International Journal for Numerical Methods in Engineering

WILEY

RESEARCH ARTICLE

A Graph Networks-Based Plastic Fracture Surrogate Model for Geomaterials

Kai Feng | Xiao-Ping Zhou

School of Civil Engineering, Chongqing University, Chongqing, China

Correspondence: Xiao-Ping Zhou (cqxpzhou@hotmail.com)

Received: 15 May 2025 | **Revised:** 4 January 2026 | **Accepted:** 7 January 2026

Keywords: computational efficiency | data-driven mechanics | graph neural networks | plastic fracture | surrogate model

ABSTRACT

Geomaterials exhibit highly nonlinear plastic deformation and fracture behaviors. Deep learning offers a promising alternative by exploiting data-driven nonlinear mappings to bypass explicit equation construction. However, existing methods have difficulty in handling noisy small-sample geotechnical data and lack systematic integration of physical priors (e.g., energy conservation, yield conditions) to ensure consistency. In addition, most alternative models are limited to material-scale predictions and need to be combined with traditional numerical methods to solve problems related to boundary conditions, which affects efficiency. This study proposes a novel graph neural network (GNN)-based surrogate model for plasticity-fracture modeling, bridging data-driven learning and physical principles. The framework encodes state information (nodes) and interactions (edges) via a graph structure, enabling efficient evolution prediction of physical fields while embedding interpretable mechanical components. Three numerical examples validate the accuracy and computational efficacy of the proposed model.

1 | Introduction

The research objects of geomaterials (such as rocks, soils, etc.) usually exhibit highly nonlinear plastic deformation and fracture behaviors, and their evolution process is jointly controlled by multiscale structural characteristics (such as pores, cracks, joints) and multiphysical field coupling effects (seepage, temperature, chemical corrosion). In the traditional theoretical framework, although the constitutive models based on continuum mechanics or fracture mechanics can partially describe the material responses, they still face significant challenges in characterizing complex path dependencies, cross-scale crack propagation, and heterogeneous geological body behaviors. At the same time, conventional numerical methods (such as the discrete element method) are difficult to calibrate model parameters when dealing with complex problems; the computational efficiency is low in large-scale engineering problem applications, and it is difficult to meet the real-time analysis needs, especially in deep underground engineering, geological disaster warning, and other scenarios. Therefore, more efficient solution strategies are urgently needed.

In recent years, the rapid development of deep learning technology [1–3] has provided new ideas for breaking through the above bottlenecks. With its powerful nonlinear mapping ability, neural networks (NNs) can directly mine implicit mechanical laws from experimental data or high-fidelity numerical simulations, thereby bypassing the complex explicit equation construction process in traditional methods [4–8]. In particular, it has methodological advantages in dealing with complex constitutive responses of heterogeneous materials [9–13].

When dealing with the history-dependent plastic behavior of materials, deep learning methods driven by historical data have also made a series of progress [14–17]. The core of this type of methods is to use a NN architecture that processes time series data, such as a recurrent neural network (RNN), to consider the influence of cumulative plastic strain. Its advantage is that it can learn directly from data. However, current research still faces key challenges: on the one hand, the strong noise and small sample characteristics of geotechnical data limit the generalization ability of the model; on the other hand, how to systematically embed mechanical prior knowledge (such as energy conservation and yield conditions) into the network architecture to ensure that the prediction results have both data-driven flexibility and physical consistency is still an unresolved problem. To address this problem, a method of predicting the plastic behavior of materials by training multiple interpretable component functions [18–20] has been proposed. This method improves the generalization ability and physical consistency of the model. However, in the above methods, the surrogate model can usually and only be applied at the material scale. It is difficult to use it alone in practical problems of geotechnical engineering and rock mechanics that need to consider changing boundary conditions. It needs to be combined with traditional numerical methods [16], which inevitably leads to programming complexity and reduces computational efficiency.

On the other hand, one prominent approach, known as physics-informed neural networks [21–23] (PINNs) based on automatic partial differentiation [24], has emerged to learn the solution of partial differential equations (PDEs) by unsupervised (or semi-supervised) training of a NN. PINNs aim to solve forward and inverse problems of PDEs by minimizing a loss function that integrates fundamental laws of physics. As a complete surrogate model, PINNs have been successfully applied in solid mechanics [25–28]. Furthermore, by introducing fracture energy in phase field approach [29–31] or peridynamic (PD) approach [32], PINNs were successfully applied in some crack problems [31, 33, 34]. PINN's latest research makes progress in solving dynamic Poisson-Nernst-Planck (PNP) equations [35], Hausdorff derivative elliptic problems [36], and fundamental solutions [37]. Compared with data-driven methods, the PINNs-based method can serve as a surrogate model to solve mechanical problems and has better interpretability. However, as a network approximate solution, the PINNs method requires achieving an approximate solution to the problem through the network training process. This can lead to high computational costs, particularly in scenarios involving crack propagation [32] and plastic behavior [38]. Therefore, the recent studies are to employ the physics-data combined machine learning methods to enhance the generalization using small training data [39, 40].

In fact, the surrogate model based on graph neural network (GNN) [41–43] has shown its potential in dealing with a complete computational domain problem with varying boundary conditions [44–46]. This type of methods takes advantage of the structured expression of graph $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$ and combines it with the ability of NN to learn data. The graph node V represents the state information of nodes (implicit) and material points (explicit) in the physical system, and the edge E represents the connection and interaction between them. Through learning data, the evolution prediction of the physical field can be successfully realized. This is especially true in dynamic problems [47], such as mesh contact [48], fluid dynamics [49], and particle flow [50]. Due to the excellent performance of GNN, GNN-based models have made a series of progress in the field of solid mechanics, such as using graphs to represent the spatial relationship of grains in polycrystalline materials [51], using multiple graphs [52] to reconstruct models of different fidelity in finite element proxy models and elastic–plastic behavior of composite materials [53]. GNN-based models also show high prediction accuracy and significant computational gains [54] when performing alternative modeling or simulation of complex systems. Recent research results have also reported progress in the prediction of crack propagation [55] and plastic behavior [56] of geomaterials using GNN-based surrogate models as alternative modeling schemes. However, the failure of geomaterials usually involves complex plastic-fracture coupling behaviors, there is still a lack of corresponding research.

Based on previous research [44, 55, 56], this paper combines GNN to propose a surrogate model for plastic fracture problems, focusing on the intersection of plastic fracture mechanics and deep learning, aiming to explore a new modeling paradigm driven by data-physics fusion to improve the prediction accuracy and computational efficiency of the mechanical behavior of complex rock and soil media. This key contribution in this paper includes: (1) a GNN-based surrogate model for plasticity-fracture problems with varying boundary conditions, (2) integration of physical constraints to enhance generalization, and (3) a unified data-physics paradigm for geomechanical simulations. The proposed approach demonstrates potential for real-time analysis in deep underground engineering and hazard warning.

The remainder of this paper is organized as follows: Section 2 briefly reviews the formulation of the improved ordinary state PD (OSB-PD) model. Section 3 proposes a novel surrogate model for the plasticity-fracture surrogate model, which is based on a GNN and integrates interpretable components. Section 4 discusses the details of network model training and testing. Section 5 takes three numerical examples. Finally, Section 6 draws the conclusions.

2 | OSB-PD Plasticity Model Considering Bond Shear Deformation

In this section, we first briefly introduce the OSB-PD model [31] that considers bond shear deformation as the learning object and comparison benchmark, namely, the OSB-PD model.

2.1 | OSB-PD Model Introduction

In OSB-PD, the concept of state is introduced to describe the motion compatibility and momentum balance between material points in $H(x)$, as shown in Figure 1.

As shown in Figure 1, a PD material point $\boldsymbol{x}$ interacts with a material point $\boldsymbol{x}'$ in its horizon with a radius of δ, and $\boldsymbol{\xi} = \boldsymbol{x}' - \boldsymbol{x}$ is the initial bond vector. The PD material points interact with each other through the bond force density, which includes material parameters. In order to describe the position of the material point in $H(x)$, which represents the neighborhood of x, the bond vector state in the reference configuration can be expressed as:

$$\underline{\boldsymbol{x}}\langle\boldsymbol{\xi}\rangle = \boldsymbol{\xi} = \boldsymbol{x}' - \boldsymbol{x} \tag{1}$$

where $\boldsymbol{\xi}$ is the relative position vector between material points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the reference configuration, and the corresponding bond scalar state is:

$$\underline{x}\langle\boldsymbol{\xi}\rangle = \left\|\boldsymbol{x}' - \boldsymbol{x}\right\| = \|\boldsymbol{\xi}\| \tag{2}$$

Similarly, the deformation vector state field is defined to describe the position of a material point $\boldsymbol{y}(\boldsymbol{x}, t)$ in the deformed configuration:

$$\underline{\boldsymbol{y}}\langle\boldsymbol{\xi}\rangle = \boldsymbol{y}(\boldsymbol{x}, t) - \boldsymbol{y}'(\boldsymbol{x}, t) = \boldsymbol{\xi} + \boldsymbol{\eta} \tag{3}$$

where $\boldsymbol{\eta} = \boldsymbol{u}\left(\boldsymbol{x}', t\right) - \boldsymbol{u}\left(\boldsymbol{x}, t\right)$ represents the relative displacement vector between material points $\boldsymbol{y}$ and $\boldsymbol{y}'$ in the deformed configuration, and the associated deformation scalar state is defined as:

$$\underline{y}\langle\boldsymbol{\xi}\rangle = \|\boldsymbol{\xi} + \boldsymbol{\eta}\| \tag{4}$$

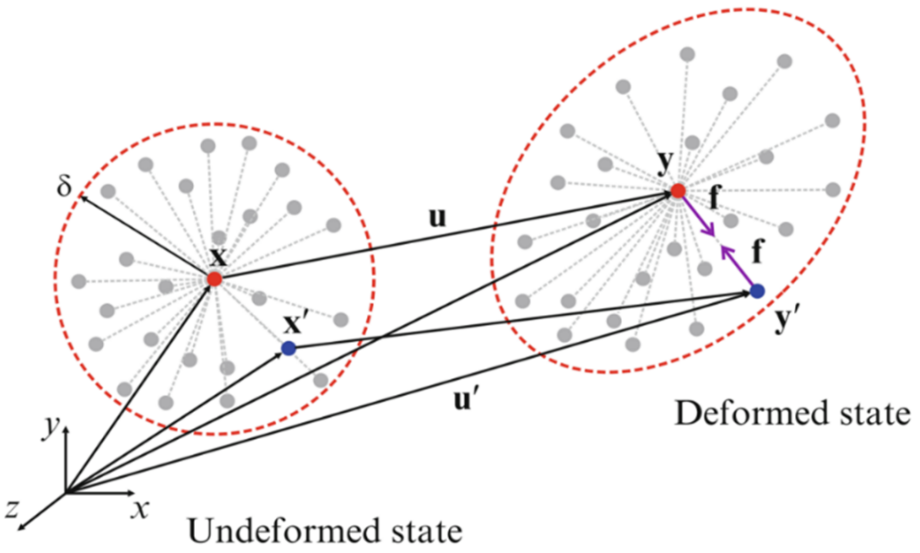


FIGURE 1 | Deformation of PD material points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the framework of the state-based peridynamics.

The extension scalar state of a bond $\boldsymbol{\xi}$ between two interacting material points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the deformed configuration is described by:

$$\underline{e}\langle\boldsymbol{\xi}\rangle = \underline{y}\langle\boldsymbol{\xi}\rangle - \underline{x}\langle\boldsymbol{\xi}\rangle = \|\boldsymbol{\xi}+\boldsymbol{\eta}\| - \|\boldsymbol{\xi}\| \tag{5}$$

According to the definition by Silling et al. [57], the scalar-valued expansion function $\Theta(\underline{e})$ is directly equal to the volume change of the neighboring material points in the neighborhood $H(\boldsymbol{x})$ of $\boldsymbol{x}$. In the 3D case, we have:

$$\Theta(\underline{e}) = \frac{3}{m}\int_{H(x)} \omega(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\| \cdot \underline{e}\langle\boldsymbol{\xi}\rangle dV_{x'} = \frac{3}{m}(\underline{\omega x}) \bullet \underline{e} \tag{6a}$$

In the 2D case, we have

$$\Theta(\underline{e}) = \frac{2}{m}\int_{H(\boldsymbol{x})} \omega(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\| \cdot \underline{e}\langle\boldsymbol{\xi}\rangle dV_{x'} = \frac{2}{m}(\underline{\omega x}) \bullet \underline{e} \tag{6b}$$

where $\|\boldsymbol{\xi}\|$ represents the influence function state that weights the contribution of each bond $\boldsymbol{\xi}$ in $H(\boldsymbol{x})$, and m is a scalar-valued weighted volume factor, which is defined as:

$$m = \int_{H(\boldsymbol{x})} \omega(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\|^2 dV_{x'} = (\underline{\omega x}) \bullet \underline{x} \tag{7}$$

To address the tensile-shear and compressive-shear crack propagation of geomaterials under compressive loads with plasticity, this paper employs an improved OSB-PD model to account for bond shear deformation [58]. In the previous state-based PDs, the volume dilatation function can be estimated by the total extension scalar state [57] and the displacement state $\underline{\boldsymbol{u}}(x,t)\langle\xi\rangle$ [59]. These approximations may encompass both the dilatation part and distortion part of the displacement components. In this study, it is assumed that the extension scalar state of a bond can be approximately partitioned into two parts: a pure dilatation part (along the $\boldsymbol{\xi}$ direction) and a pure distortion part (perpendicular to the $\boldsymbol{\xi}$ direction), as illustrated in Figure 2.

The new volume dilatation scalar-valued functions in 3D and 2D cases take the following form, respectively:

$$\Theta(\underline{e}_n) = \frac{3}{m}\int_{H(\boldsymbol{x})} \omega(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\| \cdot \underline{e}_n\langle\boldsymbol{\xi}\rangle dV_{x'} = \frac{3}{m}(\underline{\omega x}) \bullet \underline{e}_n \tag{8a}$$

$$\Theta(\underline{e}_n) = \frac{2}{m}\int_{H(\boldsymbol{x})} \omega(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\| \cdot \underline{e}_n\langle\boldsymbol{\xi}\rangle dV_{x'} = \frac{2}{m}(\underline{\omega x}) \bullet \underline{e}_n \tag{8b}$$

where $\underline{e}_n\langle\boldsymbol{\xi},\boldsymbol{\eta}\rangle$ is the normal extended scalar state, which is geometrically approximated by:

$$\underline{e}_n\langle\boldsymbol{\xi},\boldsymbol{\eta}\rangle = \|\boldsymbol{\xi}+\boldsymbol{\eta}\|cos\theta - \|\boldsymbol{\xi}\| \tag{9}$$

For OSB-PD, the integro-differential equation of motion for a material point $\boldsymbol{x}$ is:

$$\rho(\boldsymbol{x})\ddot{\boldsymbol{u}}(\boldsymbol{x},t) = \int_{H(\boldsymbol{x})} \left[\underline{\boldsymbol{T}}(\boldsymbol{x},t)\langle\boldsymbol{x}'-\boldsymbol{x}\rangle - \underline{\boldsymbol{T}}(\boldsymbol{x}',t)\langle\boldsymbol{x}-\boldsymbol{x}'\rangle\right] dV_{x'} + \boldsymbol{b}(\boldsymbol{x},t) \tag{10}$$

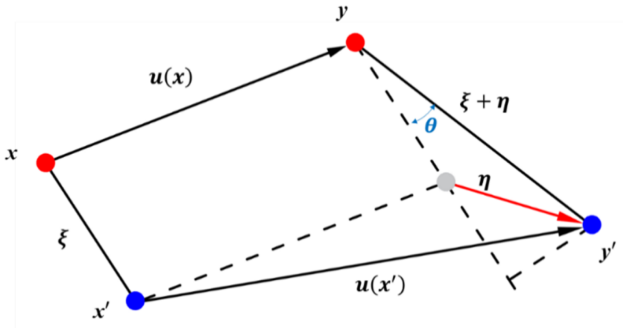


FIGURE 2 | Kinematic decomposition of volumetric and deviatoric deformation between two interacting material points.

where ρ is the material density, $\boldsymbol{x}$ is the position vector in the reference configuration, t represents time, $\underline{\boldsymbol{T}}\langle \boldsymbol{x}' - \boldsymbol{x} \rangle$ is the force state acting on the relative position vector, $\boldsymbol{x}' - \boldsymbol{x}$, $\boldsymbol{b}$ is the external force density, $\boldsymbol{u}$ is the displacement vector, $\ddot{\boldsymbol{u}}$ is the acceleration, and $H(\boldsymbol{x})$ represents the neighborhood of $\boldsymbol{x}$.

The macroscopic elastic strain energy density $\psi\left[\Theta\left(\underline{e}_n\right),\ \underline{e}^d\right]$ of the OSB-PD elastic solid can be rewritten as:

$$\begin{aligned}\psi\left[\Theta\left(\underline{e}_n\right),\ \underline{e}^d\right] &= \frac{\kappa'}{2}\Theta^2\left(\underline{e}_n\langle \boldsymbol{\xi}, \boldsymbol{\eta}\rangle\right) + \frac{\mu'}{2}\left(\underline{\omega e^d}\right) \bullet \underline{e}^d \\ &= \frac{\kappa'}{2}\Theta^2\left(\underline{e}_n\langle \boldsymbol{\xi}, \boldsymbol{\eta}\rangle\right) + \frac{\mu'}{2}\int_{H(\boldsymbol{x})} \underline{\omega}\left(\underline{e}^d\right)^2 dV_{x'}\end{aligned} \tag{11}$$

in which κ' and μ' are two PD parameters derived based on energy conservation, whose values are taken according to the literature [58]; $\underline{e}^d$ is the rotation part of the bond deformation.

Since the volume dilatation scale function in Equations (8a) and (8b) has a new definition, the new distortion part of the extension scalar state is expressed as:

$$\underline{e}^d = \underline{e} - \frac{\Theta\left(\underline{e}_n\right)}{3}\underline{x} \tag{12}$$

The elastic force density is defined as the Fréchet derivative of the PD density energy with respect to the relative displacement vector, which can be written as:

$$\underline{\boldsymbol{T}} = \nabla_{\boldsymbol{\eta}}\psi = \kappa'\Theta\frac{3\omega(\|\boldsymbol{\xi}\|)\|\boldsymbol{\xi}\|}{m} + \mu'\omega(\|\boldsymbol{\xi}\|)\underline{e}^d \tag{13}$$

2.2 | Drucker–Prager Plasticity

1. Yield function

In Equation (12), the contribution of the deformation of a single bond to volume dilatation is defined as $\underline{e}^i = \frac{\Theta\left(\underline{e}_n\right)}{3}\underline{x}$. Meanwhile, the energy density in Equation (11) can be decomposed into the following two distinct components: volume dilatation deformation and distortion deformation.

$$\psi^i = \frac{\kappa'}{2}\Theta^2 \tag{14}$$

$$\psi^d = \frac{\mu'}{2}\left(\underline{\omega e^d}\right) \bullet \underline{e}^d \tag{15}$$

The Drucker–Prager yield criterion can be written as:

$$f_y(\boldsymbol{\sigma}, \zeta) = \sqrt{J_2} + \alpha_{DP} I_1 - k_{DP}(\zeta) \tag{16}$$

In the modified OSB-PD model, I_1 and J_2 in PD form can be expressed by ψ^i and ψ^d, respectively [60]:

$$I_1 = tr(\boldsymbol{\sigma}) = sign(\Theta)\sqrt{\frac{6E}{1-2\upsilon}\psi^i} \tag{17a}$$

$$J_2 = \frac{1}{2}\mathcal{S}_{ij}\mathcal{S}_{ij} = \frac{E}{1+\upsilon}\psi^d \tag{17b}$$

The mean pressure p and the deviatoric stress (von Mises) q can be defined as:

$$p = \frac{1}{3}tr(\boldsymbol{\sigma}) = \frac{1}{3}I_1 \tag{18a}$$

$$q = \sqrt{\frac{3}{2}\mathcal{S}_{ij}\mathcal{S}_{ij}} = \sqrt{3J_2} \tag{18b}$$

Combining Equations (17a), (17b), (18a), and (18a), we have:

$$p = \frac{1}{3}sign(\Theta)\sqrt{\frac{6E}{1-2\upsilon}\psi^i} \tag{19a}$$

$$q = \sqrt{\frac{3E}{1+\upsilon}\psi^d} \tag{19b}$$

2. Flow criteria
The direction of the plastic strain increment $d\varepsilon_i^p$ can be defined based on the nonassociative plastic flow law, which is given by:

$$d\varepsilon_i^p = d\lambda\frac{\partial g(\tau)}{\partial \tau_i} = d\lambda\left(\alpha'_{DP} + \frac{\mathcal{S}_{ij}}{2\sqrt{J_2}}\right) \tag{20}$$

where $d\ \varepsilon_i^p (i = 1, 2, 3)$ is the principal plastic logarithmic strain increment, $d\lambda > 0$ is the plasticity multiplier, $g = \sqrt{J_2} + \alpha'_{DP}I_1$ is the plastic potential, and $\tau_i (i = 1, 2, 3)$ is the principal Kirchhoff stress.
Considering plastic deformation, the PD deformation $\underline{e}$ can be divided into the plastic part and the elastic part, which can be written as:

$$\underline{e} = \underline{e}^e + \underline{e}^p \tag{21}$$

where $\underline{e}^e$ and $\underline{e}^p$ are the elastic and plastic parts of the key deformation, respectively, which can be written as $\underline{e}^e = \underline{e}^{ie} + \underline{e}^{de}$, $\underline{e}^p = \underline{e}^{ip} + \underline{e}^{dp}$.
The PD flow rule can be described by incremental plastic stretching $d\underline{e}^{ip}$ and $d\underline{e}^{dp}$ in the following form:

$$d\underline{e}^{ip} = x^i\underline{e}^{ie} \tag{22a}$$

$$d\underline{e}^{dp} = x^d\underline{e}^{de} \tag{22b}$$

where x^i and x^d are undetermined parameters.
In classical continuum mechanics and PD, the expressions for incremental plastic volume dilatation strain are calculated as:

$$d\Theta^p\left(d\varepsilon_i^p\right) = d\lambda\left(3\alpha'_{DP} + \frac{\mathcal{S}_1 + \mathcal{S}_2 + \mathcal{S}_3}{2\sqrt{J_2}}\right) = 3\alpha'_{DP}d\lambda \tag{23a}$$

$$d\Theta^p\left(d\underline{e}^{ip}\right) = a\omega \bullet d\underline{e}^{ip} = x^i\Theta^e \tag{23b}$$

where $\Theta^e = a\omega \bullet \underline{e}^{ie}$, $a = \frac{3}{m}$ (the 3D case), and $x^i = \frac{3\alpha'_{DP}\ d\lambda}{\Theta^e}$ can be solved by Equations (23a) and (23b).
Similarly, the increment J_2 of plastic deformation contribution is expressed as:

$$dJ_2\left(d\varepsilon_i^p\right) = \mathcal{S}_{ij}d\mathcal{S}_{ij} = \frac{E}{1+\upsilon}\sqrt{J_2}d\lambda \tag{24a}$$

$$dJ_2\left(ds^{dp}\right) = \frac{2E}{1+\upsilon}\mu'\omega\underline{e}^{de} \bullet \underline{e}^{dp} = 2x^dJ_2 \tag{24b}$$

where $x^d = \frac{E}{2(1+\upsilon)\sqrt{J_2}}d\lambda$ is obtained from Equations (24a) and (24b).

