Structural and Multidisciplinary Optimization (2024) 67:143
https://doi.org/10.1007/s00158-024-03856-1

RESEARCH PAPER

Physics-informed neural network based topology optimization through continuous adjoint

Xueqi Zhao[1] · **Francesco Mezzadri**[2] · **Tianye Wang**[1] · **Xiaoping Qian**[1]

Received: 14 March 2024 / Revised: 17 June 2024 / Accepted: 22 July 2024 / Published online: 13 August 2024

Abstract
In this paper, we introduce a Physics-Informed Neural Networks (PINNs)-based Topology optimization method that is free from the usual finite element analysis and is applicable for both self-adjoint and non-self-adjoint problems. This approach leverages the continuous formulation of TO along with the continuous adjoint method to obtain sensitivity. Within this approach, the Deep Energy Method (DEM)—a variant of PINN-completely supersedes traditional PDE solution procedures such as a finite-element method (FEM) based solution process. We demonstrate the efficacy of the DEM-based TO framework through three benchmark TO problems: the design of a conduction-based heat sink, a compliant displacement inverter, and a compliant gripper. The results indicate that the DEM-based TO can generate optimal designs comparable to those produced by traditional FEM-based TO methods. Notably, our DEM-based TO process does not rely on FEM discretization for either state solution or sensitivity analysis. During DEM training, we obtain spatial derivatives based on Automatic Differentiation (AD) and dynamic sampling of collocation points, as opposed to the interpolated spatial derivatives from finite element shape functions or a static collocation point set. We demonstrate that, for the DEM method, when using AD to obtain spatial derivatives, an integration point set of fixed positions causes the energy loss function to be not lower-bounded. However, using a dynamically changing integration point set can resolve this issue. Additionally, we explore the impact of incorporating Fourier Feature input embedding to enhance the accuracy of DEM-based state analysis within the TO context. The source codes related to this study are available in the GitHub repository: https://github.com/xzhao399/DEM_TO.git.

Keywords Topology optimization · Physics-informed neural networks · Deep energy method · Compliant mechanisms

1 Introduction

Topology optimization (TO) is a computational design method in engineering, seeking optimal material distribution within structures under specified loads, boundary conditions, and constraints. This method significantly contributes to enhancing design efficiency and minimizing material usage, finding extensive applications in various industrial designs (Bendsoe and Sigmund 2003; Eschenauer and Olhoff 2001).

With the increasing prominence of machine learning (ML) across numerous scientific and engineering disciplines (Frank et al. 2020; Brunton and Kutz 2022), there is a growing interest in integrating ML techniques with topology optimization processes (Woldseth et al. 2022; Shin et al. 2023; Chandrasekhar and Suresh 2021).

A significant advancement in ML-based computational physics is the emergence of the physics-informed neural network (PINN) (Karniadakis et al. 2021). PINNs offer an ML-based approach to solving partial differential equations (PDEs) by training neural networks without requiring labeled data. The essence of PINNs lies in developing neural networks that closely approximate PDE solutions, with a loss function defined by the adherence to physical laws. By minimizing this loss function, the neural network's outputs are constrained to approximate PDE solutions while complying with these laws. The PINN framework has introduced novel capabilities in computational physics, particularly its mesh-free characteristic, which simplifies the management

Responsible editor: W. H. Zhang

✉ Xiaoping Qian
qian@engr.wisc.edu

[1] Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, USA

[2] Department of Engineering Enzo Ferrari, University of Modena and Reggio Emilia, Via P. Vivarelli, 10/1, Modena, Italy

of complex numerical domains where mesh generation is challenging (Jagtap and Karniadakis 2021; Xiang et al. 2022; Costabal et al. 2024). Furthermore, PINNs allow for an integration of physical models with empirical data (Raissi et al. 2019; Cai et al. 2021; He et al. 2020), enabling the neural network to simultaneously conform to observed data and the governing physical laws. This integration is particularly beneficial in scenarios where the physical understanding is incomplete, allowing for the resolution of realistic problems using experimental data (Cai et al. 2021), and facilitating the data-driven discovery of new PDEs (Raissi et al. 2019).

