International Journal of Mechanical Sciences 313 (2026) 111278

Contents lists available at ScienceDirect

International Journal of Mechanical Sciences

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

Failure mechanisms and resolution in deep energy method

Xi Wang [a,*], Jidong Zhao [b,*], Zhen-Yu Yin [c], Xiaoying Zhuang [d]

[a] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, HKSAR, China, Formerly, The Hong Kong University of Science and Technology, HKSAR, China
[b] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, HKSAR, China
[c] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, HKSAR, China
[d] College of Civil Engineering, Tongji University, Shanghai 200092, China, Institute of Photonics, Leibniz University Hannover, Hannover, 30167, Germany

ARTICLE INFO

Keywords:
Physics-informed neural networks (PINNs)
Deep energy/Ritz method (DEM/DRM)
Neural operator network
Forward and inverse analysis
Finite-element-informed regularization
Inverse uncertainty quantification

ABSTRACT

The deep energy/Ritz method (DEM/DRM) offers advantages over physics-informed neural networks (PINNs), including reduced derivative orders and accelerated training. However, DEM encounters critical failure modes in both forward and inverse analyses, with underlying mechanisms and robust remedies remaining underexplored. To our knowledge, this work presents the first formal analysis that systematically identifies two distinct DEM failure modes, forward divergence and inverse collapse, and establishes their root causes along with sound countermeasures. In forward analysis, DEM training may diverge due to artificial energy minimization, where abrupt loss reductions below the physically admissible minimum occur with catastrophic errors, which are thermodynamically infeasible but remain unclarified. We prove that this stems from numerical integration inaccuracies in neural network representations, inducing pathological overfitting with escalating complexity. In inverse problems involving unknown material parameters or Neumann boundary conditions, we reveal that DEM fails because its variational formulation with respect to such unknown parameters is not well defined. To overcome these limitations, we propose a novel Energy-Informed Neural Operator Network (EINO), integrating a new regularization technique. Our framework incorporates: (1) a finite-element-informed regularization that lower-bounds the loss by the ground-truth FEM energy to ensure stability, and (2) a deep operator architecture with two-stage training that reconstructs unknown parameters/boundary conditions by embedding inverse constraints. Comprehensive benchmarks on 2D/3D linear/nonlinear solid mechanics and diffusion problems confirm EINO's superiority over DEM. EINO resolves forward divergence even on very coarse meshes and achieves substantially lower parameter errors in inverse discovery (e.g., <2% relative error under 200% Gaussian noise). The elucidated failure mechanisms and the EINO framework collectively promote physics-constrained learning for surrogate modeling and inverse uncertainty quantification, minimizing the reliance on labeled data.

1. Introduction

Solving partial differential equations (PDEs) is fundamental to modeling and analyzing a wide range of natural and social phenomena. Classical numerical methods have made significant progress in recent years, with advancements in techniques such as the finite element method (FEM), peridynamic method [1–4], material point method [5,6], discrete element method [7–11], and discontinuous deformation analysis [12–18]. Those advanced methods can extend continuum-based methods for large-deformation problems, fracturing, or contacts [2,3,19–21]. Continuous-discontinuous methods also provide transparent and reproducible ways to simulate multi-scale transgranular fracturing directly based on mineral scale structure [22]. On the other hand, indispensable experimental techniques are also developed to capture real-world phenomena [23,24]. Meanwhile, the rise of deep learning has led to remarkable achievements in processing images, text, videos, and other data types [25–29]. This has opened up new possibilities for data-driven PDE solving, offering a novel approach beyond traditional numerical methods. Unlike the abundance of data in text and image domains, engineering datasets are typically sparse and costly to acquire. However, the principles of physics are generally more clearly defined than statistical patterns in text or images. This has led to growing

* Corresponding authors.
E-mail addresses: xiwang_chn@foxmail.com, xiwang.wang@polyu.edu.hk (X. Wang), jzhao@ust.hk (J. Zhao), zhenyu.yin@polyu.edu.hk (Z.-Y. Yin), xiaoying.zhuang@gmail.com (X. Zhuang).

https://doi.org/10.1016/j.ijmecsci.2026.111278
Received 12 August 2025; Received in revised form 12 January 2026; Accepted 15 January 2026
Available online 16 January 2026

interest in physics-informed neural network (PINN) [30–34], which integrates data with physics in the form of PDEs. PINNs have achieved significant advancements and attracted widespread academic interest.

The PINN framework enforces physical constraints by incorporating PDE residuals, boundary and initial conditions, and observational data into a composite loss function. This loss function is minimized during network training using automatic differentiation. Unlike traditional numerical methods, PINNs offer a unified framework that can solve both forward and inverse problems without requiring algorithmic modification. They also mitigate the dependency on data compared to purely data-driven approaches by leveraging physics-based regularization, thereby enhancing robustness in data-scarce scenarios. PINNs have been applied in various fields, including solid mechanics [35–39], fluid mechanics [40–45], multiphysics modeling [46–50], constitutive modeling [51–56], and enhanced prediction for hazard prevention [57,58]. In parallel with the PINN approach, which uses strong-form PDEs as a loss term, the functional, energy, or variational form of PDEs can also be employed to derive physical loss. This unlocks additional flexibility and potential for further advancements in the field.

E and Yu. (2018) [59] proposed the deep Ritz method (DRM) by using the neural network as a trial function to compute the functional. Samaniego et al. (2020) [60] proposed the deep energy method (DEM) that uses neural networks as function approximation machines and focused on computational mechanics. For simplicity, DEM in this study refers to DEM/DRM. DEM was then widely applied in solid mechanics. Nguyen-Thanh et al. (2020) [61] studied finite deformation hyperelasticitiy using DEM. Mojahedin et al. (2021) [62] applied DEM to study functionally graded Euler beams. Viscoelasticity was also implemented using DEM in a two-potential constitutive framework [63]. He et al. (2023) [64] extended the DEM with radial return algorithm for elastoplasticity. Strong discontinuity in elastoplastic solids was regularized to weak discontinuity [65]. Huang and Peng (2024) used finite difference and coordinate transformation in DEM for bending, buckling and free vibration problems of irregular Kirchhoff plates [66]. Then they performed geometrically nonlinear bending analysis of laminated thin plates [67]. Bai et al. (2025) [68] solved frictionless contact problems under large deformation. Fracture simulation was also implemented in DEM. Goswami et al. (2020) [69] used transfer learning to accelerate PINN solution to fracture phase-field modeling, and then extend it to fourth-order phase field model [70] and DeepONet [71]. This framework was then further developed [72,73]. In addition to solid mechanics, DEM has also been used to solve other problems. For example, Gao et al. (2022) [74] applied DEM to steady-state heat conduction problems. Lee et al. (2024) [75] developed an adversarial DEM for solving saddle point problems with electromechanical coupling. Lin et al. (2024) [76] investigated the deep energy approach for addressing multi-physics issues in the field of piezoelectricity. Lin et al. (2024) [77] explored the deep energy method for addressing thermoelasticity problems.

To improve the performance of DEM, novel techniques have been incorporated. Mojahedin et al. (2021) [78] transformed the physical domain to a parametric domain, a unit square, accounting for the strain gradient effects. Fuhg and Bouklas (2022) [79] introduced stresses as additional outputs of neural networks to build the mixed DEM. Rezaei et al. (2022) [80] applied the mixed formulation and incorporated both strong form loss and energy form loss as physical constraints. Wang et al. (2022) [81] employed the radial basis function to satisfy Dirichlet boundary conditions and domain decomposition for heterogeneous problems. Bai et al. (2024) [82] also used radial basis functions for displacement interpolation at arbitrary points in the problem domain. Chadha et al. (2023) [83] applied random Fourier feature mapping to reduce bias toward high frequencies and used Bayesian optimization to tune the hyper parameters. Wang et al. (2024) [84] proposed the deep complementary energy method for linear elasticity based on the principle of minimum complementary energy. Eshaghi et al. (2025) [85] incorporated Fourier neural operator in DEM but is limited to 2D rectangular domains.

Comparisons between PINNs and DEM validated DEM's superiority in efficiency and accuracy. Samaniego et al. (2020) [60] claimed that DEM requires a smaller network to predict results accurately, and DEM does not need explicit treatment of the traction-free Neumann boundaries. DEM can achieve convergence in about one-tenth the number of training iterations required by PINNs. Li et al. (2021) [86] compared the performance of PINN and DEM for modeling elastic plates. It shows that DEM triumphs at a smaller number of hyperparameters and is computationally more efficient, while PINN is less mesh dependent. In the study of Ghaffari Motlagh et al. (2023) [72], compared with PINN and variational PINN, DEM presents the most accurate solutions. Notably, the DEM can achieve errors several orders of magnitude lower than those from PINNs trained with standard optimizers, even when the latter are granted substantially more computational time [87].

While DEM is widely applied and theoretically advantageous for solving PDEs, critical limitations impede its reliability. In forward problems, DEM training is plagued by a catastrophic convergence issue: the loss collapses abruptly to values far below the true energy minimum, triggering exploding errors. This violates fundamental physics principles (e.g., minimum potential energy), compromising solution integrity. For inverse analysis, where unknown parameters (e.g., material properties or Neumann boundary conditions) should be inferred, DEM fails consistently, lacking a stable and efficient optimization framework.

Although some previous studies have reported such failure phenomena, little research has systematically clarified the fundamental causes of the forward divergence, and few unified, efficient remedies have been proposed to solve the inverse problem. For instance, He et al. (2023) attributed forward divergence to strain localization instability [88], while Xiong et al. (2025) attempted to mitigate it by introducing a stiffness matrix into the DEM framework [89]. Wang et al. (2025) [90] described such phenomena as zero energy modes. However, such explanations remain underexplored and often overlook a fundamental physical constraint: a DEM energy value below the physically admissible minimum is thermodynamically infeasible. For the inverse problem, Chen et al. (2024) identified the issue but offered no solution [91]. Thombre et al. (2025) proposed a remedy, albeit at the cost of undermining the original efficiency of DEM training [92]. This study aims to bridge these gaps by fundamentally explaining these DEM failures and developing robust solutions, thereby advancing the state of the art in machine learning and computational mechanics.

To summarize, significant research gaps still exist:

(i) The underlying mechanisms driving such physically implausible failures are poorly understood, with no systematic analysis currently available.
(ii) Existing regularization techniques cannot guarantee the prevention of physics-violating loss collapses.
(iii) Unlike PINNs, there is currently no unified DEM framework capable of efficiently handling both forward and inverse analysis.

To bridge these gaps, we present a threefold contribution:

(i) **Failure mechanism diagnosis**: Through systematic diagnosis of forward divergence and inverse collapse phenomena, we deliver a rigorous mechanistic analysis of their origins.
(ii) **Provably robust framework for forward divergence**: We propose a novel regularization method inspired by finite element method and rigorously demonstrate its efficacy in completely eliminating physically implausible forward divergence even on very coarse mesh.
(iii) **A two-stage training strategy for inverse collapse**: The proposed two-stage training strategy leverages the computational efficiency of DEM to embed physical laws into a deep operator network, enabling rapid and accurate inference of unknown

material parameters (e.g., Young's modulus, Poisson's ratio) and Neumann boundary conditions during inverse analysis.

This study is structured as follows. Section 2 briefly reviews PINN and DEM. Section 3 demonstrates the two failure modes and clarifies underlying mechanisms. Section 4 proposes the methodology as a countermeasure with provable effectiveness. Section 5 validates the remedies with solid mechanics and diffusion problems in 2D and 3D, including the conventional regularization, denser Gaussian points on the same mesh, and the proposed method. Section 6 further discusses the real energy loss and nonlinear problems.

2. PINN and DEM implementations

PINN and DEM primarily differ in their approach to PDE formulations. Both methods use neural networks to approximate field variables. PINN incorporates PDE residuals at collocation points into the loss function. However, since the variational form of a PDE inherently represents a minimization problem, DEM can directly apply minimization algorithms from deep learning to functional. In this section, we will introduce the strong form of the PDE and its corresponding functional, followed by a discussion of the common implementations of both PINN and DEM.

2.1. Governing equations and energy functionals

Assuming the strong form PDE is defined as in

$$\begin{cases} L(\boldsymbol{u}) = 0, \ \text{on } \Omega, \\ \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{g}, \ \text{on } \partial\Omega_{\mathrm{D}}, \\ \mathrm{Flux}(\boldsymbol{u}(\boldsymbol{x}))\cdot\boldsymbol{n} = \boldsymbol{h}, \ \text{on } \partial\Omega_{\mathrm{N}}, \end{cases} \tag{1}$$

where $L: \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$ is the differential operator of the field variable $\boldsymbol{u}$ with dimension n_u, $\partial\Omega_{\mathrm{D}}$ is the Dirichlet boundary, and $\partial\Omega_{\mathrm{N}}$ is the Neumann boundary. $\boldsymbol{n}$ denotes the outer normal on $\partial\Omega_{\mathrm{N}}$. $\Omega \subset \mathbb{R}^{n_x}$ denotes the physical domain with dimension n_x. $\boldsymbol{u}(\boldsymbol{x}): \Omega \to \mathbb{R}^{n_u}$ is the physical variable over the domain with n_u components at point $\boldsymbol{x}$.

When the differential operator is self-adjoint or the bilinear form is symmetric, the energy functional exists. The material parameters and Neumann boundaries are treated as knowns, since the functional is minimized only with respect to field variables. If the variational form of Eq. (1) exists, the weak solution of this strong form PDE is also the unique minimizer of

$$\begin{aligned} &\min_{\boldsymbol{u}} \Pi(\boldsymbol{u}) = \min_{\boldsymbol{u}} \int e(\boldsymbol{u}), \\ &\text{s.t}: \ \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{g}, \ \text{on } \partial\Omega_{\mathrm{D}}, \end{aligned} \tag{2}$$

where the field variable $\boldsymbol{u}$ is constrained by Dirichlet boundary conditions. Π is the functional to be minimized and is integrated over the domain. In solid mechanics, the strong form and its functional are

$$\begin{cases} L(\boldsymbol{u}) = \nabla\cdot\boldsymbol{\sigma}(\boldsymbol{u}) + \boldsymbol{f} = 0, \ \text{on } \Omega, \\ \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{g}, \ \text{on } \partial\Omega_{\mathrm{D}}, \\ \boldsymbol{\sigma}\cdot\boldsymbol{n} = \boldsymbol{h}, \ \text{on } \partial\Omega_{\mathrm{N}}, \end{cases} \tag{3}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^{n_u \times n_u}$ is stress, $\boldsymbol{f} \in \mathbb{R}^{n_u}$ is the body force, and $\boldsymbol{u}(\boldsymbol{x}) \in \mathbb{R}^{n_u}$ is displacement. For linear elasticity with small displacement assumption, stress is computed with

$$\begin{cases} \boldsymbol{\sigma} = \lambda \mathrm{tr}(\boldsymbol{\varepsilon}) + 2\mu\boldsymbol{\varepsilon}, \\ \boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\boldsymbol{u} + \boldsymbol{u}\nabla), \end{cases} \tag{4}$$

where λ and μ are Lamé constants. The corresponding functional in the variational formulation is

$$\Pi(\boldsymbol{u}) = \int \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon} \mathrm{d}\Omega - \int \boldsymbol{f}\cdot\boldsymbol{u} \mathrm{d}\Omega - \int_{\partial\Omega_{\mathrm{N}}} \boldsymbol{h}\cdot\boldsymbol{u} \mathrm{d}\Gamma. \tag{5}$$

Considering finite strain hyper elasticity, the first Piola-Kirchhoff stress $\boldsymbol{P}$ is utilized. The strong form of governing equation is

$$\begin{cases} L(\boldsymbol{u}) = \nabla\cdot\boldsymbol{P} + \boldsymbol{f} = \boldsymbol{0}, \ \text{on } \Omega, \\ \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{g}, \ \text{on } \partial\Omega_{\mathrm{D}}, \\ \boldsymbol{P}\cdot\boldsymbol{n} = \boldsymbol{h}, \ \text{on } \partial\Omega_{\mathrm{N}}, \end{cases} \tag{6}$$

where $\boldsymbol{P}$ is derived from the strain energy function Ψ by $\boldsymbol{P} = \frac{\partial\Psi}{\partial\boldsymbol{F}}$. A three-dimensional Neo-Hookean type strain energy is

$$\Psi = \frac{1}{2}\mu(\mathrm{tr}(\boldsymbol{C}) - 3) - \mu \log J + \frac{1}{2}\lambda(\log J)^2, \tag{7}$$

where $\boldsymbol{C} = \boldsymbol{F}^{\mathrm{T}}\cdot\boldsymbol{F}$ and $J = \det\boldsymbol{F}$.

$$\boldsymbol{F} = \nabla\boldsymbol{u} + \mathbf{I}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix. The functional corresponding to Eq. (6) reads

$$\Pi(\boldsymbol{u}) = \int \Psi \mathrm{d}\Omega - \int \boldsymbol{f}\cdot\boldsymbol{\varphi} \mathrm{d}\Omega - \int_{\partial\Omega_{\mathrm{N}}} \boldsymbol{h}\cdot\boldsymbol{\varphi} \mathrm{d}\Gamma, \tag{9}$$

where $\boldsymbol{\varphi}(\boldsymbol{X}) = \boldsymbol{X} + \boldsymbol{u}$. $\boldsymbol{X}$ is the referential position vector.