2.3 | Failure Criteria

In this work, the stored elastic strain energy density scalar state $\underline{w}\langle\boldsymbol{\xi}\rangle$ is decomposed into isotropic and distortion parts, namely, $\underline{w}^i\langle\boldsymbol{\xi}\rangle$ and $\underline{w}^d\langle\boldsymbol{\xi}\rangle$.

$$\underline{w}\langle\boldsymbol{\xi}\rangle = \underline{w}^i\langle\boldsymbol{\xi}\rangle + \underline{w}^d\langle\boldsymbol{\xi}\rangle \tag{25a}$$

$$\underline{w}^i\langle\boldsymbol{\xi}\rangle = \frac{1}{2}\underline{t}^i\langle\boldsymbol{\xi}\rangle \cdot \underline{e}_n\langle\boldsymbol{\xi},\boldsymbol{\eta}\rangle \tag{25b}$$

$$\underline{w}^d\langle\boldsymbol{\xi}\rangle = \frac{1}{2}\underline{t}^d\langle\boldsymbol{\xi}\rangle \bullet \underline{e}^d\langle\boldsymbol{\xi},\boldsymbol{\eta}\rangle \tag{25c}$$

According to the energy conservation law, the critical fracture energy release rate due to volume dilatation can be obtained as:

$$w_c^i\langle\boldsymbol{\xi}\rangle = \begin{cases} \frac{4G_{Ic}}{\pi\delta^4} & \text{for 3D case} \\ \frac{9G_{Ic}}{4\pi h\delta^3} & \text{for 2D case} \end{cases} \tag{26}$$

where h denotes the thickness, and G_{Ic} represents the critical mode I fracture energy release rate of solid materials.

According to the previous literature [61], the critical energy release rate due to pure shear deformation can be expressed as:

$$w_c^d\langle\boldsymbol{\xi}\rangle = \begin{cases} \frac{24G_{IIc}}{\pi\delta^4} & \text{for 3D case} \\ \frac{45G_{IIc}}{8\pi h\delta^3} & \text{for 2D case} \end{cases} \tag{27}$$

where G_{IIc} denotes the critical mode-II fracture energy release rate of solid materials.

The Benzeggagh-Kenane semi-empirical failure criterion [62] offers a unified framework to describe pure tension fracture, pure shear fracture, and mixed tension-shear fracture modes in brittle and quasi-brittle solids, which can be formulated as:

$$G_c = G_{Ic} + \left(G_{IIc} - G_{Ic}\right)\left(\frac{G_{II}}{G_{II} + G_I}\right)^{\beta} \tag{28}$$

where G_c is the critical mixed fracture energy release rate, G_I and G_{II} are the type I and type II fracture energy release rates, and β is the empirical material parameter.

Referring to the application of the Benzeggagh–Kenane semi-empirical failure criterion reported in references [58, 61, 63], and considering that geomaterials exhibit higher shear strength under compression, the critical shear failure (mode II failure) is excluded from the mixed failure criterion. Specifically, only mode I and mode I/II mixed failure modes are considered. The failure criterion employed in this paper can be expressed as:

$$w_f = min\left\{w_c^i, w_c^i + \left(w_c^d - w_c^i\right)\left(\frac{\underline{w}^d\langle\boldsymbol{\xi}\rangle}{\underline{w}^i\langle\boldsymbol{\xi}\rangle + \underline{w}^d\langle\boldsymbol{\xi}\rangle}\right)^{\beta}\right\} \tag{29}$$

In brittle solids, when the new bond-level energy failure criterion in Equation (29) is reached, the bond $\boldsymbol{\xi}$ is completely broken, and the scalar function of the bond state can be written as.

$$g\langle\boldsymbol{\xi},t\rangle = \begin{cases} 1 & \underline{w}\langle\boldsymbol{\xi}\rangle < w_f \\ 0 & \underline{w}\langle\boldsymbol{\xi}\rangle \geq w_f \end{cases} \tag{30}$$

To characterize the strain-softening behavior of quasi-brittle solids, we employ the generalized scaling function $g\langle\boldsymbol{\xi},t\rangle$ to depict the damage-softening behavior of a single bond in the postfailure state, as illustrated in Figure 3. Based on previous

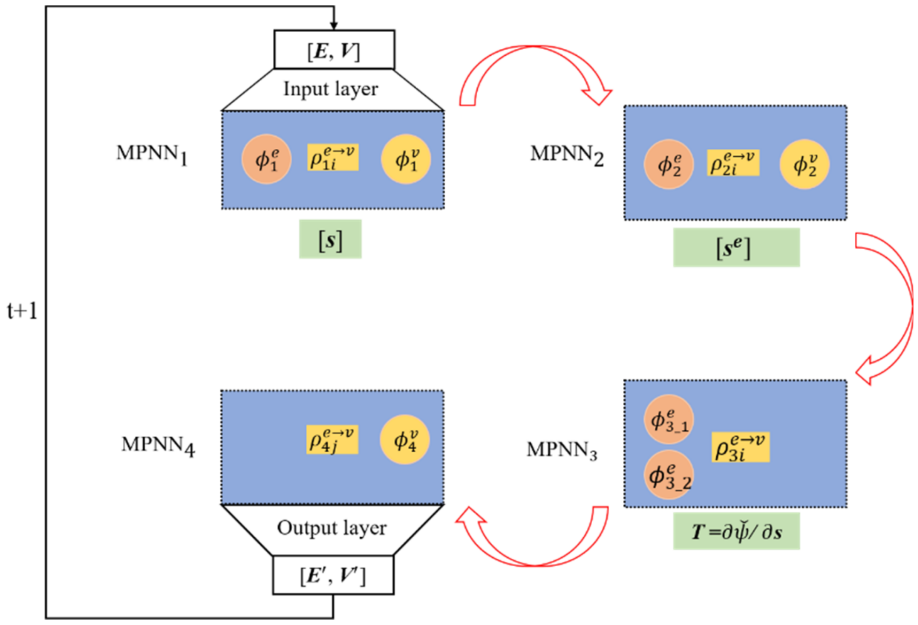


FIGURE 3 | Schematic diagram of the plastic fracture surrogate model. The four MPNN modules correspond to four distinct computational stages as defined by the procedure code: MPNN1 is designated for calculating deformation, MPNN2 for determining the plastic state, MPNN3 for evaluating unbalanced forces, and MPNN4 for updating the positional state.

studies [64], this paper employs the simplest linear softening law to describe the strain-softening behavior of quasi-brittle solid materials.

$$g\langle \boldsymbol{\xi}, t\rangle = \begin{cases} 1 & \underline{w}\langle \boldsymbol{\xi}\rangle < w_{ini} \\ 1 - \frac{\underline{w}\langle \boldsymbol{\xi}\rangle - w_{ini}}{w_f - w_{ini}} & w_{ini} < \underline{w}\langle \boldsymbol{\xi}\rangle < w_f \\ 0 & \underline{w}\langle \boldsymbol{\xi}\rangle \geq w_f \end{cases} \tag{31}$$

where w_{ini} is the initial value of the bond-level energy density at which bond damage begins, w_f represents the final value of the bond-level energy density and can be determined by Equation (29), where $\underline{w}\langle \boldsymbol{\xi}\rangle \geq w_f$ means that the interaction through this bond disappears.

According to the original nonlocal idea, the damage variable at any material point $\boldsymbol{x}$ is

$$d\langle \boldsymbol{x}, t\rangle = 1 - \frac{\int_{H(\boldsymbol{x})} g\langle \boldsymbol{\xi}, t\rangle dV_{x'}}{\int_{H(\boldsymbol{x})} dV_{x'}} \tag{32}$$

3 | Plastic Fracture Surrogate Model

In this paper, to address the issue of plastic fracture, we incorporate a fracture module based on the plastic model. Figure 3 shows a simplified schematic diagram of the proposed method. As depicted in Figure 3, the plastic fracture surrogate model comprises four Message Passing Neural Network (MPNN) modules (represented as blue modules). These modules respectively correspond to the deformation, plasticity, fracture, and control equations within the physical model. The red arrows signify the internal processing flow of the proposed model. During a complete time-step, the surrogate model updates the deformation state, plasticity state, damage state and displacement state. The green box indicates the key variable that the corresponding MPNN module is expected to acquire. It is important to note that this variable is not the specific output of the MPNN. The output of each MPNN is the updated graph $\boldsymbol{G}' = (\boldsymbol{V}', \boldsymbol{E}')$ [44]. The variable information is encapsulated within the updated graph node matrix (related to plasticity and displacement) or the updated graph edge matrix (associated with deformation and damage). The specific configurations of each MPNN will be detailed subsequently. Once all calculations are finalized at a time step t, the model is invoked once. Prior to performing calculations at the $t + 1$ time step (when the model is invoked again), the constraints of the boundary conditions should be

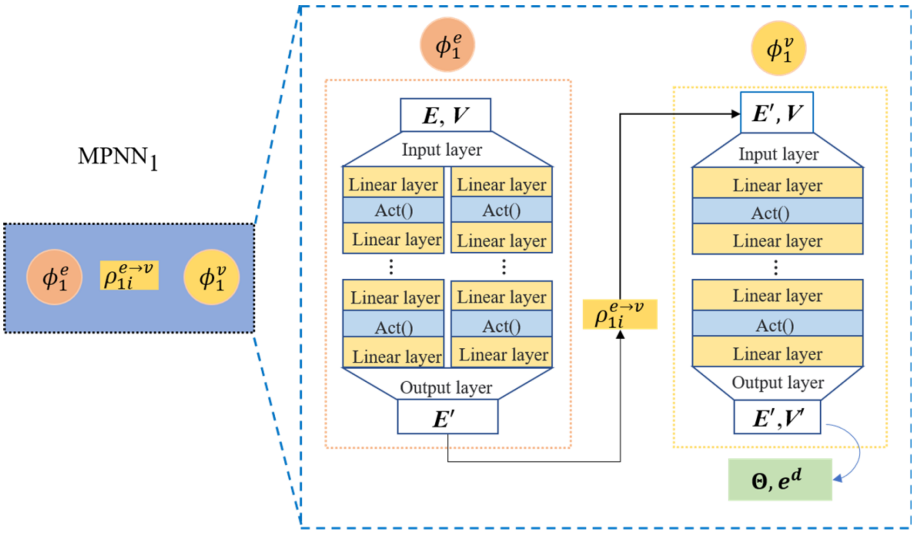


FIGURE 4 | Specific network settings of submodule MPNN1.

considered. Likewise, the information regarding the boundary conditions is implemented by updating or maintaining the corresponding quantities in the node matrix or edge matrix.

Figure 4 shows the first submodule (MPNN_1) in the plastic fracture surrogate model, which is designed to handle the corresponding deformation calculation part in the proposed physical model. As shown in Figure 4, MPNN_1 consists of three functions, namely ϕ_1^e, $\rho_{1i}^{e\to v}$, and ϕ_1^v. Among them, ϕ_1^e generates new edge distance information based on the input graph node and edge information, $\rho_{1i}^{e\to v}$ aggregates the updated distance information to the node, and then generates the deformation of all nodes and the deformation information in the edge through ϕ_1^v.

MPNN_1 calculates the bond deformation implied by the displacement field in the input node matrix $\boldsymbol{V}$ and the material point volume deformation $\boldsymbol{s}[\Theta, \underline{e}^i, \underline{e}^d]$ according to the process shown in Equations (33a), (33b), (33c). The obtained bond deformation is stored in $\boldsymbol{E}$ after the current step update, and the volume deformation is stored in $\boldsymbol{V}$ after the update.

$$E'_{1i} \leftarrow \phi_1^e(\boldsymbol{E},\boldsymbol{V}), \tag{33a}$$

$$\overline{\boldsymbol{e}}'_{1i} \leftarrow \rho_{1i}^{e\to v}\left(E'_{1i}, \boldsymbol{E}\right) \tag{33b}$$

$$\boldsymbol{s} \leftarrow \phi_1^v\left(\overline{\boldsymbol{e}}'_{1i}\right) \tag{33c}$$

Figure 5 shows the second submodule (MPNN_2) in the plastic fracture surrogate model, which is designed to handle the corresponding plastic part in the physical model. As shown in Figure 5, MPNN_2 consists of three functions, namely ϕ_2^e, $\rho_{2i}^{e\to v}$ and ϕ_2^v. Among them, ϕ_2^e generates new edge energy information based on the deformation information updated in the previous step, $\rho_{2i}^{e\to v}$ aggregates the updated energy information to the node, then ϕ_2^v is used to judge the plastic state of all nodes, to determine the plastic increment, and to update the elastic deformation information. MPNN_2 calculates the elastic energy at the material point and the corresponding first and second stress invariants according to the process shown in Equations (17a) and (17b), and determines the yield function value in combination with the internal plastic state.

$$\boldsymbol{F} \leftarrow \phi_2^e(\boldsymbol{s}, \boldsymbol{E}) \tag{34a}$$

$$\breve{\psi}^i, \breve{\psi}^d \leftarrow \rho_{2i}^{e\to v}(\boldsymbol{F}, \boldsymbol{E}) \tag{34b}$$

$$\breve{f}_y = \phi_2^v(p, q, \zeta) \tag{34c}$$

If the yield function $\breve{f}_y > 0$, the plasticity multiplier $d\lambda$ is determined by the Newton iteration method with automatic differentiation, and the plastic state is updated as:

$$d\lambda = d\lambda - \frac{\breve{f}_y}{\partial \breve{f}_y / \partial d\lambda} \tag{35a}$$

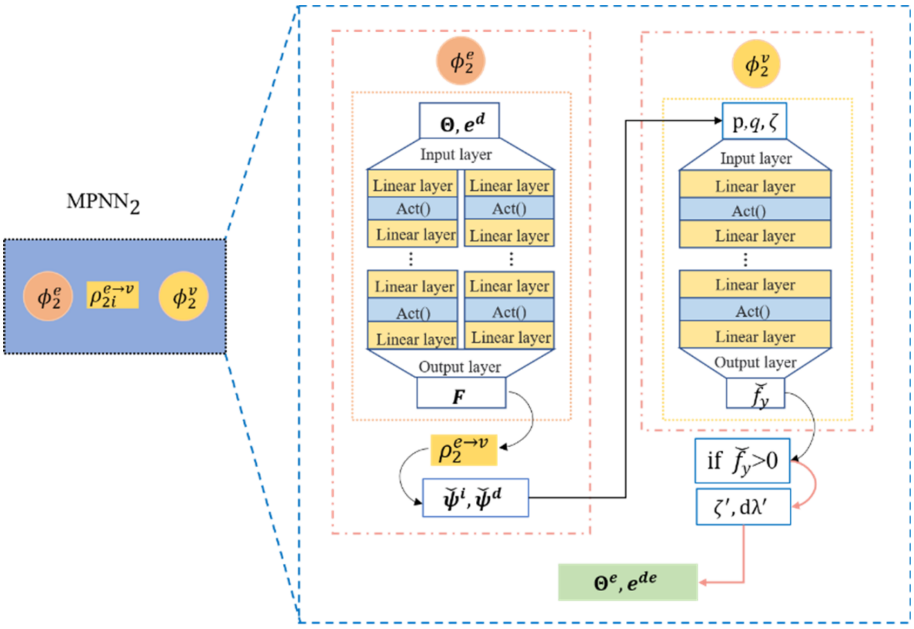


FIGURE 5 | Specific network settings of submodule MPNN2.

$$\check{f}_y(d\lambda) = \check{f}_y\left(p', q', \zeta' | \overline{\boldsymbol{W}}, \overline{\boldsymbol{b}}\right) \tag{35b}$$

Among them, $p'(d\lambda)$, $q'(d\lambda)$, $\zeta'(d\lambda)$ are calculated according to Algorithm 1, and $\overline{\boldsymbol{W}}$ and $\overline{\boldsymbol{b}}$ represent the fixed weight parameters of the trained network $\check{f}_y$.

Figure 6 shows the third submodule (MPNN_3) in the plastic fracture surrogate model, which is designed to handle the corresponding damage part in the physical model. As shown in Figure 6, MPNN_3 consists of two edge update functions $\phi^e_{3\_1}$, $\phi^e_{3\_2}$ and an edge-to-node aggregation function $\rho^{e\to v}_{3i}$. Among them, the first edge update function $\phi^e_{3\_1}$ generates new edge energy information X_1 based on the elastic deformation information updated in the previous step, and the corresponding second edge update function $\phi^e_{3\_2}$ generates edge damage information X_2 based on the elastic deformation updated in the previous step. $\rho^{e\to v}_{3i}$ aggregates the edge energy information after damage to the node and the bond force related information on the edge is obtained. It is worth noting that there is no node update function in this submodule. Considering the trainability and prediction accuracy of the proposed model, the relationship between bond force and energy is explicitly defined as the partial derivative of energy with respect to deformation, and is realized by the automatic differential counting of the surrogate network model corresponding to the function.

