DOI: 10.1111/mice.13292

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Physics-informed neural operator solver and super-resolution for solid mechanics

Chawit Kaewnuratchadasorn[1] | **Jiaji Wang**[1] | **Chul-Woo Kim**[2]

[1]Department of Civil Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China

[2]Department of Civil and Earth Resources Engineering, Kyoto University, Kyoto, Japan

Correspondence
Jiaji Wang, Department of Civil Engineering, The University of Hong Kong, Pok Fu Lam, Hong Kong, China.
Email: cewang@hku.hk

Funding information
The University of Hong Kong Start-up Fund for New Staff; The University of Hong Kong Seed Funding Programme for Basic Research for New Staff; Japan Society for the Promotion of Science, Grant/Award Number: Grant-in-Aid 22H01576

Abstract

Physics-Informed Neural Networks (PINNs) have solved numerous mechanics problems by training to minimize the loss functions of governing partial differential equations (PDEs). Despite successful development of PINNs in various systems, computational efficiency and fidelity prediction have remained profound challenges. To fill such gaps, this study proposed a Physics-Informed Neural Operator Solver (PINOS) to achieve accurate and fast simulations without any required data set. The training of PINOS adopts a weak form based on the principle of least work for static simulations and a storng form for dynamic systems in solid mechanics. Results from numerical examples indicated that PINOS is capable of approximating solutions notably faster than the benchmarks of PINNs in both static an dynamic systems. The comparisons also showed that PINOS reached a convergence speed of over 20 times faster than finite element software in two-dimensional and three-dimensional static problems. Furthermore, this study examined the zero-shot super-resolution capability by developing Super-Resolution PINOS (SR-PINOS) that was trained on a coarse mesh and validated on fine mesh. The numerical results demonstrate the great performance of the model to obtain accurate solutions with a speed up, suggesting effectiveness in increasing sampling points and scaling a simulation. This study also discusses the differentiation methods of PINOS and SR-PINOS and suggests potential implementations related to forward applications for promising machine learning methods for structural designs and optimization.

1 | INTRODUCTION

Advances in computational mechanics aim at modeling the behaviors of materials to prevent failures when a body is exposed to external loads (H. Wang & Qin, 2019). Numerical methods such as finite difference and finite element (FE) methods have been developed to compute the approximate solutions governed by partial differential equations (PDEs) of systems. However, such methods are limited when applied to large-scale modeling, design optimization, and analysis. The model updating requires huge computational time to search for the real-world system parameters. Furthermore, most of the conventional software for numerical analysis relies on the central processing unit (CPU), encountering challenges in inverse problems, optimization, and efficient utilization of graphical processing unit (GPU)-based devices.

In recent years, there have seen rapid developments in artificial intelligence (AI) with the utilization of GPU-based supercomputers. Neural networks (NN) have been successfully applied in interdisciplinary fields which include, but are not limited to, vision (O'Shea and Nash, 2015), language models (Vaswani et al., 2017), robotics (Kober et al., 2013), and engineering designs (Adeli & Yeh, 1989). In these applications, numerous researchers developed state-of-the-art architectures that are trained by available data and learn to minimize prediction error through NN parameter optimization. In civil engineering field, many studies employed advanced models for the problems associated with the innovative data-driven approach for smart structures and integrated systems (Javadinasab et al., 2021; Pezeshki et al., 2023). Perez-Ramirez et al. (2019) proposed a non-linear autoregressive exogenous (NARX)-based recurrent neural network (RNN) model for accurate non-parametric structural identification. The study trained the model by the measured data sets for a high-rise building and five-story frame where the performance in accurate prediction indicates the capability of capturing the relationship of input motions and the structural responses. Hurtado et al. (2024) presented to employ convolutional variational autoencoders (CVAE) and convolutional adversarial autoencoder (CAAE) for detecting damages on supported bridges under drive-by inspection. The models were trained on data of the bridge in a healthy condition for damage assessments in the damaged condition based on the damage index and threshold.

Despite successful attempts to apply data-driven models in systems governed by mathematical equations, the generalization and intepretability of NN have been the concerns. The infusion of physics was introduced into deep learning, known as physics-informed neural networks (PINNs) (Raissi et al., 2019). According to the universal approximation theorem by Hornik et al. (1989) and physics integration, PINNs have received gradual attention for applications in methanics, modeling, and structural engineering because of the seamless integration of data and mathematical models, computational effectiveness, and the ability to search for new intrinsic variables and representations with built-in physics constraints (Karniadakis et al., 2021). The recent studies of PINNs may be categorized into two main branches. In the first branch, PINNs are applied as a surrogate model, which learns data to represent a system by NN parameters, and is generalized through physical equations. Bai et al. (2022) implemented PINNs with the least squares weighted residual for solid mechanics in two and three dimensions. Adaptivity and domain decomposition are applied to PINNs for two-dimensional static elasticity problems (Henkes et al., 2022). E. Zhang et al. (2022) applied PINNs incorporating the mesh-free method for geometry and material identifications and investigated the performance of PINNs for compressible elasticity, deformation plasticity, and incompressible hyperelasticity. PINNs were applied in the topological optimization of structures by Jeong et al. (2023), in which energy-based loss, sensitivity analysis, and sensitivity filtering were applied to construct a new optimized design. Haghighat et al. (2021) employed PINNs with isogeometric analysis for linear elasticity application and extension to nonlinear problems through the von Mises elastoplasticity example. PINNs were also applied to the three-dimensional (3D) analysis of the elasticity, hyperelasticity, and plasticity of materials (Abueidda et al., 2021, 2022). Yadav et al. (2021) developed distributed PINNs (DPINN) by taking the meshless nature of materials, which was able to avoid the curse of dimensionality, particularly for unified computational frameworks. Song et al. (2022) proposed StructGNN-E with a graph-based neural network where the structural topologies were processed by graph isomorphism network and the physics-informed paradigm was implemented for application in structural analysis. To achieve high fidelity, the frameworks requires sufficient data, and rich data sets are typically obtained from extensive experiments or generated by numerical solvers. In addition, the applicability of models may be sensitive to the portion of different types of training data sets.

Conversely, PINNs have been adopted as numerical solver in physical problems to compute the solutions to PDEs that govern the system. The Deep Ritz method was proposed by E and Yu (2017) to solve PDEs based on the nonlinear transformation of the FCNN architecture to obtain the solution and calculate the total energy and the residual connection to avoid the vanishing gradient problem in the optimization. The Deep Ritz method was applied to examine computational cost and accuracy in the Poisson equation, a setting with the Neumann boundary condition, transfer learning of relatable solutions, and the eigenvalue problems. Variational PINN (VPINN) was developed based on the Petrov–Galerkin method by formulating the variational residual for nonlinear approximation. The proper integration-by-part in VPINN helps increase the accuracy while reducing training costs as it mitigates PINNs in taking high-order derivatives (Kharazmi et al., 2019). Kharazmi et al. (2021) also developed high-order polynomials VPINN (hp-VPINN) by implementing the high-order polynomials refinement via the projection onto high-order polynomials space and the domain decomposition of the nonoverlapping function. Wandel et al. (2022) proposed to introduce Hermite spline with CNNs to solve incompressible Navier–Stokes equations. Jagtap et al. (2020) developed conservative PINN (cPINN) by having subdomain networks connected through interface conditions. Extended PINN (XPINN) or the generalized domain decomposition of PINNs was

proposed, employing the advantages of PINN and cPINN while providing flexibility of subdomain division (Jagtap & Karniadakis, 2020). Chiu et al. (2022) implemented automatic differentiation (AD) and numerical differentiation (ND) for gradient computation in the coupled-automatic–numerical differentiation based PINN (CAN-PINN). For enhancing computations, Fang (2022) applied CNN instead of FCNN and the finite volume method for hybrid PINN. Nabian et al. (2021) proposed importance sampling to achieve efficient training of PINN, compared with uniform sampling in two-dimensional problems, including isotopic elasticity, plane stress, and transient diffusion. In static elasticity, J. Wang et al. (2023) developed exact Dirichlet boundary PINN (EPINN), which replaced the PDE loss with the principle of least work, restricted the Dirichlet boundary by approximate distance function (ADF), adopted the meshless finite difference (MFD) to compute for gradient information, and successfully reported the convergence of fine-mesh simulations.

Another family of machine learning methods is called deep operator networks proposed by Lu et al. (2021) to learn nonlinear relationships between functional spaces. Recent advances in neural operators have been reported through physics-informed DeepONet (Lu et al., 2021), graph kernel neural operator (GNO) (Anandkumar et al., 2019), U-shaped neural operator (UNO) (Rahman et al., 2023), convolutional neural operator (CNO) (Raonić et al., 2023), Koopman neural operator (KNO) (Xiong et al., 2023), wavelet neural operator (WNO) (Tripura & Chakraborty, 2023), and Fourier neural operator (FNO) (Li et al., 2021) on learning PDE spaces of physical equations. Many researchers employed neural operator, mainly FNO, in data-driven models of physical systems for forward and inverse problems in a weather model (Pathak et al., 2022), structural analysis (Kaewnuratchadasorn et al., 2024), material microstructure (Rashid et al., 2022), flood inundation forecast (Sun et al., 2023), and image classification (Johnny et al., 2022). Based on the FNO, Li et al. (2023) developed the physics-informed neural operator (PINO) by introducing physical losses to FNO, which was capable of reducing the error in Navier–Stokes equations. Although the neural operator family has shown excellent performances in various problems, there has been no work on training neural operators without labeled data as a PDE solver, which can further be fined-tuned for specific systems in space mappings.

To this end, this study proposed physics-informed neural operators solver (PINOS) trained without labeled data for simulations in solid mechanics. In static elasticity problems, we adopted the energy loss functions from the principle of least work and compared the performance with the EPINN benchmark in a 1D truss under complicated load, 2D plane stress, and 3D box problems. In dynamical systems, we applied a strong form of partial differential equation or ordinary differential equation (PDE/ODE) loss, boundary, and initial loss functions to obtain the solutions in three spring masses and wave propagation systems. Our contributions are summarized as follows:

1. This is the first work in neural operators without labeled data contexts and examines the performances of the strong and weak forms of systematic equations for both static and dynamic systems in solid mechanics.
2. In static problems, PINOS exceeded EPINN performance for 474× and 36× in 1D and 2D problems. PINOS also achieved 38× and 22× speed up in 2D and 3D problems compared to ABAQUS as a reference solution.
3. In dynamic problems, PINOS achieved 10× and 3.4× speed up in 1D and 2D problems compared to PINN, and super-resolution PINOS (SR-PINOS) achieved high performance in dynamics systems when compared to the analytical solutions.
4. This work examined zero-shot super-resolution of the solution in all problems, reporting that SR-PINOS was able to solve on large mesh sizes fast and provide accurate solutions in fine mesh sizes.