The diffusion equation describes the process by which substances (e.g., mass, heat, or momentum) undergo spontaneous transport from regions of high concentration to regions of low concentration. Assuming a constant diffusion coefficient D, the steady-state diffusion equation is

$$\begin{cases} L(u) = D\nabla^2 u = 0, \\ u(\boldsymbol{x}) = g, \ \text{on } \partial\Omega_{\mathrm{D}}, \\ -\nabla u(\boldsymbol{x})\cdot\boldsymbol{n} = h, \ \text{on } \partial\Omega_{\mathrm{N}}, \end{cases} \tag{10}$$

where u is the density of the diffusing material. The diffusion equation can reproduce the process of a broad scope of phenomena like heat conduction and Darcy flow. The corresponding functional in the variational formulation is

$$\Pi(u) = \int_{\Omega} \frac{1}{2} D|\nabla u|^2 \mathrm{d}\Omega - \int_{\partial\Omega_{\mathrm{N}}} h\cdot u \mathrm{d}\Gamma. \tag{11}$$

It should be noted that a functional and energy denote the same concept, where "functional" is the broader mathematical term used in PDEs, and "energy" refers to its specific physical meaning in certain contexts. Furthermore, the variational formulation differs from the strong or weak forms: it poses a minimization problem for the functional/energy, whose minimizer is the solution to the corresponding weak form equation.

2.2. The physics-informed neural network and deep energy/Ritz method

Fig. 1 presents the PINN and DEM framework for unified forward and inverse analysis. Major differences are marked purple and blue for PINN and DEM, respectively. The neural network $\mathcal{N}$ is deployed to approximate the field variable $\boldsymbol{u}$. Derivatives of $\boldsymbol{u}$ with respect to $\boldsymbol{x}$ will be computed using automatic differentiation, which is entailed in the physics loss $Loss_{\mathrm{physics}}$. Loss minimization using gradient descent methods is run to find the target trainable parameters. Parameters of the neural network $\mathcal{N}$ provide the solution of $\boldsymbol{u}$. PINN utilizes the strong-form residuals as physics loss, and DEM uses the functional as physics loss. The resulting differences are discussed as follows:

(1) **Higher-order derivatives of PINN versus lower-order derivatives of DEM**

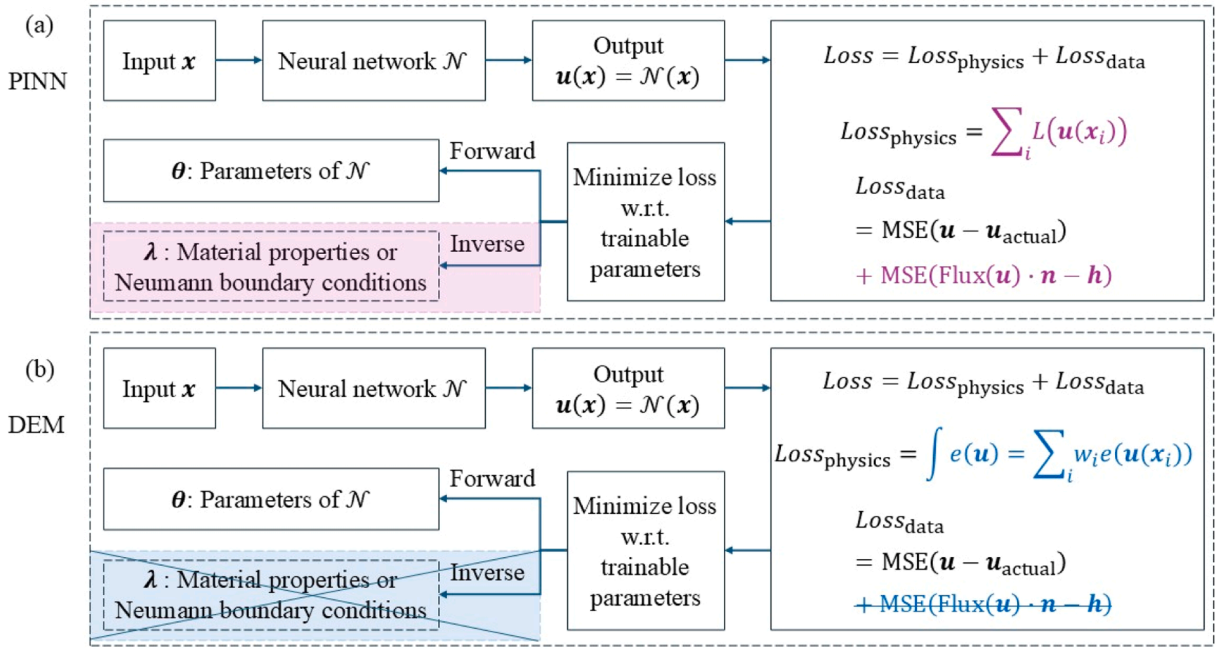

Fig. 1. Comparison of forward and inverse frameworks between PINN and DEM, with methodological distinctions highlighted by different colors. (a) PINN: It performs unified forward and inverse analysis using the same loss function, including physical loss and loss of labeled data. Physical loss is derived from PDE residuals. Data loss constrains $\boldsymbol{u}$ and flux of $\boldsymbol{u}$; (b) DEM: The physics loss is computed via numerical integration that includes the prescribed flux. The DEM data loss does not contain flux, as that is already accounted for in the physics loss. DEM is not applicable to inverse analysis in the framework.

As presented in Section 2.1, derivative order in the functional is lower than the strong-form residual. DEM leads to a weak solution that is less demanding on smoothness than those of PINN.

(2) **PINN versus DEM about the Neumann boundary conditions**
The Neumann boundary condition in DEM is naturally incorporated into the loss functional. Unlike PINN, where even zero-flux boundary conditions must be enforced through an additional loss term, the treatment of Neumann boundary conditions in DEM is intrinsically embedded in the functional, resulting in a simpler implementation.

(3) **Collocation points of PINN versus quadrature points of DEM**
Collocation points denote where constraints from PDE are imposed, a concept originating from the weighted residual method. DEM constitutes an optimization problem whose minimizer is the weak solution. The functional is evaluated by integrating over the computational domain and its boundaries. Thus, the integral must be discretized into a weighted sum of integrand evaluations at integration points.

(4) **PINN versus DEM about unified forward and inverse analysis**
PINN features unified forward and inverse analysis with nearly identical algorithms – a widely acknowledged advantage over classical numerical methods. However, for DEM, inverse analysis involving material parameters or Neumann boundary conditions is not defined in the variational statement of Eq. (2), as DEM requires a well-posed optimization problem with respect to the field variable $\boldsymbol{u}$, not other unknown parameters. The failure of DEM when applied to inverse analysis will be presented in Section 3.2.

3. Two typical failure modes in DEM

Key differences between PINN and DEM determine that DEM is more efficient and accurate than PINN as demonstrated by previous studies [60,72,86,87]. However, this advantage comes with a significant trade-off: DEM failure risks that remain uncharacterized and unclarified. This section illustrates two typical failure modes through specific examples, where underlying mechanisms and influencing factors are explained using simple mathematical formulations.

3.1. Forward divergence: failure in forward analysis

3.1.1. Identifying the failure in forward analysis

Fig. 2 depicts a commonly used setting to test PINN and DEM. An elastic cantilever beam with its left end fully constrained (fixed support) and the right end loaded by transverse distributed loading of P=1kPa. The Young's modulus and Poisson's ratio are 1×10^7Pa and 0.3, respectively. The DEM here adopts a fully connected neural network comprising 3 hidden layers, each containing 50 neurons. The neural network predictions of displacements u_x and u_y are multiplied by the input coordinate x to enforce the fixed boundary condition on the left. Fig. 2b shows the mesh to be used for numerical integration. Gaussian integration is used as in most DEM studies. There are 76 elements, and each element is assigned 4 Gaussian points, yielding 304 Gaussian points in total. The functional (namely, the total system energy) is integrated over the domain by weighted summation at Gaussian points. It was demonstrated that the combined optimization method Adam+LBFGS consistently provides a smaller loss and error than using Adam or LBFGS alone [93]. Thus, we use the combined optimizer with Adam using step size 1×10^{-3} for 5,000 steps first and then LBFGS with step size 1.0. For all experiments in this study, double-precision floating point numbers are utilized. COMSOL is used to get the reference solution. The relative error of a vector $\boldsymbol{v}$ in this study denotes the commonly used L_2 relative error that is computed with $\frac{\|\boldsymbol{v}_{\text{prediction}} - \boldsymbol{v}_{\text{real}}\|_2}{\|\boldsymbol{v}_{\text{real}}\|_2}$.

DEM simulation results of the cantilever beam are shown in Fig. 2cde, depicting the training progression via the loss function, relative error, and parameter L_2 norm. The relative error first falls below 1% at the 5,026th iteration as shown in Fig. 2f. The minimum relative error achieved is on the order of 0.1%, followed by a significant deterioration in accuracy during later stages. Surprisingly, at the 8,147$^{\text{th}}$ iteration, relative error rebounds to 10%. Fig. 2fg presents two contour snapshots of the displacements. Fig. 2f presents quite accurate results with only 1% relative error. However, the final prediction in Fig. 2g drastically

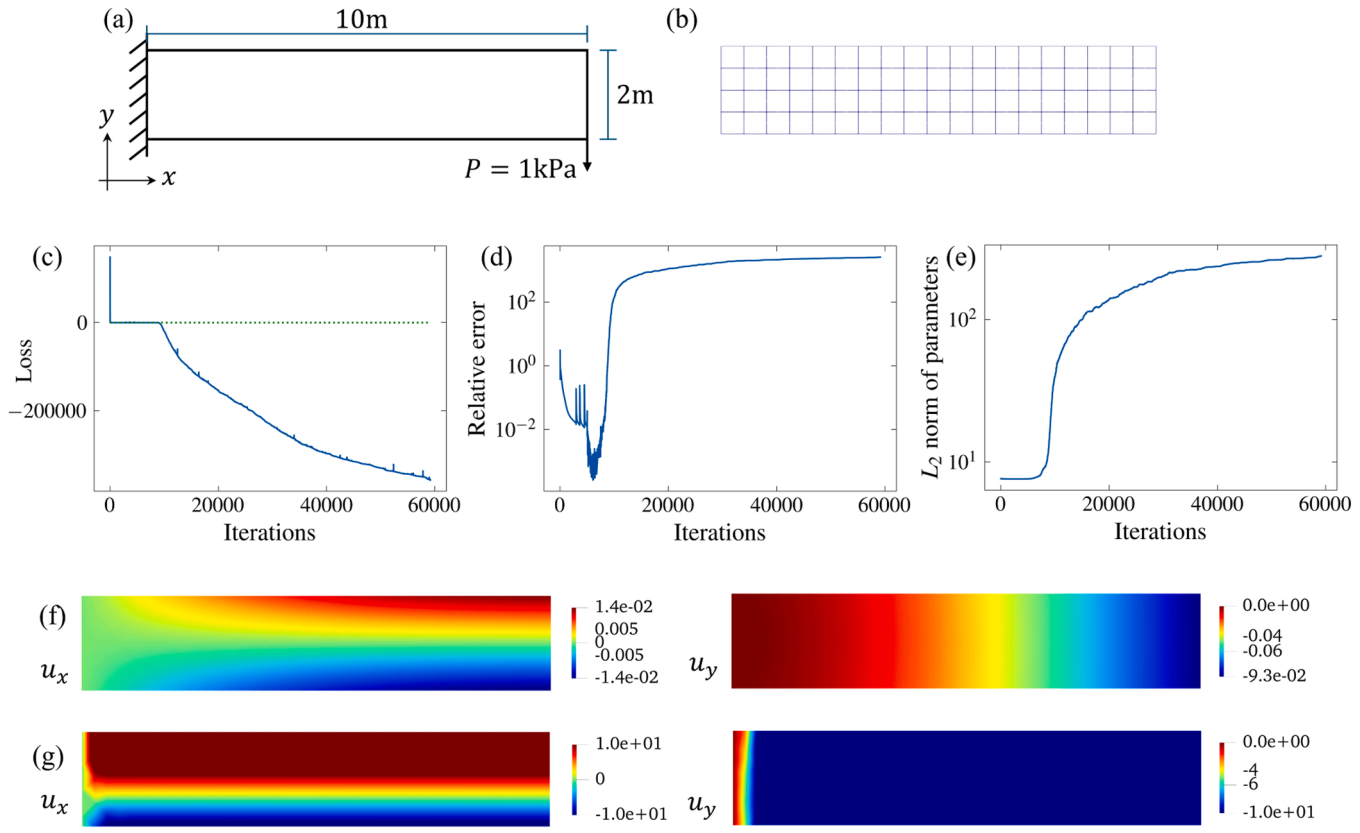

Fig. 2. Forward divergence of DEM when solving the beam problem. (a) Model configuration, including geometry and boundary conditions; (b) FEM mesh for numerical integration; (c) Loss versus number of training iterations (Green dotted line denoting the real target energy loss); (d) Relative error versus number of training iterations; (e) L_2 norm of neural network parameters versus number of training iterations; (f) Displacements u_x and u_y from DEM with 1% relative error at the 5,026$^{\text{th}}$ iteration; (g) Displacements u_x and u_y after all training at the 60,000$^{\text{th}}$ iteration.

deviates from the reference solution. Fig. 2e also indicates the increased complexity of neural network representations. Correspondingly, the loss function decreased significantly, eventually reaching values below -3.5×10^5, while the target loss value is -92. This overshooting indicates the complete failure of stable DEM convergence during forward analysis. It can also be treated as overfitting because the decrease of training loss leads to increase of validation error. The initial convergence as shown in Fig. 2f is the boundary between underfitting and overfitting.

A fundamental theoretical question emerges in the forward analysis of DEM: What mechanisms preclude the establishment of a lower bound for energy loss? This observed unbounded energy implies a potential violation of the minimum potential energy principle, revealing inherent limitations in DEM. The next section will elaborate on the failure mechanisms with simple and straightforward mathematical formulations.

3.1.2. *DEM fails due to inaccurate numerical integration*

This section employs a canonical prototype problem to elucidate the fundamental mechanisms underlying the methodological failure. Fig. 3a presents the configuration of the 1D bar under a tension F at the right end. E is the elastic modulus. A is the cross-sectional area of the bar. The analytical solution is a linear displacement field in Fig. 3b, with displacement at the right end as u_1. The potential energy Π is

$$\Pi = \int_{\Omega} \frac{1}{2}\sigma : \varepsilon \mathrm{d}\Omega - \int_{\partial\Omega_{\mathrm{N}}} h\cdot u \mathrm{d}\Gamma = \int_{x=0}^{x=L} \frac{1}{2}\left(E\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial x} A \mathrm{d}x - Fu_1, \tag{12}$$

where ε is the strain computed as $\varepsilon = \frac{\partial u}{\partial x}$. σ is the stress derived from $\sigma = E\varepsilon = E\frac{\partial u}{\partial x}$. When approximating bar displacements using a single

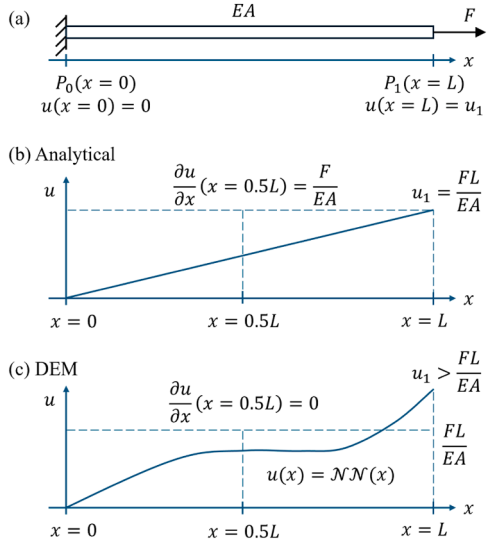

Fig. 3. The 1D bar test: (a) Configuration, (b) analytical solution, and (c) a possible divergent solution of DEM. Note: this divergent solution results in an energy loss much smaller than the minimum potential energy, which is unphysical.

element with nodes P_0 and P_1, the integral in Eq. (12) is discretized as weighted summation,

$$\Pi = \frac{EAL}{2}\left(\frac{\partial u}{\partial x}(x = 0.5L)\right)^2 - Fu_1. \tag{13}$$

The correct solution is $\frac{\partial u}{\partial x}(x=0.5L)=\frac{F}{EA}$ and $u_1=\frac{FL}{EA}$, and thus the true energy $\prod_{\text{true}}$ is

$$\prod_{\text{true}}=\frac{F^2L}{2EA}-\frac{F^2L}{EA}=-\frac{F^2L}{2EA}. \tag{14}$$

If we use a neural network to approximate the solution by $u=\mathcal{N}(x)$, it may converge to a wrong state by minimizing the potential energy Π. For example, Fig. 3c shows an erroneous solution yielded by a neural network, with $\frac{\partial u}{\partial x}(x=0.5L)=0$ and $u_1>\frac{FL}{EA}$. The corresponding potential energy is given by

$$\prod=0-Fu_1<-\frac{F^2L}{EA}<\prod_{\text{true}}=-\frac{F^2L}{2EA}. \tag{15}$$

Moreover, a neural network can produce much lower loss values below the true energy if $\frac{\partial u}{\partial x}(x=0.5L)=0$ and $u_1\gg\frac{FL}{EA}$. Thus, by minimizing the potential energy, the neural network can converge to erroneous solutions much lower than the true energy. Notably, the erroneous solution in Fig. 3c exhibits higher complexity compared to the true solution in Fig. 3b. Such phenomena are consistent with Section 3.1.2. It also indicates that such forward divergence will not go to negative infinity if all boundary conditions are Dirichlet boundaries, since the minimum elastic potential energy is zero.