MPNN_3 calculates the elastic deformation $\boldsymbol{s}^e[\Theta^e, \underline{e}^{ie}, \underline{e}^{de}]$ after updating the plastic state according to the process shown in Equations (36a), (36b), and (36c), it determines the damage state in combination with the key damage judgment network function, and it calculates the key force state and point damage value d. In aggregate function $\rho^{e\to v}_{3i}$, the subscript i indicates that the direction of information propagation in MPNN_3 is $j \to i$.

$$X_1 \leftarrow \phi^e_{3\_1}(\mathbf{s}^e, \boldsymbol{E}) \tag{36a}$$

$$X_2 \leftarrow \phi^e_{3\_2}(\mathbf{s}^e, \boldsymbol{E}) \tag{36b}$$

$$\check{\psi}^i, \check{\psi}^d, \boldsymbol{T}_i \leftarrow \rho^{e\to v}_{3i}\left(E'_{3i}, \boldsymbol{E}\right) \tag{36c}$$

Figure 7 shows the fourth submodule (MPNN_4) in the plastic fracture surrogate model, which is designed to handle the corresponding control equation part in the physical model. As shown in Figure 7, MPNN_4 consists of an edge-to-node aggregation function $\rho^{e\to v}_{4j}$ and a node update function ϕ^v_4. Among them, $\rho^{e\to v}_{4j}$ is a supplement to the aggregation function $\rho^{e\to v}_{3i}$ in MPNN_3. Since the information transmission direction corresponding to the edge aggregation function in the same MPNN is fixed ($i \leftarrow j$ or $j \leftarrow i$), in the OSB-PD model the force state of a point is not only related to the effect of j on i, but also to the effect of i on j, as shown in Equation (10). Therefore, it is necessary to superimpose the effect of i on j

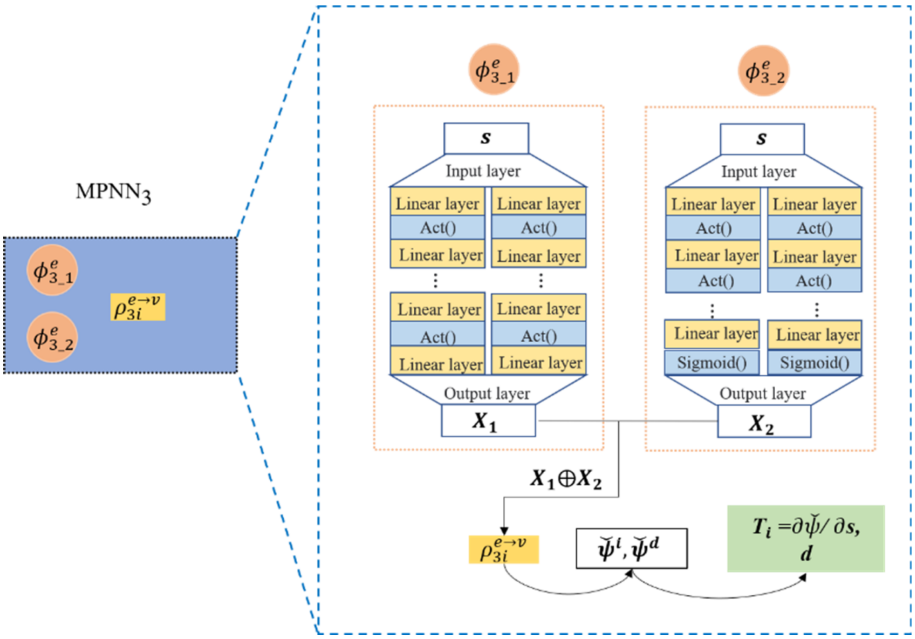


FIGURE 6 | Specific network settings of submodule MPNN3.

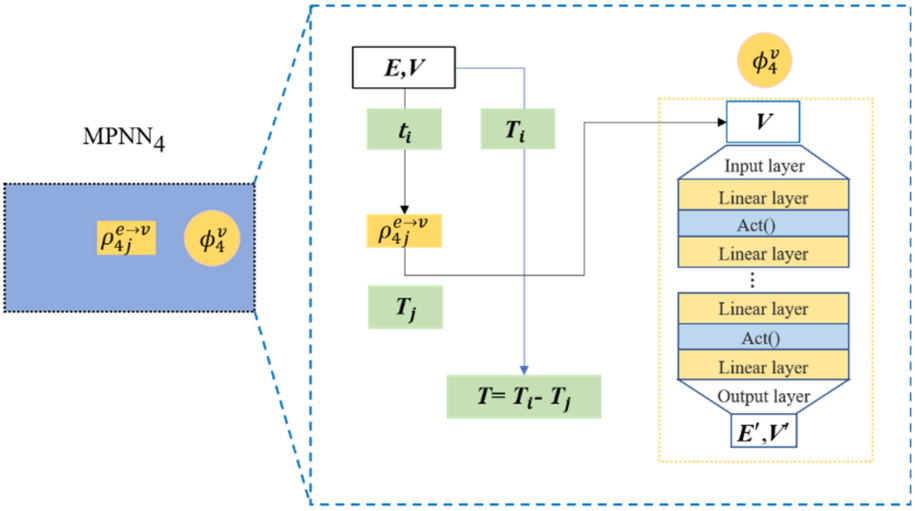


FIGURE 7 | Specific network settings of submodule MPNN4.

through $\rho^{e\to v}_{4j}$, and finally the update of the state information of the force on the node is formed. On this basis, ϕ^v_4 updates the displacement information through the position information and the state of the force on the node.

MPNN_4 calculates and updates the material point force and displacement according to the process shown in Equations (37a) and (37b). The subscript j in $\rho^{e\to v}_{4j}$ indicates that the direction of information propagation in MPNN_4 is $i \to j$.

$$\boldsymbol{T}_j \leftarrow \rho^{e\to v}_{4j}\left(\boldsymbol{T}_i, \boldsymbol{E}\right) \tag{37a}$$

$$\left(\boldsymbol{E}', \boldsymbol{V}'\right) \leftarrow \phi^v_4\left(\boldsymbol{T}_i - \boldsymbol{T}_j, \boldsymbol{E}, \boldsymbol{V}\right) \tag{37b}$$

All functions in the above modules, except for the aggregation function, need to be implemented through MLP. Considering the model training cost and the prediction accuracy of the network, in the simplified version, only the update functions in MPNN_2 and MPNN_3 can be implemented through the network. The specific network width, depth, activation function, and other hyperparameters are given in the subsequent model training and verification.

4 | Model Training and Testing

4.1 | Sobolev Training of Elastic Energy Functional

In this section, we first introduce the training methods of elastic energy function and yield function, and perform preprocessing of labeled data and verification testing of trained functions.

In order to ensure the accuracy and smoothness of the stress and tangent operators obtained from the learned energy function, the Sobolev training framework [65] is extended to high-order derivative constraints. For the training sample $i \in [1, 2, \cdots, N]$, the H_2 training objective of the energy function in Equation (11) is now the Loss function and can be written as:

$$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\gamma_1 \left\| \psi_i - \breve{\psi}_i \right\|_2^2 + \gamma_2 \left\| \frac{\partial \psi_i}{\partial s_i} - \frac{\partial \breve{\psi}_i}{\partial s_i} \right\|_2^2 + \gamma_3 \left\| \frac{\partial^2 \psi_i}{\partial s_i \partial s_i} - \frac{\partial^2 \breve{\psi}_i}{\partial s_i \partial s_i} \right\|_2^2 \right) \right) \tag{38}$$

where ψ_i represents the i-th data in the sampling space, including two components ψ^i and ψ^d; the corresponding s_i also contains two components, volume deformation Θ and bond rotation deformation $\underline{e}^d$; $\breve{\psi}_i$ corresponds to the energy of the network output, which is described by the ϕ_2^e function in MPNN_2, consisting of two independent MLPs, each of which contains an input layer with a shape of 1×20, followed by five hidden layers with a shape of 20×20, each hidden layer is followed by a $Tanh()$ activation layer, and the final output layer is 20×1; N is 10,000; input γ_1, γ_2, and γ_3 are weight coefficients of different order constraint loss terms, which can be adjusted according to the training accuracy and prediction effect parameters.

If there is no other provision, this paper takes $\gamma_1 = \gamma_2 = \gamma_3 = 1$ as the value. The initial learning rate of training is 0.0001, and the learning rate decays once every 200 epochs, with a decay coefficient of 0.1.

Figure 8 shows the network energy function training loss curve. It is found from Figure 8 that after 10,000 epoch training cycles, the loss terms of different orders tend to converge. The higher the order of the loss term, the higher the value after convergence during the training process. This is because its loss is greater than its label value; the energy is the quadratic deformation, the force is the first power of the deformation, and the normalized deformation value range is [−1,1].

Figure 9 shows the prediction results of the trained energy function on the test set. Figure 9a shows the elastic potential energy contributed by the volume dilatation part of OSB-PD, and Figure 9b shows the elastic potential energy contributed by the bond rotation deformation part. The energy is the direct output of the network, the force is the first-order derivative of the network output to the input, and the modulus is the second-order derivative of the network output to the input,

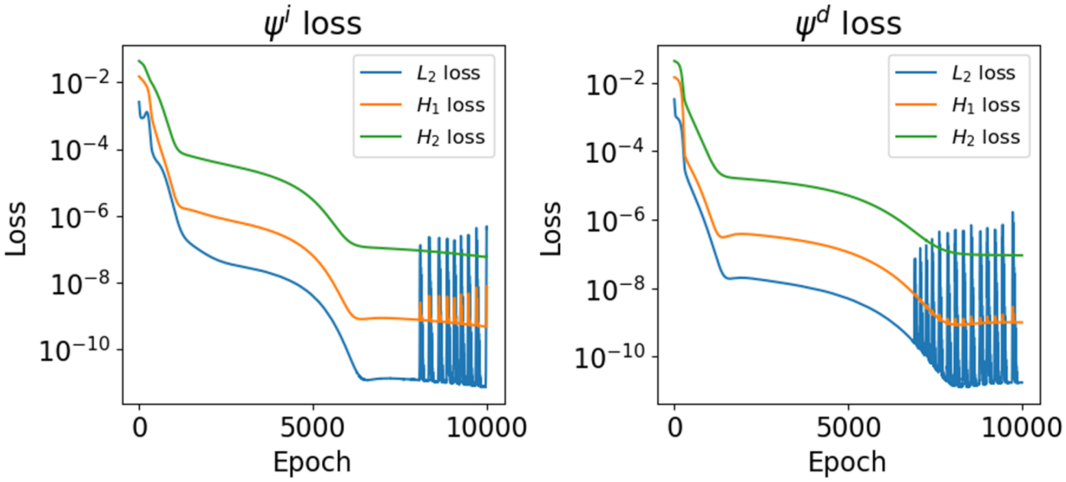


FIGURE 8 | Energy function training loss curve. The symbol L_2 denotes the loss curve derived from the conventional mean squared error, H_1 represents the loss curve incorporating a regularization term based on first-order partial derivatives, and H_2 corresponds to the loss curve that integrates both first-order and second-order loss terms.

which is realized by the network automatic differentiation technology [24]. Figure 9 shows that the network energy function has achieved high prediction accuracy on the test set, which is mainly due to the use of independent MLP training for the energy function components and the introduction of high-order loss terms in the loss function.

4.2 | Plastic Yield Function Training

In this section, we will continue to apply the concept of level set in Section 4.4 to generate a training data set for the yield function, except that the sampling and training are performed in the stress invariant space. The yield function is then assumed to be a signed distance function, which is defined as:

$$\Phi(\boldsymbol{x},\zeta)=\begin{cases} d(\boldsymbol{x}) & \text{outside } f_y \\ 0 & \text{on } f_y \\ -d(\boldsymbol{x}) & \text{inside } f_y \end{cases} \tag{39}$$

where $d(\boldsymbol{x})$ is the minimum Euclidean distance between any point $\boldsymbol{x}$ in stress space and the surface $f_y = \left\{x \in \mathbb{R}^3 | f(\boldsymbol{x}) = 0\right\}$, which is defined as:

$$d(x) = \min\left(\left|\boldsymbol{x} - \boldsymbol{x}_y\right|\right) \tag{40}$$

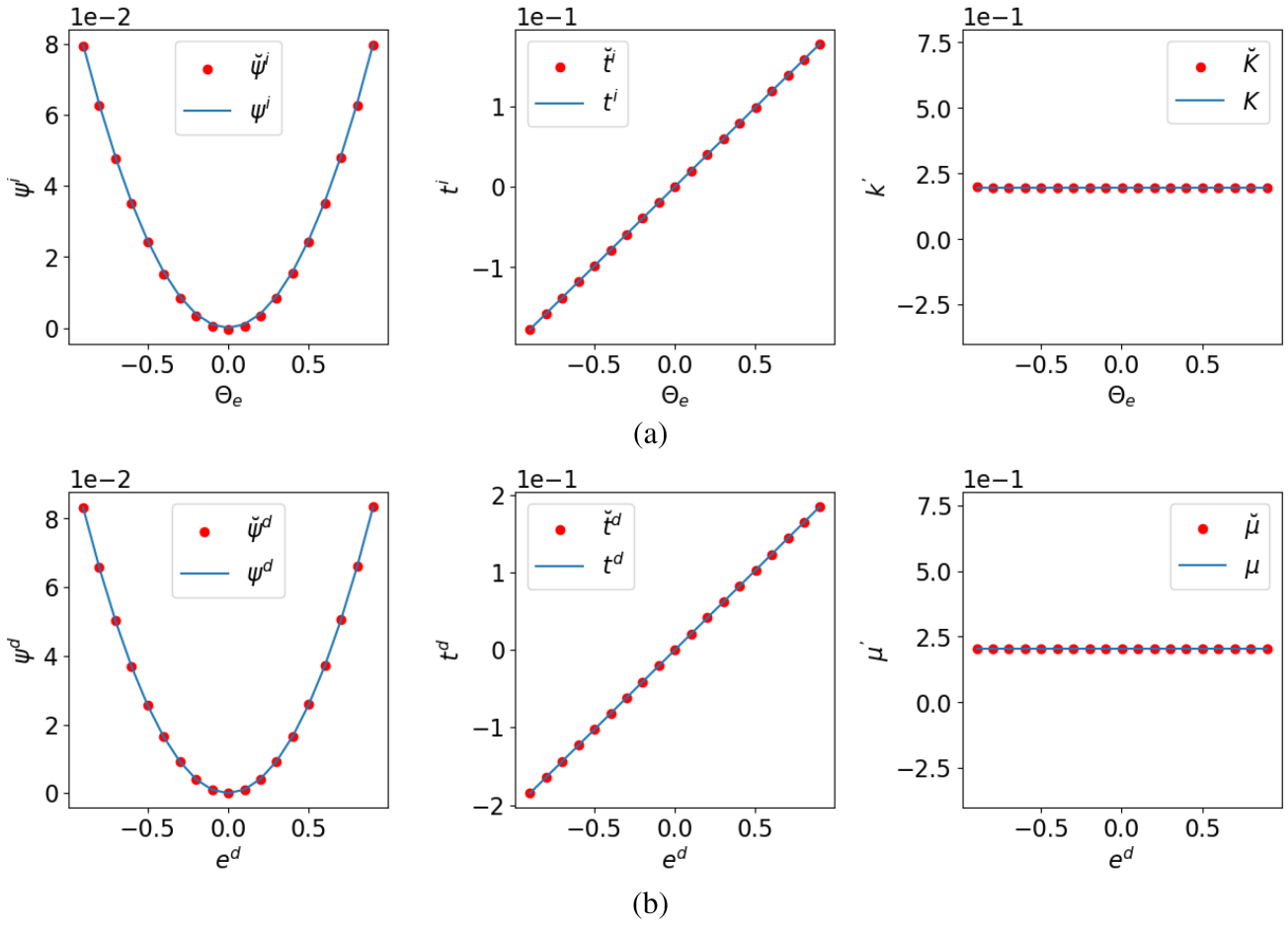


FIGURE 9 | Network energy function test after training. The blue solid line represents the true value, and the red dots indicate the predictions by the surrogate model. (a) From left to right, plots present the comparative test results of model predictions against reference data for: The value of $\breve{\psi}^i$, the first derivative of $\breve{\psi}^i$, and (c) the second derivative of $\breve{\psi}^i$. (b) From left to right, plots present the comparative test results of model predictions against reference data for: The value of $\breve{\psi}^d$, the first derivative of $\breve{\psi}^d$, and (c) the second derivative of $\breve{\psi}^d$.

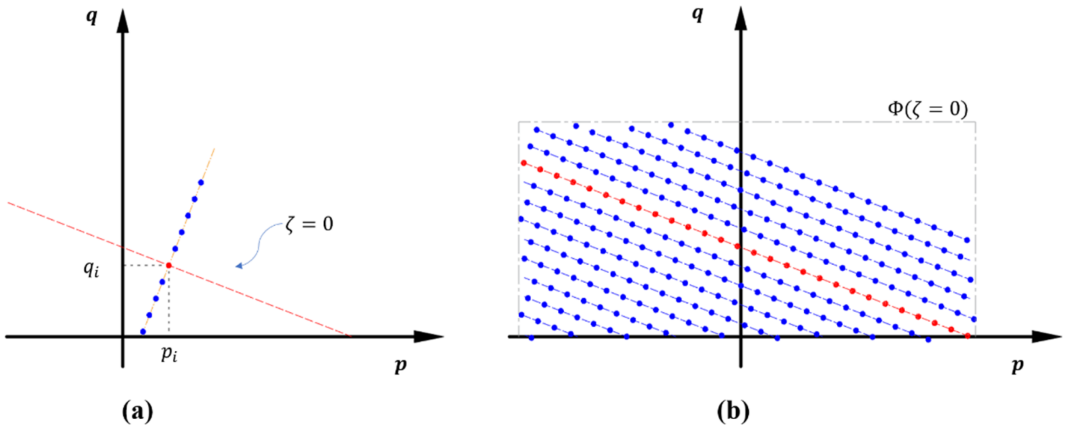


FIGURE 10 | Schematic diagram of yield function training data set generation. (a) The original yield surface is indicated by the red dashed line, with interpolation sampling points denoted by blue dots; (b) The complete sampling space used for training the yield function.

ALGORITHM 1 | Constructing the Yield Function Level Set (Data Enhancement).

Initialize an empty set of augmented training samples (p_m, q_m) for m in [0,..., $N \times$ L].
$m = 1$
for i in [0,..., N] do
  for j in [0,..., L] do
    $q_m = \frac{j\varsigma}{L} q_i$, the radius range of the signed distance function is $\left[0, \varsigma q_i\right]$
    $p_m = p_i, \zeta_m = \zeta_i$
    $f_m = \frac{j\varsigma}{L} q_i - q_i$, The signed distance function value range is $\left[-q_i, \ (\varsigma - 1) q_i\right]$
    $m = m + 1$

where $\boldsymbol{x}_y$ is the yield stress for a given value of the accumulated plastic strain ζ at time t, and the plastic internal variable ζ increases monotonically, indicating the history-dependent behavior of the material.

As shown in Figure 10, in the stress invariant space, for any stress state point (p_i, q_i) on the yield surface, $f_y(p_i, q_i) = g(p_i, q_i) - k_{DP}(\zeta) = 0$ is always satisfied. In Figure 10, the signed distance function of the yield surface can be obtained by solving the reinitialization problem of the level set to obtain the inferred yield surface (red point) and auxiliary data points (blue point). Figure 10a shows the data interpolation enhancement diagram of the yield stress state point (p_i, q_i) when $\zeta = 0$. The red dashed line represents the yield surface implied by the data, and the blue point represents the stress state point obtained by interpolation of (p_i, q_i). The yield function value of this point will be recorded in the subset $\boldsymbol{\Phi}(\boldsymbol{x}, 0)$ of the level set in the form of Equation (40). Figure 10b shows this subset, where the dashed line represents the stress range of the sampling space.

In this paper, the level set can be obtained through data collection and interpolation generation of numerical experiments. For a training data set of N initial samples, the stress measurements (p_i, q_i) at yield and the accumulated plastic strain ζ are first converted into signed distance functions, and then the level set of the signed distance function is generated by interpolation (see Algorithm 1).

For a simple MLP feedforward network, the network can be viewed as an approximation of the true yield function level set f_y as $\check{f}_y = \check{f}_y(p, q, \zeta | \boldsymbol{W}, \boldsymbol{b})$, with inputs p, q and hardening parameter ζ, parameterized by weights $\boldsymbol{W}$ and bias $\boldsymbol{b}$. The classical training objective following the L2 norm will only constrain the predicted yield function value. The corresponding training objective is to minimize the difference measured at N sample points, as shown in Equation (41):

$$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W}, \boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\gamma_4 \left\| f_{yi} - \check{f}_{yi} \right\|_2^2 \right) \right) \tag{41}$$

The second training objective can be modeled in terms of the H1 norm, constraining f and its first-order derivatives with respect to the stress states p and q. For a NN approximator parameterized as $\check{f}_y = \check{f}_y(p, q, \zeta | \boldsymbol{W}, \boldsymbol{b})$ using stress invariant

as input, the training objective for training samples $i \in [1, \cdots, N]$ takes the following form:

$$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W}, \boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\gamma_4 \left\| f_{yi} - \check{f}_{yi} \right\|_2^2 + \gamma_5 \left\| \frac{\partial f_{yi}}{\partial p_i} - \frac{\partial \check{f}_{yi}}{\partial p_i} \right\|_2^2 + \gamma_6 \left\| \frac{\partial f_{yi}}{\partial q_i} - \frac{\partial \check{f}_{yi}}{\partial q_i} \right\|_2^2 \right) \right) \tag{42}$$

where $\check{f}_{yi}$ corresponds to the energy of the network output, which is described by the ϕ_2^v function in MPNN_2, which consists of independent MLP containing an input layer of shape 3×30, followed by five hidden layers of shape 30×30, each hidden layer is followed by a $Tanh()$ activation layer, and the final output layer is 30×1; the total number of samples is $N = n_1 \times n_1 \times L$, where $n_1 = n_2 = 100$ represents the 100×100 orthogonality of the two components in the stress invariant space, and $L = 100$ represents 100 interpolation enhancements for each stress data point on the yield surface; γ_4, γ_5, γ_6 are weight coefficients of the different loss terms, which can be tuned according to the training accuracy and prediction effect parameters; $\gamma_4 = 10$, $\gamma_5 = \gamma_6 = 1$ are taken; the initial learning rate of training is 0.0002; the learning rate decays every 200 epochs with a decay coefficient of 0.1.

