

A Physics-Informed Neural Network-based Topology Optimization (PINNTO) framework for structural optimization

Hyogu Jeong^{a,1}, Jinshuai Bai^{a,1}, C.P. Batuwatta-Gamage^a, Charith Rathnayaka^{a,b}, Ying Zhou^{c,*}, YuanTong Gu^{a,*}

^a School of Mechanical, Medical and Process Engineering, Queensland University of Technology, Brisbane, QLD 4000, Australia

^b School of Science, Technology and Engineering, University of the Sunshine Coast, Petrie, QLD 4502, Australia

^c State IJR Center of Aerospace Design and Additive Manufacturing, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

ARTICLE INFO

Keywords:

Topology optimization
Physics informed neural network
Machine learning
Solid mechanics

One of the newest deep learning based works from 2023 - it moves away from the SIMP method and utilizes energy based methods

ABSTRACT

Physics-Informed Neural Networks (PINNs) have recently attracted exponentially increasing attention in the field of computational mechanics. This paper proposes a novel topology optimization framework: Physics-Informed Neural Network-based Topology Optimization (PINNTO). Unlike existing machine-learning based topology optimization frameworks, PINNTO employs an energy-based PINN to replace Finite Element Analysis (FEA) in the conventional structural topology optimization, to numerically determine the deformation states, which is a key novelty in the proposed methodology. A supervised neural network that respects governing physical laws defined via partial differential equations is trained to develop the corresponding network without any labelled data, with the intention of solving solid mechanics problems. To assess feasibility and potential of the proposed PINNTO framework, a number of topology-optimization-related case studies have been implemented. The subsequent findings illustrate that PINNTO has the ability to attain optimized topologies with neither labelled data nor FEA. In addition, it has the capability to generate comparable designs to those produced by the current successful approaches such as Solid Isotropic Material with Penalization (SIMP). Based on the results of this study, it can also be deduced that PINNTO can acquire optimal topologies for various types of complex domains given that the boundary conditions and loading configurations are correctly imposed for the associated energy-based PINN. Consequently, the proposed PINNTO framework has demonstrated promising capabilities to solve problems under conditions when the usage of FEA is challenged (if not impossible). In summary, the proposed PINNTO framework opens up a new avenue for structural design in this ‘data-rich’ age.

1. Introduction

Topology optimization (TO) has traditionally been used for a broad spectrum of design applications due to the capability of generating least-weight and high-performance structures for industrial designs [1,2]. It has been recognized as an effective method for finding the optimal material distribution within a prescribed problem domain by maximizing or minimizing the targeted objective function under specific design constraints [3–6]. Ever since the seminar paper published by Bendsøe and Kikuchi using the homogenization method [7], TO has been gaining remarkable attention altogether with theoretical and practical developments [8–11] among the Finite Element Analysis (FEA) researcher-community, leading to theoretical and practical

developments [8–11]. Accordingly, most TO problems are inherently devoted to finding the optimal topologies of two-dimensional (2-D) linear elastic structures where structural elements are regarded as porous materials at the microscopic level [12]. The different variations of TO such as Solid Isotropic Material with Penalization (SIMP) [13], Evolutionary Structural Optimization (ESO) [9], Level Set Method (LSM) [10], Homogenization method [7] have subsequently granted prosperity to the structural optimization field [14]. Although these methods have contributed significantly to the advancement of the field, there are existing challenges due to the inherent high-dimensionality of typical TO problems constrained by PDEs and inequalities [15]. These challenges often lead to massive computational costs, especially in high-resolution domains of continuum structures, due to the highly iterative

* Corresponding authors.

E-mail addresses: yingzhou@mail.nwpu.edu.cn (Y. Zhou), yuantong.gu@qut.edu.au (Y. Gu).

¹ These authors contributed equally to this work and should be considered co-first authors.

nature of TO processes [16].

Recent advancements in computational mechanics integrated with Artificial Intelligence (AI) have contributed to a paradigm shift in the field of TO towards a machine-learning-based intelligent platform, where AI draws insights from existing experimental and computational data. Such data-driven techniques have made research more efficient and begun to complement traditional statistical methods [17,18].

The standard form of data-driven topology optimization (DDTO) is to develop a predictive model which establishes a function that maps input (physical fields and loading configuration) to labelled output data (optimal topology). One example is to train a feed-forward neural network (FFNN) with a large dataset to establish a relationship between the loading configurations and the optimal topologies [19]. Subsequent studies have followed a similar trend, incorporating a more complex, deep convolutional neural network (DCNN) to generate accurate and generalizable real-time predictions of the resulting optimal topology at a lower computational cost [4–6,16,20–22]. Exemplary research done by Sosnovik and Oseledets [5] has developed an image segmentation DCNN with U-Net architecture to establish a mapping between the tensor containing early iteration result x_n and the gradient ∂x for the final optimized solution with binary classes for solid and void materials. Based on this work, Yu et al. [4] introduced a concept to describe boundary conditions as a matrix for the training of the CNN encoder-decoder model, leading to the advantage of employing CNN as an alternative to vector data usually utilised with machine learning-based models. This idea of transforming loads and boundary conditions to a matrix input has influenced the later DDTO studies relying on CNN. Subsequently, a second deep learning model, conditional generative adversary network (CGAN) is trained to refine near-optimal solutions into the higher resolution space which attains high-frequency details at a larger size. The prior work has influenced Banga et al. [20] to explore various input combinations for three-dimensional (3-D) topology optimization using a convolutional neural network (CNN). The information used for their models is the combinations of density, density gradient, boundary condition and the loading configurations represented as a matrix to evaluate the effectiveness of each input for the model performance. They have concluded that density data distribution sampled using Poisson distribution provides the best model performance. Posterior works in the same paradigm have explored different approaches to solve topology optimization. Abueidda et al. [21] have solved the material's linear elastic and hyperelastic responses for solving TO problems using a deep CNN encoder-decoder model. Their work has investigated large-and-small-deformation scenarios with and without the stress constraint. Zhang et al. [23] have proposed a trained CNN model with U-Net architecture that has a strong generalization ability for TO. Such a method can solve TO with different boundary condition problems in a negligible time, given that the model is trained solely on one boundary condition in the training dataset. They have achieved this by employing a tensor in the training dataset containing normal and shear strain fields to estimate the layout of the optimal structure.

Despite the great success witnessed in the DDTO field, some significant disadvantages remain. The DDTO approach requires sufficient neural network (NN) training that relies on a large set of sampled data. Also, the model's capability is strictly limited to the information stored within the dataset [18]. This leads to a significant drawback of generating inaccurate predictions for novel boundary condition problems due to the poor generalization of the NN framework [24,25]. In addition, the predicted topologies obtained via the trained model comprising pixel-wise errors are inevitably less effective than the actual result as they usually contain prediction errors that induce blurry edges, disconnected elements, and irregular deformations. Consequently, the predicted structural designs will have high compliance errors, despite having a low prediction error [26]. The DDTO approach requires more in-depth exploration prior to 'real-world' applications under real circumstances.

A recent publication by Raissi, et al. [27] has developed a deep learning-based PDE solver for forward and inverse computational

problems, called Physics-Informed Neural Networks (PINNs). This method has the ability to solve governing PDEs in the absence of labelled data through minimization of the PDE residual, initial conditions (IC) and boundary conditions (BC) using automatic differentiation with multi-tasking capabilities in NN. The capabilities of this novel approach have been illustrated for nonlinear PDEs such as Schrodinger, Allen-Cahn and Navier-Stokes [27,28]. The subsequent PINN-based investigations have demonstrated that it is possible to yield accurate predictions in the absence of comprehensive datasets for fluid mechanics [29], solid mechanics [30] and elastodynamics [31]. Consequently, PINN has received considerable attention in both the industry and academia while the availability of open-source, deep learning libraries such as Tensorflow and Pytorch [32] have motivated AI and computational mechanics researchers to develop novel PINN-based frameworks to solve PDEs [28,33]. In particular, Python libraries like SciANN [34], IDRLnet [35], SimNet [36] and DeepXDE [37] have been developed to make PINN more accessible and convenient for researchers. Through all these success-stories, PINN has emerged as a highly viable tool for complementing traditional scientific and engineering approaches with respect to computational mechanics [38].

Motivated by the recent success of PINN, novel TO methods using PINN have been developed as alternative approaches to traditional TO solvers [14,24,25]. The update of design variables in traditional TO has been replaced by the reparameterization of NN parameters [24]. The method leverages automatic differentiation in deep neural networks (DNNs) to replace conventional sensitivity analyses with a deep learning backpropagation problem. Chandrasekhar and Suresh [14] have proposed topology optimization using a NN (TOuNN), where the density function is parameterized via NN parameters (i.e., weights and bias). In particular, the conventional finite element (FE) solver has been implemented as an unconstrained NN loss to compute elemental stiffness and displacement. The displacement and the global stiffness matrix have then been used to calculate the total compliance as the NN's training objective that is backpropagated to update NN parameters. Chen and Shen [25] proposed a physics-informed deep learning process to solve density-based topology designs. Unlike the conventional DDTO method where the network's loss is calculated based on similarities, their method update PINN using gradient information from FEA. Zhang, et al. [24] have carried out further investigations by introducing Topology Optimization via Neural Reparameterization (TONR) based on Inverting Representation of Image (IRI), which reconstructs an image based on the image feature. In this research, TONR combines TO, IRI, and PINN to reconstruct the optimal topology based on features (compliance) that have been previously computed using a numerical FE solver. This approach has established a novel TO framework overcoming the limitations of the DDTO approach. It does not require training data to train the NN and generates feasible topology designs with performance comparable with the conventional TO methods. TONR also provides great flexibility for solving various optimization problems with the aid of automatic differentiation in NN that can handle differential operators. However, such methods still do not respond to the core purpose of combining deep learning techniques and TO, which is to develop a more efficient method for high-resolution TO problems that is also computationally intractable [39].

For the traditional FEA-based TO methods, the design of material geometry with desired functionality is very challenging due to the high dimensionality of the design space [40]. The PDE-constrained topology optimization using numerical PDE solvers (such as FEM) is prone to slow and decimal point precision errors, especially in the high dimensional domain [41]. In addition, FEA-based TO has inherent shortcomings, including the difficulty of solving TO of problems with strong nonlinearity and complex materials as well as significant challenges under dynamic loading and subsequent fracture. In addition, geometrically nonlinear TO methods suffer considerably from excessive distortion of void elements denoted by low stiffness [24,42]. As a result, it causes non-convergence of TO where the stiffness matrix become indefinite or

negative-definite [43].