The versatility of PINNs has spurred interest in developing various PINN variants, each encoding different physics aspects (Raissi et al. 2019; Haghighat et al. 2021; Kharazmi et al. 2019, 2021; Samaniego et al. 2020; Yu et al. 2018; Nguyen-Thanh et al. 2020; Fuhg and Bouklas 2022). These variants can be broadly categorized based on the nature of the encoded physics laws: strong-form PDE-based PINNs (Raissi et al. 2019; Haghighat et al. 2021), variational formulation-based PINNs (Kharazmi et al. 2019, 2021), and functional minimization formulation-based PINNs (Samaniego et al. 2020; Yu et al. 2018; Nguyen-Thanh et al. 2020; Fuhg and Bouklas 2022). The strong-form PDE-based PINN, as proposed by Raissi et al. (2019), directly inputs spatial and temporal coordinates into the neural network to output PDE solution values at those points. The loss function comprises the strong-form PDE residuals at collocation points and boundary condition discrepancies at boundary points, ensuring the network's outputs satisfy both the PDE and boundary conditions. Despite the straightforward implementation of strong-form PDE-based PINNs, their requirement for higher-order derivatives, as opposed to variational or minimal functional formulations, makes them computationally intensive. The variational formulation-based PINN, introduced by Kharazmi et al. (2019, 2021), constructs its loss function from the residuals of the PDE's variational formulation, involving lower-order derivatives and hence less computational demand. However, this approach necessitates defining a test function space, adding complexity to the implementation. The minimal functional formulation-based PINN, or Deep Energy Method (DEM), advocated by Yu et al. (2018), Nguyen-Thanh et al. (2020), and Samaniego et al. (2020), builds the loss function on a functional minimization approach. This method, applicable to a broad range of engineering problems, offers computational efficiency and simplifies implementation by avoiding the need for higher-order derivatives and test functions (Li et al. 2021), although it is limited to PDEs that can be expressed in a minimal functional form.

In recent years, several PINN-based TO frameworks have emerged, offering promising alternatives to traditional PDE solver-based TO methods (Lu et al. 2021; Jeong et al. 2023a, b; He et al. 2023; Zehnder et al. 2021). Lu et al. (2021) utilized a strong-form PDE-based PINN for TO in optics and fluid dynamics, training two neural networks concurrently for design parameterization and state analysis. The loss function, formulated via penalty and augmented Lagrangian methods, treats strong-form PDEs as constraints, achieving accurate state analysis only upon optimization completion. This approach completely replaces the sensitivity analysis in traditional TO with automatic differentiation. However, since it simultaneously solves the necessary conditions of optimality for the density parameterization network parameters as well as the state analysis network parameters, a premature termination of the approach does not provide any analysis results for the current design. Recently, it has been shown by Li et al. (2021) that the DEM has better efficiency than strong-form PDE-based PINN for solving PDEs. Utilizing DEM, Zehnder et al. (2021), He et al. (2023), and Jeong et al. (2023a, b) have also developed PINN-based TO frameworks. These frameworks adopt a Nested ANalysis and Design (NAND) method for TO, featuring a primary outer loop for design optimization and a secondary inner loop for state analysis using DEM. So far, these DEM-based TO frameworks have only been applied to compliance minimization problems. In the frameworks proposed by Zehnder et al. (2021), He et al. (2023), Jeong et al. (2023a), the design sensitivities expressions are derived from the continuous or discrete Adjoint Variable Method (AVM). Since the compliance minimization problems are self-adjoint, where the adjoint problem solution coincides with the primal problem solution, the procedure of sensitivity computation in those frameworks only requires solving the primal problem. Neither of these frameworks can be directly applied to more general non-self-adjoint topology optimization problems, for which the sensitivity calculation needs to solve an additional adjoint problem besides the primal problem. In the recent work done by Jeong et al. (2023b), the sensitivities for design update are obtained fully utilizing automatic differentiation, not dependent on the derivation of AVM. In this work, two neural networks, for state analysis and design parameterization respectively, are trained in series to solve the TO problem (Jeong et al. 2023b). However, this approach hasn't been applied to TO problems other than the minimal compliance problem. Furthermore, among those DEM-based TO frameworks, in He et al. (2023) and Jeong et al. (2023a), finite element discretization is employed for estimating design sensitivities. In He et al. (2023), the spatial derivatives of the state variables are interpolated using finite element shape functions when estimating the DEM training loss function. Thus the DEM-based TO frameworks proposed by both He et al. (2023) and Jeong et al. (2023a) are still dependent on finite element discretization of the domain.

Thus far, there has not been any DEM-based TO approach that has been validated for its effectiveness in solving

non-self-adjoint TO problems while also being free from FEM discretization. Building on these developments, our work proposes a DEM-based TO framework suitable for both self-adjoint and non-self-adjoint problems, and meanwhile not dependent on any FEM discretization for both PDE solutions and the design sensitivity calculation. In this work, the design sensitivity expressions are derived by AVM, since it is the most effective method for calculating derivatives in topology design (Bendsoe and Sigmund 2003). Both the primal equation and the adjoint equation are solved utilizing DEM. For free from FEM discretization, we use the continuous adjoint method instead of the discrete adjoint method for sensitivity analysis, and we use automatic differentiation for estimating spatial derivatives when evaluating the DEM training loss and design sensitivity. Moreover, when calculating the DEM training loss, we use a dynamic set of collocation points instead of a static set to avoid the unbounded DEM training loss issue and make the DEM solution of PDEs better regularized in the whole spatial domain.

Our DEM-based TO approach, depicted in Fig. 1, employs a nested analysis and design strategy. In each optimization cycle, two neural networks are trained by minimizing DEM losses for approximating the solutions for the primal problem and the adjoint problem respectively. With the two trained neural networks, the spatial derivatives of the primal and adjoint problems' solutions can be directly evaluated through automatic differentiation. The sensitivity in each cycle is calculated using the spatial derivatives. The design is updated using the sensitivity. The optimal design is obtained when the converge criterion is met and the optimization loop ends.