Figure 11 is the training loss curve of the network yield function, where Figure 11a,b represent the training results from Equation (41), and Figure 11c,d represent the training results from Equation (42). The left side shows the training using

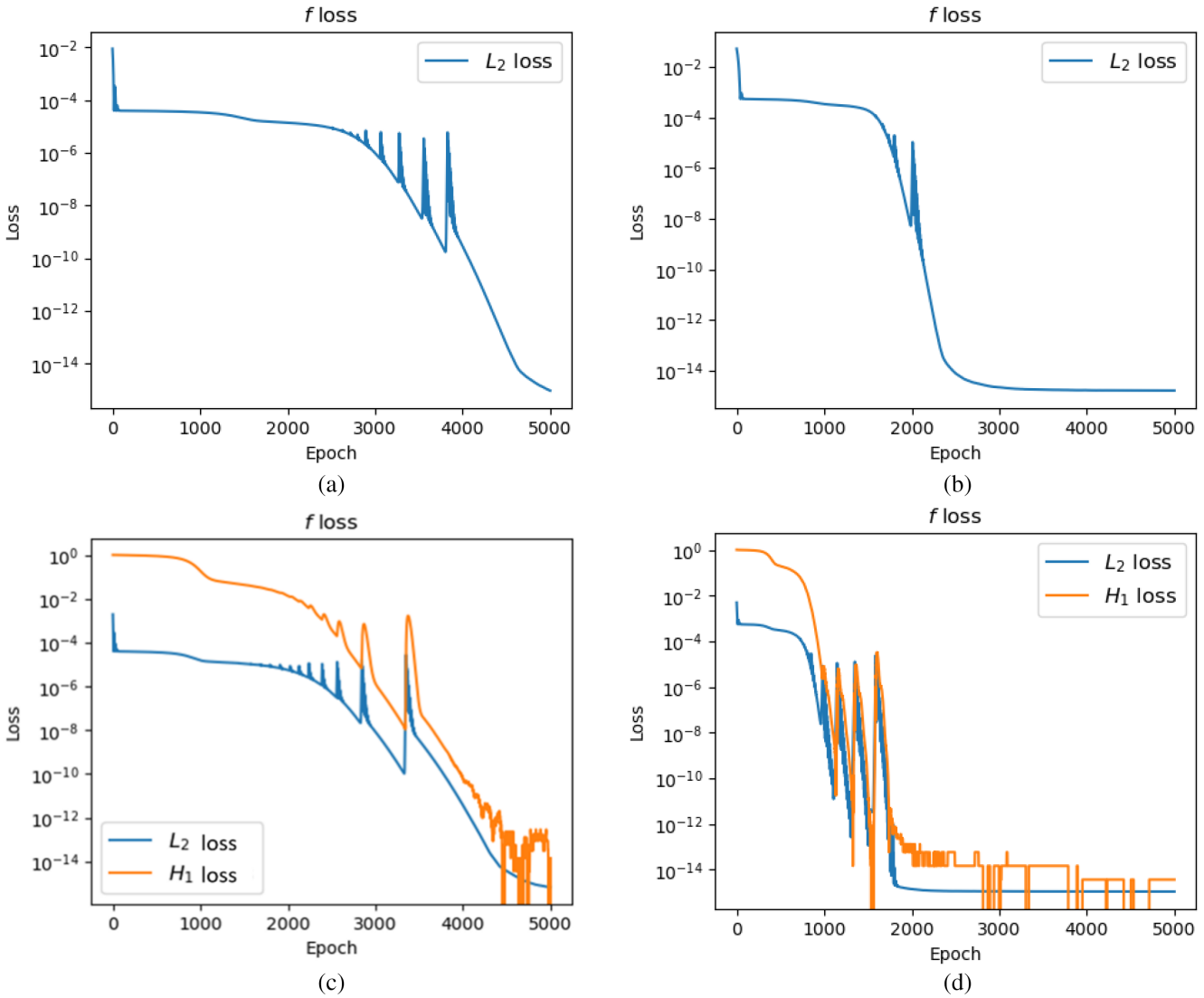


FIGURE 11 | Loss curve of yield function training under different parameters. (a) The training loss curve for the common mean squared error loss function without data processing (b) The training loss curve for the common mean squared error loss function with data normalization processing. (c) Training loss curve without data processing for loss function with added first-order partial derivative term. (d) Training loss curve with data normalization processing for loss function with added first-order partial derivative term.

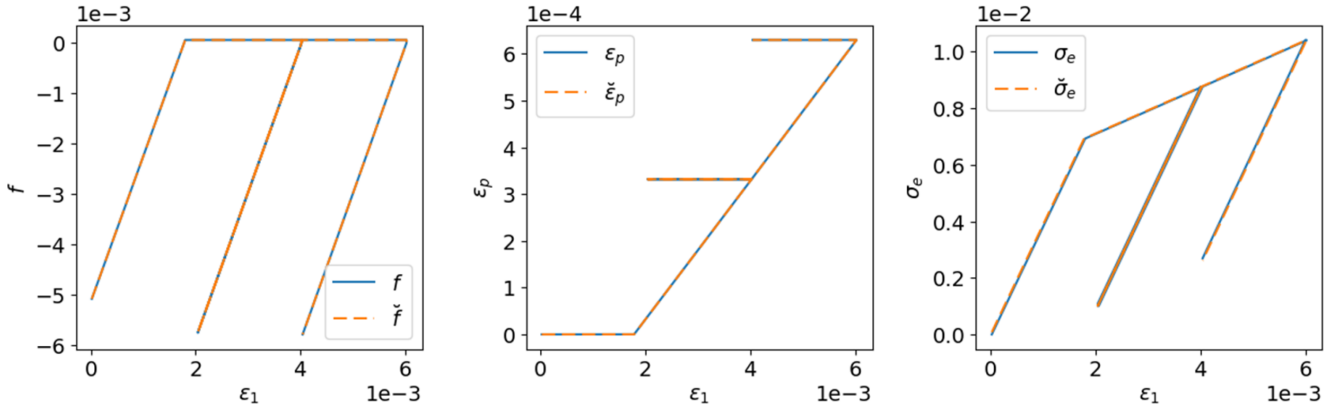


FIGURE 12 | Prediction effect of the trained network yield function on the test set. The blue solid line represents the true value, and the yellow dashed line represents the predicted value from the surrogate model. From left to right, they correspond to the predicted results of the yield function, equivalent plastic strain, and effective stress, respectively.

the original sampled data, and the right side shows the training structure after appropriate data preprocessing (normalization). As can be seen in Figure 11, data preprocessing can make the training process more efficient and convergent. At the same time, although the constraints of high-order loss will make the training process more difficult, the network approximate function is more robust and interpretable.

Figure 12 shows the prediction effect of the network yield function after training on an independent test set with a sampling number of 600. The test set is a uniaxial stress path of compression—unloading—compression. It is found from Figure 12 that the prediction accuracy of the network yield function is excellent.

4.3 | Training of Mixed Failure Function

For the third submodule (MPNN_3), $\phi^e_{3\_1}$ is the elastic energy function, and the weight parameters in ϕ^e_2 can be directly used to simplify model training and to accelerate the prediction process. The function $\phi^e_{3\_2}$ that handles damage judgment can be divided into the following two cases according to the different damage criteria in the physical model (Section 2.3).

For the brittle failure described by Equation (30), since the bond state of bond breakage is discontinuous, such as 0 or 1, it is a binary classification problem for NN judgment, and no high-order constraints are required. The training of $\phi^e_{3\_2}$ aims to minimize the binary cross entropy. The softmax layer converts the value of $\boldsymbol{X}_2$ corresponding to each edge into the fracture probability $\mathcal{P}_i$. For the training sample $i = 1 : N$, the train process can be written as follows:

$$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(y_i \cdot log\left(\mathcal{P}_i\right) + \left(1 - y_i\right) \cdot log\left(1 - \mathcal{P}_i\right) \right) \right) \tag{43}$$

where y_i is the true value of edge e_i, $y_i = 1$ means the edge is broken, and $y_i = 0$ means the edge is intact.

In this paper, $\phi^e_{3\_2}$ has a similar architecture to $\phi^e_{3\_1}$, except that the activation function of the last layer uses a different sigmoid function as the activation layer. The specific network setting consists of two independent MLPs, each of which contains an input layer with a shape of 1×20, followed by five hidden layers with a shape of 20×20, each hidden layer is followed by a $Tanh()$ activation layer, and the final output layer is 20×1. In addition, we use the Adam optimizer [66] to fine-tune the training process, with the learning rate initialized to 0.001 and reduced by 0.1 times every 100 epochs until the end of training.

For the softening damage model described by Equation (31), since the damage state of the bond is continuous from 0 to 1, it is a function approximation problem for the NN. At this time, the training of $\phi^e_{3\_2}$ aims to minimize the mean square error, and can be trained according to the following loss function:

$$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\left\| g_i - \check{g}_i \right\|_2^2 \right) \right) \tag{44}$$

in which, g_i is the damage state label value, and $\check{g}_i$ is the network prediction value.

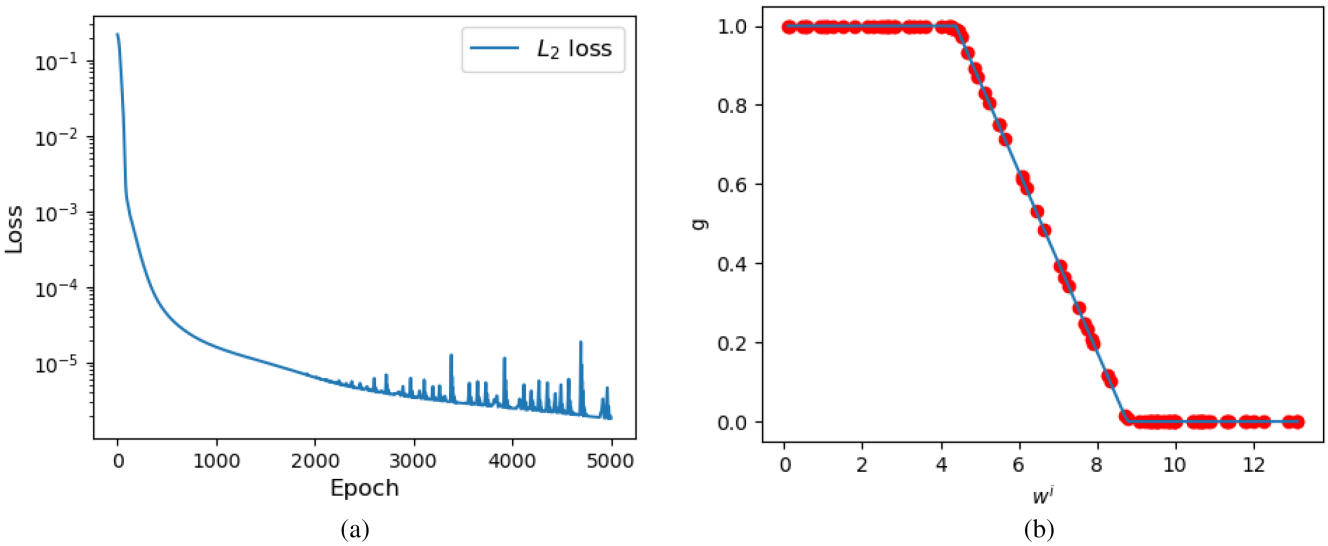


FIGURE 13 | Single factor failure function training and testing. (a) Loss curve. In (b), the blue solid line represents the true value, and the red dots represent the prediction results of the surrogate model.

In this paper, $\phi^e_{3\_2}$ consists of an independent MLP, specifically set to a 2×40 input layer, followed by 6 40×40 hidden layers, where each of the first three hidden layers is followed by a $Tanh()$ activation layer, and each of the last three hidden layers is followed by a Relu() activation layer, and finally an output layer of shape 40×1 without an activation layer. The learning rate is initialized to 0.001, and the learning rate is reduced by 0.1 every 100 epochs until the end of training.

Similarly, an independent dataset of size 40000 is used to train the damage function, which will be used in Case 1. Figure 13 shows the training loss curve and test results of the damage function of a single indicator, and the test data set is generated by independent random sampling in the strain space range, with the number of samples being $N_t = 100$. Figure 13a shows the damage function training loss curve, and Figure 13b plots the test result of the trained network on a randomly sampled data set. As shown in Figure 13b, the trained failure function makes accurate predictions on the test set, where the red dots represent prediction results by the proposed model and the blue lines represent true values.

For mixed fracture, an independent dataset of size 1000×1000 is used to train the damage function, which will be used in Cases 2 and 3. The training loss and test set validation results are shown in Figure 14. The test set is independently randomly sampled in the bond energy space, with the number of samples $N_t = 10000$.

As shown in Figure 14a, due to the increase in the complexity of the mixed cracking judgment function, its training loss decreases slowly and the minimum loss value is also large. However, after training, the network shows accurate predictions on the test data set, as shown in Figure 14b. Finally, Figure 14c shows the absolute error between the predicted value and the label value on the test set, and its absolute error is basically less than 0.02, indicating that the training NN failure function has good prediction accuracy. As a supplement, we add a failure function case using a small sample (100×100) of data with noise (0, 0.001) for training, and the number of test set samples $N_t = 1000$. The training, test and error results are plotted in Figure 14d–f. Compared with the case of using a large sample of nonnoisy data to train the failure function, except that the loss curve has significantly larger fluctuations, there is no significant difference in the test and prediction errors.

4.4 | Benchmark

A benchmark case is first performed to illustrate the convergence and prediction accuracy of the proposed method before the subsequent application examples. As illustrated in Figure 15, an isotropic square plate with a circular hole in plane stress condition is subjected to uniaxial tension along the horizontal edges [67], in which the red five-pointed star is the displacement history observation point, and the red dotted line is the displacement observation line. The dimension of the square plate is length $L = 1$ m and width $W = 1$ m. A circular hole with radius $R = 0.1$ m is located at the center of the plate. The material properties of the plate are listed as follows: Young's modulus is $E = 60$ GPa, Poisson's ratio is $\upsilon = 1/3$, and mass density is $\rho = 7000\ \mathrm{kg}/m^3$. Besides, a tensile pressure $p = 60$ MPa is applied both on the top and bottom edges.

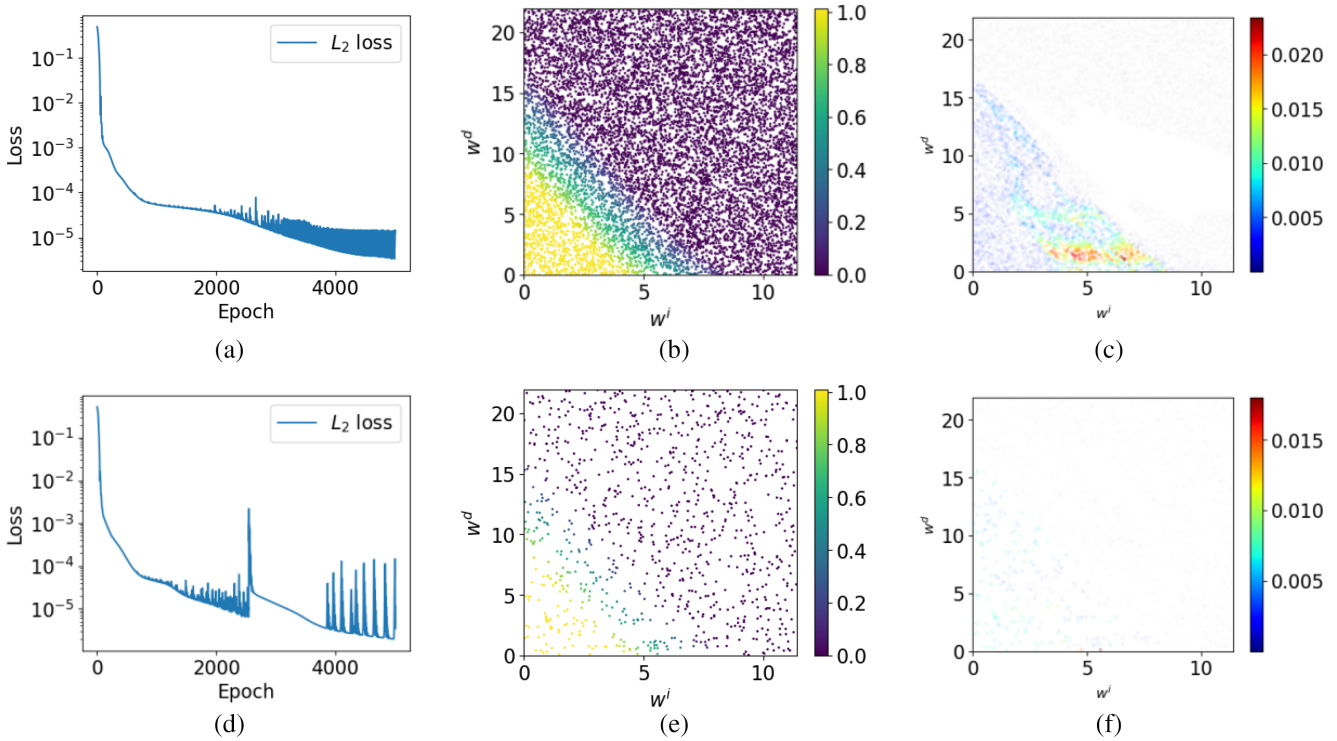


FIGURE 14 | Failure function training and testing of mixed-mode fracture (The number 1 indicates the result of a large sample of noise-free data, and the number 2 indicates the result of a small sample of noisy data). (a) Loss curve 1, (b) Test results 1, (c) Prediction error 1, (d) Loss curve 2, (e) Test results 2, (f) Prediction error 2.

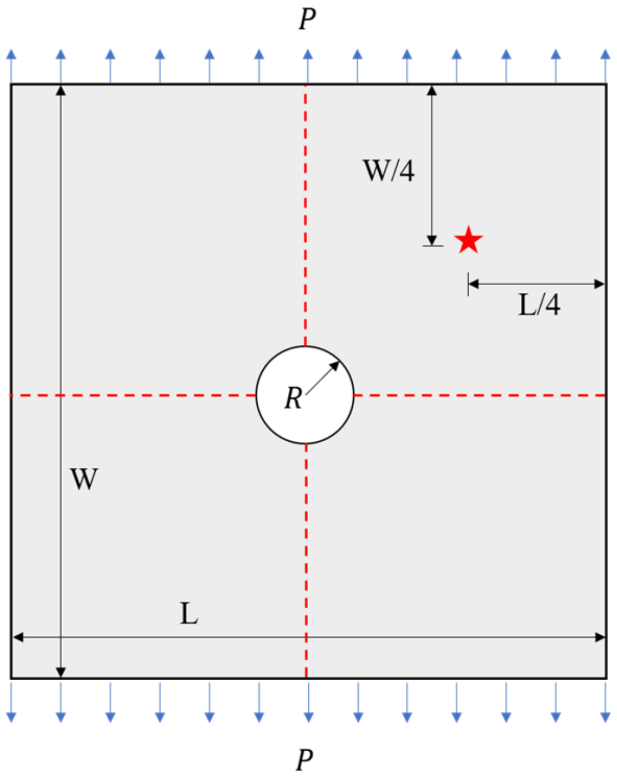


FIGURE 15 | Isotropic plate with a circular hole subjected to tensile loads.

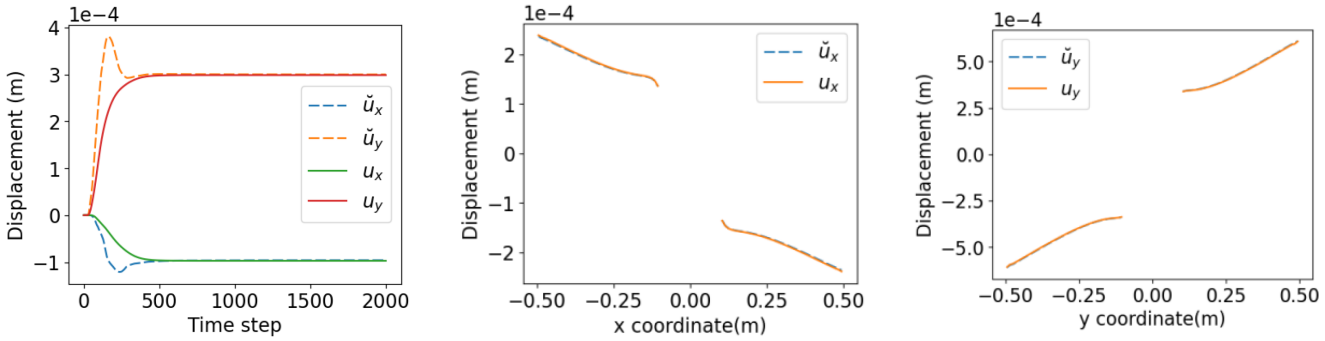


FIGURE 16 | Comparison of surrogate model prediction results $\breve{u}$ with PD results u. (a) Displacement of observation points at different time steps. (b) Horizontal centerline x-direction displacement. (c) Vertical centerline y-direction displacement.