Motivated by the aforementioned considerations, this paper explores a novel TO framework that employs PINN to replace the associated FEA component. This Physics-Informed Neural Network-based Topology Optimization (PINNTO) differs from previous traditional, data-driven and PINN approaches as this does not rely on FEA nor large datasets. Instead, the proposed method utilizes PINN to solve strain energy criterion in conjunction with binary density fields. SIMP optimization algorithm is adopted to add or remove materials systematically, so the residual of the structure evolves towards an optimum [12]. To evaluate the feasibility of the proposed PINNTO framework for solving TO problems under various settings, a number of detailed case studies have also been conducted.

This paper is organized as follows. **Section 2** introduces the workflow of the conventional SIMP algorithm, and the architecture of the energy-based PINN. **Section 3** explains the novel PINNTO setup in detail with strategies of incorporating energy-based PINN into SIMP. To illustrate the effectiveness and the potential of the proposed PINNTO framework, several case studies have been presented under **Section 4**. Finally, **Section 5** concludes the paper with a summary of key findings of the proposed computational framework, limitations and recommendations for future work.

2. Introduction of energy-based PINN and SIMP

This section provides fundamental details of the conventional SIMP algorithm and energy-based PINN framework as they are critical for developing and understanding the proposed PINNTO framework.

2.1. Energy-based PINN framework for solid mechanics

2.1.1. The principle of minimum potential energy for solid mechanics

Consider a 2-D solid mechanics problem in the domain Ω . The displacement solution of the problem is denoted as u . The total potential energy of the problem, $\Pi(u)$, is a function of u . The total potential energy can be calculated as follows,

$$\Pi(u) = E_{\text{in}} + E_{\text{ex}}, \quad (1)$$

where E_{in} is the internal strain energy and E_{ex} is the potential energy corresponding to the external forces. The internal strain energy can be calculated as,

$$E_{\text{in}} = \int_{\Omega} \frac{1}{2} \sigma_{\alpha\beta} \epsilon_{\alpha\beta} dV, \quad (2)$$

where σ and ϵ are the Cauchy stress and strain for the problem, respectively. The subscripts α , β , and γ refer to the coordinate directions. The strain can be calculated as follows,

$$\epsilon_{\alpha\beta} = \frac{1}{2} \left(\frac{\partial u_{\alpha}}{\partial x_{\beta}} + \frac{\partial u_{\beta}}{\partial x_{\alpha}} \right), \quad (3)$$

For linear elastic problems, the Cauchy stress can be obtained through the constitutive equation,

$$\sigma_{\alpha\beta} = \lambda \delta_{\alpha\beta} \epsilon_{\gamma\gamma} + 2\mu \epsilon_{\alpha\beta}, \quad (4)$$

where $\delta_{\alpha\beta}$ is the Kronecker delta function. Note that the Einstein summation convention has been applied. λ and μ are the Lamé constants. For the plain stress problem, the Lamé constants can be calculated by,

$$\begin{cases} \lambda = \frac{E\nu}{(1+\nu)(1-\nu)}, \\ \mu = \frac{E}{2(1+\nu)}, \end{cases} \quad (5)$$

where ν is the Poisson's ratio. Besides, the potential energy corresponding to the external forces can be written as,

$$E_{\text{ex}} = - \int_{\partial\Omega} \bar{u} T dA, \quad (6)$$

where T is the external force vector applied on the boundary, $\partial\Omega$, and \bar{u} is the displacement at the corresponding boundary. Referring to the principle of minimum potential energy [44], the solution of the solid mechanics problem, u , is the function that minimizes the total potential energy,

$$\min_u \Pi(u). \quad (7)$$

Hence, the actual displacement is attained at the minimum potential energy where the displacement solution is obtained by finding the stationary value of the total potential energy.

2.1.2. Physics-informed neural networks for minimizing the overall potential energy

To solve the solid mechanic problems through PINNs, an FFNN is used to approximate the displacement field as shown in Fig. 1.

The network is composed of a series of fully connected layers: an input layer; hidden layers; and an output layer. The expression of the network is given as,

$$u = N(x, y, W, b), \quad (8)$$

where x, y are the spatial coordinate variables and W, b are weights and biases in the subsequent hidden layers to process the input data through forward propagation [45]. The output layer then generates displacements (u_1, u_2) with respect to the inputs. Hence, the information from the input to the output layer is forward propagated through the following nested transformation,

$$z_i = \Phi(W_i z_{i-1} + b_i), \quad i = 1, \dots, L, \quad (9)$$

where L is the number of fully connected layers, z_0 is the input and z_L is the output of the FFNN. W_i and b_i are the weights and bias parameters of the i^{th} layer respectively. Φ is the activation function to apply nonlinearity to the network [46]. For the PINN, the hyperbolic tangent (Tanh) activation function is used for all hidden layers.

$$\Phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (10)$$

Hence the forward propagation of the network can be written as,

$$\begin{aligned} z_1 &= \Phi(W_1 z_0 + b_1); \\ z_2 &= \Phi(W_2 z_1 + b_2); \\ &\dots \\ z_L &= \Phi(W_L z_{L-1} + b_L). \end{aligned} \quad (11)$$

The built network is trained through the physics-guided loss function. Herein, the loss function is formulated based on the overall potential energy,

$$\mathcal{L} = \Pi(u) \quad (12)$$

It should be noted that the partial differential terms for the overall potential energy in the loss function are analytically obtained through automatic differentiation in the NN [44]. During the training process, the weights and biases are iteratively reparametrized to find the minimum value of the overall potential energy.

An example problem using energy-based PINN is demonstrated using a plate of $1 \text{ m} \times 1 \text{ m}$ domain which is subjected to a load and boundary conditions as depicted in Fig. 2. The square cantilever beam is fixed from one end and subjected to a traction force $T = 0.5 \text{ N}$ is applied at the other end.

The displacement fields obtained through PINN, FEA and their errors are illustrated in Fig. 3. The predicted displacements fields u_1^*, u_2^* from the energy-based PINN closely follow the corresponding FEA references with very small absolute errors for each displacement components of the solution. More detailed explanations and numerical implementations of

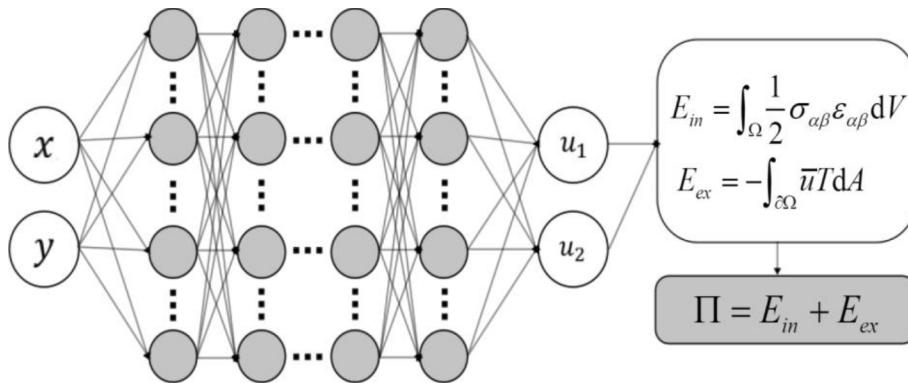


Fig. 1. Schematic diagram of energy-based PINN with spatial coordinates x, y as inputs and displacements u_1, u_2 as outputs with total potential energy as loss function.

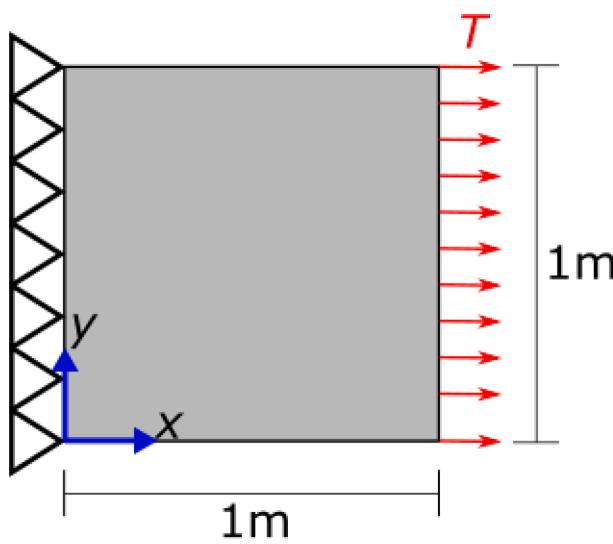


Fig. 2. Problem setup.

the energy-based PINN method can be found in [44].

