Received: 12 August 2024 | Accepted: 9 December 2024

DOI: 10.1111/mice.13405

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Geometry physics neural operator solver for solid mechanics

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

[1]Department of Civil Engineering, The University of Hong Kong, Pok Fu Lam, 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
National Natural Science Foundation of China, Grant/Award Number: 52408221; Hong Kong Innovation and Technology Support Programme, Grant/Award Number: ITS/041/23MX

Abstract

This study developed Geometry Physics neural Operator (GPO), a novel solver framework to approximate the partial differential equation (PDE) solutions for solid mechanics problems with irregular geometry and achieved a significant speedup in simulation time compared to numerical solvers. GPO leverages a weak form of PDEs based on the principle of least work, incorporates geometry information, and imposes exact Dirichlet boundary conditions within the network architecture to attain accurate and efficient modeling. This study focuses on applying GPO to model the behaviors of complicated bodies without any guided solutions or labeled training data. GPO adopts a modified Fourier neural operator as the backbone to achieve significantly improved convergence speed and to learn the complicated solution field of solid mechanics problems. Numerical experiments involved a two-dimensional plane with a hole and a three-dimensional building structure with Dirichlet boundary constraints. The results indicate that the geometry layer and exact boundary constraints in GPO significantly contribute to the convergence accuracy and speed, outperforming the previous benchmark in simulations of irregular geometry. The comparison results also showed that GPO can converge to solution fields faster than a commercial numerical solver in the structural examples. Furthermore, GPO demonstrates stronger performance than the solvers when the mesh size is smaller, and it achieves over 3× and 2× speedup for a large degree of freedom in the two-dimensional and three-dimensional examples, respectively. The limitations of nonlinearity and complicated structures are further discussed for prospective developments. The remarkable results suggest the potential modeling applications of large-scale infrastructures.

1 | INTRODUCTION

Numerical solvers achieved significant success in solving numerous mechanics and engineering problems, including forward, inverse, and optimization problems. In structural engineering, the applications of computational and optimization methods are involved in structural designs and analysis, facilitating the understanding and prediction of the behavior of complex structures subjected to external conditions (Plevris & Tsiatas 2018; Aldwaik & Adeli 2014). In the past decades, substantial advancements have been made in computational solid mechanics with various numerical methods and solid mechanics for practical applications. Conventional approaches applied to model physical phenomena governed by partial differential equations (PDEs) include the finite element method (FEM) (Hughes 2012), the meshfree method (Huerta et al. 2017), and the isogeometric analysis (Hughes et al. 2005). The FEM mostly leverages spatial discretization of finite element and temporal schemes with central processing units (CPUs). The high performance of graph processing units (GPUs) and GPU-based supercomputers may not be directly used by conventional FE methods. Consequently, conventional approaches may be computationally expensive and time-consuming for some complicated inverse problems or optimization problems, such as multiphysics systems, nonlinear topology optimization of complex geometries, and material nonlinearity with significant damage effects.

Artificial intelligence for science (AI4S) has achieved significant development over the last decade. In AI4S, a major research topic is developing physics-informed machine learning (PIML) algorithms to solve mechanics problems. In recent years, tremendous progress has been made in machine learning (ML) and deep learning (DL) in computational mechanics. Numerous robust approaches have been proposed and integrated into broad applications, including pattern recognition (Rafiei et al. 2024; Alam et al. 2020), engineering (Adeli & Yeh 1989), structural designs (Sun et al. 2021), modeling (Haghighat et al. 2021), and structural monitoring (Malekloo et al. 2022), which contribute to structural stability, reliability, efficiency, and smart structures. Pereira et al. (2020) proposed a finite element machine classifier for supervised pattern recognition in multivariate data, which adopted basis functions to form probabilistic functions. Rafiei and Adeli (2017) developed the neural dynamic classification, which can effectively discover hidden feature spaces and perform optimum feature selections based on neural dynamics optimization. Furthermore, researchers extended the ML and DL approaches to the investigation of large-scale structures (Sanni-Anibire & Olatunji 2022), cities, and regions (Yeung et al. 2022). In scientific computing of structure engineering, algorithms can be classified on the basis of their applicability to various physics and data scales. Figure 1 shows the primary groups for addressing various computational engineering problems, considering the amount of data and the level of physics. The horizontal axis shows the applicability to problems associated with extensive physics principles, and the vertical axis presents the applicability of models to problems with large datasets. The classes and categorizations were inspired by reviews by Faroughi et al. (2024) and Hao et al. (2023). Although models are designed with distinct capabilities, selecting an appropriate model for a given problem is of the most significance.

The data-driven models, also known as black boxes in the ML domain, rely on empirical data and statistical methods without explicit consideration of underlying physical principles. The data-driven techniques are commonly applied to achieve various objectives, including accelerations of modeling through surrogate models, sensitivity analysis, probabilistic programming, and inverse problems (Faroughi et al. 2024). Esteghamati and Flint (2021) proposed to develop surrogate models for predicting the seismic vulnerability and assessing the environmental impacts with topological variables of a building. Jiang and Adeli (2008) developed the dynamic fuzzy neuroemulator to predict the structural displacement response. The datasets were obtained from the nonlinear structural analysis with consideration of geometrical and material nonlinearities of structures.

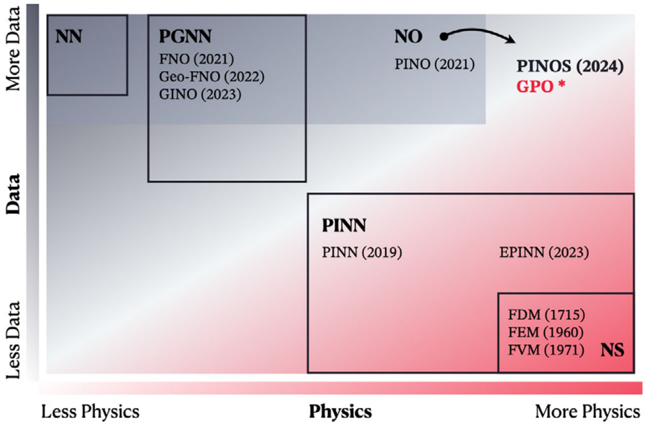


FIGURE 1 Categorization of computational models in scientific computing. Main categories include neural networks (NN), physics-guided neural networks (PGNN), physics-informed neural networks (PINN), neural operators (NO), and numerical solver (NS).

The study suggested integrating the models to narrow the design space in the practical designs. A study by Liao et al. (2023) proposed an integration approach of wavelet transformation and hierarchical clustering to balance the training datasets for improving the prediction accuracy of long short-term memory surrogate models. Perez-Ramirez et al. (2019) integrated a recurrent neural network with the nonlinear autoregressive exogenous model to predict structural dynamic responses of a five-story frame and a high-rise building under the ground motion excitation, demonstrating a robust performance in modeling the responses of the nonlinear structures. Li et al. (2024) trained a continuous-time state-space neural network to represent a physical model while fitting the experimental data of an energy dissipation device and applied it through MATLAB Simulink to predict the dynamic signal of a structure by historical earthquake excitation. On the other side, the physics-based models or numerical solvers are fundamental in computational physics and have been extensively studied over recent decades. The models derived from various physical principles and mathematical techniques represent another alternative in scientific modeling, commonly adopted to investigate a phenomenon when the principles are known but data are extremely limited. Practically, the advances in sensing technology provide more access to real-world information on structures. To this end, the integration between data-driven and physics-based models, known as physics-guided and physics-informed neural networks, helps bridge the gap between simulation and reality, leading to a digital twin of structures (Faroughi et al. 2024).

Physics-guided neural networks (PGNN) establish surrogate relationships among physical parameters governed by the physics principle of a particular system. The method trains a neural network to achieve surrogate mappings using datasets obtained from computer-based simulations, which may be stochastic or deterministic. In the domain of structural engineering, many works adopt PGNN to accelerate the computational cost caused by traditional solvers. R. Zhang et al. (2020) developed a physics-guided convolution neural network and applied it to learn a mapping between ground motion input and seismic response output of a structure for data-driven seismic response modeling from a limited dataset, reporting the superior performance over non-physics-guided networks. Huang et al. (2022) proposed a physics-guided deep convolution neural network in a physics-guided manner using datasets from FE simulations and measured data, achieving greatly improved accuracy compared to a network trained with only measured data. Taghizadeh et al. (2024) introduced the hierarchical multifidelity graph neural network with a curriculum learning–based training method to train an accurate surrogate model while reducing the computational cost of training. Miao et al. (2024) developed a physics-guided conditional generative adversarial network to reconstruct an artificial response diagram in the time domain and mapped it to response-compatible ground motion using a neural network. The results were verified with nonlinear time-history analysis under six different structures and various conditions. Kaewnuratchadasorn et al. (2023) developed DL-based models to predict dynamic responses of a bridge from its damage defined by the reduction of stiffness, in which the responses are triggered when a vehicle runs along the bridge. The study showed that the Fourier neural operator (FNO) was able to speed up nearly 20 times compared to a numerical solver for damage analysis in inverse problems. Despite tremendous developments in model designs and successful applications, the generalizability and interpretability need a large dataset. In a problem such as structural modeling in which physics principles are known, regularizing physics into the model will lead to less computational cost and more generalizable inferences (Wang & Yu 2023).