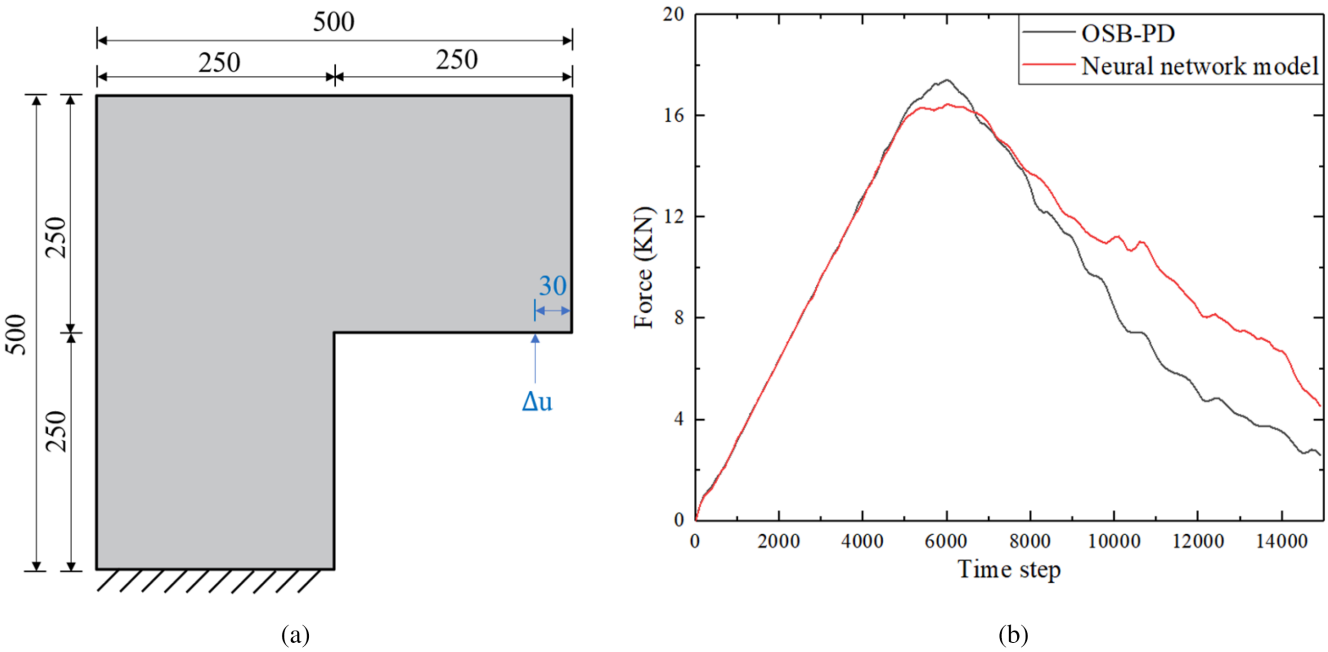


FIGURE 17 | A L-shaped plate under bending. (a) Case 1 Geometric Conditions. (b) Case 1 Comparison of load–displacement curves of two calculation methods.

The prediction effect of the model on the benchmark example is shown in Figure 16. It can be seen from Figure 16 that although the model prediction results deviate from the PD calculation results in the first 200 steps, they quickly converge to the PD calculation results in the subsequent time steps. Finally, the relative errors of the predicted displacement in the x-direction and y-direction at the observation point are 1.2% and 0.5%, respectively; the L2 error norms of the predicted displacement in the x-direction and y-direction on the observation line are 0.7% and 0.4%, respectively.

5 | Numerical Examples

5.1 | Case 1: L-Shaped Plate Subjected to Tension

In order to verify the ability of the proposed method to simulate plastic fracture problems, this section first simulates and predicts Case 1, a L-shaped tension bending plate. Figure 17a shows the geometric characteristics of the L-shaped plate [68], where the dimensions are in mm. The bottom is constrained by a fixed displacement of 0, and the right side is loaded with concentrated displacement. The distance from the vertical end face is 30 mm. The vertical line load is applied to the horizontal surface of the lower part of the horizontal leg; the direction of action is opposite to the gravity, and the

TABLE 1 | Case 1 material properties.

Young's modulus E (GPa)	Poisson's ratio υ	G_{Ic}(N/m)	Cohesion c(MPa)	Internal friction angle φ (°)
20	0.27	130	20	35

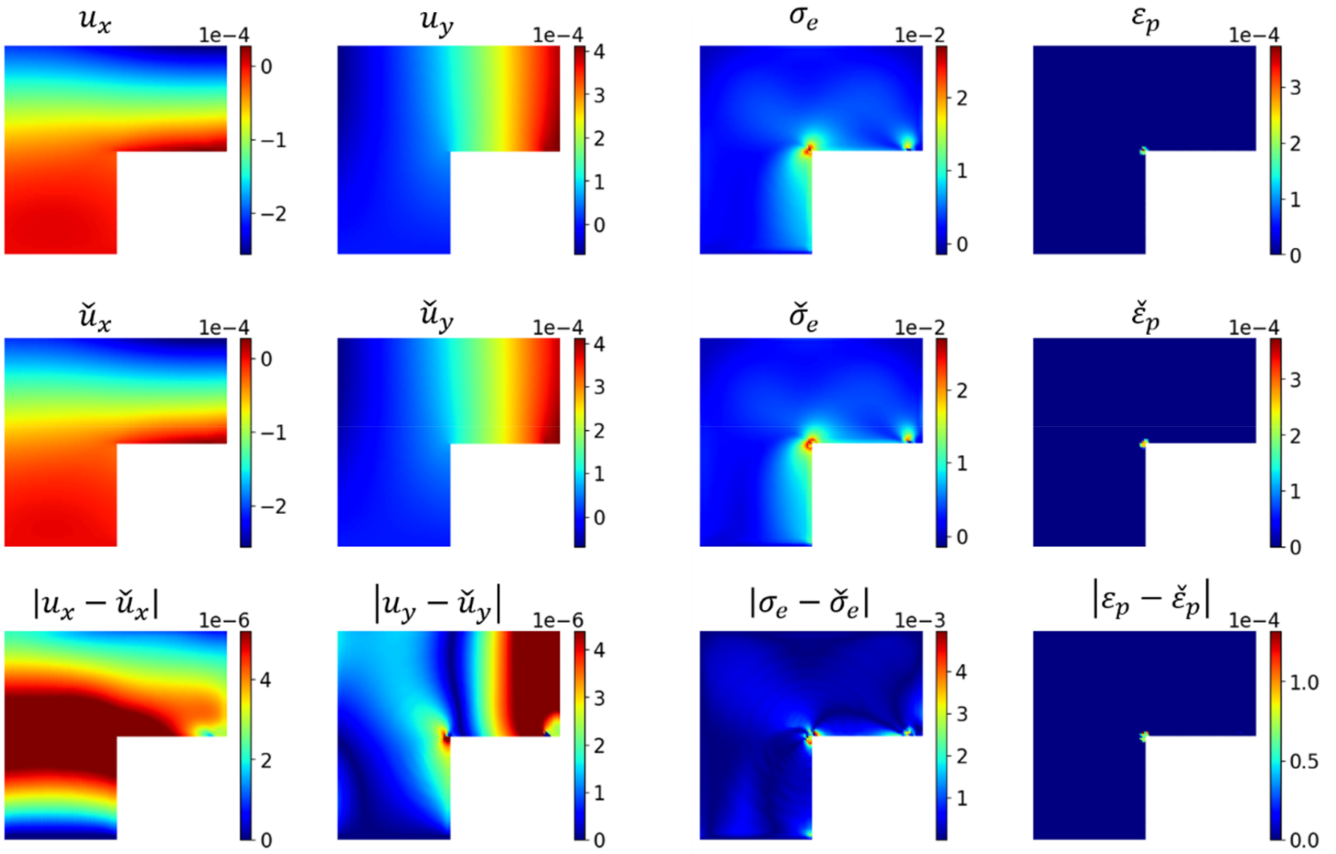


FIGURE 18 | Comparison of the proposed method and PD calculation results. (First row; left to right) PD solutions: u_x, u_y (Unit: M) ,σ_e (in GPa) and ε_p; (second row) the proposed model: $\breve{u}_x$, $\breve{u}_y$, $\breve{\sigma}_e$ and $\breve{\varepsilon}_p$; (Third rows) absolute error between PD and model: $|u_x - \breve{u}_x|$, $|u_y - \breve{u}_y|$, $|\sigma_e - \breve{\sigma}_e|$ and $|\varepsilon_p - \breve{\varepsilon}_p|$.

load is applied by displacement control. The material properties are referenced from previous studies [69], as shown in Table 1.

The material has elastic–plastic deformation behavior, and the strain hardening rule is $k_{DP}(\varepsilon_p) = k_{DP}(1 + 800\varepsilon_p)$. The simulation is carried out under the displacement control mode $\Delta u = \upsilon \times d_t \times n_t$ with $\upsilon = 1 \times 10^{-7} m/s$ and $d_t = 1s$, the material point spacing used for discretization in the PD model is $d_x = 0.005$ m, the neighborhood radius is $\delta = 3.015 \times d_x$, and a total of 7500 material points are used to discretize the computational domain. Figure 17b shows comparison of load–displacement curves obtained by PD and the proposed method. It is found from Figure 17b that load–displacement curve obtained by the proposed method is in good agreement with that obtained by OSB-PD, verifying the efficiency of the proposed method.

Figure 18 shows the crack initiation stage ($n_t = 4000$), and the network model prediction results are compared with the PD numerical results. As can be seen in Figure 18, after 4000 steps of forward prediction, the surrogate model can maintain high-precision predictions. The absolute error of the displacement field is about 1% of the true value. The maximum absolute error of the displacement in the x-direction is concentrated at the right-angle turning point of the L-shaped plate, while the maximum absolute error of the displacement in the y-direction is concentrated above the right loading end, both of which are places where the displacement components change dramatically (larger gradients). Due to the slight differences in the prediction of the network model at each step, the displacement transfer is not completely consistent

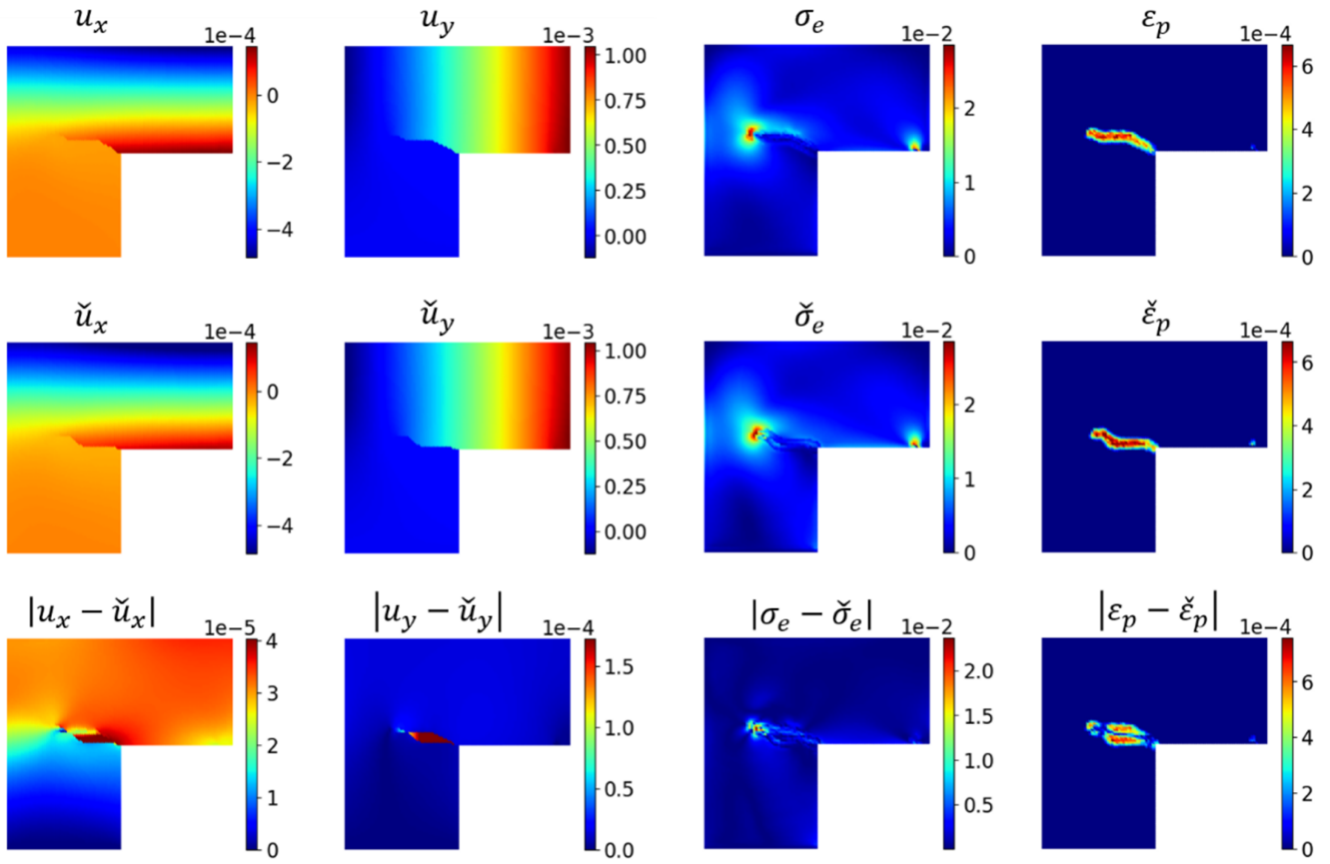


FIGURE 19 | Comparison of the final results predicted by the model and calculated by the PD code. (First row; left to right) PD solutions: u_x, u_y (Unit: M) ,σ_e (Unit: GPa) and ε_p; (second row) the proposed model: $\breve{u}_x$, $\breve{u}_y$, $\breve{\sigma}_e$ and $\breve{\varepsilon}_p$; (Third rows) absolute error between PD and model: $|u_x - \breve{u}_x|$, $\left|u_y - \breve{u}_y\right|$, $|\sigma_e - \breve{\sigma}_e|$ and $\left|\varepsilon_p - \breve{\varepsilon}_p\right|$.

with the numerical solution. After considering plasticity, the absolute error of the effective stress predicted by the proposed model is about 10% of the true value. This is mainly because the crack initiation occurs.

Figure 19 shows the comparison between the surrogate model prediction results and the PD calculation results at the end of loading ($n_t = 10000$). As shown in Figure 19, considering plasticity and fracture, after 10,000 steps of forward prediction, the surrogate model still maintains a high accuracy and successfully predicts the crack propagation path. The absolute error of the displacement field is about 10% of the true value, and the maximum error occurs near the crack propagation path. Due to the noncontinuity of the displacement on both sides of the crack, the difference in the crack path leads to a large absolute error in the displacement of the material points near the path. The distribution law of the effective stress and cumulative plastic strain predicted by the proposed model is basically consistent with the PD calculation results. Except for the slight stress concentration at the loading point, the maximum cumulative plastic strain is mainly concentrated at the crack tip and on both sides of the crack path.

Figure 20 shows the comparison of crack propagation paths at different time steps. As shown in Figure 20, the network surrogate model successfully predicts the initiation of cracks. The paths are slightly different during crack propagation. The main reason is that the displacement is slightly different during the transmission process, while the accumulated plastic strain and damage judgment are irreversible effects. As shown in Figure 20, although the crack path angle predicted by the surrogate model is smaller than the PD numerical calculation result at $n_t = 6000$ steps, and it leads to continuous differences in the crack path at the subsequent $n_t = 8000$ steps, the crack path obtained by the proposed model is close to the PD numerical result at $n_t = 10000$ steps. Overall, the crack path of this example is successfully predicted.

To quantitatively evaluate the prediction accuracy of the surrogate model, Table 2 shows the L_2 error norms between the predicted physical field quantities at two steps ($n_t = 4000$ and $n_t = 10000$) and the results calculated by the OSB—PD code. Since the crack propagation path in this example is not stable, although the crack propagation directions predicted by the

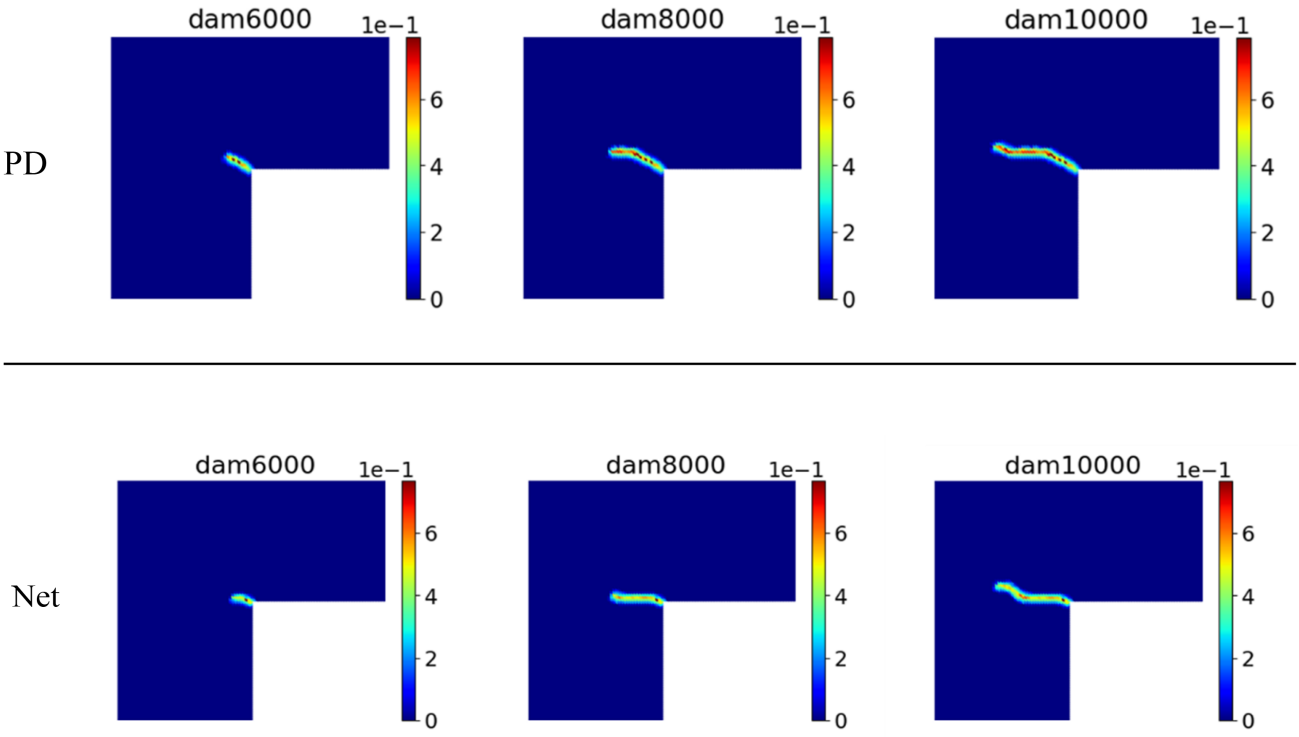


FIGURE 20 | Comparison of crack path prediction results at different time steps.

TABLE 2 | L_2 error norms of surrogacy model prediction results at different time steps in case 1.

Time steps	$L_2(u_x)$	$L_2(u_y)$	$L_2(\sigma_e)$	$L_2(\varepsilon_p)$
4000	4.16%	1.6%	3.5%	7.1%
10,000	11.8%	5.3%	27.53%	68.2%

two methods are basically the same, there is a certain deviation. As a result, the plastic state quantity mainly distributed along the crack propagation path has a large L_2 error norm.

