communications engineering

A Nature Portfolio journal

Article

https://doi.org/10.1038/s44172-025-00476-5

Information-distilled physics informed deep learning for high order differential inverse problems with extreme discontinuities

Check for updates

Mingsheng Peng [1,2,3] & Hesheng Tang [1,2,3]

Standard physics informed deep learning and their enhanced variants encounter challenges in addressing inverse problems characterized by extreme discontinuities and high-order parameterized differential equations due to the use of globally smooth activation functions, especially when the unknown parameters exhibit spatially distributed characteristics. Phenomena such as discontinuous loads, boundary truncations, and abrupt changes in material properties introduce singularities in the derivatives, which in turn lead to ill-conditioned information in the gradient flow. To address these limitations, here we propose an information-distilled physics-informed deep-learning framework that combines reduced-order modeling, multi-level domain decomposition, and an ill-conditioning-suppression mechanism. The framework captures rapid variations in variables within highly localized regions induced by discontinuities. Through an information propagation mechanism and information distillation, the ill-conditioned information in the gradient flow of the system is suppressed. Even in scenarios where specific subnetworks fail, the framework preserves the accuracy of the majority of subnetworks.

Differential equations with nonlinear characteristics and parameter modeling capabilities can mathematically describe various physical phenomena. Solving differential equations with fixed parameters is termed forward problem. The solution to forward problem[1–3] can dynamically capture the behavioral changes of systems under different conditions. Inferring unknown parameters from observational data is termed inverse problem. Solving inverse problems is widely applied in fields such as health monitoring and parameter identification. However, inverse problems are generally more challenging than forward problems due to their ill-posed nature, noise interference, difficulty in data acquisition, and modeling complexity[4].

In recent years, deep learning has been applied to various fields and has achieved wide adoption[5,6]. Physics-Informed Neural Networks (PINNs)[7] are machine learning techniques used to solve forward and inverse problems by embedding the constraints of differential equations into neural networks. Due to the grid-free nature and noise resistance of PINNs, and the fact that solving inverse problems with PINNs is as straightforward as solving forward problems, this approach holds great promise for overcoming the limitations of traditional numerical methods[8–12].

However, with the rapid development and widespread use of PINNs, some inherent defects have emerged. These defects stem from the solution approach of PINNs, which involves computing dependent differential operators, optimizing a global loss function, and simultaneously training multiple tasks[13–15]. Certain enhanced PINNs lack the capability to handle strong discontinuities effectively. Based on Neural Tangent Kernel theory, an adaptive weighting technique is proposed to automatically assign suitable weights to various terms of the loss function[16]. Trial functions are used to project the loss function into high-dimensional spaces, producing satisfactory results in approximating high-frequency functions[17]. Structural modifications of neural networks enhance the efficiency of solving high-frequency problems with PINNs[18]. The non-dimensionalization of differential equations enhances computational stability[19]. Deep reduced-order PINNs[20] and Multi-level PINNs (M-PINNs)[21] flexibly handle various boundary conditions (BCs) by applying hard constraints on boundaries, demonstrating improved performance in solving discontinuous and high-order problems.

Extreme discontinuities—defined as localized jumps whose magnitude tends to infinity within a very narrow region—pose challenges that existing

[1]Department of Disaster Mitigation for Structures, College of Civil Engineering, Tongji University, Shanghai, China. [2]State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University, Shanghai, China. [3]These authors jointly supervised this work: Mingsheng Peng, Hesheng Tang. e-mail: thstj@tongji.edu.cn

pre-decomposition methods cannot satisfactorily resolve, whereas they cope well with ordinary discontinuities. Extended Physics-Informed Neural Networks (XPINNs)[22] employ domain decomposition techniques, utilizing distinct networks to decompose the computational domain, handling problems with intricate computational domains. Diao et al.[23] applied domain decomposition-based PINNs to solve multi-material solid mechanics problems. Sarma et al.[24] reduced training overhead and achieved comparable accuracies to traditional domain decomposition methods by sharing network parameters between subnetworks. Hu et al.[25] improved the multi-network structure by using augmented coordinate variables to label the pieces of each subdomain. Liu et al.[26] introduced the Rankine–Hugoniot relation, which is equivalent to the weak form of the conservation laws near discontinuities, enhancing the shock-capturing performance of PINNs. Peng et al.[27] achieved automatic domain decomposition for high-order ODEs; however, the approach has not yet been extended to inverse problems or PDEs.

Higher-order derivatives exacerbate the complexity of solving problems with discontinuities. Additional loss function terms and residual points for interface conditions between subnetworks are needed for domain decomposition PINNs. When discontinuities occur in one order of physical quantity, the remaining orders of quantities that satisfy continuity conditions are also decomposed. Reconnecting them using loss functions is an inefficient approach. Moreover, using soft constraints for continuity conditions in subnetworks inherently has precision limitations. When the region containing the discontinuity is extremely small, referred to as an extreme discontinuity, the solution often exhibits a significant order-of-magnitude difference compared with other subdomains, leading to optimization difficulties. Furthermore, due to the limited number of training points in this region, the risk of overfitting arises[28].

Ill-posedness is a common challenge encountered in inverse problems. In the inversion of spatially distributed parameters, division by zero can result in severe numerical instabilities. Concentrated loads, material discontinuities, and boundary truncations further exacerbate the singularities, making the problem even more challenging to solve. The singularities in the system response caused by discontinuities can propagate to the unknown parameters, leading to a failure in the inversion. The identified parameters exhibit huge errors at the boundaries, some studies directly treat the boundary conditions of unknown parameters as known[29]. Even M-PINNs also encounter singularity issues caused by boundary truncation during the gradient flow propagation process. Methods utilizing PINNs for inverse problems lack the capability to identify parameters with spatial distribution in high-order differential equations that contain extreme discontinuities[10,30,31].

To address these challenges, a multi-level domain decomposition and Reduced-order PINNs (DR-PINNs) with an ill-conditioning-suppressing mechanism is proposed to solve high order problems with extreme discontinuities and spatial distributed unknown parameters. The contributions of this paper are summarized as follows:

1) Reduced-order refers to the transformation of high-order differential equations into a series of lower-order differential equation systems. Based on this transformation, corresponding subnetworks are designed according to the specific characteristics of variables at different orders. Corresponding subnetworks are tailored according to the characteristics of variables with different orders. Compared to M-PINNs, each order of physical quantity is flexibly assigned a distinct multi-level domain decomposition scheme, effectively avoiding the singularities induced by discontinuities and minimizing additional computational overhead.

2) In extreme discontinuities, it is difficult to enforce interface conditions for variables with large order-of-magnitude differences using soft constraints. To address this issue, the interface condition hard constraint technique is used. Unlike traditional XPINNs, interface conditions for variables of any order can be enforced through hard constraints, avoiding the need for additional loss function settings and residual point collection, which enhances the solution efficiency.

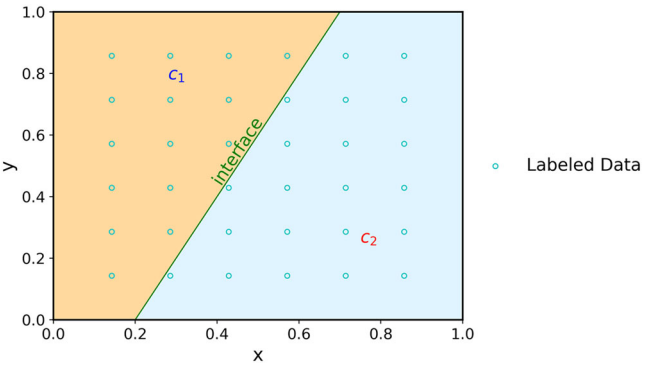


Fig. 1 | Poisson problem with a slanted material discontinuity. Unit-square domain ($0 \le x,\ y \le 1$) used to test DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks). A green line splits the domain into two sub-regions with diffusion coefficients c_1 (beige, upper-left) and c_2 (light blue, lower-right). Cyan open circles denote the labeled solution points supplied for training.

3) A suppression function for information distillation and blocking ill-conditioned information propagation is introduced to improve the quality of gradient flow. The term information distillation differs from its conventional usage. The DR-PINNs framework incorporates a mechanism that suppresses such ill-conditioning by replacing inaccurate gradients with more reliable ones. The singularity issues encountered during the inversion of unknown parameters are addressed. The efficient information propagation mechanism ensures that DR-PINNs can handle uncertain information in inverse problems. Even if some networks malfunction, DR-PINNs can still guarantee the accuracy of most networks, whereas original PINNs exhibit a complete inability.

Results

Application of DR-PINNs to Poisson Equation

Widely used in fields such as electrostatics, heat conduction, and fluid dynamics, the Poisson equation serves as a fundamental model for describing steady-state distributions influenced by spatially varying sources. Consider the following two-dimensional Poisson equation:

$$\frac{\partial}{\partial x}\left(c(x,y)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(c(x,y)\frac{\partial u}{\partial y}\right)=1, x\in[0,1], y\in[0,1] \quad (1)$$

with the boundary conditions

$$u(0,y)=u(1,y)=u(x,0)=u(x,1)=0 \quad (2)$$

As shown in Fig. 1, to introduce discontinuous characteristics, the coefficient c is defined in the following piecewise forms

$$c=\begin{cases}1, & 4x-2y\le 1\\ 2, & 4x-2y>1\end{cases} \quad (3)$$

The value of u obtained from a uniformly distributed 6×6 grid is utilized as labeled input to guide the training process. A total of 2000 training points were randomly sampled within the computational domain. A fully connected neural network with three hidden layers and 50 neurons per layer was employed to output the required physical quantities, using x and y as inputs. Another fully connected network with three hidden layers and 20 neurons per layer was used to predict the values at the interface, using $4x-2y-0.5$ as input. The training process was conducted using a learning rate of 10^{-3} for 20,000 epochs, followed by an additional 30,000 epochs with a reduced learning rate of 10^{-4}.