2.2. Solid Isotropic Material with Penalization (SIMP) for intermediate densities

SIMP is a density-based topology optimization method, that aims to find an optimum design under specified volume constraints [19,47]. In the SIMP approach, the problem domain is generally discretised into square finite elements, ρ_i ($i = 1, \dots, N$). The design variable, ρ_i , is introduced to represent the proportion of the material within the corresponding element. The mechanical properties of the element are obtained using a power-law interpolation function [13]. The power-law penalizes the intermediate densities which drives the elemental densities to converge towards 0–1 solution. The mathematical formulation of SIMP for the compliance minimization problem is described as follows:

$$\begin{aligned} \min : C(\rho) &= \mathbf{U}^T \mathbf{K} \mathbf{U}, \\ \text{s.t.} : \mathbf{K} \mathbf{U} &= \mathbf{F}, \\ V(\rho)/V_0 &= f, \\ 0 \leq \rho_i \leq 1, i &= 1 \dots N, \end{aligned} \quad (13)$$

where C is the mean compliance, \mathbf{U} is the global displacement vector, \mathbf{F} is the external force vector and \mathbf{K} is the global stiffness matrix. f is the

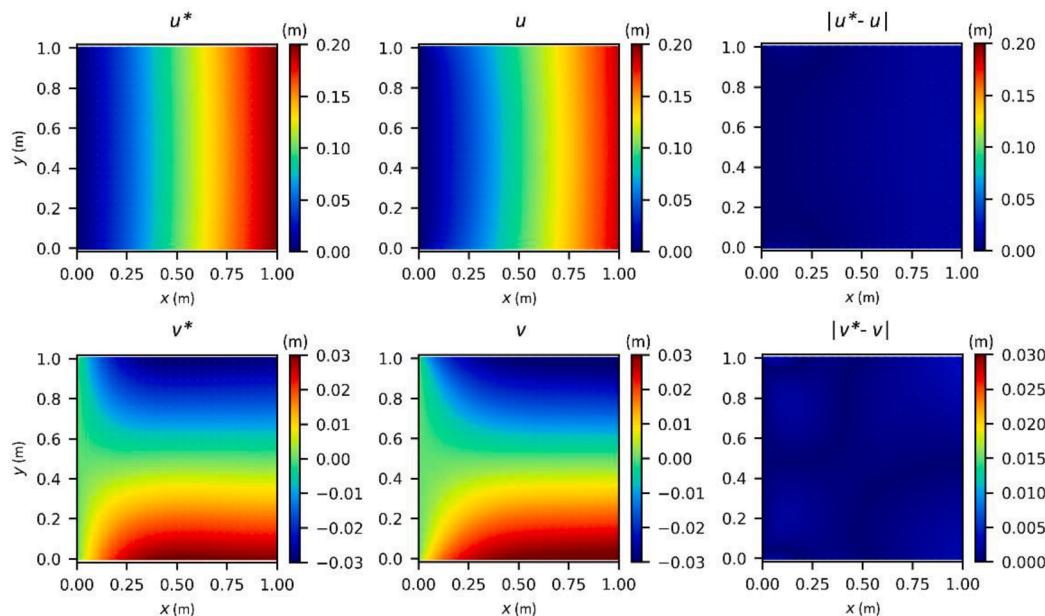


Fig. 3. PINN solution of displacement fields u_1^*, u_2^* against FEA references u_1, u_2 .

prescribed volume fraction, $V(\rho)$ and V_0 are the material and design space volume, respectively. Besides, the density of the given element and Young's modulus of the corresponding element have the following heuristic relationship for the modified SIMP approach [48]:

$$E_i(\rho_i) = E_{\min} + (E_0 - E_{\min})\rho_i^p, \quad (14)$$

where E_0 denotes Young's modulus of the solid material and E_{\min} is the very small stiffness assigned to the void elements to avoid the stiffness matrix from becoming singular. p is the penalization factor to encourage 0 and 1 solutions of the design variables. Generally, $p = 3$ is selected as the penalization factor [49].

The update of the design variables in conventional TO depends on the sensitivity number, which is the first-order derivative of the objective function with respect to the design variable [50,51]. When i^{th} element is removed, the change of the total strain energy (mean compliance) is equivalent to the elemental sensitivity number ΔC_i :

$$\frac{\partial c}{\partial \rho_i} = -p(\rho_i)^{p-1}(E_0 - E_{\min})\mathbf{u}_i^T \mathbf{k}_e \mathbf{u}_i, \quad (15)$$

where \mathbf{u}_i is the elemental displacement vector and \mathbf{k}_e is the elemental stiffness matrix. The material volume V with respect to the densities is described as:

$$\frac{\partial V}{\partial \rho_i} = 1, \quad (16)$$

where Eq. (16) is equal to 1 based on the assumption that each element consists of unit volume.

A subsequent filter scheme is adopted for elemental sensitivities to suppress checkerboard problems [3,52]. In the filter scheme, the nodal sensitivity numbers are modified based on the neighbouring elements within the circle of the sub-domain under a prescribed filter radius r_{\min} [49]. Hence, the neighbouring sensitivity numbers are smoothed by the filtering technique as follows [53]:

$$\widehat{\frac{\partial c}{\partial \rho_i}} = \frac{1}{\max(\gamma, \rho_i) \sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} \rho_i \frac{\partial c}{\partial \rho_i}, \quad (17)$$

where N_e is the set of elements in which, the distance between the $\Delta(e,i)$ and element e is less than the filter radius r_{\min} . γ is a small positive value to avoid zero division. H_e is the weight factor that is defined as the following:

$$H_e = \max(0, r_{\min} - \Delta(e, i)). \quad (18)$$

The compliance minimization problem in Eq. (13) is solved using the Optimality Criteria (OC) method that is a heuristic material updating scheme described as follows [13]:

$$\rho_i^{\text{new}} = \begin{cases} \max(0, \rho_i - m), & \rho_i B_i^\eta \leq \max(0, \rho_i - m), \\ \min(1, \rho_i + m), & \rho_i B_i^\eta \geq \min(1, \rho_i + m), \\ \rho_i B_i^\eta, & \text{otherwise.} \end{cases} \quad (19)$$

Where m is a positive move limit, $\eta = 0.5$ is a numerical damping coefficient, and B_e is calculated using optimality condition described as:

$$B_e = \frac{-\frac{\partial c}{\partial \rho_i}}{\lambda \frac{\partial V}{\partial \rho_i}}, \quad (20)$$

where the appropriate Lagrangian multiplier λ is selected using the bisection method to satisfy the prescribed volume constraints f . The detailed implementation of SIMP can be found in [48,52].

3. Physics-Informed Neural Network based Topology Optimization (PINNTO) framework

This section presents the fundamentals of the PINNTO framework. In

addition, the specifics of the corresponding numerical implementation have also been delineated.

3.1. PINN for topology optimization

PINNTO is a combination of PINN and SIMP, which have been discussed under Section 2 of this paper. Herein, energy-based PINN is selected among other approaches like PDE-based as it is computationally more efficient while achieving satisfactory accuracy of actual FEA simulation [44]. Moreover, considering the change of the topology design during the optimization process, the energy-based model can easily deal with newly generated or disappeared boundaries. Hence, in such manner, we choose energy-based PINN in the proposed PINNTO framework. For the sake of clarity and simplicity, we elucidate the process of PINNTO using a typical linear elastic TO problem.

3.1.1. PINN dataset and density interpolation scheme

Prior to the optimization process, the given computational domain is firstly discretised into square elements with corresponding sample points as illustrated in Fig. 4. These sample points are generated to form 2-D spatial coordinates of the mesh, which are used to calculate the overall potential energy of the given structure. Unlike DDTO, generating a large dataset to train NN is not necessary for PINN as training is solely relied on the governing physics equations that are embedded in the loss function. Instead, sampling the spatial coordinates of the computational domain x_i, y_i are sufficient for the network input [44]. Each sample point contains density, ρ_i^* that is later used to compute Young's modulus at the point.

Subsequently, the density at each sample point is obtained via a density interpolation scheme as shown in Fig. 5.

The proposed density interpolation scheme needs to be adopted to allocate elemental density at each sample point ρ_i^* . Such a scheme utilizes zero padding and max pooling operation to the design domain to evaluate densities at each sample point. This allows mesh-free characteristics of the energy-based PINN domain defined through uniformly distributed sample points to interact with surrounding elemental densities ρ_i in the PINN computational domain [54].

3.1.2. PINNTO optimization

In PINNTO, the minimization of mean compliance is used as the objective function, and the other design constraints in the volume fraction and density in SIMP remain the same. Therefore, minimizing the operation of the mean compliance for PINNTO can be expressed as,

$$\begin{aligned} \min : C(\rho) &= \mathbf{U}^T \mathbf{K} \mathbf{U}; \\ \text{s.t.} : \mathbf{K} \mathbf{U} &= \mathbf{F}, \\ V(\rho)/V_0 &= f, \\ 0 \leq \rho_i \leq 1, & \quad i = 1 \dots N \end{aligned} \quad (21)$$

where C is the objective in terms of the strain energy. FFNN is used to

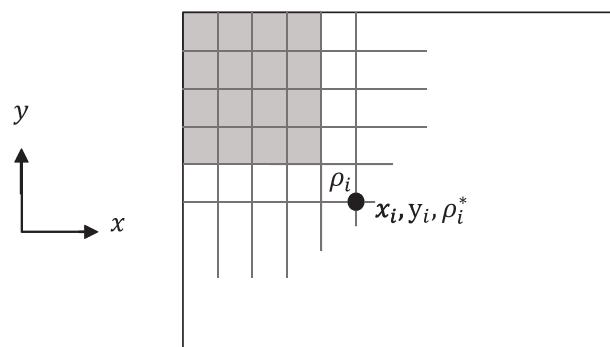


Fig. 4. A 2-D computational domain discretised into sample points x and y with corresponding density ρ where i indicates the element number.

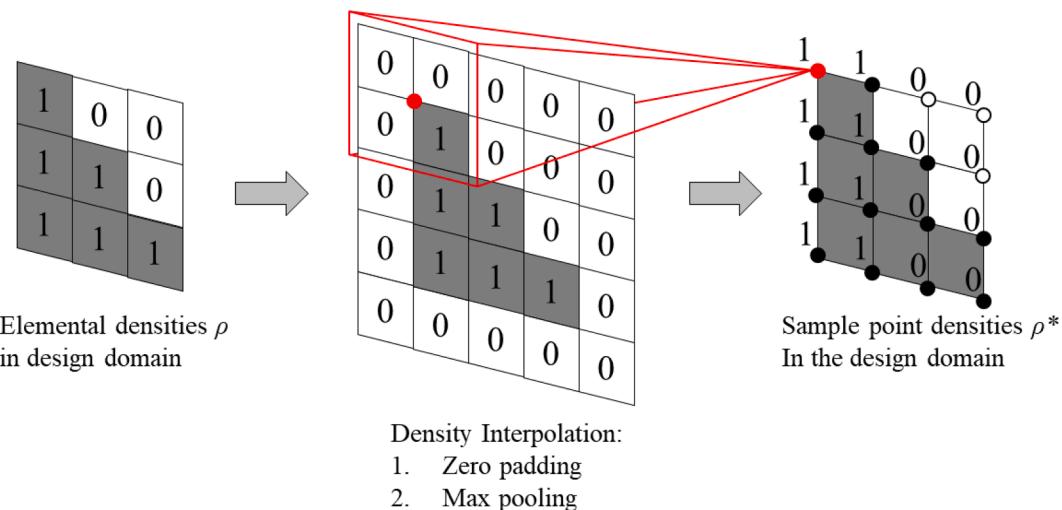


Fig. 5. Density interpolation scheme. The elemental densities in the design domain are transformed into sample point densities using the max pooling operation.

obtain the displacement fields of the current design through PINN within a given iteration. The displacement fields are used to calculate the stress and strain of the current design are calculated through Eq. (3) and Eq. (4). Note that the material interpolation scheme from SIMP as in Eq. (14) is applied to relate the density ratio with the corresponding Young's modulus.