We illustrate the effectiveness of our approach through three examples: heat sink design, compliant displacement inverter, and compliant gripper designs. Key highlights of our proposed framework include:

- Versatility in addressing both self-adjoint and non-self-adjoint TO problems.
- Obviating finite element discretization, reducing computational complexity.
- A fully unsupervised learning-based approach to TO that eliminates the need for offline data set generation.
- High accuracy of DEM solutions in solving both the primal equation and adjoint equation.

The remainder of the paper is organized as follows: Section 2 reviews the density-based TO method and benchmark TO problems. Section 3 introduces the deep energy method and then discusses the issues one may encounter in DEM implementation for spatial derivative estimation and collocation point sampling. Section 4 validates the efficacy of our proposed framework with three benchmark TO problems, and discuss the performance of the proposed framework on state analysis accuracy and efficiency. Also, we do two ablation studies to show the contributions of Fourier Feature input embedding and dynamic training set. Section 5 summarizes findings and future work directions.

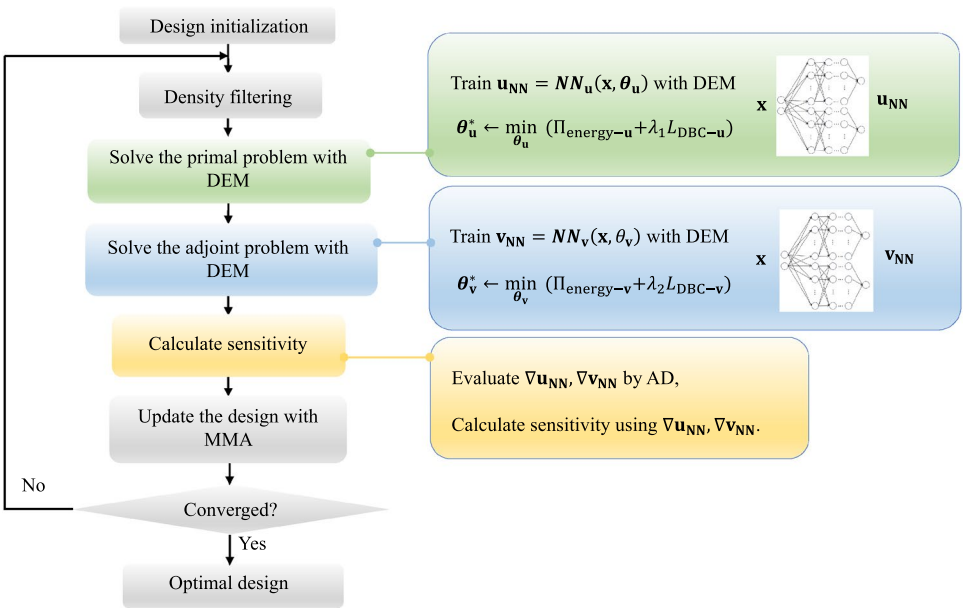


Fig. 1 The flowchart diagram of the proposed DEM-based TO framework

2 Topology optimization

2.1 Problem formulation of TO

A design optimization aims to find an optimal design variable, γ, that minimizes an objective function subject to constraints, which can be formulated as:

$$\min_{\gamma} \quad f_{\text{obj}}(\gamma) = \mathcal{F}(\gamma, \mathbf{u}(\gamma)) \tag{1}$$

$$\text{s.t.} \quad D(\mathbf{u}(\gamma)) = 0, \tag{2}$$

$$g_j(\gamma) \leq 0, \quad j = 1, 2, \ldots, p, \tag{3}$$

where $\mathbf{u}$ is the state variable linked to the design variable γ via the state equation $D(\mathbf{u}(\gamma)) = 0$, and g_j are the inequality constraints.

Topology optimization (TO) seeks the best material distribution within a design domain, denoted by the density function $\gamma \in [0, 1]$, where 0 indicates material absence and 1 indicates material presence as is shown in Fig. 2a.

The design representation in TO is akin to a black-and-white image as shown in Fig. 2b, where each 'pixel' corresponds to a constant value of the density function, γ. The parameterization of the density function is given by:

$$\gamma(\mathbf{x}) = \sum_{i=1}^{N_{\text{ele}}} f_{\gamma_{ei}}(\mathbf{x}), \quad f_{\gamma_{ei}}(\mathbf{x}) = \begin{cases} \gamma_{ei} & \text{if } \mathbf{x} \in \Omega_{ei}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The optimization problem thus becomes the optimization of the variable set $\{\gamma_{ei} | i = 1, 2, \ldots, N_{\text{ele}}\}$.

Material properties for intermediate value of γ are interpolated using the Solid Isotropic Material with Penalization (SIMP) method:

$$\begin{aligned} \kappa(\gamma) =& \kappa_{\min} + (\kappa_0 - \kappa_{\min})\gamma^p, \\ & \kappa(\gamma = 0) = \kappa_{\min}, \quad \kappa(\gamma = 1) = \kappa_0, \end{aligned} \tag{5}$$

where $\kappa(\gamma)$ is the interpolated material property, $\kappa_{\min}$ is the property for void, κ_0 for solid, and p is the penalization exponent, typically chosen as $p = 3$.