The fundamental reason for this failure results from the discretization of the energy in Eq. (12). The accuracy of numerical integration techniques (e.g., Gaussian quadrature commonly implemented in DEM frameworks) becomes unreliable when applied to integrands containing highly complex functions. However, the neural network architecture inherently constitutes a highly complex nonlinear function whose expressiveness remains challenging to quantify due to its opaque internal representations. To be more specific, $\frac{\partial u}{\partial x}$ from the neural network for elastic potential energy $\int_{x=0}^{x=L}\frac{1}{2}\left(E\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial x}A\mathrm{d}x$ in Eq. (12) can be very complex and numerical integration cannot capture the correct solution. $\frac{1}{2}\left(E\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial x}$ of Fig. 3c is clearly nonzero, but the neural network can lead to localized loss minimization exclusively at integration points while potentially violating physics elsewhere in the domain. For example, Fig. 3c sets $\frac{\partial u}{\partial x}=0$ at integration points, leading to zero elastic potential energy $\int_{x=0}^{x=L}\frac{1}{2}\left(E\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial x}A\mathrm{d}x$ In summary, inaccurate numerical integration results in the failure of DEM in forward analysis.

3.1.3. Sensitivity of mesh density

This section explores the influence of mesh density on the failure in forward analysis. The settings are from Section 3.1.1, except for mesh density. The study considered various scenarios with 20, 45, 76, 168, 500, and 2,000 elements, respectively, with each scenario run five times. Fig. 4 displays the iteration threshold (mean and standard deviation) at which relative error regresses to 10% as a function of spatial discretization in DEM training. This can also be regarded as overfitting and the complexity of neural network exceeds the power of numerical integration. A reduced iteration threshold correlates with increased susceptibility to training failure. As the mesh density increases, DEM becomes less prone to fail in training for forward analysis. This is attributed to the fact that a finer mesh enhances the accuracy of numerical integration.

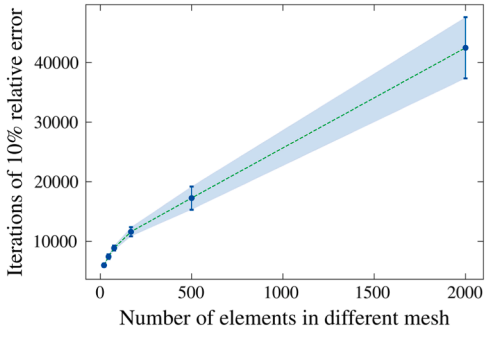

Fig. 4. Influence of mesh density on the forward divergence of DEM: Number of iterations for diverging to 10% relative error versus number of elements in different mesh (error bars represent ±1 standard deviation).

3.1.4. Influence of neural network architecture

This section explores the width and depth of a neural network with respect to the number of iterations at which the relative error regresses to 10%. The rest settings are identical to Section 3.1.1. Fig. 5 reports the mean iteration numbers and standard deviations of different architectures. Fig. 5a demonstrates a clear trend that fewer layers or neurons contribute to less susceptibility to training failure. This is attributed to the fact that numerical integration can handle simpler neural networks more accurately due to diminished complexity in representation. Fig. 5b displays the standard deviation of five runs for each configuration.

3.2. Inverse collapse: failure in inverse analysis

3.2.1. Identifying failure mode in inverse analysis for material parameters and Neumann boundary conditions

Fig. 1 in Section 2.2 illustrates the unified forward and inverse analysis of PINN. However, in DEM, inverse analysis is not defined for the variational statement. This section demonstrates the failure in inverse analysis using DEM. The cantilever beam case in Section 3.1.1 is utilized for testing inverse analysis. The displacements at mesh nodes serve as labeled data. Two scenarios, unknown elastic modulus and unknown boundary force are tested. We use the combined optimizer with Adam using learning rate 1×10^{-1} for 5,000 steps first and then LBFGS with step size 1.0. Fig. 6a and Fig. 6b present the results of inverse analysis for modulus. Young's modulus is initialized with the true value 1×10^{7}Pa. However, during training, the modulus progressively deviates from its initial ground truth. Finally, the energy loss and modulus exhibit unbounded negative divergence during later training stages. Fig. 6c and Fig. 6d reveal identical divergence patterns in the inverse analysis of material parameters and Neumann boundary conditions. The reference energy loss is -92, while the final energy loss is far below -1×10^{6}. The Neumann boundary force also plunges towards negative infinity. The underlying reason for such failure in DEM inverse analysis will be elaborated in the next section.

3.2.2. DEM fails due to undefined functional

As discussed in Section 2.2, for DEM method, the inverse analysis for material parameters and Neumann boundary conditions are not defined in the variational statement of Eq. (2). DEM depends on a well-defined optimization problem with respect to the function of field variable $\boldsymbol{u}$, not other unknown parameters. Thus, unexpected failure behaviors will be observed when trying to unify forward and inverse analysis like that in PINN.

Taking the 1D bar test introduced in Section 3.1.2 as an example, the potential energy is $\Pi=\int_{x=0}^{x=L}\frac{1}{2}\left(E\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial x}A\mathrm{d}x-Fu_1$. Inverse analysis usually requires solving unknown material parameter E and Neumann boundary F. However, minimizing the potential energy with respect to unknown E and F cannot solve the inverse problem. As a function of E and F, Π is not lower bounded, and thus minimizing Π with respect to E and F will drive the loss into negative infinity. However, for Dirichlet boundary conditions, they are essentially unknown field variables at boundaries, and thus their inverse analysis is not limited like Neumann

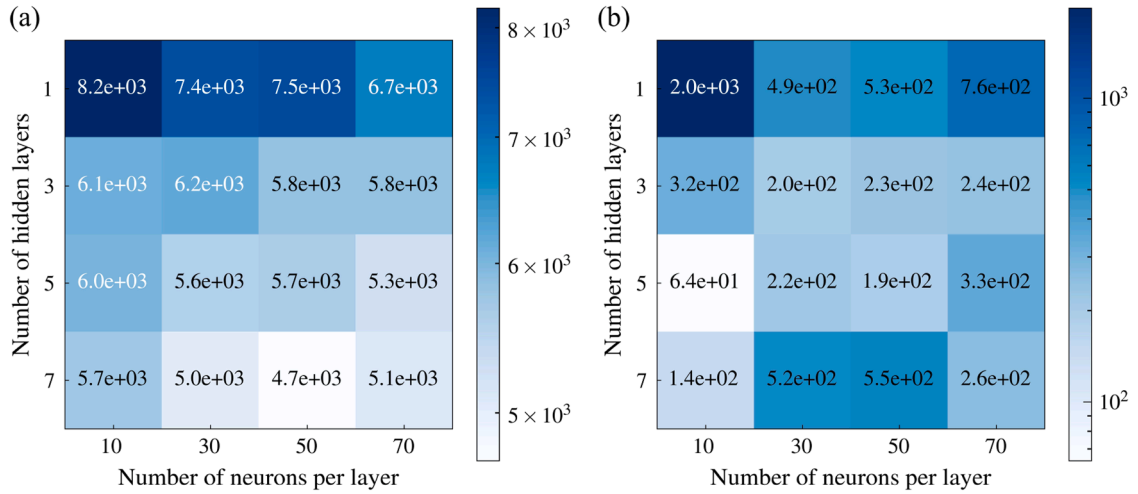

Fig. 5. Influence of neural network architecture on the forward divergence of DEM: Heat map of number of iterations for diverging to 10% relative error with respect to different neural network architectures (specified by hidden layers and neurons per layer): (a) Mean value of number of iterations; (b) Standard deviation of number of iterations.

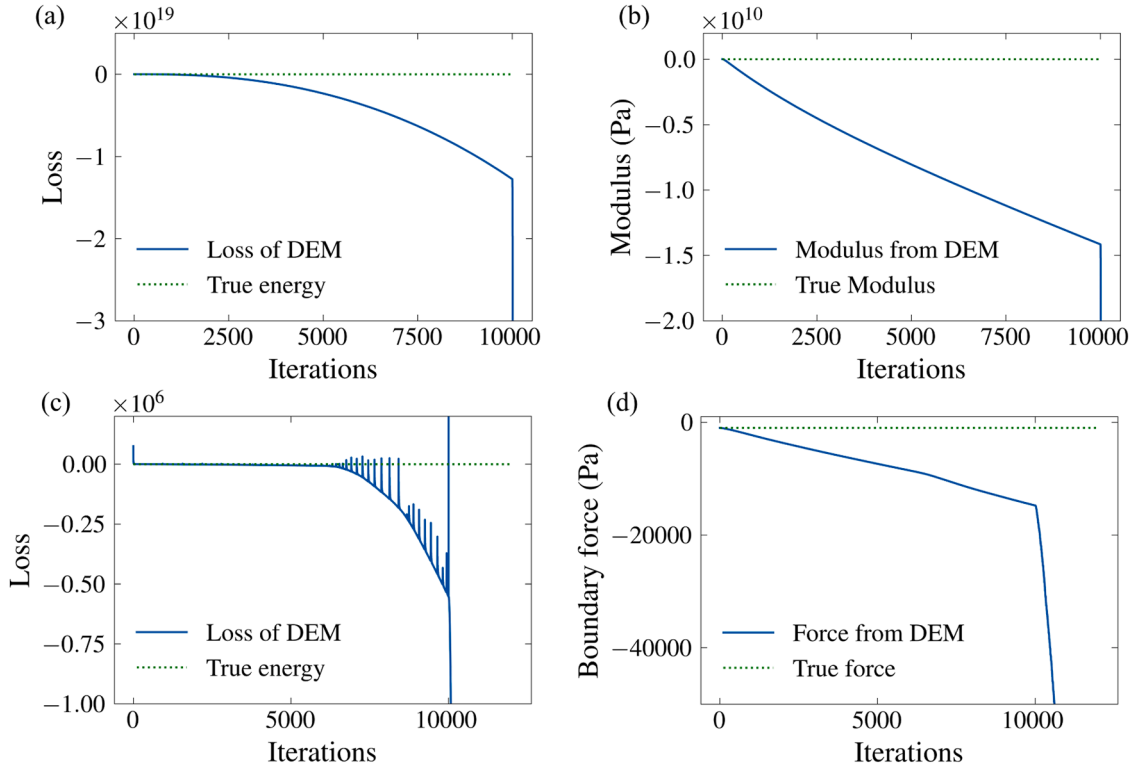

Fig. 6. Inverse collapse of DEM to solve unknown material parameters and Neumann boundary conditions (Green dotted line denoting the true value). (a) Losses versus training iterations in inverse analysis for unknown Modulus; (b) Modulus versus training iterations; (c) Losses versus training iterations in inverse analysis for unknown Neumann boundary condition; (d) Neumann boundary force versus training iterations.

boundary and material parameters.

4. EINO: methodology and robustness

As established, DEM is superior in efficiency and accuracy compared to PINN. Yet, the variational formulation is a well-defined optimization problem solely with respect to unknown displacements and relies on precise numerical integration. Consequently, DEM exhibits divergence risks in forward analysis and is inherently ill-suited for inverse analysis. This study proposes the Energy-Informed Neural Operator Network (EINO) as a provably robust framework.

4.1. EINO with novel finite-element-informed regularization and two-stage training strategy

During the training of neural networks, it is common to add additional regularization terms to reduce model complexity and promote smoothness, such as L_1 regularization (Lasso regularization) and L_2 regularization (Ridge regression). Nevertheless, these methods do not ensure that the output of a neural network will necessarily fall within the

intended solution space. As presented in Fig. 7, we deploy the finite-element-informed regularization to constrain the solution space of neural networks to be the same as that of finite element methods. Compared to conventional L_1/L_2 regularization, the finite-element-informed regularization does not add extra loss terms. Applying the finite-element approximation after the neural network, field variables are determined by interpolation functions on elements. Assuming the nodes of an element are $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_{\text{node}}}\}$, the regularized solution $\boldsymbol{u}_{\text{reg}}$ and derivatives $\frac{\mathrm{d}^n \boldsymbol{u}_{\text{reg}}}{\mathrm{d}\boldsymbol{x}^n}$ are interpolated from the nodal variables $\{\boldsymbol{u}_1(\boldsymbol{x}_1), \boldsymbol{u}_2(\boldsymbol{x}_2), \ldots, \boldsymbol{u}_{n_{\text{node}}}(\boldsymbol{x}_{n_{\text{node}}})\}$,

$$\begin{cases} \boldsymbol{u}_{\text{reg}} = \sum_{i=1}^{n_{\text{node}}} l_i \boldsymbol{u}_i, \\ \dfrac{\mathrm{d}^n \boldsymbol{u}_{\text{reg}}}{\mathrm{d}\boldsymbol{x}^n} = \sum_{i=1}^{n_{\text{node}}} \dfrac{\mathrm{d}^n l_i}{\mathrm{d}\boldsymbol{x}^n} \boldsymbol{u}_i, \end{cases} \tag{16}$$

where l_i $(i = 1 \ldots n_{\text{node}})$ are nodal basis functions to interpolate nodal solutions of the neural network over the element. Since field variables are governed by nodal values, Dirichlet boundary conditions can be enforced straightforwardly by assigning target values to boundary nodes. In this study, we implement two kinds of elements, namely the second-order serendipity quadrilateral element and the second-order Lagrange tetrahedral element as elaborated in the Appendix.

For general elements, the coordinate transformation should be applied to derivatives and integral. Partial derivatives with respect to general coordinates $\boldsymbol{x}=\{x, y, z\}$ can be transferred to those with respect to reference coordinates $\bar{\boldsymbol{x}} = \{\bar{x}, \bar{y}, \bar{z}\}$,

$$\begin{cases} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial}{\partial \bar{x}} \\ \frac{\partial}{\partial \bar{y}} \\ \frac{\partial}{\partial \bar{z}} \end{bmatrix}, \\ \int \mathrm{d}x\mathrm{d}y\mathrm{d}z = \int |\boldsymbol{J}| \mathrm{d}\bar{x}\mathrm{d}\bar{y}\mathrm{d}\bar{z}, \\ J = \begin{bmatrix} \frac{\partial x}{\partial \bar{x}} & \frac{\partial x}{\partial \bar{y}} & \frac{\partial x}{\partial \bar{z}} \\ \frac{\partial y}{\partial \bar{x}} & \frac{\partial y}{\partial \bar{y}} & \frac{\partial y}{\partial \bar{z}} \\ \frac{\partial z}{\partial \bar{x}} & \frac{\partial z}{\partial \bar{y}} & \frac{\partial z}{\partial \bar{z}} \end{bmatrix}, \end{cases} \tag{17}$$

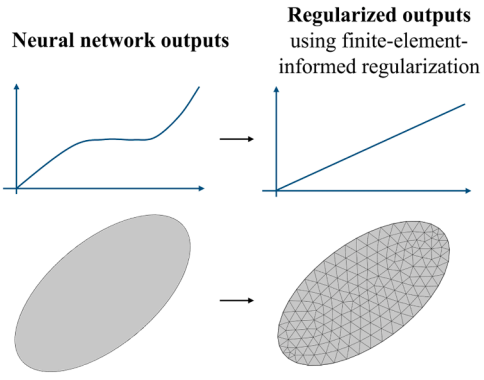

Fig. 7. The finite-element-informed regularization. Note that the output of blackbox neural network is regularized to the finite-element solution space (The regularization of 1D function above and 2D function below).

where $\boldsymbol{J}$ is the Jacobian matrix from reference coordinates to general coordinates, and $|\boldsymbol{J}|$ is its determinant.

Fig. 8 demonstrates the two-stage training strategy for EINO. The deep operator network (DeepONet) [94] is adopted in EINO. It leverages a universal approximation theorem to learn mappings between function spaces, enabling efficient and accurate modeling of complex physical systems. Its advantages include high generalization across diverse PDEs and reduced computational cost compared to traditional numerical solvers. DeepONet employs dual deep neural networks: a branch net encoding input function like material parameters and boundary conditions $\boldsymbol{\lambda}$, and a trunk net encoding continuous output domain coordinate $\boldsymbol{x}$. Their outputs are combined via a dot product to approximate nonlinear operators $\boldsymbol{u}(\boldsymbol{\lambda})(\boldsymbol{x})$, enabling learning mappings between infinite-dimensional function spaces. In this study, branch and trunk nets are set as fully-connected neural networks as shown in

$$\begin{cases} \left[\boldsymbol{b}_1^i, \boldsymbol{b}_2^i, \cdots, \boldsymbol{b}_{n_{hidden}}^i\right] = \left(\boldsymbol{A}_{n_{branch}} \circ \sigma_{n_{branch}-1} \circ \boldsymbol{A}_{n_{branch}-1} \circ \cdots \circ \sigma_1 \circ \boldsymbol{A}_1\right)_i (\boldsymbol{\lambda}), \\ \left[\boldsymbol{t}_1^i, \boldsymbol{t}_2^i, \cdots, \boldsymbol{t}_{n_{hidden}}^i\right] = \left(\boldsymbol{A}_{n_{trunk}} \circ \sigma_{n_{trunk}-1} \circ \boldsymbol{A}_{n_{trunk}-1} \circ \cdots \circ \sigma_1 \circ \boldsymbol{A}_1\right)_i (\boldsymbol{x}), \\ \boldsymbol{u}_i(\boldsymbol{\lambda})(\boldsymbol{x}) = \sum_{k=1}^{n_{hidden}} \boldsymbol{b}_k^i \boldsymbol{t}_k^i. \end{cases} \tag{18}$$

If there is no need for inverse analysis of material parameters and boundaries $\boldsymbol{\lambda}$, we can ignore the branch net and the DeepONet is reduced to a fully-connected neural network.

