

# DEDEM: Discontinuity Embedded Deep Energy Method for solving fracture mechanics problems

Luyang Zhao<sup>a</sup>, Qian Shao<sup>a,b\*</sup>

<sup>a</sup>Department of Engineering Mechanics, School of Civil Engineering, Wuhan University,  
430072 Wuhan, China

<sup>b</sup>Wuhan University Shenzhen Research Institute, 518057 Shenzhen, China

## Abstract

Physics-Informed Neural Networks (PINNs) have aroused great attention for its ability to address forward and inverse problems of partial differential equations. However, approximating discontinuous functions by neural networks poses a considerable challenge, which results in high computational demands and low accuracy to solve fracture mechanics problems within standard PINNs framework. In this paper, we present a novel method called Discontinuity Embedded Deep Energy Method (DEDEM) for modeling fracture mechanics problems. In this method, interfaces and internal boundaries with weak/strong discontinuities are represented by discontinuous functions constructed by signed distance functions, then the representations are embedded to the input of the neural network so that specific discontinuous features can be imposed to the neural network solution. Results demonstrate that DEDEM can accurately model the mechanical behaviors of cracks on a large variety of fracture problems. Besides, it is also found that DEDEM achieves significantly higher computational efficiency and accuracy than the existing methods based on domain decomposition techniques.

**Keywords:** Physics-informed neural networks; Deep energy method; Fracture mechanics; Bimaterial interface; Discontinuity

## 1 Introduction

Fracture mechanics analysis is of utmost importance for prediction of failure mode and assessment of service life in engineering materials and structures. Numerical simulations of the fracture behaviour involve solving partial differential equations (PDEs) with singularities at the crack tip and jump discontinuities across the crack surface, which pose significant challenges for numerical modeling, especially for the classic finite element method (FEM) [1] which is the most popular tool for solid mechanics modeling. Over the past decades, various numerical tools have been developed to model the mechanical behaviour of the crack, including the Extended Finite Element Method (XFEM) [2], boundary Element Method (BEM) [3], phase field method (PF) [4] and peridynamics [5].

The advent of increased computational power has led to a resurgence in the use of neural networks (NNs) for solving PDEs, decades after the initial pioneering work [6]. Thanks to the

---

\*Corresponding author. E-mail address: qian.shao@whu.edu.cn.

universal approximator property of NNs [7], the Physics-Informed Neural Networks (PINNs) framework proposed by Rassi et al. [8] is able to deal with forward and inverse problems involving PDEs by employing NNs as function approximators. Subsequently, PINNs have found success in a variety of PDE-governed physical systems, including fluid mechanics [9], heat transfer problems [10] and cardiovascular flows [11]. Additionally, Haghighe et al. [12] extended the application of PINNs to solving forward problems, inversion and surrogate modeling for solid mechanics problems. The standard PINNs framework involves enforcing PDEs and initial/boundary conditions by minimizing squared errors on collocation points inside the domain and on the boundaries, respectively. An alternative training strategy is the Deep Ritz Method (DRM) [13], which focuses on addressing the variational formulations of PDEs with NNs. DRM can more efficiently solve the PDEs as the method requires lower orders of gradients and automatically satisfies Neumann boundary conditions. Subsequently, DRM is introduced to solving solid mechanics problems, and the implementation is called deep energy method (DEM) [14], whose application involves elastic plate [15], hyperelasticity [16] and topology optimization [17].

Nonetheless, the application of PINNs approach encounters formidable challenges in tackling problems involving strong and weak discontinuities as feed-forward networks have limited ability on discontinuous functions [18]. The difficulties also present challenges in solving fracture problems. Many attempts to modeling fracture avoid directly describing the discontinuities. Some researchers utilize the phase field model [14, 19–21], where cracks are viewed as material damage and dispersed to be process zones with finite width; Other researchers utilize peridynamics theories [22, 23], where cracking behaviors are modeled by non-local interactions. Despite success on modeling crack behavior achieved by these methods, they cannot directly deal with ideal line cracks. The question of modeling strong and weak discontinuities problems with PINNs still not be answered.

One straightforward approach to overcome such challenge is to use domain decomposition techniques, where the entire domain is split into several sub-domains by the interfaces and each subdomain is approximated using an individual neural network. The initial attempt at domain decomposition was made by conservative-PINNs (CPINNs) [24], followed by the subsequent work extended-PINNs (XPINNs) [25] proposed by Rassi et al. Then Wu et al. [26] proposed interfaced neural networks (INN) to improve the efficiency and accuracy for domain decomposition techniques. He et al. [27] applied domain decomposition to solve 3D elliptic interface problems. Yu et al. [28] used CPINNs to solve multi-material problems in solid mechanics. Jang et al. [29] applied augmented Lagrangian term to accelerate the convergence for solving interface problems. Domain decomposition techniques are then be employed to solve fracture mechanics problems. Wang et al. [30] proposed conservative energy method (CENN) to employ domain decomposition within DEM framework, which successfully simulated mode III crack and heterogeneous problems. Gu et al. [31] incorporated enrichment functions to precisely capture the singular crack-tip field and calculate the stress intensity factors (SIFs) of 2D in-plane cracks. Then the same technique is utilized to solve interface fracture problems [32] and fatigue crack growth [33].

However, several challenges remain in domain decomposition based method:

(1) *Intractable hyperparameter tuning*: The interface conditions between sub-domains are enforced through additional penalties in the loss function. It has been proven that the hyperparameters associated with these loss terms are crucial for the successful training of both strong and weak forms [24, 30]. Balancing these hyperparameters becomes particularly challenging for too many loss terms introduced by domain decomposition.

(2) *High computational cost*: Training separate NNs for each sub-domain results in an increased number of trainable parameters. Without the implementation of parallelization techniques [34], this computational burden will reach an impractical level for training sub-networks in serial. Besides, competition among multiple loss terms also significantly impacts the training efficiency [35], as much more training epochs are required to achieve convergence.

(3) *High risk for training failure*: The competition among multiple loss terms mentioned above also significantly impacts the accuracy. The imbalanced multi-objective optimisation process sometimes leads to training failures caused by gradient pathologies [36].

To overcome these challenges, efforts have been made to modeling discontinuities bypassing the domain decomposition techniques to improve the efficiency and accuracy. Wang et al. [37] first investigated DRM on solving interface problems with discontinuous coefficients. Hu et al. [38] proposed DCSNN to approximate piece-wise continuous function by introducing additional feature inputs to the network. Subsequently, Tseng et al. [39] enable the neural network to capture weak discontinuities for elliptic interface problems using the same approach. Yao et al. [40] solved multi-material diffusion equation with one single network by domain separation strategy, and the discontinuities are provided by the separated inputs to the network. Sarma et al. [41] proposed interface PINNs (I-PINNs) to model interface problems with modified NN architecture. It should be noted that there still lacks investigation on modeling internal boundaries, so that problems with ideal line crack cannot be solved.

In this paper, we propose the Discontinuity Embedded Deep Energy Method (DEDEM), which is applied to obtain weak solutions for fracture mechanics problems. Within our framework, interfaces and cracks are tracked by signed distance functions (SDF), then strong and weak discontinuities are imposed to the neural network solution through embedding the discontinuities into the inputs. Our method offers the following advantages:

(1) *Flexibility*: Our approach for constructing discontinuities can be readily applied to model arbitrary crack patterns, including complex curved cracks, crossing cracks and bi-material interface cracks. Besides, such description by SDF is easy for implementation.

(2) *Efficiency and Accuracy*: Without competition between interface conditions and governing equations during the training process, the neural network converges much faster and higher accuracy can be achieved. Besides, crack-tip singularities can be automatically captured by NNs without any special treatment.

(3) *Convenience for postprocessing*: PINNs provide a meshless solution in the whole domain, which offer convenience for any subsequent calculation, including acquiring stress intensity factors and computing the direction for crack propagation.

The paper is organized as follows. Section 3 introduces the governing equations for discontinuous elastic problems and basic idea of PINNs method. Section 3 details the proposed

DEDEM for modeling problems with strong/weak discontinuities. Then the implementation details and four examples are presented in Section 4 to show the applicability of DEDEM on various fracture problems. Subsequently, Section 5 discussed some possible extensions of our proposed method. Finally, a brief summary is given in Section 6.

## 2 Preliminaries

### 2.1 Governing equations for discontinuous elasticity problems

The governing equation of a quasi-static mechanical system is given by the equilibrium equations under prescribed boundary conditions:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} &= 0 \text{ in } \Omega, \\ \mathbf{u} &= \bar{\mathbf{u}} \text{ on } \Gamma_u, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \bar{\mathbf{t}} \text{ on } \Gamma_t, \end{aligned} \tag{1}$$

where  $\boldsymbol{\sigma}$  is the stress tensor,  $\mathbf{b}$  is the body force,  $\mathbf{u}$  is the displacement vector,  $\mathbf{n}$  is the unit outward normal,  $\bar{\mathbf{u}}$  and  $\bar{\mathbf{t}}$  respectively denotes for the displacement and force boundary conditions;  $\Omega$  is the computational domain and  $\Gamma_u$  and  $\Gamma_t$  are respectively essential and natural boundaries. Then the constitutive relation of linear elastic problems can be formulated by:

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}, \tag{2}$$

where  $\mathbf{C}$  is the stiffness tensor which can be defined by the Young's modulus  $E$  and the Poisson's ratio  $\nu$  for linear elastic problems.  $\boldsymbol{\varepsilon}$  is the strain tensor:

$$\boldsymbol{\varepsilon} = \nabla_s \mathbf{u} \text{ with } \varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{3}$$

where  $\nabla_s$  is the symmetric part of the gradient operator.

In order to derive the governing equation for discontinuous elasticity problems, here we consider a discontinuous problem shown in Figure 1 (a). The whole domain  $\Omega$  is divided into  $\Omega^+$  and  $\Omega^-$  by the interface  $\Gamma_d$ , and the external boundaries, where  $\Gamma_d = \Gamma_{d_1} \cup \Gamma_{d_2}$ . Here both of the strong and weak discontinuities exist, as the solution is discontinuous across the crack surface  $\Gamma_{d_1}$ , and the normal derivative is discontinuous across the material interface  $\Gamma_{d_2}$ . The interface conditions on  $\Gamma_d$  can be formulated as:

$$\begin{aligned} \boldsymbol{\sigma}^+ \cdot \mathbf{n}^+ + \boldsymbol{\sigma}^- \cdot \mathbf{n}^- &= [\bar{\mathbf{t}}_d] \text{ on } \Gamma_{d_1}, \\ \boldsymbol{\sigma}^+ \cdot \mathbf{n}^+ + \boldsymbol{\sigma}^- \cdot \mathbf{n}^- &= 0 \text{ on } \Gamma_{d_2}, \\ \mathbf{u}^+ - \mathbf{u}^- &= 0 \text{ on } \Gamma_{d_2}, \end{aligned} \tag{4}$$

where  $[\bar{\mathbf{t}}_d]$  denotes the traction acting on the crack surface. Here we let  $\mathbf{n}_{\Gamma_d} = \mathbf{n}^+ = -\mathbf{n}^-$  on  $\Gamma_d$ . Since the differential operator of Eq. (1) is linear and self-adjoint [1], then the weak

formulation can be derived by introducing a trial function  $\mathbf{u}$  and a test function  $\delta\mathbf{u}$ :

$$\int_{\Omega} \delta\mathbf{u} \cdot (\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) + \mathbf{b}) d\Omega = 0, \quad (5)$$

where both of the test and trial function must satisfy the essential boundary conditions. Then integration by parts is used:

$$\int_{\Omega} \delta(\nabla \mathbf{u}) : \boldsymbol{\sigma}(\mathbf{u}) d\Omega - \int_{\Omega} \nabla \cdot (\delta\mathbf{u} \cdot \boldsymbol{\sigma}(\mathbf{u})) d\Omega - \int_{\Omega} \delta\mathbf{u} \cdot \mathbf{b} d\Omega = 0. \quad (6)$$