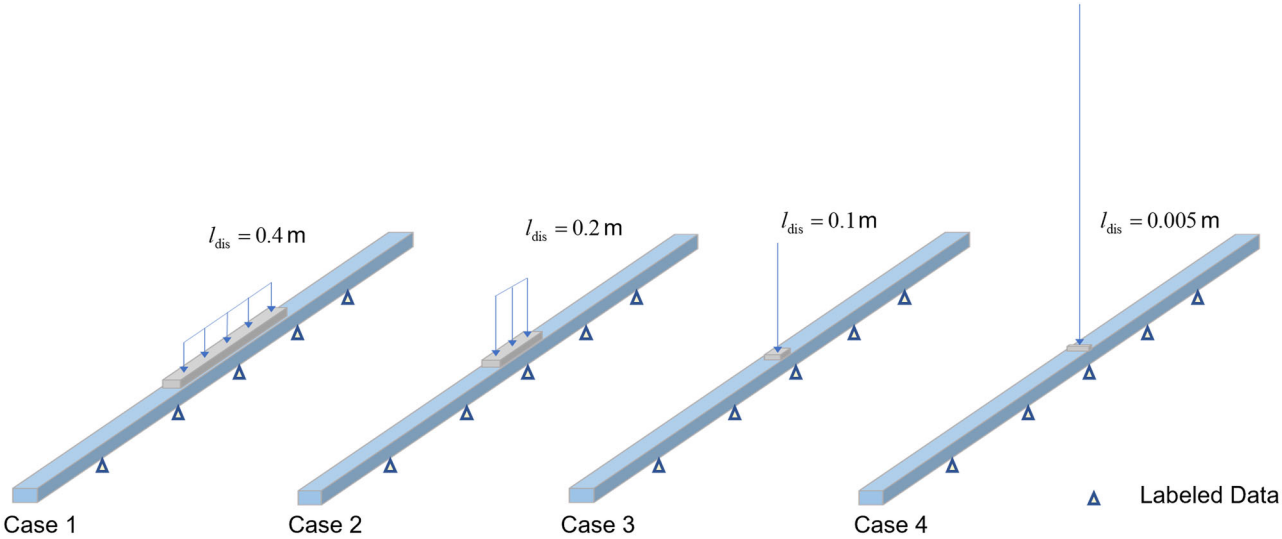


Fig. 2 | Euler beams under concentrated loads. The varying characteristics of the loads result in system discontinuities with distinct features. Four beam cases with decreasing load-distribution length l_{dis} = 0.40 m, 0.20 m, 0.10 m, 0.005 m, respectively. Downward blue arrows indicate the applied concentrated load, and hollow triangles mark the displacement data points supplied for training. Shorter l_{dis} values produce sharper discontinuities in the structural response.

Supplementary Fig. 1 illustrates the iterative results during training, including the identified unknown parameters and the loss function. In the early stages of training, the loss function decreased rapidly, and the unknown parameters in different regions quickly converged to accurate values. As the learning rate decreased, the loss function continued to decline, and the parameter c converged to its true value with minimal error. Supplementary Fig. 3 presents the results of representative variables. DR-PINNs successfully captured distinct features of the target functions using a simple fully connected architecture, without requiring extensive hyperparameter tuning.

Application of DR-PINNs to beam problems subjected to static loads

The Euler-Bernoulli beam is a typical high-order problem, where each order of the derivative may exhibit discontinuities corresponding to specific engineering scenarios. Taking beam problems with extreme discontinuities and unknown spatially distributed parameters as examples, the accuracy and efficiency of the proposed method are demonstrated. The governing equation of the Euler-Bernoulli beam under static load is given by

$$-\frac{d^2}{dx^2}\left(EI(x)\frac{d^2w(x)}{dx^2}\right)+q(x)=0 \tag{4}$$

where $EI(x)$ is the flexural stiffness, $w(x)$ is the transverse displacement, and $q(x)$ is the transverse load. And

$$\varphi=-\frac{dw}{dx} \tag{5}$$

$$M=EI\frac{d\varphi}{dx} \tag{6}$$

$$V=\frac{dM}{dx} \tag{7}$$

$$q=-\frac{dV}{dx} \tag{8}$$

Equations (5) to (8) shows the relationships among transverse displacement w, slope φ, bending moment M, shear force V, and spatially distributed load q. In practical engineering, due to effects of supports and concentrated loads, the shear force in a beam often exhibits discontinuities. The direct cause of shear force discontinuities is that bending moment does not satisfy the C^1 continuity condition, while slope and deflection exhibit smooth characteristics and discontinuities in high-order derivatives.

Consider the Euler-Bernoulli beam under a load as shown in Fig. 2. The beam is simply supported at both ends and has a length of L = 1.5 m and a cross-sectional width b = 50 mm. The cross-sectional height is h = 50 mm and the elastic modulus is E = 3 GPa. The load magnitude for Case 1 to Case 4 is $F = 200\,N$. The region where the concentrated load is applied extends across $(L/2-l_{dis}/2, L/2+l_{dis}/2]$. l_{dis} is defined as 0.4 m, 0.2 m, 0.1 m and 0.005 m, respectively.$q(x)$ is given by

$$q(x)=\frac{\delta(x)F}{l_{dis}} \tag{9}$$

where

$$\delta(x)=\begin{cases}0, & x\subset[0,L/2-l_{dis}/2]\cup(L/2+l_{dis}/2,1.5]\\ 1, & x\subset(L/2-l_{dis}/2,L/2+l_{dis}/2]\end{cases} \tag{10}$$

As the load distribution becomes more concentrated, the magnitude of the load increases, resulting in more pronounced discontinuities. Additionally, certain physical quantities in the midspan region exhibit large order-of-magnitude differences compared with other regions, leading to increased computational complexity. The flexural stiffness of the material undergoes the following alteration

$$EI(x)=EI\times[0.75+(x/L-0.5)^2] \tag{11}$$

In M-PINNs, networks $\hat{w}$, $\hat{\varphi}$, $\hat{M}$ and $\hat{V}$ represent w, φ, M and V. Related studies employ this framework for inverse problems without discontinuities[31]. However, the present study finds that curvature $d\varphi/dx$, as a bridge between the geometric equation and the equilibrium equation, directly affects the inversion results of stiffness and plays a crucial role. Let curvature $d\varphi/dx$ be denoted as C. Due to the coupling between EI and C, boundary conditions applied through M cannot be accurately imposed on C, but instead affect the predictions of EI. Therefore, a network $\hat{C}$ for approximating curvature is introduced.

Because the concentrated load produces two sharp discontinuities in the internal fields, the beam domain is partitioned into three sub-intervals

(The detailed procedure and notation are provided in the Methods section). The specific loss functions and network configurations are provided in the Supplementary Notes 1. The loss function term directly relied upon for the inversion of flexural stiffness is given

$$\mathcal{L}_{\mathcal{F}_3}^{DR} = \frac{1}{N_{\mathcal{F}_3}} \sum_{x \in \mathcal{T}_{\mathcal{F}_3}} \left| EI \cdot \sum_{j=1}^{n+1} \hat{C}^{\{j\}} \cdot \chi_{\Omega_2^{\{j\}}} - \sum_{j=1}^{n+1} \hat{M}^{\{j\}} \cdot \chi_{\Omega_3^{\{j\}}} \right|^2 \tag{12}$$

When $\mathcal{L}_{\mathcal{F}_3}^{DR}$ is sufficiently small, and $\hat{C} \neq 0$, the following equation holds:

$$EI = \frac{\sum_{j=1}^{n+1} \hat{M}^{\{j\}} \cdot \chi_{\Omega_3^{\{j\}}}}{\sum_{j=1}^{n+1} \hat{C}^{\{j\}} \cdot \chi_{\Omega_2^{\{j\}}}} \tag{13}$$

Equation (13) fundamentally elucidates the principle of identifying spatially distributed flexural stiffness, emphasizing the critical roles of bending moment and curvature in the inversion process. In order to improve training results, it is necessary to eliminate the propagation of pathological information caused by soft BCs in areas where curvature $\hat{C}$ is zero, specifically the transmission of $d\hat{\varphi}/dx$. According to Eq. (30) in the Methods section, the term $\mathcal{L}_{\mathcal{F}_2}^{DR}$ in Eq. (S1) is modified as $\mathcal{L}_{\mathcal{F}_2}^{DR} := \mathcal{W}(\varepsilon) \times \mathcal{L}_{\mathcal{F}_2}^{DR}$.

To thoroughly demonstrate the competitiveness of DR-PINNs, traditional PINNs and domain decomposition PINNs (XPINNs) were also applied to solve this problem. Under the traditional PINNs framework, the external load at mid-span can be introduced in the form of a delta function, as shown in Eq. (10), denoted as PINNs-Delt. Alternatively, a Gaussian function can be used to achieve a smoother external load, denoted as PINNs-Gaus. The loss functions of different methods under varying l_{dis} are illustrated in Supplementary Fig. 2. Across various conditions, DR-PINNs consistently exhibit the lowest loss function values.

The graphical results for $l_{\text{dis}} = 0.4$ m are presented in Fig. 3, while those for $l_{\text{dis}} = 0.005$ m are shown in Fig. 4. The L_2 norm errors for more different l_{dis} values and solution methods are summarized in Supplementary Table. When l_{dis} decreases to 0.2, although the accuracy of XPINNs slightly declines, it remains comparable to that at $l_{\text{dis}} = 0.4$ m, achieving an accuracy of 10^{-3} for lower-order physical quantities and 10^{-2} for higher-order physical quantities. However, the parameter identification of PINNs-Gaus and PINNs-Delt completely fails. When l_{dis} further decreases to 0.1 m, parameter identification fails completely for PINNs-Gaus, PINNs-Delt, and XPINNs, despite careful tuning of hyperparameters. The remaining physical quantities, particularly higher-order ones, produce highly inaccurate results.

In contrast, DR-PINNs consistently maintain excellent performance regardless of the variations in l_{dis}. Through domain decomposition and the use of hard constraints at subnetwork interfaces in the Methods section, DR-PINNs mitigate singularities caused by discontinuities, reduce computational burden, and enhance accuracy and efficiency. By suppressing the propagation of ill-conditioned information at boundaries (i.e., setting $\varepsilon \neq 0$), the accuracy of the inferred spatially distributed unknown parameters is further improved. To fully demonstrate DR-PINNs' ability to solve extreme discontinuities, l_{dis} is further reduced to 0.005 m, with the graphical results shown in Fig. 4. DR-PINNs continue to achieve high overall accuracy, especially after suppressing the propagation of ill-conditioned boundary information.

Since Eq. (4) is equivalent to Eqs. (5)–(8), Eq. (13) also explains the principle behind flexural stiffness identification in original PINNs, which means C and M are also the critical information for flexural stiffness estimation in PINNs-Delt and PINNs-Gaus. In original PINNs, the output of higher-order physical fields depends on the unidirectional operation of the network. This implies that once the identification of flexural stiffness (EI) fails, the higher-order physical fields (M) are also adversely affected. During the parameter update process, PINNs will not only update the network outputting the flexural stiffness but may also update the deflection network to reduce the PDE residual. This results in situations where PINNs struggle to achieve sufficiently small residuals on both the labeled data and the governing PDEs simultaneously.