A density filter is applied to achieve a mesh-independent design (Bendsoe and Sigmund 2003). The filtered density $\tilde{\gamma}_{ei}$ is computed as:

$$\tilde{\gamma}_{ei} = \frac{\sum_{j \in N_{e,i}} w_j v_j \gamma_{ej}}{\sum_{j \in N_{e,i}} w_j v_j}, \tag{6}$$

where $N_{e,i}$ is the neighborhood set of elements within the filtering domain around Ω_{ei}, a circular region with radius R. The center of element Ω_{ei} is the circle's center. Element Ω_{ej} has a center at $\mathbf{x}_{cj}$ within $N_{e,i}$. v_j is the area of Ω_{ej}, and w_j is the weight for γ_{ej} in computing $\tilde{\gamma}_{ei}$:

$$w_j = R - \|\mathbf{x}_{cj} - \mathbf{x}_{ci}\|. \tag{7}$$

2.2 Sensitivity analysis using the continuous adjoint method

Sensitivity analysis is fundamental for gradient-based optimization algorithms, guiding the iterative process towards an optimal solution by quantifying how small changes in design variables affect the objective or constraints. The adjoint variable method is the most effective method for calculating sensitivities in topology optimization, which typically involves a large number of design variables and a moderate number of constraints. Bendsoe and Sigmund (2003) in this study, the adjoint sensitivity analysis is conducted at the continuous level since the utilization of DEM requires a continuous form of the PDE.

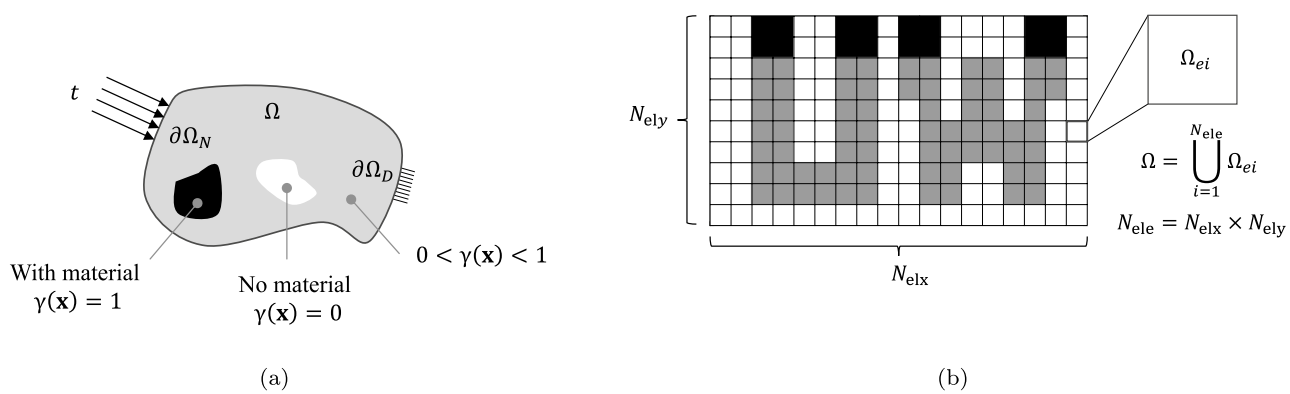


Fig. 2 Illustrations of **a** the density function and **b** design parameterization

For the PDE-constrained optimization problem in (1), if one calculates the sensitivity $\partial f_{\text{obj}}/\partial\gamma$ using the chain rule:

$$\boldsymbol{\nabla} f_{\text{obj}}(\gamma) = \boldsymbol{\nabla}_{\gamma}\mathcal{F}(\gamma, \mathbf{u}(\gamma)) + \boldsymbol{\nabla}_{\mathbf{u}}\mathcal{F}(\mathbf{u}(\gamma))\boldsymbol{\nabla}_{\gamma}\mathbf{u}(\gamma). \tag{8}$$

The term $\boldsymbol{\nabla}_{\gamma}\mathbf{u}(\gamma)$ is challenging to compute due to the implicit dependence of $\mathbf{u}$ on γ.

The AVM simplifies this by introducing a Lagrangian functional L:

$$L(\mathbf{u}, \gamma, \mathbf{v}) = \mathcal{F}(\gamma, \mathbf{u}(\gamma)) + \int_{\Omega} D(\mathbf{u}) \cdot \mathbf{v}\, \mathrm{d}\mathbf{x}, \tag{9}$$

where $\mathbf{v}$ is the adjoint variable. The gradient ∇f_{obj} for a given γ is obtained from:

$$\int_{\Omega} \nabla f_{\text{obj}}(\gamma)\delta\gamma\, \mathrm{d}\mathbf{x} = \mathrm{d}_{\gamma}L(\mathbf{u}(\gamma), \gamma, \mathbf{v}; \delta\gamma), \tag{10}$$

where $\mathbf{v}$ is determined by solving the adjoint equation $\mathrm{d}_{\mathbf{u}}L(\mathbf{u}, \gamma, \mathbf{v}; \delta\mathbf{u}) = 0$.