The divergence theorem is applied respectively in  $\Omega^+$  and  $\Omega^-$  and interface conditions are utilized, the second term of Eq. (6) can be formulated as:

$$\begin{aligned} \int_{\Omega} \nabla \cdot (\delta\mathbf{u} \cdot \boldsymbol{\sigma}) d\Omega &= \int_{\Omega^+} \nabla \cdot (\delta\mathbf{u} \cdot \boldsymbol{\sigma}) d\Omega + \int_{\Omega^-} \nabla \cdot (\delta\mathbf{u} \cdot \boldsymbol{\sigma}) d\Omega \\ &= \int_{\Gamma^+ \cup \Gamma_{d1} \cup \Gamma_{d2}} \delta\mathbf{u}^+ \cdot \boldsymbol{\sigma}^+ \cdot \mathbf{n}^+ d\Gamma + \int_{\Gamma^- \cup \Gamma_{d1} \cup \Gamma_{d2}} \delta\mathbf{u}^- \cdot \boldsymbol{\sigma}^- \cdot \mathbf{n}^- d\Gamma \\ &= \int_{\Gamma} \delta\mathbf{u} \cdot \boldsymbol{\sigma} \cdot \mathbf{n} d\Gamma + \int_{\Gamma_{d1} \cup \Gamma_{d2}} \delta\mathbf{u} \cdot (\boldsymbol{\sigma}^+ \cdot \mathbf{n}^+ + \boldsymbol{\sigma}^- \cdot \mathbf{n}^-) d\Gamma \\ &= \int_{\Gamma} \delta\mathbf{u} \cdot \bar{\mathbf{t}} d\Gamma + \int_{\Gamma_{d2}} \delta\mathbf{u} \cdot [\bar{\mathbf{t}}_d] d\Gamma. \end{aligned} \quad (7)$$

By utilizing the symmetry of stress tensor on the first term and Eq. (7) on the second term, the weak form of governing equation for the discontinuities problem can be written as:

$$\delta J(\mathbf{u}) = \int_{\Omega} \delta\boldsymbol{\varepsilon}(\mathbf{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) d\Omega - \int_{\Gamma_t} \delta\mathbf{u} \cdot \bar{\mathbf{t}} d\Gamma - \int_{\Gamma_{d2}} \delta\mathbf{u} \cdot [\bar{\mathbf{t}}_d] d\Gamma - \int_{\Omega} \delta\mathbf{u} \cdot \mathbf{b} d\Omega = 0, \quad (8)$$

where  $J(\mathbf{u})$  is the total potential energy:

$$J(\mathbf{u}) = \int_{\Omega} \frac{1}{2} \boldsymbol{\varepsilon}(\mathbf{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) d\Omega - \int_{\Gamma_t} \mathbf{u} \cdot \bar{\mathbf{t}} d\Gamma - \int_{\Gamma_{d2}} \mathbf{u} \cdot [\bar{\mathbf{t}}_d] d\Gamma - \int_{\Omega} \mathbf{u} \cdot \mathbf{b} d\Omega. \quad (9)$$

The first term stands for the strain energy, and the rest terms respectively stands for the external energy contributed by surface traction on external boundaries and internal boundaries, and the body force inside domain. It should be noted that the formulation of the principle of virtual work is equivalent to minimize  $J(\mathbf{u})$ .

## 2.2 Physics-informed Neural Networks

In Physics-Informed Neural Networks (PINNs) framework, solutions of PDEs are approximated by deep learning techniques. Deep Neural Networks (DNNs), which have been proven to be universal approximators [7], are the primary choice for PINNs. A DNN can be formulated as a composition of simple transformations:

$$y = NN(x) = T^{(n)}(T^{(n-1)}(\dots T^{(1)}(x))), \quad (10)$$

where  $x$  is the input, and  $y$  is the output.  $T$  is one single hidden layer, which is a combination of a linear transformation and an activation function:

$$y^{(i)} = T^{(i)}(x^{(i)}) = \sigma(W^{(i)}x^{(i-1)} + b^{(i)}), \quad (11)$$

where  $x^{(i)}$  and  $y^{(i)}$  are the inputs and outputs of the  $i - th$  layer, respectively, while  $W$  and  $b$  are trainable parameters that are updated by gradient descent and back propagation algorithms. The activation function introduces nonlinearity to the neural network. Then we consider a PDE with given boundary conditions on  $\partial\Omega := \cup_{n=1}^M \Gamma_i$ :

$$\begin{aligned} \mathcal{L}(\mathbf{u}) &= f \text{ in } \Omega, \\ \mathcal{G}(\mathbf{u}) &= g_i \text{ on } \Gamma_i \ (i = 1, 2, 3, \dots, M), \end{aligned} \quad (12)$$

where  $\mathcal{L}$  is a known differential operator, and  $\mathcal{G}$  denotes arbitrary boundaries conditions, including Dirichlet and Neumann boundaries. Let  $\mathbf{u}$  be the approximation of the neural networks, the solution of the PDE can be approximated by the neural network by minimizing the following loss function:

$$\mathbf{u}(\mathbf{x}; \theta) = \arg \min_{\theta} L(\theta) := \underbrace{\frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \mathcal{L}\mathbf{u}(x^{(i)}) - f \right\|^2}_{\text{PDE loss}} + \underbrace{\sum_{j=1}^M \frac{1}{N_b^{(j)}} \sum_{i=1}^{N_b^{(j)}} \left\| \mathcal{G}\mathbf{u}(x^{(i)}) - g_j \right\|^2}_{\text{B.C. loss}}, \quad (13)$$

where  $\theta$  is the learnable parameters. If Eq. (12) has a variational form, the solution can also be obtained by:

$$\mathbf{u}(\mathbf{x}; \theta) = \arg \min_{\theta} I(\theta) := J(\mathbf{u}) + \beta \int_{\Gamma_u} \|(\mathbf{u} - \bar{\mathbf{u}})\|^2 d\Gamma_u, \quad (14)$$

where  $\Gamma_u$  is the boundaries with essential boundary conditions  $\bar{\mathbf{u}}$ , and  $\beta$  is the lagrangian multiplier, which is a hyper-parameter needs to be tuned. Using the energy form of PDE for training offers two advantages over strong form: 1.Training with energy form requires lower orders of derivative than strong form, leading to a substantial reduction in computational cost during the back propagation training process. 2.Neumann boundary conditions can be automatically satisfied, so that less loss terms and hyperparameters are required. By substituting  $J(\mathbf{u}(\mathbf{x}; \theta))$  with Eq. (9), linear elasticity problems can be solved. In this paper, we choose to use the weak form to train the network. The NN approximation is trained by the loss function composed of potential energy and essential boundary conditions. It should be noted that the traction-free conditions can be satisfied without specific treatment, which offers convenience in implementation.

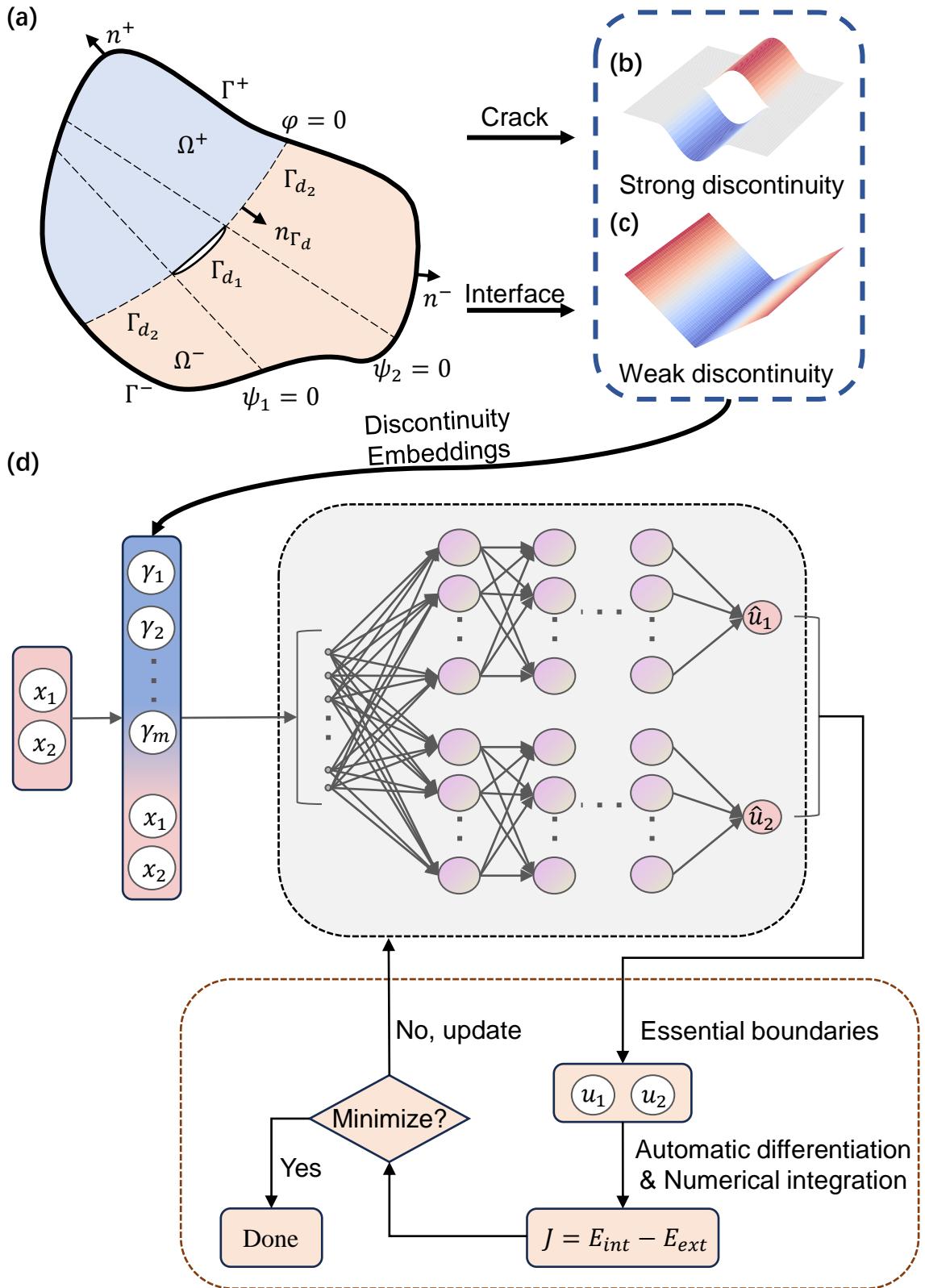


Figure 1: The schematic of DEDEM: (a) The schematic of a problem with a crack; (b) strong and (c) weak discontinuous functions constructed from SDF; (d) The whole training workflow.