In this paper, Section 2 describes PINOS in FNO architectures and physics losses for solid mechanics problems. Section 3 illustrates the performance of PINOS with weak form loss in static elasticity problems, while Section 4 reports that with a strong form in the dynamic problems. Section 5 provides comparisons of models and discusses the potential and limitations of PINOS and SR-PINOS. Lastly, Section 6 summarizes the contributions of our work quantitatively and discusses the applications of PINOS in field mapping without data in solid mechanics.

2 | PHYSICS-INFORMED NEURAL OPERATOR FOR SOLID MECHANICS

This section provides explanations of the neural operator and zero-shot super-resolution applications in solid mechanics. Then, the physics-informed losses in static and dynamics systems of solid mechanics are described.

2.1 | Neural operators

As Hornik et al. (1989) established the universal approximation of NN, Kovachki et al. (2023) proposed a generalization of NN that learns operators called neural operator as $\mathcal{G}_\theta$, formulated as Equation (1)

$$\mathcal{G}_\theta := \mathcal{Q} \circ (\mathcal{W}_L + \mathcal{K}_L) \circ \cdots \circ \alpha (\mathcal{W}_1 + \mathcal{K}_1) \circ \mathcal{P} \quad (1)$$

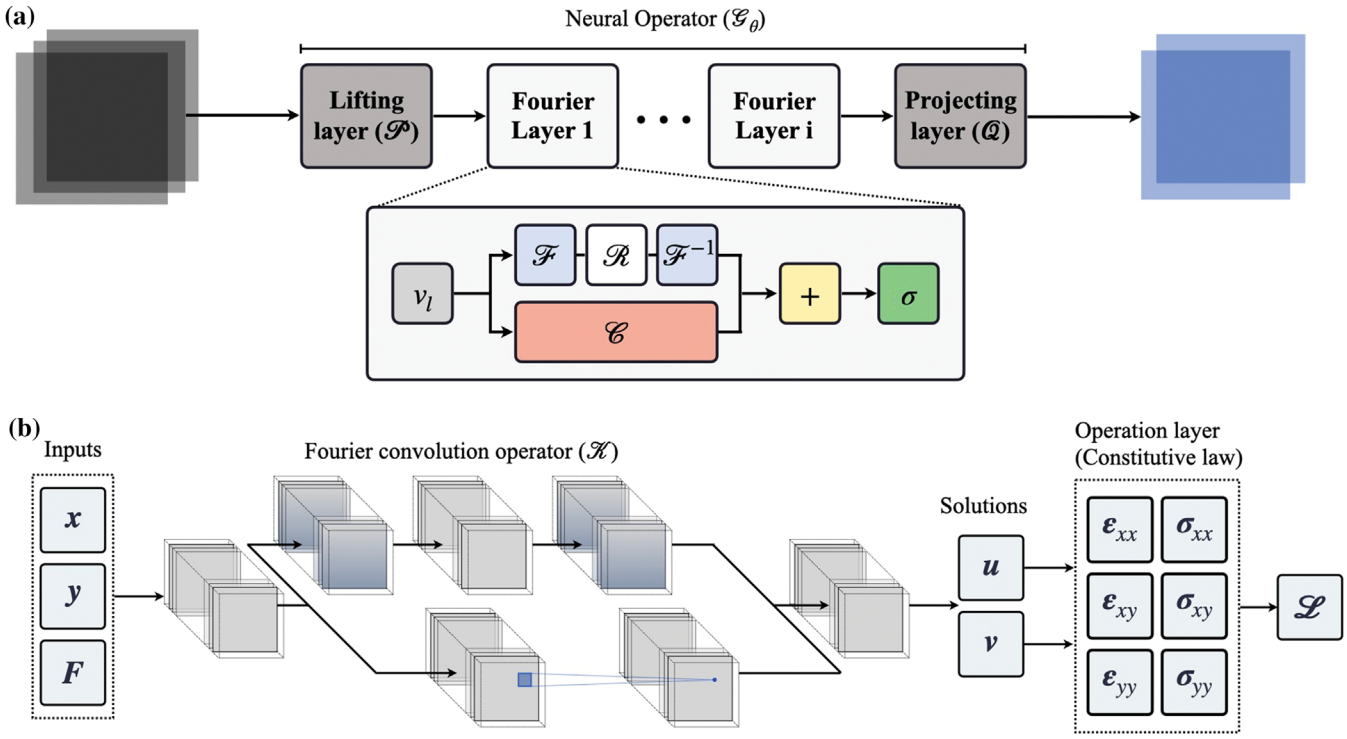


FIGURE 1 Illustration of Physics-Informed Neural Operator Solver (PINOS) architecture for solid mechanics. (a) Schematic of Fourier Neural Operator ($\mathcal{G}_\theta$). (b) A simple PINOS with one Fourier layer. The inputs contain coordinate and initial parameters. The inputs are lifted, passed through the Fourier convolution operator, and projected to the solution output. The operational layer calculates finite-difference derivatives and physical parameters to obtain losses ($\mathcal{L}$) of the govering equation, boundary conditions, and initial conditions.

where $\mathcal{P}$ and $\mathcal{Q}$ are functions parameterized by neural networks. α is a non-linear activation function. In FNO, $\mathcal{K}$ is a Fourier convolution operator.

$$(\mathcal{K}(a;\phi)v_t)(x) = \mathcal{F}^{-1}\left(\left(\mathcal{R}\left(\kappa_\phi\right)\cdot\mathcal{F}\left(v_t\right)\right)\right)(x) \qquad (2)$$

where x are coordinates in the input domains; $\mathcal{F}$ denotes Fourier transform of a function f mapping $D \to \mathbb{R}^n$ for $j = 1, \ldots, n$; $\mathcal{F}^{-1}$ is the inverse Fourier transform; and κ_ϕ is parameterized in Fourier space. The formulation was based on the convolution theorem of the kernel integral operator by Fourier and inverse Fourier transforms in Equations (3) and (4).

$$(\mathcal{F}f)_j(k) = \int_D f_j(x)\, \mathrm{e}^{-2i\pi\langle x,k\rangle} dx, \qquad (3)$$

$$(\mathcal{F}^{-1}f)_j(x) = \int_D f_j(k)\, \mathrm{e}^{2i\pi\langle x,k\rangle} dk, \qquad (4)$$

PINO was proposed by using the architecture of the FNO with data loss and the additional constraints of physics law in the loss (Li et al., 2023). Therefore, in the context of our paper, PINOS is training FNO by only physics losses, which will be explained in the following sections.

PINOS consists of six layers of computational operation as shown in Figure 1. The first layer is the input layer, which consists of input variable fields such as coordinates (x, y, z) and/or time (t). Next, the lifting layer ($\mathcal{P}$) transforms the coordinate fields into the same dimension as Fourier layers. The Fourier layer ($\mathcal{K}$ and $\mathcal{W}$), also called the Fourier block, contains Fourier transformation, linear, and inverse Fourier transformation as depicted in Equation (2) with separated convolutional parts. Two operational paths in the Fourier layers are summed, and a non-linear activation function (α) is applied before passing to the next layer. In our study, the number of Fourier layers needed depends on the complexity of the problem. It is noted that activation functions are applied to all Fourier layers except the last latyer. After blocks of Fourier layers, the projecting layer ($\mathcal{Q}$) converts to the dimension the same as the input. This means the output layer, which is the solution variable field, has the same dimension as the input variable fields. The inputs and outputs can be 1D, 2D, or 3D coordinates, which will be associated with the calculation of derivatives by finite-difference differentiation. The last layer is the operational layer, which calculates derivatives and loss functions. The operational layer in PINOS can be specific to each system, and the elastic linearity is considered in this paper for static simulations. The gradient descent optimization algorithms of parameters in PINOS adopt Adam optimizer in learning of loss functions explained in

Section 2.3. In this study, we apply PINOS for 1D, 2D, and 3D coordinates in static problems and 1D temporal and 2D spatial-temporal fields in dynamic problems, as reported in the next section.

2.2 | Zero-shot super-resolution applications

The zero-shot super-resolution refers to an ability to enhance the resolution of the solution fields without training with the pair of low-resolution inputs and high-resolution outputs. The ability also involves learning with larger mesh sizes and predicting the solution in smaller mesh sizes. As per Li et al. (2021), FNO showed the zero-shot super-resolution performance in Navier–Stokes equations with data training. It was reported that the FNO was trained with $64 \times 64 \times 20$ resolution and capable of approximating the outputs of $256 \times 256 \times 80$ resolution on the same spatial and time frames.

In solid mechanics, zero-shot super-resolution is a challenging task that would reduce computational cost drastically. The conventional finite-element methods (FEM) encountered exponential costs in solving solutions on a finer mesh size. The accuracy trade-off is not favorable in the microanalysis of a structural member. Similarly, the task is demanding in physics-informed machine learning (PIML). Since FNO showed the performance in fluid mechanics, this work also utilizes and investigates the performances in both static and dynamic problems for computational cost and accuracy.

It is important to emphasize that we developed SR-PINOS and investigated two features in terms of computations, which are relative L_2 error and convergence time.

2.3 | Loss functions in solid mechanics

In FEM analysis, governing equations are classified into two types: strong form and weak form. While a strong form is represented by the PDE of a specific system, a weak form is derived from the strong form by integration or a variational approach. In this work, the principle of least work is employed as a weak-form governing equation in static elasticity problems, and the strong-form governing equations are utilized for dynamics problems. This section will introduce physics loss in PINOS of this paper.

The FNOs are trained by loss functions for specific solid mechanics problems. In our study, a training loss ($\mathcal{L}$) consists of three components as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{system} + \beta \mathcal{L}_{BC} + \lambda \mathcal{L}_{IC}, \tag{5}$$

where $\mathcal{L}_{system}$ is restricted by governing equations of the system; $\mathcal{L}_{BC}$ denotes the boundary condition loss; $\mathcal{L}_{IC}$ is the initial condition loss, which is added in the dynamical problem; and α, β, λ are the weights of each loss. To further optimize PINOS, it may be more efficient to use loss-balancing techniques such as SoftAdapt (Heydari et al., 2019) and ReLoBRaLo or the multiobjective loss balancing (Bischof & Kraus, 2021) that will adjust the weights of loss during training.