The ideal solution is to establish spatially distributed flexural stiffness based on the information of curvature and moment, rather than having the information of curvature and moment influenced by flexural stiffness, which has low accuracy. To demonstrate the superior information propagation mechanism of DR-PINNs, the flexural stiffness $\widehat{EI}$ was intentionally set to zero. After 20,000 epochs of training, the output results of DR-PINNs are shown in Supplementary Fig. 5. It can be observed that, despite a marked deviation of $\widehat{EI}$ from its true value, the remaining networks maintain very high accuracy. This is due to the fact that the physical fields generated by the DR-PINNs are directly produced by the network, rather than being derived from network gradients. The output of networks depends not only on the gradient information from higher-level networks but also closely on its gradient information and that of lower-level networks. Therefore, DR-PINNs can solve for high-accuracy bending moment networks according to the governing equation (physics) and solve for a curvature network based on the deflection (label data). It is important to note that physics and data are not separated, but interact in a clearer manner. Consequently, the error will be corrected and significantly limited during the propagation process. The clever information dissemination mechanism makes the information of bending moment and curvature independent of the flexural stiffness.

When $l_{\text{dis}} = 0.005$, the detailed solution information of DR-PINNs under different values of ε is compared during the training. Figure 5 shows the identification process when $\varepsilon = 0$ with iteration counts of (a) 120k and (b) 480k; and when $\varepsilon = 0.1$ with iteration counts of (c) 120k and (d) 480k. Figure 5e presents the relative error of $\widehat{EI}$. The comparison includes the ratio of the bending moment $\hat{M}$ under hard constraints to the curvature $\hat{C}$ under hard constraints (Hard BC), the ratio of the bending moment $\hat{M}$ under hard constraints to the curvature $d\hat{\varphi}/dx$ under soft constraints (Soft BC), the flexural stiffness value directly output by the $\widehat{EI}$, and (d) the relative error of $\widehat{EI}$.

When $\varepsilon = 0$, the rotational gradient information $d\hat{\varphi}/dx$ under soft constraints has relatively low accuracy in regions where the curvature is close to 0, leading to inefficiency and instability in recognition. On the one hand, it is challenging to obtain high-gradient information from rare labeled data in locations with minimal impact. On the other hand, the boundary $d\hat{\varphi}/dx$ should be 0, but soft constraint formula can only make $d\hat{\varphi}/dx$ as close to 0 as possible, and it cannot exactly equal 0. When $d\hat{\varphi}/dx$ is as small as possible but not equal to zero, small fluctuations can cause significant changes and oscillations in the ratio of $\hat{M}$ to $d\hat{\varphi}/dx$, exhibiting strong pathological characteristics. In contrast, the ratio of the $\hat{M}$ to the curvature $\hat{C}$ under hard constraints is relatively stable and does not show oscillatory characteristics at the boundary. But due to the bidirectional nature of information propagation, the hard constraint information is influenced by the soft constraint information and low-precision flexural stiffness information. And the pathological characteristics of soft constraints directly affect the predicted value $\widehat{EI}$ through the information propagation path. To reduce the residuals, $\widehat{EI}$ tends to capture values with large magnitudes that are far away from and oscillate near the true value. The output results of $\widehat{EI}$ are closely related to the ratio of $\hat{M}$ to $d\hat{\varphi}/dx$ at the boundary residual points. In regions where the $\hat{C}$ is zero, the low-accuracy gradient information of curvature results in low-precision identification of flexural stiffness. This manifests as prolonged capture of erroneous information and stagnation in optimization.

When $\varepsilon = 0.1$, the pathological nature of soft BCs still exists. However, by suppressing $\mathcal{L}_{\mathcal{F}_2}^{DR}$, the transmission path of the pathological information to $\mathcal{L}_{\mathcal{F}_3}^{DR}$ and then to $\widehat{EI}$ is blocked. As a result, the more accurate hard BCs information is no longer affected by the pathological information and quickly outputs more accurate flexural stiffness. By the 120k-th iteration, the deviation of $\widehat{EI}$ from the true value at the boundary has been limited to a certain extent. At the 480k-th iteration, the accuracy of $\widehat{EI}$ at both the boundary and across the entire domain further improves. After eliminating

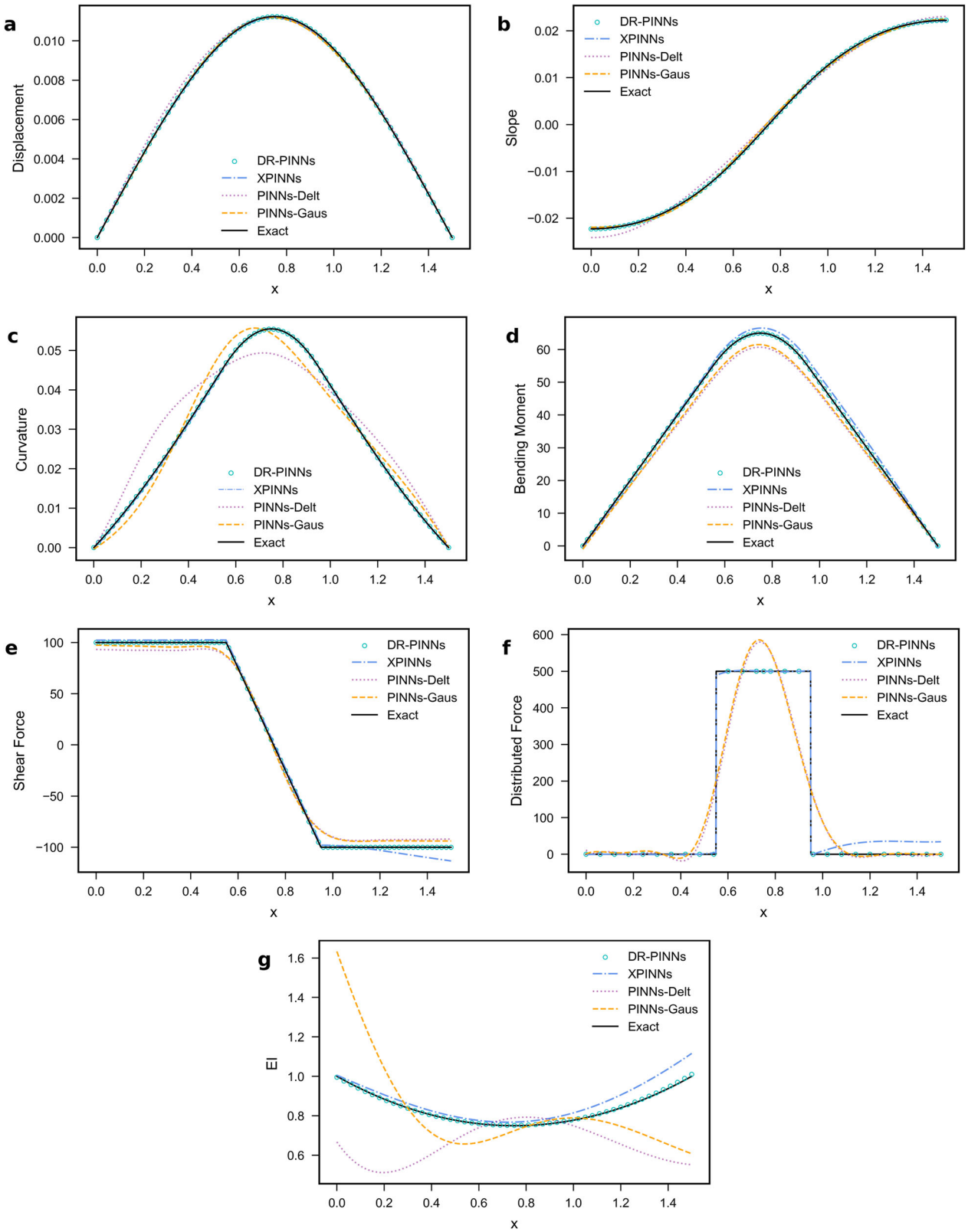


Fig. 3 | The solution results of different method for a case with a mild discontinuity. a Transverse displacement $w(x)$. **b** Slope $\varphi(x) = dw(x)/dx$. **c** Curvature $C(x) = d\varphi(x)/dx$. **d** Bending moment $M(x) = EI(x) \cdot C(x)$. **e** Shear force $V(x) = dM(x)/dx$. **f** Distributed load $q(x) = dV(x)/dx$. **g** Flexural rigidity $EI(x)$ identified by each method. Cyan open circles — DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks); blue dash-dot line — XPINNs (extended physics-informed neural networks); purple dotted line — PINNs-Delt (physics-informed neural network in which the concentrated load is represented by a Dirac-delta function); orange dashed line — PINNs-Gaus (physics-informed neural network in which the concentrated load is approximated by a Gaussian function); black solid line — exact solution. Here, EI is normalized.

