FEM and PINN. The relative error accuracies of both methods are compared in Table 3, which reveals that the computational accuracy of S-FEM coupled with PINN is improved by about an order of magnitude over FEM coupled with PINN.

![Figure 7. The output of neural network in the inversion of elastic-plastic material parameters.](image1)

![Figure 8. Convergence process of different methods for the inversion of elastic-plastic material parameters: (a) Young’s modulus $E$, (b) yield stress $\sigma_s$.](image2)
Table 3. Relative errors of elastoplastic material parameters obtained by inversion of different methods and true values.

<table>
<thead>
<tr>
<th>Parameters (MPa)</th>
<th>Inverse Results (MPa)</th>
<th>Relative Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FEM-PINN</td>
<td>S-FEM-PINN</td>
</tr>
<tr>
<td>E</td>
<td>1.974</td>
<td>2.001</td>
</tr>
<tr>
<td>$\sigma_s$</td>
<td>0.2219</td>
<td>0.2335</td>
</tr>
</tbody>
</table>

In solving the elastic-plastic inverse problem, achieving computational accuracy while considering computational efficiency is of utmost importance. Thus, a comparison was conducted between the computational time of the FEM couple with PINN and the S-FEM couple with PINN methods. The time required to solve the elastic-plastic inversion problem using both coupling methods comprises two main components: (1) synthetic data time and (2) PINN inversion time. The time taken for the inversion algorithm to reach convergence was recorded. The results indicated that the S-FEM coupled with PINN method achieved convergence for both material parameters in 65.92 s, whereas the FEM coupled with PINN method took 71.91 s.

4.2. Coupling S-FEM and PINN for the Inversion of Different Material Parameters with Transfer Learning

In this section, the performance of the transfer learning-based coupling of S-FEM and PINN when inverting different material parameters is investigated through three examples. In Section 4.2.1, the accuracy and efficiency of the transfer learning-based coupling of S-FEM and PINN when inverting various linear elastic plane strain material parameters are demonstrated. In Section 4.2.2, the accuracy and efficiency of the transfer learning-based coupling of S-FEM and PINN when inverting material parameters for elastoplastic plane stress problems are presented. Finally, in Section 4.2.3, the accuracy and efficiency of the coupling of S-FEM and PINN based on transfer learning when inverting different 3D elastoplastic material parameters are demonstrated.

4.2.1. Inversion of Different Parameters on a 2D Elastic Plate

An elastic plate, shown in Figure 9a, is considered here as a plane strain problem. The elastic plate has a fixed bottom, left and right boundary conditions denoted as $\sigma_{xx} = 0$ and $u_y = 0$, and a top boundary condition denoted as $u_x = 0$. The plate is subjected to a varying distributed load denoted as $q = (\lambda + 2\mu) \sin(\pi x)$. The material parameters of the elastic plate satisfy Equations (12)–(14), where the Lamé parameter is $\lambda = 1.0$ MPa, $\mu = \{0.1, 0.25, 0.5, 0.75, 1.0\}$ MPa, and $\mu$ is inverted as an unknown parameter. First, the S-FEM synthetic dataset with material parameters $\lambda = 1.0$ MPa and $\mu = 0.5$ MPa was created. The S-FEM data were then coupled with the PINN for pre-training. The neural network parameters of PINN are set as shown in Table 4. The initial value of the unknown parameter $\mu$ is set to 1.0 MPa.

Table 4. The neural network parameters for the linear elastic parameter inversion.

<table>
<thead>
<tr>
<th>Neural Network Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layers</td>
<td>[40] × 4</td>
</tr>
<tr>
<td>Activation Functions</td>
<td>tanh</td>
</tr>
<tr>
<td>Initial Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Epochs</td>
<td>10,000</td>
</tr>
<tr>
<td>Initializer</td>
<td>Glorot uniform</td>
</tr>
</tbody>
</table>
According to the geometric conditions and boundary conditions of the elastic plate shown in Figure 9a, the S-FEM is used to discretize the problem domain into triangular elements, and the coordinates of 289 nodes are obtained. The parameter combination of $\lambda = 1.0$ MPa and $\mu = 0.5$ MPa is then substituted into the S-FEM to compute the displacements and stresses of the nodes, which are used as the known data for pre-training the PINN. The pre-trained neural network parameters, including the weights and biases of the neural network, as well as the inversion parameters $\mu$, are saved to enable their transfer for the inversion of different material parameters. The value of parameter $\mu$ is modified to 0.1, 0.25, 0.75, and 1.0 MPa for four different parameter groups while keeping the remaining material parameters and loading conditions constant. Subsequently, four groups of data are synthesized using S-FEM. These four datasets are utilized to train the PINN for parameter inversion in two different approaches. The first method involved randomly initializing the neural network (without transfer learning), while the second method involved initializing the neural network using the saved pre-trained model parameters for the inversion of material parameters (with transfer learning).