2.3.1 | Weak form: Principle of least work with the Dirichlet boundary condition

The principle of least work states that the solution to the static mechanics problem is such that the total energy, which is defined as the strain energy minus the external work done by an external force at force boundary and body force, reaches a minimum (Office, 1923). In accordance with the theorem, the principle of least work loss function is formulated as Equation (6).

$$\mathcal{L}_{system}(\mathbf{u}; \mathbf{f}, \mathbf{t}) = \frac{1}{2} \int_{\Omega} \sigma(\mathbf{u}) \cdot \varepsilon(\mathbf{u}) d\Omega - \left(\int_{\Omega} \mathbf{u} \cdot \mathbf{f} d\Omega + \int_{\Delta\Omega_N} \mathbf{u} \cdot \bar{\mathbf{t}} d\Delta\Omega_N \right), \tag{6}$$

where $\mathbf{u}$ denotes the displacement solution field; σ and ε are stress and strain tensors calculated from the displacement; $\mathbf{f}$ represents the force, and $\mathbf{t}$ is the traction applied to the system. PINOS adopts the total energy of the system in the system loss function ($\mathcal{L}_{system}$). The principle of least work is applied in the machine learning system, which indicates that PINOS will optimize to find the minimum total work of the system and assures the one unique solution of a static solid mechanic problem. The reasons for adopting the principle of least work in PINOS include the following. Firstly, there is no report on using weak-form governing equations in PINOS of solid mechanics. Energy is more sensitive to the PDE loss; thus minimization of energy of the system may reduce convergence time. In a study of EPINN, J. Wang et al. (2023) also employed the principle of least work on PINN and reported converging performance and significant reduction of convergence rate compared to the original PINN with a strong form of governing equation. Many researchers also reported the difficulty in accurate derivatives affected the convergence of models. In PINOS, the derivatives will be directly calculated using finite differences, thus will be efficient in solving the solution of a system. Therefore, the theorem of least work is applied to static elasticity problems in this study.

2.3.2 | Strong form: PDE with boundary and initial conditions

In dynamic problems, PINOS adopts PDE loss for the system ($\mathcal{L}_{system}$). Unlike the static elasticity problems which may be represented by energy-governing equations and minimized, the solutions in dynamic problems are in the time domain. Many models encounter difficulty in solving the governing equations. Although the variational formulation may be derived as the energy-governing equation of a dynamic system, the current work focuses on a strong form due to directly representing the governing system.

In this study, we applied PINOS in solving ordinary differential equations (ODE) of three spring masses and wave equations. The ODE is considered a complex 1D temporal problem as three solutions from three equations need to be solved. The wave equation is considered a 2D spatial-temporal problem (x, t) which is in a second-order form. The equations will be explained specifically in the section of the problems. However, in general, the formulation is derived as follows. Let f be an ODE in Equation (7).

$$f : u_t + R(u) = 0, \tag{7}$$

where u_t and $R(u)$ are a time derivative and a function of u, respectively. The system loss of the governing equation or the ODE loss is formulated as follows:

$$\mathcal{L}_{system} = \int_D |f(u(t))| dt. \tag{8}$$

The system loss ($\mathcal{L}_{system}$) will need to converge to near zero. PINOS will use an optimizer to obtain the model parameters to obtain the solutions of a system.

2.3.3 | Boundary and initial conditions

In the context of numerical methods, boundary and initial conditions may be divided into hard constraints and soft constraints, which are represented in the mathematical formulation of a model. While the boundary conditions are enforced into the model for the hard constraints, the soft-constraint models incorporate the solver into some degree of flexibility.

We consider using soft constraints of boundary and initial conditions in our study. Similar to the system loss ($\mathcal{L}_{system}$), PINOS also learns to minimize the loss of boundary ($\mathcal{L}_{BC}$) and initial conditions ($\mathcal{L}_{IC}$). This learning is considered a soft constraint because PINOS solves the solution that the boundary will not meet the exact boundary but will be in close proximity. The minimization of the losses in PINOS will lead to the learning of a system using PIML, and the solution will converge. However, it is also important to mention the weights of the boundary and initial conditions. This study adopts the weights of the boundary and initial conditions with the accountable dimensions of the solution. The boundary and initial conditions only calculate the solution at the starting or at the boundary while the system loss takes all the solution. Therefore, to balance the weights of the boundary and initial conditions with the system loss, we add the weights according to the dimension of the solution.

2.3.4 | Validation loss function

To validate the performance of a model, we employed the validation loss function to compare the result errors between the solution from a model and the reference solution. In this paper, the reference solutions are analytical solutions in a truss under complicated load, spring-mass system, and wave propagation while the solutions from FEA using ABAQUS are used in a plane under eccentric stress and box problems. We employed the relative errors, which were defined by L. Zhang et al. (2021), including relative L_2 and H_1 errors. The error functions are expressed in Equations (9) and (10) as follows:

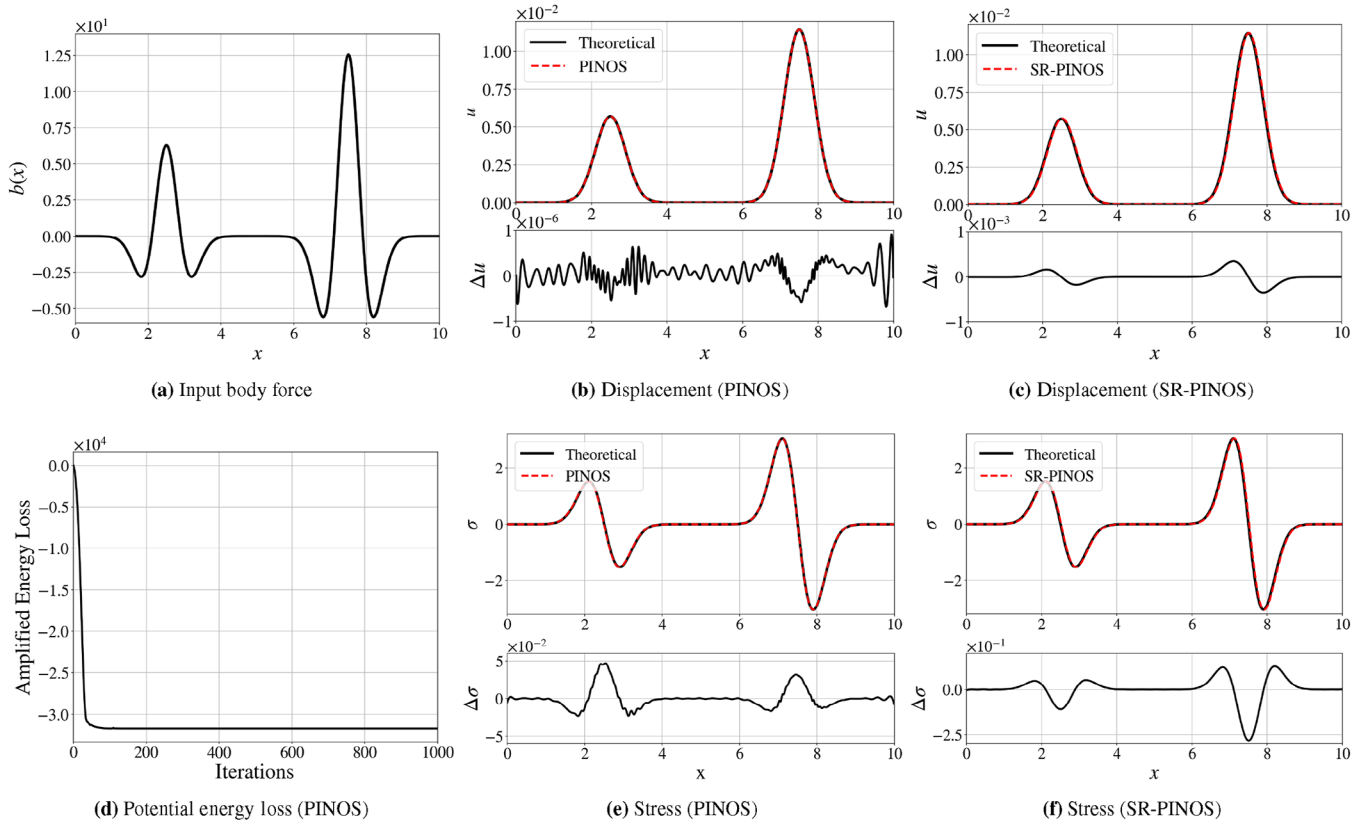
$$\|e\|_{L_2} = \frac{\sqrt{\int_\Omega (u_r - u_m)^2 dx}}{\sqrt{\int_\Omega (u_r)^2 dx}} \tag{9}$$

$$\|e\|_{H_1} = \frac{\sqrt{\int_\Omega (u_r - u_m)^2 dx + \int_\Omega (du/dx_r - du/dx_m)^2 dx}}{\sqrt{\int_\Omega (u_r)^2 dx + \int_\Omega (du/dx_r)^2 dx}} \tag{10}$$

where u_r is the reference solution from ABAQUS or analytical solution and u_m is the solution from the machine-learning model for all x in Ω domain of interest. In this study, the relaitvel L_2 error is adopted to justify the convergence of models where the solutions are converged at the time when the relative error reduces to less than 0.1. In the higher dimensions, the solutions of all directions are validated.

3 | APPLICATIONS OF PHYSICS-INFORMED NEURAL OPERATOR IN STATIC ELASTICITY PROBLEMS

The section reports the performance of PINOS in numerical solver and super-resolution tasks in three static elasticity problems, which include a deformation of a truss loaded



(a) Input body force **(b)** Displacement (PINOS) **(c)** Displacement (SR-PINOS)

(d) Potential energy loss (PINOS) **(e)** Stress (PINOS) **(f)** Stress (SR-PINOS)

FIGURE 2 The input and results of PINOS and SR-PINOS on the 1D loaded truss problem. PINOS was trained and validated with a truss with $\Delta x = 0.01$ m. SR-PINOS was trained with a truss with $\Delta x = 0.05$ m and validated on a truss with $\Delta x = 0.01$ m.

by complicated force, a deformation of a plane stressed by partial displacement boundary conditions, and a deformation of a box stressed by local displacement boundary conditions. In this study, PINOS was conducted using the Intel Xeon Platinum 8368 (Icelake/2.4 GHz 38 cores) two CPUs, and a single unit of NVIDIA A100 on the Supercomputer for Quest to Unsolved Interdisciplinary Datascience (SQUID) at Osaka University. The hyperparameters of PINOS, including the number of Fourier layers, modes, widths, and learning rate, were selected according to the complexity of a problem. This is reported separately in each subsection.

3.1 | One-dimensional truss under complicated load

The performance of 1D PINOS was investigated in the deformation of a truss under complicated loads. This study considered the problem reported by L. Zhang et al. (2021) in the hierarchical deep-learning neural networks (HiDeNN) and investigated by J. Wang et al. (2023) using EPINN on Nvidia Modulus. The problem considers a truss element or one-dimensional bar with following properties. In the setting, Young's modulus, the cross-sectional area, and the length are defined as $E = 175$, $A = 1$, and $l = 10$, respectively. The load (b) on the truss is written in Equation (11) in the horizontal (x) direction, illustrated in Figure 2a.