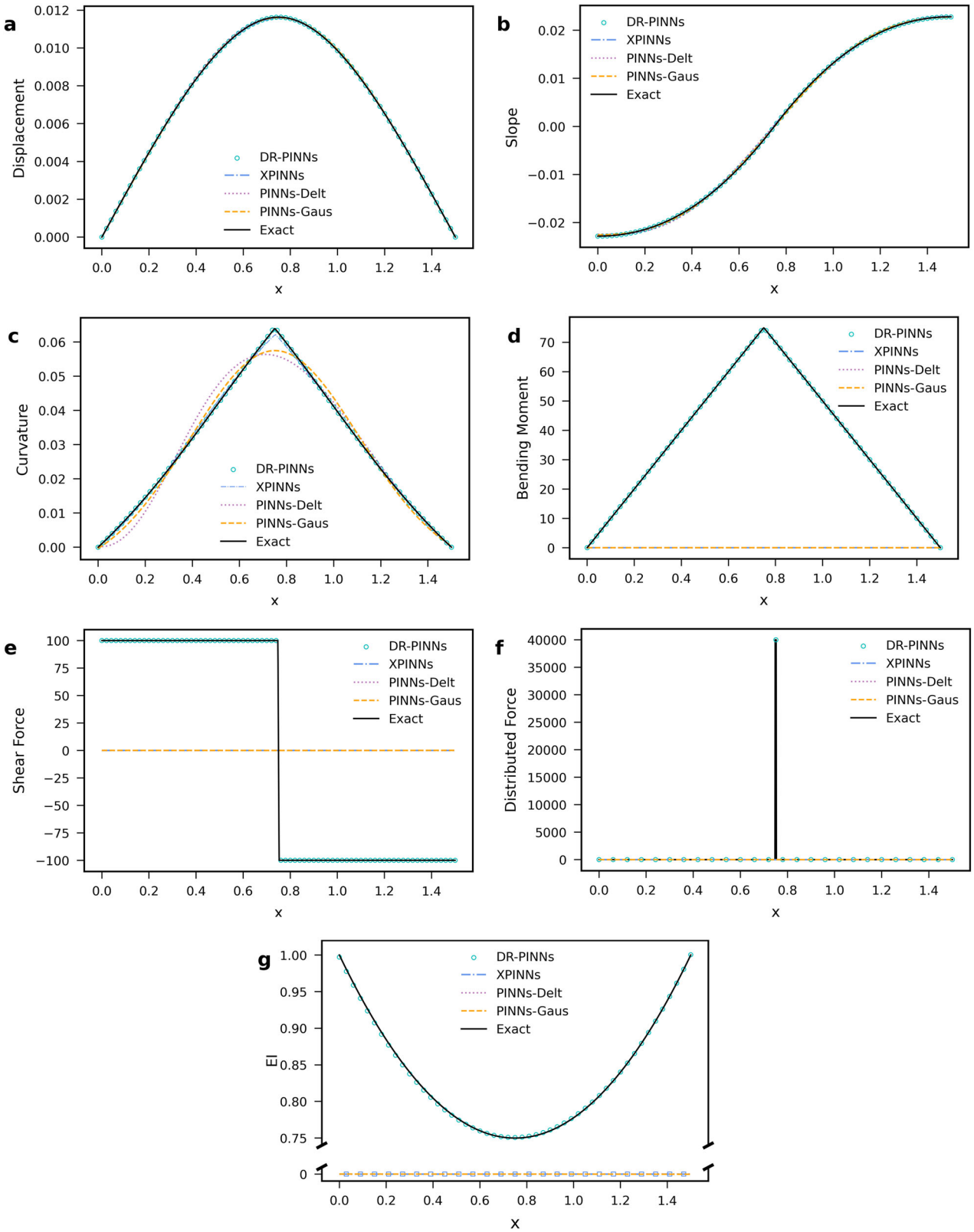


Fig. 4 | The solution results of different method for a case with an extreme discontinuity. a Transverse displacement $w(x)$. **b** Slope $\varphi(x) = dw(x)/dx$. **c** Curvature $C(x) = d\varphi(x)/dx$. **d** Bending moment $M(x) = EI(x) \cdot C(x)$. **e** Shear force $V(x) = dM(x)/dx$. **f** Distributed load $q(x) = dV(x)/dx$. **g** Flexural rigidity $EI(x)$ identified by each method. Cyan open circles — DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks); blue dash-dot line — XPINNs (extended physics-informed neural networks); purple dotted line — PINNs-Delt (physics-informed neural network in which the concentrated load is represented by a Dirac-delta function); orange dashed line — PINNs-Gaus (physics-informed neural network in which the concentrated load is approximated by a Gaussian function); black solid line — exact solution. Here, EI is normalized.

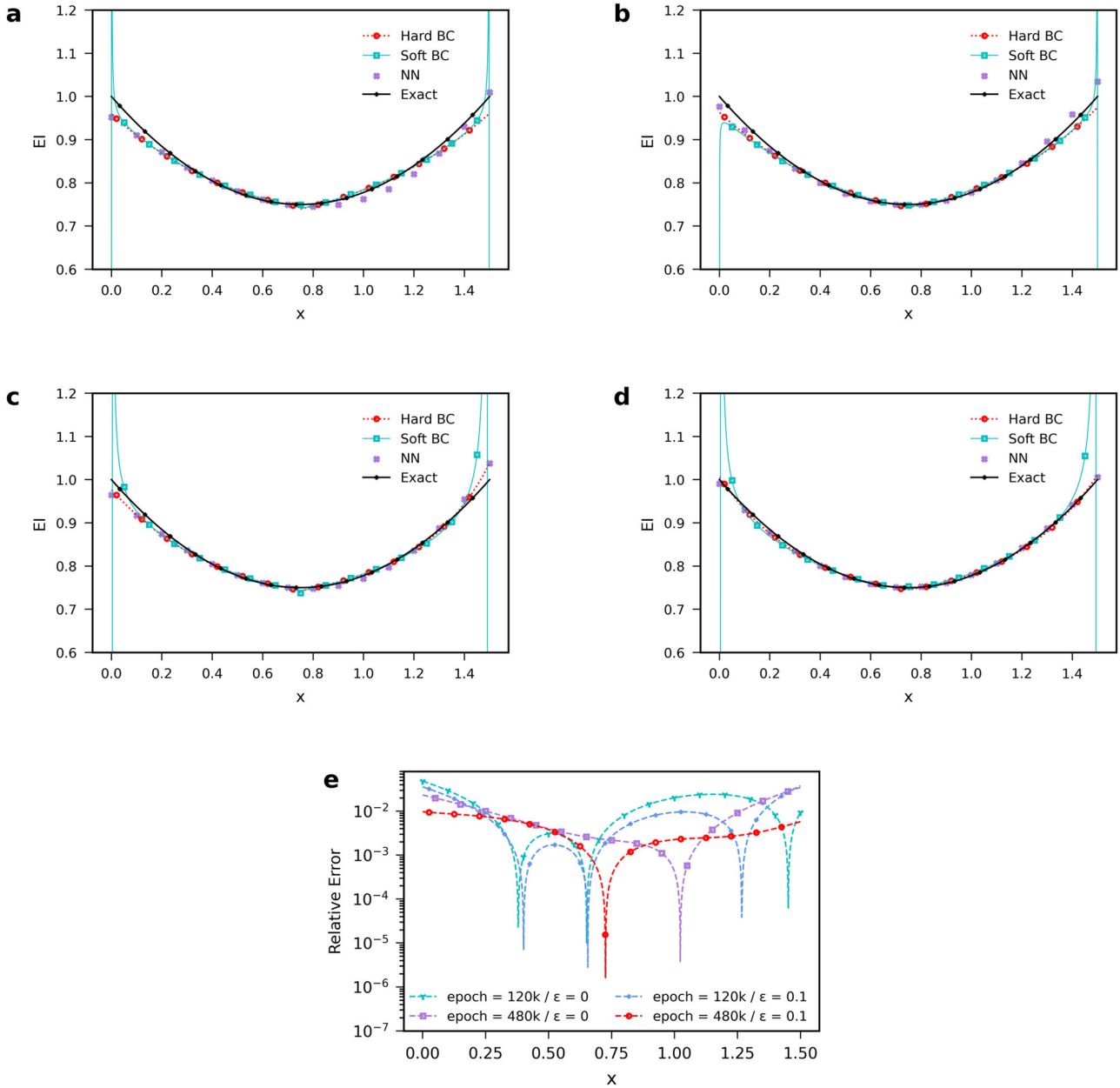


Fig. 5 | Effect of the suppression function $\mathcal{W}(\varepsilon)$ on the learned flexural-rigidity field for a case with an extreme discontinuity. **a** $EI(x)$ after 100k training iterations, $\varepsilon = 0$. **b** $EI(x)$ after 480k iterations, $\varepsilon = 0$. **c** $EI(x)$ after 100k iterations, $\varepsilon = 0.1$. **d** $EI(x)$ after 480k iterations, $\varepsilon = 0.1$. Red circles, hard BC (boundary condition); cyan squares, soft BC; purple "×", direct network output by $\widehat{EI}$; black line, exact solution. **e** Point-wise relative error $|EI_{\text{pred}} - EI_{\text{exact}}|/\, EI_{\text{exact}}$ for the four cases in **a**–**d**. Here, EI is normalized.

the pathological information near the regions where $\hat{M} = 0$, training no longer stagnates, and accuracy improves.

Application of DR-PINNs to beam problems subjected to dynamic loads

In contrast to ODEs, PDEs involve differentiation with respect to different dimensions of the variables, which makes parameter estimation more challenging. Traditional PINNs encounter greater challenges when solving PDEs with extreme discontinuities and spatially distributed unknown parameters. The beam problem under dynamic loads is a typical high-order partial differential equation.

Consider the beam shown in Fig. 2, with a material density of ρ = 1180 kg m^{-3} and a concentrated load $F(t) = -30\times\ \sin(10\pi t)$ N at mid-span, where t belongs to $[0,\ T]$ and $T = 0.2$ s. The width of the concentrated force is considered to be 6 mm, i.e., $l_{\text{dis}} = 0.006$ m. Other parameters are the same as those in Fig. 2. Under dynamic loads, the control equation for the beam is

$$\rho A(x)\frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left(EI(x)\frac{\partial^2 w(x,t)}{\partial x^2}\right) = f(x,t) \tag{14}$$

$f(x,t)$ is the dynamic load. Here,

$$f(x,t) = \delta(x)\times F(t) \tag{15}$$

where

$$\delta(x) = \begin{cases} 0, & x \subset [0, 0.747)\cup[0.753, 1.5] \\ 1, & x \subset [0.747, 0.753) \end{cases} \tag{16}$$

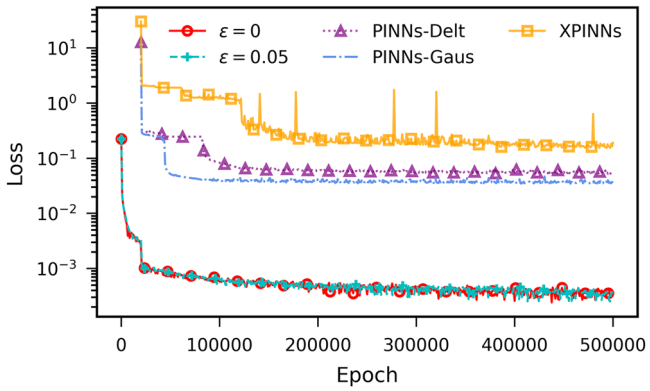


Fig. 6 | Training loss under dynamic loading for five PINN strategies. Log-scaled loss versus epoch (0–500k) for DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks) with $\varepsilon = 0$ (red circles) and $\varepsilon = 0.05$ (cyan crosses), PINNs-Delt, in which the concentrated load is represented by a Dirac-delta function (purple dotted line with triangles), PINNs-Gaus, in which the load is approximated by a Gaussian function (blue dash–dot line), and XPINNs (extended physics-informed neural networks; orange squares). DR-PINNs achieves more than two orders-of-magnitude lower final loss than the other variants.

The specific loss functions and network configurations are provided in the Supplementary Notes 2. An adaptive activation function technique is employed[32]. The loss functions under dynamic load of different methods are plotted in Fig. 6. After 300,000 iterations, the loss function nearly ceases to decrease, resembling the static force scenario. The loss functions of XPINNs, PINNs-Gaus, and PINNs-Delt are higher than that of DR-PINNs, which fully demonstrates the competitiveness of DR-PINNs in solving PDEs with extreme discontinuities. The fluctuation of $\widehat{EI}$ is caused by the division form of Eq. (13). However, the inversion principle of $\widehat{EI}$ in the loss function is in the multiplication form as shown in Eq. (S5), resulting in the loss function being insensitive to fluctuations of $\widehat{EI}$ at the boundaries. This effect is evident in the loss-curve plot: although increasing ε from 0 to 0.05 improves the identification accuracy of $\widehat{EI}$, the two loss curves remain almost indistinguishable.