Figure 10 illustrates the convergence process of the four sets of parameters during parameter inversion, comparing the results with and without the utilization of transfer learning. Table 5 presents a comprehensive comparison of the relative errors between the material parameters obtained with and without transfer learning, relative to their true values, for the four parameter sets. The comparison results indicate that the use of transfer learning significantly enhances the convergence speed and accuracy of the inversion. Figure 11 displays a comparison of the convergence of the loss functions for coupling S-FEM and PINN with and without transfer learning. The outcomes reveal that the coupling of S-FEM and PINN with transfer learning converges better and faster on the new data set.

Table 5. Relative errors of inversion results obtained with and without transfer learning compared to the true values in a 2D elastic plate.

<table>
<thead>
<tr>
<th>Parameters (MPa)</th>
<th>Inversion Results (MPa)</th>
<th>Relative Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without Transfer Learning</td>
<td>With Transfer Learning</td>
</tr>
<tr>
<td>$\mu = 0.1$</td>
<td>0.0987</td>
<td>0.0986</td>
</tr>
<tr>
<td>$\mu = 0.25$</td>
<td>0.2469</td>
<td>0.2471</td>
</tr>
<tr>
<td>$\mu = 0.75$</td>
<td>0.7419</td>
<td>0.7416</td>
</tr>
<tr>
<td>$\mu = 1.0$</td>
<td>0.9829</td>
<td>0.9885</td>
</tr>
</tbody>
</table>
Figure 10. Results of inversion parameters with and without transfer learning after 10,000 epochs for: (a) $\mu = 0.1$ MPa, (b) $\mu = 0.25$ MPa, (c) $\mu = 0.75$ MPa, (d) $\mu = 1.0$ MPa.

Figure 11. Loss function convergence of coupling S-FEM and PINN with and without transfer learning: (a) $\mu = 0.1$ MPa, (b) $\mu = 0.25$ MPa, (c) $\mu = 0.75$ MPa, (d) $\mu = 1.0$ MPa.

The convergence times for the parameter inversions of the four sets are recorded and depicted in Figure 12. The results demonstrate that the convergence time of inversion is reduced by at least half after using transfer learning. In the experiments of $\mu = 1.0$ MPa, the convergence time of parameter inversion is found to be consistent with and without transfer learning. The reason behind this is that the initial value of the parameter to be inverted, $\mu$, was set to 1.0 MPa during the random initialization of the training model. As a result, the inversion of the parameter without using transfer learning still achieves good performance. It has been demonstrated that the selection of initial values for the unknown parameters during parameter inversion has a significant impact on the inversion results, and appropriately chosen initial values can greatly improve computational efficiency. The
The concept behind transfer learning is similar, as it involves using the pre-trained model parameters to initialize the neural network parameters of the new model.

Figure 12. Comparison of inversion convergence time with and without transfer learning in a 2D elastic plate.

4.2.2. Inversion of Different Parameters on a 2D Elastic-Plastic Beam

Considering the geometric and boundary conditions of the elastoplastic problem shown in Figure 5a, the results of the coupling S-FEM and PINN inversion parameter combinations $E = 2.0$ MPa and $\sigma_s = 0.235$ MPa in Section 4.1 are used as the pre-trained model results. Four sets of data are generated using S-FEM by altering the values of parameter $E$ to 1.6, 1.8, 2.2, and 2.4 MPa while keeping the remaining material parameters and loading conditions constant. PINN is employed for parameter inversion using these four datasets, both with and without transfer learning. In one case, the neural network is randomly initialized (without transfer learning), while in the other case, the neural network is initialized using the saved pre-trained model parameters (with transfer learning).

In Figure 13, a comparison of the convergence processes of the four sets of parameters with and without transfer learning inversion is presented. In Table 6, a detailed comparison of the relative errors of the material parameters obtained with and without transfer learning inversion to their true values for the four sets of parameters is provided. The comparison results indicate that the convergence of parameter inversion is significantly faster and more accurate when transfer learning is employed. Additionally, the convergence time of the parameter inversion for the four sets was recorded and compared in Figure 14. The results demonstrate that the use of transfer learning leads to a reduction in convergence time by at least 50%.