Physics-informed neural networks (PINN) incorporate physics principles or governing equations into learning the loss functions to ensure the models' predictions adhere to the underlying physical constraints (Raissi et al. 2019; Karniadakis et al. 2021). Cross et al. (2021) demonstrated the improvement in the predictions of the gray-box models that fuse physics-based models with data-driven approaches in structural analysis and monitoring. Meng et al. (2023) integrated PINN with the first-order reliability method to address the optimization in structural reliability directly without solving the governing PDEs. Jeong et al. (2023) proposed PINN-based topology optimization (PINNTO) as an alternative to FEA in determining the deformation states by training energy-based physics laws without labeled data. The work suggested that PINNTO is capable of producing comparable designs to the current successful approaches. Zhang et al. (2022) introduced a PINN-based framework to identify multiple structural features and predict the physical characteristics of voids in linearly elastic, hyperelastic, and plastic materials. Moradi et al. (2023) applied novel PINN architecture that follows parallel and sequential natures and focuses on system identification and state prediction. Gao et al. (2024) proposed the physics-informed graph-assisted auto-encoder (PGI-AE) and integrated it with a residual graph convolution neural network classifier (RGCNNC) to monitor an urban flow field from sparse sensor data. RGCNNC was proposed to determine wind attack angle and graph features for PGI-AE, which leverage graph neural networks and physics loss of Navier–Stokes equations. Wang et al. (2024) adopted PINN for both forward and inverse problems in soil and lining predictions. Multiple PDE losses are formed in soil and lining PINN (SL-PINN) as a result of multiple bodies which include surrounding soils and lining. The work showed excellent performance of SL-PINN in various contact conditions by verification with FEM. In structural engineering, Song et al. (2023) developed the structural graph neural network-elastic (StructGNN-E) model which integrates the underlying mechanics' principle, graph isomorphism network, and neural network optimization for structural analysis. Fu et al. (2024) proposed a physics-informed deep reinforcement learning framework by utilizing reinforcement learning to learn the policy of structural design. The approach introduces physics-informed reward functions to reinforcement to learn and achieve automatic structural design. Kapoor et al. (2024) adopted PINN to solve forward and inverse problems of a dynamic system of beams and vehicles governed by the Euler–Bernoulli and Timoshenko PDEs.

Rao et al. (2021) applied PINN to approximate the quantity of interest in a PDE with hard constraints of boundary conditions for static and dynamic analysis of two-dimensional (2D) objects in computational mechanics problems without training data. A recent study proposed the exact Dirichlet boundary physics-informed neural network (EPINN) by training with energy-based loss and imposing the exact Dirichlet boundary conditions to obtain a solution with fast convergence in solid mechanics in one-dimensional, two-dimensional, and three-dimensional (3D) cases (Wang et al. 2023). Abueidda et al. (2021) proposed a deep collocation method as a mesh-free method and showed accurate modeling under different constitutive models, including linear elasticity, hyper-elasticity, and plasticity of a 3D beam. The method was further developed by implementing Fourier transform into the architecture and achieved relatively more rapid convergence Abueidda et al. (2023). Kaewnuratchadasorn et al. (2024) developed a physics-informed neural operator solver (PINOS) based on FNO and the principle of least work and reported remarkable convergences of solutions compared to PINN and comparable performance as the ABAQUS numerical solver. Although there are huge developments in computational solid mechanics, a gap in practical structural modeling is still challenging. The neural network–based models encounter complex geometry but have slow convergence, while the neural operators–based approaches are competitively fast but face challenges in irregular huge objects.

In this study, a physics-informed approach based on a neural operator architecture is proposed to address complex geometry and large computational time. This work aims to achieve efficient simulations for structural modeling by training the model with only physics principles and without labeled data. We establish the Geometry Physics neural Operator solver (GPO) by implementing the technical developments as follows.

1. This study develops a neural operator solver with geometry information in the architecture which allows the model to restrict the solving domain and explicitly understand the irregular object.
2. This study develops the boundary constraint function in the neural operator to meet the exact known boundaries and accelerates the solution convergence.
3. This study applies a neural operator to solve for solutions in irregular geometry in 2D and 3D solid mechanics problems with complex boundary conditions. The results showed that GPO is faster than a commercial solver ABAQUS in all cases reported.

This paper describes the detailed derivation of GPO features for solid mechanics problems in Section 2. Sections 3 and 4 report the performance of GPO in comparison to the previous benchmark PINOS in a 2D plane with a hole and a 3D building structure. Section 5 discusses the contributions of the added features related to the convergence. Section 6 summarizes the work of this study in structural simulations.

2 | GEOMETRY PHYSICS NEURAL OPERATOR

This section explains the architecture of the GPO. First, the neural operator solver utilizes a DL structure as a numerical solver for optimization. Second, the principle of least work, which is a variational form of a governing equation in static solid mechanics, is introduced, and its connection to the solution and derivative operations is elucidated. Next, the constraint functions for exact Dirichlet boundary conditions are formulated to restrict the known conditions of a system. Finally, the geometry layer that encodes the area of interest in neural network architecture and the derivative operation is described.

2.1 | Neural operator solver

This work focuses on developing a neural operator family for solving PDEs of complicated geometry. The neural operator was derived according to the universal approximation theorem of operators (Goswami et al. 2022; Hornik et al. 1989). In this work, we adopted the neural operator architecture implemented with the Fourier transform called FNO as a benchmark of the operator in a physics-informed manner (Z. Li et al. 2021; Z. Li, Zheng, et al. 2023). In addition, the concept of training a lightweight neural operator to achieve solutions to PDEs was employed from PINOS in solid mechanics (Kaewnuratchadasorn et al. 2024)

Neural operators ($\mathcal{N}$) are formulated to achieve a generalization of neural networks in learning represented operators. As shown in Figure 2, a neural operator is mainly composed of a lifting transformation neural network layer function ($\mathcal{P}$), kernel integral operators ($\mathcal{K}$), linear operator ($\mathcal{W}$), and a projecting transformation network ($\mathcal{Q}$). Equation (1) writes a formulation of GPO in the generalization function.

$$\mathcal{N}_\theta := \mathcal{Q} \circ (\mathcal{W}_L + \mathcal{K}_L) \circ \cdots \circ \sigma_1 (\mathcal{W}_1 + \mathcal{K}_1) \circ \mathcal{P}, \quad (1)$$

where σ_i is an activation function at layer i except for the last layer L. This work selects the Gaussian error linear unit from an intensive literature review and numerical experiments to achieve higher accuracy of solutions in nonlinear problems. We may define a Fourier layer i as $\sigma_i(\mathcal{W}_i + \mathcal{K}_i)$ and choose both linear and kernel integral operators. In this work, the core architecture of GPO utilizes a specific form of the kernel integral operator ($\mathcal{K}$) based on the

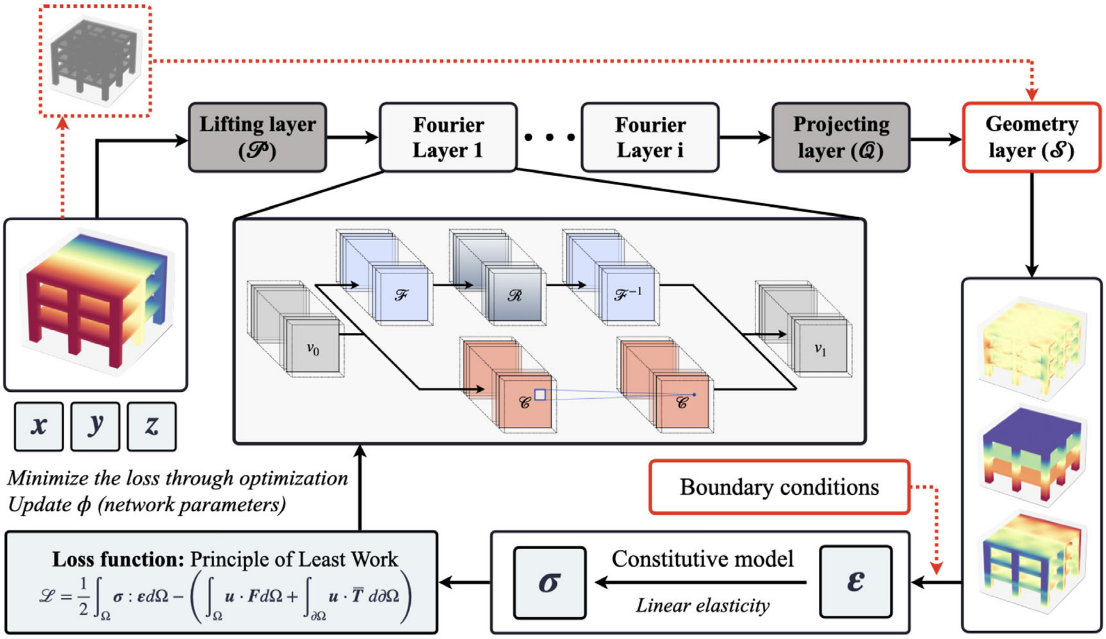

Lifting layer ($\mathscr{P}$)
Fourier Layer 1
Fourier Layer i
Projecting layer ($\mathscr{Q}$)
Geometry layer ($\mathscr{S}$)
v_0
$\mathscr{F}$
$\mathscr{R}$
$\mathscr{F}^{-1}$
$\mathscr{C}$
$\mathscr{C}$
v_1
x
y
z
Minimize the loss through optimization
Update ϕ (network parameters)
Boundary conditions
Loss function: Principle of Least Work
$\mathscr{L} = \frac{1}{2}\int_{\Omega} \sigma : \varepsilon d\Omega - \left(\int_{\Omega} u \cdot F d\Omega + \int_{\partial\Omega} u \cdot \bar{T}\, d\partial\Omega\right)$
σ
Constitutive model
Linear elasticity
ε