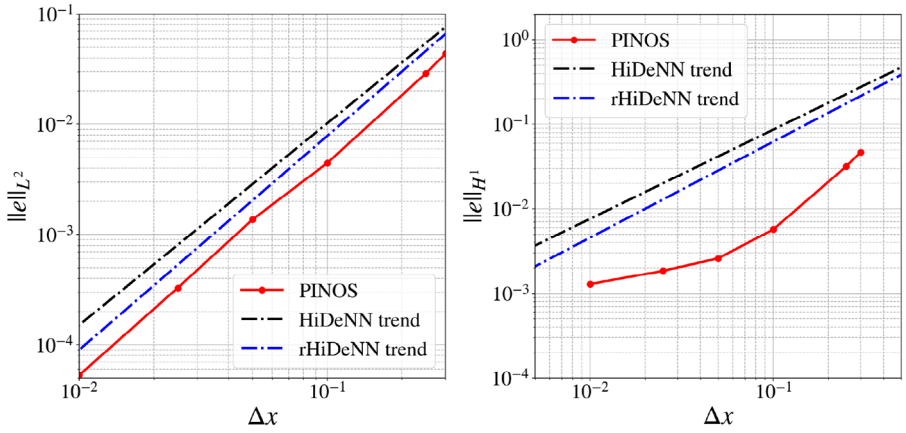
$$b(x) = -\frac{4\pi^2(x-2.5)^2 - 2\pi}{e^{\pi(x-2.5)^2}} - \frac{8\pi^2(x-7.5)^2 - 4\pi}{e^{\pi(x-7.5)^2}} \tag{11}$$

As written in energy loss (Equation 6), the potential energy loss ($\mathcal{L}_{system}$) may be written in terms of longitudinal displacement (u) for one dimension as Equation (12).

$$\mathcal{L}_{system}(u) = \frac{1}{2}\int_{\Omega} AE\left(\frac{du}{dx}\right)^2 dx - \int_{\Omega} ub(x)dx, \forall u \in U, \tag{12}$$

where Ω refers to all elements in the x direction. The analytical solution of the vertical displacement of the truss is provided in Equation (13), and the derivative (u_x) is shown in Equation (14), which will be used in calculations of the strain ($\varepsilon = u_x/L$) and the stress ($\sigma = Eu_x/L$).

$$u(x) = \frac{1}{AE}\left(e^{-\pi(x-2.5)^2} - e^{-6.25\pi}\right) + \frac{2}{AE}\left(e^{-\pi(x-7.5)^2 - e^{-56.25\pi}}\right) - \frac{e^{-6.25\pi} - e^{-56.25\pi}}{10AE}x, \tag{13}$$



(a) Relative L_2 error **(b)** Relative H_1 error

FIGURE 3 Error analysis of PINOS with different mesh sizes.

$$\frac{du}{dx} = \frac{2}{AE}\left(-\pi\, e^{\pi(x-2.5)^2}(x-2.5)\right) + \frac{4}{AE}\left(-\pi e^{-\pi(x-7.5)^2}(x-7.5)\right) - \frac{e^{-6.25\pi} - e^{-56.26\pi}}{10AE}. \tag{14}$$

The architecture of the neural operator in this problem is 1D FNO with the following hyperparameters. The number of Fourier layers is four; modes are 16; widths are 64; learning rate is 0.0005; the stop epoch is 1000. The performance of PINOS is reported in Figure 2. PINOS was trained and validated on a truss with mesh size $\Delta x = 0.01$ m of length 10. The displacement solution and the differences from the analytical solution (Equation 13) are shown in Figure 2b, and the stress and the differences from the analytical solution (Equation 13) are depicted in Figure 2e. The training rate of PINOS was 222 iterations/secod (it/s) on SQUID. The convergence was observed around epoch 200, thus PINOS took around 0.9 s to solve the 1D problem.

This study also performed the super-resolution and reported in Figure 2c,f. SR-PINOS, which was trained on a truss with mesh size $\Delta x = 0.05$, however, was validated on a truss with mesh size $\Delta x = 0.01$. It is observed that SR-PINOS was capable of providing accurate solutions, while mesh size increased by 5× for displacement and stress. The solution converged at 0.33 s or at 75 iterations with a training speed of 226 it/s. This suggests a promising application of SR-PINOS in the super-resolution task, which aims to provide justification for the model on a microscale investigation of the material deformation in solid mechanics. The time and accuracy of PINOS and SR-PINOS are compared along with other problems in Section 5.

Figure 3 shows the error analysis of PINOS. We show the convergence rate of L_2 and H_1 errors when the mesh size is reduced. A linear trend is observed in the relative L_2 error, while an exponential trend is observed in the relative H_1 error in the log–log plot between the errors and mesh size. In comparison, we extracted the trend of the HiDeNN model from L. Zhang et al. (2021). It is observed that PINOS is more accurate than HiDeNN in both L_2 and H_1 relative errors. In the HiDeNN model, the numbers of nodes were 23, 45, 89, 177, and 353 nodes. In PINOS, we considered the mesh size of $\Delta x = 0.3, 0.25, 0.1, 0.05, 0.025, 0.01$, which corresponded to 33, 40, 100, 200, 400, and 1000 nodes. The complexity of 1000 nodes is high in solid mechanics to obtain an accurate solution, and PINOS was able to obtain the solution within 0.9 s, which suggests potential application in more complex problems.

3.2 | Two-dimensional plane stress under eccentric tension

The deformation of a plane under eccentric tension aims to investigate the performance of 2D PINOS as reported in studies of PINN (Rao et al., 2020) and EPINN (J. Wang et al., 2023) without labeled data. Given a plane length and height of 1 m. The material properties include elastic modulus ($E = 10$ MPa) and Poisson ratio ($\nu = 0.2$). The boundary conditions include the fix at the bottom line and vertical tension displacement on the left half of the top line. We define horizontal displacement as u and vertical displacement as v. The potential energy of the plane is calculated according to Equation (6), where stress (σ) and strain (ϵ) are formulated as Equations (15) and (16), respectively.

$$\sigma_{xy} = \lambda \varepsilon_{zz} \delta_{xy} + 2\mu \varepsilon_{xy} \tag{15}$$

$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{16}$$

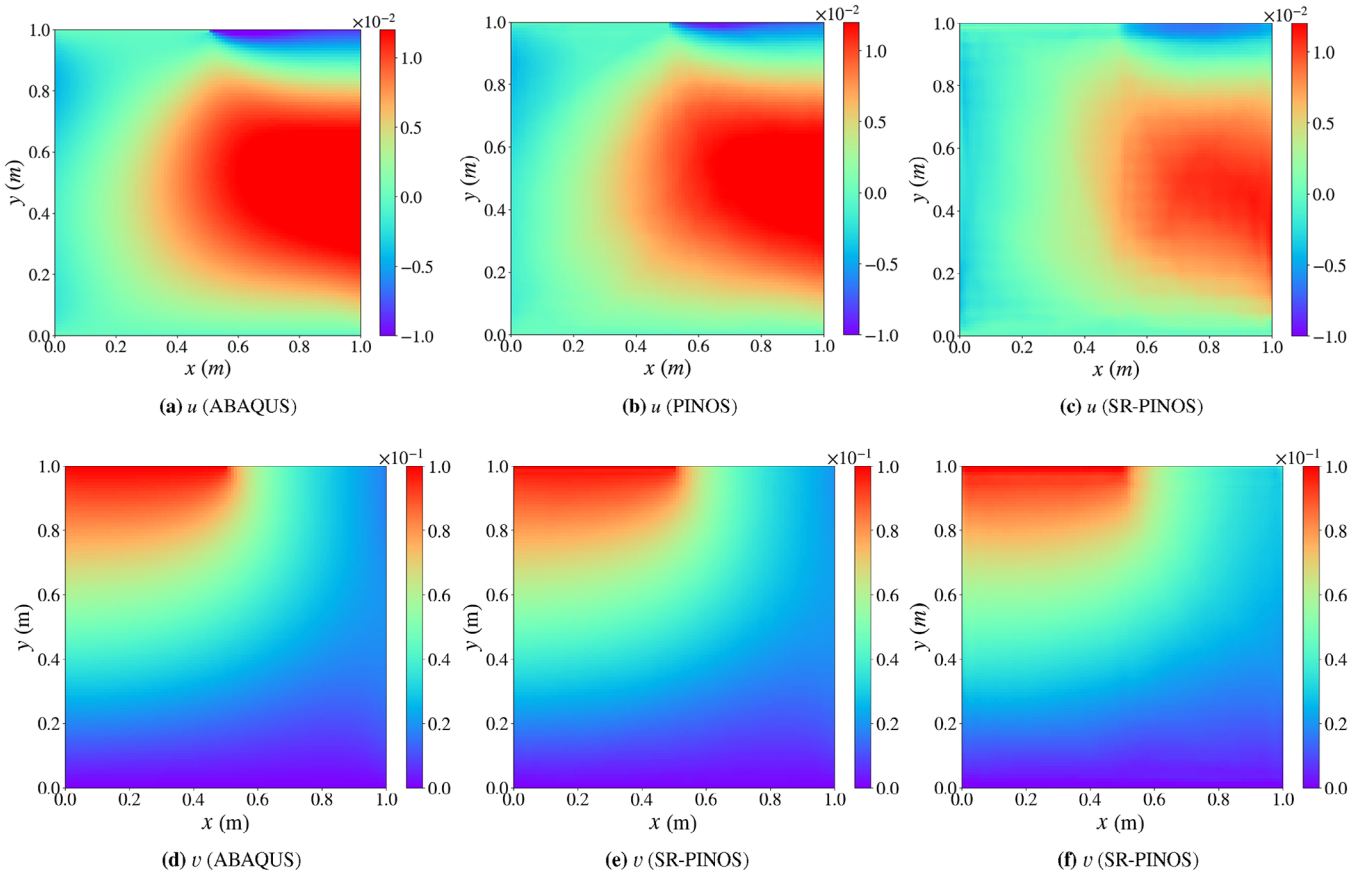


FIGURE 4 Results of PINOS and SR-PINOS on the 2D plane under eccentric loads. The horizontal and vertical displacements are denoted by u and v, respectively. The reference solution was obtained from the ABAQUS. PINOS solution was trained and obtained on the fine mesh of $\Delta x, \Delta y = 0.01$ m. SR-PINOS was trained on $\Delta x, \Delta y = 0.02$ m mesh, and the solution was obtained at a mesh size $\Delta x, \Delta y = 0.01$ m.

where x, y, z represent the axial directions and λ, μ are the Lamé parameters, formulated as in Equation (16),

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{17}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{18}$$

In this 2D problem, stress in the z-direction is not considered ($\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$). The architecture of PINOS in this setting adopted 2D FNO with 5 Fourier layers, 16 modes in two dimensions, 64 widths, a learning rate of 0.0005, and a stop epoch at 20,000. The plane is divided into mesh size $\Delta x, \Delta y = 0.01$ m. As the analytical solutions were not available, the reference solutions (the horizontal and vertical displacements) were obtained from the FE simulation in ABAQUS software. Figure 4 illustrates the solver and super-resolution performances of PINOS, compared to the ABAQUS FE simulation. PINOS was trained and validated on mesh size $\Delta x, \Delta y = 0.01$ m, while SR-PINOS was trained on mesh size $\Delta x, \Delta y = 0.02$ m but validated on $\Delta x, \Delta y = 0.01$ m mesh. The results indicate that PINOS and SR-PINOS showed great performance as observed in the comparisons. The comparison of horizontal displacements is reported in Figure 4a–c and that of vertical displacements is reported in Figure 4d–f. The relative least square (L_2) errors on horizontal and vertical solution fields are 0.091 and 0.018 from PINOS. The errors of solutions are 0.298 and 0.067 from SR-PINOS. The training rates of PINOS and SR-PINOS are 32.15 it/s and 50.98 it/s. The energy and boundary losses converged at 1000 iterations, thus the convergence times are 31.1 and 19.6 s while ABAQUS took 265 s to obtain the solution. The accuracy and computational time of PINOS and SR-PINOS are compared with the benchmark in Section 5. Therefore, the performance of PINOS and SR-PINOS suggested potential applications in the investigations of the deformation of solid mechanics.