Table 6. Relative errors of inversion results obtained with and without transfer learning compared to the true values in a 2D elastic-plastic beam.

<table>
<thead>
<tr>
<th>Parameters (MPa)</th>
<th>Inversion Results (MPa)</th>
<th>Relative Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without Transfer Learning</td>
<td>With Transfer Learning</td>
</tr>
<tr>
<td>$E = 1.6$</td>
<td>1.5801</td>
<td>1.5995</td>
</tr>
<tr>
<td>$E = 1.8$</td>
<td>1.7977</td>
<td>1.7999</td>
</tr>
<tr>
<td>$E = 2.2$</td>
<td>2.1446</td>
<td>2.1962</td>
</tr>
<tr>
<td>$E = 2.4$</td>
<td>2.3882</td>
<td>2.3834</td>
</tr>
</tbody>
</table>
Figure 13. Results of inversion parameters with and without transfer learning after 10,000 epochs for: (a) $E = 1.6$ MPa, (b) $E = 1.8$ MPa, (c) $E = 2.2$ MPa, (d) $E = 2.4$ MPa.

Figure 14. Comparison of inversion convergence time with and without transfer learning in a 2D elastic-plastic beam.

4.2.3. Inversion of Different Parameters on a 3D Elastic-Plastic Beam

A 3D elastic-plastic cantilever beam configuration is illustrated in Figure 15, where the left end is fixed and the top is subjected to a uniform load of $q = -0.001$ N/mm$^2$. The material properties of the 3D elastoplastic cantilever beam are defined by Equation (24), which describes a power-hardening stress–strain relationship with a power-hardening exponent of $m = 0.1$. The material parameters are Young’s modulus $E = \{1.5, 1.75, 2.0, 2.25, 2.5\}$ MPa, and the yield stress is $\sigma_y = 0.235$ MPa; these are treated as unknown variables and determined through the inversion process using S-FEM coupled with PINN. First, the S-FEM synthetic dataset with material parameters $E = 2.0$ MPa and $\sigma_y = 0.235$ MPa was created. The S-FEM data were then coupled with the PINN for pre-training. The specific
neural network parameters of PINN are outlined in Table 2. The initial values for the unknown parameters $E$ and $\sigma_s$ are set to 1.0 MPa and 0.1 MPa, respectively.

Figure 15. Geometry and boundary conditions of the 3D elastic-plastic beam.

According to the geometric conditions and boundary conditions of the elastoplastic cantilever beam shown in Figure 15, the coordinate information of 458 nodes is obtained by S-FEM to discretize the problem domain. The displacements and stresses of the nodes were calculated by substituting the parameter combination $E = 2.0$ MPa and $\sigma_s = 0.235$ MPa into S-FEM and coupling it with PINN for pre-training. Four groups of data were synthesized using S-FEM by changing the data of parameter $E$ to 1.5, 1.75, 2.25, and 2.5 MPa, while keeping the remaining material parameters and loading conditions unchanged. The four datasets were directly coupled to PINN for parameter inversion by randomly initializing the neural network (without transfer learning) and for the inversion of material parameters after initializing the neural network using the saved pre-trained model parameters (with transfer learning).

The convergence processes of the inversion with and without transfer learning for the four sets of parameters are compared in Figure 16. The relative errors of the four sets of parameters obtained by inversion with and without transfer learning are compared with the true values in Table 7. The comparison results show that the convergence of the inversion is faster and the accuracy of the inversion can be guaranteed after using transfer learning. The convergence times of the four sets of parameter inversions are recorded and compared in Figure 17. The results show that the convergence time of the inversions is reduced by about 50% on average after using transfer learning.

Table 7. Relative errors of inversion results obtained with and without transfer learning compared to the true values in a 3D elastic-plastic beam.

<table>
<thead>
<tr>
<th>Parameters (MPa)</th>
<th>Inversion Results (MPa)</th>
<th>Relative Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without Transfer Learning</td>
<td>With Transfer Learning</td>
</tr>
<tr>
<td>$E = 1.5$</td>
<td>1.540</td>
<td>1.538</td>
</tr>
<tr>
<td>$E = 1.75$</td>
<td>1.778</td>
<td>1.790</td>
</tr>
<tr>
<td>$E = 2.25$</td>
<td>2.312</td>
<td>2.289</td>
</tr>
<tr>
<td>$E = 2.5$</td>
<td>2.531</td>
<td>2.536</td>
</tr>
</tbody>
</table>
Figure 16. Results of inversion parameters with and without transfer learning after 10,000 epochs for: (a) $E = 1.5$ MPa, (b) $E = 1.75$ MPa, (c) $E = 2.25$ MPa, (d) $E = 2.5$ MPa.