Subsequent sensitivity analysis and sensitivity filtering are evaluated using the approximated displacement field obtained through energy-based PINN as described in Eq. (15) and (17) respectively. The following optimality criterion process expressed in Eq. (19) determines the elemental densities for the next topology optimization iteration. This optimization process is repeated until the topology converges with less than a 1 % change in optimization. The process flowchart of PINNTO is shown in Fig. 6.

3.2. Numerical implementation

The loss function in PINN is constructed based on the overall potential energy that is numerically integrated to calculate Eq. (1). This energy integration is approximated through the means of energy densities of sample points inside the computational domain. For numerical integration, the strain energy of a given sample point is calculated as,

$$E_{in}(\rho_i^*) = \frac{1}{2} \varepsilon_{i,\alpha\beta} \sigma_{i,\alpha\beta} S_i, \quad (22)$$

where S_i is the mean area of the corresponding collocation point. Hence, the overall strain energy of the design is,

$$E_{in} = \sum_i E_{in}(\rho_i^*) = \sum_i \frac{1}{2} \varepsilon_{i,\alpha\beta} \sigma_{i,\alpha\beta} S_i. \quad (23)$$

Besides, the potential energy of the external force is numerically calculated by,

$$E_{ex} = - \sum_i \bar{u}_{i,a} T_{i,al} l_i, \quad (24)$$

where l_i is the mean interval of the sample points on the boundary. The sample points that are required for this Neumann boundary condition are generated right on the location where the forces are applied. Since all the Neumann boundary conditions presented in this work are concentrated forces, for every numerical example, one sample point is generated at exactly where the concentrated force is being applied. A similar approach can be made for the distributed traction force, by generating several sample points along the traction boundary line or surface.

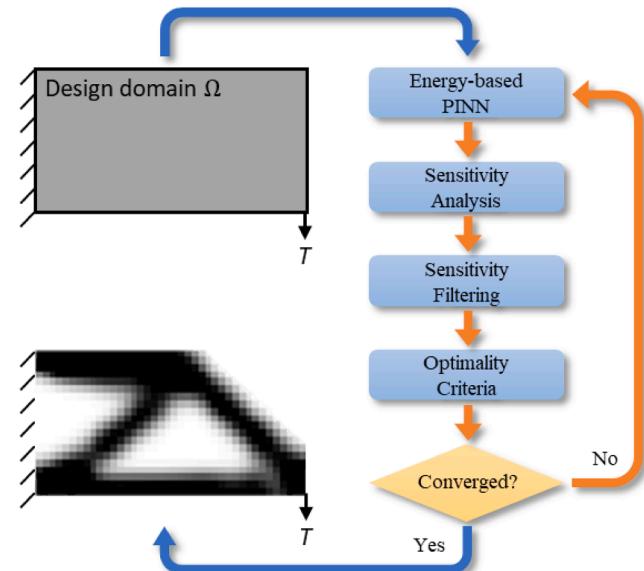


Fig. 6. The flowchart of PINNTO framework. After defining PINNTO parameters, the proposed PINNTO uses Energy-based PINN to replace FEA to obtain the displacement fields of the structure. Subsequent sensitivity analysis and filtering are performed with approximated displacement fields from Energy-based PINN to construct a new optimized design until the convergence guided by the Optimality criterion is satisfied.

The Dirichlet boundary conditions are imposed using the hard boundary condition scheme proposed by Samaniego, et al. [54]. In this method, the sample points can be generated as long as they can cover the boundary lines without the obvious vacancy. In particular, this technique can overcome the problems with complex boundary constraints that weakly imposed boundary condition with Lagrangian multipliers fails to satisfy [31]. Hence, the final displacements for the cantilever beam can then be obtained as follows:

$$\begin{aligned} u_1^*, u_2^* &= N(x, y, W, b), \\ u_1 &= u_1^* \cdot x, \\ u_2 &= u_2^* \cdot x, \end{aligned} \quad (25)$$

where u_1^* , u_2^* are the intermediate displacement outputs in x , y directions, respectively. These intermediate outputs are multiplied by the x coordinates of the sample points. In this manner, the displacement

boundary conditions are naturally satisfied [44].

Note that all the NNs used in this work are built on the PyTorch library (version 1.9.1 + cpu) [32]. Depending on the problem, the sizes of NNs need to be modified to assure high performance and efficient training. In this paper, we have used eight hidden layers with 80 neurons for each layer to approximate the displacement fields. Xavier initialization method [55] is applied to initialize NN parameters in the hidden layers. The Adam optimizer [56] is used to minimize the value of the energy-based loss function. The network is set to train for 1000 epochs with full batch size to compute the gradient updates of all sample points per epoch. The initial learning rate is $lr = 0.001$ with adaptive scheduler to gradually reduce the learning rate by a factor of 0.5 if no improvement is found after 10 epochs in the loss history. Other parameters for the NNs are set as default settings if not explicitly mentioned. All numerical cases related to this paper have been implemented on Intel® Core™ i7-10700 CPU @ 2.90 GHz with 16.0 GB DDR4 RAM. Lastly, Python (version 3.9.5) [57] is used throughout this paper to develop the PINNTO algorithm.

4. Numerical examples

This section provides case studies carried out to validate the PINNTO, investigate the sensitivity of PINNTO framework on different computational variations and present the capabilities of PINNTO in complex problems.

4.1. Validation

The performance of the proposed PINNTO is analysed by comparing the structure designs obtained from both PINNTO and 88-line MATLAB implementation of SIMP-based optimization [52]. We first consider the classic compliance minimization problems described in Eq. (21). The domain of the beam is $2 \text{ m} \times 1 \text{ m}$ in size and is discretised into 40×20

elements. The material properties and target volume fraction v_f are listed in Fig. 7 with the fixed Poisson's ratio of $\nu = 0.3$ and $E_{\min} = 1 \times 10^{-9} \text{ Pa}$ for all cases. The TO parameters are filter radius $r_{\min} = 3$, penalization factor $p = 3$, and the allowable convergence error of $\tau = 0.01 \%$.

Minor variations can be observed in the structural designs between the two approaches due to the different displacement fields obtained via energy-based PINN and FEA. In terms of the objective function value (compliance), the difference between the two approaches is marginally low with less than $2 \sim 3 \%$ in compliance for most cases.

The convergence plots for both PINNTO and SIMP are illustrated in Fig. 8. For both the frameworks, the mean compliance plots display decreasing trends as the design variables converge towards the 0–1 solution. In addition, Cases 1 and 2 illustrate similar trends for both PINNTO and minor differences in the number of iterative convergences. In particular, the PINNTO in Case 3 exhibits faster convergence in TO than in SIMP, whereas in Case 4 is vice versa. This disparity between the convergence rate is due to the approximation difference of the displacement functions. There exist relative approximation errors of displacement field between energy-based PINN and FEA [54]. This causes creating different topologies and compliances after every optimization step.

The loss convergence plots of energy-based PINN at the first PINNTO iteration are illustrated in Fig. 9. As training progress, the loss of all four cases decreases at decreasing rate and begins to converge after 20 epochs.

The convergence history of case 1 is illustrated in Fig. 10. The cantilever beam subjected to a mid-right load converges from grey elements to black and white solution as the number of iterations increases. The corresponding topology simultaneously evolves towards its optimum throughout the convergence process.

Table 1 summarises the computational efficiency characteristics for all four cases. The computational time requirements for PINNTO exceeds that of the SIMP method. The state-of-art PINN models are still in the

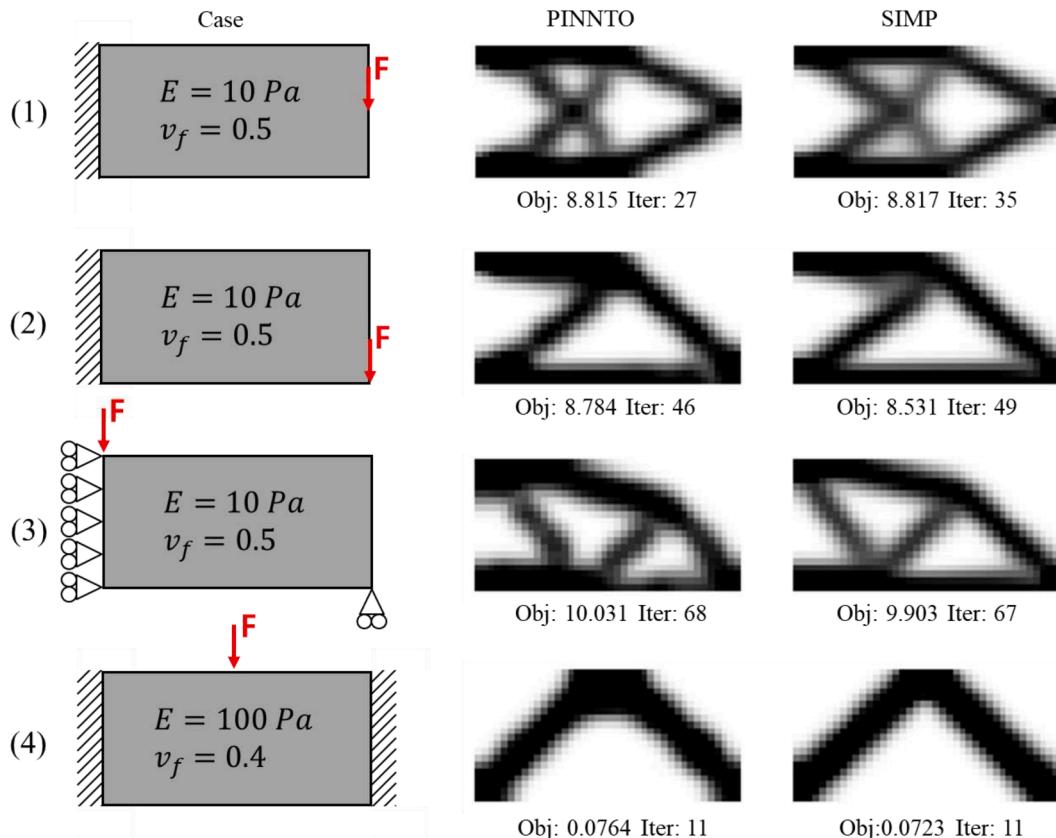


Fig. 7. Validation of PINNTO through classic compliance minimization problems with different boundary conditions.