FIGURE 2 A schematic diagram of GPO with a modified Fourier neural operator (FNO) backbone to approximate the solutions from input coordinates and geometry. The Geometry Physics neural Operator (GPO) consists of a lifting layer ($\mathcal{P}$), Fourier convolution operator ($\mathcal{K}$), convolution network ($\mathcal{W}$), projecting layer ($\mathcal{Q}$), geometry layer ($\mathcal{S}$), exact boundary constraints, and physics-informed loss function.

convolution theorem, called the Fourier convolution operator (Z. Li et al. 2021), which consists of fast Fourier transform ($\mathcal{F}$), linear transform ($\mathcal{R}$) parameterized by network parameters (ϕ), and inverse Fourier Transforms ($\mathcal{F}^{-1}$) as formulated in Equation (2):

$$(\mathcal{K})(x) = \mathcal{F}^{-1}(\mathcal{R} \cdot (\mathcal{F}v_l))(x). \tag{2}$$

In this work, all layers in a neural operator are trainable, and the hyper-parameters will be designed and reported specifically in each problem as 2D and 3D irregular objects do not require the same number to obtain a solution. It is also significant to highlight that neural operators may be built with a combination of different operators, which includes graph-based, low-rank, Fourier convolution, and wavelet transform (Anandkumar et al. 2019; Kovachki et al. 2023; Z. Li et al. 2021; Tripura & Chakraborty 2023). This work adopted the Fourier convolution operator to report in GPO as proven and validated in Kaewnuratchadasorn et al. (2024) in consideration of solution construction and convergence ability when training the network with a physics-informed loss function.

2.2 | Physics-informed loss functions

Our GPO focuses on training to solve solutions for solid mechanics problems. A general form of PDEs in a system with linear elasticity behavior is written as Equation (3):

$$\frac{\partial \sigma_n}{\partial n} + \frac{\partial \tau_{nm}}{\partial m} + \frac{\partial \tau_{nk}}{\partial k} + f_n = 0. \tag{3}$$

where n, m, and k refer to the Cartesian coordinates x, y, and z. Many works were conducted using a general form of PDE, also called strong form, to train a model to obtain a solution to mechanical problems. Models were trained to approach the physics-informed loss function to zero to obtain the system solutions. A recent study by J. Wang et al. (2023) demonstrated that a different form of PDE, the so-called weak form, provided greater efficiency for PINN. The weak form adopts the concept of the principle of least work, stating the total energy of the system is at the minimum. The total energy is defined as the difference between internal strain energy and the external work done by forces on the body. The concept incorporates the nature of training the neural operator as it is minimizing the loss function, thus the energy loss function will be minimized to the least possible. Per the theorem, Equation (4) formulates the loss function ($\mathcal{L}_{system}$) derived from the principle of least work.

$$\mathcal{L}_{system}(\mathbf{u}; \mathbf{f}, \mathbf{t}) = \frac{1}{2}\int_{\Omega} \sigma(\mathbf{u}) \cdot \varepsilon(\mathbf{u}) d\Omega - \left(\int_{\Omega} \mathbf{u} \cdot \mathbf{f} d\Omega + \int_{\Delta\Omega_N} \mathbf{u} \cdot \overline{\tau} d\Delta\Omega_N\right), \tag{4}$$

where **u** symbolizes the displacement solution of a deformation; σ and ε represent stress and strain tensors, respectively, calculated from the displacement. The details will be further explained in the later sections. **f** and $\boldsymbol{\tau}$ stand for forces and tractions applied to a system. The calculations are based on the linear elasticity equations, where 2D and 3D cases have different assumptions. Linear elasticity equations will involve obtaining derivatives of the solutions. In this work, we adopted the finite-difference differentiation to efficiently obtain the derivatives as computational experiments reported that automatic differentiation was unable to obtain the accurate derivatives of a solution without the labeled data training of the FNO-based architecture.

2.3 | Exact boundary-constrained functions

In addition to a governing equation, the boundary condition is a prerequisite to understanding the behavior of a static system. Commonly, many works defined a loss function, called boundary loss function, to impose the solution according to a predefined boundary condition. The total loss function ($\mathcal{L}$) in physics-informed ML for static solid mechanics can be written as Equation (5),

$$\mathcal{L} = \mathcal{L}_{system} + \omega\mathcal{L}_{BC}. \tag{5}$$

where $\mathcal{L}_{system}$ is restricted by governing equations of the system. As mentioned earlier, the governing equations may be chosen as a strong form or a weak form, in which the choices depend on the model architecture and affect the convergence rate and accuracy of the solution. $\mathcal{L}_{BC}$ denotes the boundary condition loss, and ω is the weight of the boundary loss.

This approach to train a neural network to impose boundary conditions is called soft constraint training. However, numerous recent works argued that such training poses disadvantages. First, it is necessary to determine the weights for particular problems, which may vary by systems, the sensitivity of neural network models, the number of degrees of freedom, and other hyperparameters. Furthermore, the loss will not be generalized and practical when applied to a new problem set. Second, training two dependent losses is not computationally efficient as the imposing boundary conditions will influence the calculation of the governing equation. Third, the solutions do not meet the exact boundary condition. There have been many attempts to address the issues. Heydari et al. (2019) proposed a dynamic balance of loss functions and regularization, allowing adjustment of the weights of multiple losses of governing equations and boundary conditions in the context of PINN. Bischof and Kraus (2021) developed loss balancing of multiobjective optimization called ReLoBRaLo, inspired by GradNorm (Chen et al. 2018) and learning rate annealing. The study introduced a random lookback mechanism into exponential decay to determine previous steps for scaling calculation. These methods mitigate the difficulty in the first concern; however, the second and third concerns still remain to impact the convergence speed.

In response to this issue, many researchers proposed various concepts to strictly constraint boundary conditions, classified as hard constraint training. Rao et al. (2021) imposed an exact boundary condition by introducing a composite scheme of neural networks for general solutions, distance functions, and boundary conditions. A study by Sukumar and Srivastava (2022) proposed to implement a trial function and multiply with a PINN approximation to exactly satisfy Dirichlet boundary conditions. The hard boundary constraints by the approximate distant function (ADF) in the PINN approach were proposed in the EPINN (J. Wang et al. 2023). The results on static deformations in solid mechanics showed that EPINN outperformed conventional PINNs.

Instead of confiding upon the ADFs that assume smoothness, this work utilizes direct step-wise functions that explicitly prescribe values at specific boundaries and maintain the solutions solved by GPO at other arbitrary coordinates. These step-wise functions built by boundary information ensure that the boundary conditions are precisely enforced, even at nondifferentiable boundaries. By integrating these exact boundary functions into the GPO architecture, we assure the network's capability to accurately capture the behavior of the solution near the boundaries as well as enhance the stability and convergence of the GPO training process. The exact restriction of the boundary conditions results in the drop of the term $\mathcal{L}_{BC}$ in Equation (5), remaining only the system energy loss function and leading to a great reduction of computational time caused by minimizing the influencing multiple losses. We illustrate the effectiveness of our approach by conducting numerical experiments on previous problems solved by PINOS (Kaewnuratchadasorn et al. 2024), demonstrating its applicability and advantages in scenarios where traditional methods struggle to achieve satisfactory results. The importance of the exact boundary-constrained functions is further illustrated in Section 5.

2.4 | Geometry layer

A key feature of GPO in solving differential equations of more complex objects is the geometry layer where the shape geometry information is implemented in a neural network parameter layer. A geometry function ($\mathcal{S}$) formulates GPO as Equation (6) as regards the neural operator in Equation (1)

$$\mathcal{G}_\theta := \mathcal{S} \circ [\mathcal{Q} \circ (W_L + K_L) \circ \cdots \circ \sigma_1 (W_1 + K_1) \circ \mathcal{P}]. \quad (6)$$

The formulation of GPO ($\mathcal{G}_\theta$) implements a nontrainable geometry layer to restrict the exact geometry of an object by a mesh grid, instead of learning geometry from initial mesh inputs. Related works adopted a network or kernel operator to learn the geometry information from the mesh based on the universal approximation theorem in fitting data-driven problems (Z. Li et al. 2022; Z. Li, Kovachki, et al. 2023). Our work needs the geometry layer to restrict the solution field, reduce the computational cost of learning the geometry, and speed up the convergence time. The geometry layer plays a pivotal role in the GPO architecture that ensures the restricted computation of the solutions within the defined geometry. The layer function operates the masking mechanism, where points outside the specified geometry are excluded in the solution calculation by the neural operator and derivative computation within the framework. The geometry function ($\mathcal{S}$) evaluates arbitrary interior points to 1 and neglects the calculation for points outside a defined geometry, which ensures that the formulation of total gradients is influenced only by points in the valid domain. In this work, the geometry layer is built upon the known shape where the number of nontraining neural parameters is equal to the mesh of the solution fields. This formulation, thus, enforces adherence to the geometry constraints throughout the training and inference processes.