3.3 | Three-dimensional box under eccentric tension

This section examines PINOS in solving the 3D solution fields, which refer to displacements in x, y, and z directions. A 3D elastic box is restricted to local tension in a quarter of the surface at the top, while the bottom surface

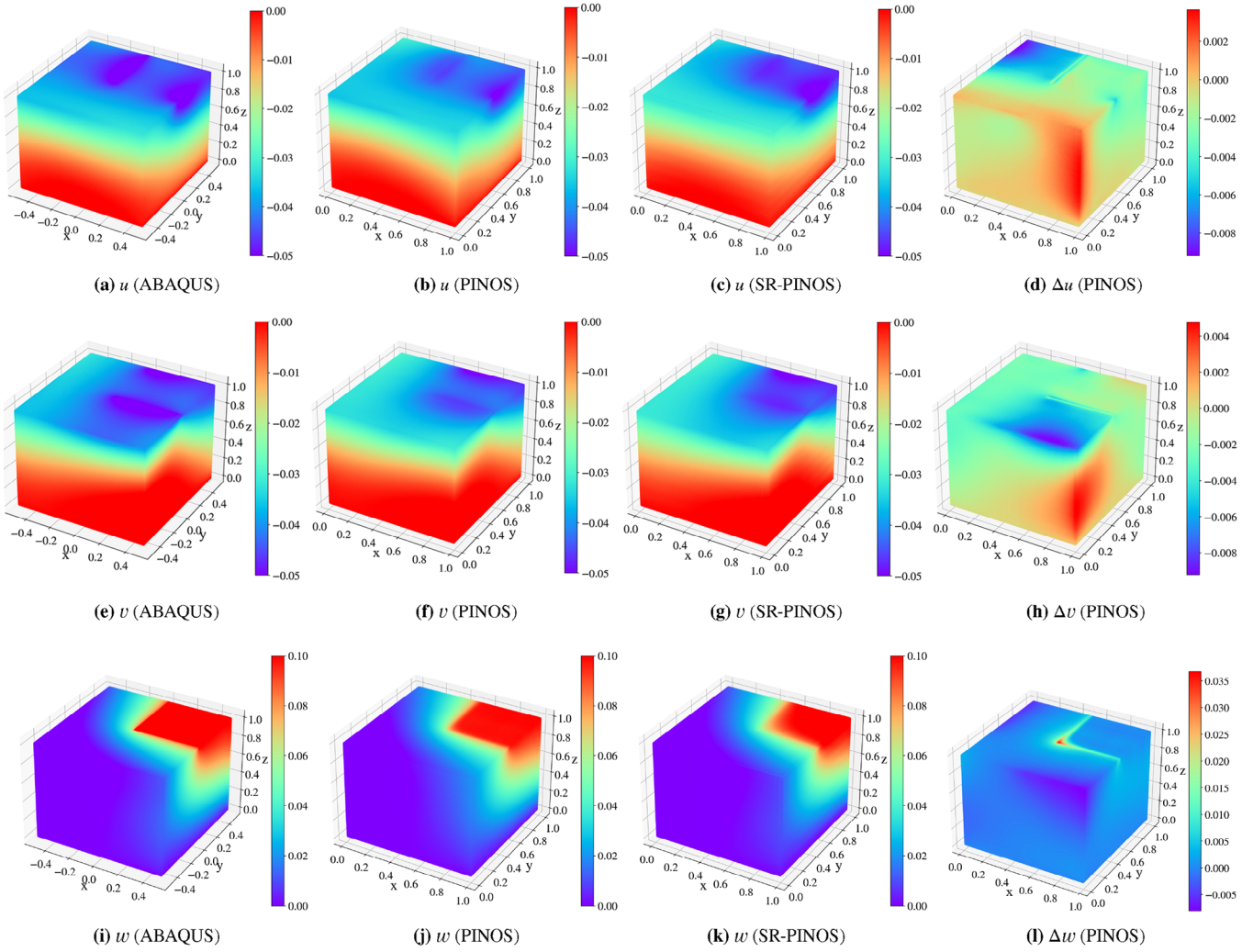
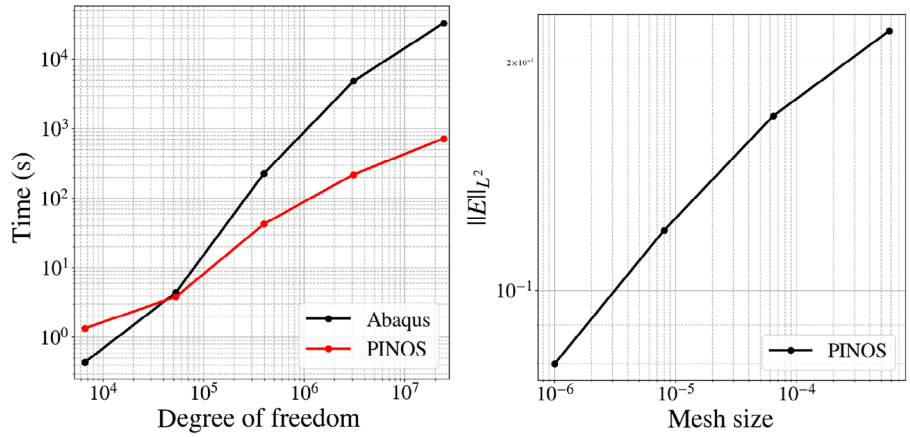


FIGURE 5 Results of PINOS on the 3D box under eccentric stress. The displacements in the x, y, and z directions are denoted by u, v, and w, respectively. The reference solution was obtained from ABAQUS. PINOS solution was trained and obtained on the fine mesh with $\Delta x, \Delta y, \Delta z = 0.01$ m. SR-PINOS was trained on $\Delta x, \Delta y, \Delta z = 0.02$ m mesh, and the solution was obtained at $\Delta x, \Delta y, \Delta z = 0.01$ m mesh size.

is fixed. A 1 m cube has physical properties as follows. Elastic modulus is set to be $E = 3$ MPa, and a Poisson ratio uses $\nu = 0.4$. The stress (σ) and strain (ϵ) are calculated as Equations (15) and (16), and potential energy loss is formulated as in Equation (6). In this problem, displacements and stress occur in all directions.

Figure 5 shows the solutions obtained from ABAQUS, PINOS, and SR-PINOS. As the theoretical solution to this problem is not available, the reference solutions were obtained from ABAQUS simulation as illustrated in Figure 5a,e, and i) for the displacement in x, y, and z directions at 4840 s using one million C3D8R element (3D first-order element with reduced integration) at a mesh size of 0.01. The architecture of PINOS adopted 3D FNO with 5 Fourier layers, 8 modes in three dimensions, 64 widths, a learning rate of 0.001, and a stop epoch at 2000. In PINOS, the training of a box with a mesh size of $\Delta x, \Delta y, \Delta z = 0.01$ m was conducted, and the solutions were obtained as shown in Figure 5b,f, and j. The SR-PINOS was trained by the physics loss of a box of $\Delta x, \Delta y, \Delta z = 0.02$ m mesh, and solutions were obtained by the input of a box of $\Delta x, \Delta y, \Delta z = 0.01$ m mesh, depicted in Figure 5c,g, and k. PINOS and SR-PINOS converged at 500 iterations with rates of 2.27 and 10.94 it/s, respectively. Therefore, the convergence time of PINOS was 220 s or 22× of ABAQUS where the relative L_2 errors of the solutions are 0.09, 0.10, and 0.05 for displacements in the x, y, and z directions, respectively. The convergence time of SR-PINOS is 45.5 s or approximately 4.8× of PINOS. The relative L_2 errors of the solution are 0.10, 0.08, and 0.15 for x, y, and z directions, respectively. It is important to point out that the error of displacement in the y direction (v) of SR-PINOS is less than that of PINOS. The benchmark comparison emphasizes the performances



(a) Convergence rate analysis **(b)** Relative L_2 Error Analysis

FIGURE 6 PINOS performance in the box problem. (a) PINOS is faster than ABAQUS when the mesh size is over $\Delta x = 0.04$ m. (b) The trend of the relative L_2 error converges when the mesh size is reduced.

of PINOS and SR-PINOS, which are further discussed in Section 5.

Figure 6 extended the validation of the performance of PINOS in terms of the convergence rate and relative L_2 error. In Figure 6a, the comparison between PINOS and an FE model by ABAQUS is demonstrated, in which the mesh size varied from $\Delta x = 1/12.5$ m to $\Delta x = 1/100$ m. We observed that at large mesh size ($\Delta x = 1/12.5$ m), ABAQUS is faster than PINOS. However, PINOS starts to exceed FEM speed when the mesh size is $1/25$ m. When the number of nodes is large in a small mesh size, PINOS also showed an exceeding performance compared to ABAQUS. This shows the great potential of PINOS as a numerical solver of complicated 3D problems, reducing simulation time while obtaining high accuracy of the solution.

4 | APPLICATIONS OF PHYSICS-INFORMED NEURAL OPERATOR IN DYNAMIC SOLID MECHANICS PROBLEMS

In this section, we expanded our examination of PINOS as a numerical solver and its performance in the super-resolution task of dynamic solid mechanics. Our investigations were conducted on two specific problems: the behaviors of spring masses and the propagation of waves. We adopted the strong form of the governing equations as mentioned in Section 2.3.2 and evaluated PINOS and SR-PINOS with the analytical solutions of the problems. We carried out the numerical experiments on the SQUID platform with the Intel Xeon Platinum 8368 (Icelake/2.4 GHz 38 cores) two CPUs, and a single unit

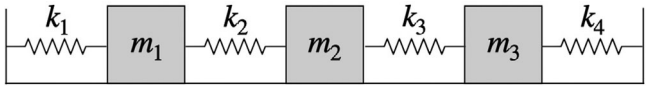


FIGURE 7 An oscillation system of three masses connected with springs.

of NVIDIA A100. The reference architecture of PINOS was retrieved from the FNO model (Li et al., 2021), and the hyperparameters were adjusted and reported in each subsection.