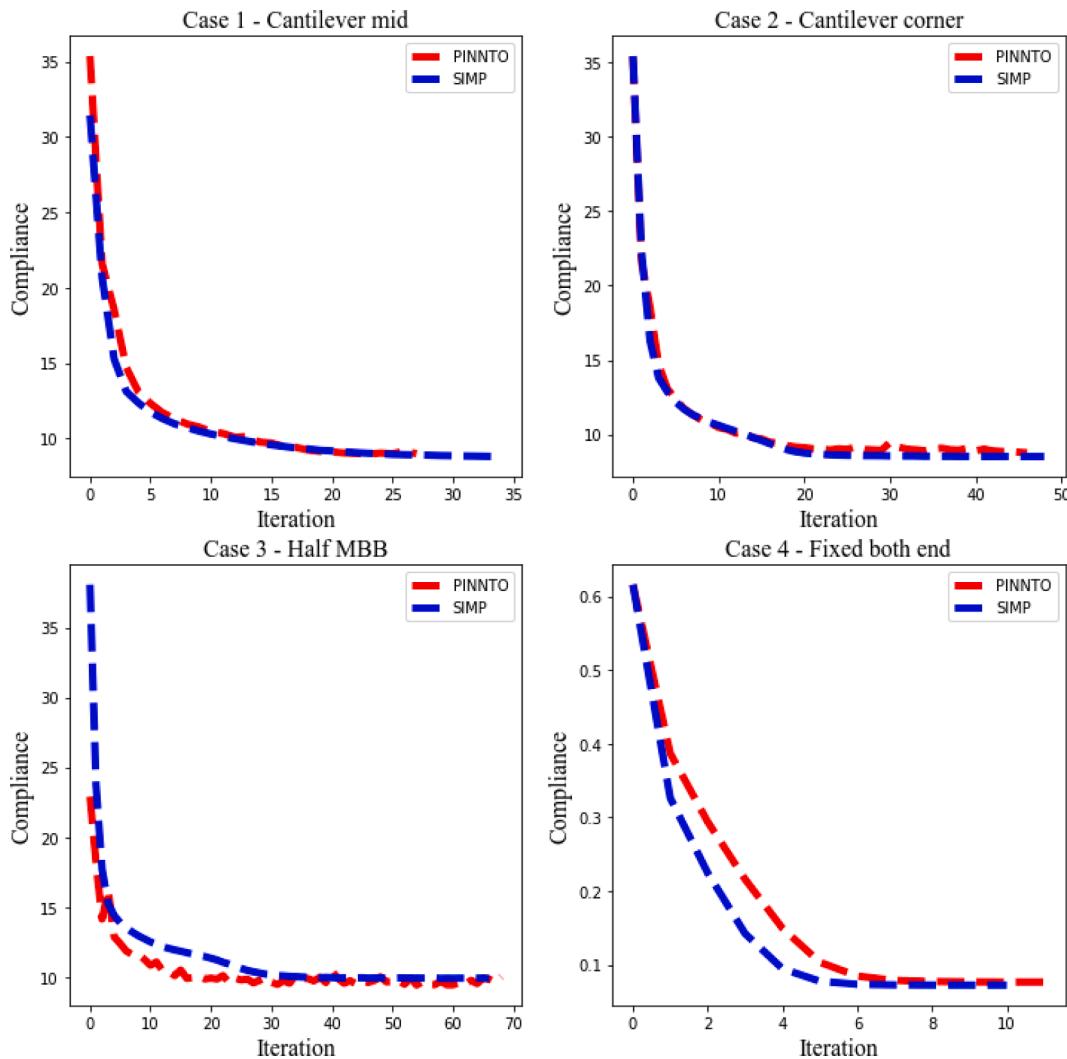


Fig. 8. Convergence plot for problems listed in Fig. 7.

process of being developed to be more computationally efficient due to the challenges existing in the training and loss minimization processes [58]. Subsequently, computational-efficiency improvements of PINNTO can be given as a potential avenue for directly-relevant future research in this domain. Despite the significant computational cost differences, the objective value performance difference is minimal between the two methods. The percentage difference for the first three cases is less than 3 %. In particular, PINNTO demonstrates lower compliance than SIMP for Case 1 with fewer iterations. Case 4 shows the largest percentage % difference of 5.51 % due to the relatively higher complexity of geometrical setup and boundary condition in PINN. The expected displacement field is highly nonlinear that is challenging to approximate using standard FFNN used in PINN [44]. Consequently, it causes the largest approximation errors on the displacement field than in other cases.

4.2. Effect of neural network size

The size of the neural network cannot be estimated as it is problem-dependent [59]. Therefore, in this section, we study the effects of NN size on the PINNTO topologies. To illustrate this, a tip-cantilever problem with varying NN sizes (hidden layer \times hidden nodes) is formulated as shown in Fig. 11. The parameters are kept constants for all cases: Young's modulus $E_0 = 10$ Pa and Poisson's ratio of $\nu = 0.3$. The TO parameters are set to: filter radius $r_{\min} = 3$ and penalization factor $p = 3$.

The design domain is $2\text{ m} \times 1\text{ m}$ discretised into 80×40 mesh. Based on the results shown in Fig. 12, the computed topologies obtained under various NN sizes illustrate different designs.

As predicted, the objective value decreases with the increase of NN size. In particular, the resulting topologies fail to be optimized for smaller NN size, as seen in Fig. 12(a) and (b). Thus, it can be concluded that the quality depends on the NN size. The computational time per iteration increases as the number of floating point operations in hidden layers and nodes increases [60]. Based on various NN sizes in PINNTO, the observed performance indicates that at least a size of 8×80 is required to generate reliable topology at the minimum objective value and computational cost. This NN size is also influenced by the TO problems, for example, more complex TO problems such as multiple loading cases require a larger NN size.

4.3. Effect of mesh size

Next, we analyse the effect of mesh size on the performance of PINNTO. The computed topologies obtained for various mesh sizes for the square cantilever beam are illustrated in Fig. 13. The mechanical and TO parameters are kept identical to the case study in Section 4.2. Herein, we observe the topology changes for various mesh sizes (width \times height) due to the mesh-dependent SIMP algorithm [61]. The PINNTO illustrates similar structures as SIMP with minor irregular branches can be seen for finer mesh sizes due to the approximation errors. Finally, the

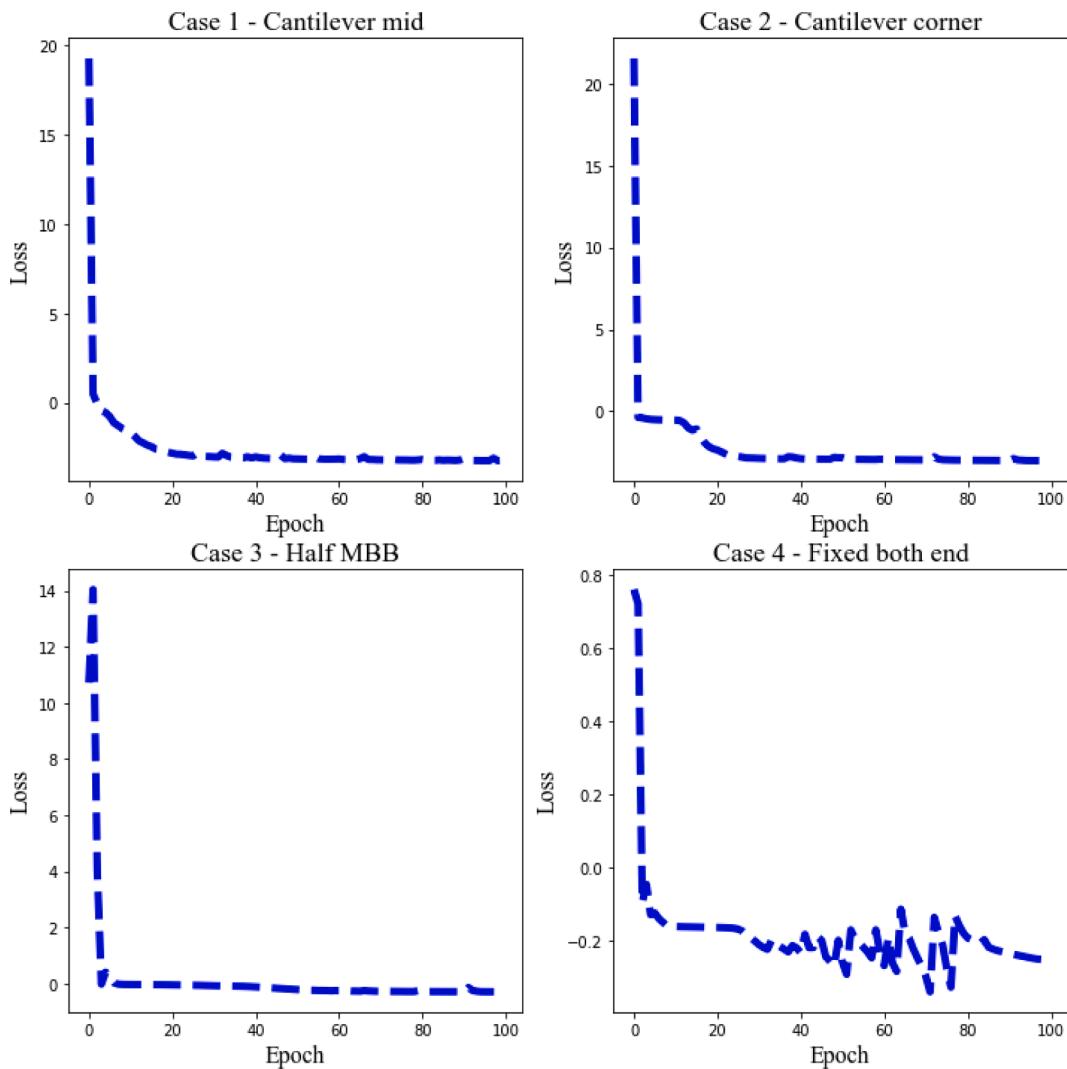


Fig. 9. Loss convergence history of problems listed in Fig. 7 at the first iteration.