The geometry layer consists of untrainable neural network parameters of a mask tensor (M), which is formulated from a set of points $P \in R^{d \times n}$ in the geometry of an irregular structure, where n is the number of points and d is the dimension of a problem. A numerical mask tensor was formulated as a regular grid tensor from P and can be expressed as

$$M_{i,j,k} = \begin{cases} 1 & \text{if } (x_i, y_j, z_k) \in P \\ 0 & \text{otherwise,} \end{cases}$$

where, i, j, k are the index in the mask tensor $M \in R^{w,h,s}$ which was obtained based on the grid of a structure. Similar to the numerical mask tensor, the bool mask tensor was constructed and applied to the solution to ensure the solution was not calculated outside the geometry. The solution is obtained from the element-wise multiplication of the output from the Fourier and projection layers and the geometry layer, as a neural parameter function.

3 | TWO-DIMENSIONAL PROBLEMS: A PLANE WITH A HOLE IN PLANE STRESS STATE

This section examines the performance of the GPO model as a computational solver for approximating solutions to solid mechanics problems involving 2D objects with irregular geometries. The GPO model is compared with PINOS as a benchmark DL-based model in solving static solid mechanic problems. In the numerical experiment, a plane featuring a square hole was selected as the irregular 2D object. We demonstrate the performance of the GPO model using examples of a plane with a hole subject to displacement and force boundaries. It is significant to highlight again that while PINOS learns displacement boundaries, GPO imposes the conditions through the ADF method. However, both models are designed to learn the force conditions of a system.

The potential energy of the plane was computed based on Equation (4), wherein stress (σ) and strain (ε) were formulated as Equations (7) and (8).

$$\sigma_{xy} = \lambda \varepsilon_{zz} \delta_{xy} + 2\mu \varepsilon_{xy}, \quad (7)$$

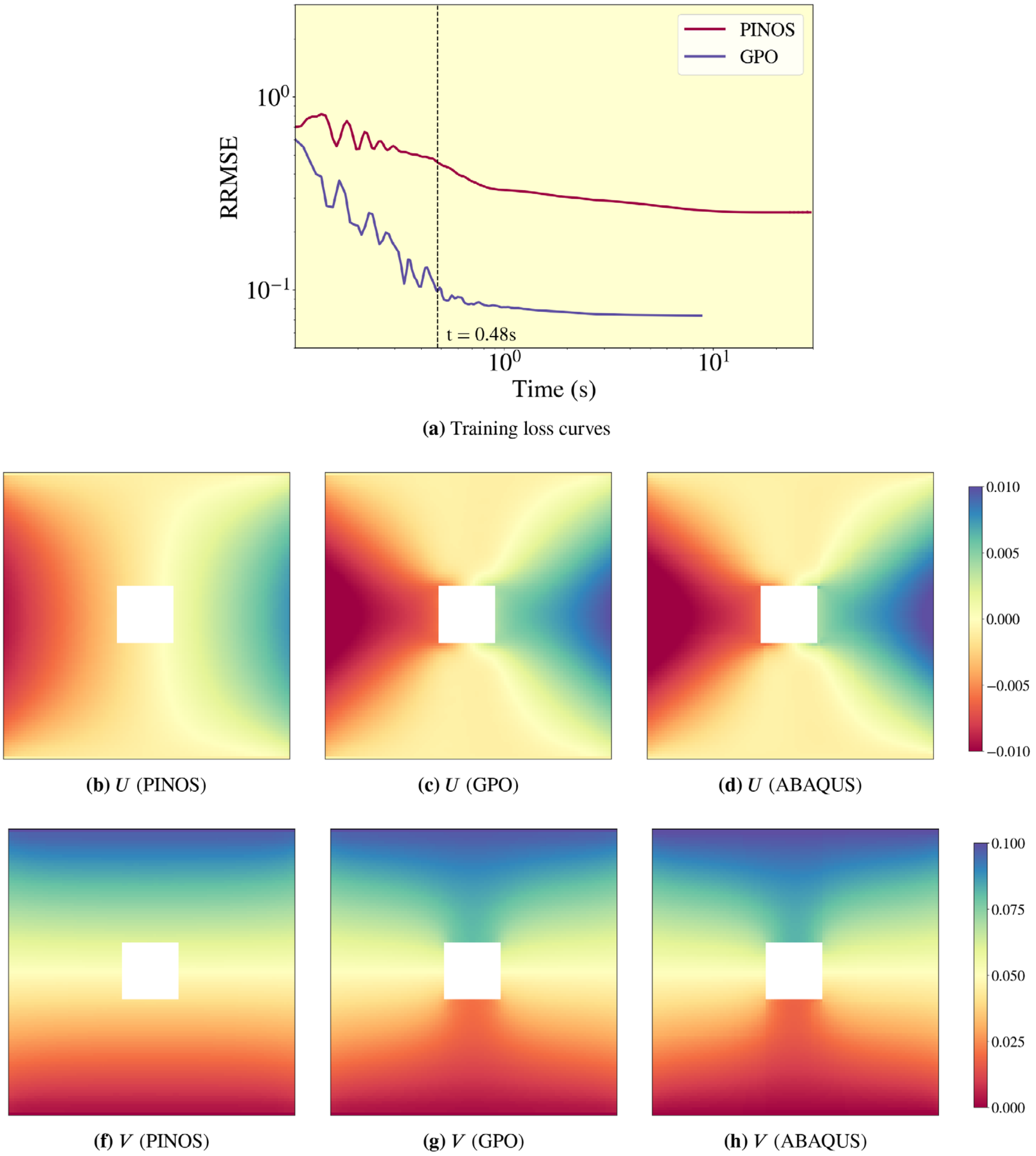
$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right), \quad (8)$$

where x, y, and z symbolize the axial directions, and the Lamé parameters are represented by λ and μ, calculated as in Equations (9) and (10) from the elastic modulus (E) and the Poisson's ratio (ν).

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad (9)$$

$$\mu = \frac{E}{2(1+\nu)}. \quad (10)$$

The computational experiments of PINOS and GPO were conducted using the Supercomputer for Quest to Unsolved Interdisciplinary Data Science (SQUID), equipped with Intel Xeon Platinum 8368 CPUs (2.4 GHz, 38 cores) and NVIDIA A100. The reports in Sections 3.1 and 3.2 adopted the FNO backbone with 2 Fourier layers, 8 modes of Fourier, and 64 hidden dimensions of kernel representation. The learning rate begins at 0.001 and decays at 0.95 rate every 100 steps. The FE model in ABAQUS was conducted on the Intel i9-13900KF CPU RAM 128 GB with no GPU acceleration. The mesh grid for the plane with a hole was structured as a regular rectangle with a resolution of 0.01 m mesh size. The relative root mean square error (RRMSE) was adopted to validate the accuracy of the solutions from GPO or PINOS compared to the reference solution from ABAQUS.



(a) Training loss curves

(b) *U* (PINOS) **(c)** *U* (GPO) **(d)** *U* (ABAQUS)

(f) *V* (PINOS) **(g)** *V* (GPO) **(h)** *V* (ABAQUS)