4.1 | Ordinary differential equations of spring masses

PINOS as a numerical solver of a dynamic solution of the spring-mass system is performed. The system consists of three masses connected by four massless springs on a frictionless horizontal surface. We examined the dynamic displacement (u) and velocity ($\dot{u}$) within 10 s of 1/100 s time step from the initial condition of the ODEs written in Equation (19). The masses are 1 kg ($m_1 = m_2 = m_3 = 1$ kg), and the spring constants are set to be $k_1 = k_4 = 2$ N/m and $k_2 = k_3 = 1$ N/m as illustrated in Figure 7.

$$\begin{aligned}
\mathcal{L}_{ODE_1} &: m_1\ddot{u}_1(t) + k_1u_1(t) - k_2\left(u_2(t) - u_1(t)\right) = 0,\\
\mathcal{L}_{ODE_2} &: m_2\ddot{u}_2(t) + k_2\left(u_2(t) - u_1(t)\right) - k_3\left(u_3(t) - u_2(t)\right) = 0,\\
\mathcal{L}_{ODE_3} &: m_3\ddot{u}_3(t) + k_3\left(u_3(t) - u_2(t)\right) + k_4u_3(t) = 0.
\end{aligned} \tag{19}$$

The initial condition is given when m_1 is pulled 1 m to the x direction ($u_1 = 1$) and other masses are not ($u_2 = u_3 = 0$), and all are not moving. The initial conditions are written

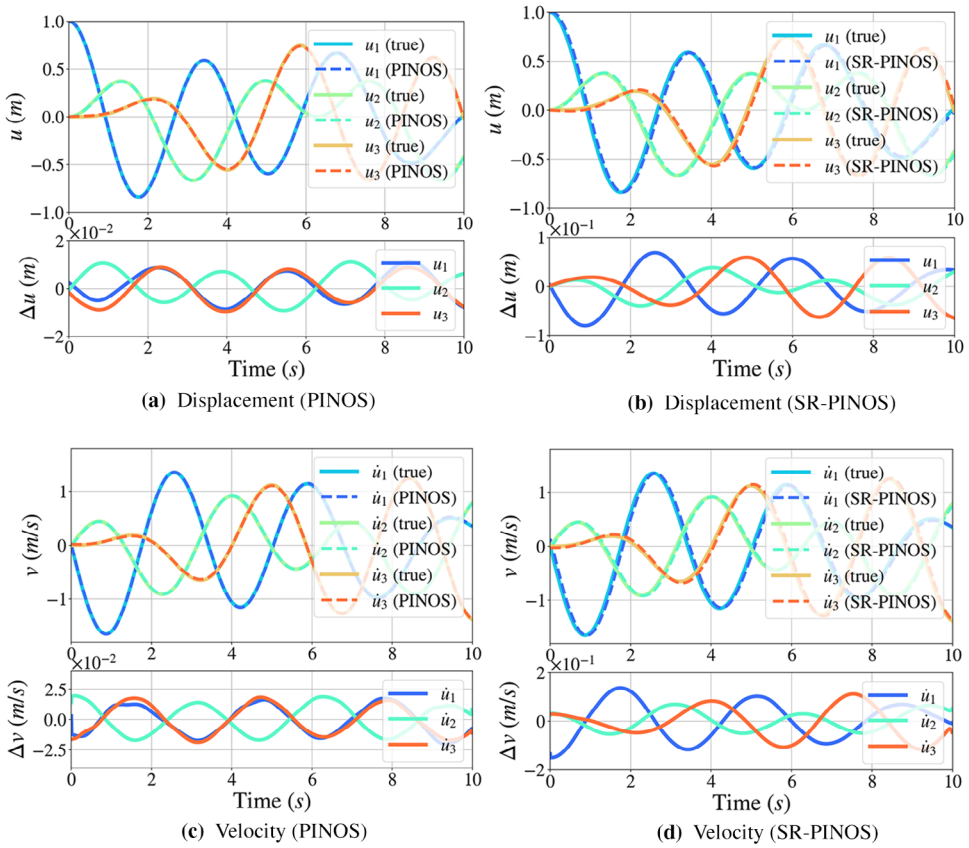


FIGURE 8 Results of PINOS and SR-PINOS on the 1D ODE of spring masses. The displacements of m_1, m_2, and m_3 are denoted by u_1, u_2, and u_3, respectively. The reference solution was obtained from an analytical solution. PINOS solution was trained and obtained on the fine time step with $\Delta t = 0.01$ s. SR-PINOS was trained on $\Delta t = 0.05$ s time step, and the solution was obtained at $\Delta t = 0.01$ s time step.

in Equation (20).

$$\begin{aligned} \mathcal{L}_{IC_1} &: u_1(0) = 1, \dot{u}_2(0) = 0, \dot{u}_3(0) = 0, \\ \mathcal{L}_{IC_2} &: \dot{u}_1(0) = 0, \dot{u}_2(0) = 0, \dot{u}_3(0) = 0. \end{aligned} \tag{20}$$

The analytical solutions are derived in Equation (21) for displacements, and their derivatives by time are the velocity solutions.

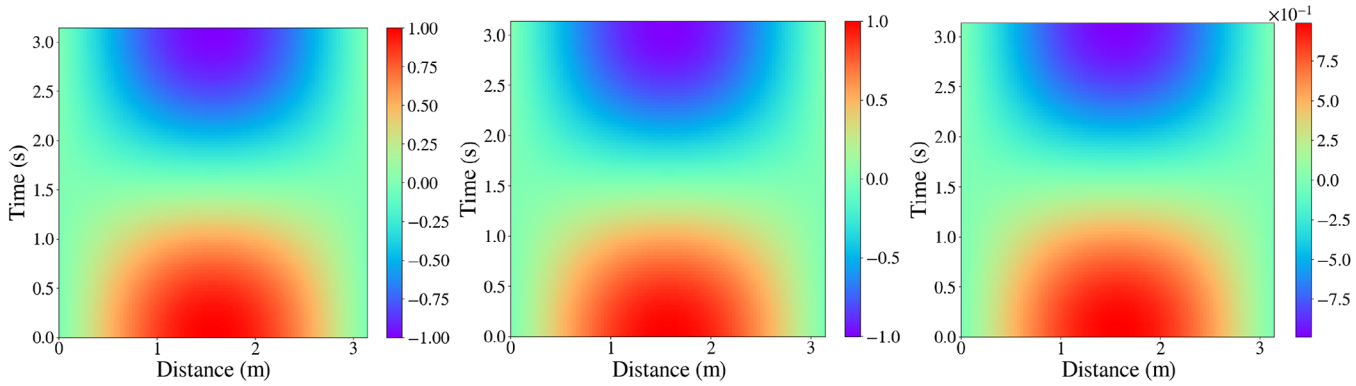
$$\begin{aligned} u_1(t) &= (1/6)\cos(t) + (1/2)\cos(\sqrt{3}t) + (1/3)\cos(2t), \\ u_2(t) &= (2/6)\cos(t) - (1/3)\cos(2t), \\ u_3(t) &= (1/6)\cos(t) - (1/2)\cos(\sqrt{3}t) + (1/3)\cos(2t). \end{aligned} \tag{21}$$

Figure 8 depicts the results of the numerical experiments. In Figure 8a,c, the displacement and velocity are represented at the top of the subfigures, and the bottom of the subfigures shows the errors when the solutions from PINOS were compared to the analytical solutions. Figure 8a shows the ODE loss, and Figure 8d shows the validation loss. It was observed that PINOS learned to meet the initial conditions within a few hundred iterations, while it took 4000 epochs to gradually converge the validation loss. PINOS converged at 26.15 s or 4000 iterations with a rate of 152.94 it/s. The relative L_2 errors of the dynamic displacements are approximately 0.01.

To investigate the super-resolution ability in a strong form of governing equation loss in a dynamic system, SR-PINOS was trained with 200 timesteps but validated with 1000 steps of 10 s. The model converged at 22.62 s or 3500 iterations with a rate of 154.67 it/s. The comparisons of displacements and velocities are illustrated in Figure 8b,d. The resolution of input length is increased by 5×, and SR-PINOS showed good performance with a relative L_2 loss of nearly 0.09 for the displacements. However, the training time of SR-PINOS is not significantly different from that of PINOS. The results of SR-PINOS suggested the capability of the nonlinear interpolation of the solution solver, which may provide benefits in the solid mechanics dynamic system.

4.2 | Wave propagation

In solid mechanics, the study of the propagation of waves has been applied to seismic analysis and vibration testing. In this work, we adopted PINOS models to solve the



(a) Displacement solution (Analytic) **(b)** Displacement solution (PINOS) **(c)** Displacement solution (SR-PINOS)

FIGURE 9 Results of PINOS and SR-PINOS on the wave equations. The reference solution was obtained from an analytical solution. PINOS solution was trained and obtained on the fine step with $\Delta t = \pi/100$ s and $\Delta x = \pi/100$ m. SR-PINOS was trained on $\Delta t = \pi/50$ s and $\Delta x = \pi/50$ m step, and the solution was obtained at $\Delta t = \pi/100$ s and $\Delta x = \pi/100$ m step.

problem. The wave propagation is considered a 2D problem because the displacement solution field (u) is in the spatial (x) and time domain (t). The propagation of waves is physically written in Equation (22).

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \tag{22}$$

where c is wave speed, set to be 1 for our numerical experiment. The boundary and initial conditions of the numerical experiments are set to be Equation (23).

$$\begin{aligned} \mathcal{L}_{BC} &: u(0,t) = 0, u(\pi,t) = 0, \\ \mathcal{L}_{IC} &: u(x,0) = \sin(x). \end{aligned} \tag{23}$$

PINOS examined the propagation in $x \in [0, \pi]$ and $t \in [0, \pi]$. The spatial and time frames are divided into 100 intervals and steps. To solve the 2D solution, we employed the following model. The parameters and architecture of 2D FNO in this problem included 4 Fourier layers, 12 modes for both dimensions, 32 widths, a learning rate of 0.0001, and a stop epoch at 10,000. The analytical solution of the wave propagation system is expressed in Equation (24).

$$u = \sin\left(\frac{\pi x}{L}\right)\cos\left(\frac{\pi t}{T}\right) = \sin(x)\cos(t). \tag{24}$$

Figure 9 shows the numerical results and comparison of PINOS and SR-PINOS with the analytical results. PINOS was trained and validated on the step with $\Delta t = \pi/100$ s and $\Delta x = \pi/100$ m. SR-PINOS was trained on the input and output sizes of $\Delta t = \pi/50$ s and $\Delta x = \pi/50$ m step and validated on a $\Delta t = \pi/100$ s and $\Delta x = \pi/100$ m.