Utilizing design parameterization from (4), the derivative $\partial f_{\text{obj}}/\partial\gamma_{ei}$ is:

$$\frac{\partial f_{\text{obj}}}{\partial \gamma_{ei}} = \int_{\Omega_{ei}} \nabla f_{\text{obj}}(\gamma)\, \mathrm{d}\mathbf{x}. \tag{11}$$

For sensitivity approximation, Gaussian quadrature is employed:

$$\int_{\Omega_{ei}} (\cdot)\, \mathrm{d}\mathbf{x} \approx \sum_{j=1}^{N_{\text{quad}}} (\cdot)|_{\mathbf{x}=\mathbf{x}_{\text{quad}_{ij}}} w_{ij}, \tag{12}$$

where $(\cdot)|_{\mathbf{x}=\mathbf{x}_{\text{quad}_{ji}}}$ is the integrand value at the jth Gaussian point in Ω_i, and w_{ij} is the corresponding weight. Here, $N_{\text{quad}} = 4$ for second-order quadrature in 2D elements.

With the density filter, the sensitivity of the objective function with respect to the design variables is calculated using the chain rule:

$$\frac{\partial f_{\text{obj}}}{\partial \gamma_j} = \sum_{i \in N_{e,j}} \frac{\partial f_{\text{obj}}}{\partial \tilde{\gamma}_i} \frac{\partial \tilde{\gamma}_i}{\partial \gamma_j}, \tag{13}$$

where the term $\partial\tilde{\gamma}_i/\partial\gamma_j$ is:

$$\frac{\partial \tilde{\gamma}_i}{\partial \gamma_j} = \frac{w_j v_j}{\sum_{k \in N_{e,i}} w_k v_k}. \tag{14}$$

The same approach is used for the sensitivities of other constraints, like volume constraints, with respect to the design variables γ.

2.3 Benchmark TO problems

To demonstrate the effectiveness of the proposed DEM-based TO framework, three benchmark TO problems are used. The problem set includes: (1) Heat sink optimization, (2) compliant displacement inverter design, and (3) compliant gripper design. The math formulations and boundary conditions for the three benchmark problems are shown in this subsection. The detailed derivations of the adjoint equations and sensitivity expressions for the three benchmark problems can be found in Appendix C.

2.3.1 Heat sink optimization

The goal of this problem is to optimize the efficiency of heat transfer away from the heat sources, subject to the material volume constraint. In continuous form, the heat sink optimization problem can be written as:

$$\min_{\gamma} \quad f_{\text{obj}}(\gamma) = \int_{\Omega} \kappa(\gamma)|\boldsymbol{\nabla} T|^2 \mathrm{d}\boldsymbol{x}, \tag{15}$$

$$\text{s.t.} \quad -\boldsymbol{\nabla}\cdot(\kappa\boldsymbol{\nabla} T) = s, \quad \text{for} \quad \boldsymbol{x} \in \Omega, \tag{16}$$

$$-\kappa\boldsymbol{\nabla} T \cdot \mathbf{n} = q, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_N, \tag{17}$$

$$T = T_d, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_D, \tag{18}$$

$$\frac{\int_{\Omega} \gamma \mathrm{d}\boldsymbol{x}}{V} \le V_f, \tag{19}$$

$$0 \le \gamma \le 1. \tag{20}$$

Here the objective functional is thermal compliance. In the direct state equations (15)–(20), T denotes temperature, κ represents the heat conductivity, s denotes the heat source, T_d represents the specified temperature on the Dirichlet boundary Γ_D, q represents the heat flux specified on the Neumann boundary Γ_N, and s represents the heat source in the volume.

The Eq. (19) is the material volume constraint, in which $\int_{\Omega} \gamma \mathrm{d}\boldsymbol{x}$ is the evaluation of the material volume, V is the volume of the design domain, V_f is a pre-specified volume constraint. The Eq. (20) denotes the upper bound and lower bound of the design variable γ. In this study, we consider

$$s = 0.001, \tag{21}$$

$$q = 0, \tag{22}$$

$$T_d = 0. \tag{23}$$

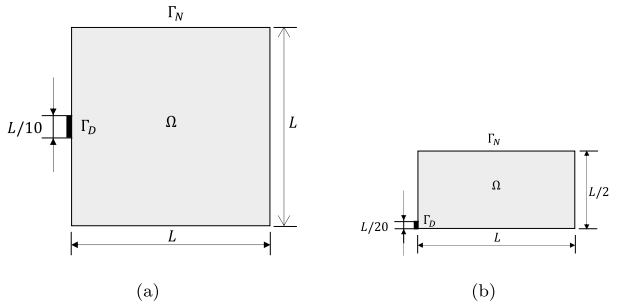


Fig. 3 Design domain and boundary conditions for the primal equations in the heat sink optimization problem. **a** Original design domain and boundary conditions. **b** Design domain and boundary conditions after applying a symmetry treatment