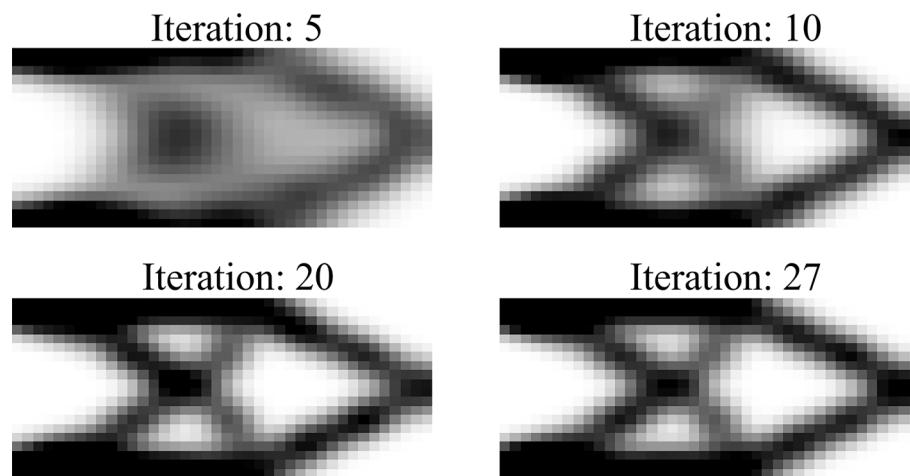


Fig. 10. Convergence history for problem in Fig. 7 case 1.

compliances comparison between PINNTO and SIMP is plotted in Fig. 14 to examine the mechanical performance of PINNTO under the various mesh sizes.

The compliances plot shows a correlated trend between SIMP and

PINNTO. In particular, PINNTO has lower compliance for the coarse mesh (i.e., 10×10). The remaining mesh sizes indicate no significant difference in compliance between SIMP and PINNTO. This indicates that PINNTO is more reliable than SIMP in the coarse mesh setting. The

Table 1

Number of iterations, time taken and objective value for each case in Fig. 7.

Case	No. of Iteration		Time (iter/total) - sec		Objective value		
	PINNTO	SIMP	PINNTO	SIMP	PINNTO	SIMP	% difference
1	27	35	33.89/915.02	0.02/0.81	8.815	8.817	0.02
2	46	49	34.37/1,581.02	0.02/1.19	8.784	8.531	2.92
3	68	67	32.01/2,176.68	0.02/1.56	10.03	9.903	1.28
4	11	11	36.26/398.86	0.03/0.31	0.076	0.072	5.51

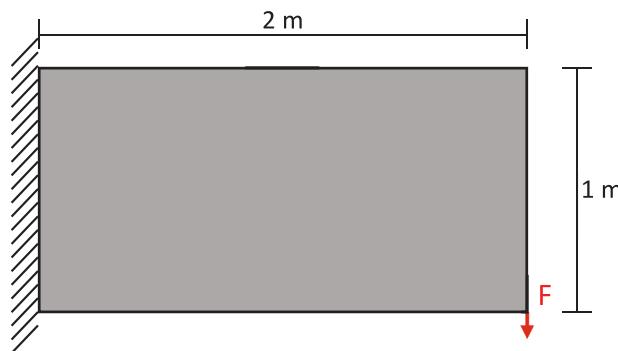


Fig. 11. Tip-loaded cantilever beam problem.

displacement field's accuracy is highly dependent on its mesh size [62]. The FEA may lead to less accurate results for coarse mesh, whereas PINN can approximate consistent displacement variables independent of the mesh size.

4.4. Passive elements

To further analyse the capabilities of PINNTO, an additional case study has been conducted. The passive elements (i.e., elements with fixed densities of zero or one) are added to the design domain by setting zero densities at the circular vacancy during the OC process, as shown in Fig. 15 (a). In this sense, the uniformly distributed sample points in the design domain can be modelled as an irregular geometry while PINN remains consistent for imposing the boundary constraints. This,

however, does not ensure uniformly sampled points near the circular hole as shown in Fig. 15 (a) as the sample points near the circular boundary cannot exactly be placed on the circular boundary by using uniformly distributed sample points. However, no additional treatment is required for the circular vacancy inside the computational domain as the free boundary doesn't require to be explicitly written in the loss function in the energy-based PINN for computational mechanics. It is worth noting that traction stresses on free boundaries equal to zero. Consequently, the potential energy of external force on those free boundaries also equal zeroes despite having its displacement.

To implement the vacancy inside the computational domain, the well-known topology optimisation technique, so-called the passive element technique, is used in which the sample points inside the circular vacancy are fixed to zero densities during the optimization process. By so doing, the internal energy from those vacancy sample points always equal to zero.

In addition, the rough boundary (with zigzag shapes) will appear when using the low-resolution uniformly distributed sample points. This can be circumvented by applying the high resolution to 61×41 uniformly distributed sample points to accurately improve the shape of the circular hole.

The mechanical properties for this case are: Young's modulus $E_0 = 10$ Pa, and Poisson's ratio of $\nu = 0.3$. The TO parameters are filter radius $r_{\min} = 3$ and penalization factor $p = 3$. The design domain is $1.5 \text{ m} \times 1 \text{ m}$ that is discretised into 60×40 mesh. As shown in Fig. 15 (b), the optimized topology is obtained using PINNTO, where the solid elements are appropriately distributed outside the passive elements. Imposing this design constraint increases compliance by 9.4 % compared to the design without passive elements [14].

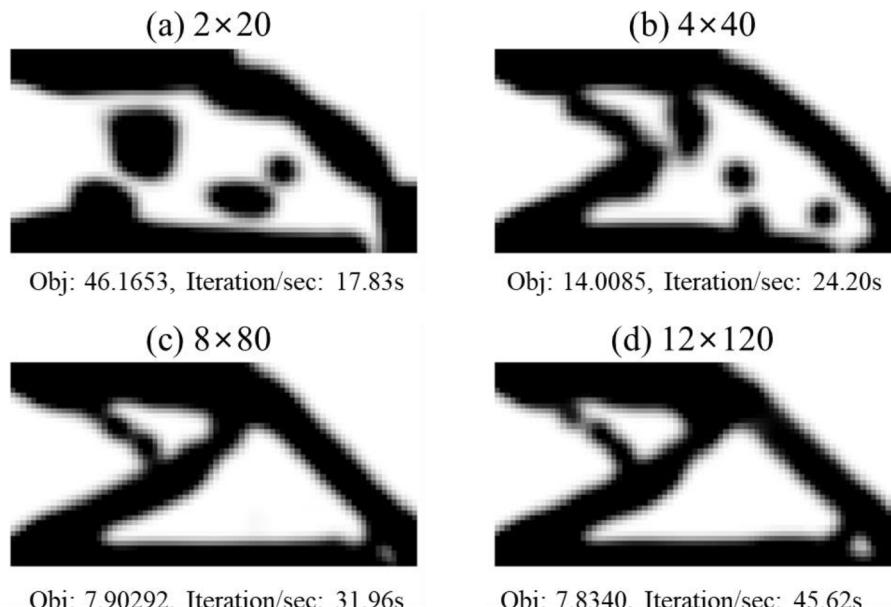


Fig. 12. PINNTO optimal topologies under various NN sizes: hidden layers \times hidden nodes.

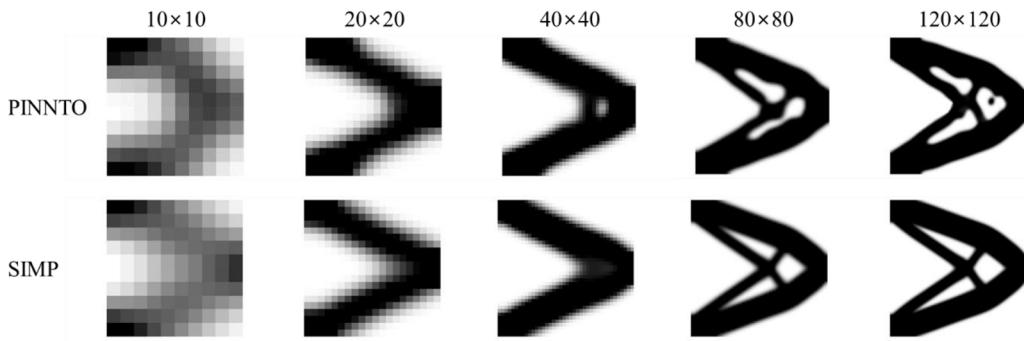


Fig. 13. PINNTO optimal topologies under various mesh sizes: width \times height.

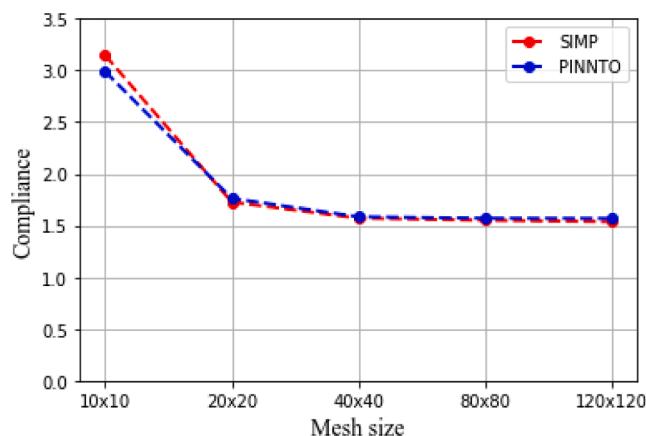


Fig. 14. Comparing compliance between SIMP and PINNTO for varying mesh sizes.