### 3 Discontinuity Embedded Deep Energy Method

#### 3.1 Overview of the proposed method

In standard PINNs, the solution is approximated by a neural network, where the inputs are the spatial coordinates of the sampling points. It has been proven theoretically [18] and experimentally [41] that the output solutions can hardly output discontinuous solutions. This makes it challenging to approximate discontinuous physical fields with PINNs. In DEDEM, we introduce a transformation of spatial coordinates and embed them into the input of the neural network:

$$\mathbf{u}(\mathbf{x}) = u_{NN}(\mathbf{x}, \boldsymbol{\gamma}(\mathbf{x})), \quad (15)$$

where  $u_{NN}$  represents the neural network,  $\mathbf{x}$  is the spatial coordinates of points, and  $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, \dots, \gamma_n\}$  denotes the introduced embedding to describe the discontinuities inside the domain. The derivative of the output respect to a spatial coordinate  $x_i$  can be given by the chain rule:

$$\frac{\partial \mathbf{u}(\mathbf{x})}{\partial x_i} = \frac{\partial u_{NN}(\mathbf{x}, \boldsymbol{\gamma}(\mathbf{x}))}{\partial x_i} + \frac{\partial u_{NN}(\mathbf{x}, \boldsymbol{\gamma}(\mathbf{x}))}{\partial \boldsymbol{\gamma}(\mathbf{x})} \cdot \frac{\partial \boldsymbol{\gamma}(\mathbf{x})}{\partial x_i}. \quad (16)$$

It is obvious that if the embedding  $\boldsymbol{\gamma}(\mathbf{x})$  and the derivative of embedding  $\frac{\partial \boldsymbol{\gamma}(\mathbf{x})}{\partial x_i}$  is constructed to satisfy specific features, the outputs of NN can be imposed to satisfy specific discontinuities. In this section, we provide a detailed description of the construction of embeddings for describing strong and weak discontinuities across the interfaces.

#### 3.2 Strong discontinuities on opening interfaces

Strong discontinuities occur in crack problems as displacements are discontinuous across the crack surface. Displacements on one side are distinct from the other side on the crack surface, leading to multiple values correspond to one spatial coordinate. The displacement approximation must have non-zero solution jumps across the crack, which can be satisfied when  $\boldsymbol{\gamma}$  is selected to be  $\boldsymbol{\gamma}(\mathbf{x}^+) \neq \boldsymbol{\gamma}(\mathbf{x}^-)$ :

$$[\![\mathbf{u}]\!] = \mathbf{u}^+ - \mathbf{u}^- = NN(\mathbf{x}, \boldsymbol{\gamma}^+(\mathbf{x})) - NN(\mathbf{x}, \boldsymbol{\gamma}^-(\mathbf{x})) \neq 0. \text{ on crack surface} \quad (17)$$

This can be simply achieved by selecting  $\boldsymbol{\gamma}$  as the sign function:

$$\boldsymbol{\gamma} = \text{sgn}(\varphi), \quad (18)$$

where  $\text{sgn}(\cdot)$  takes the following form:

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases}, \quad (19)$$

and  $\varphi$  is the signed distance functions (SDF) to track and describe the crack surface. SDF techniques is widely used for tracking and describing surfaces and interfaces in XFEM [42]. An example of an interior crack is given in Figure 2 (a), and the described surface is the green line, of which the  $\varphi$  can be given by:

$$\varphi(\mathbf{x}) = \text{sgn}(\mathbf{n} \cdot (\mathbf{x} - \hat{\mathbf{x}})) \|\mathbf{x} - \hat{\mathbf{x}}\|, \quad (20)$$

where  $\hat{\mathbf{x}}$  denotes for the closest point on the surface to the point  $\mathbf{x}$ ,  $\mathbf{n}$  is the outward normal, and  $\|\mathbf{x} - \hat{\mathbf{x}}\|$  is the distance of the point  $\mathbf{x}$  to the surface.

Since the crack do not divide the domain into two distinct parts completely, the sign function needs to be modified to guarantee the smoothness in front of the crack tip. This can be achieved by modifying the embedding with SDF for describing the tangential extension of the crack tip. For an interior crack, the embedding can be given by:

$$\gamma = F(\psi_1, \psi_2) \cdot \text{sgn}(\varphi), \quad (21)$$

where  $\psi_1, \psi_2$  are the SDF of the crack tips on tangential direction, and  $F(\psi_1, \psi_2)$  requires to have C1 continuity to ensure the flux continuities in front of the crack tip. The selection of  $F(\cdot)$  is arbitrary, it can be constructed by interpolation, B-splines or trigonometric functions. Here we use the Rectified Linear Unit (ReLU) function for convenience:

$$\gamma = \text{ReLU}^2(\psi_1 \psi_2) \cdot \text{sgn}(\varphi), \quad (22)$$

The  $F(\psi_1, \psi_2)$  for the example is shown in Figure 2 (c) , and the crack embedding is shown in Figure 2 (d). The introduced function jumps across the crack as well as keeping continuous elsewhere. Besides, the first derivative with respect to the spatial coordinate can be expressed as:

$$\frac{\partial \gamma}{\partial x_i} = \text{sgn}(\varphi) \text{ReLU}(\psi_1 \psi_2) \sum_{j=1}^2 \frac{\partial \psi_j}{\partial x_i}, \quad (23)$$

which is also continuous to guarantee the continuity of derivatives in front of the crack tip.

**Remark .** The boundary conditions on crack surface are always intractable for PINNs method based on strong form of governing equations. In most cases, the boundary conditions on crack surface are traction free boundaries. Since the neural network solution is separated alongside the crack surface by discontinuity embeddings, the nature boundary conditions can be automatically satisfied when the network is trained by energy formulation.

### 3.3 Weak discontinuities

Weak discontinuities occur in heterogeneous solid mechanics problems. In these problems, displacements are continuous across the material interface, while the normal derivative has a jump discontinuity. Solving interface PDE problems with such weak discontinuities has been widely discussed in PINN framework [24, 26, 27, 30, 37–41]. Here we use a ramp function proposed

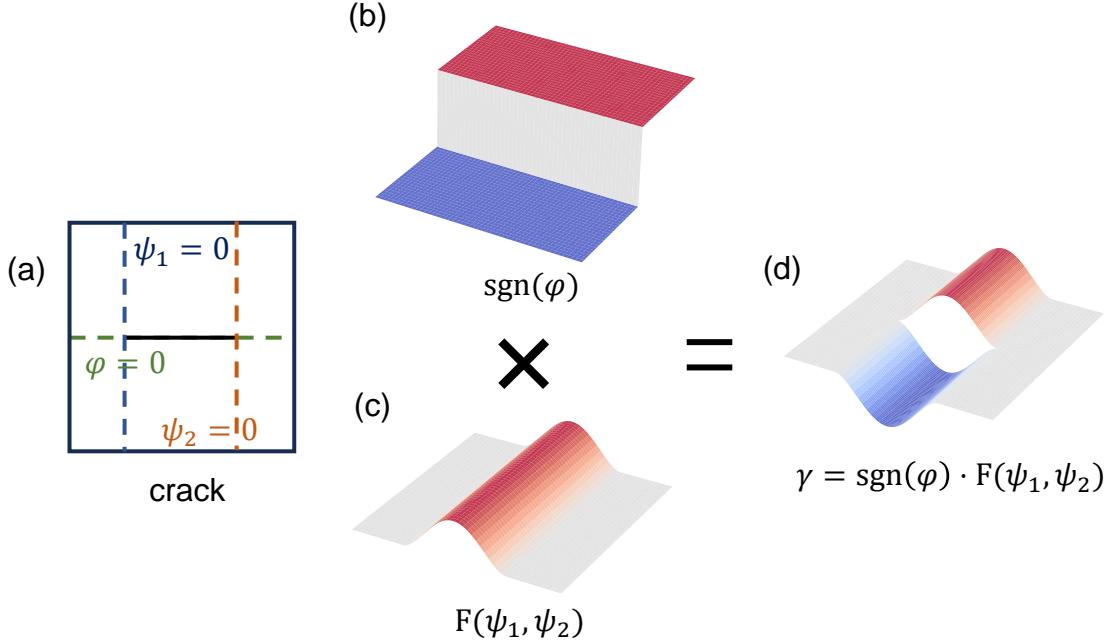


Figure 2: An example of the embedding of a crack: (a) A domain with a crack; (b) the sign function constructed from  $\varphi$  to separate the crack surface and (c) continuous functions constructed from  $\psi$  to guarantee the continuity outside the crack; (d) The constructed embedding  $\gamma$  for describing the crack.

by Tseng [39] to produce jump normal derivative:

$$\gamma_{inter} = |\varphi|, \quad (24)$$

where  $\varphi$  is the SDF to track the material interface. The neural network approximation  $NN(\mathbf{x}, \gamma_{inter})$  keeps continuous across the interface. Then the normal derivative can be directly derived by the chain rule:

$$\frac{\partial \mathbf{u}}{\partial n} = \frac{\partial NN(\mathbf{x}, \gamma_{inter})}{\partial \mathbf{x}} \cdot \mathbf{n} + \frac{\partial NN(\mathbf{x}, \gamma_{inter})}{\partial \gamma_{inter}} \cdot \frac{\partial \gamma_{inter}}{\partial n}, \quad (25)$$

where  $\partial \gamma_{inter}/\partial n = \text{sgn}(\varphi)$ . It can be observed that the jump in normal derivative is allowed:

$$\llbracket \frac{\partial \mathbf{u}}{\partial n} \rrbracket = \frac{\partial \mathbf{u}(\mathbf{x}^+)}{\partial n} - \frac{\partial \mathbf{u}(\mathbf{x}^-)}{\partial n} = 2 \cdot \frac{\partial NN(\mathbf{x}^+)}{\partial \gamma_{inter}}. \quad (26)$$

## 4 Numerical implementation

### 4.1 Implementation details

Suitable NN architecture should be constructed for field approximation. The selection of activation function is essential in the NN architecture, as the activation function decides the non-linearity. The ReLU function which is commonly used in deep learning tasks is not suggested in DEM framework, as its first derivative is discontinuous and the second derivative vanishes,

which brings difficulties for derive a smooth solution. Here, the tanh function is applied in our implementation for its smoothness. Theoretically, a deeper NN has a stronger expressive ability, but it may suffer from gradient vanishing with tanh activation functions. One technique to mitigate this is the use of residual connections [43]. The  $i$ -th residual block of the NN can be expressed as:

$$y^{(i+1)} = y^{(i)} + G(y^{(i)}, W^{(i)}), \quad (27)$$

where  $G$  represents the residual mapping with trainable parameters to be learned. In the following cases, two neural networks containing one fully-connected layer and two residual blocks are respectively used for approximating the displacements components  $u_1$  and  $u_2$  for 2D problems. Each of the residual blocks contains two fully connected layers, and each fully connected layer used in the NN has 30 neurons in all cases.

Then the NN approximation has to fulfill the essential boundary conditions a-priori to guarantee the accuracy with DEM framework [30, 44]. Generally, the essential boundary conditions can be imposed by "hard constraints" on the neural network approximation. Let  $\mathbf{u}(\mathbf{x}) = NN(\mathbf{x}, \gamma(\mathbf{x}))$  rewritten to be  $\hat{\mathbf{u}}(\mathbf{x})$ , the hard constraints can be given by the following transformation:

$$\mathbf{u}(\mathbf{x}) = A(\mathbf{x}) \cdot \hat{\mathbf{u}}(\mathbf{x}) + B(\mathbf{x}), \quad (28)$$

where  $A(\mathbf{x})$  and  $B(\mathbf{x})$  are constructed to satisfy the given boundary conditions. The hard constraints can be directly constructed for simple cases [45]. For complex boundaries, it can be decided either by radius basis functions [30], pre-trained neural networks [46] or distance functions [47].

Finally, the NN is trained to minimize the potential energy, which involves numerical integration for approximating the strain energy. The integration accuracy is essential to the calculation as the integration error will accumulate during the training epochs. In this paper, the composite trapezoidal quadrature rule is implemented. Despite that the trapezoidal rule requires grids in regular domain, it can be implemented on irregular geometries by transforming the domain into a regular reference domain [44]. Besides, the sampling grids are also refined in the vicinity of the crack tips due to the higher integration accuracy required by the singular mechanical behavior. During the training process, the Adam algorithm [48] is applied to optimize the trainable parameters. The initial learning rate is set to be 0.02, and will be dropped by 0.5 every 5000 epochs. Besides, the training will be stopped in advance if no improvement is observed in 1000 epochs. The pytorch library [49] is used for implementation, and the NNs are trained on one Nvdia RTX 3080ti.