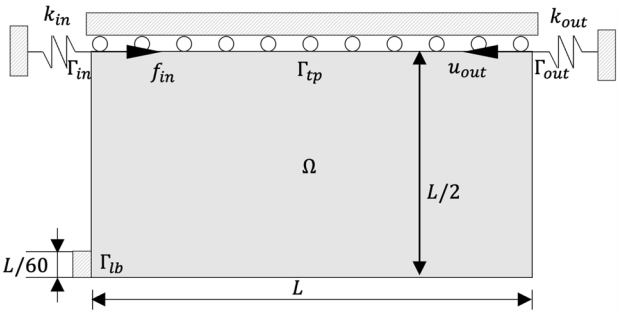


Fig. 4 Design domain and boundary conditions for the compliant mechanism displacement inverter design problem

The heat conductivity κ is interpolated by using the SIMP scheme (5).

The design domain and boundary conditions of the heat sink optimization problem are shown in Fig. 3a. Figure 3b shows the reduction of the domain by applying a symmetry treatment.

2.3.2 Compliant displacement inverter design

The goal of the compliant displacement inverter design problem is to maximize the output displacement u_{out} at the output port, subject to a volume constraint. The design domain, loading conditions, and boundary conditions are depicted in Fig. 4. In continuous formulation, the optimization problem can be written as:

$$\min_{\gamma} \quad f_{\text{obj}} = -u_{\text{out}} = \int_{\Gamma_{\text{out}}} \boldsymbol{u} \cdot (\delta(\boldsymbol{x} - \boldsymbol{x}_{\text{out}})\boldsymbol{e}_{\boldsymbol{x}}) \mathrm{d}\boldsymbol{s}, \tag{24}$$

$$\text{s.t.} \quad \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \boldsymbol{b} = 0, \boldsymbol{x} \in \Omega, \tag{25}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}, \boldsymbol{x} \in \Gamma_N, \tag{26}$$

$$\boldsymbol{u} = 0, \boldsymbol{x} \in \Gamma_D, \tag{27}$$

$$\frac{\int_{\Omega} \gamma \mathrm{d}\boldsymbol{x}}{V} \leq V_f, \tag{28}$$

$$0 \leq \gamma \leq 1. \tag{29}$$

Here the objective functional is the x-displacement at the output port on the top-right corner, and the notation $\boldsymbol{e}_{\boldsymbol{x}}$ represents the unit vector in the x-axis direction. The Eq. (25) denotes the linear elasticity equation, in which $\boldsymbol{\sigma}$ denotes the stress tensor, $\boldsymbol{b}$ denotes the body force. The $\delta(\boldsymbol{x} - \boldsymbol{x}_{\text{out}})$ denotes a direct delta function. The system is subject to Neumann boundary condition (26) and Dirichlet boundary condition (27). The stress $\boldsymbol{\sigma}$ can be computed by using the constitutive law:

$$\boldsymbol{\sigma} = \frac{E}{1+\nu}\boldsymbol{\epsilon} + \frac{E\nu}{(1+\nu)(1-2\nu)} tr(\boldsymbol{\epsilon})\boldsymbol{I}, \tag{30}$$

in which E, ν denotes the material properties Young's modulus and Poisson ratio respectively, $\boldsymbol{\epsilon}$ denotes the stress tensor. Considering small deformation, the strain tensor can be given by:

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right). \tag{31}$$

In this study, we consider:

$$\boldsymbol{b} = \mathbf{0}, \tag{32}$$

$$\nu = 0.3. \tag{33}$$

The Young's modulus E is interpolated by using the SIMP scheme (5).

2.3.3 Compliant gripper design

As for the compliant displacement inverter design, the goal of the compliant gripper design problem is also to maximize the u_{out} at the output port subject to a volume constraint. The design domain, loading conditions, and boundary conditions are depicted in Fig. 5a. The continuous formulation, constitutive relationship, strain tensor expression, and material properties are the same as the Eqs. (24)–(33) for the compliant displacement inverter problem.

Different from the compliant displacement inverter design problem, the compliant gripper case has two passive regions for design in the top-right corner.

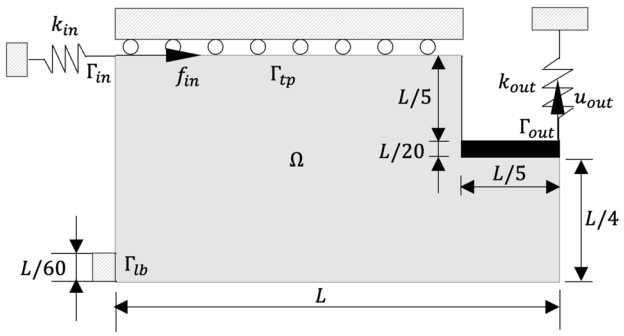


Fig. 5 Design domain and boundary conditions for the compliant mechanism gripper design problem

3 Deep energy method for solving PDE

In this section, we introduce the implementation of the Deep Energy Method. For illustrative purposes, the 2D heat conduction problem in Eqs. (16)–(18) is used as an example (Fig. 6).