Figure 7 shows the results of different physical variations after 500k iterations, as well as the absolute errors. Supplementary Fig. 6 provides snapshots of various physical quantities at $t = 0.15$ s along with the L_2 Norm of spatially distributed flexural stiffness $EI(x)$. In Supplementary Fig. 6a–f), DR-PINNs are configured with $\varepsilon = 0.05$. DR-PINNs achieve good predictions for all quantities. This would be nearly impossible for other methods. When ε is non-zero, although the loss function does not exhibit a noticeable decrease, the accuracy of $\widehat{EI}$ improves noticeably throughout the training process. Additionally, the overall accuracy of the spatially distributed parameter inversion reaches 4.28E−03 when $\varepsilon \neq 0$, whereas the accuracy is only 1.15E-02 when $\varepsilon = 0$. As shown in Supplementary Fig. 6 (h), when the suppression function is applied, the identified $\widehat{EI}$ not only achieves higher final accuracy but also maintains superior accuracy and efficiency throughout the entire training process.

Application of DR-PINNs to thin plate problems

Thin plate structures are widely present in various engineering fields, including civil engineering, aerospace engineering, and marine engineering, such as glass curtain walls in high-rise buildings[33] and aircraft fuselages[34]. When subjected to external loads or environmental corrosion, thin plate structures may develop cracks, corrosion, and other forms of structural damage. These damages gradually expand and worsen over time, reducing the load-bearing capacity of the structure and increasing potential risks.

The governing equation of thin plates is a typical high-order differential equation with higher dimensionality, where responses in different directions are coupled. Sudden changes in parameters such as stiffness and mass introduce discontinuities into the system, further complicating the problem. In this section, a thin plate problem with discontinuous material properties is used to verify the superiority of DR-PINNs in solving discontinuous problems. The equilibrium equation of internal forces on the thin plate cross-section satisfies

$$\frac{\partial^2}{\partial x^2}\left(D(x,y)\frac{\partial^2 w}{\partial x^2}\right) + 2\frac{\partial^2}{\partial x \partial y}\left(D(x,y)\frac{\partial^2 w}{\partial x \partial y}\right) + \frac{\partial^2}{\partial y^2}\left(D(x,y)\frac{\partial^2 w}{\partial y^2}\right) + \nu\left[\frac{\partial^2 D(x,y)}{\partial x^2}\frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 D(x,y)}{\partial y^2}\frac{\partial^2 w}{\partial x^2} - 2\frac{\partial^2 D(x,y)}{\partial x \partial y}\frac{\partial^2 w}{\partial x \partial y}\right] - q = 0 \tag{17}$$

where

$$D(x,y) = \frac{E(x,y)h^3}{12(1-\nu^2)} \tag{18}$$

In the equation, w represents the deflection, q denotes the applied load, D is the bending stiffness, ν is the Poisson's ratio, and E represents the elastic modulus of the thin plate.

Consider a rectangular thin plate with four simply supported edges, as shown in Supplementary Fig. 4. The plate has a length of $a = 1$ m, a width of $b = 1$ m, and a thickness of $h = 0.05$ m. The material density is $\rho = 7800\,\mathrm{kg\,m^{-3}}$, and the plate is subjected to a uniform load of $q = 1000\,N$. The Poisson's ratio is $\nu = 0.3$. The discontinuity is introduced via a stepwise reduction in the elastic modulus $E(x, y)$, specifically

$$E = \begin{cases} 2.06 \times 10^{11} & x \le 0.5 \\ 1.03 \times 10^{11}, & x > 0.5 \end{cases} \tag{19}$$

The displacement data obtained from a uniformly distributed 10×10 grid is utilized as labeled inputs to guide the training process. The parameter ε is set to 0.05. A total of 2000 training points are randomly sampled within the computational domain. A fully connected neural network with three hidden layers and 100 neurons per layer is used to predict the target physical quantities, taking x, y as input. Additionally, a separate fully connected network with three hidden layers and 20 neurons per layer takes y as input to predict the values at the interface. The model is trained for 60,000 epochs using a learning rate of 10^{-3}, followed by another 40,000 epochs with a reduced learning rate of 10^{-4}. At the 80,000th epoch, 200 new residual points are added using the Residual-based Adaptive Refinement sampling algorithm[11] to further enhance training accuracy.

Figure 8 illustrates the iterative results during training, including the identified elastic modulus and the loss function. At the early stage of training, the loss function decreases rapidly, and the elastic moduli in different regions quickly converge to accurate values. As the learning rate is reduced, the loss function continues to decline, and the elastic modulus converges to the true values with minimal error. In contrast, XPINNs exhibit a consistently higher loss throughout training, and the identified value of E_1 deviates substantially from the ground truth. Supplementary Fig. 7 presents the results of representative variables. Due to the discontinuous distribution of the elastic modulus, the corresponding variables exhibit different characteristics, ranging from smooth behavior to strong discontinuities. Despite this complexity, DR-PINNs are capable of producing functions with varying features using a fully connected network, without requiring extensive parameter tuning.

To quantitatively evaluate the computational efficiency of the proposed DR-PINNs framework, a comparison of runtime with the baseline XPINNs method is presented in Table 1 across a variety of problems, including the Poisson equation, static and dynamic beam problems, and a plate problem involving discontinuities. While DR-PINNs and XPINNs perform similarly on relatively simple problems such as the Poisson equation, DR-PINNs demonstrate substantial improvements in computational efficiency as the complexity of the problem increases. In particular, for high-order problems with strong discontinuities, such as the dynamic beam and plate problems, DR-PINNs reduce runtime by nearly 50% and 82%, respectively, highlighting their scalability and suitability for complex inverse problems.

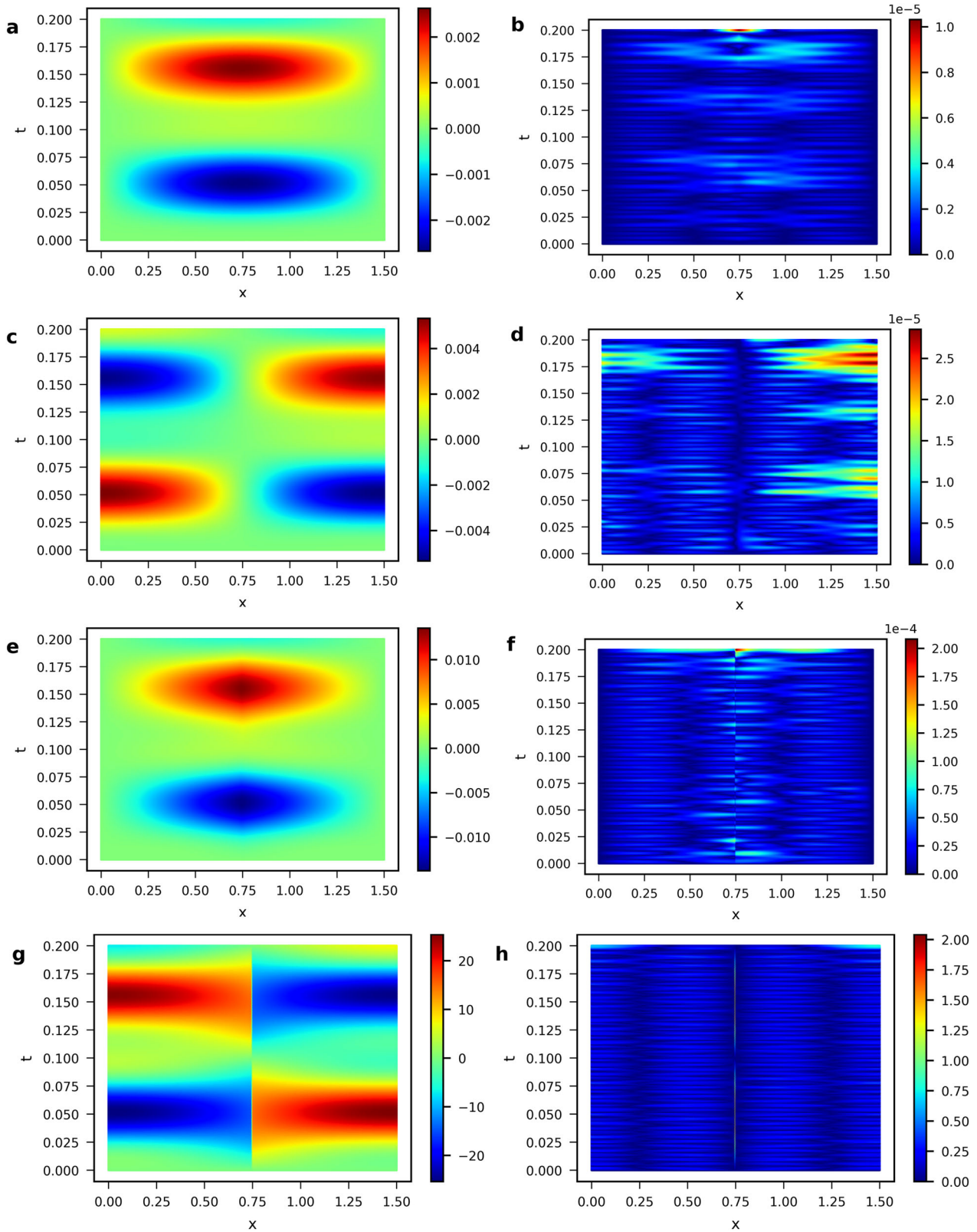


Fig. 7 | Spatio-temporal fields recovered by DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks) and their point-wise errors. a Displacement $w(x,t)$. **b** Absolute error of $w(x,t)$. **c** Slope $\varphi(x,t) = \partial w(x,t)/\partial x$. **d** Absolute error of $\varphi(x,t)$. **e** Curvature $C(x,t) = \partial\varphi(x,t)/\partial x$. **f** Absolute error of $C(x,t)$. **g** Shear force $V(x,t) = \partial M(x,t)/\partial x$. **h** Absolute error of $V(x,t)$.