## 4.2 A center crack in homogeneous plate

The first example is a center crack in a homogeneous plate under remote tension shown in Figure 3(a). The plate is in plane strain condition with the Poisson ratio  $\nu=0.3$  and Young's modulus  $E=100\text{GPa}$  under the tension  $\sigma=10\text{MPa}$  at the top and bottom of the plate. The

length and width are taken to be  $b=1\text{m}$ , and the crack with the length of  $a$  is located at  $y=0$ . Taking the symmetry into consideration, only the right-hand part is taken for calculation. The left edge is fixed to be  $u_1 = 0$ , and the lower edge is fixed to be  $u_2 = 0$ . The discontinuity embedding for describing the crack in a half plate can be written as:

$$\gamma(\mathbf{x}) = \text{ReLU}^2(a - x_1) \cdot \text{sgn}(x_2). \quad (29)$$

We first take  $a = 0.5\text{m}$  as an example. The value of embedding is plotted in Section 4.2(d), where discontinuous values can be observed across the crack surface. Then the neural networks are trained by 15000 epochs with  $80 \times 100$  sampling grid points shown in Figure 3(b). The hard constrain for essential boundary conditions is given by:

$$u_1 = \hat{u}_1 \cdot x_1, \quad u_2 = \hat{u}_2 \cdot (x_2 + 1)/2. \quad (30)$$

To evaluate the results, FEM is performed on a refined mesh. Since the plate is primarily subjected to a mode I crack, we focus on the displacement component  $u_2$  and the stress component  $\sigma_{22}$ . The results predicted by DEDEM are presented in Section 4.2(e, g), with the point-wise absolute errors compared to FEM calculations shown in Section 4.2(f, h). The results demonstrate that the jump discontinuity in displacement is accurately captured. Additionally, due to the strong non-linearity of the NN, the singular stress field near the crack tip is accurately resolved without requiring any special treatment at the crack tip.

To further evaluate the proposed method, stress intensity factors (SIFs) are calculated. SIFs are key to predict the failure and propagation of crack. Since the jump discontinuities has been obtained, the SIFs can then be derived by displacement extrapolation method [51] with the crack opening displacements  $\delta_i = u_i^+ - u_i^-$  on the crack surface [52]:

$$\delta_1 = K_2 \frac{\kappa + 1}{\mu} \sqrt{\frac{r}{2\pi}} + O(r^{3/2}), \quad \delta_2 = K_1 \frac{\kappa + 1}{\mu} \sqrt{\frac{r}{2\pi}} + O(r^{3/2}), \quad (31)$$

where  $\kappa = 3 - 4\nu$  is the kolosov constant for plane strain problems,  $\mu_i = E_i/2(1 + \nu_i)$  is the shear modulus, and  $r$  is the distance of the points on the crack surface to the crack tip. Details of the SIF calculation method can be seen in appendix A. We vary the crack length from 0.1m to 0.9m and calculate the corresponding  $K_1$  values. For cases where  $a/b > 0.2$ , points located between  $0.3a$  and  $0.35a$  behind the crack are selected for the SIF calculation, while points between  $0.4a$  and  $0.8a$  are chosen for cases where  $a/b \leq 0.2$ . The results are compared with the reference solution given by [50]:

$$K_1 = \sigma \sqrt{\pi a} \left[ 1 - 0.025 \left( \frac{a}{b} \right)^2 + 0.06 \left( \frac{a}{b} \right)^4 \right] \sqrt{\sec \frac{\pi a}{2b}}. \quad (32)$$

A comparison of the calculated  $K_1$  values with the analytical solution is given in Figure 3 (b), with the relative errors listed in Table 1. The calculations show excellent agreement with the reference solution, with relative errors no greater than two percent for cases where  $a/b > 0.1$ . The largest relative error occurs for the shortest crack, and a detailed discussion of this phenomenon

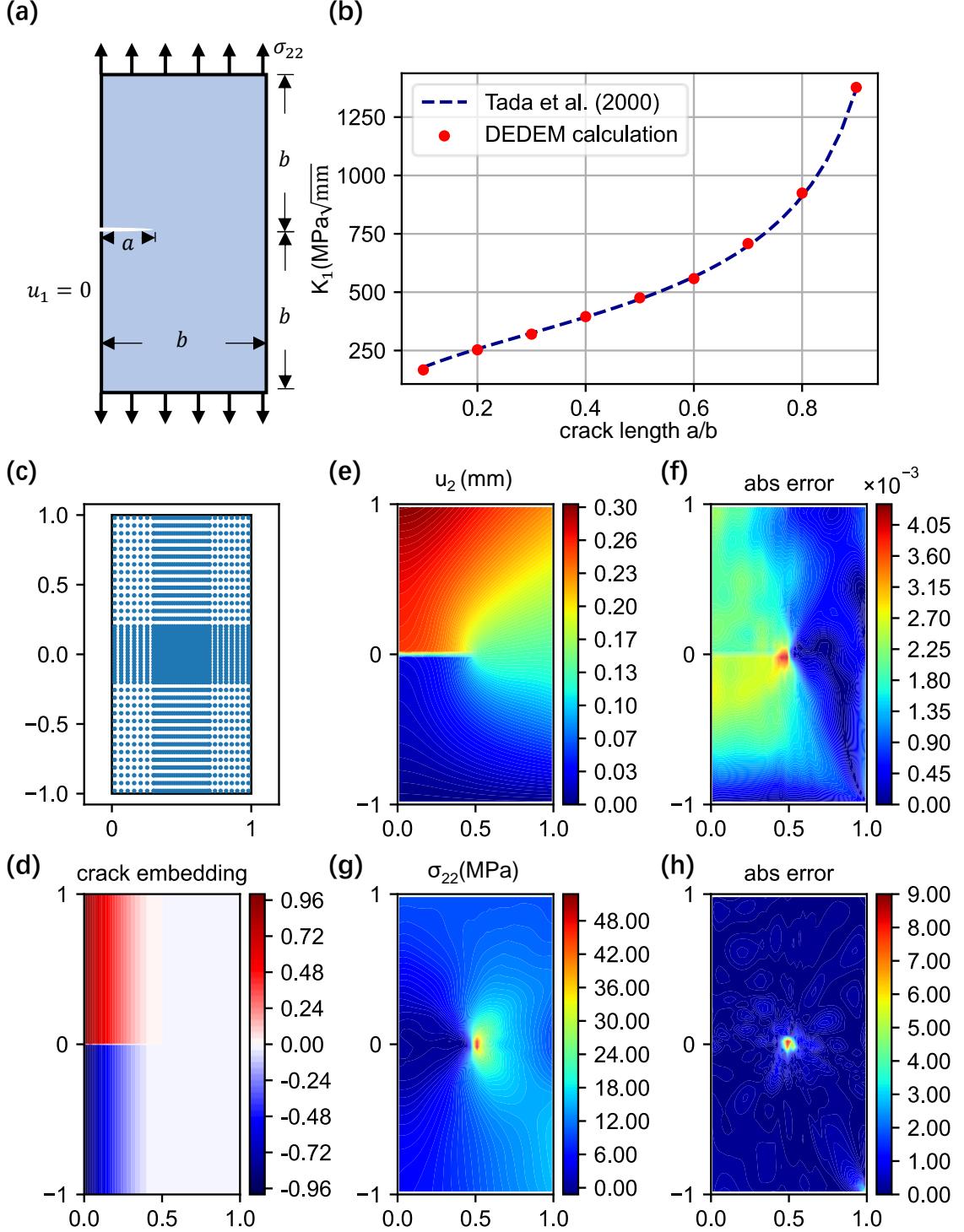


Figure 3: A center crack in a homogeneous plate: (a) the geometry of the symmetric part for computation; (b) Comparison of  $K_1$  with the reference solution [50]; (c) Sampling points for numerical integration and (d) the crack embedding for the case  $a/b = 0.5$ ; Predicted solutions and absolute error compared with FEM for the case  $a = 0.5\text{m}$ : displacement  $u_2$  (e,f) and stress  $\sigma_{22}$  (g,h).

is provided in Section 5.

Table 1:  $K_1$  for a center crack in homogeneous plate

a/b	Reference [50] (MPa $\sqrt{\text{mm}}$ )	Calculation (MPa $\sqrt{\text{mm}}$ )	Error (%)
0.1	178.3	167.1	6.27
0.2	256.8	253.1	1.43
0.3	324.7	320.1	1.41
0.4	393.1	395.3	0.54
0.5	470.1	475.9	1.22
0.6	565.6	558.1	1.32
0.7	697.5	708.1	1.53
0.8	909.6	924.4	1.63
0.9	1370.1	1377.1	0.51

### 4.3 A bi-material interface crack

Analysis for interfacial cracks play an important role in evaluating reliability and mechanical properties for multi-layer structures, as these structures often encounter failure induced by the initiation and propagation of interface cracks resulting from material property mismatches between adjacent layers [53]. In the second example, we investigate the applicability of DEDEM to bi-material interface cracks. Here we consider a bi-material plate in plane stress condition under the tensile loading  $\sigma_{22} = 1\text{ MPa}$  with a center crack at the material interface. Similar to the previous example Section 4.2, only the right half part is considered and the essential boundary conditions are also same to the previous example Section 4.2. The geometry is taken to be  $b = 1\text{ m}$  and the crack length is taken to be  $a = 0.5\text{ m}$ . Since the normal derivative is discontinuous across the material interface, an embedding is constructed to be input into the neural network to track the weak discontinuity:

$$\gamma_{\text{interface}}(\mathbf{x}) = |x_2|. \quad (33)$$

Besides, the crack embedding for strong discontinuities across the crack is also required, which is set to be same as the previous example. The Poisson's ratio is set to be  $\nu_1 = \nu_2 = 0.3$ , and Young's modulus is set to be  $E_2 = 1\text{ GPa}$ , and  $E_1 = 10\text{ GPa}$ . The NN is trained with  $100 \times 150$  sampling points with 15000 maximum training epochs. Results of DEDEM and absolute errors compared with FEM on displacement and stress are plotted in Figure 4(c-j), illustrating high accuracy achieved by the proposed method. It can be observed that the oscillatory crack tip behaviour, which requires complicated crack tip enrichment in classic XFEM [54], can be automatically captured without special treatment by the NN solution in the DEDEM framework. Displacement extrapolation is also applied for computing the SIFs. The elastic mismatch across the material interface results in inseparable tensile and shear effects of interface crack [56], making the SIFs cannot be simply decoupled into mode **I** and mode **II** cracks. The relationship