4.5. Multiple loading problem

The performance of PINNTO is investigated for multiple loading scenarios of a domain in this case study. The $1 \text{ m} \times 1 \text{ m}$ square cantilever beam domain is fixed from one end and subjected to two diagonally outward loads ($F_1 = F_2 = 1$) from the top and bottom right corners of the square cantilever beam, as outlined in Fig. 16(a). The corresponding optimal topology is shown in Fig. 16(b). The mechanical and TO parameters are kept identical to Section 4.4. The target volume fraction v_f is set to 25 % to test PINNTO under a small volume fraction. Lastly, the design domain is set to $1 \text{ m} \times 1 \text{ m}$ which is discretised into 50×50 mesh. The optimal design is shown in Fig. 16(b). Through this result, it is evident that PINN can be used to solve the multi-loading problem. In particular, the small volume fraction can also be handled with PINNTO.

4.6. Three-dimensional (3-D) PINNTO

As a final exercise, we consider the 3D implementations of PINNTO. Modifications in PINN have been made to solve 3D problems for an isotropic material where stress and strain z-direction is no longer negligible. In doing so, an additional NN has been added to approximate displacement in the z-direction and is included in the training optimiser. As a consequence, the computational time will be increased due to increased NN parameters and input dataset than in 2-D problems. In particular, the PINN hyper-parameters need to be carefully chosen prior to the NN training due to the robustness issues of the complex 3D problems. All other hyper-parameters and NN setups remain the same as in the previously discussed 2-D cases. Moreover, the training process used for 3-D will remain the same as in 2-D problems.

A tip-cantilever problem subjected to a distributed loading is considered as shown in Fig. 17. The design domain is $6 \text{ m} \times 2 \text{ m} \times 0.4 \text{ m}$ which is discretised into $60 \times 20 \times 4$ hexahedral mesh. The 3-D topology and objective value obtained using the SIMP algorithm coded by Liu and Tovar [48] are illustrated in Fig. 18(a) for the sake of a better comparison. The optimal topology obtained using 3-D PINNTO is illustrated in Fig. 18(b).

The two topologies obtained using 3D SIMP and PINNTO illustrates comparable shape with less than 5 % difference in objective values. This difference is not surprising as the current state of PINN suffers from the neural network used to develop PINN. The limited ability of the FFNN to predict highly non-uniform displacements or strain fields, especially for 3D cases [44]. Nonetheless, while the two topologies are very similar, the number of iterations is significantly lower for PINNTO. This indicates that PINNTO can achieve faster convergence compared to SIMP for specific design problems such as 3D cantilever beams. The faster convergence in 3D PINNTO design is due to the density interpolation scheme in the PINNTO algorithm. The design with a small void region is transformed into the solid when the solid elements surround it. Consequently, the updated design is likely to be identical to the previous one if

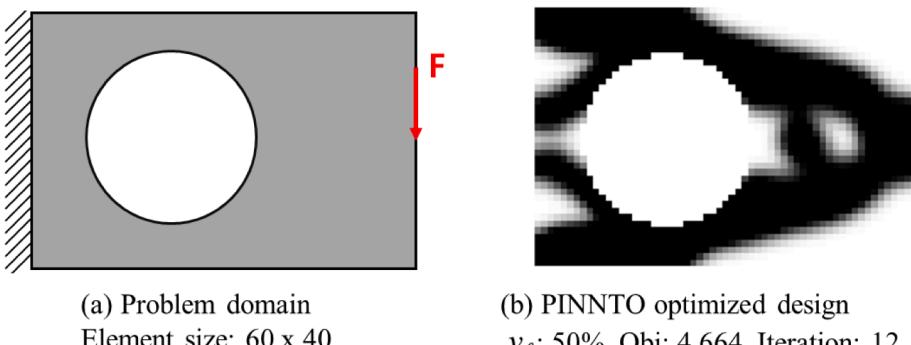


Fig. 15. PINNTO problem with passive elements. (a) the cantilever beam design domain subjected to an external load. The domain consists of uniformly distributed sample points where sample points within the circular hole are passive elements (fixed to zero densities). (b) Corresponding optimal topology using PINNTO.

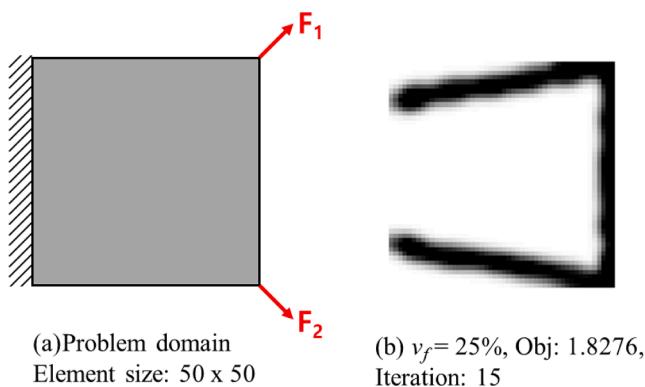


Fig. 16. PINNTO problem with multiple loading scenarios. (a) the square cantilever beam design domain subjected to two external loads. (b) Corresponding optimal topology using PINNTO.

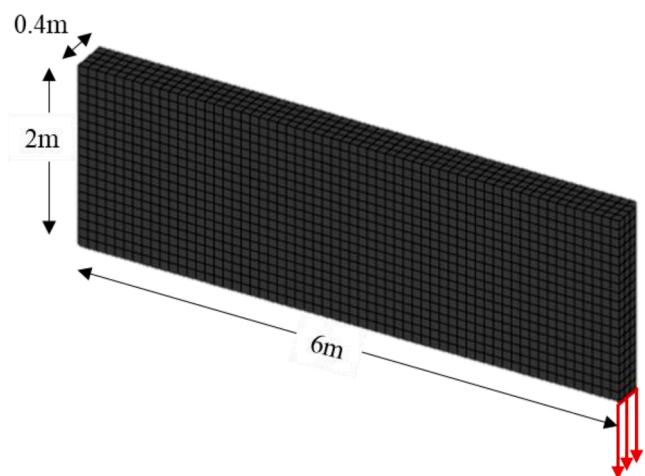


Fig. 17. 3-D cantilever beam subjected to distributed load at the edge.

no significant changes have been made. In conclusion, despite the PINN still being in its developmental phase, attaining accurate 3D PINNTO results is still possible using recently published energy-based PINN model [54,63].

5. Conclusion

In this paper, a novel Physics-Informed Neural Network-based Topology Optimization (PINNTO) framework has been developed. The proposed PINNTO framework is a combination of Topology

Optimization (TO) and Physics-Informed Neural Networks (PINNs). One of the key novelties in the PINNTO framework is the replacement of the FEA component in the traditional TO approach with PINNs. Consequently, PINNTO possesses the capability to overcome inherent shortcomings of FEA-based topology optimization, such as the difficulty of solving nonlinear problems and dealing with complicated materials and domains. To demonstrate the effectiveness and the potential of the proposed PINNTO framework, a number of case studies have been conducted. The numerical examples indicate that PINNTO has the ability to attain the optimized topology without the assistance of labelled data or FEA. Based on this investigation, it can also be deduced that PINNTO can acquire optimal topologies for various types of complex domains including 3D cases, given that the boundary conditions and the loading configurations are correctly imposed for the associated energy-based PINN. In summary, the proposed PINNTO opens up a new avenue for structural design in this ‘data-rich’ age.

There are several open questions to be answered before PINNTO can be further extended for wider application. Firstly, fluctuations of mean compliance could occur in TO due to the usage of PINN instead of FEA. A more stable and advanced Neural Network could be employed in PINNTO, such as using an adaptive activation function for creating NN [64]. Secondly, as the loss function plays a vital role in PINN, more studies could be conducted to develop more robust and effective approaches for the determination of loss function. Thirdly, the hyperparameters used in PINN are problem-based. More investigations should be undertaken for these hyperparameters. Finally, more complex problems including nonlinear and dynamic problems, for which FEA is hard to be employed, should be studied to demonstrate the unique strengths of PINNTO.

It should be noted that the current study is targeted to linear static problems only. In our future work, we will investigate PINN-based topology optimization frameworks for more challenging topology optimization problems such as high-resolution 3D and nonlinear dynamic problems with various force/boundary constraints. Moreover, we will focus on improving the robustness of the PINN by integrating advanced numerical techniques.

CRediT authorship contribution statement

Hyogu Jeong: Conceptualization, Methodology, Formal analysis, Writing – original draft. **Jinshuai Bai:** Conceptualization, Methodology, Formal analysis, Writing – review & editing. **C.P. Batuwatta-Gamage:** Writing – review & editing. **Charith Rathnayaka:** Writing – review & editing. **Ying Zhou:** Formal analysis, Writing – review & editing. **YuanTong Gu:** Conceptualization, Formal analysis, Writing – review & editing, Supervision.

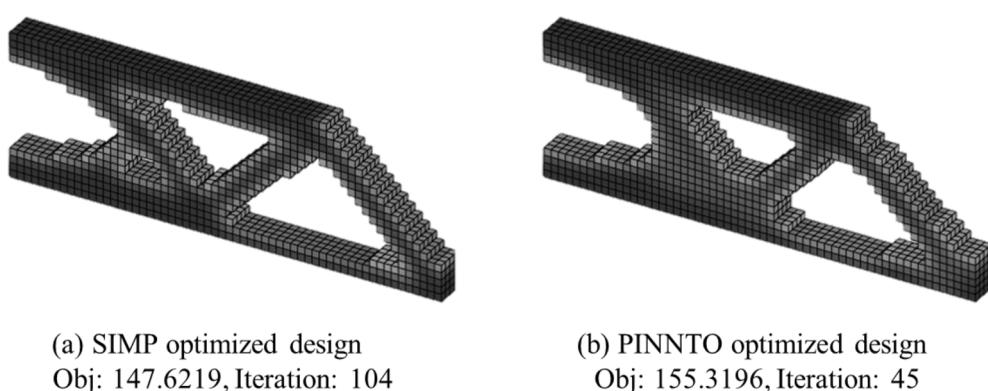


Fig. 18. Optimal topologies obtained using. (a) 3D SIMP code by Liu and Tovar [48] (b) PINNTO.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgements

Support from the ARC Grants (IC190100020 and DP200102546), and QUT Postgraduate Research Award (QUTPRA) and the High-Performance Computing (HPC) resources provided by the Queensland University of Technology (QUT) are gratefully acknowledged.