3.1 Deep neural network representation of PDE solution

In the DEM for 2D heat conduction, the solution T is approximated by a neural network $T_{\text{NN}} = \text{NN}_T(\mathbf{x};\boldsymbol{\theta}_T)$, where $\mathbf{x}$ are spatial inputs and $\boldsymbol{\theta}_T$ represents the network's trainable parameters.

Information flow in the network from input to output is given by:

$$a_j^l = f_{\text{act}}\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right), \tag{34}$$

where a_k^l is the output of the kth neuron in the lth layer, w_{jk}^l and b_j^l are the weights and biases, and f_{act} is the activation function introducing non-linearity.

In this study, Fourier Feature Embeddings (FFE) are employed to help the DEM model learn the high-frequency components in the solution more effectively so that enhance the accuracy of DEM solutions (Wang et al. 2021). For more details about FFE and its applications on PINN, we recommend readers refer to references (Tancik et al. 2020) and Wang et al. (2021). FFE introduces a random Fourier mapping $\boldsymbol{\phi}(\mathbf{x})$ to the inputs, expressed as:

$$\boldsymbol{\phi}(\mathbf{x}) = [\cos(2\pi\mathbf{B}\mathbf{x}), \sin(2\pi\mathbf{B}\mathbf{x})]^{\text{T}}, \tag{35}$$

where $\mathbf{B} \in \mathbb{R}^{m\times d}$, with its elements sampled from $\mathcal{N}(0, \sigma_{\text{FFE}}^2)$. Here, m represents the number of Fourier features, and d indicates the dimensionality of the input.

3.2 Definition of the DEM loss function

The calculus of variation identifies an important class of PDEs whose solutions can be gained by minimizing the corresponding energy functional (Evans 2022; Le Dret and Lucquin 2016). The DEM loss function approximates this energy functional to solve the corresponding PDE. For the heat conduction problem, the minimal functional problem is:

$$\min_T J(T) = \frac{1}{2}\int_\Omega \kappa|\nabla T|^2 \mathrm{d}\mathbf{x} - \int_\Omega sT\mathrm{d}\mathbf{x}, \tag{36}$$

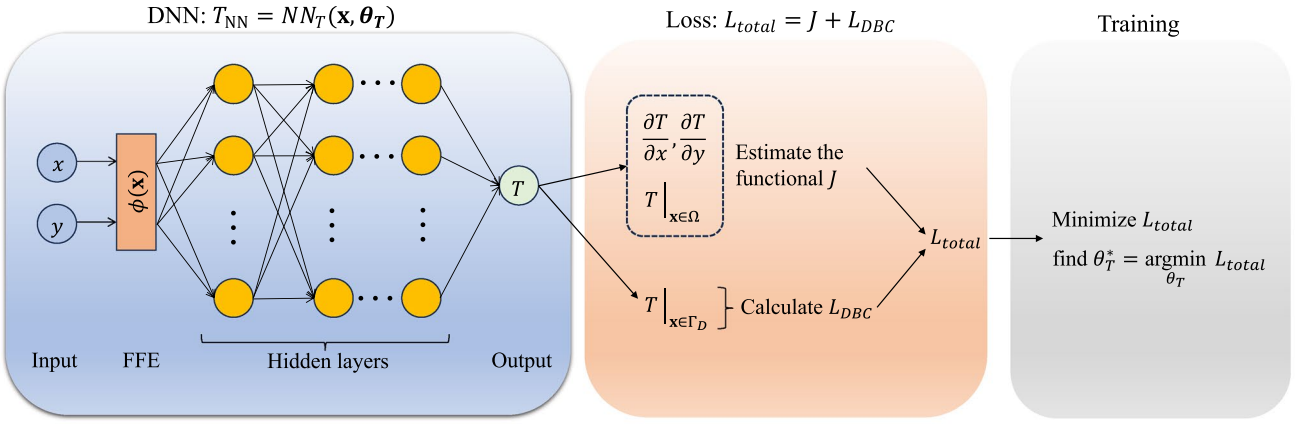


Fig. 6 Computational framework of DEM for solving the equilibrium temperature distribution in the 2D heat conduction problem (16)–(18)

subject to $T = 0$ on Γ_D. This minimal functional formulation can be derived from the strong-form PDE, the steps of deriving the expression (36) can be found in Appendix A.

The DEM loss function, L_{total}, incorporates $J(T)$ and a penalty for Dirichlet boundary conditions (L_{DBC}):

$$L_{\text{total}} = J(T_{\text{NN}}) + L_{\text{DBC}}, \tag{37}$$

where $J(T_{\text{NN}})$ is the DEM energy loss and L_{DBC} enforces boundary conditions. The optimal parameters $\boldsymbol{\theta}_{\boldsymbol{T}}^{*}$ minimize L_{total}, determined iteratively using gradient-based optimization. The gradient $\partial L_{\text{total}}/\partial \boldsymbol{\theta}_{\boldsymbol{T}}$ is evaluated by AD.