As depicted in Fig. 8, EINO training includes offline and online stages. Fig. 8a presents the offline stage for training a neural network to solve the forward problem. The output $\boldsymbol{u}$ from neural network is not directly used in the loss function, instead, the regularized $\boldsymbol{u}_{\text{reg}}$ will be applied to compute the numerical integration. Unlike the blackbox outputs of a neural network, whose representational complexity makes them difficult to evaluate, $\boldsymbol{u}_{\text{reg}}$ is interpretable and can be easily handled by numerical integration. However, the data loss term MSE($\boldsymbol{u}_{\text{reg}}or\boldsymbol{u}$ − $\boldsymbol{u}_{\text{actual}}$) does not involve integration and thus can use $\boldsymbol{u}_{\text{reg}}$ or $\boldsymbol{u}$ interchangeably. Parameters of the neural network $\boldsymbol{\theta}$ are updated by minimizing the total loss. Notably, this approach can eliminate the need for extensive labeled datasets, enabling the training of models across diverse boundary conditions and material parameters without reliance on data labels. This constitutes a promising strategy for training large models for scientific computing. On the other hand, if material parameters and boundary conditions are all set, the branch net can be omitted and the DeepONet degrades back to a fully connected neural network like that in DEM.

The offline stage embeds physics into the neural network, and thus during the online stage of Fig. 8b, inverse analysis can be readily performed by minimizing the data loss with respect to $\boldsymbol{\lambda}$. The finite-element-informed regularization in the online stage is optional because there is no integration. However, the finite-element-informed regularization possesses an additional advantage in strictly enforcing the Dirichlet boundary conditions. Since EINO only needs to minimize data loss with respect to unknown $\boldsymbol{\lambda}$ in the inverse training, it will be much faster than minimizing both data and physics loss with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ like the PINN inverse analysis in Fig. 1. Leveraging DEM's inherent computational efficiency over PINNs in forward training, the proposed EINO framework effectively exploits the energy formulation of PDEs, significantly enhancing the robustness for efficient forward modeling and inverse analysis.

4.2. EINO preventing DEM failures

The EINO framework in Fig. 8 incorporates built-in mechanisms that effectively prevent typical failure modes of DEM for both forward and inverse problems, as discussed in Section 3. To demonstrate this capability, we employ the bar test configuration shown in Fig. 9, which shares identical settings with Fig. 3 (Section 3.1.2).

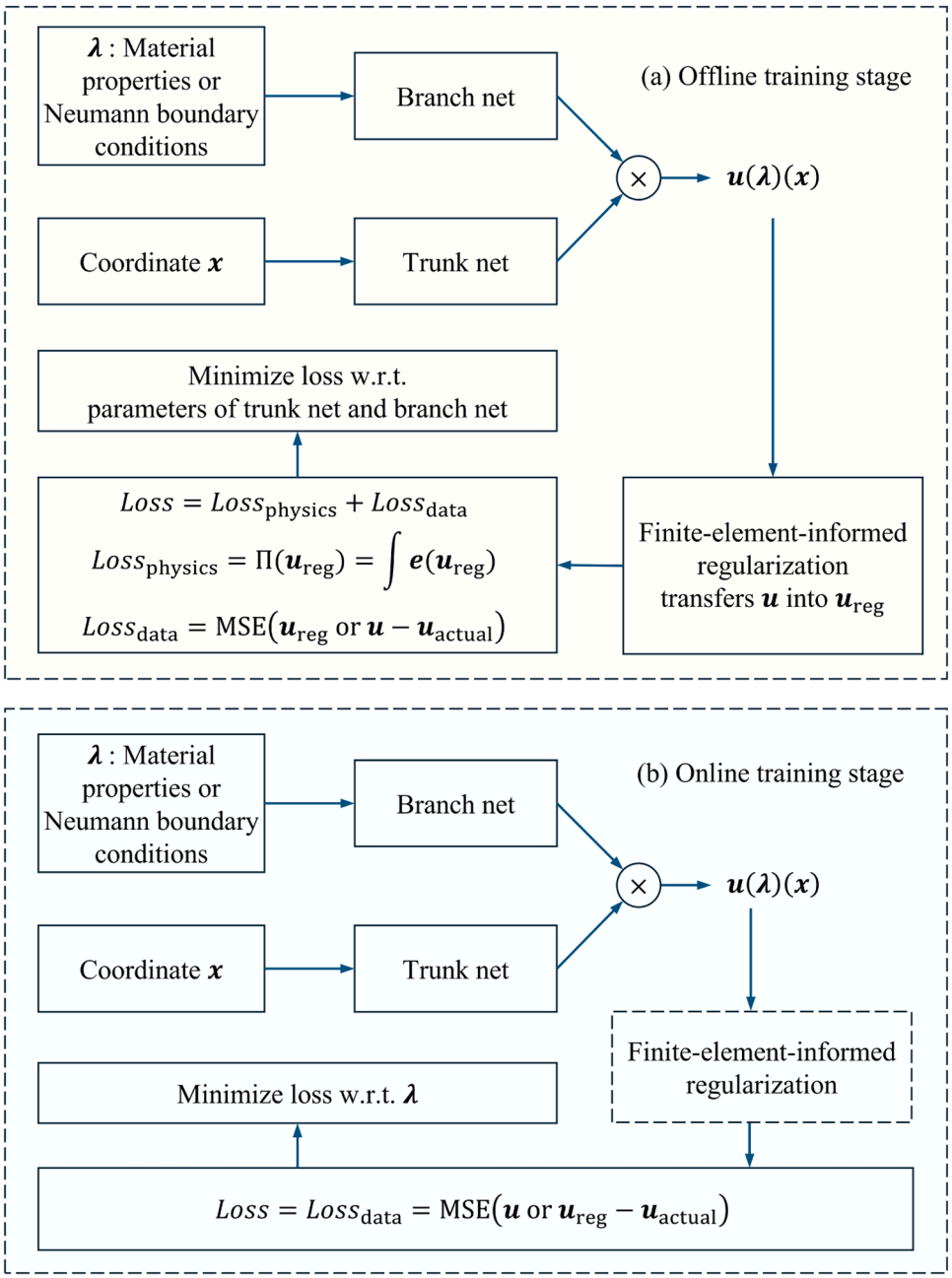

Fig. 8. EINO with a two-stage training strategy. (a) Offline training stage to train a DeepONet to learn the mapping between different parameters λ and solutions; (b) Online training stage: The process involves an online training stage to solve the unknown solution and the parameter λ simultaneously, given only sparse data constraints. Note that forward analysis in online stage is just the direct inference of the pretrained DeepONet without training.

Fig. 9b displays the erroneous DEM solution that is represented by a neural network $u = \mathcal{N}(x)$. As demonstrated in Section 3.1.2, this erroneous solution can drive energy loss into negative infinity if $\frac{\partial u}{\partial x}(x = 0.5L)$ is close to zero and u_1 increases towards positive infinity. Such failures seem to violate the minimum energy principle, and the fundamental cause is the blackbox nature of neural networks, which can hardly be accurately captured by numerical integration. Notably, EINO adopts the finite-element-informed regularization. Fig. 9c demonstrated the regularized solution u_{reg} by linear basis functions. The field displacement will be interpolated by the nodal basis function $l_1(x)$ and the nodal displacement $u_1 = \mathcal{N}(x = L)$:

$$u_{\text{reg}}(x) = \sum_{i=1}^{n_{\text{node}}} l_i(x) u_i = \frac{x}{L} u_1 = \frac{x}{L} \mathcal{N}(x = L). \tag{19}$$

Potential energy is a quadratic function of nodal predictions,

$$\Pi(u_{\text{reg}}) = \frac{EAL}{2} \left(\frac{\partial u_{\text{reg}}}{\partial x}(x = 0.5L) \right)^2 - F u_1 = \frac{EA u_1^2}{2L} - F u_1. \tag{20}$$

$u_1 = \frac{FL}{EA}$ minimizes $\Pi(u_{\text{reg}})$ to $-\frac{F^2 L}{2EA}$, which is the true potential energy of the corresponding FEM. By optimizing the loss function $\Pi(u_{\text{reg}})$ with respect to nodal predictions u_i, neural network $\mathcal{N}(x)$ in EINO will not produce a loss smaller than the true energy like that in Section 3.1.1. Since the regularized solution u_{reg} shares the same space with FEM, $\Pi(u_{\text{reg}})$ will also be lower bounded by the minimum functional value of FEM. Eq. (21) presents that $\Pi(u_{\text{reg}})$ is a function of u_i and u_i is a function of $\boldsymbol{\theta}$. Thus, the gradient of loss with respect to neural network parameters $\boldsymbol{\theta}$ is computed using the chain rule,

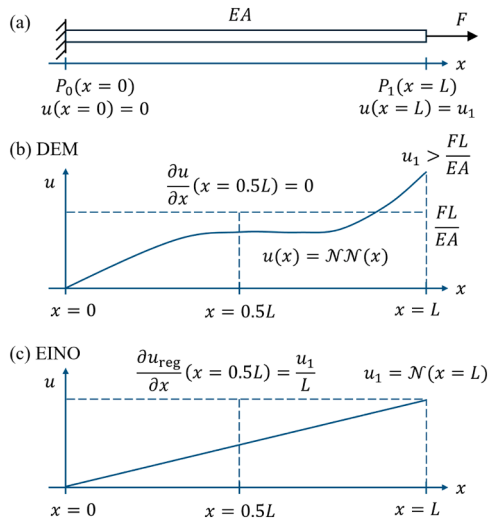

Fig. 9. The 1D bar test: Configuration, diverged DEM solution, and regularized EINO solution. (a) Configuration of the test; (b) Diverged solution of DEM leading to much smaller loss than true energy; (c) The correct solution of EINO due to finite-element informed regularization.

$$\begin{cases} \Pi(\boldsymbol{u}_{\text{reg}}) = \int e\left(\sum_{i=1}^{n_{\text{node}}} l_i(\boldsymbol{x}) u_i(\boldsymbol{\theta})\right) \mathrm{d}\boldsymbol{x}, \\ \dfrac{\partial \Pi}{\partial \boldsymbol{\theta}} = \dfrac{\partial \Pi}{\partial \boldsymbol{u}_{\text{reg}}} \dfrac{\partial \boldsymbol{u}_{\text{reg}}}{\partial \boldsymbol{\theta}}. \end{cases} \tag{21}$$

The derivation of $\frac{\partial \Pi}{\partial \boldsymbol{u}_{\text{reg}}}$ is equivalent to computing the residuals in FEM. With finite-element-informed regularization, $\Pi(\boldsymbol{u}_{\text{reg}})$ denotes the same functional computed using FEM and thus is lower-bounded by the ground-truth FEM energy. Once Π reaches minimum, $\frac{\partial \Pi}{\partial \boldsymbol{u}_{\text{reg}}}$ becomes zero and the neural network ceases training, preventing overfitting. $\boldsymbol{u}_{\text{reg}}$ is a function of neural network parameters in EINO because the nodal variables are predicted by a neural network. Thus, during forward analysis, EINO does not diverge after attaining the correct solution of FEM.

Clearly, EINO avoids the inverse analysis failures characteristic of DEM. DEM fails because the minimization problem in the variational statement is not defined with respect to unknown material parameters and boundary conditions. EINO already learns the mappings from coordinates, material parameters, and boundary conditions to solutions during offline stage training. Online inverse training will not minimize the energy functional with respect to unknown material parameters and boundary conditions, and thus EINO can completely prevent the inverse collapses in DEM.

In summary, DEM exhibits two fundamental limitations: (1) Forward divergence stemming from numerical integration inaccuracies when handling complex neural network representations, and (2) inverse collapse caused by its variational formulation being defined exclusively for unknown field variables (undefined optimization with respect to material parameters and Neumann boundaries). EINO overcomes these through: (1) Finite-element-informed regularization that guarantees integration accuracy by construction, with training automatically terminating at the finite element solution's energy minimum; (2) A decoupled training paradigm where offline learning establishes parameter-to-solution mappings while preserving DEM's rapid training advantages, enabling online inverse analysis via direct data loss minimization - eliminating the need for problematic energy functional optimization.

5. Validation of remedies

This section presents a comprehensive validation of EINO along with alternative approaches to address the two failure modes inherent in DEM. The first two subsections examine the efficacy of L_2 regularization and enhanced integration point density strategies. The last two subsections validate EINO for solving failures in forward and inverse analysis, respectively.

5.1. L_2 regularization

This section explores the influence of L_2 regularization. L_2 regularization, also known as Ridge regularization, is a widely used technique in machine learning to prevent overfitting by adding a penalty term to the loss function. This term is the sum of the squared magnitudes of the model parameters (weights), scaled by a hyperparameter that we refer to as regularization coefficient γ, expressed as $\gamma\sum_i \theta_i^2$. By penalizing large weights, L_2 regularization promotes the development of simpler models. Given that the failure of DEM in inverse analysis stems from inaccurate integration, L_2 regularization can facilitate more precise integration by encouraging simpler representations. The beam test with 20 elements described in Section 3.1.1 is used to investigate the effectiveness of L_2 regularization as a remedy. Different regularization coefficients are tested. Fig. 10a presents the iteration threshold (mean and standard deviation) at which relative error regresses to 10% as a function of γ at $[1\times10^{-3}, 1\times10^{-2}, 1\times10^{-1}, 2\times10^{-1}]$. The trend is straightforward, and we can conclude that L_2 regularization helps mitigate failure during forward analysis. However, DEM still fails. We further increase γ to further punish the complexity of neural network. Fig. 10b presents the final relative error for each run of $\gamma = [3\times10^{-1}, 4\times10^{-1}, 5\times10^{-1}, 1\times10^{0}]$. Those larger than 1.0 denote failure in forward analysis. We set the maximum total iterations as 120,000. With the increase of γ, the number of failed runs decreases, at the sacrifice of accuracy. It can be concluded that L_2 regularization can postpone the failure iterations of forward analysis in DEM. However, the value of regularization coefficient γ is hard to quantify for each case. Small γ is not effective enough, but large γ decreases accuracy. It cannot guarantee all runs will converge and thus can only be employed as a temporary remedial measure. As detailed in Section 5.3, the proposed EINO effectively addresses the failure problem without requiring hyperparameter tuning or compromising accuracy.

5.2. Increasing integration points while keeping the same mesh

One common method to enhance the accuracy of DEM is to increase the mesh density. However, denser mesh discretization incurs higher computational costs, and mesh generation has long been a challenging aspect of traditional numerical methods. Therefore, we propose to only add more integration points while keeping the same mesh. Simply adding more Gaussian points does not incur higher interpolation order or increasing elements because it only increases the accuracy of numerical integration. The same configuration of the beam test is used as in Section 5.1. The mesh has only 20 elements, but we increase the number of Gaussian points in each element from 4 to 25. Fig. 11 presents the iteration threshold (mean and standard deviation) at which relative error regresses to 10% as a function of total Gaussian points. This increasing trend supports the beneficial effect of increasing the number of Gaussian points. Compared with remeshing, adding more Gaussian points to each cell is simpler and more efficient to implement. Nevertheless, the failure in forward analysis cannot be entirely avoided.

5.3. Numerical experiments for solving failure in forward analysis

This section validates the proposed EINO for solving failure in forward analysis. Two experiments are conducted. The first is a classical cantilever beam test, and the second is a diffusion problem that covers a broad scope of phenomena.

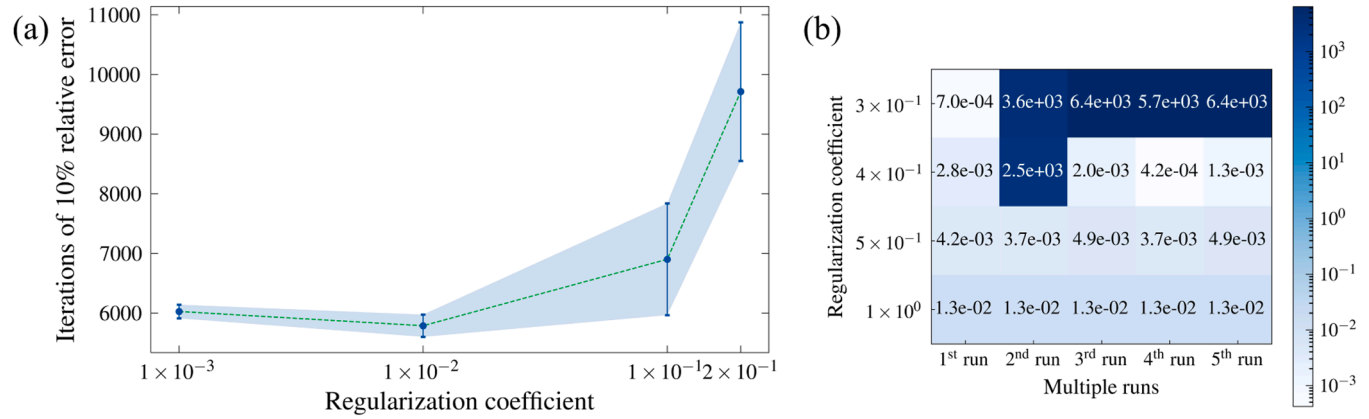

Fig. 10. Influence of different regularization coefficients on the forward divergence of DEM. (a) Number of iterations for diverging to 10% relative error versus regularization coefficient (error bars represent ±1 standard deviation); (b) Heatmap of relative errors after each run in the case of different regularization coefficients.