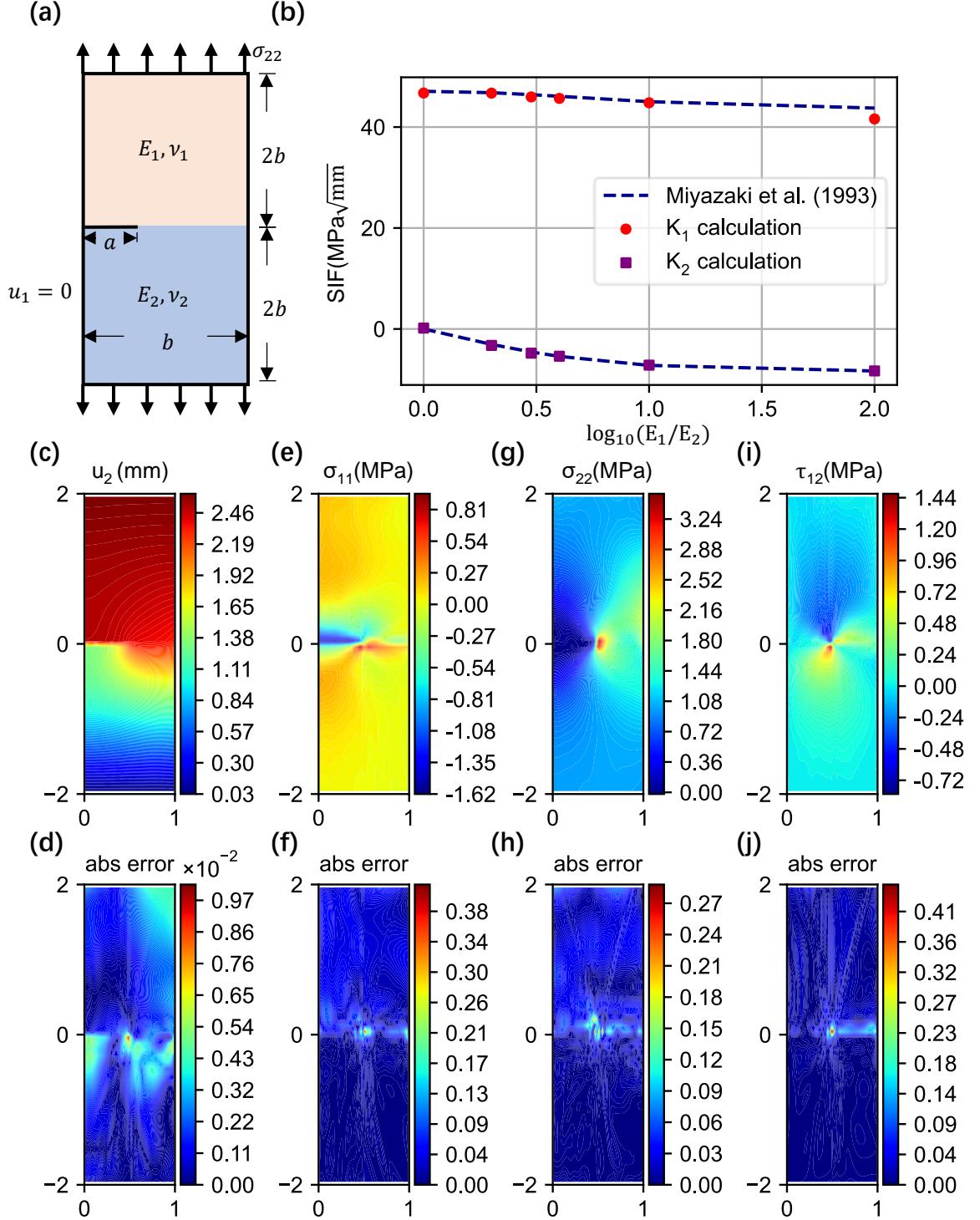


Figure 4: A bi-material interface crack: (a) the geometry of the symmetric part for computation; (b) Comparison of SIFs with the reference solution [55]; (c-j): Predicted displacement and stress and absolute error compared with FEM for the case  $E_1/E_2 = 10$ .

of crack opening displacements and the complex SIF  $\mathbf{K} = K_1 + iK_2$  is given by [57]:

$$\delta_2 + i\delta_1 = \frac{8\mathbf{K}r^i\varepsilon\sqrt{r/2\pi}}{(1+2i\varepsilon)\cosh(\pi\varepsilon)E^*} + O(r^{3/2+i\varepsilon}), \quad (34)$$

where  $i = \sqrt{-1}$  and  $\omega$  is a dimensionless bi-material constant defined by the second Dundurs parameter [58]:

$$\omega = \frac{1}{2\pi} \ln \left( \frac{1-\beta}{1+\beta} \right), \quad \beta = \frac{\mu_1(\kappa_2 - 1) - \mu_2(\kappa_1 - 1)}{\mu_1(\kappa_2 + 1) + \mu_2(\kappa_1 - 1)}, \quad (35)$$

where  $\mu_m$  and  $\kappa_m$  are the shear modulus and Kolosov constant for material  $m$  ( $m = 1, 2$ ), and  $\kappa_m = (3 - \nu_m)/(1 + \nu_m)$  for plane stress problems. We change the  $E_1$  from 1GPa to 100GPa, while keeping the  $E_2$  to be 1GPa. The computed  $K_1$  and  $K_2$  are shown in Figure 4 (b) and Table 2. It can be observed that the value of  $K_2$  does not equal to 0 even if the plate is under pure tensile loading. Besides, our results agrees well with the calculations of [55].

Table 2: SIFs for a bi-material interface crack.

$E_1/E_2$	K <sub>1</sub>			K <sub>2</sub>		
	Reference [55] (MPa $\sqrt{\text{mm}}$ )	Calculation (MPa $\sqrt{\text{mm}}$ )	Error (%)	Reference [55] (MPa $\sqrt{\text{mm}}$ )	Calculation (MPa $\sqrt{\text{mm}}$ )	Error (%)
1	47.08	46.74	0.73	0.00	0.00	/
2	46.81	46.72	0.20	-3.05	-3.25	6.72
3	46.41	45.97	0.96	-4.56	-4.79	5.16
4	46.09	45.67	0.91	-5.43	-5.40	0.55
10	45.02	44.80	0.49	-7.21	-7.17	0.65
100	43.76	41.61	4.91	-8.32	-8.31	0.17

#### 4.4 Multiple cracks

In this section, we aim to investigate the performance of our proposed method on problems with multiple cracks. Here we consider a square plate with a length of 2m shown in Figure 5 (a), with the center of the plate at the coordinate (0,0). The plate is under plane stress conditions with a uniform tensile load of  $\sigma = 10$  MPa applied to the top. The bottom of the plate is fixed with  $u_2 = 0$ , and the right and left sides are fixed with  $u_1 = 0$ . Two cracks intersect at the center of the plate: the first crack runs from the point (-0.5m, -0.5m) to (0.5m, 0.5m), and the second crack runs from (-0.5m, 0.5m) to (0.5m, -0.5m). The Young's modulus is  $E = 100$  GPa, and the Poisson's ratio is  $\nu = 0.3$ .

To demonstrate the efficiency and accuracy of DEDEM, we compare it with CPINN [24] and its energy form, CENN [30]. The domain decomposition strategy for CPINN and CENN is shown in Figure 5 (b). The entire domain is divided into four subdomains along the interfaces denoted by the red lines ahead of the crack tips. The neural network structure used for each subdomain is the same as in DEDEM but with fewer neurons in each layer (15 neurons for CPINN/CENN and 30 neurons for DEDEM) to maintain a similar total number of trainable parameters. A uniform grid with  $N_{pde} = 62500$  internal points is used for training in all three methods. For DEDEM, Two crack embeddings are used for respectively describing the discontinuities across

each of the crack, which are depicted in Figure 5 (c,d) Besides, hard constraints for essential boundary conditions are employed in all the methods:

$$u_1 = \hat{u}_1 \cdot (x_1 + 1) \cdot (x_1 - 1)/4, \quad u_2 = \hat{u}_2 \cdot (x_2 + 1)/2. \quad (36)$$

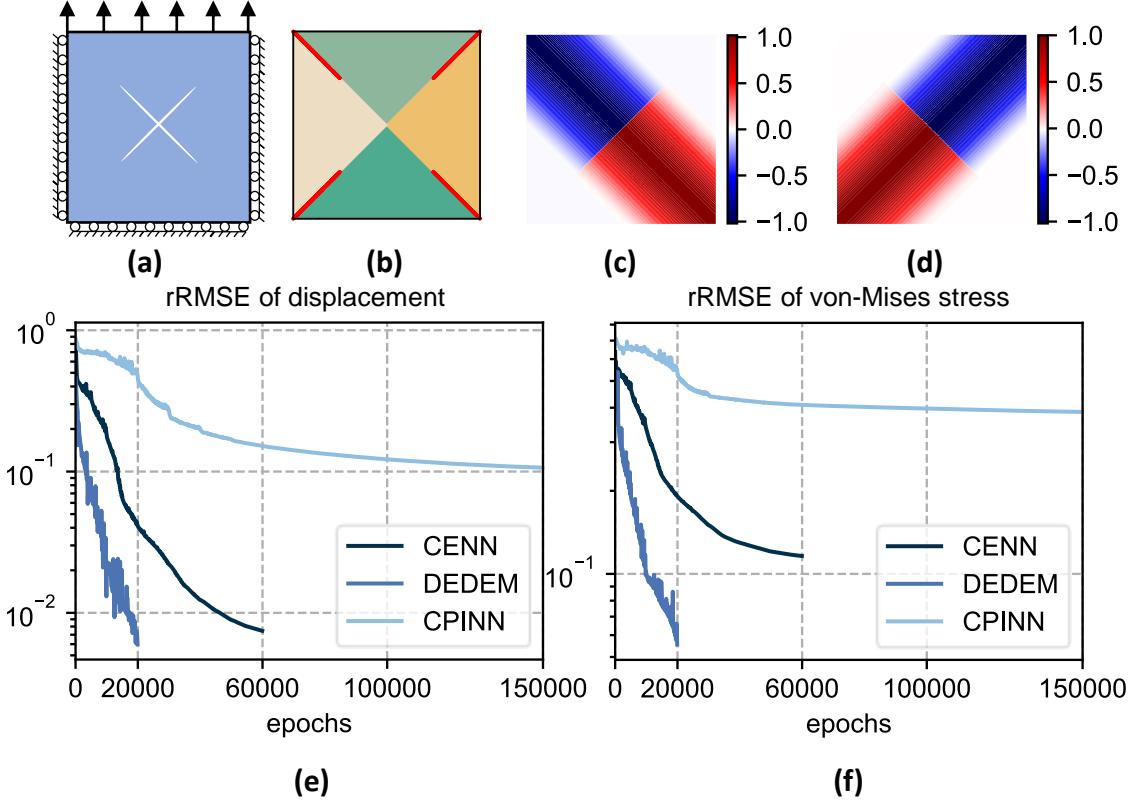


Figure 5: A plate with multiple cracks: (a) The geometry; (b) The domain decomposition strategy for CENN and CPINN; (c-d) The embeddings to describe cracks for DEDEM; The evolution history of rRMSE on (e) displacement and (f) von-Mises stress predicted by CENN, CPINN and DEDEM.

The training and optimization strategy for all the methods are same, which is detailed in Section 4.1. The loss of CPINN is given by:

$$\begin{aligned} L_{\text{CPINN}} = & \underbrace{\lambda_1 \sum_{i=1}^4 \frac{1}{N_{pde}} \sum_{j=1}^{N_{pde}} \|\nabla \cdot \boldsymbol{\sigma}\|^2}_{\text{equilibrium}} + \underbrace{\lambda_2 \sum_{i=1}^4 \frac{1}{N_b} \sum_{j=1}^{N_b} \|\boldsymbol{\sigma} \cdot \mathbf{n} - \bar{\mathbf{t}}\|^2}_{\text{external boundaries}} + \underbrace{\lambda_3 \sum_{i=1}^8 \frac{1}{N_b} \sum_{j=1}^{N_b} \|\boldsymbol{\sigma} \cdot \mathbf{n}\|^2}_{\text{crack surfaces}} \\ & + \underbrace{\lambda_4 \sum_{i=1}^4 \frac{1}{N_I} \sum_{j=1}^{N_I} \|\mathbf{u}^+ - \mathbf{u}^-\|^2 + \lambda_5 \sum_{i=1}^4 \frac{1}{N_I} \sum_{j=1}^{N_I} \|(\boldsymbol{\sigma}^+ - \boldsymbol{\sigma}^-) \cdot \mathbf{n}\|^2}_{\text{interfaces}}, \end{aligned} \quad (37)$$

where  $N_b$  is the number of sampling points on each boundary, which is taken to be 250. The Neumann boundary conditions on all of the external boundaries and the crack surfaces needs to

be taken into consideration, and boundary conditions on each crack surface needs to be considered twice by two subdomains.  $N_I$  is the number of points on each interface, which is taken to be 500.  $(\cdot)^+$  and  $(\cdot)^-$  denotes the solution on different side of the interface predicted by different NNs. The weights of the loss terms  $\lambda_i$  is crucial to the training. Although several techniques has been developed [35, 36, 59–61], it remains to be cumbersome to tune these hyperparameters. In this case, we empirically set  $\{\lambda_i\}_{i=1}^5 = \{10, 1, 1, 1000, 1\}$ . For CENN, the loss function is given by:

$$L_{\text{CENN}} = J(\mathbf{u}) + \lambda_I \sum_{i=1}^4 \frac{1}{N_I} \sum_{j=1}^{N_I} \|\mathbf{u}^+ - \mathbf{u}^-\|^2, \quad (38)$$

where  $J(\mathbf{u})$  is the potential energy shown in Eq. (9), and  $\lambda_I = -1000 \ln(\tanh(N_I/N_{\text{domain}}))$  as suggested in [30]. For DEDEM, two separate embeddings are introduced to respectively represent the two cracks, and only  $J(\mathbf{u})$  is required in loss function since no domain decomposition is applied and essential boundary conditions has been imposed. Details of the hyperparameters can be seen in Table 3.