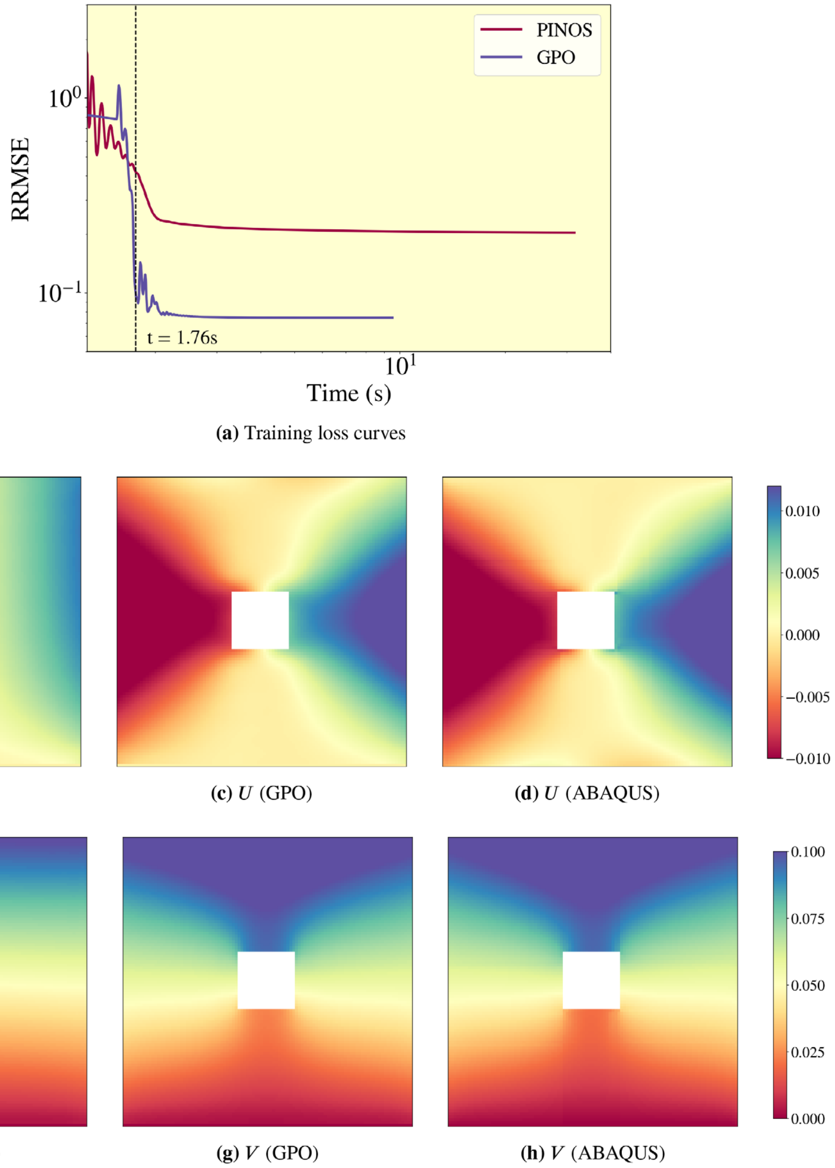
FIGURE 3 Results of GPO for a plane with a hole under fixed displacement constraints. *U* and *V* denote the horizontal and vertical displacements, respectively. The reference solution was obtained from a numerical solver in ABAQUS software where the mesh size is 0.01 m.

3.1 | Displacement boundaries

A comparative analysis between the GPO and PINOS models was conducted to determine the deformation behaviors of a plane with a hole subjected to eccentric tension. In the computational experiment, a square plane measuring 1 m on each side with a 0.2-m square hole was used. The material characteristics include the elastic modulus (E) of 10 MPa and the Poisson ratio (ν) of 0.2. The displacement constraints were applied on the top and bottom edges of the plane such that the bottom side was fixed in both horizontal and vertical directions at 0 m, while the top edge was fixed to accommodate a vertical deformation of 0.1 m.

Figure 3 presents a comparative analysis of GPO and PINOS against a reference solution derived from the numerical solver ABAQUS. This figure illustrates the horizontal and vertical deformation behaviors as captured by ABAQUS, PINOS, and GPO. It was noticed that PINOS struggled to accurately model deformations near the hole sections in the horizontal and vertical directions. In contrast, the geometry layers and operations within GPO effectively captured the complex geometry of the object and provided accurate approximations close to and around the hole areas.

As shown in the loss curve of Figure 3, GPO outperformed PINOS. Evidently, PINOS did not reduce its RRMSE to 0.1, whereas GPO achieved this benchmark within 2.7 s and further reduced it to 0.03. While ABAQUS reached a solution in 4.4 s, the GPO model was able to converge to an accurate solution approximation in just 2.7 s.



(a) Training loss curves

(b) U (PINOS) **(c)** U (GPO) **(d)** U (ABAQUS)

(f) V (PINOS) **(g)** V (GPO) **(h)** V (ABAQUS)

FIGURE 4 Results of GPO for a plane with a hole under a loading condition on the top surface. U and V denote the horizontal and vertical displacements, respectively. The reference solution was obtained from a numerical solver in ABAQUS software where the mesh size is 0.01 m.

3.2 | Force constraints

The numerical experiment solves the displacement behavior solution when force is applied at a boundary of a 2D irregular object. The plane with a hole was subjected to 10 MN force at the top surface, while the bottom surface was fixed in the horizontal and vertical directions. The physical parameters of the plane with a hole were the same as an example in Section 3.1. In this example, the object is fixed only at the bottom.

Figure 4 presents a comparative analysis of GPO and PINOS against a reference solution derived from the numerical solver ABAQUS of the plane with a hole subjected to loadings on the top surface. The figure illustrates the horizontal and vertical deformation behaviors as captured by ABAQUS, PINOS, and GPO. It was observed that PINOS struggled to accurately model deformations near the hole sections horizontally and vertically, similar to the previous case sample. In contrast, the geometry layers and operations within GPO effectively captured the complex geometry of the object and provided accurate approximations close to and around the hole areas. As shown in the loss curve of Figure 4, GPO outperformed PINOS. PINOS did not reduce the RRMSE to 0.1, whereas GPO achieved this benchmark within 1.8 s and further reduced it to 0.03. While ABAQUS solved the solution within 4.4 s, the

GPO model was able to converge to an accurate solution approximation in only 1.8 s.

4 | THREE-DIMENSIONAL PROBLEMS: BUILDING STRUCTURE

This section reports the performance of the GPO model as a computational solver for approximating solutions to solid mechanics problems involving 3D objects with irregular geometries. We compare GPO with a DL-based benchmark model, namely PINOS, for static deformations in 3D structural problems.

For the 3D structure, we consider a building with three stories. Each square story has a 6.5-m width and 0.4-m thickness. In total, the length and width of the structure are 6.5 m, and the height of the structure is 9 m. The material properties of the concrete for the three-story building structure include a mass density of 2.5 t/mm^3, elastic modulus (E) of 35,000 MPa, and a Poisson ratio of 0.2. This section comprises two examples, namely displacement and force boundaries.

In 3D solid mechanics, the stress tensor is calculated from Equations (7) and (8) where the Lamé parameters are formulated as in Equations (9) and (10), as mentioned earlier. The section reports the computational experiments on SQUID for PINOS and GPO. The core architecture of both models reported in Sections 4.1 and 4.2 applies an FNO with 4 Fourier layers, 16 modes, and 64 widths. The learning rate of 0.001 is selected and decays at 0.95 rate every 100 steps. The reference solutions were obtained from ABAQUS software on an Intel i7-10870H CPU without GPU acceleration. The mesh for the building structure was structured as a cube with a mesh of 0.1 m.

4.1 | Displacement boundaries

In this numerical experiment, we compare GPO and PINOS in a task to obtain a deformation solution for a building structure subjected to eccentric tension. The displacement constraints were applied at the top and bottom boundaries of the structure. The bottom boundary was fixed in all directions at 0 m. The top boundary was fixed to accommodate a deformation of 0.8 m in the directions of x, y, and z.

GPO and PINOS utilized an FNO with 4 Fourier layers, 16 Fourier modes, and 64 hidden dimensions of kernel representation and operated at a learning rate of 0.001. The mesh grid for the structure was structured as a regular rectangle with a resolution of 0.1 m.