5.2 | Case 2: The Plate Containing Multicracks Subjected to Compression

The proposed model is further demonstrated to be capable of simulating the propagation and penetration of complex cracks under compression. Case 2 is a plate with multiple cracks [29] under uniaxial compression. The geometric conditions are $L = 0.2$ m, $H = 0.2$ m, $a = 0.02$ m, $b = 0.05$ m, and $c = 0.07$ m, as shown in Figure 21. The simulation is carried out in a displacement control mode of $\Delta u = v \times d_t \times n_t$, with each displacement increment $v = 5 \times 10^{-8} m/s$, $d_t = 1s$ and a maximum value $n_t = 5000$. The material point spacing used for discretization is $d_x = 0.0013$ m, the neighborhood radius is $\delta = 3.015 \times d_x$, and a total of 23,400 material points are used to discretize the computational domain. The corresponding material property values are shown in Table 3. Figure 21b shows comparison of load–displacement curves obtained by OSB-PD and the proposed method. It is found from Figure 21b that load–displacement curve obtained by the proposed method is in good agreement with OSB-PD result, verifying the efficiency of the proposed method.

Figure 22 shows the comparison between the network model prediction results and the PD numerical results before crack propagation ($n_t = 2000$). As shown in Figure 22, after 2000 steps of forward prediction, the physical engine model can be learned to maintain high-precision predictions. The absolute error of the displacement field in the y direction is about 1% of the true value, and the maximum error is concentrated near the crack tip; while the maximum absolute error of the displacement in the x direction is concentrated on the two side boundaries, and the absolute value is slightly larger. When the crack propagates, the proposed model predicts the effective stress and cumulative plastic strain with good accuracy,

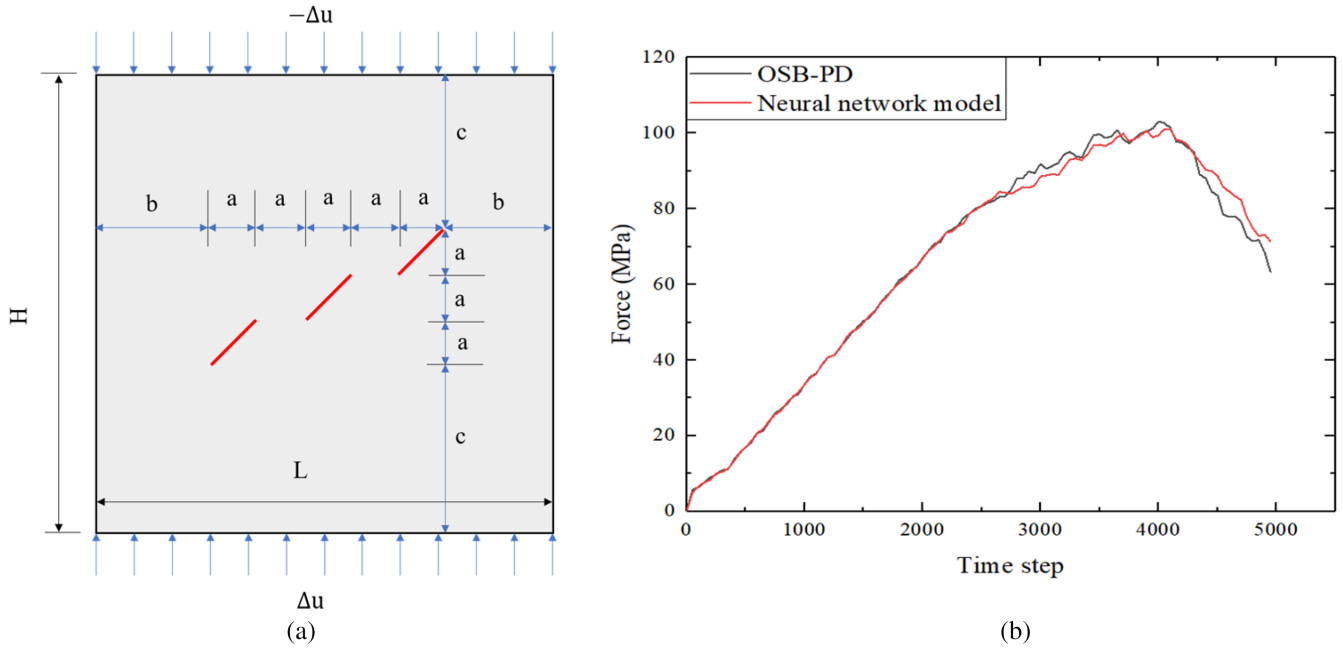


FIGURE 21 | A plate containing multicracks subjected to compression. (a) Case 2 Geometric Conditions. (b) Case 2 Comparison of load–displacement curves of two calculation methods.

TABLE 3 | Case 2 material properties.

Young's modulus E (GPa)	Poisson's ratio υ	G_{Ic}(N/m)	G_{IIc}(N/m)	Cohesion c(MPa)	Internal friction angle φ (°)
20	0.27	50	150	10	30

successfully reproducing the distribution law of effective stress and plastic strain at the prefabricated crack tip, with an absolute error between the PD results and predicted result being about 5%.

Figure 23 shows the displacement predicted by the model for the final case 2 ($n_t = 5000$) and the final result of PD calculation. As shown in Figure 23, the displacement field predicted by the proposed model maintains a high accuracy, and the maximum absolute error is one order of magnitude lower than the true value. The maximum error is concentrated near the new crack on the right, which is mainly caused by the strong discontinuity at the crack propagation path. The effective stress and accumulated plastic strain predicted by the network model are highly consistent with the results obtained by PD, and the maximum absolute error is concentrated near the new crack, which is about 10% of the true value.

Figure 24 shows the comparison between the crack propagation path predicted by the proposed model and the PD calculation results at different time steps ($n_t = 3000, n_t = 4000, n_t = 5000$) for Case 2. As can be seen in Figure 24, at different time steps, the crack path predicted by the proposed model maintains a high consistency with the PD calculation results. The crack propagates in two modes, namely, tensile and shear modes, from the prefabricated crack tip. The direction perpendicular to the prefabricated crack is controlled by wing crack, while the direction parallel to the prefabricated crack is controlled by shear crack.

Table 4 shows the L_2 error norms of the surrogate model predictions and the OSB-PD code calculation results for Case 2 at two steps. As shown in Table 4, the proposed model's prediction accuracy is good when the crack begins to propagate, and the accuracy increases slightly as the crack propagates. Since the crack propagation path in this example is relatively stable, the final error accumulation is not significant.

5.3 | Case 3: Analysis of Slope Instability

In order to illustrate the ability of the proposed method to deal with plastic crack propagation problems, this section relies on an actual engineering slope to perform numerical calculations and predictive analysis of the proposed network model.

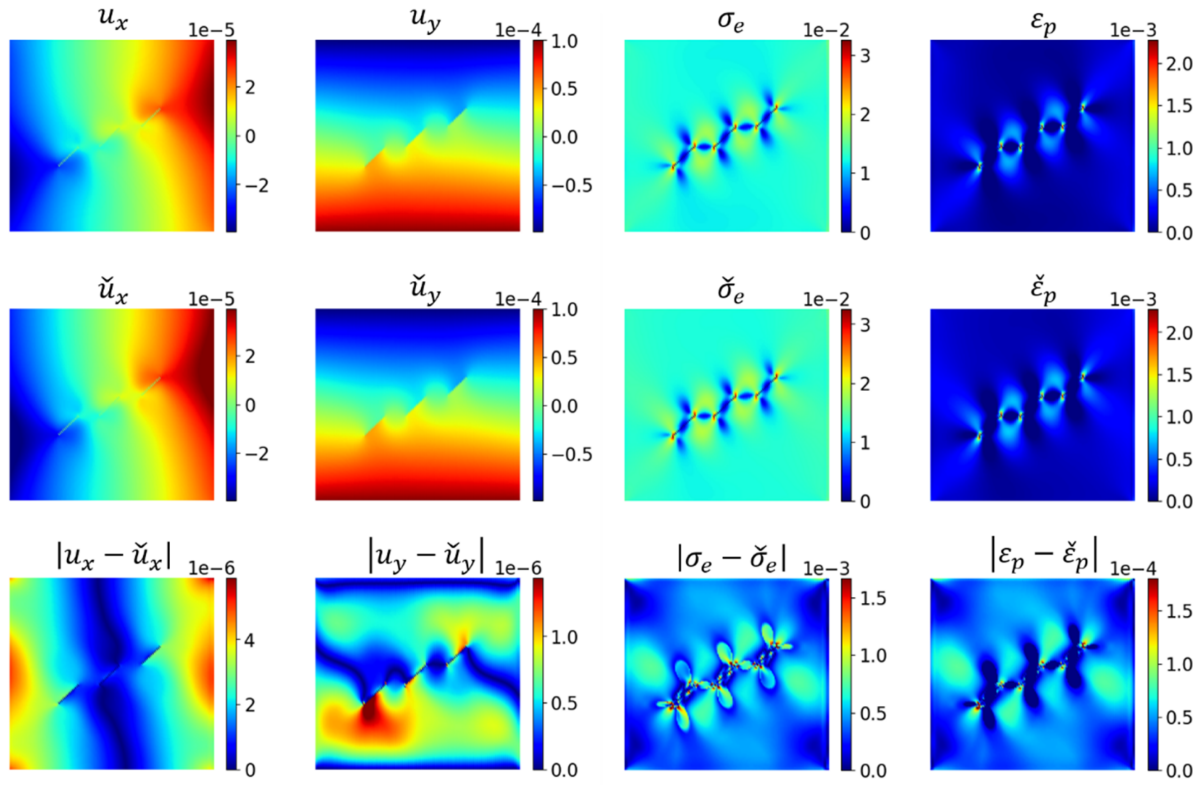


FIGURE 22 | Comparison of the proposed model prediction results and PD calculation results for Case 2. (First row; left to right) PD solutions: u_x, u_y (Unit: M) ,σ_e (Unit: GPa) and ε_p; (second row) the proposed model: $\breve{u}_x$, $\breve{u}_y$, $\breve{\sigma}_e$ and $\breve{\varepsilon}_p$; (Third rows) absolute error between PD and model: $|u_x - \breve{u}_x|$, $|u_y - \breve{u}_y|$, $|\sigma_e - \breve{\sigma}_e|$ and $|\varepsilon_p - \breve{\varepsilon}_p|$.

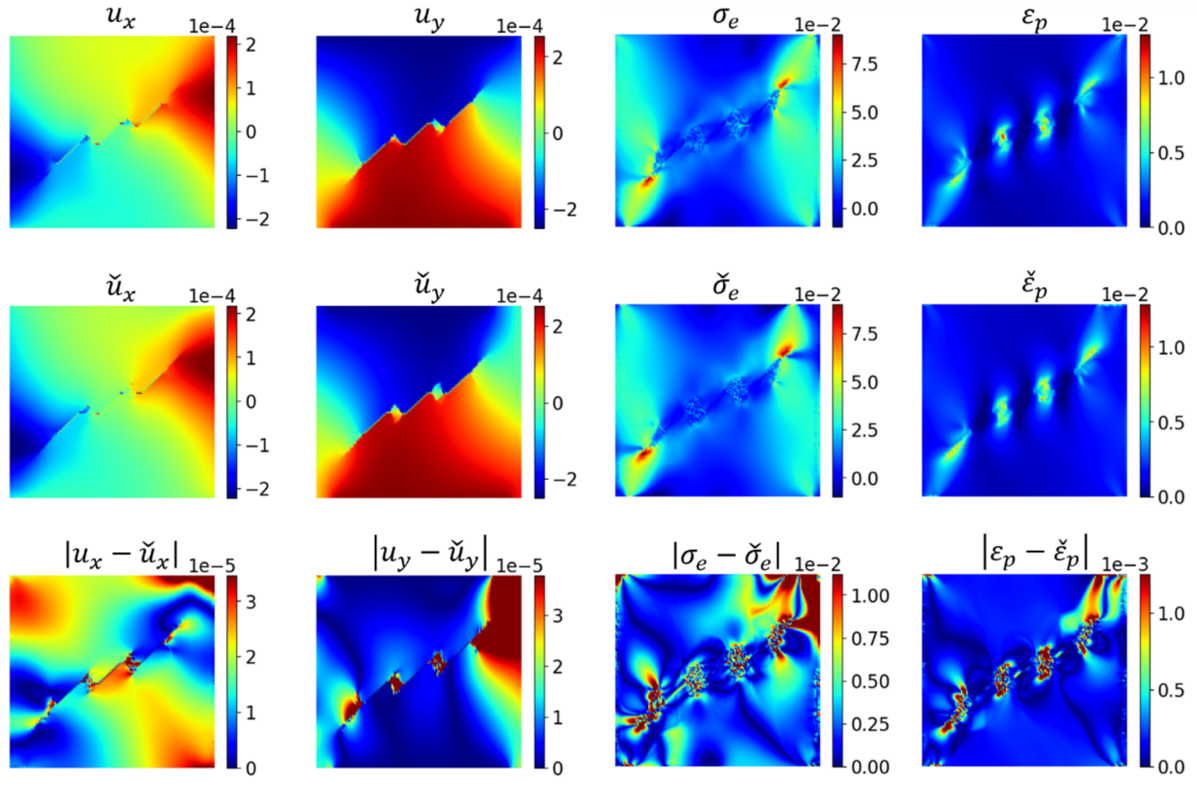


FIGURE 23 | Comparison of the final results predicted by the model in Case 2 and the PD calculation results. (First row; left to right) PD solutions: u_x, u_y (Unit: M) ,σ_e (Unit: GPa) and ε_p; (second row) the proposed model: $\breve{u}_x$, $\breve{u}_y$, $\breve{\sigma}_e$ and $\breve{\varepsilon}_p$; (Third rows) absolute error between PD and model: $|u_x - \breve{u}_x|$, $|u_y - \breve{u}_y|$, $|\sigma_e - \breve{\sigma}_e|$ and $|\varepsilon_p - \breve{\varepsilon}_p|$.

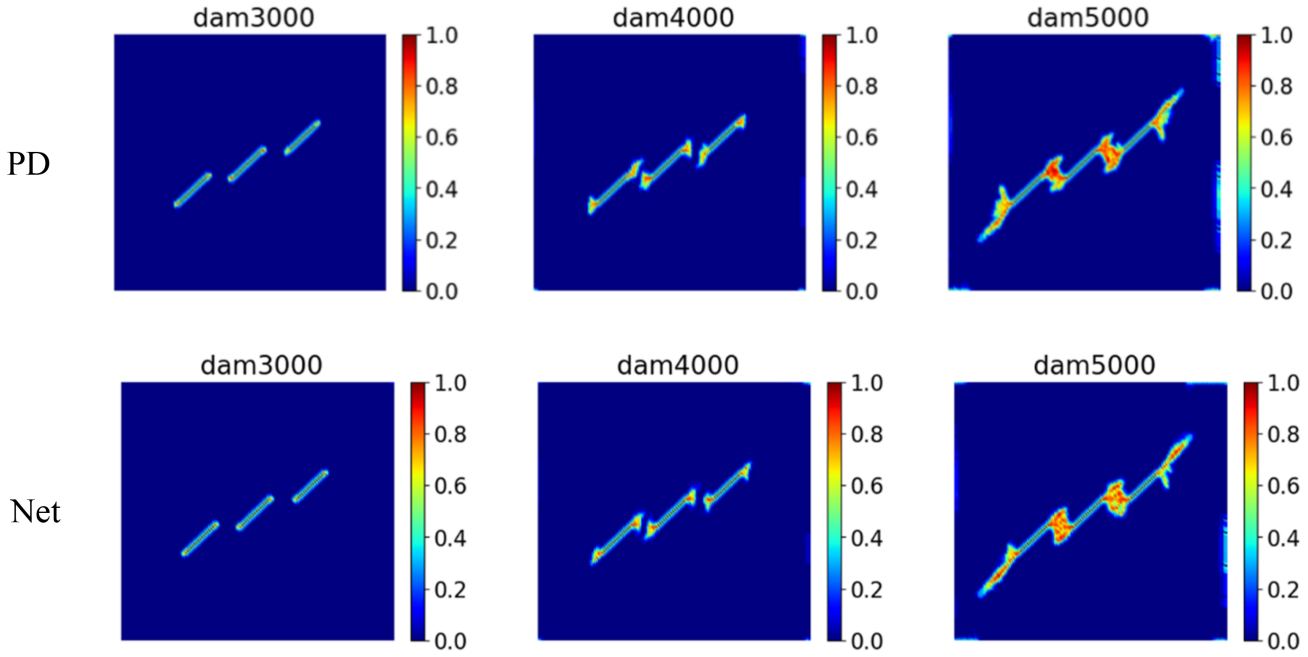


FIGURE 24 | Comparison of crack propagation process.

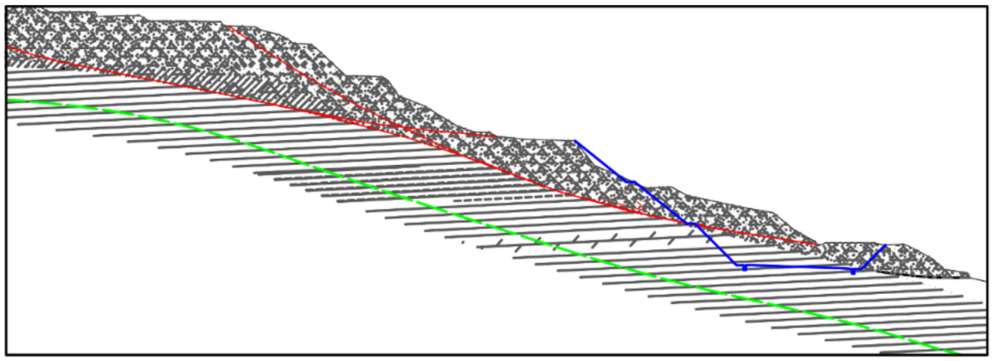
TABLE 4 | L_2 error norms of surrogacy model prediction results at different time steps in case 2.

Time steps	$L_2(u_x)$	$L_2(u_y)$	$L_2(\sigma_e)$	$L_2(\varepsilon_p)$
2000	6.7%	1.1%	3.4%	5.7%
5000	10.9%	6.2%	10.5%	8.9%

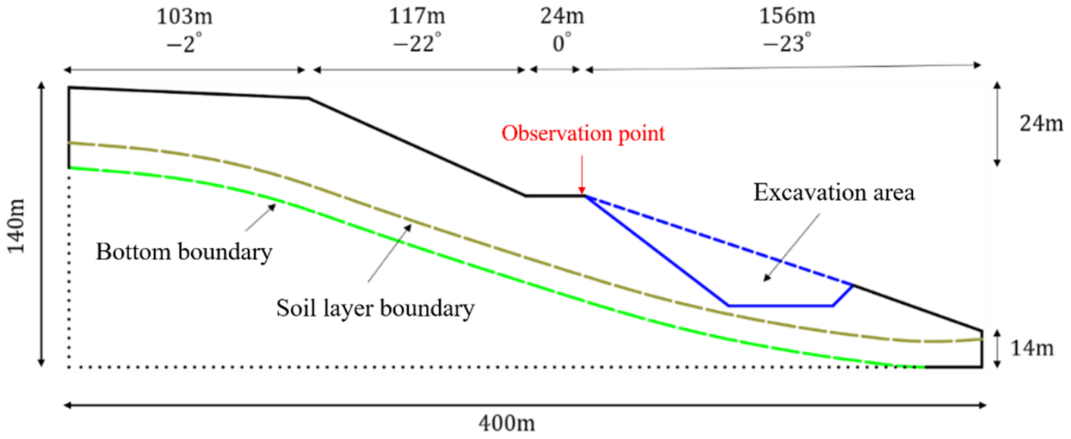
Figure 25a shows the excavation profile design of a highway embankment, where the blue line represents the embankment profile line after the designed excavation, and the red line represents the sliding surface after the original natural slope begins to deform and become unstable after rainfall. The original slope profile is a three-step shape. To simplify the problem, this example only considers the lower two slopes where serious instability occurs. The surface of the slope is mainly composed of gravel blocks, silty clay and gravel, and the base layer at the bottom of the slope is mainly shale, carbonaceous shale and mudstone. Figure 25b shows the simplified slope geometry, where the green dashed line represents the bottom boundary of the slope in numerical modeling, the light green dashed line represents the bottom boundary of the surface-breaking accumulation layer, the blue line represents the excavation area, and the slope geometry parameters are shown in Figure 25b. To optimize the calculation efficiency, the area between the green dashed line and the black dashed line is not considered in the modeling. Figure 25c is the numerical model after the computational domain is discretized, in which s_c represents the strength reduction factor of different materials.

Yield function parameter $\alpha_{DP} = \frac{2\sin(\varphi)}{\sqrt{3}(3+\sin(\varphi))}$, initial value of hardening parameter $K_{DP} = \frac{6c\cos(\varphi)}{\sqrt{3}(3+\sin(\varphi))}$, and hardening criterion $K_{DP}(\zeta) = K_{DP}(1 + 800\zeta)$. Other PD numerical parameters are as follows: material point spacing $d_x = 0.5$, neighborhood range $\delta = 3.015 \times d_x$, and the total number of material points is 61,805. The calculation condition is divided into two stages: in the first stage, the displacement and stress are calculated by applying the self-weight of soils; in the second stage, the displacement and plastic state as well as slope instability and failure are calculated considering construction excavation ($n_t = 1000$), and the total calculation time step is $n_t = 1500$. According to the geological survey results, the material properties of the slope body are shown in Table 5.