Table 3: Hyperparameters and comparison of various methods for the cross crack problem.

		CPINN	CENN	DEDEM
Hyperparameters	subdomains	4	4	/
	trainable parameters	4328	4238	4082
	internal points	62500	62500	62500
	boundary points	3000	250	250
	interface points	4000	4000	/
Results	training epochs	150000	60000	20000
	time (seconds/100 epochs)	12.41	3.94	2.35
	rRMSE of $\bar{u}$	0.107	0.008	0.0006
	rRMSE of $\bar{\sigma}$	0.386	0.116	0.056

The training time for every 100 epochs are also shown in Table 3. Note that the efficiency of all these three methods can be further improved. CPINN and CENN can be accelerated by parallelization techniques [34] and DEDEM can be accelerated by manually pre-computing the spatial derivative of embeddings. The accuracy of the results are evaluated by the relative rooted mean square error (rRMSE) compared with FEM. The rRMSE of the physical field  $\phi$  is given by:

$$\text{rRMSE}(\phi) = \sqrt{\frac{\int_{\Omega} \|\phi - \phi_{\text{FEM}}\|^2 d\Omega}{\int_{\Omega} \|\phi_{\text{FEM}}\|^2 d\Omega}}. \quad (39)$$

To evaluate the accuracy of the three methods, we compare the displacement  $\bar{u} = \sqrt{u_1^2 + u_2^2}$  and von-Mises stress  $\bar{\sigma} = \sqrt{(\sigma_{11} - \sigma_{22})^2 / 2 + 3\sigma_{12}^2}$ . The training histories are shown in Figure 6

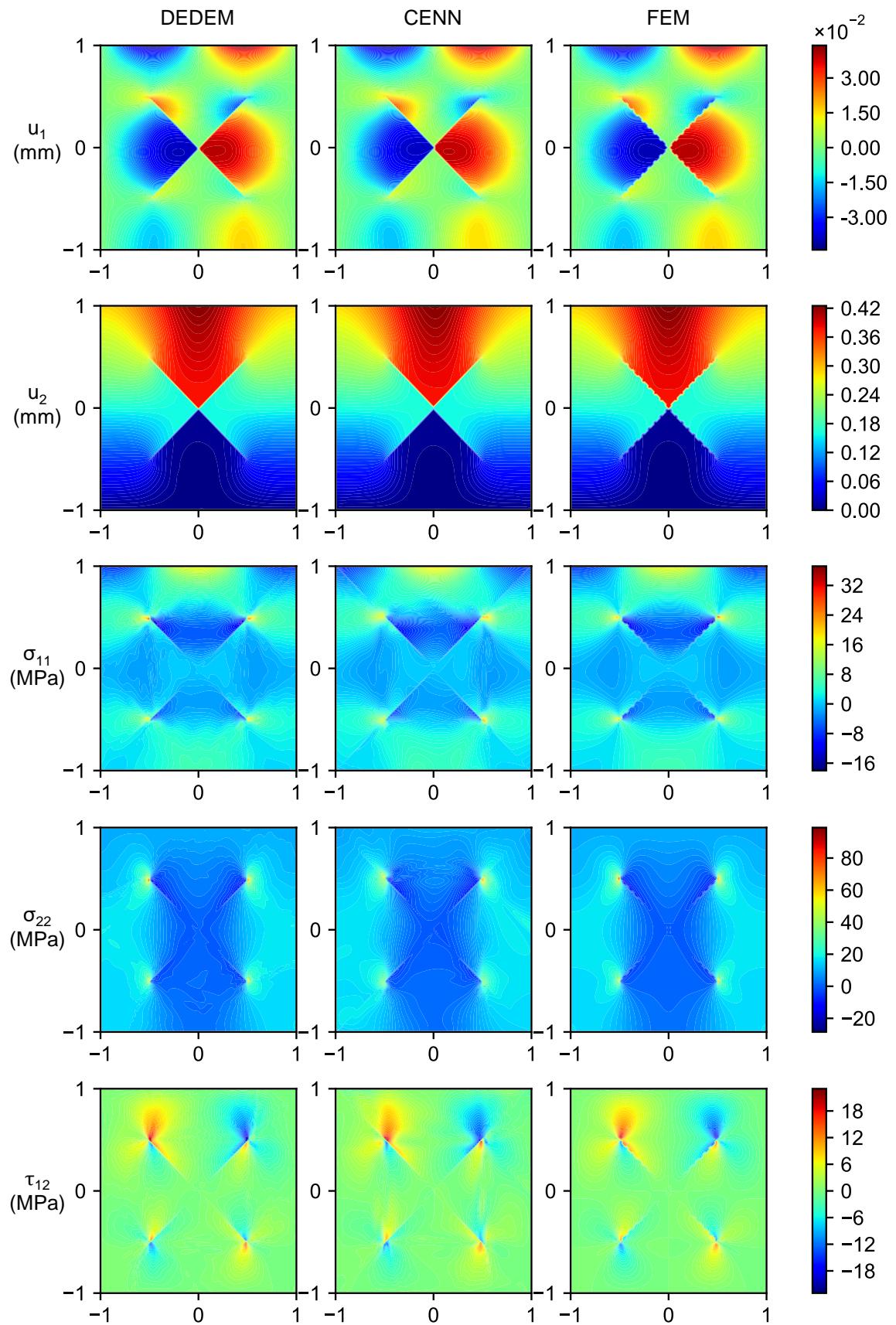


Figure 6: Comparison of displacement and stress components of DEDEM, CENN and FEM.

(e,f), indicating that the DEDEM method achieves best accuracy both on the stress and displacement. Besides, it can be observed that domain decomposition methods require much more training epochs than DEDEM owing to the competition between different loss terms. Detail comparisons of displacement and stress components of DEDEM, CENN and FEM are shown in Figure 6. The displacement of DEDEM and CENN achieves similar accuracy, while DEDEM performs better on stress prediction, especially in the vicinity of the crack tips.

#### 4.5 Crack propagation under shear loading

Since SIFs can be calculated from the predicted displacements, the direction of the crack growth can be determined according to the fracture criterion, and the propagation path can then be predicted via iteratively computing the mechanical response of the growing crack. In the last example, we investigate the applicability of DEDEM on predicting the crack path of an edge shear crack. The problem we consider here is shown in Figure 7 (a). The plate is fixed to be  $u_1 = 0$  on the bottom, and  $u_2 = 0$  on all of the four sides. There is a fixed load  $\tau = 5\text{MPa}$  on top of the plate during the propagation process. The plate is in plane stress condition, and the material properties are set to be  $E=200\text{GPa}$ ,  $\nu=0.3$ .

We assume that the loading process is quasi-static, and no unstable propagation occurs during the process. The maximum circumferential stress hypothesis [62] is applied to determine the propagation direction. Details can be seen in appendix B. The growth in crack length is set to be 0.15m, and the embedding to describe the crack will be updated for each step. A fixed uniform grid with  $250 \times 250$  sampling points is employed for training during the whole process. The maximum training epochs in each step is set to be 20000. The predicted crack path is shown in Figure 7 (a). Crack kinking is observed in the first step, then the crack eventually extends to the right part of the bottom side.

Due to the similarities in the mechanical responses of different steps, transfer learning techniques can be applied to accelerate the training process [19]. The trained NN parameters of the first step is used as the initial parameters for training the current step. The comparison of total training epochs of training with/without transfer learning is shown in Figure 7 (b), showing that transfer learning effectively reduced the computational cost. It should be noted that the training process can be further accelerated by using the NN parameters in previous step as the initial parameters as more similarities exist in adjacent steps. However, training failure may probably occur due to the error accumulated in previous steps. Therefore, we do not adopt such transfer learning strategy to ensure the robustness. The displacements and stress in different computation steps are shown in Figure 7 (c), illustrating that our method can successfully simulate the whole cracking process.

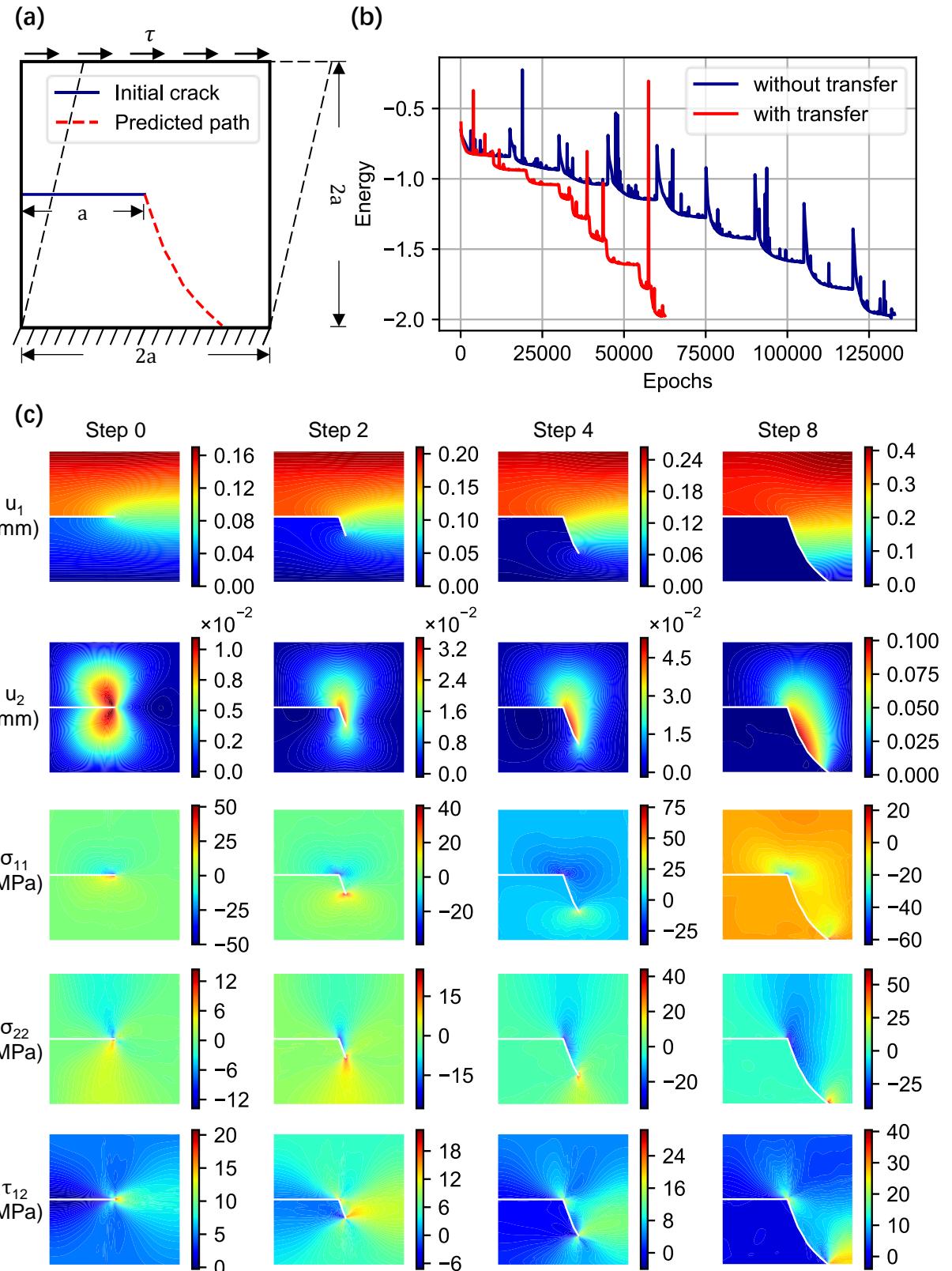


Figure 7: Crack propagation under shear loading: (a) The geometry and the predicted path; (b) Training history with/without transfer learning; (c) Results of displacement and stress components in different steps.