Figure 9b,c shows the displacement solution on mesh size $\Delta t = \pi/100$ s and $\Delta x = \pi/100$ m by PINOS and SR-PINOS, respectively. The relative L_2 error of PINOS is 0.002, and the relative L_2 error of SR-PINOS is 0.013. PINOS converged at 17.41 s or 2500 iterations with a rate of 143.62 it/s, and SR-PINOS converged at 13.31 s or 2000 iterations with a rate of 152.31 it/s. The results emphasize the potential of PINOS and SR-PINOS in the dynamic simulation of solid mechanics. The comparison of PINOS and SR-PINOS is further reported in the next section.

5 | DISCUSSION, COMPARISON, AND EXTENSIONS

The previous sections present the performance of PINOS and SR-PINOS in static elasticity and dynamic problems for single and multidimensions. This section aims to provide a quantitative comparison of PINOS and SR-PINOS in terms of accuracy and computational convergence time. The accuracies were determined by a relative L_2 error by reference solutions of a problem, and the convergence times were obtained when the relative L_2 error converged to 0.1 in the training. The models were experimented on the Intel Xeon Platinum 8368 (Icelake/2.4 GHz 38 cores) two CPUs and a single unit of NVIDIA A100 at Osaka University SQUID. The reference solutions in the truss were obtained in the analytical form in Equations (13) and (14). The reference solutions in plane and box problems were obtained from the ABAQUS standard solver (ABAQUS 2023) (Smith, 2009) conducted on a single Intel i7-10870H CPU using the double-precision FE solver without GPU acceleration. The reference solutions in ODE and wave propagation were calculated in Equations (21) and (24), respectively.

Table 1 shows the comparison of PINOS and SR-PINOS in solid mechanics problems. PINOS generally showed more accurate solutions than SR-PINOS when compared to the reference solutions. In the truss under complicated load, PINOS converged to the relative L_2 error of 0.00005

TABLE 1 The comparison of PINOS, SR-PINOS, EPINN, and PINN.

		Relative L_2 error				Convergence time (s)			
Problems	**Solution**	**PINOS**	**SR-PINOS**	**EPINN**	**PINN**	**PINOS**	**SR-PINOS**	**EPINN**	**PINN**
Truss	u	0.00005	0.035	0.022	0.161	0.90	0.33	427	5957
Plane	u	0.091	0.250	0.286	0.8	7	6	253	5957
	v	0.018	0.054	0.092	0.9	7	6	253	5957
Box	u	0.092	0.101	-	-	220	45.5	-	-
	v	0.102	0.106	-	-	220	45.5	-	-
	w	0.052	0.150	-	-	220	45.5	-	-
Spring	u_1	0.012	0.092	-	0.019	26.2	12.1	-	266
	u_2	0.007	0.080	-	0.014	26.2	12.1	-	266
	u_3	0.012	0.092	-	0.013	26.2	12.1	-	266
Wave	u	0.002	0.013	-	0.003	17.7	13.3	-	60

Abbreviations: EPINN, exact Dirichlet boundary physics-informed neural network; PINN, physics-informed neural network; PINOS, physics-informed neural operators solver; SR-PINOS, super-resolution physics-informed neural operators solver.

at 0.9 s, and SR-PINOS converged to 0.035 after 0.33 s. J. Wang et al. (2023) reported that EPINN converged to 0.022 after 427 s on the same GPU computing. In the plane under eccentric stress, the solutions obtained from PINOS were up to 0.091 and 0.018 relative L_2 errors, and the convergence time of PINOS was 7 s. SR-PINOS converged after 6 s with higher relative L_2 errors of 0.25 and 0.05 in horizontal and vertical displacement solutions. In the study of J. Wang et al. (2023), EPINN converged after 253 s and obtained the relative L_2 errors of 0.14 and 0.045 for the solutions while PINN did not converge after 5957 s. The ABAQUS reference was obtained after 265 s of numerical solving. In the box deformation, the reference solutions were obtained after 4840 s. However, our PINOS converged after 220 s, and the relative L_2 errors of solutions in the x, y, and z directions are 0.092, 0.102, and 0.054, respectively. The convergence time of SR-PINOS is nearly 5× faster than PINOS; however, the solution errors are 0.101, 0.083, and 0.156, respectively. These results emphasize the excellent performance of PINOS and SR-PINOS in terms of accuracy and time.

In dynamic problems, the relative L_2 errors of PINOS in the spring-mass system are 0.012, 0.007, and 0.012 for displacement solutions of three masses, and the relative L_2 errors of SR-PINOS are nearly 0.09. The convergence times of PINOS and SR-PINOS are near with less than 4 s difference. In the same problem, PINN converged at 266 s with an average relative L_2 error of 0.014. This highlighted that PINOS exceeded PINN performance, and the performance of SR-PINOS is nearly comparable. In wave propagation, PINOS converged after 26.2 s with 0.002 relative L_2 error. The convergence time of SR-PINOS was only 22.2 s at a coarser mesh, and the relative L_2 error of the solution was 0.013 in the same mesh as PINOS. The convergence rates were observed to be relatively higher in dynamic problems where time is sensitive. Although PINOS did not show considerable improvement in terms of accuracy, PINOS exceeded PINN performance in terms of training time. It might be suggested to utilize the weak forms such as the variational approach for PINOS to learn the least energy dissipation of a system.

As PINOS outperforms PINN in truss and plane tasks, the results are related to layers in the architecture. We discuss that the advantages of PINOS include the FNO structure and operational layers. In the structure, the inputs and outputs are mesh-based which allows the computation to directly obtain the solution function $(x, y) \longrightarrow u(x, y)$. Furthermore, the solution allows the operational layer to obtain the derivatives in a precise manner, which further leads to the exact computation for loss and optimization. Another advantage of PINOS from the architecture is the super-resolution capability. The mesh-based structure of inputs and outputs allows the FNO architecture to provide super-resolution results, which contributes to the reduction of computational time in complicated solid mechanics problems such as the box problem with mesh size 0.01. While this paper shows the computational efficiency of the proposed PINOS in static and dynamic problems, the current developments are restricted by two major limitations. First, the model encounters difficulty in the irregular body and may not be efficient for the computation related to complex geometric shapes. In response to the limitation, geometry-aware and geometry-informed architecture, namely Geo-FNO and GINO, were proposed in a data-driven training (Li et al., 2022; Li et al., 2023). In simulation-free methods, it is required to incorporate geometry information of an irregular shape in the architecture of PINOS. Second, the current proposed model does not consider non-linear problems. The neural-operator architecture has been observed to have the capability of approximating operators raised in non-linear PDEs in data-rich systems (Li et al., 2022). Thus,

it is suggested to implement non-linear relationships of materials in solid mechanics such as hyper-elasticity to the physics-informed operation. With the capablity to rapidly reconstruct solution domains, the non-linearity and discovery of non-linear relationships will be exciting areas to explore for the neural operator-based solver.

In addition to the current limitations, this study suggests implementations to enhance the efficiency of the framework, particularly in terms of accuracy and speed. A fine-tuning approach to reduce simulation time in complicated problems using the super-resolution and function mapping capability of the FNO model. The approach will start by training PINOS on a larger mesh, and when it converges we then train the model on a smaller mesh. By the suggested approach, the convergence time will be shortened as the learning on a larger mesh is faster, and PINOS will be able to transfer the learning on a smaller mesh. Furthremore, in the current development, the boundary and initial conditions are soft constrained, as formulated in the loss functions shown in Equation (5). The hard constraints can be further studied to strongly enforce the exact boundary and initial constraints to the framework. In EPINN, J Wang et al. (2023) reported a notable reduction of convergence speed because incorportaing the known constraints information to the framework will reduce the computational resources of the PINNs that learn the loss functions of boundary and initial conditions.

6 | CONCLUSIONS

This study proposes PINOS as a numerical solver and SR-PINOS models for efficient and accurate simulations in solid mechanics. Our framework adopts FNO architecture and the super-resolution capability with the weak form and the strong form of governing equations in static and dynamic systems, respectively. Numerical experiments assured superior performances over PINN and ABAQUS software. The major contributions of this study are summarised as follows:

1. This is the first work in neural operators without labeled data contexts and examines the performances of the strong and weak forms of governing equations in solid mechanics.
2. PINOS exceeded PINN performance for 474× and 36× in 1D and 2D static problems. PINOS also achieved 38× and 22× speed up of ABAQUS in 2D and 3D static problems.
3. PINOS exceeded PINN performance at 10× and 3.4× in 1D and 2D dynamic problems.
4. The performance of SR-PINOS also demonstrated high efficiency in static and dynamic simulations.

It is noteworthy that while this work shows remarkable results in modeling tasks in solid mechanics, the future scope can be extended to necessary developments which include complex geometry, non-linear materials, and computational enhancement of loss learning. These challenges can be considered as the prospective goals of the work, and addressing these limitations could result in a more robust framework. Consequently, PINOS and SR-PINOS will achieve significant contributions to applications in engineering structures, which include but are not limited to interesting directions such as design optimization, structural analysis, and digital twins.

ACKNOWLEDGMENTS

The work presented in this paper was financially supported by the University of Hong Kong (HKU) Seed Funding Programme for Basic Research for New Staff, HKU start-up funding for new staff, and the computing services of Osaka University SQUID supercomputer. This work is partially supported by the Grant-in-Aid for JSPS (Grant No: 22H10576). The findings in the paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the sponsor.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interest.