References

- [1] Huang X, Xie Y-M. A further review of ESO type methods for topology optimization. *Struct Multidiscip Optim* 2010;41:671–83.
- [2] Zhu J-H, Zhang W-H, Xia L. Topology Optimization in Aircraft and Aerospace Structures Design. *Arch Comput Meth Eng* 2016;23:595–622.
- [3] Xia L, Xia Q, Huang X, Xie YM. Bi-directional Evolutionary Structural Optimization on Advanced Structures and Materials: A Comprehensive Review. *Arch Comput Meth Eng* 2018;25:437–78.
- [4] Yu Y, Hur T, Jung J, Jang IG. Deep learning for determining a near-optimal topological design without any iteration. *Struct Multidiscip Optim* 2018;59:787–99.
- [5] Sosnovik I, Oseledets I. Neural networks for topology optimization. *Russ J Numer Anal Math Model* 2019;34:215–23.
- [6] Nie Z, Lin T, Jiang H, Kara LB. TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain. *J Mech Des* 2021;143:1–13.
- [7] Bendsoe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 1988;71:197–224.
- [8] Allaire G, Jouve F, Maillot H. Topology optimization for minimum stress design with the homogenization method. *Struct Multidiscip Optim* 2004;28.
- [9] Xie YM, Steven GP. A simple evolutionary procedure for structural optimization. *Comput Struct* 1993;49:885–96.
- [10] Wang MY, Wang X, Guo D. A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 2003;192:227–46.
- [11] Guo X, Zhang W, Zhong W. Doing Topology Optimization Explicitly and Geometrically—A New Moving Morphable Components Based Framework. *J Appl Mech* 2014;81:081009.
- [12] Yang XY, Xie YM, Steven GP, Querin OM. Bidirectional Evolutionary Method for Stiffness Optimization. *AIAA J* 1999;37:1483–8.
- [13] Bendsoe MP. Optimal shape design as a material distribution problem. *Structural Optimization* 1989;1:193–202.
- [14] Chandrasekhar A, Suresh K. TOuNN: Topology Optimization using Neural Networks. *Struct Multidiscip Optim* 2020.
- [15] Lu L, Pestourie R, Yao W, Wang Z, Verdugo F, Johnson SG. Physics-Informed Neural Networks with Hard Constraints for Inverse Design. *SIAM J Sci Comput* 2021;43: B1105–B1132.
- [16] Behzadi MM, Ilies HT. Real-Time Topology Optimization in 3D via Deep Transfer Learning. *Comput Aided Des* 2021;135:103014.
- [17] Halilaj E, Rajagopal A, Fiterau M, Hicks JL, Hastie TJ, Delp SL. Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. *J Biomech* 2018;81:1–11.
- [18] Zhou Y, Zhan H, Zhang W, Zhu J, Bai J, Wang Q, et al. A new data-driven topology optimization framework for structural optimization. *Comput Struct* 2020;239:106310.
- [19] Ulu E, Zhang R, Kara LB. A data-driven investigation and estimation of optimal topologies under variable loading configurations. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 2015;4:61–72.
- [20] Banga S, Gehani H, Bhilare S, Patel S, Kara L. 3D Topology Optimization using Convolutional Neural Networks. arXiv pre-print server, 2018.
- [21] Abueidda DW, Koric S, Sobh NA. Topology optimization of 2D structures with nonlinearities using deep learning. *Comput Struct* 2020;237.
- [22] Zheng S, Fan H, Zhang Z, Tian Z, Jia K. Accurate and real-time structural topology prediction driven by deep learning under moving morphable component-based framework. *App Math Model* 2021;97:522–35.
- [23] Zhang Y, Peng B, Zhou X, Xiang C, Wang D. A deep Convolutional Neural Network for topology optimization with strong generalization ability. arXiv pre-print server, 2020.
- [24] Zhang Z, Li Y, Zhou W, Chen X, Yao W, Zhao Y. TONR: An exploration for a novel way combining neural network with topology optimization. *Comput Methods Appl Mech Eng* 2021;386:114083.
- [25] Chen L, Shen M-H-H. A New Topology Optimization Approach by Physics-Informed Deep Learning Process. *Adv Sci, Technol Eng Syst J* 2021;6:233–40.
- [26] Ates GC, Gorguluarslan RM. Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. *Struct Multidiscip Optim* 2021;63:1927–50.
- [27] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 2019;378:686–707.
- [28] Bai J, Zhou Y, Ma Y, Jeong H, Zhan H, Rathnayaka C, et al. A general Neural Particle Method for hydrodynamics modeling. *Comput Methods Appl Mech Eng* 2022;393:114740.
- [29] Sun L, Gao H, Pan S, Wang J-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput Methods Appl Mech Eng* 2020;361:112732.
- [30] Haghhighat E, Raissi M, Moura A, Gomez H, Juanes R. A deep learning framework for solution and discovery in solid mechanics. arXiv pre-print server, 2020.
- [31] Rao C, Sun H, Liu Y. Physics-Informed Deep Learning for Computational Elastodynamics without Labeled Data. *J Eng Mech* 2021;147:04021043.
- [32] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin ZM, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai JJ, Chintala S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Adv Neur In*, vol. 32, 2019.
- [33] Batuwatta-Gamage CP, Rathnayaka CM, Karunasena HCP, Wijerathne WDCC, Jeong H, Welsh ZG, et al. A physics-informed neural network-based surrogate framework to predict moisture concentration and shrinkage of a plant cell during drying. *J Food Eng* Nov 2022;332.
- [34] Haghhighat E, Juanes R. SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Comput Methods Appl Mech Eng* 2021;373:113552.
- [35] Peng W, Zhang J, Zhou W, Zhao X, Yao W, Chen X. IDRLnet: A Physics-Informed Neural Network Library. arXiv pre-print server, 2021.
- [36] Hennigh O, Narasimhan S, Nabian MA, Subramaniam A, Tangsali K, Fang Z, Rietmann M, Byeon W, Choudhry S. NVIDIA SimNet™: An AI-Accelerated Multi-Physics Simulation Framework. ed: Springer International Publishing, 2021, pp. 447–61.
- [37] Lu L, Meng X, Mao Z, Karniadakis GE. DeepXDE: A Deep Learning Library for Solving Differential Equations. *SIAM Rev* 2021;63:208–28.
- [38] Markidis S. The Old and the New: Can Physics-Informed Deep-Learning Replace Traditional Linear Solvers? *Front Big Data* 2021;4:669097.
- [39] Xue L, Liu J, Wen G, Wang H. Efficient, high-resolution topology optimization method based on convolutional neural networks. *Front Mech Eng* 2021;16:80–96.
- [40] Lu L, Pestourie R, Yao WJ, Wang ZC, Verdugo F, Johnson SG. Physics-informed neural networks with hard constraints for inverse DESIGNast. *SIAM J Sci Comput* 2021–02–09 2021;43: B1105–B1132.
- [41] Margossian CC. A review of automatic differentiation and its efficient implementation. *WIREs Data Min Knowl Discover* 2019;9:e1305.
- [42] Kumar P, Schmidlein C, Larsen NB, Sigmund O. Topology optimization and 3D printing of large deformation compliant mechanisms for straining biological tissues. *Struct Multidiscip Optim* 2021;63:1351–66.
- [43] Buhl T, Pedersen CBW, Sigmund O. Stiffness design of geometrically nonlinear structures using topology optimization. *Struct Multidiscip Optim* 2000/04/01 2000;19:93–104.
- [44] Li W, Bazant MZ, Zhu J. A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches. *Comput Methods Appl Mech Eng* 2021;383:113933.
- [45] Kulathunga N, Ranasinghe NR, Vrinceanu D, Kinsman Z, Huang L, Wang Y. Effects of Nonlinearity and Network Architecture on the Performance of Supervised Neural Networks. *Algorithms* 2021;14:51.
- [46] Kulathunga N, Nishath, Vrinceanu D, Kinsman Z, Huang L, Wang Y. Effects of the Nonlinearity in Activation Functions on the Performance of Deep Learning Models. arXiv pre-print server, 2020.
- [47] Alvarez HA, Zambrano HR, Silva OM. Influence of Density-Based Topology Optimization Parameters on the Design of Periodic Cellular Materials. *Materials* 2019;12:3736.
- [48] Liu K, Tovar A. An efficient 3D topology optimization code written in Matlab. *Struct Multidiscip Optim* 2014;50:1175–96.
- [49] Huang X, Xie YM. Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. *Comput Mech* 2009;43:393–401.
- [50] Huang X, Xie YM. Optimal design of periodic structures using evolutionary topology optimization. *Struct Multidiscip Optim* 2008;36:597–606.
- [51] Zuo ZH, Xie YM. A simple and compact Python code for complex 3D topology optimization. *Adv Eng Softw* 2015;85:1–11.
- [52] Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code. *Struct Multidiscip Optim* 2010;43: 1–16.
- [53] Huang X, Xie YM. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elem Anal Des* 2007;43:1039–49.
- [54] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput Methods Appl Mech Eng* 2020;362:112790.
- [55] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. presented at the Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 2010.
- [56] Kingma D, Ba J. Adam: A Method for Stochastic Optimization. International Conference on Learning Representations, 12/22 2014.

- [57] Rossum GV. Python reference manual. In Department of Computer Science [CS], ed: CWI, 1995.
- [58] Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE. Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mech Sin* 2021;37:1727–38.
- [59] Lappas G. “Estimating the Size of Neural Networks from the Number of Available Training Data,” ed: Springer. Berlin Heidelberg 2007:68–77.
- [60] Justus V, Brennan J, Bonner S, McGough AS. Predicting the Computational Cost of Deep Learning Models.
- [61] Rozvany G. The SIMP method in topology optimization - Theoretical background, advantages and new applications. In: 8th Symposium on Multidisciplinary Analysis and Optimization, ed: American Institute of Aeronautics and Astronautics, 2000.
- [62] Ghavidel A, Mousavi SR, Rashki M. The Effect of FEM Mesh Density on the Failure Probability Analysis of Structures. *KSCE J Civ Eng* 2018;22:2370–83.
- [63] Guo M, Haghigat E. Energy-Based Error Bound of Physics-Informed Neural Network Solutions in Elasticity. *J Eng Mech* 2022;148.
- [64] Jagtap AD, Kawaguchi K, Karniadakis GE. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *J Comput Phys* 2020; 404:109136.