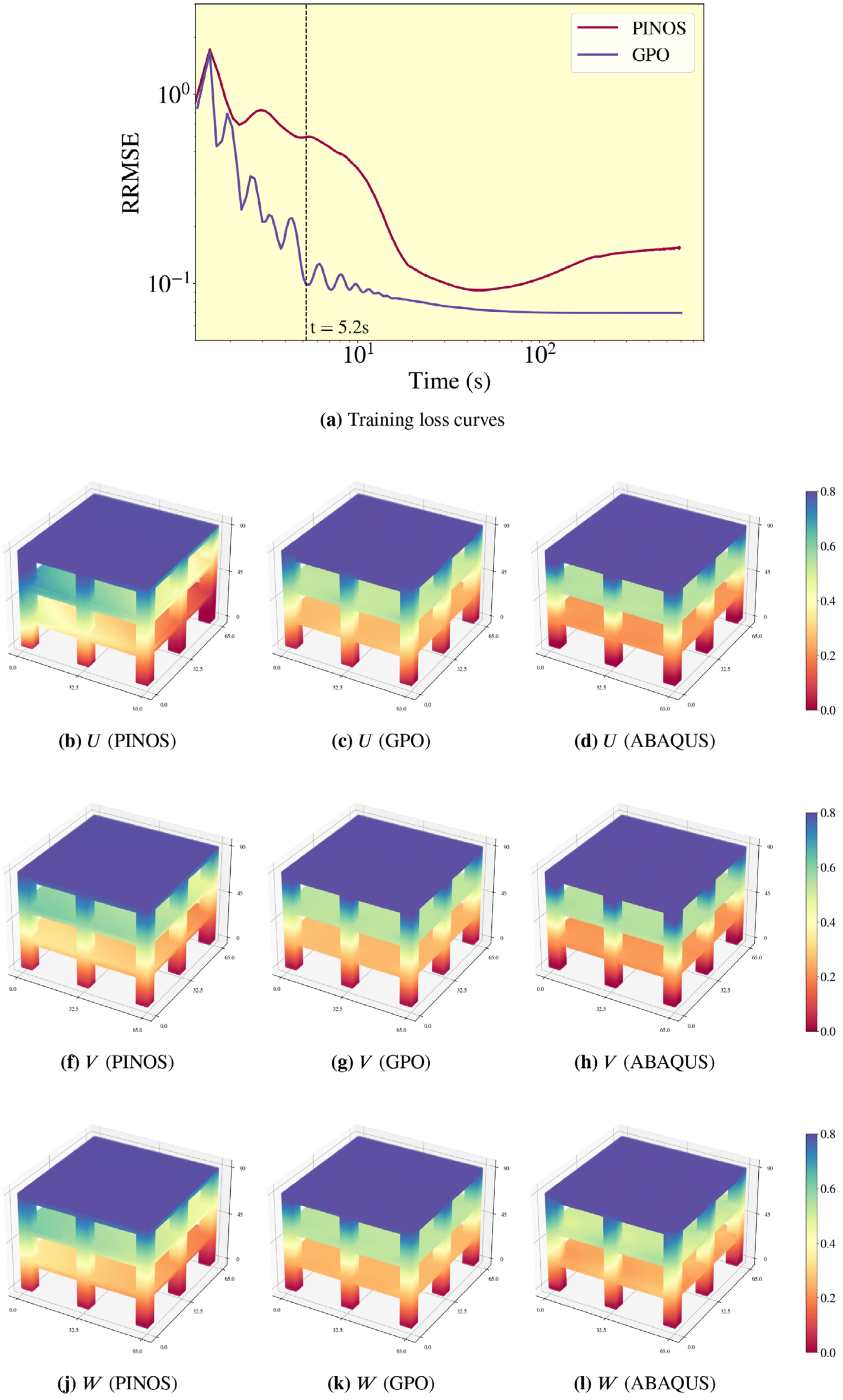
Figure 5 shows a comparative analysis of GPO and PINOS against a reference solution from the numerical solver ABAQUS. Noticeably, the deformation solutions in the x, y, and z directions are well captured in GPO in comparison to the solutions obtained from PINOS. In PINOS, the solution was able to converge after 44 s; however, the RRMSE rose up over 0.1. On the other hand, GPO was able to converge and remained converged after only 5.2 s. The reference solution from ABAQUS was obtained after 7.7 s. Furthermore, it is observed a swing error curve after PINOS converged, while RRMSE of GPO is observed to be gradually reducing and steady. This implies that training PINOS longer may result in a poor performance of the model. Hence, the steady convergence of GPO indicates the high reliability of the model as it ensures the minimum errors of the converged solution. This numerical experiment suggests the potential application of GPO over a commercial numerical solver, namely ABAQUS, in speeding up complex structure examples nearly 2 times.

4.2 | Force constraints

This example aims to obtain the deformation solution of a building loaded with a constant loading on the top boundary. A 100MN y-direction load is applied at the top surface of the object, while the bottom boundary is fixed for all directions and not subjected to any additional external load. We then compared GPO and PINOS in solving the solution to this problem.

Figure 6 illustrates the performance of GPO and PINOS with a reference solution from ABAQUS. The solutions and errors of the deformation in the x, y, and z directions were depicted in the figure. It is observed that PINOS could not converge to an accurate solution with an L2 loss of less than 0.1, but GPO converged within only 11.2 s. The reference-solving time of ABAQUS was nearly 14 s. In addition, the L2 losses in the x and z directions are relatively larger than those in the y direction. This may be caused by the solutions in x and z, which are relatively smaller than y. In fact, the y-direction load does not have a large influence on the deformation of the building in the x-direction, given the elastic property of the building. Therefore, the RRMSE of the solution in the y-direction is more considered in this example. In terms of convergence time, the solution obtained from GPO was as fast as the ABAQUS, suggesting a potential tool for solving static solid mechanics problems.

The previous sections provided quantitative comparisons of GPO and PINOS in solving solutions of deformations under different circumstances. From the numerical experiments, it is sufficient to conclude that GPO with geometry layers and operations in the derivative contributed to achieving a more general solver for more complex geometries in two and three dimensions. The ADF and the geometry layers also help boost the



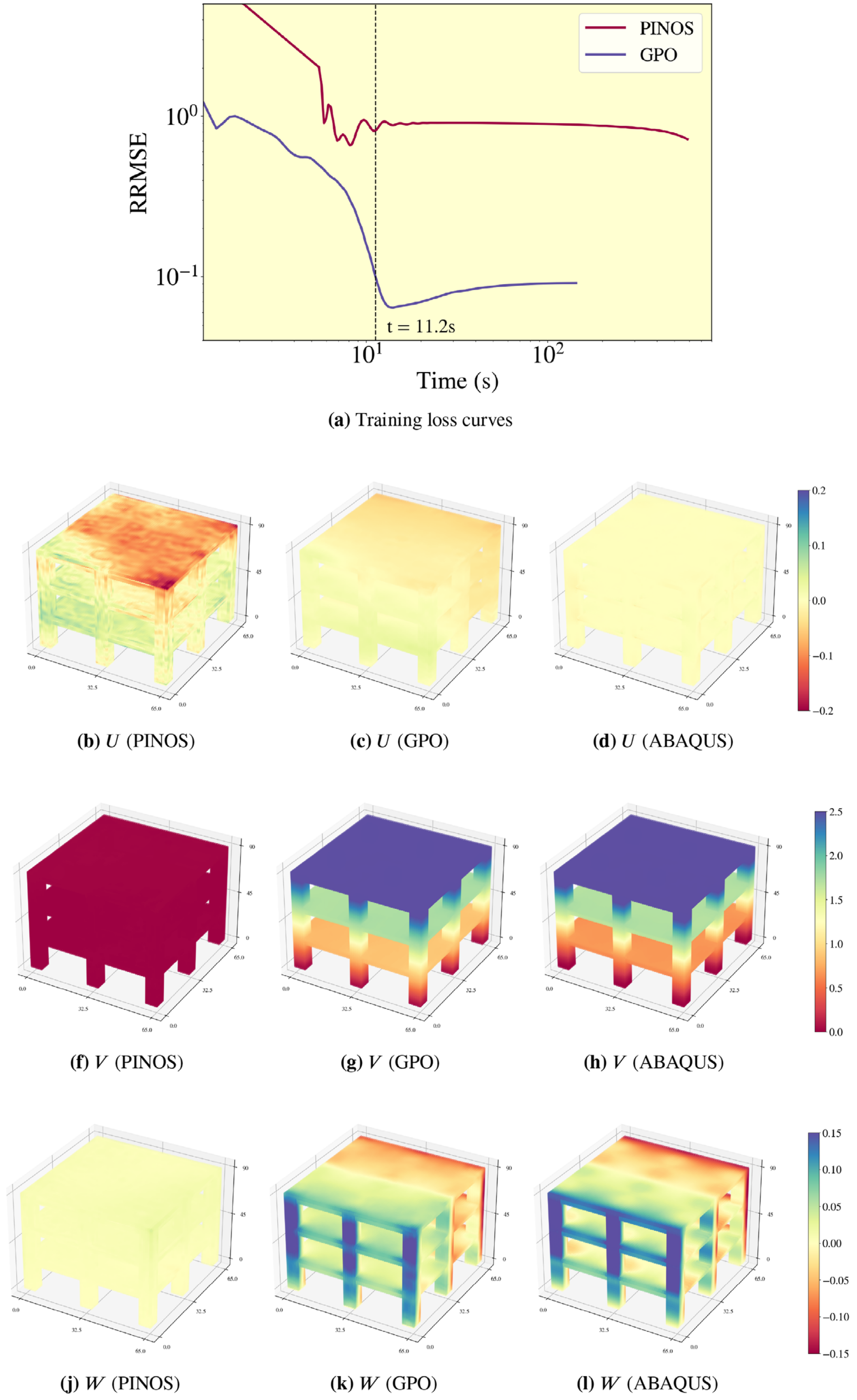
(a) Training loss curves

(b) U (PINOS) **(c)** U (GPO) **(d)** U (ABAQUS)

(f) V (PINOS) **(g)** V (GPO) **(h)** V (ABAQUS)

(j) W (PINOS) **(k)** W (GPO) **(l)** W (ABAQUS)

FIGURE 5 Results of GPO for a three-story building under fixed displacement constraints. U, V, and W denote the displacement in the x, y, and z directions, respectively. The reference solution was obtained from a numerical solver in ABAQUS software with a mesh size of 0.1 m.



(a) Training loss curves

(b) U (PINOS)

(c) U (GPO)

(d) U (ABAQUS)

(f) V (PINOS)

(g) V (GPO)

(h) V (ABAQUS)

(j) W (PINOS)

(k) W (GPO)

(l) W (ABAQUS)

FIGURE 6 Results of GPO for a three-story building under a loading condition on the top surface. U, V, and W denote the displacement in the x, y, and z directions, respectively. The reference solution was obtained from a numerical solver in ABAQUS software with a mesh size of 0.1 m.

TABLE 1 Influences of mesh size on the performance of GPO in both 2D plane and 3D structures under fixed displacement boundaries.