REFERENCES

Adeli, H., & Yeh, C. (1989). Perceptron Learning in Engineering Design. *Computer-Aided Civil and Infrastructure Engineering*, *4*(4), 247–256. Portico. https://doi.org/10.1111/j.1467-8667.1989.tb00026.x

Abueidda, D. W., Koric, S., Guleryuz, E., & Sobh, N. A. (2022). Enhanced physics-informed neural networks for hyperelasticity. *International Journal for Numerical Methods in Engineering*, *124*(7), 1585–1601. https://doi.org/10.1002/nme.7176

Abueidda, D. W., Lu, Q., & Koric, S. (2021). Meshless physics-informed deep learning method for three-dimensional solid mechanics. *International Journal for Numerical Methods in Engineering*, *122*(23), 7182–7201. https://doi.org/10.1002/nme.6828

Anandkumar, A., Azizzadenesheli, K., Bhattacharya, K., Kovachki, N., Li, Z., Liu, B., & Stuart, A. (2019). Neural operator: Graph kernel network for partial differential equations. Paper presented at the ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations. https://openreview.net/forum?id=fg2ZFmXFO3

Bai, J., Rabczuk, T., Gupta, A., Alzubaidi, L., & Gu, Y. (2022). A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics. *Computational Mechanics*, *71*, 1–20. https://doi.org/10.1007/s00466-022-02252-0

Bischof, R., & Kraus, M. (2021). *Multi-objective loss balancing for physics-informed deep learning* [Unpublished manuscript]. https://doi.org/10.13140/RG.2.2.20057.24169

Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. (2021). Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, *37*, 1727–1738. https://doi.org/10.1007/s10409-021-01148-1

Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, *143*(6), 060801. https://doi.org/10.1115/1.4050542

Chiu, P.-H., Wong, J. C., Ooi, C., Dao, M. H., & Ong, Y.-S. (2022). Can-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method. *Computer Methods in Applied Mechanics and Engineering*, *395*, 114909. https://doi.org/10.1016/j.cma.2022.114909

Weinan, E., & Bing, Y. (2017). The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, *6*, 1–12. https://doi.org/10.1007/s40304-018-0127-z

Fang, Z. (2022). A high-efficient hybrid physics-informed neural networks based on convolutional neural network. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(10), 5514–5526. https://doi.org/10.1109/TNNLS.2021.307087

Haghighat, E., Raissi, M., Moure, A., Gomez, H., & Juanes, R. (2021). A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, *379*, 113741. https://doi.org/10.1016/j.cma.2021.113741

Henkes, A., Wessels, H., & Mahnken, R. (2022). Physics informed neural networks for continuum micromechanics. *Computer Methods in Applied Mechanics and Engineering*, *393*, 114790. https://doi.org/10.1016/j.cma.2022.114790

Heydari, A. A., Thompson, C. A., & Mehmood, A. (2019). *Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions*. arXiv. https://doi.org/10.48550/arXiv.1912.12355

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hurtado, D., Costabal, F., Yang, Y., Perdikaris, P., & Kuhl, E. (2020). Physics-informed neural networks for cardiac activation mapping. *Frontiers of Physics*, *8*, 42. https://doi.org/10.3389/fphy.2020.00042

Hurtado, A. C., Alamdari, M. M., Atroshchenko, E., Chang, K. C., & Kim, C. W. (2024). A data-driven methodology for bridge indirect health monitoring using unsupervised computer vision. *Mechanical Systems and Signal Processing*, *210*, 111109. https://doi.org/10.1016/j.ymssp.2024.111109

Jagtap, A., & Karniadakis, G. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, *28*, 2002–2041. https://doi.org/10.4208/cicp.OA-2020-0164

Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, *365*, 113028. https://doi.org/10.1016/j.cma.2020.113028

Javadinasab Hormozabad, S., Gutierrez Soto, M., & Adeli, H. (2021). Integrating structural control, health monitoring, and energy harvesting for smart cities. *Expert Systems*, *38*(8). Portico. https://doi.org/10.1111/exsy.12845

Jeong, H., Bai, J., Batuwatta-Gamage, C., Rathnayaka, C., Zhou, Y., & Gu, Y. (2023). A physics-informed neural network-based topology optimization (PINNTO) framework for structural optimization. *Engineering Structures*, *278*, 115484. https://doi.org/10.1016/j.engstruct.2022.115484

Jiang, X., Wang, D., Fan, Q., Zhang, M., Lu, C., & Lau, A. P. T. (2022). Physics-informed neural network for nonlinear dynamics in fiber optics. *Laser & Photonics Reviews*, *16*(9), 2100483. https://doi.org/10.1002/lpor.202100483

Johnny, W., Brigido, H., Ladeira, M., & Souza, J. C. F. (2022). Fourier neural operator for image classification. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–6). IEEE. https://doi.org/10.23919/CISTI54924.2022.9820128

Kaewnuratchadasorn, C., Wang, J., & Kim, C.-W. (2024). Neural operator for structural simulation and bridge health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *39*, 872–890. https://doi.org/10.1111/mice.13105

Kapoor, T., Wang, H., Núñez, A., & Dollevoet, R. (2023). *Physics-informed machine learning for moving load problems*. arXiv. https://doi.org/10.48550/arXiv.2304.00369

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Kharazmi, E., Zhang, Z., & Karniadakis, G. E. (2019). Variational physics-informed neural networks for solving partial differential equations. arXiv. http://arxiv.org/abs/1912.00873

Kharazmi, E., Zhang, Z., & Karniadakis, G. E. (2021). hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, *374*, 113547. https://doi.org/10.1016/j.cma.2020.113547

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274. https://doi.org/10.1177/0278364913495721

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2023). Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, *24*(89), 1–97. https://doi.org/10.5555/3648699.3648788

Li, Z., Huang, D. Z., Liu, B., & Anandkumar, A. (2022). *Fourier neural operator with learned deformations for PDEs on general geometries*. arXiv. https://doi.org/10.48550/arXiv.2207.05209

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. Paper presented at the International Conference on Learning Representations (ICLR 2021). https://openreview.net/forum?id=c8P9NQVtmnO

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., & Anandkumar, A. (2023). *Physics-informed neural operator for learning partial differential equations*. https://arxiv.org/abs/2111.03794

Li, Z., Kovachki, N. B., Choy, C., Li, B., Kossaifi, J., Otta, S. P., Nabian, M. A., Stadler, M., Hundt, C., Azizzadenesheli, K., & Anandkumar, A. (2023). Geometry-Informed Neural Operator for Large-Scale 3D PDEs (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2309.00583

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. (2021). Learning nonlinear operators via Deeponet based on the universal

approximation theorem of operators. *Nature Machine Intelligence*, *3*, 218–229.

Nabian, M. A., Gladstone, R. J., & Meidani, H. (2021). Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, *36*(8), 962–977. Portico. https://doi.org/10.1111/mice.12685

Office, Great Britain. His Majesty's Stationery Office. (1923). *On Castigliano's theorem of least work, and the principle of St. Venant: May 1922*. Reports and memoranda. H.M. Stationery Office. https://books.google.co.th/books?id=tQEyzgEACAAJ

O'Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks*. arXiv. http://arxiv.org/abs/1511.08458

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, P., Kashinath, K., & Anandkumar, A. (2022). *FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators*. arXiv. https://doi.org/10.48550/arXiv.2202.11214

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., & Romero-Troncoso, R. J. (2019). Recurrent neural network model with Bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, *178*, 603–615. https://doi.org/10.1016/j.engstruct.2018.10.065

Pezeshki, H., Adeli, H., Pavlou, D., & Siriwardane, S. C. (2023). State of the art in structural health monitoring of offshore and marine structures. *Proceedings of the Institution of Civil Engineers - Maritime Engineering*, *176*(2), 89–108. https://doi.org/10.1680/jmaen.2022.027

Rahman, M. A., Ross, Z. E., & Azizzadenesheli, K. (2023). *U-no: U-shaped neural operators*. https://arxiv.org/abs/2204.11127

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Rao, C., Sun, H., & Liu, Y. (2021). Physics-informed deep learning for computational elastodynamics without labeled data. *Journal of Engineering Mechanics*, *147*(8), 04021043. https://doi.org/10.1061/(ASCE)EM.1943-7889.0001947

Raonić, B., Molinaro, R., Ryck, T. D., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., & de Bézenac, E. (2023). *Convolutional neural operators for robust and accurate learning of PDEs*. arXiv. https://doi.org/10.48550/arXiv.2302.01178

Rashid, M. M., Pittie, T., Chakraborty, S., & Krishnan, N. A. (2022). Learning the stress-strain fields in digital composites using Fourier neural operator. *iScience*, *25*(11), 105452. https://doi.org/10.1016/j.isci.2022.105452

Sherstinsky, A. (2018). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. arXiv, http://arxiv.org/abs/1808.03314

Smith, M. (2009). *ABAQUS/Standard user's manual, version 6.9*. Dassault Systèmes Simulia Corp.

Song, L., Wang, C., Fan, J., & Lu, H. (2022). Elastic structural analysis based on graph neural network without labeled data. *Computer-Aided Civil and Infrastructure Engineering*, *38*(10), 1307–1323. Portico. https://doi.org/10.1111/mice.12944

Sun, A. Y., Li, Z., Lee, W., Huang, Q., Scanlon, B. R., & Dawson, C. (2023). Rapid flood inundation forecast using Fourier neural operator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops* (pp. 3733–3739). IEEE.

Tripura, T., & Chakraborty, S. (2023). Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, *404*, 115783. https://doi.org/10.1016/j.cma.2022.115783

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. U., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30, pp. 6000–6010). Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wandel, N., Weinmann, M., Neidlin, M., & Klein, R. (2022). Spline-pinn: Approaching pdes without data using fast, physics-informed hermite-spline cnns. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*, 8529–8538. https://doi.org/10.1609/aaai.v36i8.20830

Wang, H., & Qin, Q.-H. (2019). Mechanics of solids and structures. In H. Wang, & Q.-H. Qin, (Eds.), *Methods of fundamental solutions in solid mechanics* (pp. 53–90). Elsevier. https://doi.org/10.1016/B978-0-12-818283-3.00002-6

Wang, J., Mo, Y., Izzuddin, B., & Kim, C.-W. (2023). Exact Dirichlet boundary physics-informed neural network EPINN for solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, *414*, 116184. https://doi.org/10.1016/j.cma.2023.116184

Xiong, W., Huang, X., Zhang, Z., Deng, R., Sun, P., & Tian, Y. (2023). *Koopman neural operator as a mesh-free solver of non-linear partial differential equations*. arXiv. https://doi.org/10.48550/arXiv.2301.10022

Yadav, G., Natarajan, S., & Srinivasan, B. (2021). Distributed PINN for linear elasticity – A unified approach for smooth, singular, compressible and incompressible media. *International Journal of Computational Methods*, *19*, 2142008. https://doi.org/10.1142/S0219876221420081

Zhang, E., Dao, M., Karniadakis, G. E., & Suresh, S. (2022). Analyses of internal structures and defects in materials using physics-informed neural networks. *Science Advances*, *8*(7), eabk0644. https://doi.org/10.1126/sciadv.abk0644

Zhang, L., Cheng, L., Li, H., Gao, J., Yu, C., Domel, R., Yang, Y., Tang, S., & Liu, W. K. (2021). Hierarchical deep-learning neural networks: Finite elements and beyond. *Computational Mechanics*, *67*(1), 207–230. https://doi.org/10.1007/s00466-020-01928-9

How to cite this article: Kaewnuratchadasorn, C., Wang, J., & Kim, C.-W. (2024). Physics-informed neural operator solver and super-resolution for solid mechanics. *Computer-Aided Civil and Infrastructure Engineering*, 1–17. https://doi.org/10.1111/mice.13292