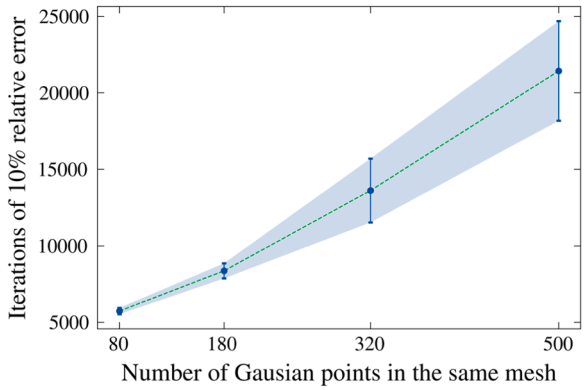

Fig. 11. Number of iterations for diverging to 10% relative error versus number of Gaussian points in the same mesh (error bars represent ±1 standard deviation).

5.3.1. *Cantilever beam*

The same configuration as in Section 5.1 is taken to check if EINO can handle the failure during forward analysis. For EINO, the maximum number of iterations is set as 110,000. Quadratic serendipity elements are deployed for regularization in EINO. Fig. 12 compares DEM and EINO with respect to loss, relative error, and L_2 norm versus training iterations. DEM diverges to 10% relative error at the 6,002nd iteration with plunging loss far below the true energy loss, exploding error, and surging complexity of neural network representation. As a striking contrast, EINO maintains robust loss, accuracy, and L_2 norm of neural network parameters, even after the number of training iterations reaches 110,000. To further validate EINO, we run the test 10 times, finally, the mean relative error is 0.33%, and the standard deviation is 0.033%. The robust and accurate performance also supports the proposed failure mechanisms in Section 3.1.2 and discussions in Section 5.2.

5.3.2. *Diffusion problem*

The diffusion equation describes a broad scope of processes by which substances (e.g., mass, heat, or momentum) undergo spontaneous transport from regions of high concentration to regions of low concentration. Thus, the diffusion equation as described in Eq. (10) is solved by minimizing the functional in Eq. (11). The diffusion coefficient D is set as $1\text{m}^2\text{s}^{-1}$. As shown in Fig. 13a, the left boundary of the domain is Dirichlet boundary with u=1. The bottom boundary is Neumann boundary with $D\frac{\partial u}{\partial y}=1$. The rest are zero flux boundaries. Since material parameters and boundaries are all known, the DEM and EINO here adopt a fully connected neural network comprising 3 hidden layers, each containing 50 neurons. There are 12 elements with 4 Gaussian points in each element. In DEM, the neural network prediction of u is multiplied by the input coordinate x to enforce the fixed boundary condition on the left. Quadratic serendipity elements are deployed for regularization in EINO. Dirichlet boundary conditions are set by assigning corresponding values to boundary nodes. We use the combined optimizer with Adam using step size 1×10^{-3} for 5,000 steps first and then LBFGS with step size 1.0.

Fig. 13b-d juxtapose DEM with EINO about loss, relative error, and L_2 norm versus training iterations. DEM diverges from the very beginning with the loss far below the true energy loss, exploding error, and surging complexity of neural network representation. As a striking contrast, EINO maintains robust loss, accuracy, and L_2 norm of neural network parameters, even after the number of training iterations reaches 17,000. To further validate EINO, we run the test 10 times, finally, the mean relative error is 0.27%, and the standard deviation is 0.030%. Fig. 13e–f depict the contour of u from DEM and EINO, respectively. The DEM diverges with a complex neural network representation featuring abrupt transitions and discontinuities of field variables. EINO, benefiting from the finite-element-informed regularization, maintains high accuracy during the later stage of training. The robust and accurate performance supports the proposed failure mechanism, namely that inaccurate numerical integration leads to the failure of forward analysis, as discussed in Section 3.1.2. Discussions in Section 5.2 are also validated that finite-element-informed regularization guarantees accurate integration and stops training after reaching the reference energy.

5.4. *Numerical experiments for solving failure in inverse analysis*

Previous section validates EINO in solving the failure problem of forward analysis. This section further checks EINO for solving inverse analysis. The first stage is to bake the physics information from energy minimization into the neural network. The second stage performs inverse analysis by minimizing data loss in accordance with the physics-informed neural network. Configuration of the cantilever beam test in Section 3.1.1 with 168 quadratic serendipity elements is deployed. The trunk network consists of four hidden layers, each containing 50 neurons. The branch network comprises three hidden layers, with 50 neurons in each layer.

5.4.1. *Offline training stage*

In the training stage, there are 16 different sets of input λ, which are obtained by combining four different moduli $[5, 10, 15, 20]\times10^6$Pa with four different Neumann boundary conditions $[-500, -1000, -1500, -2000]$Pa. We use the combined optimizer with Adam using step size

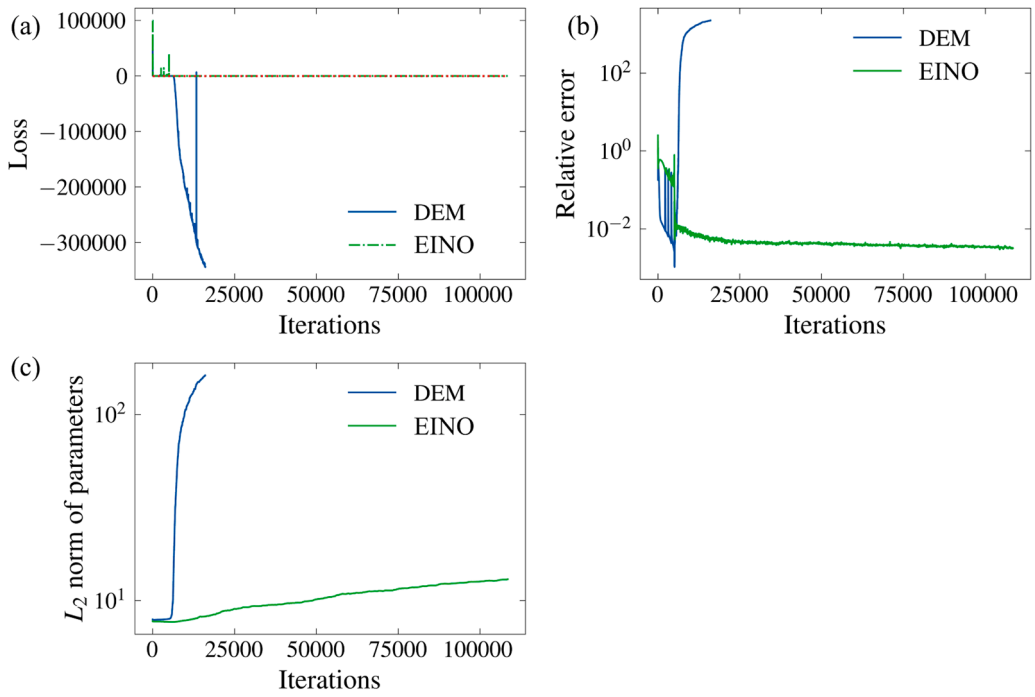

Fig. 12. Results of the cantilever beam test of DEM and EINO (DEM diverges while EINO does not): Loss, relative error, and parameter norm. (a) Losses versus training iterations (Red dotted line denoting the real target energy loss; DEM diverges while EINO converges); (b) Relative errors versus training iterations (DEM diverges while EINO converges); (c) L_2 norm of neural network parameters versus training iterations (DEM diverges while EINO converges).

1×10^{-3} for 10,000 steps first and then LBFGS with step size 1.0 for 40,000 steps. The final relative error of all predictions is 0.33%. Only energy loss. No labeled data is needed during offline training.

5.4.2. *Inverse analysis at online stage*

During online stage, the displacements u_x and u_y are used as labeled data. We adopt the combined optimizer with Adam using step size 1×10^{-2} for 200 steps first and then LBFGS with step size 1.0. First, two tests for inverse analysis of unknown modulus and boundary force are performed. Fig. 14 demonstrates the results of these two tests. Fig. 14a compares the inverse analysis for modulus. EINO converges quickly, while DEM diverges from the target value. Fig. 14c compares the final displacements of DEM and EINO. EINO reconstructs the displacements accurately, but DEM fails with undefined values. Fig. 14b compares the inverse analysis for Neumann boundary condition. DEM seems not aware of the correct boundary force and decreases towards negative infinity. EINO still converges to the correct solution in 200 iterations. Fig. 14d compares the final displacements of DEM and EINO from the inverse analysis of boundary force. EINO achieves satisfactory accuracy and solves the failure of inverse analysis in DEM.

To further check EINO for inverse analysis in different settings, different moduli [5,10, 15, 20]$\times10^6$Pa and different Neumann boundary conditions [−500, −1000, −1500, −2000]Pa are tested by different combinations. Relative errors of predicted modulus and boundary force from inverse analysis in different settings of EINO are presented in Fig. 15ab. The results further validate EINO's ability of inverse analysis in a broader scope.

Finally, we examine the performance of the inverse analysis in scenarios involving unseen parameters. Fig. 15c presents the relative error of modulus versus different modulus values. When the modulus is between (5×10^6Pa, 2×10^7Pa), the inverse analysis exhibits high accuracy with relative error always smaller than 2%. If the modulus is not within the range of training settings. The inverse analysis still yields satisfactory accuracy with relative error smaller than 10% when the modulus is between (2×10^7Pa, 4×10^7Pa). Fig. 15d presents the relative error of Neumann boundary force versus different force values. When the force is between (−2000Pa, −500Pa), the inverse analysis exihibits high accuracy with relative error always smaller than 1%. If the modulus is not within the range of training settings. The inverse analysis still yields satisfactory accuracy with relative error smaller than 10% when the modulus is between (−4000Pa, −2000Pa). In summary, EINO effectively addresses the inverse analysis problem in cases where DEM consistently fails. Furthermore, EINO demonstrates satisfactory accuracy when applied to previously unseen data.

5.4.3. *Inverse uncertainty quantification*

The previous section has proved the effectiveness of EINO in inverse analysis. This section further leverages its performance in inverse uncertainty quantification. With the same settings as Section 5.4.2, Gaussian noises with different relative magnitudes M_n are added into the clean data following $u_\mathrm{n}=u+M_\mathrm{n}\sigma(u)r$, where u_n is the labeled data with noise, $\sigma(u)$ denotes standard deviation of the field variable u, and r represents samples from standard Gaussian distribution. The target Modulus is 1×10^7Pa, and the target Neumann boundary loading is -1×10^3Pa. Fig. 16a and Fig. 16b present the corrupted labeled data with 200% Gaussian noise. 10%, 20%, 50%, 100%, and 200% Gaussian noises are tested 10 times, respectively. The mean and standard deviation of Modulus and Neumann boundary loadings from inverse analysis of EINO are delineated in the curves of Fig. 16c and Fig. 16d. Larger noise induces larger error and deviation. EINO shows remarkable robustness in recovering accurate parameters from highly noisy data. EINO preserves physics information without any labeled data and can serve as reliable surrogates for inverse uncertainty quantification.

5.5. *Three-dimensional investigations on unstructured grids*

This section further validates EINO in 3D non-rectangular domains. As shown in Fig. 17a, the pipe elbow, modeled as a quarter section of a three-dimensional torus, is a ubiquitous connector in engineering. Geometrically, it comprises a curved tubular structure with a constant cross-sectional radius revolving around an axis. The left boundary is Dirichlet boundary with u=1. The flux at the bottom boundary is set as

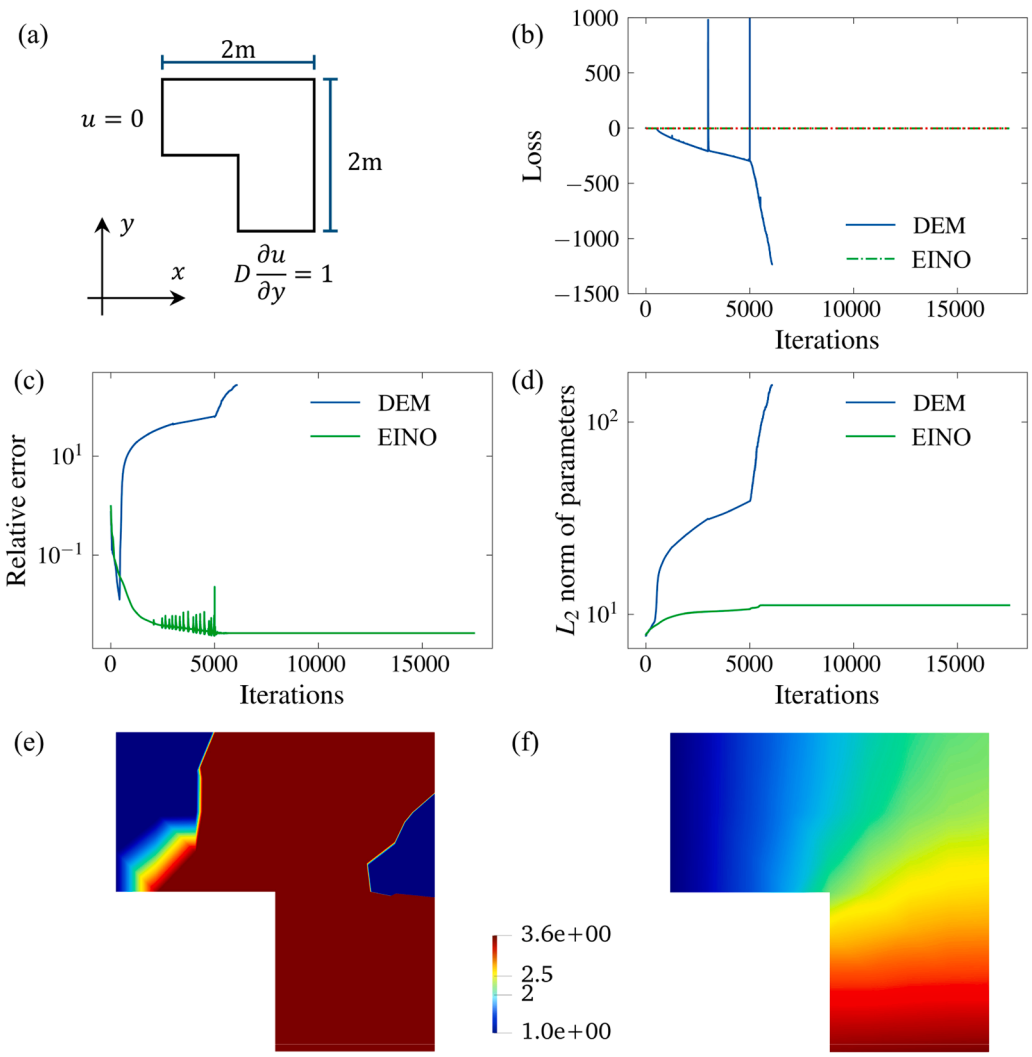

Fig. 13. Comparison between DEM and EINO on the forward analysis of diffusion problem: DEM fails while EINO succeeds. (a) Configuration of the diffusion domain; (b) Loss versus training iterations (Red dotted line denoting the real target energy loss; DEM diverges while EINO converges); (c) Relative error versus training iterations (DEM yields large error while EINO converges to zero error); (d) L_2 norm of neural network parameters versus training iterations (DEM overfits while EINO does not); (e) Contour of u from DEM (diverged solution); (f) Contour of u from EINO (relative error = 0.26%).

$D\frac{\partial u}{\partial z} = 1$. D is set as 1. Fig. 17b presents the unstructured grid used in FEM and EINO, which contains 1,779 tetrahedral elements with second-order Lagrange basis functions. Each element is assigned with 4 Gaussian points.

The DEM here adopts a fully connected neural network comprising 3 hidden layers, each containing 50 neurons. The neural network predictions of u are multiplied by input coordinates $(x+2)$ to enforce the Dirichlet boundary condition on the left. DEM and EINO use the same network and numerical integration. Both DEM and EINO are trained 100 steps with Adam (step size 1×10^{-3}) first and then LBFGS (step size 1.0). Results on the surface and selected slices are demonstrated in Fig. 17. Fig. 17a lists DEM and EINO when the relative error is 1%. After that, the results at the 492nd iteration are compared in Fig. 17b. The relative error of DEM rebounds after reaching 1%. On the contrary, EINO is robust in the training, with high accuracy of 0.09% relative error. Subsequently, Fig. 17c juxtaposes results at the 1,428th iteration. DEM training is undermined by a drastically large relative error of 89,300%, and only the Dirichlet boundary is accurate due to hard enforcement. Conversely, EINO accuracy further enhances, and the relative error is only 0.02%. DEM training stops and EINO continues to the 6,300th iteration as shown in Fig. 17d. EINO stops training with a relative error of 0.009%. These comparisons further validate EINO in 3D non-rectangular geometry. The accuracy and robustness of EINO is as expected since it is rigorously supported by formal analysis.

6. Discussions

The preceding numerical experiments have validated the effectiveness of EINO in addressing the forward divergence and inverse collapse issues faced by DEM. To further enhance the understanding of EINO's strengths and limitations, this chapter addresses the following questions: (1) Although the computed energy loss is lower than the true minimum, what is the real energy loss given that known physical laws cannot be violated? (2) The previous examples involved linear problems. Does EINO remain equally applicable to nonlinear problems? (3) Is EINO the only solution for DEM problems? What are the limitations of EINO?