Problems	Mesh size (m)	DOFs	GPO (s)	ABAQUS (s)	RRMSE
2D	1/100	19,680	0.48	1	0.074
	1/200	77,760	1.05	1.1	0.052
	1/400	309,120	3.08	6.9	0.040
	1/1000	1,924,800	35.81	62.9	0.027
	1/2000	7,689,600	100.7	304	0.021
3D	1/10	240,516	5.2	9.3	0.052
	1/20	1,595,295	64.67	134.5	0.074

Abbreviations: 2D, two-dimensional; 3D, three-dimensional; DOFs, degrees of freedom; GPO, Geometry Physics neural Operator; RRMSE, relative root mean square error.

convergence time. In addition to the comparisons between GPO and PINOS, we also demonstrate the convergence time of GPO solving the solution without any labeled data. It is great to emphasize that the inference time of GPO is not the same as the convergence time during the training. In all examples, GPO converging to a solution is faster than ABAQUS solving the solution nearly two times for coarse mesh. As mentioned in Kaewnuratchadasorn et al. (2024), PINOS, an FNO-based solver, highly outperforms ABAQUS in the problem of fine mesh, suggesting the potential of GPO in irregular geometry with fine mesh.

To further observe the computational efficiency of GPO in various mesh sizes, numerical investigations have been conducted on both 2D and 3D cases. In the 2D plane with a hole in a plane stress state, the variation of mesh sizes is shown in Table 1 with the number of degrees of freedom (DOFs) calculated in the ABAQUS software. It is observed that the convergence time of GPO is faster than ABAQUS in all mesh sizes, ranging from 1 to 3 times speed up. In addition, GPO performed better when the mesh was finer as the RRMSE reduced followingly. In the 3D problem, comparisons between GPO and ABAQUS are conducted at only two different mesh sizes. In both cases, solving a system using GPO is faster than solving it with ABAQUS nearly two times. Although convergences of solution were observed in both cases, the RRMSE was observed to be larger when the mesh size decreased.

5 | DISCUSSION

This section discusses the significance of geometry layers in FNO architecture and the contributions of ADF to convergence time. Furthermore, this section includes a discussion of the limitations and potential applications of GPO in the simulations of large-scale structures.

Investigations with respect to two key features of GPO were conducted. The technical investigations focused on the influence of the exact boundary constraints imposed by step-wise functions and the influence of the geometry layer built by the geometry information of the object. As GPO was derived by PINOS for solid mechanics problems by adding the two key features, GPO was examined without each feature. First, PINOS with exact boundary constraints but without a geometry information layer was validated. PINOS with a geometry information layer without exact boundary constraints was evaluated, or the model may be called GPO without exact boundary constraints. These two models are compared with GPO in a plane with a hole in the stress state problem.

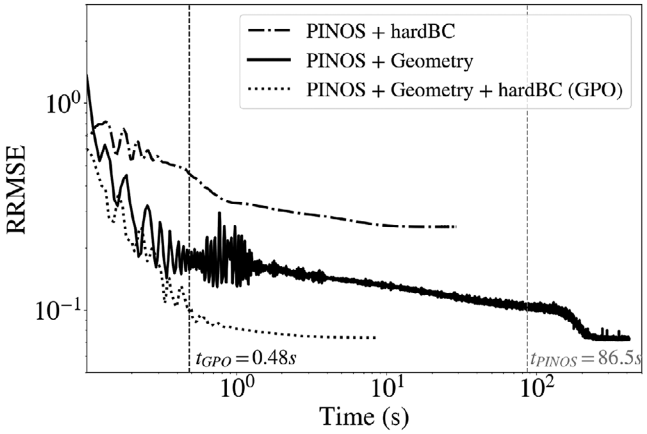


FIGURE 7 Comparison of models in the solution convergence to investigate the contribution of the geometry layer and the constraint boundary function in GPO

Figure 7 illustrates the performances of the mentioned three models. It is observed that PINOS with hard boundary constraints did not converge to an accurate solution and GPO without imposing the hard boundaries took 86.5 s, remarkably longer than GPO (ref: 0.48 s) for a convergence. The results indicated that the geometry layer significantly contributes to the convergence, while the hard boundary functions notably improve speedup. The investigations indicated contributions of geometry layers in the solution convergence and boundary constraint function in the convergence speed.

While the results in Sections 3 and 4 showed excellent performance of GPO in simulations of irregular structures, the current limitations of GPO include the number of DOFs, the regular grid, and the nonlinearity of a complicated structure. GPO cannot solve large-scale simulations that require a large number of DOFs because of the memory limitation. Although GPO has advantages in computational time, the current development considers grid tensor of inputs and outputs, consuming large memory. This limitation may be tackled by sectioning a large-scale structure into smaller partitions with geometry and boundary conditions, yet should be investigated in future development. Furthermore, GPO considers linear elastic materials. Therefore, the development and analysis of GPO for simulations associated with the nonlinearity of the materials should be considered in the prospective development.

For future applications, the zero-shot super-resolution ability was proven in FNO architecture and shown in various research (Rashid et al. 2022). As GPO contains the Fourier layers, the ability will help approximate solutions at fine mesh with training on a coarse mesh. As per Kaewnuratchadasorn et al. (2024), PINOS was able to transfer a solution with sufficient accuracy (within convergence range) to a mesh finer at 2 times. This may imply that GPO can analyze the deformation with smaller units of an irregular object with largely reduced computational cost. Therefore, an application of the super-resolution GPO (SR-GPO) is to solve a large-scale structure at a coarse mesh and transfer to obtain a solution when inferring the solution at a fine mesh. In addition, a limitation of simulating a large-scale structure is the computational cost as the number of degrees of freedom is exceedingly huge. A few recent studies investigate and analyze the transfer learning for accelerating simulations by PINN that transfer to different boundary conditions or geometrical parameters in different fields (Lippert et al. 2024; Y. Wang, Zhang, et al. 2024; Xu et al. 2023). Transfer learning can be adopted and integrated with GPO. After the completion of training of a structure on a large grid, GPO will be able to investigate the structure on a small grid during the inference. In addition, training GPOs aims to achieve the convergence of deformation solutions to a specific problem. As a numerical solver, we may apply GPO to generate structural simulations in different scenarios. A pretrained GPO that converges to a solution in a condition will be further trained to solve for the solution in another condition within the same shape of the object. The training of pretrained GPOs will reduce the convergence time in similar problems, which will greatly save the computational cost for problems requesting a large number of simulations. Further implementation of ML surrogate models may be conducted by cooperating with the training of the pretrained GPO.

6 | CONCLUSIONS

In this study, GPO is introduced to solve differential equations of systems characterized by irregular geometry for efficient structural simulations. This work developed GPO as a solver by PIML and demonstrated its suitability in the modeling of engineering structures. Our training framework regularizes GPO with a weak form of governing equations by minimizing the energy-based loss function of a system in solid mechanics. In addition, exact boundary constraint functions are implemented on GPO to strictly satisfy the Dirichlet boundary condition. In our experiments, the convergence time of GPO is 3× and 2× faster than ABAQUS in 2D and 3D irregular structures under static loads. The relative performance of GPO compared to ABAQUS also increases when the mesh size decreases.

This work contributes to reducing the gap between computational solid mechanics and structural simulations. The quantitative comparisons of performances between GPO, PINOS, and a commercial solver ABAQUS demonstrate that GPO is a potential candidate for an accurate and fast solver for general geometries of structures under different constraints. To the best of our knowledge, GPO achieves the first PIML model with faster simulation compared with commercial numerical solvers in relatively large simulations of structures in static loads as numerous models cannot solve in a large DOF with fast convergence. The achievement of GPO as a solver to forward problems will contribute to structural applications in topological design, analysis, and inverse problems health monitoring. The current development of GPO is limited by the size of the discretized mesh or the number of nodes required in the system. Our framework may be extended to the dynamic studies in structural engineering for applications in fields related to seismic responses. Addressing these extensions is an exciting area for future work in artificial intelligence-integrated structural engineering.