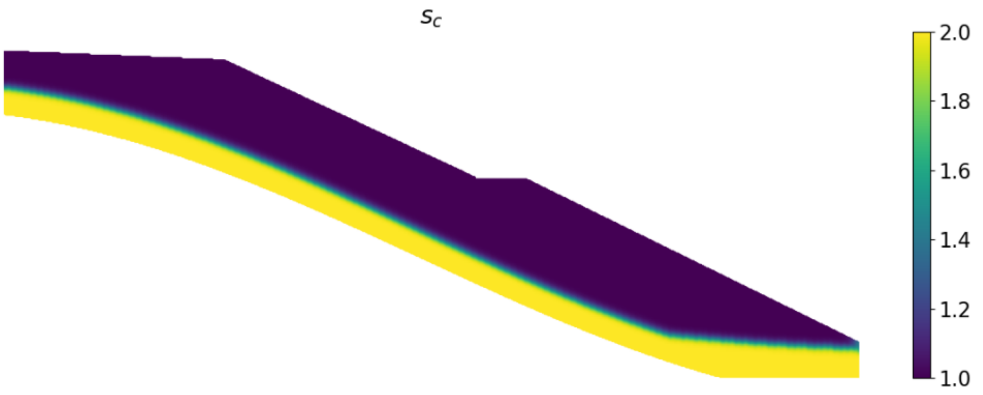
Figure 26 shows the displacement record curve of the new slope top formed after the excavation of the natural slope, and its location is marked by the red arrow in Figure 25b. As shown in Figure 26, the displacement of the observation point increase significantly after $n_t > 1300$, indicating that the slope becomes unstable. Compared with the numerical results obtained by OSB-PD, the displacement of the observation point in the surrogate model increases more slowly.



(a)



(b)



(c)

FIGURE 25 | Plot of engineering slope. (a) Geological conditions of slope engineering cases (b) Geometric dimensions of the numerical model (c) Distribution of strength coefficient s_c.

TABLE 5 | Case 3 material properties.

Young's modulus E (GPa)	Poisson's ratio υ	G_{Ic} (N/m)	G_{IIc} (N/m)	Cohesion c (MPa)	Internal friction angle φ (°)
0.015	0.28	7.2	16.8	0.013	14

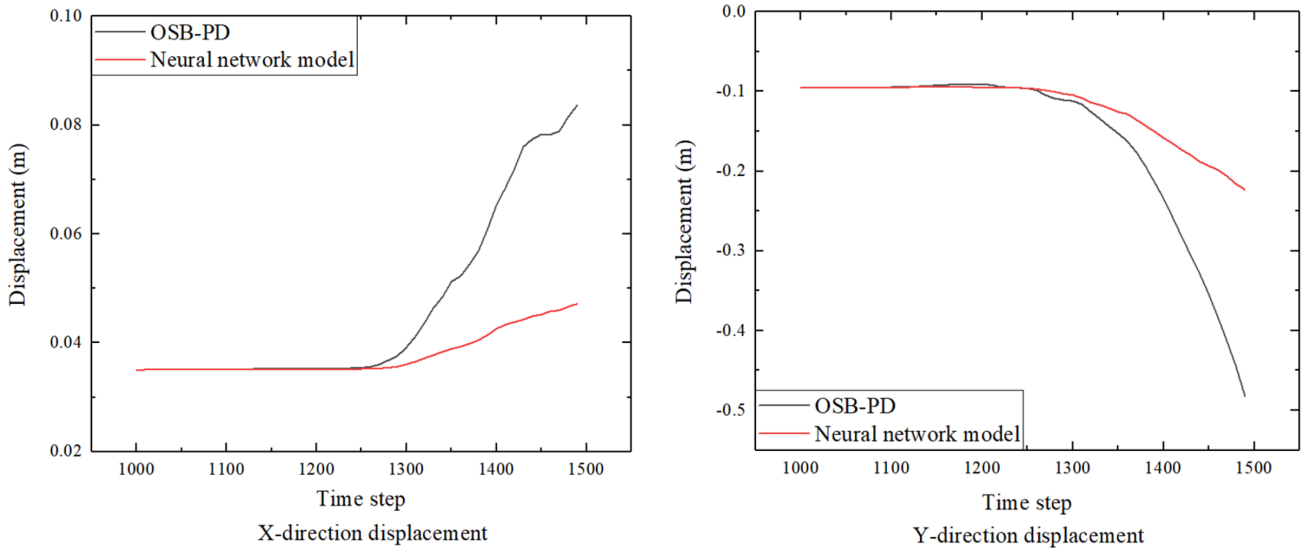


FIGURE 26 | Displacement records of observation point.

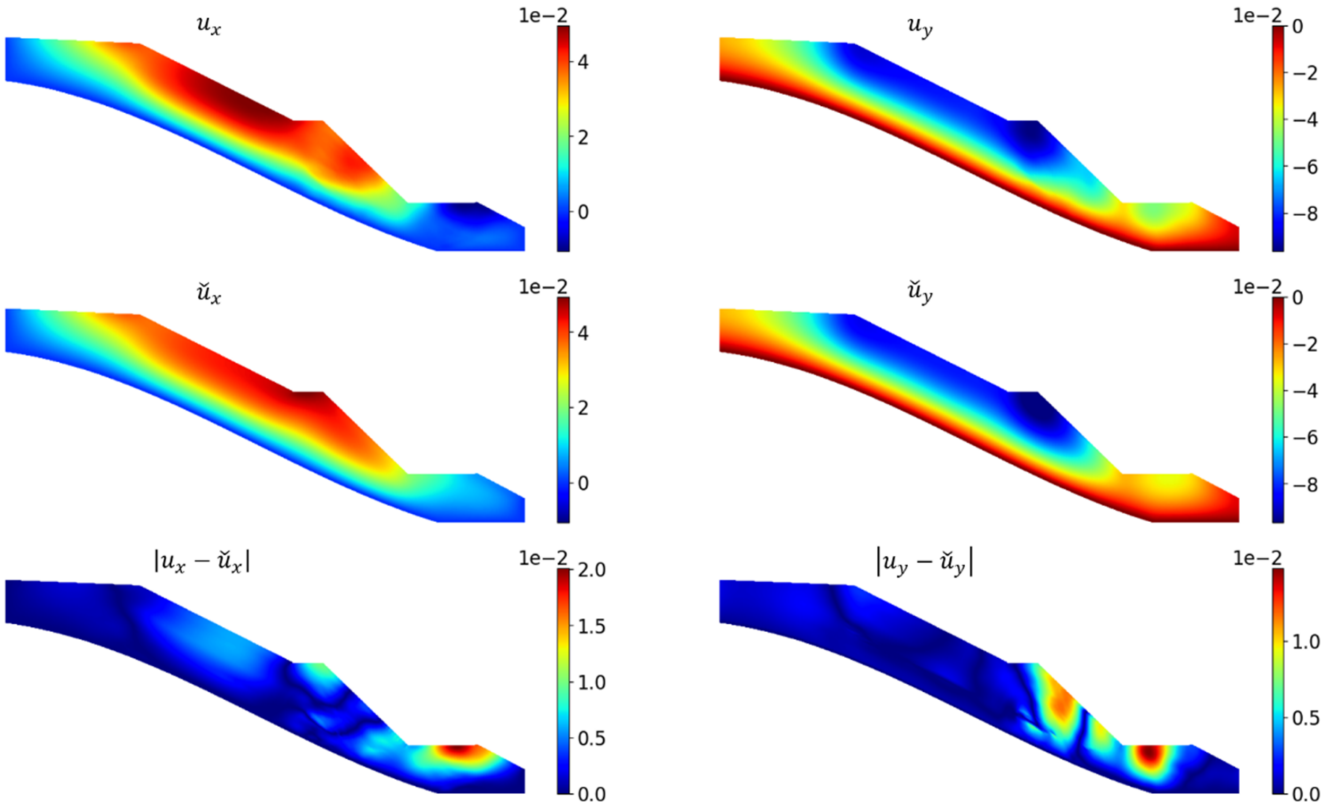


FIGURE 27 | Comparison between the proposed model prediction displacement $\breve{u}$ (Unit:M) and PD calculation u after slope excavation.

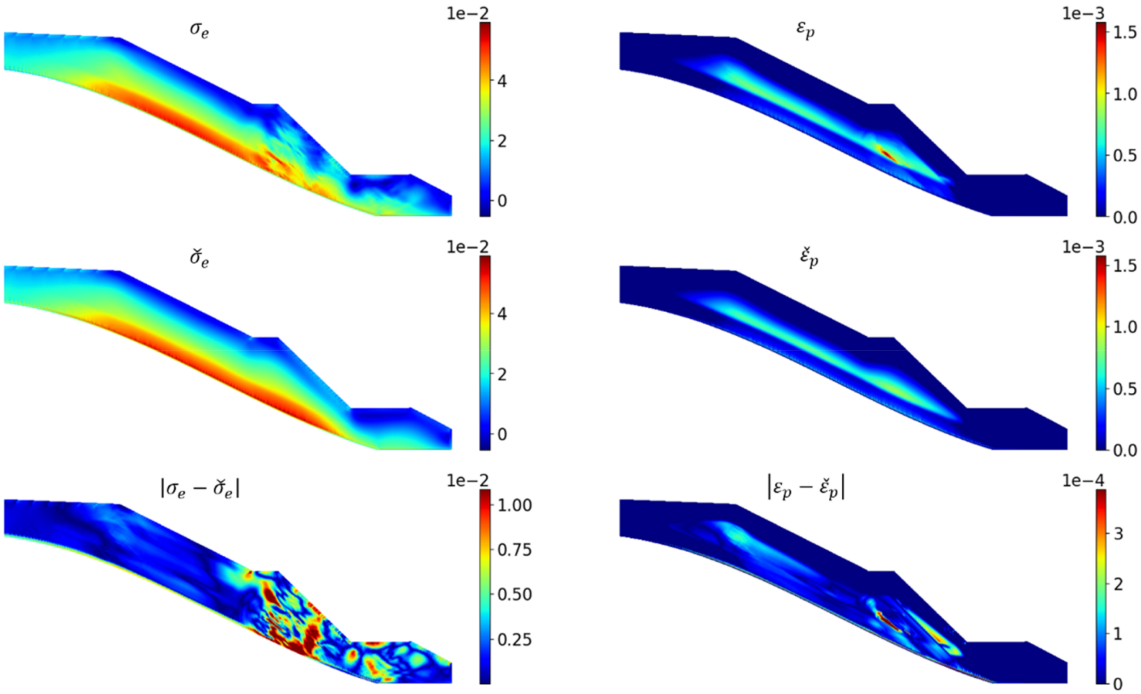
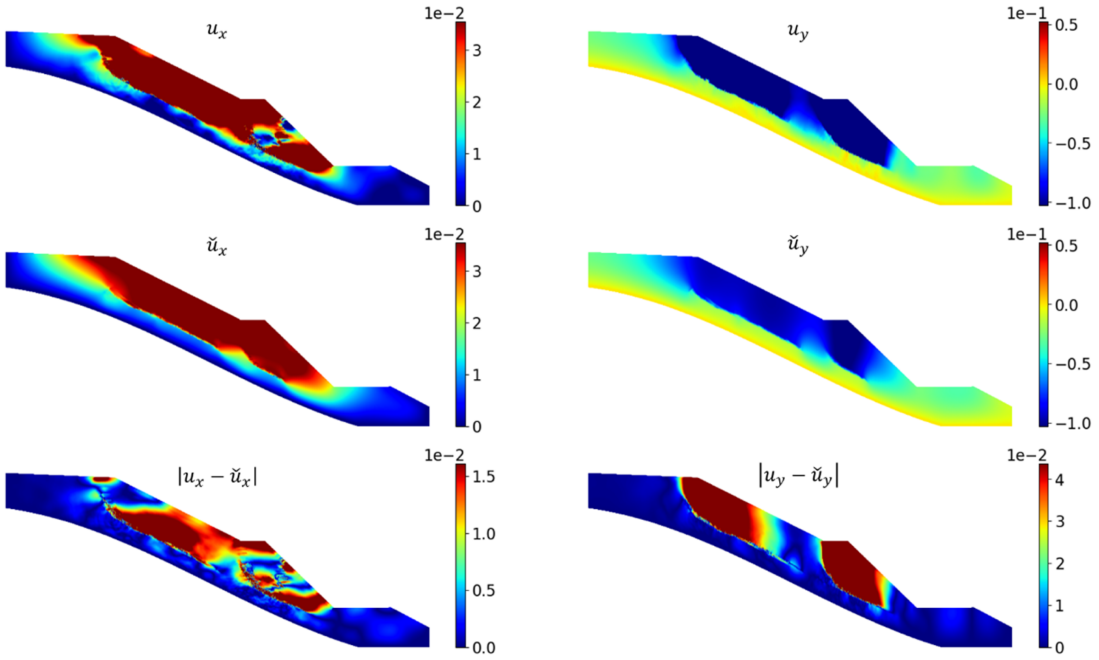


FIGURE 28 | Comparison of the proposed model prediction $\check{\sigma}_e$ (Unit：MPa) and $\check{\varepsilon}_p$, and PD calculation σ_e (Unit：MPa) and ε_p after slope excavation.

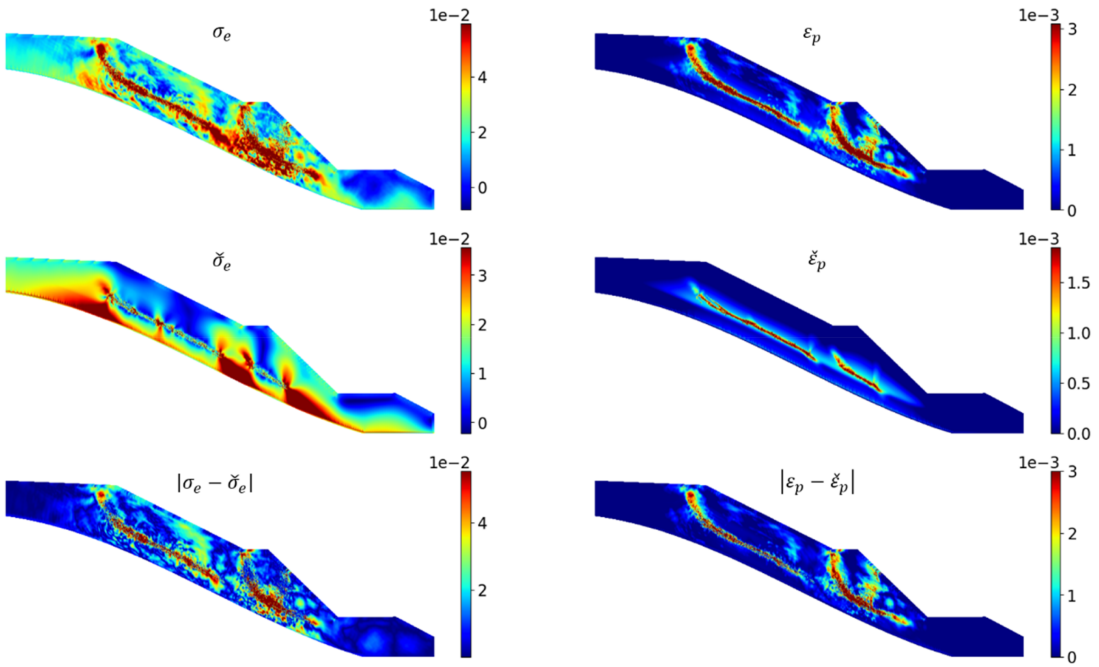
Figure 27 shows the comparison of the proposed model prediction displacement and PD calculation results after the roadbed slope is excavated ($n_t = 1200$). The first line is the displacement component contour u_x and u_y calculated by PD, the second line is the prediction result of the proposed network model, and the third line is the absolute error between the proposed model prediction result and the PD calculation result. It is found from Figure 27 that after 1200 steps of forward prediction, the network model still maintains a high accuracy, and the proposed model prediction accuracy of the main deformation area of the slope after excavation is good. The maximum error in the x direction is concentrated at the excavation of the slope foot. The reason is that the displacement gradient of the PD calculation result at the newly formed slope foot after excavation is more concentrated, while the proposed network model prediction result is more continuous and smooth; the maximum error in the y direction has a certain error on the inner side of the newly excavated slope in addition to the newly formed slope foot.

Figure 28 shows a comparison of the model-predicted effective stress, cumulative plastic strain and PD calculation results after the excavation of the roadbed slope ($n_t = 1200$). The first line is the PD result σ_e and ε_p, the second line is the proposed model prediction result, and the third line is the prediction error. As can be seen in Figure 28, the distribution laws of the proposed model prediction effective stress field and the cumulative plastic strain are basically consistent. The PD calculation result has a larger extreme value, while the proposed model prediction result is relatively more uniform and smoother. The maximum error is concentrated in the lower half of the slope where the deformation is severe. The maximum cumulative plastic strain error is basically below 10% except for the area with the most concentrated plastic strain.

Figure 29 shows a comparison between the proposed model prediction results and PD calculation results after the excavation of the roadbed slope. Figure 29a shows the displacement field result; the first line is the PD calculated displacement component, and the second line is the proposed model prediction result. Similarly, Figure 29b shows the effective stress and cumulative plastic strain results. As can be seen in Figure 30, when the final slope is unstable and damaged, although the displacement value predicted by the proposed model is smaller than the PD calculation result, the instability area of the slope is basically the same. The main reason for the difference is that the damage extension area predicted by the proposed model is smaller than the PD calculation result, resulting in a large difference in displacement after instability and damage. From the plastic strain contour, it can be seen that the cumulative plastic strain concentration zone predicted by the proposed model is basically consistent with the PD calculation result.



(a)



(b)

FIGURE 29 | Comparison between the final instability result of slope excavation predicted by the proposed model and the PD calculation result. (a) Displacement (Unit:M), $\breve{u}$ represents the prediction result of the surrogate model, and u represents the calculation result of the PD code (b) Equivalent stress (Unit:MPa) and accumulated plastic strain, $\breve{\sigma}_e$ and $\breve{\varepsilon}_p$ represent the prediction result of the surrogate model, σ_e and ε_p represent the PD calculation result.

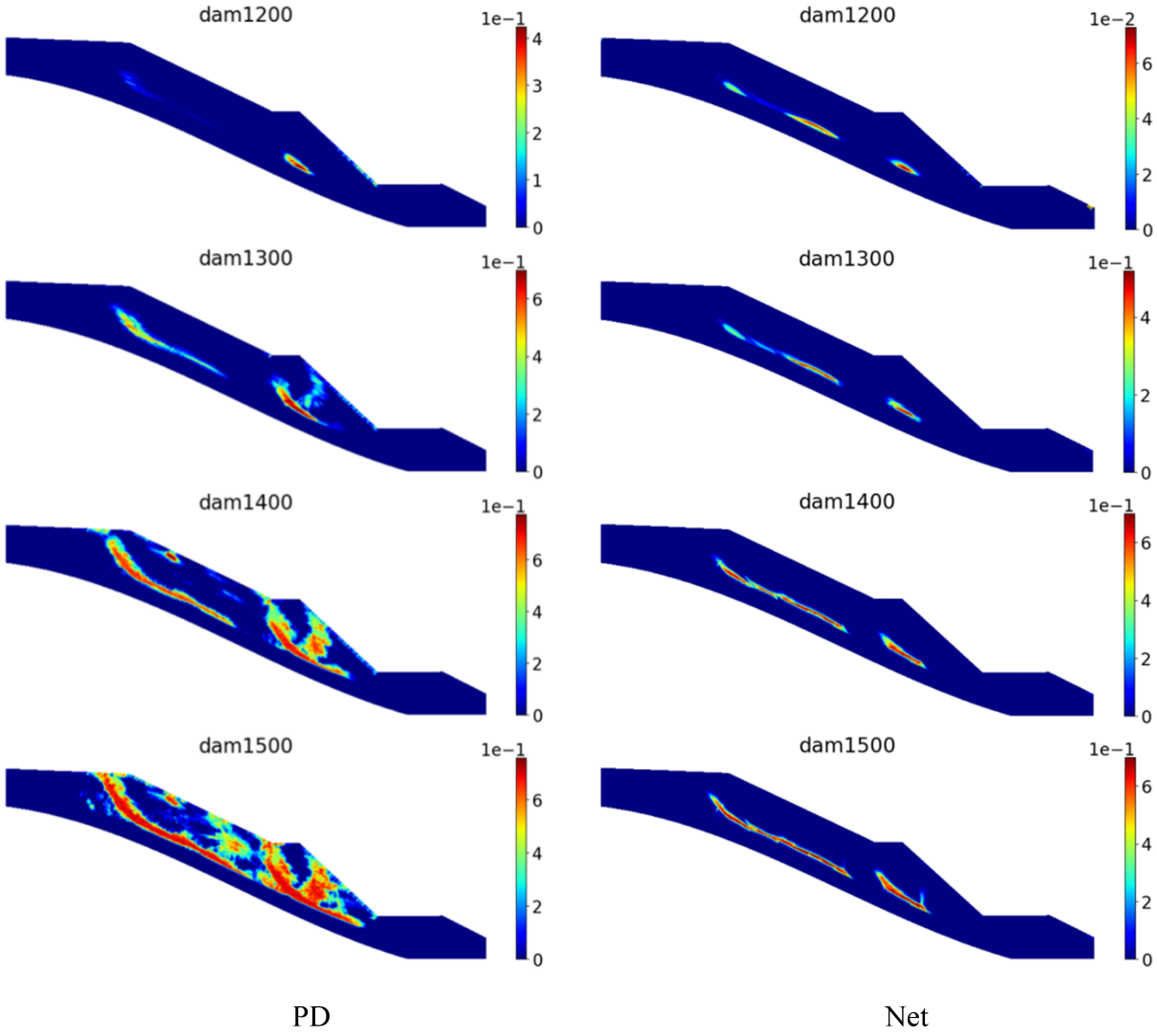


FIGURE 30 | Comparison of damage at different time steps.