6.1. What is the real energy loss?

The diverged DEM, as shown in previous sections, exhibits unphysical energy loss that is far below the true minimum energy. To better support the findings that forward divergence attributes the loss collapse to numerical integration, we compare the energy loss from the very sparse mesh with 20 elements and dense mesh with 2,000 elements in

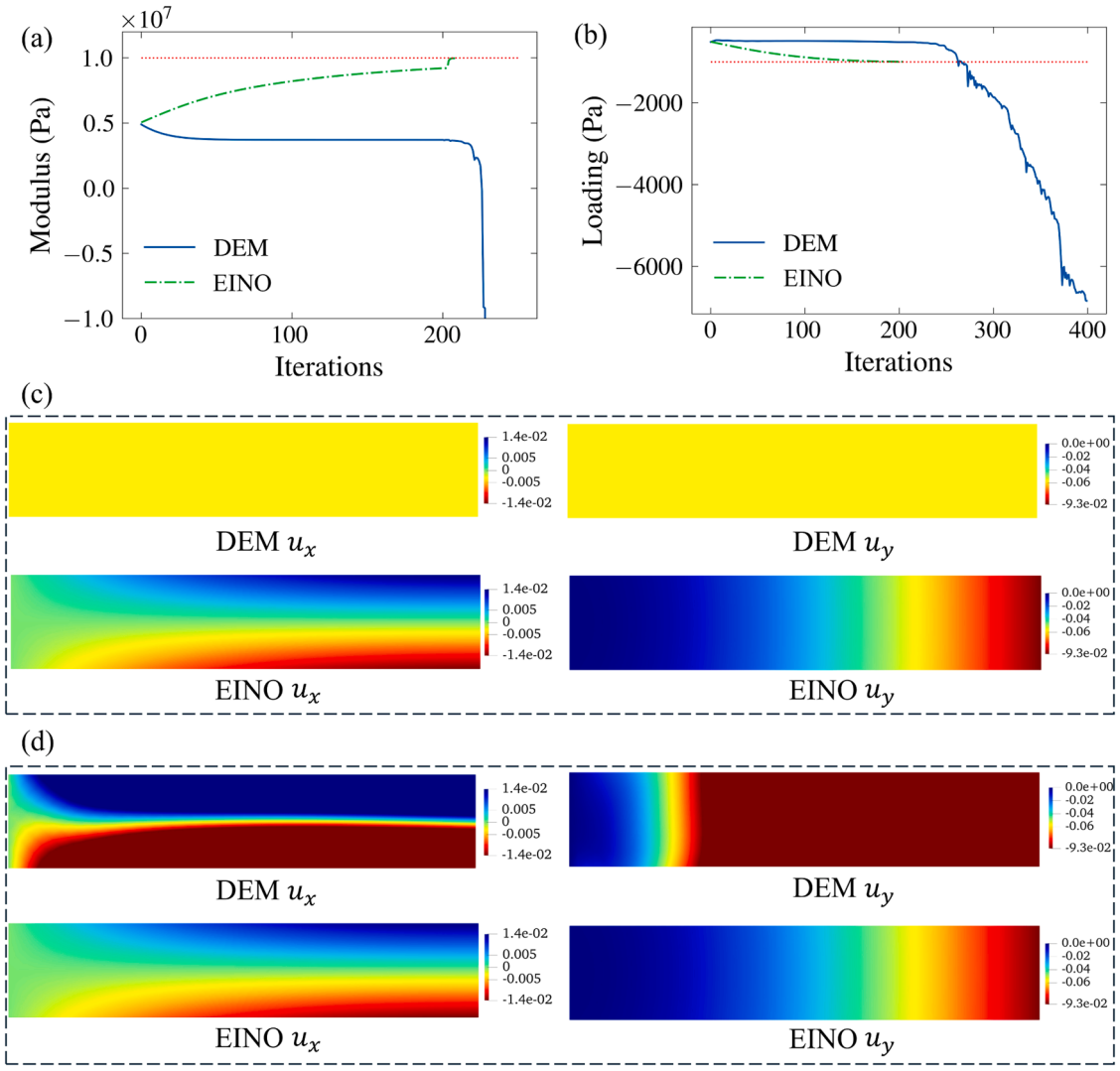

Fig. 14. Comparison between DEM and EINO on the inverse analysis of cantilever beam: DEM fails while EINO succeeds. (a) Modulus versus number of training iterations (Red dotted line denoting the real target Modulus. DEM diverges while EINO converges to the correct Modulus); (b) Neumann boundary loading versus number of training iterations (Red dotted line denoting the real target Neumann boundary. DEM diverges while EINO converges to the correct Neumann boundary loading); (c) Contour of displacements after the inverse analysis of unknown modulus (Relative error of DEM is infinite. Relative error of EINO is 0.34%); (d) Contour of displacements after the inverse analysis of unknown loading at the Neumann boundary (Relative error of DEM is 1040%. Relative error of EINO is 0.37%).

Fig. 18a. The same cantilever beam case in Section 5.3.1 is reproduced. Integration in DEM is computed using the coarse mesh, and the reference real energy on dense mesh from FEM using predictions from the same DEM neural network is taken for comparison. Fig. 18b gives the DEM energy loss and real energy loss. The difference between these two curves indicates errors of numerical integration. From the beginning, DEM energy loss on coarse mesh matches well with the real energy loss on dense mesh. At about the 5,500th step, DEM loss diverges far below the physical constraint of the minimum potential energy. In Fig. 18c, the relative error increases significantly due to the integration error as indicated in Fig. 18b. It also shows that real energy will not violate the principle of minimum potential energy. The discrepancy between DEM energy and real energy supports the claim that such forward divergence is caused by inaccurate integration. Diverged results of those in Section 3.1.3 also show that a denser mesh can contribute to later divergence. In comparison, the training of EINO in Fig. 18d does not diverge after 50,000 iterations even on the very coarse mesh, which is well corroborated by the proof in Section 4.2.

6.2. *What about nonlinear problems?*

To further corroborate EINO in the case of nonlinear problems on a coarse mesh, the same geometry and boundaries of the cantilever beam in Section 5.3.1 are adopted with nonlinear hyperelastic constitutive model as introduced in Eqs. (6)-(9). The Young's modulus and Poisson's ratio are 1×10^7Pa and 0.3, respectively. To validate EINO's robustness, a very coarse mesh with 20 elements in Fig. 19a is used for discretization. Fig. 19b demonstrates the robust convergence of EINO and the forward divergence of DEM. EINO performs robustly even for 10,000 steps, and the robustness for more iterations can still be guaranteed due to the proof in Section 4.2. Relative errors of DEM and EINO in Fig. 19c also support EINO's superior performance. Similarly, Fig. 19d presents norms of the parameters for both DEM and EINO. EINO's sound performance corroborates the novel finite-element-informed regularization. Fig. 19e and Fig. 19f juxtapose the diverged DEM solution and EINO's accurate results. This section proves EINO's ability to solve nonlinear problems. Section 4.2 demonstrates that EINO's strong performance is not limited to specific problem types or network architecture.

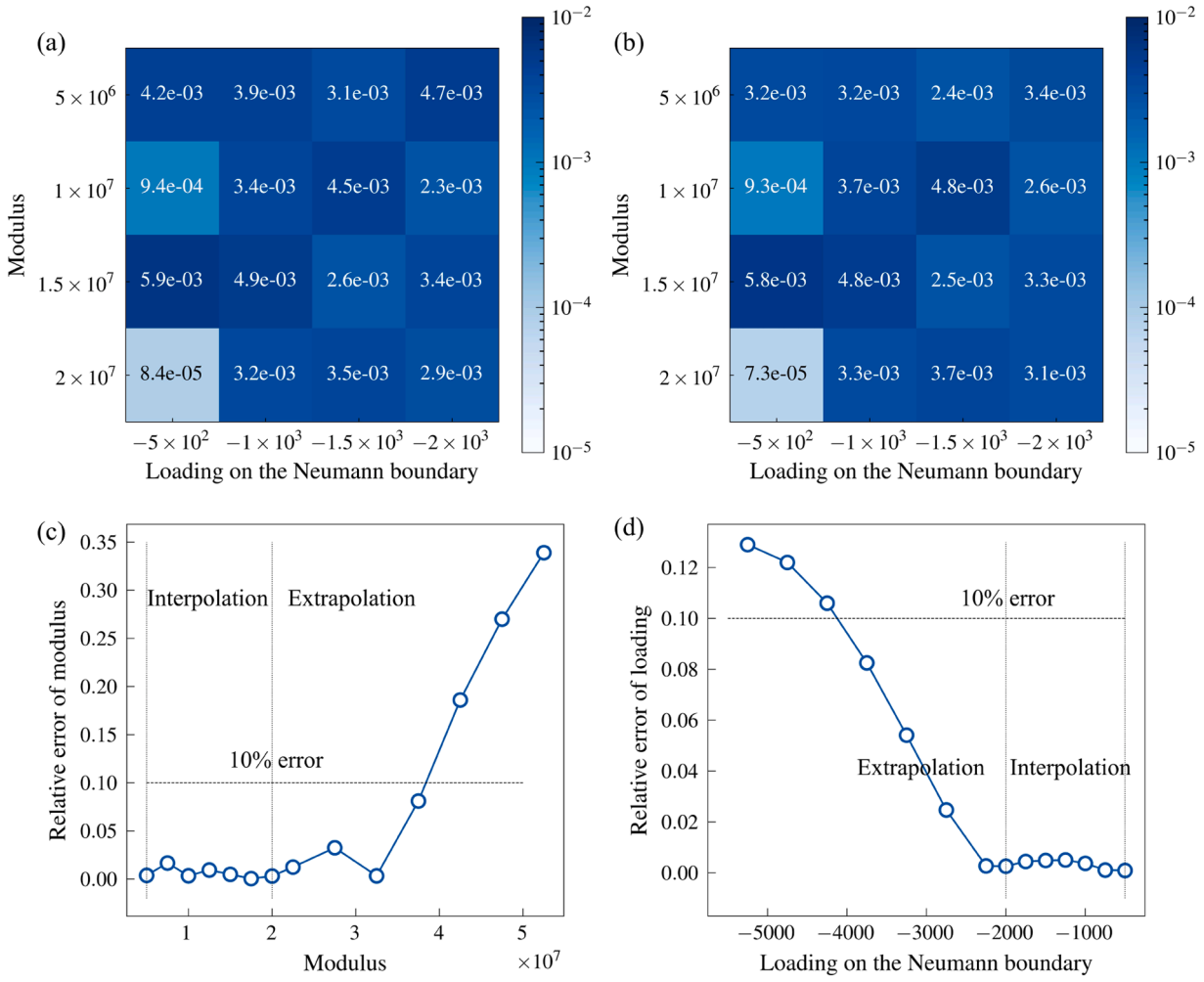

Fig. 15. Inverse analysis using EINO in different settings of unknown Modulus and Neumann boundary loadings for the cantilever beam problem. (a) Heatmap of relative errors of Modulus; (b) Heatmap of relative errors of Neumann boundary forces; (c) Relative errors of Modulus (Neumann boundary force is set as −1000Pa. Modulus is tested for extrapolation scenarios); (d) Relative errors of Neumann boundary loadings (Modulus is set as 1×10^{-7}Pa. Neumann boundary loading is tested for extrapolation scenarios).

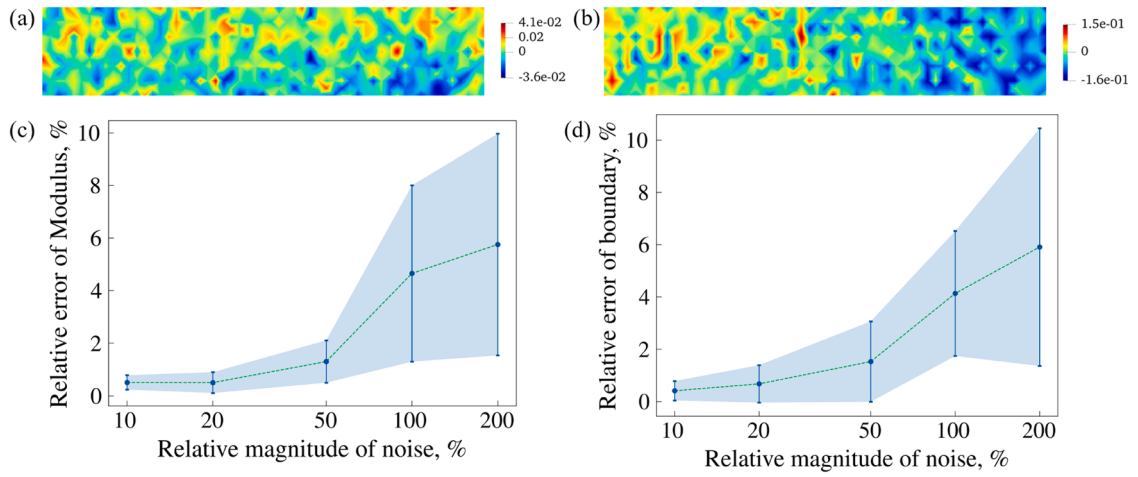

Fig. 16. Inverse analysis of Modulus and Neumann boundary loadings using noisy data. (a) Displacements data along the *x* axis with 200% Gaussian noise; (b) Displacements data along the *y* axis with 200% Gaussian noise; (c) Relative errors of Modulus under different magnitudes of noise; (d) Relative errors of Neumann boundary loadings under different magnitudes of noise (error bars represent ±1 standard deviation).

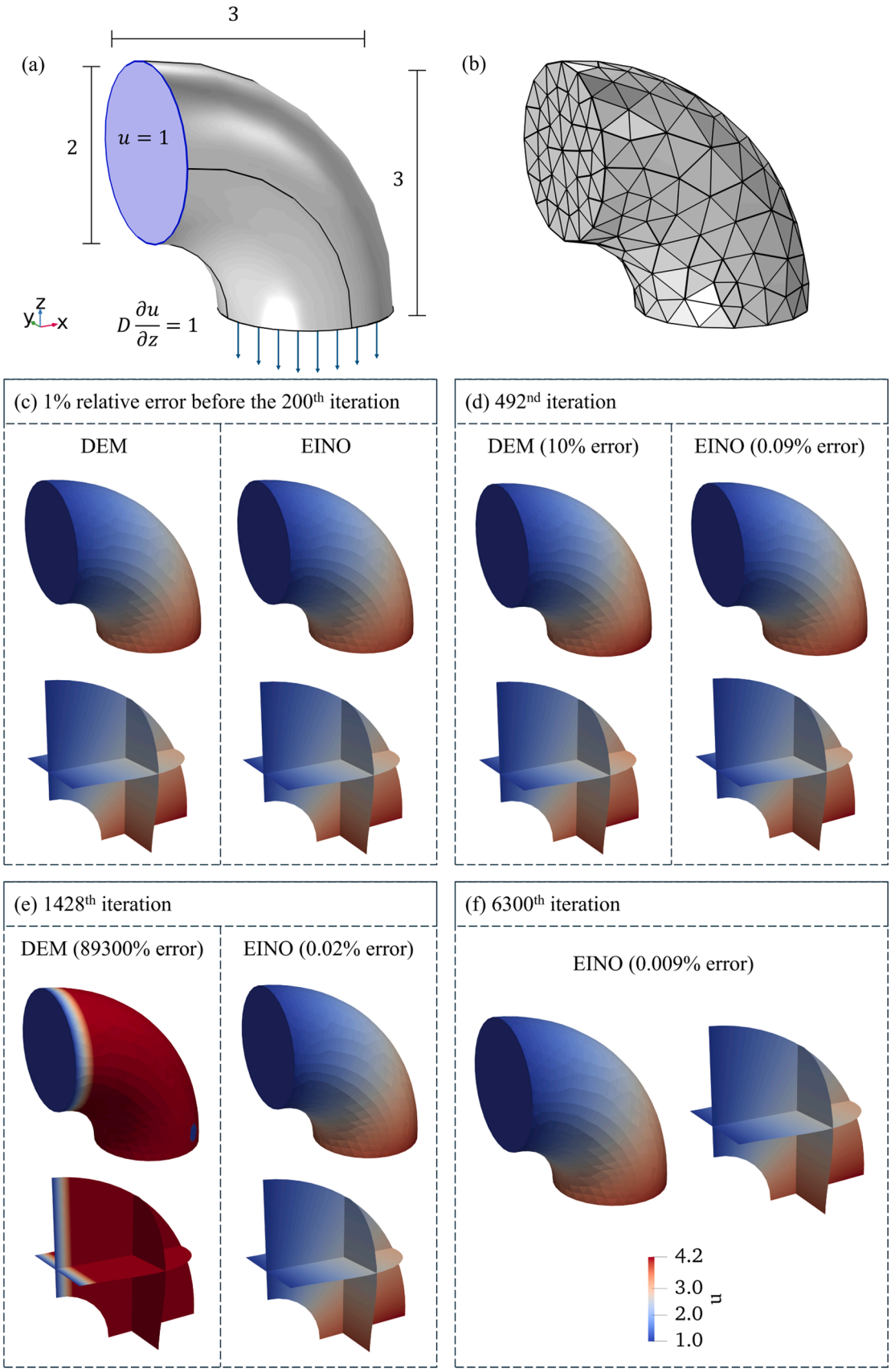

Fig. 17. Comparisons between DEM and EINO in the 3D diffusion problem: DEM fails while EINO succeeds. (a) Configuration of the 3D diffusion problem; (b) FEM mesh for numerical integration; (c) DEM and EINO results when the relative error is 1%; (d) DEM and EINO results at the 492nd iteration. DEM error is large (10% error) while EINO is still accurate (0.09% error); (e) DEM and EINO results at the 1428th iteration. DEM totally fails (89,300% error) while EINO is still accurate (0.02% error); (f) EINO results at the 6,300nd iteration.