ACKNOWLEDGMENTS

The work in this paper is financially supported by the National Natural Science Foundation of China Project 52408221, Hong Kong Innovation and Technology Support Programme (Mid-stream, theme-based, ITS/041/23MX), and the computing services of Osaka University SQUID supercomputer and HKU HPC2021 cluster. 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

Abueidda, D. W., Koric, S., Guleryuz, E., & Sobh, N. A. (2023). 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

Adeli, H., & Yeh, C. (1989). Perceptron learning in engineering design. *Microcomputers in Civil Engineering*, *4*(4), 247–256.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690. https://doi.org/10.1007/s00521-019-04359-7

Aldwaik, M., & Adeli, H. (2014). Advances in optimization of highrise building structures. *Structural and Multidisciplinary Optimization*, *50*(6), 899–919. https://doi.org/10.1007/s00158-014-1148-1

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

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

Chen, Z., Badrinarayanan, V., Lee, C.-Y., & Rabinovich, A. (2018). *Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks*. arXiv. https://arxiv.org/abs/1711.02257

Cross, E. J., Gibson, S. J., Jones, M. R., Pitchforth, D. J., Zhang, S., & Rogers, T. J. (2021). Physics-informed machine learning for structural health monitoring. In A. Cury, D. Ribeiro, F. Ubertini, & M.D. Todd (Eds.), *Structural health monitoring based on data science techniques* (pp. 347–367). Springer International Publishing. https://doi.org/10.1007/978-3-030-81716-9-17

Faroughi, S. A., Pawar, N. M., Fernandes, C., Raissi, M., Das, S., Kalantari, N. K., & Mahjour, S. K. (2024). Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics. *Journal of Computing and Information Science in Engineering*, *24*(4), 040802. https://doi.org/10.1115/1.4064449

Fu, B., Gao, Y., & Wang, W. (2024). A physics-informed deep reinforcement learning framework for autonomous steel frame structure design. *Computer-Aided Civil and Infrastructure Engineering*, *39*(20), 3125–3144. https://doi.org/10.1111/mice.13276

Gao, H., Hu, G., Zhang, D., Jiang, W., Tse, K. T., Kwok, K. C. S., & Kareem, A. (2024). Urban wind field prediction based on sparse sensors and physics-informed graph-assisted autoencoder. *Computer-Aided Civil and Infrastructure Engineering*, *39*(10), 1409–1430. https://doi.org/10.1111/mice.13147

Goswami, S., Bora, A., Yu, Y., & Karniadakis, G. E. (2022). *Physics-informed deep neural operator networks*. ArXiv. https://doi.org/10.48550/arXiv.2207.05748

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.2021.11374

Hao, Z., Liu, S., Zhang, Y., Ying, C., Feng, Y., Su, H., & Zhu, J. (2023). *Physics-informed machine learning: A survey on problems, methods and applications*. arXiv. https://arxiv.org/abs/2211.08064

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

Huang, Z., Yin, X., & Liu, Y. (2022). Physics-guided deep neural network for structural damage identification. *Ocean Engineering*, *260*, 112073. https://doi.org/10.1016/j.oceaneng.2022.112073

Huerta, A., Belytschko, T., Fernández-Méndez, S., Rabczuk, T., Zhuang, X., & Arroyo, M. (2017). Meshfree methods. In E. Stein, R. Borst, & T. J. R. Hughes (Eds.), *Encyclopedia of computational mechanics* (2nd ed., pp. 1–38). John Wiley & Sons. https://doi.org/10.1002/9781119176817.ecm2005

Hughes, T. (2012). *The finite element method: Linear Static and dynamic finite element analysis*. Dover Civil and Mechanical Engineering. Dover Publications.

Hughes, T., Cottrell, J., & Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, *194*(39), 4135–4195. https://doi.org/10.1016/j.cma.2004.10.008

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., & Adeli, H. (2008). Dynamic fuzzy wavelet neuroemulator for non-linear control of irregular building structures. *International Journal for Numerical Methods in Engineering*, *74*(7), 1045–1066. https://doi.org/10.1002/nme.2195

Kaewnuratchadasorn, C., Wang, J., & Kim, C.-W. (2023). Neural operator for structural simulation and bridge health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *39*(6), 872–890. https://doi.org/10.1111/mice.13105

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*, *39*(22), 3435–3451. https://doi.org/10.1111/mice.13292

Kapoor, T., Wang, H., Núnez, A., & Dollevoet, R. (2024). Physics-informed neural networks for solving forward and inverse problems in complex beam systems. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(5), 5981–5995. https://doi.org/10.1109/tnnls.2023.3310585

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

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, H.-W., Hao, S., Ni, Y.-Q., Wang, Y.-W., & Xu, Z.-D. (2024). Hybrid structural analysis integrating physical model and continuous-time state-space neural network model. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13282

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. https://openreview.net/forum?id=c8P9NQVtmnO

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. Paper presented at the Thirty-Seventh Conference on Neural Information Processing Systems. https://openreview.net/forum?id=86dXbqT5Ua

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. arXiv. https://arxiv.org/abs/2111.03794

Liao, Y., Zhang, R., Wu, G., & Sun, H. (2023). A frequency-based ground motion clustering approach for data-driven surrogate modeling of bridges. *Journal of Engineering Mechanics*, *149*(9), 04023069. https://doi.org/10.1061/JENMDT.EMENG-6812

Lippert, J. R., von Tresckow, M., De Gersem, H., & Loukrezis, D. (2024). Transfer learning-based physics-informed neural networks for magnetostatic field simulation with domain variations. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, *37*(4), e3264. https://doi.org/10.1002/jnm.3264

Malekloo, A., Ozer, E., AlHamaydeh, M., & Girolami, M. (2022). Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, *21*(4), 1906–1955. https://doi.org/10.1177/14759217211036880

Meng, Z., Qian, Q., Xu, M., Yu, B., Yıldız, A. R., & Mirjalili, S. (2023). PINN-form: A new physics-informed neural network for reliability analysis with partial differential equation. *Computer Methods in Applied Mechanics and Engineering*, *414*, 116172. https://doi.org/10.1016/j.cma.2023.116172

Miao, Y., Kang, H., Hou, W., Liu, Y., Zhang, Y., & Wang, C. (2024). A response-compatible ground motion generation method using physics-guided neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *39*(15), 2350–2366. https://doi.org/10.1111/mice.13194

Moradi, S., Duran, B., Azam, S. E., & Mofid, M. (2023). Novel physics-informed artificial neural network architectures for system and input identification of structural dynamics PDEs. *Buildings*, *13*(3), 650. https://doi.org/10.3390/buildings13030650

Sanni-Anibire, R. M. Z. M. O., & Olatunji, S. O. (2022). Machine learning model for delay risk assessment in tall building projects. *International Journal of Construction Management*, *22*(11), 2134–2143. https://doi.org/10.1080/15623599.2020.1768326

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMA: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404. https://doi.org/10.1007/s00521-019-04146-4

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

Plevris, V., & Tsiatas, G. C. (2018). Computational structural engineering: Past achievements and future challenges. *Frontiers in Built Environment*, *4*, 21. https://doi.org/10.3389/fbuil.2018.00021

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083. https://doi.org/10.1109/TNNLS.2017.2682102

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2024). Self-supervised learning for near-wild cognitive workload estimation. *Journal of Medical Systems*, *48*(1), 107. https://doi.org/10.1007/s10916-024-02122-7

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

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

Song, L.-H., Wang, C., Fan, J.-S., & Lu, H.-M. (2023). Elastic structural analysis based on graph neural network without labeled data. *Computer-Aided Civil and Infrastructure Engineering*, *38*(10), 1307–1323. https://doi.org/10.1111/mice.12944

Sukumar, N., & Srivastava, A. (2022). Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, *389*, 114333. https://doi.org/10.1016/j.cma.2021.114333

Sun, H., Burton, H. V., & Huang, H. (2021). Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering*, *33*, 101816. https://doi.org/10.1016/j.jobe.2020.101816

Taghizadeh, M., Nabian, M. A., & Alemazkoor, N. (2024). Multifidelity graph neural networks for efficient and accurate mesh-based partial differential equations surrogate modeling. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13312

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

Wang, G., Fang, Q., Wang, J., Li, Q. M., Chen, J. Y., & Liu, Y. (2024). Estimation of load for tunnel lining in elastic soil using physics-informed neural network. *Computer-Aided Civil and Infrastructure Engineering*, *39*(17), 2701–2718. https://doi.org/10.1111/mice.13208

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

Wang, R., & Yu, R. (2023). Physics-guided deep learning for dynamical systems: A survey. arXiv. http://arxiv.org/abs/2107.01272

Wang, Y., Zhang, H., Lai, C., & Hu, X. (2024). Transfer learning Fourier neural operator for solving parametric frequency-domain wave equations. *IEEE Transactions on Geoscience and Remote Sensing*, *62*, 1–11. https://doi.org/10.1109/TGRS.2024.3440199

Xu, C., Cao, B. T., Yuan, Y., & Meschke, G. (2023). Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios. *Computer Methods in Applied Mechanics and Engineering*, *405*, 115852. https://doi.org/10.1016/j.cma.2022.115852

Yeung, Y.-H., Barajas-Solano, D. A., & Tartakovsky, A. M. (2022). Physics-informed machine learning method for large-scale data assimilation problems. *Water Resources Research*, *58*(5), e2021WR031023. E2021WR031023 2021WR031023. https://doi.org/10.1029/2021WR031023

Esteghamati, M. Z., & Flint, M. M. (2021). Developing data-driven surrogate models for holistic performance-based assessment of mid-rise RC frame buildings at early design. *Engineering Structures*, *245*, 112971. https://doi.org/10.1016/j.engstruct.2021.112971

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, R., Liu, Y., & Sun, H. (2020). Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, *215*, 110704. https://doi.org/10.1016/j.engstruct.2020.110704

How to cite this article: Kaewnuratchadasorn, C., Wang, J., Kim, C.-W., & Deng, X. (2025). Geometry physics neural operator solver for solid mechanics. *Computer-Aided Civil and Infrastructure Engineering*, 1–17. https://doi.org/10.1111/mice.13405