TABLE 6 | L_2 error norms of surrogacy model prediction results at different time steps in case 3.

Time steps	$L_2(u_x)$	$L_2(u_y)$	$L_2(\sigma_e)$	$L_2(\varepsilon_p)$
1200	6.9%	2.7%	5.5%	8.7%
1500	25.4%	34.9%	44.5%	50.4%

Figure 30 shows the comparison between the damage predicted by the proposed model and the PD calculation results at different time steps after the excavation of the roadbed slope. As shown in Figure 30, after the roadbed excavation, compression-shear cracks initiate in the lower part of the original natural slope. As time increases, shear-tensile cracks initiate in the upper part of the slope, and the compression-shear cracks in the lower part of the slope propagate toward the surface of the slope. In the subsequent process of slope instability development, the upper cracks coalesce with lower cracks, causing the slope to completely lose stability and damage. The damage process predicted by the proposed model is basically consistent with the PD calculation results.

Table 6 shows the L_2 error norms of the two-step alternative model predictions and the OSB-PD solution in case 3. It is found from Table 6 that the proposed model's prediction accuracy is good when the crack begins to propagate ($n_t = 1200$). However, the prediction error increases significantly when the slope completely fails ($n_t = 1500$). This main reason is that the slope failure yields the unbalanced sliding force and causes the slope instability, ultimately leading to the high prediction error. However, the ultimate failure mode remains consistent.

TABLE 7 | Comparison of computational efficiency.

Cases	Number of material points	Calculating time steps	Average computation time (s)		Efficiency advantage
			OSB-PD	The proposed model	
Case1	7500	15,000	1768	184.6	9.6X
Case2	23,400	5000	1876	52.5	35.7X
Case3	61,805	1500	1488	33.3	44.7X

5.4 | Advantage of Computational Efficiency

To illustrate the efficiency advantage of the proposed method, the computational cost of the two schemes in the three cases mentioned above is compared. For each case, the two schemes are calculated 10 times and the average value is considered. Parameters that are strongly correlated with the computational cost, namely the number of material points and the computational time step, are given. The specifications of the computing hardware are as follows: the central processing unit (CPU) is an AMD 5950X, featuring 16 cores and 32 threads, and the graphics processing unit (GPU) is an RTX 3080 model with 12GB of video memory. Table 7 shows the computational efficiency of the surrogate model is 10 to dozens of times faster than the benchmark OSB-PD code, and its efficiency advantage increases with increasing the number of material points.

6 | Conclusions

In this paper, a novel GNN-based plastic fracture surrogate model is proposed to derive the OSB-PD Drucker-Prager model by incorporating interpretable components into the training process. To ensure the accuracy of the model prediction, we first perform Sobolev training on the energy function. Subsequently, the yield function is trained using level sets generated from historical data to evaluate the yield behavior of the material. By applying Newton iteration to the yield function of the network, we determine the plastic increment and update the plastic state accordingly. Based on the above research, this paper couples the data-trained hybrid failure network module to realize the plastic-fracture coupling model. Numerical results show that the displacement field, equivalent plastic stress, and cumulative plastic strain predicted by the proposed model are consistent with those obtained by the OSB-PD model. Finally, the application value of the proposed method in practical engineering is demonstrated through the case of slope excavation instability. Future work will consider introducing new training methods, such as attention mechanisms, to improve long-range prediction accuracy.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (No. W2511036), the Natural Science Foundation of Chongqing, China (No. CSTB2023NSCQ-LZX0025), and the Fundamental Research Funds for the Central Universities (No. 2025CDJZKCGJ-01)

Funding

This work was supported by the Natural Science Foundation of Chongqing, China (CSTB2023NSCQ-LZX0025), the National Natural Science Foundation of China (W2511036), and the Fundamental Research Funds for the Central Universities (2025CDJZKCGJ-01).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation* 9, no. 8 (1997): 1735–1780.

2. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature* 521, no. 7553 (2015): 436–444.

3. S. Q. Wu, G. Meunier, O. Chadebec, Q. X. Li, and L. Chamoin, "Learning Dynamics of Nonlinear Field-Circuit Coupled Problems With a Physics-Data Combined Model," *International Journal for Numerical Methods in Engineering* 126, no. 5 (2025): e70015.

4. T. Kirchdoerfer and M. Ortiz, "Data-Driven Computational Mechanics," *Computer Methods in Applied Mechanics and Engineering* 304 (2016): 81–101.

5. M. A. Bessa, R. Bostanabad, Z. Liu, et al., "A Framework for Data-Driven Analysis of Materials Under Uncertainty: Countering the Curse of Dimensionality," *Computer Methods in Applied Mechanics and Engineering* 320 (2017): 633–667.

6. R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, and M. Ortiz, "Model-Free Data-Driven Inelasticity," *Computer Methods in Applied Mechanics and Engineering* 350 (2019): 81–99.

7. X. L. Yu and X. P. Zhou, "A Data-Driven Bond-Based Peridynamic Model Derived From Group Method of Data Handling Neural Network With Genetic Algorithm," *International Journal for Numerical Methods in Engineering* 123, no. 22 (2024): 5618–5651.

8. A. Galetzka, D. Loukrezis, and D. G. Herbert, "Data-Driven Solvers for Strongly Nonlinear Material Response," *International Journal for Numerical Methods in Engineering* 122, no. 6 (2021): 1538–1562.

9. M. I. Latypov and S. R. Kalidindi, "Data-Driven Reduced Order Models for Effective Yield Strength and Partitioning of Strain in Multiphase Materials," *Journal of Computational Physics* 346 (2017): 242–261.

10. A. L. Frankel, R. E. Jones, C. Alleman, and J. A. Templeton, "Predicting the Mechanical Response of Oligocrystals With Deep Learning," *Computational Materials Science* 169 (2019): 109099.

11. O. L. Kafka, C. Yu, M. Shakoor, Z. L. Liu, G. J. Wagner, and W. K. Liu, "Data-Driven Mechanistic Modeling of Influence of Microstructure on High-Cycle Fatigue Life of Nickel Titanium," *JOM* 70, no. 7 (2018): 1154–1158.

12. X. Liu, F. Gasco, J. Goodsell, and W. B. Yu, "Initial Failure Strength Prediction of Woven Composites Using a New Yarn Failure Criterion Constructed by Deep Learning," *Composite Structures* 230 (2019): 111505.

13. Z. L. Liu, C. T. Wu, and M. Koishi, "A Deep Material Network for Multiscale Topology Learning and Accelerated Nonlinear Modeling of Heterogeneous Materials," *Computer Methods in Applied Mechanics and Engineering* 345 (2019): 1138–1168.

14. M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa, "Deep Learning Predicts Path-Dependent Plasticity," *Proceedings of the National Academy of Sciences* 116, no. 52 (2019): 26414–26420.

15. M. Stoffel, F. Bamer, and B. Markert, "Neural Network Based Constitutive Modeling of Nonlinear Viscoplastic Structural Response," *Mechanics Research Communications* 95 (2019): 85–88.

16. Y. Heider, K. Wang, and W. C. Sun, "SO(3)-invariance of Informed-Graph-Based Deep Neural Network for Anisotropic Elastoplastic Materials," *Computer Methods in Applied Mechanics and Engineering* 363 (2020): 112875.

17. A. Koeppe, F. Bamer, and B. Markert, "An Intelligent Nonlinear Meta Element for Elastoplastic Continua: Deep Learning Using a New Time-Distributed Residual U-Net Architecture," *Computer Methods in Applied Mechanics and Engineering* 366 (2020): 113088.

18. N. N. Vlassis and W. C. Sun, "Sobolev Training of Thermodynamic-Informed Neural Networks for Interpretable Elasto-Plasticity Models With Level Set Hardening," *Computer Methods in Applied Mechanics and Engineering* 377 (2021): 113695.

19. N. N. Vlassis and W. C. Sun, "Component-Based Machine Learning Paradigm for Discovering Rate-Dependent and Pressure-Sensitive Level-Set Plasticity Models," *Journal of Applied Mechanics* 89, no. 2 (2021): 1–13.

20. N. N. Vlassis and W. C. Sun, "Geometric Learning for Computational Mechanics Part II: Graph Embedding for Interpretable Multiscale Plasticity," *Computer Methods in Applied Mechanics and Engineering* 404 (2023): 115768.

21. M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden Fluid Mechanics: Learning Velocity and Pressure Fields From Flow Visualizations," *Science* 367, no. 6481 (2020): 1026–1030.

22. E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A Physics-Informed Deep Learning Framework for Inversion and Surrogate Modeling in Solid Mechanics," *Computer Methods in Applied Mechanics and Engineering* 379 (2021): 113741.

23. D. W. Abueidda, S. Koric, E. Guleryuz, and N. A. Sobh, "Enhanced Physics-Informed Neural Networks for Hyperelasticity," *International Journal for Numerical Methods in Engineering* 124, no. 7 (2023): 1585–1601.

24. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic Differentiation in Machine Learning: A Survey," *Journal of Machine Learning Research* 18, no. 1 (2017): 5595–5637.

25. J. N. Fuhg and N. Bouklas, "The Mixed Deep Energy Method for Resolving Concentration Features in Finite Strain Hyperelasticity," *Journal of Computational Physics* 451 (2022): 110839.

26. E. Haghighat, A. C. Bekar, E. Madenci, and R. Juanes, "A Nonlocal Physics-Informed Deep Learning Framework Using the Peridynamic Differential Operator," *Computer Methods in Applied Mechanics and Engineering* 385 (2021): 114012.

27. D. W. Abueidda, Q. Y. Lu, and S. Koric, "Meshless Physics-Informed Deep Learning Method for Three-Dimensional Solid Mechanics," *International Journal for Numerical Methods in Engineering* 122, no. 23 (2021): 7182–7201.

28. E. Haghighat and R. Juanes, "SciANN: A Keras/TensorFlow Wrapper for Scientific Computations and Physics-Informed Deep Learning Using Artificial Neural Networks," *Computer Methods in Applied Mechanics and Engineering* 373 (2021): 113552.

29. M. Manav, R. Molinaro, S. Mishra, and L. De Lorenzis, "Phase-Field Modeling of Fracture With Physics-Informed Deep Learning," *Computer Methods in Applied Mechanics and Engineering* 429 (2024): 117104.

30. S. Goswami, M. L. Yin, Y. Yu, and G. E. Karniadakis, "A Physics-Informed Variational DeepONet for Predicting Crack Path in Quasi-Brittle Materials," *Computer Methods in Applied Mechanics and Engineering* 391 (2022): 114587.

31. X. L. Yu and X. P. Zhou, "A Nonlocal Energy-Informed Neural Network Based on Peridynamics for Elastic Solids With Discontinuities," *Computational Mechanics* 73, no. 2 (2024): 233–255.

32. X. P. Zhou and X. L. Yu, "Transfer Learning Enhanced Nonlocal Energy-Informed Neural Network for Quasi-Static Fracture in Rock-Like Materials," *Computer Methods in Applied Mechanics and Engineering* 430 (2024): 117226.

33. G. Muraoka, T. Toyoshi, Y. Arai, and Y. Wada, "Crack Propagation Prediction Using Partial Differential Equations-Based Neural Network With Discovered Governing Equations," *International Journal for Numerical Methods in Engineering* 126, no. 1 (2025): e7665.

34. E. Samaniego, C. Anitescu, S. Goswami, et al., "An Energy Approach to the Solution of Partial Differential Equations in Computational Mechanics via Machine Learning: Concepts, Implementation and Applications," *Computer Methods in Applied Mechanics and Engineering* 362 (2020): 112790.

35. L. Qiu, F. J. Wang, Y. J. Liang, and Q. H. Qin, "Physics-Informed Radial Basis Function Network Based on Hausdorff Fractal Distance for Solving Hausdorff Derivative Elliptic Problems," *Computers and Mathematics with Applications* 183 (2025): 271–286.

36. X. J. Huang, F. J. Wang, B. R. Zhang, and H. Q. Liu, "Enriched Physics-Informed Neural Networks for Dynamic Poisson-Nernst-Planck Systems," *Mathematics and Computers in Simulation* 237 (2025): 231–246.

37. F. J. Wang, X. Li, H. Q. Liu, L. Qin, and X. X. Yue, "An Adaptive Method of Fundamental Solutions Using Physics-Informed Neural Networks," *Engineering Analysis with Boundary Elements* 178 (2025): 106295.

38. A. M. Roy, G. Suman, S. Veera, and A. Raymundo, "Physics-Infused Deep Neural Network for Solution of Non-Associative Drucker–Prager Elastoplastic Constitutive Model," *Journal of the Mechanics and Physics of Solids* 185 (2024): 105570.

39. J. L. Fu, D. H. Xiao, R. Fu, et al., "Physics-Data Combined Machine Learning for Parametric Reduced-Order Modelling of Nonlinear Dynamical Systems in Small-Data Regimes," *Computer Methods in Applied Mechanics and Engineering* 404 (2023): 115771.

40. X. Y. Pan and D. H. Xiao, "Domain Decomposition for Physics-Data Combined Neural Network Based Parametric Reduced Order Modelling," *Journal of Computational Physics* 519 (2024): 113452.

41. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks* 20, no. 1 (2009): 61–80.

42. M. Defferrard, X. Bresson, P. Vandergheynst, et al., "Convolutional Neural Networks on Graphs With Fast Localized Spectral Filtering," 2016 Advances in Neural Information Processing Systems 29.

43. J. Gilmer, S. Samuel S, R. F. Patrick, O. Vinyals, and D. George E, "Neural Message Passing for Quantum Chemistry," 2017 34th International Conference on Machine Learning, PMLR.

44. X. P. Zhou and K. Feng, "MPNN Based Graph Networks as Learnable Physics Engines for Deformation and Crack Propagation in Solid Mechanics," *International Journal of Solids and Structures* 291 (2024): 112695.

45. B. Feng and X. P. Zhou, "The Novel Physics-Enhanced Graph Neural Network for Phase-Field Fracture Modelling," *Computer Methods in Applied Mechanics and Engineering* 446 (2025): 118284.

46. T. Xie, A. France-Lanord, Y. M. Wang, Y. Shao-Horn, and J. C. Grossman, "Graph Dynamical Networks for Unsupervised Learning of Atomic Scale Dynamics in Materials," *Nature Communications* 10, no. 1 (2019): 2667.

47. A. Gruber, M. Gunzburger, L. L. Ju, and Z. Wang, "A Comparison of Neural Network Architectures for Data-Driven Reduced-Order Modeling," *Computer Methods in Applied Mechanics and Engineering* 393 (2022): 114764.

48. T. Pfaff, M. Fortunato, A. Sanchez-Gonzales, and P. Battaglia, "Learning Mesh-Based Simulation With Graph Networks," preprint, arXiv, 2020,:2010.03409.

49. A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, "Learning to Simulate Complex Physics With Graph Networks," 2020 PMLR, International Conference on Machine Learning.

50. Y. Choi and K. Kumar, "Graph Neural Network-Based Surrogate Model for Granular Flows," *Computers and Geotechnics* 166 (2024): 106015.

51. M. Dai, M. F. Demirel, Y. Liang, and J. M. Hu, "Graph Neural Networks for an Accurate and Interpretable Prediction of the Properties of Polycrystalline Materials," *npj Computational Materials* 7, no. 1 (2021): 103.

52. N. Black and A. R. Najafi, "Learning Finite Element Convergence With the Multi-Fidelity Graph Neural Network," *Computer Methods in Applied Mechanics and Engineering* 397 (2022): 115120.

53. M. Maurizi, C. Gao, and F. Berto, "Predicting Stress, Strain and Deformation Fields in Materials and Structures With Graph Neural Networks," *Scientific Reports* 12, no. 1 (2022): 21834.

54. B. Feng and X. P. Zhou, "The Novel Graph Transformer-Based Surrogate Model for Learning Physical Systems," *Computer Methods in Applied Mechanics and Engineering* 432 (2024): 117410.

55. K. Feng and X. P. Zhou, "A Novel Graph Networks Based Learnable Physics Engines for Crack Propagation and Coalescence in Solid Mechanics," *Engineering Fracture Mechanics* 315 (2025): 110800.

56. X. P. Zhou and K. Feng, "The Novel Learnable Physics Engines for Interpretable Elastoplastic Models of Geomaterials Based on the Message Passing Neural Network," *International Journal of Rock Mechanics and Mining Sciences* 194 (2025): 106244.

57. S. A. Silling, M. Epton, O. Weckner, J. Xue, and E. Askari, "Peridynamic States and Constitutive Modeling," *Journal of Elasticity* 88, no. 2 (2007): 151–184.

58. Y. Wang and W. Wu, "A Bond-Level Energy-Based Peridynamics for Mixed-Mode Fracture in Rocks," *Computer Methods in Applied Mechanics and Engineering* 414 (2023): 116169.

59. F. Han, G. Lubineau, Y. Azdoud, and A. Askari, "A Morphing Approach to Couple State-Based Peridynamics With Classical Continuum Mechanics," *Computer Methods in Applied Mechanics and Engineering* 301 (2016): 336–358.

60. T. Zhang, X. P. Zhou, and Q. H. Qian, "Drucker-Prager Plasticity Model in the Framework of OSB-PD Theory With Shear Deformation," *Engineering With Computers* 39, no. 2 (2023): 1395–1414.

61. H. Zhang, X. Zhang, and P. Z. Qiao, "A New Peridynamic Mixed-Mode Bond Failure Model for Interface Delamination and Homogeneous Materials Fracture Analysis," *Computer Methods in Applied Mechanics and Engineering* 379 (2021): 113728.

62. Q. V. Bui, "A Modified Benzeggagh-Kenane Fracture Criterion for Mixed-Mode Delamination," *Journal of Composite Materials* 45, no. 4 (2010): 389–413.

63. S. J. Liu, Y. T. Wang, C. Peng, and W. Wu, "A Thermodynamically Consistent Phase Field Model for Mixed-Mode Fracture in Rock-Like Materials," *Computer Methods in Applied Mechanics and Engineering* 392 (2022): 114642.

64. D. Yang, X. Q. He, J. Q. Zhu, and Z. W. Bie, "A Novel Damage Model in the Peridynamics-Based Cohesive Zone Method (PD-CZM) for Mixed Mode Fracture With Its Implicit Implementation," *Computer Methods in Applied Mechanics and Engineering* 377 (2021): 113721.

65. W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev Training for Neural Networks," 2017 in 31st International Conference on Neural Information Processing Systems.

66. D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, https://doi.org/10.48550/arXiv.1412.6980. (2014).

67. X. L. Yu and X. P. Zhou, "A Nonlocal Energy-Informed Neural Network for Isotropic Elastic Solids With Cracks Under Thermomechanical Loads," *International Journal for Numerical Methods in Engineering* 124 (2023): 3935–3963.

68. B. Winkler, G. Hofstetter, and H. Lehar, "Application of a Constitutive Model for Concrete to the Analysis of a Precast Segmental Tunnel Lining," *International Journal for Numerical and Analytical Methods in Geomechanics* 28, no. 7/8 (2004): 797–819.

69. D. Yang, X. Q. He, S. H. Yi, and X. F. Liu, "An Improved Ordinary State-Based Peridynamic Model for Cohesive Crack Growth in Quasi-Brittle Materials," *International Journal of Mechanical Sciences* 153 (2019): 402–415.