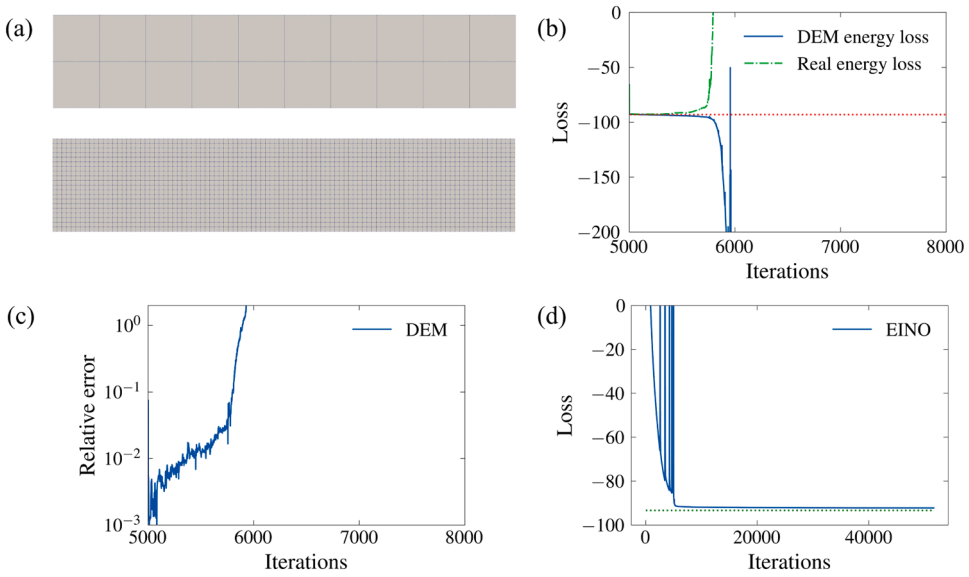

Fig. 18. Exploration of diverged unphysical energy loss of DEM at coarse mesh, its actual energy at dense mesh, and results of EINO at coarse mesh. (a) The coarse mesh with 20 elements and the dense mesh with 2,000 elements; (b) Unphysical DEM energy loss at the coarse mesh and the corresponding real energy loss. The real energy loss is computed using the dense mesh (red dotted line denoting the target energy loss); (c) Relative error of DEM at the coarse mesh; (d) The energy loss of EINO on the coarse mesh (green dotted line denoting the real target energy loss).

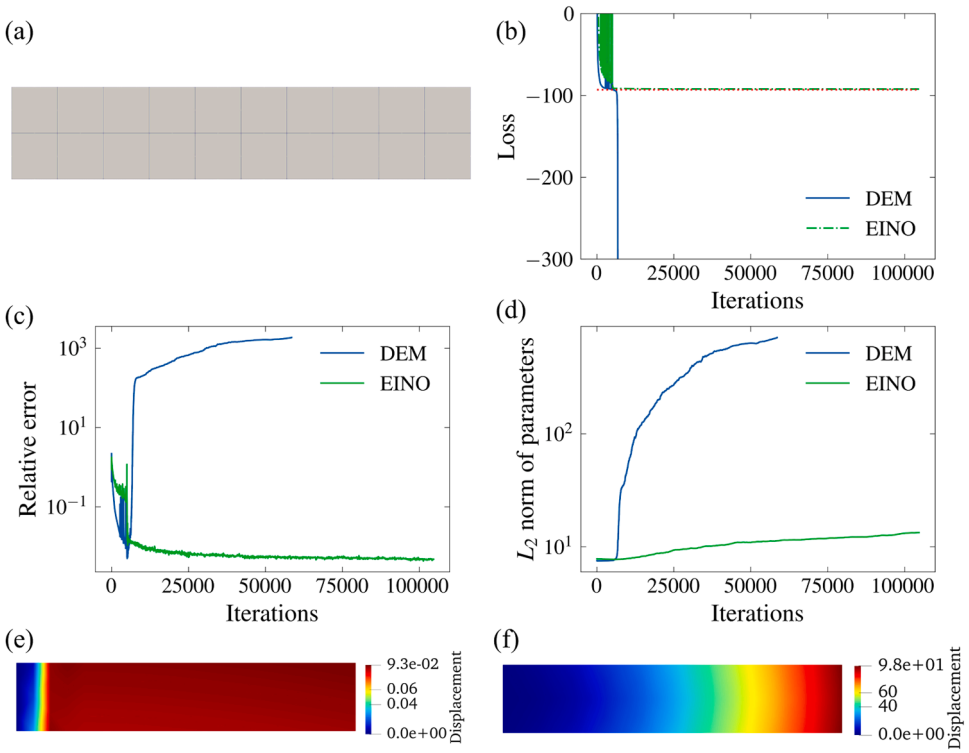

Fig. 19. Solving nonlinear hyperelastic cantilever beam using DEM and EINO. (a) The coarse mesh used for energy integration; (b) Loss curve of DEM and EINO (Red dotted line denoting the real target energy loss); (c) Relative errors of DEM and EINO; (d) L_2 norm of neural network parameters from DEM and EINO; (e) DEM displacement solution; (f) Displacement results of EINO.

6.3. Limitations and alternatives

It should be noted that the proposed solution is not the only viable one. Some other alternatives might be effective when leveraging more formulations like constrained optimization with priors/regularizes on parameters or mixed complementary energy forms. For example, Thombre et al. (2025) [92] used the energy gradient instead of energy to impose physics constraints for inverse analysis. However, such formulation is somewhat more computational demanding than minimizing energy since higher-order derivatives are entailed.

On the other hand, as shown in Section 3.1.3, Section 3.1.4, Section 5.1, and Section 5.2, increasing mesh density, adding Gaussian points, shallower or narrower neural networks, and adding L_2 regularizations can mitigate the forward divergence. Early stopping can also circumvent

this problem to some extent. Thus, using adaptive quadrature and residual-based integration refinement will certainly enhance DEM's performance. The finite-element-informed regularization, though proved effective even on very coarse meshes without additional requirements on neural network architecture, can constrain the solution's representation power to that of the underlying finite element basis. However, such limitations are whitebox, and we can adjust the representation complexity as we need for a balance between accuracy and efficiency. Moreover, some other challenges that are not unique to DEM still need to be addressed, like the spectral bias and difficulties in long-range extrapolation.

7. Conclusions

While the Deep Energy Method (DEM) demonstrates superior performance over Physics-Informed Neural Networks (PINNs) in prior studies, it suffers from two critical failure modes: (1) divergence in forward analysis due to inaccurate numerical integration, and (2) collapse in inverse analysis because the minimization problem in variational formulation is undefined for unknown material parameters or Neumann boundary conditions. This study systematically diagnoses these failure phenomena and establishes mechanistic remedies.

Key findings from the study reveal that forward divergence stems from integration inaccuracies when evaluating complex neural network outputs, which explains why strategies like mesh refinement, architectural simplification, L_2 regularization, or increased integration points, all improving integration fidelity, delay but do not entirely prevent failure. It also elucidates why the calculated loss substantially deviates below the true energy loss, a physically implausible phenomenon that violates the minimum potential energy principle. On the other hand, we revisit the rigorous variational formulation that defines a minimizer only with respect to target field variables, which supports efficient forward training but renders inverse analysis entirely infeasible. The proposed Energy-Informed Neural Operator Network (EINO) helps to overcome these limitations with the novel design of two features: (1) Finite-element-informed regularization constrains outputs within intended solution space, ensuring numerically stable integration; and (2) a two-stage operator-learning strategy of offline learning based on embedded energy principles via DeepONet and online learning to solve inverse problems by minimizing data loss.

The robustness of EINO is further validated through comprehensive benchmarks. In forward analysis, it maintains sustained accuracy via precise numerical integration, effectively preventing the divergence observed in DEM. For inverse problems, EINO reliably infers material parameters and boundary conditions in scenarios where DEM fails catastrophically, and demonstrates robust performance in uncertainty quantification even under significant noise. By preserving the efficiency of DEM while resolving its failure modes, EINO establishes a more robust and accurate framework for physics-constrained learning. Future work will target its extension to multiphysics problems and integration with advanced neural architectures.

Acknowledgments

This work is financially supported by National Natural Science Foundation of China (Key Project #52439001), Research Grants Council of Hong Kong (GRF #16212724, GRF #16206322, CRF C7085-24 G, RIF R6008-24, TRS #T22-607/24-N, TRS #T22-606/23-R).

CRediT authorship contribution statement

Xi Wang: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Jidong Zhao:** Writing – review & editing, Funding acquisition, Conceptualization. **Zhen-Yu Yin:** Writing – review & editing, Supervision, Software, Conceptualization. **Xiaoying Zhuang:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

For a reference quadrilateral element in Fig. 20a, the nodes $\overline{\boldsymbol{x}}_i$ and corresponding nodal basis functions $\bar{l}_i$ are [95]

$$\begin{cases} \overline{\boldsymbol{x}}_0 = (0,0), \overline{\boldsymbol{x}}_1 = (1,0), \overline{\boldsymbol{x}}_2 = (0,1), \overline{\boldsymbol{x}}_3 = (1,1), \\ \overline{\boldsymbol{x}}_4 = \left(\frac{1}{2},0\right), \overline{\boldsymbol{x}}_5 = \left(0,\frac{1}{2}\right), \overline{\boldsymbol{x}}_6 = \left(1,\frac{1}{2}\right), \overline{\boldsymbol{x}}_7 = \left(\frac{1}{2},1\right), \\ \bar{l}_0 = -3\overline{x}^2\overline{y} + 3\overline{x}^2 - 3\overline{x}\overline{y}^2 + 7\overline{x}\overline{y} - 4\overline{x} + 3\overline{y}^2 - 4\overline{y} + 1, \\ \bar{l}_1 = \overline{x}\left(-3\overline{x}\overline{y} + 3\overline{x} + 3\overline{y}^2 - \overline{y} - 2\right), \\ \bar{l}_2 = \overline{y}\left(3\overline{x}^2 - 3\overline{x}\overline{y} - \overline{x} + 3\overline{y} - 2\right), \\ \bar{l}_3 = \overline{x}\overline{y}(3\overline{x} + 3\overline{y} - 5), \\ \bar{l}_4 = 6\overline{x}(\overline{x}\overline{y} - \overline{x} - \overline{y} + 1),\ \bar{l}_5 = 6\overline{y}(\overline{x}\overline{y} - \overline{x} - \overline{y} + 1), \\ \bar{l}_6 = 6\overline{x}\overline{y}(1 - \overline{y}), \bar{l}_7 = 6\overline{x}\overline{y}(1 - \overline{x}). \end{cases} \tag{22}$$

For a 3D reference Lagrange element in Fig. 20 Fig. 7b, the nodes $\overline{\boldsymbol{x}}_i$ and corresponding nodal basis functions $\bar{l}_i$ are

$$
\begin{cases}
\overline{\boldsymbol{x}}_0 = (0,0,0), \overline{\boldsymbol{x}}_1 = (1,0,0), \overline{\boldsymbol{x}}_2 = (0,1,0), \overline{\boldsymbol{x}}_3 = (0,0,1), \overline{\boldsymbol{x}}_4 = \left(0, \frac{1}{2}, \frac{1}{2}\right), \\
\overline{\boldsymbol{x}}_5 = \left(\frac{1}{2}, 0, \frac{1}{2}\right), \overline{\boldsymbol{x}}_6 = \left(\frac{1}{2}, \frac{1}{2}, 0\right), \overline{\boldsymbol{x}}_7 = \left(0, 0, \frac{1}{2}\right), \overline{\boldsymbol{x}}_8 = \left(0, \frac{1}{2}, 0\right), \overline{\boldsymbol{x}}_9 = \left(\frac{1}{2}, 0, 0\right), \\
\bar{l}_0 = 2\overline{x}^2 + 4\overline{xy} + 4\overline{xz} - 3\overline{x} + 2\overline{y}^2 + 4\overline{yz} - 3\overline{y} + 2\overline{z}^2 - 3\overline{z} + 1, \\
\bar{l}_1 = \overline{x}(2\overline{x} - 1), l_2 = \overline{y}(2\overline{y} - 1), \bar{l}_3 = \overline{z}(2\overline{z} - 1), \\
\bar{l}_4 = 4\overline{yz}, \bar{l}_5 = 4\overline{xz}, \bar{l}_6 = 4\overline{xy}, \\
\bar{l}_7 = 4\overline{z}(-\overline{x} - \overline{y} - \overline{z} + 1), \bar{l}_8 = 4\overline{y}(-\overline{x} - \overline{y} - \overline{z} + 1), \bar{l}_9 = 4\overline{x}(-\overline{x} - \overline{y} - \overline{z} + 1).
\end{cases} \tag{23}
$$

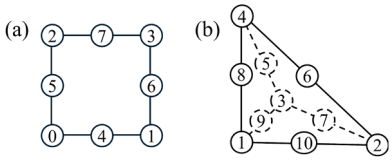

Fig. 20. Elements for finite-element-informed regularization and numerical integration in EINO. (a) The second-order serendipity quadrilateral element with 8 nodes; (b) The second-order Lagrange tetrahedral element with 11 nodes.

Data availability

The relevant software is under development, and it will be released at [ht tps://github.com/xiwang0706/Projects-of-xiwang/tree/main] in the future.

References

[1] Yang C, Zhu F, Zhao J. A multi-horizon fully coupled thermo-mechanical peridynamics. J Mech Phys Solids 2024:105758. https://doi.org/10.1016/j.jmps.2024.105758.

[2] Wang L, Vuik C, Hajibeygi H. A stabilized mixed-FE scheme for frictional contact and shear failure analyses in deformable fractured media. Eng Fract Mech 2022;267:108427.

[3] Wang L, Yin ZY, Chen W. Peridynamic modelling of dynamic damage and fragmentation of cracked solids during impact contact. Eng Fract Mech 2024;295:109784.

[4] Yang C, Zhu F, Zhao J. Coupled total- and semi-lagrangian peridynamics for modelling fluid-driven fracturing in solids. Comput Meth Appl Mech Eng 2024;419:116580. https://doi.org/10.1016/j.cma.2023.116580.

[5] Yu J, Zhao J, Liang W, Zhao S. A semi-implicit material point method for coupled thermo-hydro-mechanical simulation of saturated porous media in large deformation. Comput Meth Appl Mech Eng 2024;418:116462. https://doi.org/10.1016/j.cma.2023.116462.

[6] Yu J, Zhao J, Zhao S, Liang W. Thermo-hydro-mechanical coupled material point method for modeling freezing and thawing of porous media. Int J Numer Anal Methods Geomech 2024;48:3308–49. https://doi.org/10.1002/nag.3794.

[7] Zhao J, Zhao S, Luding S. The role of particle shape in computational modelling of granular matter. Nat Rev Phys 2023:1–21. https://doi.org/10.1038/s42254-023-00617-9.

[8] Alkin B., Kronlachner T., Papa S., Pirker S., Lichtenegger T., Brandstetter J. NeuralDEM – real-time simulation of industrial particulate flows 2025. https://doi.org/10.48550/arXiv.2411.09678.

[9] Zhao S, Zhao J, Liang W, Niu F. Multiscale modeling of coupled thermo-mechanical behavior of granular media in large deformation and flow. Comput Geotech 2022;149:104855. https://doi.org/10.1016/j.compgeo.2022.104855.

[10] Lai Z, Zhao S, Zhao J, Huang L. Signed distance field framework for unified DEM modeling of granular media with arbitrary particle shapes. Comput Mech 2022;70:763–83. https://doi.org/10.1007/s00466-022-02220-8.

[11] Kong Y, Guan M, Li X, Zhao J, Yan H. How flexible, slit and rigid barriers mitigate two-phase geophysical mass flows: a numerical appraisal. J Geophys Res: Earth Surf 2022;127. https://doi.org/10.1029/2021JF006587. e2021JF006587.

[12] Wang X, Wu W, Zhu H, Lin J-S, Zhang H. Contact detection between polygonal blocks based on a novel multi-cover system for discontinuous deformation analysis. Comput Geotech 2019;111:56–65. https://doi.org/10.1016/j.compgeo.2019.03.004.

[13] Wang X, Wu W, Zhu H, Zhang H, Lin J-S, Bobet A. A global direct search method for high-fidelity contact detection between arbitrarily shaped three-dimensional convex polyhedral blocks. Comput Geotech 2022;150:104891. https://doi.org/10.1016/j.compgeo.2022.104891.

[14] Wang X, Wu W, Zhu H, Zhang H. Three-dimensional discontinuous deformation analysis derived from the virtual work principle with a simplex integral on the boundary. Comput Geotech 2022;146:104710. https://doi.org/10.1016/j.compgeo.2022.104710.

[15] Wu W, Wang X, Zhu H, Shou K-J, Lin J-S, Zhang H. Improvements in DDA program for rockslides with local in-circle contact method and modified open-close iteration. Eng Geol 2020;265:105433. https://doi.org/10.1016/j.enggeo.2019.105433.

[16] Wang X, Wu W, Zhu H, Zhang H, Lin J-S. The last entrance plane method for contact indeterminacy between convex polyhedral blocks. Comput Geotech 2020;117:103283. https://doi.org/10.1016/j.compgeo.2019.103283.

[17] Wang X, Wu W, Zhu H, Liu F, Zhang H, Lin J-S. Three-dimensional discontinuous deformation analysis with explicit contact formulation and block-wise multicore CPU acceleration. Comput Geotech 2021;139:104410. https://doi.org/10.1016/j.compgeo.2021.104410.

[18] Wu W, Wang X, Zhu H, Zhang H, Lin J-S, Bobet A. Improved friction force calculation with an augmented open-close iteration formulation in discontinuous deformation analysis. Comput Geotech 2021;130:103932. https://doi.org/10.1016/j.compgeo.2020.103932.

[19] Jin YF, Yin ZY, Zhou XW. Two-phase two-layer SNS-PFEM for hydromechanical: geotechnical large deformation problems. Comput Methods Appl Mech Eng 2024;418:116542.