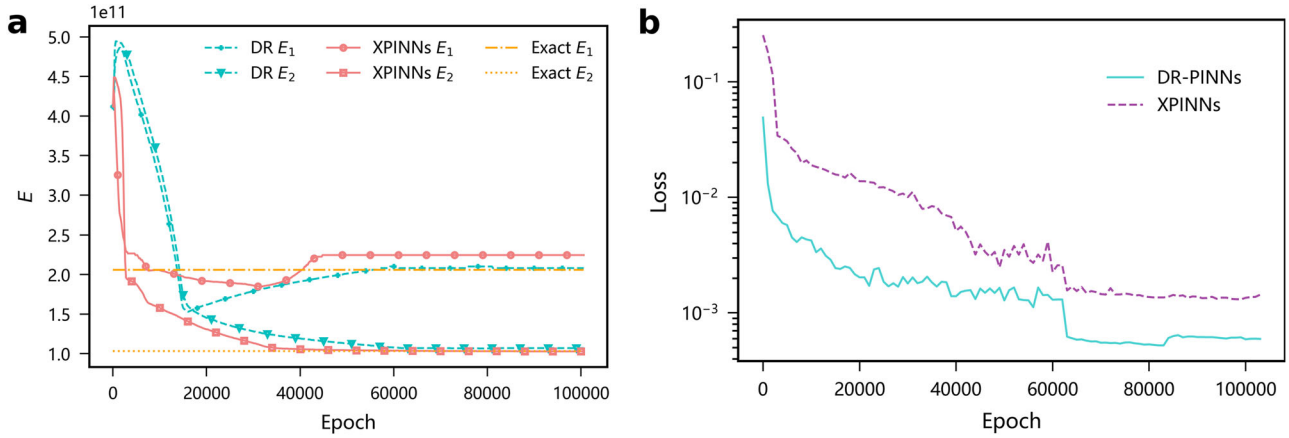


Fig. 8 | Iterative results for the thin plate problem. a Evolution of the two spatially varying elastic modulus E_1 and E_2, over 100k training epochs. Cyan circles, DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks); purple squares, XPINNs (extended physics-informed neural networks). **b** Log-scaled training loss versus epoch for the two methods (cyan solid line, DR-PINNs; purple dashed line, XPINNs). DR-PINNs reaches a loss more than one order of magnitude lower than XPINNs.

Table 1 | Comparison of Runtime (in seconds) for Different Methods on Various Problems

Method	Poisson	Beam (Static)	Beam (Dynamic)	Plate
XPINNs	94	2272	19,764	5872
DR-PINNs	94	2186	10,106	1078
Ratio	1.00	0.96	0.51	0.18

Discussion

By combining reduced-order algorithms and multi-level domain decomposition strategies, a physics informed deep learning named DR-PINNs with an ill-conditioning-suppressing mechanism for information distillation and blocking ill-conditioned information propagation is proposed. This method offers substantial advantages in addressing high order problems with extreme discontinuities and unknown parameters. While traditional PINNs and domain decomposition-based variants fail to address such extreme cases with sufficient accuracy, DR-PINNs offer both superior solution precision and enhanced computational efficiency. Division by zero and boundary truncation lead to challenges such as singularities encountered by PINNs at boundary locations. The core of DR-PINNs is to correct the boundary conditions of the gradient flow during information transmission. When the benefit of correcting the boundaries outweighs the errors caused by the transmission of gradient information in other regions, DR-PINNs exhibit high accuracy. The accuracy of the inversion results is strongly correlated with the ε in the suppression function within a certain range. The suppression function can effectively prevent the propagation of pathological information from soft constraints and enhance the stability of the training results, avoiding singularities. The output of DR-PINNs is influenced not only by the upper network but also by corrections from the lower network. This advanced information propagation mechanism ensures high efficiency and accuracy in parameter inversion.

This study has certain limitations that warrant further investigation. It focuses solely on disruptions induced by a single factor, whereas concurrent disruptions involving multiple physical quantities represent a promising direction for future research. Moreover, the disruption locations are predefined; exploring automatic localization of unknown disruptions and autonomous partitioning of network frameworks offers considerable practical value and deserves further exploration. In addition, the position, quantity, and type of labeled data have shown intriguing influences on the inversion results. A more comprehensive discussion and in-depth analysis of the role of labeled data will be conducted in future studies.

Methods

Framework of DR-PINNs

We start from the special form[27]

$$a_0\hat{\lambda}_0(x)\frac{d^m u(x)}{dx^m}+\cdots+a_r\hat{\lambda}_r(x)\frac{d^{m-r}u(x)}{dx^{m-r}}+\cdots+a_m\hat{\lambda}_m(x)u(x)-q(x)=0 \tag{20}$$

where $u(x)$ is the state variable, the coefficients $\hat{\lambda}_r(x)$ may be unknown, a_r are constant scalars, and $q(x)$ is an external source. Without loss of generality, this template captures the high-order differential operators and extreme discontinuities encountered in structural mechanics, diffusion–reaction systems, and multiphysics identification tasks. Equation (20) is denoted as $\mathcal{F}$ and is subject to the boundary condition

$$\mathcal{BC}(x;u)=0 \tag{21}$$

To avoid the numerical instability that accompanies direct high-order differentiation, we introduce auxiliary variables $\hat{v}_0(x;\boldsymbol{\theta})=u$, …, $\hat{v}_m(x;\boldsymbol{\theta})=u^{(m)}$, thereby recasting (20) as a coupled first-order system. The regularity of each derivative, exposed by this reduced-order re-parameterization, guides an order-aware domain decomposition: For every $\hat{v}_r(x;\boldsymbol{\theta})$ we estimate its jump magnitude. If $\hat{v}_r(x;\boldsymbol{\theta})$ is sufficiently smooth, a single global sub-network suffices. Otherwise, we partition the computational domain at the detected discontinuities $x_i^{\{j\}}$ and assign an independent local sub-network $\hat{v}_r^j(x;\boldsymbol{\theta})$ to each sub-interval $\Omega_i^{\{j\}}=[x_i^{\{j-1\}},x_i^{\{j\}})$.

The composite approximation reads $\hat{v}_r(x;\boldsymbol{\theta})=\sum_j\chi_{\Omega_r^{\{j\}}}(x)\cdot\hat{v}_r^{\{j\}}(x;\boldsymbol{\theta})$, where $\chi_{\Omega_r^{\{j\}}}$ is the indicator of $\Omega_r^{\{j\}}$. By confining steep gradients to small subnetworks while keeping smooth regions lightweight, DR-PINNs achieve both expressiveness and computational efficiency. All BCs can be implemented through specific network hard constraints instead of using regularization. For example, the clamped condition at $x=0$ and L can be imposed by redefining the raw network output as $\hat{v}_2(x;\boldsymbol{\theta})\cdot x\cdot(x-L)$. Therefore, the loss function of DR-PINNs is composed only of $\mathcal{L}_{\mathcal{F}_r}$, and its

form is

$$\begin{cases} \mathcal{L}_{\mathcal{F}_0}^{DR} = \frac{1}{N_{\mathcal{F}_0}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_0}} \left| \hat{v}_0 - v_0 \right|^2 \\ \mathcal{L}_{\mathcal{F}_1}^{DR} = \frac{1}{N_{\mathcal{F}_1}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_1}} \left| \frac{d\hat{v}_0}{dx} - \hat{v}_1 \right|^2 \\ \vdots \\ \mathcal{L}_{\mathcal{F}_r}^{DR} = \frac{1}{N_{\mathcal{F}_r}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_r}} \left| \frac{d\hat{v}_{r-1}}{dx} - \sum\limits_{j=1}^{n+1} \hat{v}_r^{\{j\}} \cdot \chi_{\Omega_r^{\{j\}}} \right|^2 \\ \mathcal{L}_{\mathcal{F}_{r+1}}^{DR} = \frac{1}{N_{\mathcal{F}_{r+1}}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_{r+1}}} \left| \frac{\sum_{j=1}^{n+1} d\hat{v}_r^{\{j\}} \cdot \chi_{\Omega_r^{\{j\}}}}{dx} - \hat{v}_{r+1} \right|^2 \\ \vdots \\ \mathcal{L}_{\mathcal{F}_k}^{DR} = \frac{1}{N_{\mathcal{F}_k}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_k}} \left| \frac{d\hat{v}_{k-1}}{dx} - \hat{v}_k \right|^2 \\ \mathcal{L}_{\mathcal{F}_{k+1}}^{DR} = \frac{1}{N_{\mathcal{F}_{k+1}}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_{k+1}}} \left| \mathcal{F}(x; \hat{v}_0, \cdots, \hat{v}_k; \hat{\boldsymbol{\lambda}}) \right|^2 \end{cases} \tag{22}$$

where $w_{\mathcal{F}r}$ and $\mathcal{T}_{\mathcal{F}r}$ represent the weights and sets of sampling points for the different equations and

$$\begin{cases} \mathcal{L}_{\mathcal{F}_0}^{DR} = \frac{1}{N_{\mathcal{F}_0}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_0}} \left| \hat{v}_0 - v_0 \right|^2 \\ \mathcal{L}_{\mathcal{F}_1}^{DR} = \frac{1}{N_{\mathcal{F}_1}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_1}} \left| \frac{d\hat{v}_0}{dx} - \hat{v}_1 \right|^2 \\ \vdots \\ \mathcal{L}_{\mathcal{F}_r}^{DR} = \frac{1}{N_{\mathcal{F}_r}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_r}} \left| \frac{d\hat{v}_{r-1}}{dx} - \sum\limits_{j=1}^{n+1} \hat{v}_r^{\{j\}} \cdot \chi_{\Omega_r^{\{j\}}} \right|^2 \\ \mathcal{L}_{\mathcal{F}_{r+1}}^{DR} = \frac{1}{N_{\mathcal{F}_{r+1}}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_{r+1}}} \left| \frac{\sum_{j=1}^{n+1} d\hat{v}_r^{\{j\}} \cdot \chi_{\Omega_r^{\{j\}}}}{dx} - \hat{v}_{r+1} \right|^2 \\ \vdots \\ \mathcal{L}_{\mathcal{F}_k}^{DR} = \frac{1}{N_{\mathcal{F}_k}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_k}} \left| \frac{d\hat{v}_{k-1}}{dx} - \hat{v}_k \right|^2 \\ \mathcal{L}_{\mathcal{F}_{k+1}}^{DR} = \frac{1}{N_{\mathcal{F}_{k+1}}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_{k+1}}} \left| \mathcal{F}(x; \hat{v}_0, \cdots, \hat{v}_k; \hat{\boldsymbol{\lambda}}) \right|^2 \end{cases} \tag{23}$$

v_0 denotes the labeled data, $\mathcal{T}_{\mathcal{F}_r}$ denotes the set of training points, and $N_{\mathcal{F}_r}$ represents the number of sampling points.