## 5 Discussion

### 5.1 Heterogeneous problem

For heterogeneous problems, weak discontinuities occur at the material interfaces between matrix and inclusion, resulting in low accuracy for stress prediction in DEM. Our proposed DEDEM method can improve the accuracy at the material interface without additional computational cost. Here we take a plate with a circle inclusion at the center shown in Figure 8(a) as an example to compare the accuracy of DEDEM and DEM on stress prediction. The elastic property of the center inclusion is  $E_{in} = 100\text{GPa}$ ;  $\nu = 0.3$ , while the outer matrix is taken to be  $E_{mat} = 20\text{GPa}$ ;  $\nu = 0.3$ . The embedding for describing the material interface can be written as:

$$\gamma(x) = |d(x) - r|, \quad (40)$$

where  $d(x) = \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}$  is the distance to the center of the inclusion, and  $r = 0.25\text{m}$  is the radius of the inclusion. Both of the methods are trained with 22500 uniformly distributed points. DEDEM converged at around 12000 epochs, and DEM converged at around 20000 epochs. The results of von-Mises stress and the absolute error to FEM calculation are shown in Figure 8(b). It can be observed that DEDEM achieves better accuracy near the material interface.

It should be noted that the low accuracy of DEM near the material interface is caused by the feature of MLP, as MLP with smooth activation functions lacks the ability to fit non-smooth functions. Therefore, if non-smooth trial functions are selected, such as the Kolmogorov-Arnold Networks (KAN) [63] constructed by piece-wise splines, weak discontinuities can be automatically learned by NN approximation [64].

### 5.2 Crack-tip singularity

In many classic numerical methods, such as XFEM and BEM, special singular basis are necessary to model the crack-tip mechanical behavior. However, thanks to the powerful expression ability of NN, singularities can be automatically captured with high accuracy by NN solutions without any special treatment in DEDEM framework as shown in previous examples. It should be pointed out that explicit singularity descriptions can also be incorporated into our proposed method to further improve the efficiency and accuracy. One of the feasible method is to introduce fundamental solutions of the near-tip field to the NN solution [31–33]. However, such method has several shortcomings: (1) It adds trouble for constructing the hard constrain of essential boundary conditions, which is pivotal for training with energy form; (2) The fundamental solution for different crack types, such as homogeneous cracks and interface cracks, are different, which adds complexity for implementation, especially for crack propagation. Another method is to introduce an embedding into the input to impose the singularity feature to the NN solution. However, we found that this approach greatly increases the non-linearity of the NN, which often results in training failure caused by integration errors. In conclusion, despite

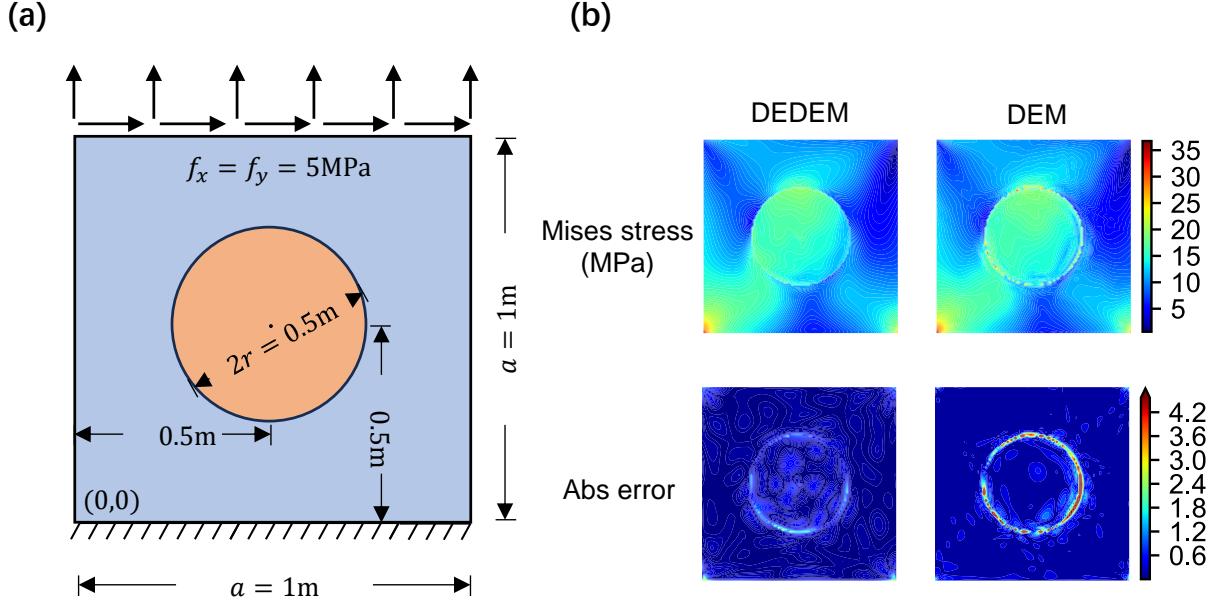


Figure 8: A plate with round inclusion: (a) The geometry; (b) Comparison of von-Mises stress predicted by DEDEM and DEM.

that it is nonessential to embed crack tip singularity in DEDEM, it remains to be an open problem to find a robust and reliable method to incorporate more priori knowledge of the crack tip mechanical behavior to improve the efficiency.

### 5.3 Short cracks

In Section 4.2, we found that DEDEM achieves lowest accuracy on samples where the size of the crack is relatively small compared with the geometry. This is because of the following reasons: 1. Short cracks has little influence to the total potential energy, whose feature may be ignored during the training process of minimizing the potential energy; 2. The small crack can be viewed as a local high frequency feature, and it has been reported that NN performs poorly in fitting high frequency components [35]. To this end, our proposed method can be improved by incorporating approaches aiming to high frequency problems, for example, using specific NNs to focus on specific local patterns [65, 66].

### 5.4 Integration accuracy

Integration accuracy is vital to the training success. In this paper, dense integration points are set, and local refinement strategy is employed at the crack tips. However, due to the lack of theoretical guidance on the integration accuracy of NN, no strategy can completely guarantee the reliability of the integration, which means that the probability of training failure caused by integration error and gradient pathology cannot be completely eliminated. Similar phenomena have also been reported by some researches for solving crack problems from damage perspective [21]. To avoid the trouble of integration, some researches substitute the automatic differentiation by mesh discretization to compute the derivatives, when the NN is only used to

predict the nodal displacements [20]. However, such method gives up the mesh-free property of PINNs method. In a word, there still lacks reliable integration scheme within DEM framework.

## 6 Conclusion

In this paper, we propose a novel method called discontinuity embedded deep energy method (DEDEM) to solve discontinuous elastic problems via physics-informed machine learning. In this method, the cracks and interfaces are described by discontinuous functions constructed by signed distance functions, and then the NN solutions are imposed to satisfy specific discontinuous features by embedding the discontinuous description to the inputs of NN. The proposed method is demonstrated to be efficient and accurate for modeling the strong discontinuities across the crack, singularity at the crack tip and weak discontinuities across the material interfaces. In addition, DEDEM is much more computationally efficient for solving problems with ideal line cracks compared with the state-of-the-art PINNs methods based on domain decomposition techniques as no additional loss term is required for dealing with the discontinuities; Besides, DEDEM is easy for implementation, and arbitrary discontinuous interfaces and internal boundaries can be tracked and solved.

The proposed method can be extended in several ways in the future. First, DEDEM can be applied to [solve 3D fracture problems](#); In addition, the proposed method can be easily employed to solve multi-physics fracture problems since the method view the cracks as ideal line cracks and no special treatment is required for the crack tip; Besides, the method can be further developed to solve dynamic fracture; Furthermore, a data-efficient surrogate model can be established for fracture problems by combining the operator learning frameworks [67, 68]. In conclusion, DEDEM provides a new idea for solving discontinuous PDEs within PINNs framework. We believe that the proposed method is promising to be applied to forward calculation, inverse modeling and surrogate modeling for a large variety of fracture problems.

## A Calculation of the stress intensity factors

Stress intensity factors (SIFs) are crucial parameters to fracture problems for determine the crack propagation and material failure. Here, we use the displacement extrapolation method to derive SIFs. The main idea of this method is to utilize the crack opening displacement (COD) to fit the SIFs. For homogeneous and linear elastic materials, the relationship of COD and SIFs are given by Eq. (31). The relation can be rewritten as:

$$\begin{Bmatrix} \tilde{K}_1(r) \\ \tilde{K}_2(r) \end{Bmatrix} = \frac{\mu}{\kappa+1} \sqrt{\frac{2\pi}{r}} \begin{Bmatrix} \delta_2 \\ \delta_1 \end{Bmatrix} = \begin{Bmatrix} K_1 \\ K_2 \end{Bmatrix} + O(r) \approx \begin{Bmatrix} K_1 \\ K_2 \end{Bmatrix} + \begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix} r. \quad (\text{A.1})$$

Then linear regression technique is employed to fit  $\tilde{K}_i(r)$  ( $i = 1, 2$ ), and the SIFs can then be decided by the intercept:

$$\begin{Bmatrix} K_1 \\ K_2 \end{Bmatrix} = \frac{\mu\sqrt{2\pi}}{\kappa + 1} \lim_{r \rightarrow 0} \frac{1}{\sqrt{r}} \begin{Bmatrix} \delta_2 \\ \delta_1 \end{Bmatrix}. \quad (\text{A.2})$$

Similarly, for bi-material interface cracks, the SIFs can be calculated by using Eq. (34). The real and imaginary parts are separated, and the relation can be rewritten as:

$$\begin{Bmatrix} \tilde{K}_1(r) \\ \tilde{K}_2(r) \end{Bmatrix} = D \begin{bmatrix} -2\varepsilon \cos Q + \sin Q & \cos Q + 2\varepsilon \sin Q \\ \cos Q + 2\varepsilon \sin Q & 2\varepsilon \cos Q - \sin Q \end{bmatrix} \begin{Bmatrix} \delta_2 \\ \delta_1 \end{Bmatrix} \approx \begin{Bmatrix} K_1 \\ K_2 \end{Bmatrix} + \begin{Bmatrix} c_1 \sin Q \\ c_2 \cos Q \end{Bmatrix} r, \quad (\text{A.3})$$

where

$$D = \frac{2\mu_1\mu_2 \cosh(\pi\varepsilon)}{\mu_1(1 + \kappa_2) + \mu_2(1 + \kappa_1)} \sqrt{\frac{2\pi}{r}}, \quad Q = \varepsilon \ln r. \quad (\text{A.4})$$

## B Maximum circumferential stress criterion

After obtaining the SIFs, the propagation direction of the crack can be determined by fracture criterion. In this paper, the maximum circumferential stress criterion is applied. The asymptotic near-tip circumferential stress takes the following form:

$$\sigma_\theta = \frac{1}{2\sqrt{2\pi r}} \cos \frac{\theta}{2} [K_1(1 + \cos \theta) - 3K_2 \sin \theta], \quad (\text{B.1})$$

where  $r$  and  $\theta$  are the local polar coordinates to the crack tip. The criterion assume that the crack will propagate from its tip in a direction  $\theta_c$  where the  $\sigma_\theta$  reaches maximum. The maximum circumferential stress satisfies the following expression:

$$\frac{\partial \sigma_\theta}{\partial \theta} = \frac{-3}{4\sqrt{2\pi r}} \cos \frac{\theta}{2} [K_1 \sin \theta + K_2(3 \cos \theta - 1)] = 0. \quad (\text{B.2})$$

Then  $\theta_c$  can be given by:

$$K_1 \sin \theta_c + K_2(3 \cos \theta_c - 1) = 0. \quad (\text{B.3})$$

Solving this equation gives:

$$\theta_c = \arccos \frac{3K_2^2 \pm \sqrt{K_1^4 + 8K_1^2K_2^2}}{K_1^2 + 9K_2^2}. \quad (\text{B.4})$$

## References

- [1] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, *The finite element method: its basis and fundamentals*, Elsevier, 2005.