[20] Jin YF, Yin ZY, Zhou XW, Liu FT. A stable node-based smoothed PFEM for :solving geotechnical large deformation 2D problems. Comput Methods Appl Mech Eng 2021;387:114179.

[21] Jin YF, Yin ZY. Two-phase PFEM with stable nodal integration for large deformation hydromechanical coupled geotechnical problems. Comput Methods Appl Mech Eng 2022;392:114660.

[22] Li XF, Li HB, Zhao J. The role of transgranular capability in grain-based modelling of crystalline rocks. Comput. Geotech. 2019;110:161–83.

[23] Wei M, Dai F, Liu Y, Jiang R. A fracture model for assessing tensile mode crack growth resistance of rocks. J Rock Mech Geotech Eng 2023;15(2):395–411.

[24] Wei M, Dai F, Liu Y, Li A, Yan Z. Influences of loading method and notch type on rock fracture toughness measurements: from the perspectives of T-stress and fracture process zone. Rock Mech Rock Eng 2021;54(9):4965–86.

[25] Ho J., Jain A., Abbeel P. Denoising diffusion probabilistic models 2020. https://doi.org/10.48550/arXiv.2006.11239.

[26] Hoogeboom E., Gritsenko A.A., Bastings J., Poole B., Berg R van den, Salimans T. Autoregressive diffusion models 2022. https://doi.org/10.48550/arXiv.2110.02037.

[27] Phung H., Dao Q., Dao T., Phan H., Metaxas D., Tran A. DiMSUM: Diffusion Mamba – A scalable and unified spatial-frequency method for image generation 2025. https://doi.org/10.48550/arXiv.2411.04168.

[28] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Curran Associates Inc.; 2017. p. 6000–10.

[29] Su L, Zuo X, Li R, Wang X, Zhao H, Huang B. A systematic review for transformer-based long-term series forecasting. Artif Intell Rev 2025;58:80. https://doi.org/10.1007/s10462-024-11044-2.

[30] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045.

[31] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3:422–40. https://doi.org/10.1038/s42254-021-00314-5.

[32] Urbán JF, Stefanou P, Pons JA. Unveiling the optimization process of physics informed neural networks: how accurate and competitive can PINNs be? J Comput Phys 2025;523:113656. https://doi.org/10.1016/j.jcp.2024.113656.

[33] Toscano JD, Oommen V, Varghese AJ, Zou Z, Ahmadi Daryakenari N, Wu C, et al. From PINNs to PIKANs: recent advances in physics-informed machine learning. Mach Learn Comput Sci Eng 2025;1:15. https://doi.org/10.1007/s44379-025-00015-1.

[34] Shi Y, Wei P, Feng K, Feng D-C, Beer M. A survey on machine learning approaches for uncertainty quantification of engineering systems. Mach Learn Comput Sci Eng 2025;1:11. https://doi.org/10.1007/s44379-024-00011-x.

[35] Diao Y, Yang J, Zhang Y, Zhang D, Du Y. Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology. Comput Meth Appl Mech Eng 2023;413:116120. https://doi.org/10.1016/j.cma.2023.116120.

[36] Haghighat E, Raissi M, Moure A, Gomez H, Juanes R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput Meth Appl Mech Eng 2021;379:113741. https://doi.org/10.1016/j.cma.2021.113741.

[37] Wang X, Yin Z-Y, Wu W, Zhu H-H. Differentiable finite element method with Galerkin discretization for fast and accurate inverse analysis of multidimensional heterogeneous engineering structures. Comput Methods Appl Mech Eng 2025;437:117755. https://doi.org/10.1016/j.cma.2025.117755.

[38] Wang X, Yin Z-Y. Interpretable physics-encoded finite element network to handle concentration features and multi-material heterogeneity in hyperelasticity. Comput Meth Appl Mech Eng 2024;431:117268. https://doi.org/10.1016/j.cma.2024.117268.

[39] Wang L, Liu G, Wang G, Zhang KM-PINN. A mesh-based physics-informed neural network for linear elastic problems in solid mechanics. Int J Numer Methods Eng 2024:e7444. https://doi.org/10.1002/nme.7444. n/a.

[40] Rao C, Sun H, Liu Y. Physics-informed deep learning for incompressible laminar flows. Theor Appl Mech Lett 2020;10:207–12. https://doi.org/10.1016/j.taml.2020.01.039.

[41] Orera J, Ramírez J, García-Navarro P, Murillo J. RoePINNs: an integration of advanced CFD solvers with physics-informed Neural Networks and application in arterial flow modeling. Comput Methods Appl Mech Eng 2025;440:117933. https://doi.org/10.1016/j.cma.2025.117933.

[42] Wang S., Sankaran S., Perdikaris P. Respecting causality is all you need for training physics-informed neural networks 2022. https://doi.org/10.48550/arXiv.2203.07404.

[43] Sun X, Cao W, Shan X, Liu Y, Zhang W. A generalized framework for integrating machine learning into computational fluid dynamics. J Comput Sci 2024;82:102404. https://doi.org/10.1016/j.jocs.2024.102404.

[44] Yan 闫畅 C, Xu 许盛峰 S, Sun 孙振旭 Z, Guo 郭迪龙 D, Ju 鞠胜军 S, Huang 黄仁芳 R, et al. Exploring hidden flow structures from sparse data through deep-learning-strengthened proper orthogonal decomposition. Phys Fluids 2023;35:037119. https://doi.org/10.1063/5.0138287.

[45] Ren P, Rao C, Liu Y, Ma Z, Wang Q, Wang J-X, et al. PhySR: physics-informed Deep Super-resolution for spatiotemporal data. J Comput Phys 2023:112438. https://doi.org/10.1016/j.jcp.2023.112438.

[46] Yu X-L, Zhou X-P. A nonlocal energy-informed neural network for isotropic elastic solids with cracks under thermomechanical loads. Int J Numer Methods Eng 2023;124:3935–63. https://doi.org/10.1002/nme.7296.

[47] Shukla K., Zou Z., Chan C.H., Pandey A., Wang Z., Karniadakis G.E. NeuroSEM: a hybrid framework for simulating multiphysics problems by coupling PINNs and spectral elements 2024. https://doi.org/10.48550/arXiv.2407.21217.

[48] Khadijeh M, Cerqueglini V, Kasbergen C, Erkens S, Varveri A. Multistage physics informed neural network for solving coupled multiphysics problems in material degradation and fluid dynamics. Eng Comput 2025. https://doi.org/10.1007/s00366-025-02174-4.

[49] Pan J, Xiao X, Guo L, Feng X. A high resolution physics-informed neural networks for high-dimensional convection–diffusion-reaction equations. Appl Soft Comput 2023:110872. https://doi.org/10.1016/j.asoc.2023.110872.

[50] Harandi A, Moeineddin A, Kaliske M, Reese S, Rezaei S. Mixed formulation of physics-informed neural networks for thermo-mechanically coupled systems and heterogeneous domains. Numer Meth Eng 2023:e7388. https://doi.org/10.1002/nme.7388.

[51] Zhang P, Karapiperis K, Weeger O. *t*-PiNet: a thermodynamics-informed hierarchical learning for discovering constitutive relations of geomaterials. J Mech Phys Solids 2025;197:106049. https://doi.org/10.1016/j.jmps.2025.106049.

[52] Qu T, Zhao J, Guan S, Feng Y. Data-driven multiscale modelling of granular materials via knowledge transfer and sharing. Int J Plast 2023;171:103786. https://doi.org/10.1016/j.ijplas.2023.103786.

[53] Zhongbo Y, Hien PL. Pre-trained transformer model as a surrogate in multiscale computational homogenization framework for elastoplastic composite materials subjected to generic loading paths. Comput Methods Appl Mech Eng 2024;421:116745. https://doi.org/10.1016/j.cma.2024.116745.

[54] Wang J, Ma G, Qu T, Guan S, Zhou W, GPM-PeNN Chang X. A generalized plasticity model-based data-driven constitutive modeling framework using physics-encoded neural network. Comput Methods Appl Mech Eng 2025;436:117694. https://doi.org/10.1016/j.cma.2024.117694.

[55] Haghighat E, Abouali S, Vaziri R. Constitutive model characterization and discovery using physics-informed deep learning. Eng Appl Artif Intell 2023;120:105828. https://doi.org/10.1016/j.engappai.2023.105828.

[56] Lourenço R, Georgieva P, Andrade-Campos A. Data-driven elastoplastic constitutive modelling with physics-informed RNNs using the Virtual Fields Method for indirect training. Comput Methods Appl Mech Eng 2025;440:117935. https://doi.org/10.1016/j.cma.2025.117935.

[57] Li C, Han Z, Li Y, Li M, Wang W, Dou J, Chen G. Physical information-fused deep learning model ensembled with a subregion-specific sampling method for predicting flood dynamics. J Hydrol 2023;620:129465.

[58] Han Z, Long G, Li C, Li Y, Su B, Xu L, Chen G. Enhancing multi-step-ahead prediction of wave propagation with the CAE-LSTM model: a novel deep learning-based approach to flood dynamics. GNHR 2025;16(1):2588708.

[59] E W, Yu B. The Deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. Commun Math Stat 2018;6:1–12. https://doi.org/10.1007/s40304-018-0127-z.

[60] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. Comput Methods Appl Mech Eng 2020;362:112790. https://doi.org/10.1016/j.cma.2019.112790.

[61] Nguyen-Thanh VM, Zhuang X, Rabczuk T. A deep energy method for finite deformation hyperelasticity. Eur J Mech - A/Solids 2020;80:103874. https://doi.org/10.1016/j.euromechsol.2019.103874.

[62] Mojahedin A, Salavati M, Rabczuk T. A deep energy method for functionally graded porous beams. J Zhejiang Univ Sci A 2021;22:492–8. https://doi.org/10.1631/jzus.A2000317.

[63] Abueidda DW, Koric S, RA Al-Rub, Parrott CM, James KA, Sobh NA. A deep learning energy method for hyperelasticity and viscoelasticity. Eur J Mech - A/Solids 2022;95:104639. https://doi.org/10.1016/j.euromechsol.2022.104639.

[64] He J, Abueidda D, Abu Al-Rub R, Koric S, Jasiuk I. A deep learning energy-based method for classical elastoplasticity. Int J Plast 2023;162:103531. https://doi.org/10.1016/j.ijplas.2023.103531.

[65] León O, Rivera V, Vázquez-Patiño A, Ulloa J, Samaniego E. Exploring energy minimization to model strain localization as a strong discontinuity using Physics Informed Neural Networks. Comput Methods Appl Mech Eng 2025;436:117724. https://doi.org/10.1016/j.cma.2024.117724.

[66] Huang Z, Peng L. An improved plate deep energy method for the bending, buckling and free vibration problems of irregular Kirchhoff plates. Eng Struct 2024;301:117235. https://doi.org/10.1016/j.engstruct.2023.117235.

[67] Huang Z-M, Peng L-X. Geometrically nonlinear bending analysis of laminated thin plates based on classical laminated plate theory and deep energy method. Compos Struct 2024;344:118314. https://doi.org/10.1016/j.compstruct.2024.118314.

[68] Bai J, Lin Z, Wang Y, Wen J, Liu Y, Rabczuk T, et al. Energy-based physics-informed neural network for frictionless contact problems under large deformation. Comput Methods Appl Mech Eng 2025;437:117787. https://doi.org/10.1016/j.cma.2025.117787.

[69] Goswami S, Anitescu C, Chakraborty S, Rabczuk T. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor Appl Fract Mech 2020;106:102447. https://doi.org/10.1016/j.tafmec.2019.102447.

[70] Goswami S, Anitescu C, Rabczuk T. Adaptive fourth-order phase field analysis using deep energy minimization. Theor Appl Fract Mech 2020;107:102527. https://doi.org/10.1016/j.tafmec.2020.102527.

[71] Goswami S, Yin M, Yu Y, Karniadakis GE. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. Comput Methods Appl Mech Eng 2022;391:114587. https://doi.org/10.1016/j.cma.2022.114587.

[72] Ghaffari Motlagh Y, Jimack PK, de Borst R. Deep learning phase-field model for brittle fractures. Int J Numer Methods Eng 2023;124:620–38. https://doi.org/10.1002/nme.7135.

[73] Manav M, Molinaro R, Mishra S, De Lorenzis L. Phase-field modeling of fracture with physics-informed deep learning. Comput Methods Appl Mech Eng 2024;429:117104. https://doi.org/10.1016/j.cma.2024.117104.

[74] Gao H, Zuo W, Feng Z, Yang J, Li T, Hu P. DHEM: a deep heat energy method for steady-state heat conduction problems. J Mech Sci Technol 2022;36:5777–91. https://doi.org/10.1007/s12206-022-1039-0.

[75] Lee S-W, Truong-Quoc C, Ro Y, Kim D-N. Adversarial deep energy method for solving saddle point problems involving dielectric elastomers. Comput Methods Appl Mech Eng 2024;421:116825. https://doi.org/10.1016/j.cma.2024.116825.

[76] Lin KC, Hu CH, Wang KC. Innovative deep energy method for piezoelectricity problems. Appl Math Model 2024;126:405–19.

[77] Lin K-C, Wang K-C, Hu C-H. Investigating deep energy method applications in thermoelasticity. Eng Anal Bound Elem 2024;159:302–14. https://doi.org/10.1016/j.enganabound.2023.12.012.

[78] Nguyen-Thanh VM, Anitescu C, Alajlan N, Rabczuk T, Zhuang X. Parametric deep energy approach for elasticity accounting for strain gradient effects. Comput Methods Appl Mech Eng 2021;386:114096. https://doi.org/10.1016/j.cma.2021.114096.

[79] Fuhg JN, Bouklas N. The mixed Deep energy method for resolving concentration features in finite strain hyperelasticity. J Comput Phys 2022;451:110839. https://doi.org/10.1016/j.jcp.2021.110839.

[80] Rezaei S, Harandi A, Moeineddin A, Xu B-X, Reese S. A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: comparison with finite element method. Comput Methods Appl Mech Eng 2022;401:115616. https://doi.org/10.1016/j.cma.2022.115616.

[81] Wang Y, Sun J, Li W, Lu Z, Liu YCENN. Conservative energy method based on neural networks with subdomains for solving variational problems involving heterogeneous and complex geometries. Comput Methods Appl Mech Eng 2022;400:115491. https://doi.org/10.1016/j.cma.2022.115491.

[82] Bai J, Liu G-R, Rabczuk T, Wang Y, Feng X-Q, Gu Y. A robust radial point interpolation method empowered with neural network solvers (RPIM-NNS) for nonlinear solid mechanics. Comput Methods Appl Mech Eng 2024;429:117159. https://doi.org/10.1016/j.cma.2024.117159.

[83] Chadha C, He J, Abueidda D, Koric S, Guleryuz E, Jasiuk I. Improving the accuracy of the deep energy method. Acta Mech 2023;234:5975–98. https://doi.org/10.1007/s00707-023-03691-3.

[84] Wang Y, Sun J, Rabczuk T, Liu Y. DCEM: a deep complementary energy method for linear elasticity. Int J Numer Methods Eng 2024;125:e7585. https://doi.org/10.1002/nme.7585.

[85] Eshaghi M.S., Anitescu C., Thombre M., Wang Y., Zhuang X., Rabczuk T. Variational physics-informed neural operator (VINO) for solving partial differential equations 2025. https://doi.org/10.48550/arXiv.2411.06587.

[86] Li W, Bazant MZ, Zhu J. A physics-guided neural network framework for elastic plates: comparison of governing equations-based and energy-based approaches. Comput Methods Appl Mech Eng 2021;383:113933. https://doi.org/10.1016/j.cma.2021.113933.
[87] Müller J, Zeinhofer M. Achieving high accuracy with PINNs via energy natural gradient descent. In: Proceedings of the 40th International Conference on Machine Learning. PMLR; 2023. p. 25471–85.
[88] He J, Abueidda D, Koric S, Jasiuk I. On the use of graph neural networks and shape-function-based gradient computation in the deep energy method. Int J Numer Methods Eng 2023;124:864–79. https://doi.org/10.1002/nme.7146.
[89] Xiong W, Long X, Bordas SPA, Jiang C. The deep finite element method: a deep learning framework integrating the physics-informed neural networks with the finite element method. Comput Methods Appl Mech Eng 2025;436:117681. https://doi.org/10.1016/j.cma.2024.117681.
[90] Wang Y., Lin Y., Goswami S., Zhao L., Zhang H., Bai J., et al. Towards unified AI-driven fracture mechanics: the extended deep energy method (XDEM) 2025. https://doi.org/10.48550/arXiv.2511.05888.
[91] Chen X-X, Zhang P, Yin Z-Y. A comprehensive investigation of physics-informed learning in forward and inverse analysis of elastic and elastoplastic footing. Comput Geotech 2025;181:107110. https://doi.org/10.1016/j.compgeo.2025.107110.
[92] Thombre M, Anitescu C, Bharadwaja B, Wang Y, Rabczuk T, Alankar A. Energy-based methods for solving forward and inverse linear elasticity problems in 2D structures. Comput Struct 2025;316:107899. https://doi.org/10.1016/j.compstruc.2025.107899.
[93] Rathore P., Lei W., Frangella Z., Lu L., Udell M. Challenges in training PINNs: a loss landscape perspective 2024. https://doi.org/10.48550/arXiv.2402.01868.
[94] Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell 2021;3:218–29. https://doi.org/10.1038/s42256-021-00302-5.
[95] Arnold DN, Awanou G. The Serendipity Family of finite elements. Found Comput Math 2011;11:337–44. https://doi.org/10.1007/s10208-011-9087-3.