Figure 9 encapsulates the entire DR-PINNs workflow. Figure 9a visualizes the first step of the workflow: an "order map" that scans the target derivatives from left to right and classifies their spatial regularity. The map reveals that the primary field and the second-order derivative remain globally smooth, whereas the first-order derivative develops finite jumps and the third-order derivative is punctuated by needle-like spikes. Guided by this diagnosis, DR-PINNs either keep the domain intact or insert cuts at the detected jump locations. Each resulting sub-interval is paired with an independent sub-network, and triangular symbols at the interfaces indicate the hard constraints—continuity or prescribed jump—that will later be imposed exactly. Figure 9b follows the information as it enters the computational graph. The left column shows typical subnet configurations: a single network is sufficient when a field is smooth, whereas a quantity with two jumps is serviced by a trio of parallel subnetworks. These subnetworks feed into the reduced-order chain down the middle of the panel, where auxiliary variables convert the original m-th-order problem into a coupled first-order system. Blue arrows mark the normal forward- and back-propagation of gradients, while the red path denotes the information-distillation filter that suppresses ill-conditioned gradients before they contaminate higher-order channels. On the right, physics residuals and data misfits are gathered into the total loss, which then drives a standard gradient-descent update loop.

Interface conditions hard constrain handling method

Traditional PINNs variants approximate interface conditions by adding penalty terms to the loss, an approach that forces a dense sampling of collocation points close to discontinuities and often leads to ill-conditioned optimization. Building on the BC-embedded technique of M-PINNs, an interface conditions hard constrain handling method between subnetworks is employed here, ensuring that DR-PINNs have the ability to capture extreme discontinuities. Let three adjacent sub-domains $\Omega_i^{\{j-1\}}$, $\Omega_i^{\{j\}}$ and $\Omega_i^{\{j+1\}}$ meet at $x = x_i^{\{j-1\}}, x_i^{\{j\}}$. Denote by $\hat{v}_i^{\{j-1\}}(x;\boldsymbol{\theta})$, $\hat{v}_i^{\{j\}}(x;\boldsymbol{\theta})$ and $\hat{v}_i^{\{j+1\}}(x;\boldsymbol{\theta})$ the raw outputs of their respective subnetworks. We construct a corrective term using a minimal set of Lagrange basis functions $\{\ell_k(x)\}$ supported on a narrow stencil around $x = x_i^{\{j-1\}}, x_i^{\{j\}}$ and define the constrained outputs

$$\begin{cases} \tilde{v}_i^{\{j-1\}}(x) = v_i^{\{j\}}(x) \cdot (x - x_i^{\{j-1\}}) + b_{j-1} \\ \tilde{v}_i^{\{j\}}(x) = v_i^{\{j\}}(x) \cdot (x - x_i^{\{j-1\}}) \cdot (x - x_i^{\{j\}}) + \sum\limits_{k}^{j-1,j} b_k \ell_k(x) \\ \tilde{v}_i^{\{j+1\}}(x) = v_i^{\{j+1\}}(x) \cdot (x - x_i^{\{j\}}) + b_j \end{cases} \tag{24}$$

Here b_k are trainable coefficients that are updated through back-propagation, whereas $\ell_k(x)$ are predefined Lagrange basis functions constructed with the interpolation nodes $x = x_i^{\{j-1\}}, x_i^{\{j\}}$ on the two sides, defined as

$$\begin{cases} \ell_{j-1}(x) = \frac{x - x_i^{\{j\}}}{x_i^{\{j-1\}} - x_i^{\{j\}}} \\ \ell_j(x) = \frac{x - x_i^{\{j-1\}}}{x_i^{\{j\}} - x_i^{\{j-1\}}} \end{cases} \tag{25}$$

the shape of $\ell_k(x)$ is fixed once these nodes are chosen and therefore does not contribute additional parameters to the optimization. Because the corrective term is embedded algebraically in the forward pass, the prescribed continuity condition is satisfied a priori. Moreover, after reformulating the original m-th-order equation as a coupled first-order system, the same algebraic enforcement can be applied uniformly to every auxiliary variable—hence to subnetworks of any derivative order. Two favorable properties follow: since the constraint is embedded, collocation points need not be clustered near $x_i^{\{j\}}$; a uniform or Latin-hypercube sampling suffices. The hard constraint removes the extremely large residual gradients that typically plague penalty-based formulations, yielding faster and more stable convergence.

As long as b_k is replaced with networks that take different inputs, the method can handle problems in different dimensions.

Suppression function for information distillation and blocking ill-conditioned information propagation

The ill-posedness in inverse problems has been a highly challenging issue. When unknown parameters exhibit spatial distribution characteristics or appear within differential operators, it presents substantial challenges. Without loss of generality, consider the special form of Eq. (20) below

$$a_0\lambda_0(x)\frac{d^m u(x)}{dx^m} + \cdots + a_r \frac{d^{r_2}}{dx^{r_2}}(\hat{\lambda}_r(x)\frac{d^{m-r_1}u}{dx^{m-r_1}}) + \cdots + a_m\lambda_m(x)u(x) - q(x) = 0 \tag{26}$$

Where $r_1 + r_2 = r$ and $\hat{\lambda}_r(x)$ is the unknown spatially distributed parameter. a_r, $\lambda_0(x)$, …, $\lambda_{r-1}(x)$, $\lambda_{r+1}(x)$, …, $\lambda_m(x)$ are known

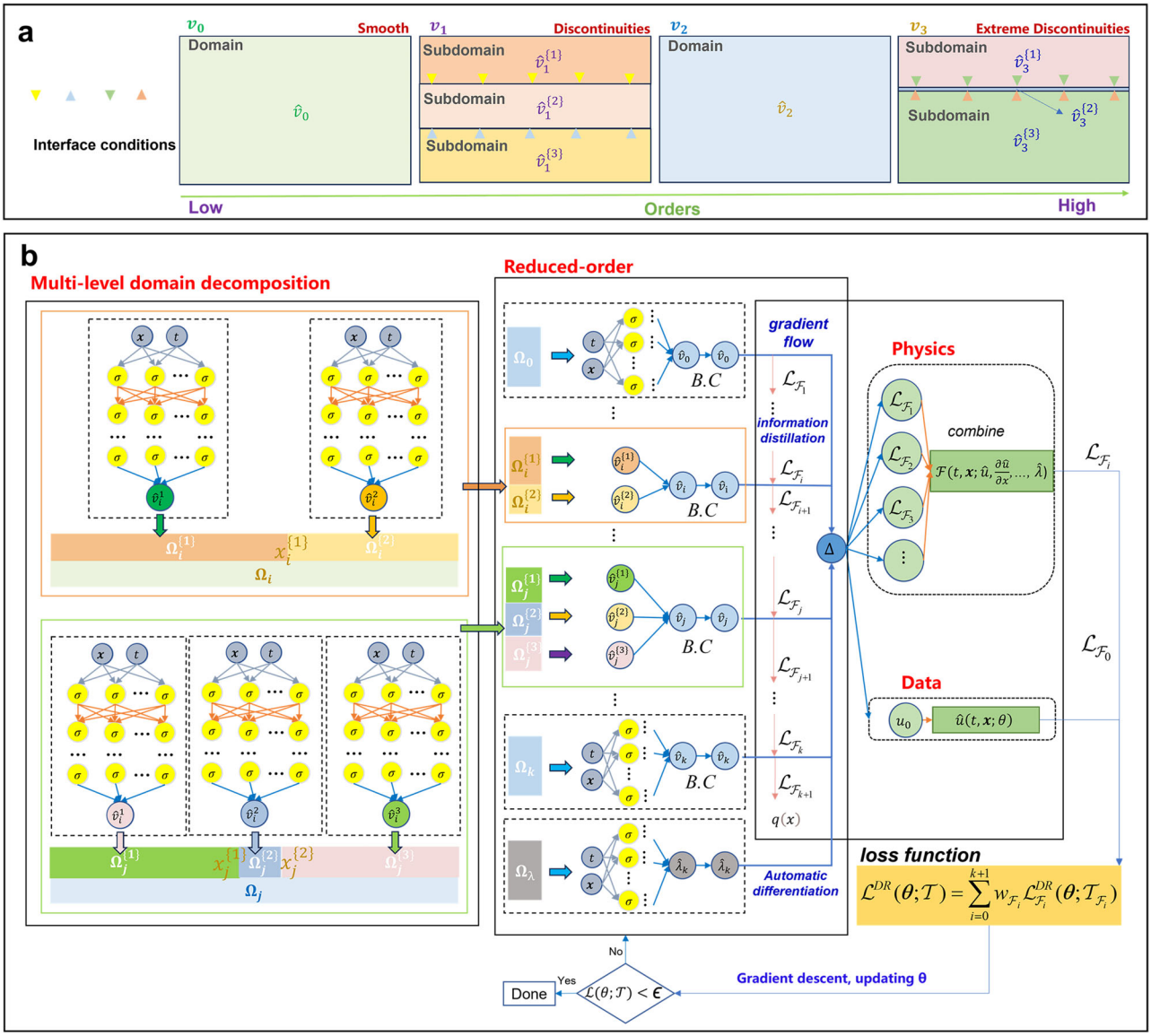


Fig. 9 | DR-PINNs (multi-level domain decomposition and Reduced-order physics-informed neural networks) concept for high-order problems with extreme discontinuities. a Schematic illustrating those physical quantities of different derivative orders experience distinct discontinuity patterns. **b** DR-PINNs architecture, in which order-specific subnetworks are assigned to each interval according to the discontinuity characteristics of the corresponding physical quantity.

parameters. Denote

$$K(x) = \hat{\lambda}_r(x)\frac{d^{m-r_1}u}{dx^{m-r_1}} \tag{27}$$

$\hat{v}_i$, $\hat{K}$ and $\hat{\lambda}$ were used to approximate $d^i u/dx^i$, $K(x)$ and $\lambda(x)$, $i = 1, \ldots, k$. The loss function is defined as

$$\begin{cases} \mathcal{L}_{\mathcal{F}_i}^{DR} = \frac{1}{N_{\mathcal{F}_i}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_i}} w_{\mathcal{F}_i} \left| \frac{d\hat{v}_{i-1}}{dx} - \hat{v}_i \right|^2 (i = 1, \ldots, k) \\ \mathcal{L}_{\mathcal{F}_{k+1}}^{DR} = \frac{1}{N_{\mathcal{F}_{k+1}}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_{k+1}}} w_{\mathcal{F}_{k+1}} \left| \mathcal{F}(x; \hat{v}_0, \cdots, \hat{v}_k; \hat{\lambda}(x); \frac{d^{r_2}\hat{K}(x)}{dx^{r_2}}) \right|^2 \\ \mathcal{L}_{\mathcal{F}_{k+2}}^{DR} = \frac{1}{N_{\mathcal{F}_{k+2}}} \sum\limits_{x \in \mathcal{T}_{\mathcal{F}_{k+2}}} w_{\mathcal{F}_{k+2}} \left| \hat{K}(x) - \hat{\lambda}_r(x)\hat{v}_{m-r1} \right|^2 \end{cases} \tag{28}$$

when $\hat{v}_{m-r_1} \neq 0$ and $\mathcal{L}_{\mathcal{F}_{k+2}}^{DR}$ is sufficiently small, the principle of identifying $\lambda(x)$ is

$$\hat{\lambda}(x) = \frac{\hat{K}(x)}{\hat{v}_{m-r_1}} \tag{29}$$

In DR-PINNs, there are two different representations of the same physical field: the gradient information output from the upper-layer network, and the lower-level network output. The former is a soft constraint, i.e., $d\hat{v}_{m-r_1-1}/dx$, while the latter is a hard constraint, i.e., $\hat{v}_{m-r_1}$. Since $\hat{v}_{m-r_1}$ can be strictly be zero, the ratio of $\hat{K}(x)$ to $\hat{v}_{m-r_1}$ avoids numerical instability. However, $d\hat{v}_{m-r_1-1}/dx$ can only approach zero at the boundaries, leading to pronounced numerical instability in the ratio of $\hat{K}(x)$ to $d\hat{v}_{m-r_1-1}/dx$.