- [2] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *International Journal for Numerical Methods in Engineering* 46 (1) (1999) 131 – 150.
- [3] M. Aliabadi, Boundary element formulations in fracture mechanics, *Applied Mechanics Reviews* 50 (2) (1997) 83 – 96.
- [4] G. A. Francfort, J.-J. Marigo, Revisiting brittle fracture as an energy minimization problem, *Journal of the Mechanics and Physics of Solids* 46 (8) (1998) 1319–1342.
- [5] S. A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling, *Journal of elasticity* 88 (2007) 151–184.
- [6] H. Lee, I. S. Kang, Neural algorithm for solving differential equations, *Journal of Computational Physics* 91 (1) (1990) 110–131.
- [7] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural networks* 2 (5) (1989) 359–366.
- [8] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics* 378 (2019) 686–707.
- [9] S. Cai, Z. Mao, Z. Wang, M. Yin, G. E. Karniadakis, Physics-informed neural networks (pinns) for fluid mechanics: A review, *Acta Mechanica Sinica* 37 (12) (2021) 1727–1738.
- [10] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks for heat transfer problems, *Journal of Heat Transfer* 143 (6) (2021) 060801.
- [11] G. Kissas, Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, P. Perdikaris, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks, *Computer Methods in Applied Mechanics and Engineering* 358 (2020) 112623.
- [12] E. Haghigat, M. Raissi, A. Moure, H. Gomez, 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.
- [13] B. Yu, W. E, The deep ritz method: a deep learning-based numerical algorithm for solving variational problems, *Communications in Mathematics and Statistics* 6 (1) (2018) 1–12.
- [14] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, 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.

- [15] W. Li, M. Z. Bazant, J. Zhu, A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches, Computer Methods in Applied Mechanics and Engineering 383 (2021) 113933.
- [16] J. N. Fuhr, N. Bouklas, The mixed deep energy method for resolving concentration features in finite strain hyperelasticity, Journal of Computational Physics 451 (2022) 110839.
- [17] H. Jeong, C. Batuwatta-Gamage, J. Bai, Y. M. Xie, C. Rathnayaka, Y. Zhou, Y. Gu, A complete physics-informed neural network-based framework for structural topology optimization, Computer Methods in Applied Mechanics and Engineering 417 (2023) 116401.
- [18] B. Llanas, S. Lantarón, F. J. Sáinz, Constructive approximation of discontinuous functions by neural networks, Neural Processing Letters 27 (2008) 209–226.
- [19] S. Goswami, C. Anitescu, S. Chakraborty, T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, Theoretical and Applied Fracture Mechanics 106 (2020) 102447.
- [20] M. Manav, R. Molinaro, S. Mishra, L. De Lorenzis, Phase-field modeling of fracture with physics-informed deep learning, Computer Methods in Applied Mechanics and Engineering 429 (2024) 117104.
- [21] B. Zheng, T. Li, H. Qi, L. Gao, X. Liu, L. Yuan, Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data, International Journal of Mechanical Sciences 223 (2022) 107282.
- [22] X.-L. Yu, X.-P. Zhou, A nonlocal energy-informed neural network based on peridynamics for elastic solids with discontinuities, Computational Mechanics (2023) 1–23.
- [23] L. Ning, Z. Cai, H. Dong, Y. Liu, W. Wang, Physics-informed neural network frameworks for crack simulation based on minimized peridynamic potential energy, Computer Methods in Applied Mechanics and Engineering 417 (2023) 116430.
- [24] A. D. Jagtap, E. Kharazmi, G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113028.
- [25] A. D. Jagtap, G. E. Karniadakis, Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, Communications in Computational Physics 28 (5) (2020) 2002 – 2041.
- [26] S. Wu, B. Lu, Inn: Interfaced neural networks as an accessible meshless approach for solving interface pde problems, Journal of Computational Physics 470 (2022) 111588.
- [27] C. He, X. Hu, L. Mu, A mesh-free method using piecewise deep neural network for elliptic interface problems, Journal of Computational and Applied Mathematics 412 (2022) 114358.

- [28] Y. Diao, J. Yang, Y. Zhang, D. Zhang, Y. Du, Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology, Computer Methods in Applied Mechanics and Engineering 413 (2023) 116120.
- [29] D.-K. Jang, K. Kim, H. H. Kim, Partitioned neural network approximation for partial differential equations enhanced with lagrange multipliers and localized loss functions, Computer Methods in Applied Mechanics and Engineering 429 (2024) 117168.
- [30] Y. Wang, J. Sun, W. Li, Z. Lu, Y. Liu, Cenn: Conservative energy method based on neural networks with subdomains for solving variational problems involving heterogeneous and complex geometries, Computer Methods in Applied Mechanics and Engineering 400 (2022) 115491.
- [31] Y. Gu, C. Zhang, P. Zhang, M. V. Golub, B. Yu, Enriched physics-informed neural networks for 2d in-plane crack analysis: Theory and matlab code, International Journal of Solids and Structures 276 (2023) 112321.
- [32] Y. Gu, L. Xie, W. Qu, S. Zhao, Interface crack analysis in 2d bounded dissimilar materials using an enriched physics-informed neural networks, Engineering Analysis with Boundary Elements 163 (2024) 465–473.
- [33] Z. Chen, Y. Dai, Y. Liu, Crack propagation simulation and overload fatigue life prediction via enhanced physics-informed neural networks, International Journal of Fatigue 186 (2024) 108382.
- [34] K. Shukla, A. D. Jagtap, G. E. Karniadakis, Parallel physics-informed neural networks via domain decomposition, Journal of Computational Physics 447 (2021) 110683.
- [35] S. Wang, X. Yu, P. Perdikaris, When and why pinns fail to train: A neural tangent kernel perspective, Journal of Computational Physics 449 (2022) 110768.
- [36] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, SIAM Journal on Scientific Computing 43 (5) (2021) A3055–A3081.
- [37] Z. Wang, Z. Zhang, A mesh-free method for interface problems using the deep learning approach, Journal of Computational Physics 400 (2020) 108963.
- [38] W.-F. Hu, T.-S. Lin, M.-C. Lai, A discontinuity capturing shallow neural network for elliptic interface problems, Journal of Computational Physics 469 (2022) 111576.
- [39] Y.-H. Tseng, T.-S. Lin, W.-F. Hu, M.-C. Lai, A cusp-capturing pinn for elliptic interface problems, Journal of Computational Physics 491 (2023) 112359.
- [40] Y. Yao, J. Guo, T. Gu, A deep learning method for multi-material diffusion problems based on physics-informed neural networks, arXiv:2305.09125 (2023).

- [41] A. K. Sarma, S. Roy, C. Annavarapu, P. Roy, S. Jagannathan, Interface pinns (i-pinns): A physics-informed neural networks framework for interface problems, Computer Methods in Applied Mechanics and Engineering 429 (2024) 117135.
- [42] T. Belytschko, N. Moës, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements, International Journal for Numerical Methods in Engineering 50 (4) (2001) 993–1013.
- [43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [44] V. M. Nguyen-Thanh, C. Anitescu, N. Alajlan, T. Rabczuk, X. Zhuang, Parametric deep energy approach for elasticity accounting for strain gradient effects, Computer Methods in Applied Mechanics and Engineering 386 (2021) 114096.
- [45] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE transactions on neural networks 9 (5) (1998) 987–1000.
- [46] H. Sheng, C. Yang, Pfnn: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries, Journal of Computational Physics 428 (2021) 110085.
- [47] N. Sukumar, A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114333.
- [48] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2017).
- [49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, arXiv:1912.01703 (2019).
- [50] H. Tada, P. C. Paris, G. R. Irwin, The Stress Analysis of Cracks Handbook, Third Edition, ASME Press, 2000.
- [51] S. Chan, I. Tuba, W. Wilson, On the finite element method in linear fracture mechanics, Engineering fracture mechanics 2 (1) (1970) 1–17.
- [52] T. L. Anderson, Fracture mechanics: fundamentals and applications, CRC press, 2005.
- [53] Z. Suo, J. W. Hutchinson, Interface crack between two elastic layers, International Journal of Fracture 43 (1) (1990) 1 – 18.
- [54] N. Sukumar, Z. Huang, J.-H. Prévost, Z. Suo, Partition of unity enrichment for bimaterial interface cracks, International Journal for Numerical Methods in Engineering 59 (8) (2004) 1075 – 1102.

- [55] N. Miyazaki, T. Ikeda, T. Soda, T. Munakata, Stress intensity factor analysis of interface crack using boundary element method: Application of virtual crack extension method, JSME international journal. Ser. A, Mechanics and material engineering 36 (1) (1993) 36–42.
- [56] J. Rice, Elastic fracture mechanics concepts for interfacial cracks, Journal of Applied Mechanics, Transactions ASME 55 (1) (1988) 98 – 103.
- [57] J. Hutchinson, Z. Suo, Mixed mode cracking in layered materials, Advances in Applied Mechanics 29 (C) (1991) 63 – 191.
- [58] J. Dundurs, Edge-bonded dissimilar orthogonal elastic wedges under normal and shear loading, Journal of Applied Mechanics, Transactions ASME 36 (3) (1964) 650 – 651.
- [59] R. Bischof, M. Kraus, Multi-objective loss balancing for physics-informed deep learning, arXiv:2110.09813 (2021).
- [60] L. D. McClenny, U. M. Braga-Neto, Self-adaptive physics-informed neural networks, Journal of Computational Physics 474 (2023) 111722.
- [61] Z. Xiang, W. Peng, X. Liu, W. Yao, Self-adaptive loss balanced physics-informed neural networks, Neurocomputing 496 (2022) 11–34.
- [62] F. Erdogan, G. Sih, On the crack extension in plates under plane loading and transverse shear, Journal of Basic Engineering 85 (12 1963).
- [63] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacić, T. Y. Hou, M. Tegmark, Kan: Kolmogorov-arnold networks, arXiv:2404.19756 (2024).
- [64] Y. Wang, J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, Y. Liu, Kolmogorov arnold informed neural network: A physics-informed deep learning framework for solving pdes based on kolmogorov arnold networks, arXiv:2406.11045 (2024).
- [65] B. Moseley, A. Markham, T. Nissen-Meyer, Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations, Advances in Computational Mathematics 49 (4) (2023) 62.
- [66] V. Dolean, A. Heinlein, S. Mishra, B. Moseley, Multilevel domain decomposition-based architectures for physics-informed neural networks, Computer Methods in Applied Mechanics and Engineering 429 (2024) 117116.
- [67] L. Lu, P. Jin, G. E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, arXiv:1910.03193 (2019).
- [68] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv:2010.08895 (2020).