Figure 17. Comparison of inversion convergence time with and without transfer learning in a 3D elastic-plastic beam.

5. Discussion

The transfer learning-based coupling of S-FEM and PINN proposed in this paper has the advantage of achieving high computational accuracy using only a small dataset in the pre-training phase. This is particularly useful in practical engineering applications where obtaining a large amount of monitoring data can be difficult due to the high measurement cost or measurement difficulty. The proposed method is well-suited for the inversion of material parameters for small datasets. The use of coupling S-FEM and PINN to pre-train the model allows for higher computational accuracy and efficiency compared to using traditional FEM and PINN to pre-train the model. The robustness of PINN to noise in the data ensures the validity of the model even in the presence of errors. However, the
findings of this study demonstrate that coupling PINN with high-quality synthetic data can lead to a significant improvement in computational accuracy. The strain smoothing technique utilized in the S-FEM enables the synthesis of data in low-order linear elements with greater computational accuracy compared to the FEM in bilinear elements [35]. By coupling S-FEM with PINN, we were able to enhance the accuracy of the computational results for the elastic-plastic inverse problem without increasing the computational cost.

The computational results presented in Section 4.2 demonstrate that our proposed method, incorporating transfer learning, exhibits a minimum twofold increase in computational efficiency compared to the approach without transfer learning. By employing coupled S-FEM with PINN during the pre-training phase, our method successfully mitigates computational errors to approximately 2%. Although the improvement in computational accuracy of our method is not significant in some cases, the reduction in computational cost is still significant. Transfer learning leverages pre-trained models to enable neural networks to acquire generic physics knowledge from related physical problems. This process enhances their ability to formulate and model new problems effectively. By incorporating a priori knowledge acquired during the pre-training phase, transfer learning facilitates more accurate and reliable inversion results in the solution process for the new problem. Additionally, the utilization of a pre-trained model enables the provision of a favorable initial state for the new problem, thereby expediting the convergence process. This paper adopts a strategy where the neural network parameters obtained during the pre-training stage are preserved as the initial parameters for the new model. Therefore, when applied to a new dataset, the model converges quickly to the optimal solution.

While the transfer learning-based S-FEM coupled PINN method proposed in this paper demonstrates remarkable performance in the inversion of various elastoplastic material parameters, it is essential to address the challenge of avoiding negative transfer in the transfer learning process. Transfer learning algorithms typically rely on the assumption that the source and target domains possess some level of interrelation [37,68]. However, if this assumption does not hold, negative transfer can occur, potentially resulting in inferior performance compared to scenarios where no transfer learning is employed [37,68]. In this paper, we assume that the models requiring parameter inversion share identical properties and conditions, except for the material parameters. This assumption ensures the correlation between the source and target domains, which is fundamental for effective transfer learning. Furthermore, the dataset necessary for the inversion using PINN is synthesized using S-FEM. It is important to acknowledge that, like any numerical method, S-FEM relies on certain assumptions and simplifications when compared to the actual problem. Therefore, there will be inherent limitations when synthesizing data using different numerical methods.

One of the key advantages of PINN lies in its ability to incorporate both physical information and data constraints during training, leading to solutions that satisfy both aspects. However, a challenge arises from the presence of multiple loss terms in PINN, which encompass both data-driven and physically driven components. The weights assigned to these loss terms significantly impact the computational results [26,69]. In this paper, we employ a fine-tuning approach to adjust the weights of the loss terms based on the computational results. However, for more complex problems, adopting a better weighting scheme, particularly an adaptive one, can be advantageous in enhancing overall convergence. Additionally, PINN, similar to other deep learning models, is specifically designed for particular problems. Developing effective neural network architectures often relies on user experience, which can be a time-intensive process. Furthermore, it is crucial to consider that a design approach successful in one problem scenario may not yield the same results in other problems. Thus, the impact of various design approaches on the results should be uniformly considered [26,69].

In our future research work, we intend to explore and investigate more combinations of numerical methods with PINN to solve various mechanical inverse problems. We aim to compare the applicable scenarios of different methods and avoid the limitation of the
scope of application of a single numerical method to synthesize data. Moreover, we plan to incorporate an adaptive weighting strategy to enhance the overall convergence of the method. This integration will enable the proposed method to be effectively applied to complex problems. Furthermore, we aspire to apply our method to a real engineering case, providing an opportunity to evaluate its practical utility and efficacy in real-world scenarios. These future research directions will contribute to a deeper understanding of the capabilities and limitations of the proposed method, as well as its potential for real-world applications.