Although, as shown in Eqs. (28) and (29), the output of $\hat{\lambda}(x)$ is only related to the hard constraint $\hat{v}_{m-r_1}$, the information of the soft-constrained

$d\hat{v}_{m-r_1-1}/dx$ is also transmitted to $\hat{v}_{m-r_1}$ through $\mathcal{L}^{DR}_{\mathcal{F}m-r_1}$, and then to $\hat{\lambda}(x)$ through $\mathcal{L}^{DR}_{\mathcal{F}_{k+2}}$. The pathological characteristics of soft constraints directly affect the ratio of $\hat{K}(x)$ to $d\hat{v}_{m-r_1-1}/dx$, and affect the predicted value $\hat{\lambda}(x)$ through the information propagation path.

When $d\hat{v}_{m-r_1-1}/dx$ approaches or equals zero at an internal point x_{domain} within the computational domain, the value of Eq. (29) near x_{domain} imposes constraints on the inversion value at x_{domain}, preventing large deviations in $\hat{\lambda}(x)$. However, when $d\hat{v}_{m-r_1-1}/dx$ approaches or equals zero at a boundary point x_{BC} of the computational domain, the absence of values outside the boundary to provide constraints leads to severe numerical instability in Eq. (29) at x_{BC}.

To ameliorate the problem, $\mathcal{L}^{DR}_{\mathcal{F}_{m-r_1}}$ is modified as

$$\mathcal{L}^{DR}_{\mathcal{F}_{m-r_1}} := \mathcal{W}(\varepsilon) \times \mathcal{L}^{DR}_{\mathcal{F}_{m-r_1}} \tag{30}$$

where ε is the prescribed threshold value associated with the suppression range for $\hat{v}_{m-r_1}$ at boundaries, and

$$\mathcal{W}(\varepsilon) = \begin{cases} 0, & x \in [0, \varepsilon) \cup [L-\varepsilon, L] \\ 1, & x \in [\varepsilon, L-\varepsilon) \end{cases} \tag{31}$$

The purpose of setting $\mathcal{W}(\varepsilon)$ is to suppress the propagation of pathological information during the parameter update process and to protect the information from hard BCs, thereby enhancing the quality of information. The core of DR-PINNs for addressing discontinuities and high-order differential equations with unknown spatially distributed parameters lies in correcting the boundary conditions of different physical fields during the information propagation process, thereby achieving the goal of information distillation.

The DR-PINN for solving inverse problems involving extreme discontinuities and high order parameterized differential equations is summarized in Supplementary B: Algorithm.

Data availability

The simulated datasets were generated with MATLAB and are used to verify the computational results predicted by the DR-PINNs framework. All data required to reproduce the findings are packaged together with the accompanying code.

Code availability

The source codes supporting this research are available on GitHub: https://github.com/PMingsheng/DR-PINNs.git.

Received: 20 January 2025; Accepted: 16 July 2025;
Published online: 01 September 2025

References

1. Lippe, P., Veeling, B., Perdikaris, P., Turner, R. & Brandstetter, J. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Adv. Neural Inf. Process. Syst.* **36**, 67398–67433 (2023).
2. Chen, Y., Hosseini, B., Owhadi, H. & Stuart, A. M. Solving and learning nonlinear PDEs with Gaussian processes. *J. Comput. Phys.* **447**, 110668 (2021).
3. Brunton, S. L. & Kutz, J. N. Promising directions of machine learning for partial differential equations. *Nat. Comput. Sci.* **4**, 483–494 (2024).
4. Givoli, D. A tutorial on the adjoint method for inverse problems. *Comput. Methods Appl. Mech. Eng.* **380**, 113810 (2021).
5. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
6. Lake, B. M., Salakhutdinov, R. & Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science* **350**, 1332–1338 (2015).
7. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
8. Liao, Y., Tang, H. & Xie, L. A deep modal model for reconstructing the VIV response of a flexible cylinder with sparse sensing data. *Ocean Eng.* **326**, 120871 (2025).
9. Yu, J., Lu, L., Meng, X. & Karniadakis, G. E. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Comput. Methods Appl. Mech. Eng.* **393**, 114823 (2022).
10. Hu, H., Qi, L. & Chao, X. Physics-informed Neural Networks (PINN) for computational solid mechanics: Numerical frameworks and applications. *Thin-Walled Struct.* **205**, 112495 (2024).
11. Lu, L., Meng, X., Mao, Z. & Karniadakis, G. E. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* **63**, 208–228 (2021).
12. Kaewnuratchadasorn, C., Wang, J. & Kim, C. W. Physics-informed neural operator solver and super-resolution for solid mechanics. *Comput.-Aided Civ. Infrastruct. Eng.* **39**, 3435–3451 (2024).
13. Hou, J., Li, Y. & Ying, S. Enhancing PINNs for solving PDEs via adaptive collocation point movement and adaptive loss weighting. *Nonlinear Dyn.* **111**, 15233–15261 (2023).
14. Ferrer-Sánchez, A., Martín-Guerrero, J. D., de Austri-Bazan, R. R., Torres-Forné, A. & Font, J. A. Gradient-annihilated PINNs for solving Riemann problems: application to relativistic hydrodynamics. *Comput. Methods Appl. Mech. Eng.* **424**, 116906 (2024).
15. Söyleyici, C. & Ünver, H. Ö A Physics-Informed Deep Neural Network based beam vibration framework for simulation and parameter identification. *Eng. Appl. Artif. Intell.* **141**, 109804 (2025).
16. Wang, S., Yu, X. & Perdikaris, P. When and why PINNs fail to train: A neural tangent kernel perspective. *J. Comput. Phys.* **449**, 110768 (2022).
17. Kharazmi, E., Zhang, Z. & Karniadakis, G. E. M. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Comput. Methods Appl. Mech. Eng.* **374**, 113547 (2021).
18. Chang, Z., Li, K., Zou, X. & Xiang, X. High order deep neural network for solving high frequency partial differential equations. *Commun. Comput. Phys.* **31**, 370–397 (2022).
19. Lou, Q., Meng, X. & Karniadakis, G. E. Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann-BGK formulation. *J. Comput. Phys.* **447**, 110676 (2021).
20. Luong, K. A., Le-Duc, T. & Lee, J. Deep reduced-order least-square method—A parallel neural network structure for solving beam problems. *Thin-Walled Struct.* **191**, 111044 (2023).
21. He, W., Li, J., Kong, X. & Deng, L. Multi-level physics informed deep learning for solving partial differential equations in computational structural mechanics. *Commun. Eng.* **3**, 151 (2024).
22. Jagtap, A. D. & Karniadakis, G. E. Extended Physics-informed Neural Networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Commun. Computational Phys.* **28**, 2002–2041 (2021).
23. 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. Methods Appl. Mech. Eng.* **413**, 116120 (2023).
24. Sarma, A. K., Roy, S., Annavarapu, C., Roy, P. & Jagannathan, S. Interface PINNs (I-PINNs): A physics-informed neural networks framework for interface problems. *Comput. Methods Appl. Mech. Eng.* **429**, 117135 (2024).
25. Hu, W.-F., Lin, T.-S. & Lai, M.-C. A discontinuity capturing shallow neural network for elliptic interface problems. *J. Comput. Phys.* **469**, 111576 (2022).
26. Liu, L. et al. Discontinuity Computing Using Physics-Informed Neural Networks. *J. Sci. Comput.* **98**, 22 (2023).
27. Peng, M., Tang, H. & Kou, Y. Adversarial and self-adaptive domain decomposition physics-informed neural networks for high-order differential equations with discontinuities. *Eng. Appl. Artif. Intell.* **145**, 110156 (2025).

28. Hu, Z., Jagtap, A. D., Karniadakis, G. E. & Kawaguchi, K. Augmented Physics-Informed Neural Networks (APINNs): A gating network-based soft domain decomposition methodology. *Eng. Appl. Artif. Intell.* **126**, 107183 (2023).
29. Miao, Z. & Chen, Y. VC-PINN: Variable coefficient physics-informed neural network for forward and inverse problems of PDEs with variable coefficient. *Phys. D: Nonlinear Phenom.* **456**, 133945 (2023).
30. Bhowmick, S. & Nagarajaiah, S. Physics-guided identification of Euler–Bernoulli beam PDE model from full-field displacement response with SimultaNeous basis function Approximation and Parameter Estimation (SNAPE). *Eng. Struct.* **289**, 116231 (2023).
31. Luong K. A., Le-Duc T., Lee S., Lee J. A novel normalized reduced-order physics-informed neural network for solving inverse problems. *Eng. Comput.*, 1–20 (2024).
32. Wang, H., Lu, L., Song, S. & Huang, G. Learning specialized activation functions for physics-informed neural networks. *Commun. Comput. Phys.* **34**, 869–906 (2023).
33. Cella, N. & Bedon, C. Numerical modelling of global/local mechanisms and sensitivity analysis for the seismic vulnerability assessment of glass curtain walls. *Eng. Struct.* **319**, 118859 (2024).
34. Ren, R., Ma, X., Yue, H., Yang, F. & Lu, Y. Stiffness enhancement methods for thin-walled aircraft structures: A review. *Thin-Walled Struct.* **201**, 111995 (2024).

Acknowledgements

The authors would like to acknowledge the financial support from the National Natural Science Foundation of China (Grant No. 52378184).

Author contributions

Conceptualization: Mingsheng Peng and Hesheng Tang; Investigation: Mingsheng Peng; Visualization: Mingsheng Peng; Supervision: Hesheng Tang; Writing—original draft: Mingsheng Peng; Writing—review and editing: Mingsheng Peng and Hesheng Tang.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44172-025-00476-5.

Correspondence and requests for materials should be addressed to Hesheng Tang.

Peer review information *Communications Engineering* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editors: [Damien Querlioz] and [Miranda Vinay and Rosamund Daw]. Peer reviewer reports are available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025