6. Conclusions

In this paper, we propose a method for the transfer learning-based coupling of S-FEM and PINN for elastic-plastic material parameter inversions. The main idea is to synthesize a small-scale dataset using S-FEM and then combine it with PINN for pre-training, based on which the pre-trained model is retrained as the initial state of the new dataset. By using the saved parameters from the pre-trained model, the initial state of the neural network is no longer randomly initialized, leading to faster convergence of the training process and improving the overall efficiency of the inversion. The accuracy and efficiency of the coupling of S-FEM and PINN in terms of pre-training results are compared with those of the coupling of conventional FEM and PINN in this paper through an example of inverting the parameters of elastoplastic materials. The comparison results indicate that the coupling of S-FEM and PINN achieves higher computational accuracy and efficiency than the coupling of the conventional FEM and PINN. We demonstrate through linear elastic and elastoplastic material parameter inversion that the transfer learning-based coupling of S-FEM and PINN is at least twice as computationally efficient as the coupling of S-FEM and PINN without transfer learning. Our method is well-suited for the inversion of various elastoplastic material parameters in real-world engineering applications, leading to a reduction in the amount of required monitoring data and significant savings in computational costs. In future research, we plan to investigate and compare the performance of various numerical methods coupled with PINN for solving a wide range of complex mechanical inverse problems relevant to engineering applications.


Funding: This research was supported by the National Natural Science Foundation of China (Grant Numbers: 42277161).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editor and the reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

\( \sigma_{ij} \)  Stress Tensor
\( \sigma_{ij,i} \)  Partial Derivative of the Stress Tensor
\( \varepsilon_{ij} \)  Strain Tensor
\( u_i \)  Displacement
\( u_{i,j} \)  Partial Derivative of the Displacement
\( f_i \)  Volume Force
\( \bar{\sigma} \)  Equivalent Stress
\( \sigma_s \)  Yield Stress
\ell \quad \text{Equivalent Strain} \\
\sigma_m \quad \text{Mean Stress} \\
\varepsilon_m \quad \text{Mean Strain} \\
\eta \quad \text{Power Hardening Index} \\
E \quad \text{Young’s Modulus} \\
\nu \quad \text{Poisson’s Ratio} \\
\lambda \quad \text{Lamé Parameter 1} \\
\mu \quad \text{Lamé Parameter 2 / Shear Modulus} \\

\textbf{Abbreviations} \\
cPINNs \quad \text{conservation PINNs} \\
CS-FEM \quad \text{Cell-Based Smoothed Finite Element Method} \\
ES-FEM \quad \text{Edge-Based Smoothed Finite Element Method} \\
FDM \quad \text{Finite Difference Method} \\
FEM \quad \text{Finite Element Method} \\
FEM-PINN \quad \text{Coupling FEM and PINN} \\
fPINNs \quad \text{fractional-order PINNs} \\
FS-FEM \quad \text{Face-Based Smoothed Finite Element Method} \\
HS-FEM \quad \text{Hybrid Smoothed Finite Element Method} \\
NS-FEM \quad \text{Node-Based Smoothed Finite Element Method} \\
PDEs \quad \text{Partial Differential Equations} \\
PINN \quad \text{Physics-Informed Neural Network} \\
S-FEM \quad \text{Smoothed Finite Element Method} \\
S-FEM-PINN \quad \text{Coupling S-FEM and PINN} \\
vPINNs \quad \text{variational PINNs} \\

\textbf{References} \\
3. Rakesh; Salo, M. The fixed angle scattering problem and wave equation inverse problems with two measurements. *Inverse Probl.* 2020, 36, 035005. [CrossRef] \\
25. Liu, Q.; Lei, Y.; Yin, X.; Lei, J.; Pan, Y.; Sun, L. Development and application of a novel probabilistic back-analysis framework for geotechnical parameters in shield tunneling based on the surrogate model and Bayesian theory. *Acta Geotech.* 2023, 1–23. [CrossRef]


58. Westergaard, H.M. *Theory of Elasticity and Plasticity*; Harvard University Press: Cambridge, MA, USA, 1952. [CrossRef]


66. Li, Y.; Lv, W.; Li, G.; Zang, H. Macro and micro damage analysis and parameter inversion of HTPB adhesive Interface based on DIC and FEMU. *Compos. Interfaces* 2023. [CrossRef]