For calculating the DEM energy loss, we use a dynamic integration point set for numerical integration, which means the positions of integration points are updated repeatedly during the DEM training. The Monte Carlo integration is utilized since it is more convenient for using a dynamic integration point set. In Monte Carlo integration, since each point has the same weight, there is no need to update weights after each change in the point set. With Monte Carlo integration, the evaluation of $J(T_{\text{NN}})$ is given by:

$$J(T_{\text{NN}}) = \frac{1}{2}\frac{Q_\Omega}{N_\Omega}\sum_{i=1}^{N_\Omega}\kappa_i|\nabla T_{\text{NN}}|_i^2 - \frac{Q_\Omega}{N_\Omega}\sum_{i=1}^{N_\Omega}s_i T_{\text{NN},i}, \tag{38}$$

where $Q_\Omega = \int_\Omega \mathbf{dx}$, N_Ω represents the number of Monte Carlo integration points within Ω, which are randomly sampled from a uniform distribution, and the subscript i denotes evaluation at the ith integration point $\mathbf{x}_i$.

The penalty term L_{DBC} is calculated by:

$$L_{\text{DBC}} = \lambda\frac{1}{N_{\Gamma_D}}\sum_{i=1}^{N_{\Gamma_D}}(T_{\text{NN}i})^2 \tag{39}$$

in which N_{Γ_D} denotes the number of points uniformly sampled on the Dirichlet boundary, and λ denotes the penalty coefficient. In this work, we simply take $\lambda = 1$.

For estimating J with the formula (38), one needs κ_i on the integration points, which can be calculated through the SIMP formula (5) with γ_i. For any sampled integration point $\boldsymbol{x_i}$, the density value is given by $\gamma_i = \gamma(\boldsymbol{x_i})$ using formula (4). In this study, the value of γ_i is obtained using the Nearest-neighbor interpolator (Rukundo and Cao 2012) in the Scipy package. The reason we use the Nearest-neighbor interpolator is to efficiently evaluate the density function values at the integration points after each update of the dynamic integration point set. Figure 7 demonstrates that the use of the Nearest-neighbor interpolator gives the same value as the original function $\gamma(\boldsymbol{x})$. In Fig. 7b, red dots mark element centers. The Nearest-neighbor algorithm lets the γ value at any sampled point equal the γ value at the nearest element center, which gives the same value as $\gamma(\boldsymbol{x})$.

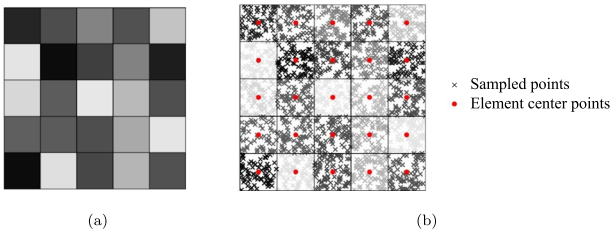


Fig. 7 Demonstration of using the Nearest-neighbor interpolator to obtain γ_i: **a** Original density function $\gamma(\boldsymbol{x})$. **b** The value of γ_i at the sampled points given by the Nearest-neighbor interpolator. (Color figure online)

3.3 Distinctions in our implementation and issues in DEM training

Key distinctions in our implementation include:

- For spatial derivative evaluation: Unlike methods relying on FEM discretization for ∇T_{NN} (e.g., He et al. 2023), our approach employs AD for direct and efficient computation. FEM discretization-based spatial derivative evaluation results in not adequate regularization of T_{NN}, which is only at the FEM nodes. The accuracy of the neural network prediction T_{NN} inside the FEM elements is not assured.
- For integration point sampling: In contrast to the static integration point sets utilized in studies such as He et al. (2023) and Jeong et al. (2023a), our method uses dynamic integration point set, which means randomly resampling the integration points every 10 epochs. A static integration points set makes the DEM energy loss function not lower unbounded when combined with AD-based spatial derivatives, while the dynamic integration point set solves this issue.

Overall, our implementation is distinct in that it uses AD to obtain spatial derivatives and uses a dynamic integration point set. This combination solved the key issues caused by using the FEM discretization-based spatial derivatives ∇T_{NN} and by using a static integration point set. We use a 1-D Poisson equation problem to explain the reasons for these issues.

3.4 Justification of the limitations on FEM discretization-based ∇T_{NN} and static integration points

Consider the 1-D Poisson's equation:

$$-\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} = 2, \quad x \in [0, 1], \tag{40}$$

$$T = 0, \quad \text{for } x = 0, \tag{41}$$

$$\frac{\mathrm{d}T}{\mathrm{d}x} = 0, \quad \text{for } x = 1, \tag{42}$$

with the analytical solution:

$$T(x) = -x^2 + 2x. \tag{43}$$

When solved with DEM, the total loss function L_{total} is defined as:

$$L_{\text{total}} = \frac{1}{2N_\Omega} \sum_{i=1}^{N_\Omega} \left(\frac{\mathrm{d}T_{\text{NN}}}{\mathrm{d}x} \right)_i^2 - \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} 2T_{\text{NN}i} + \lambda T_{\text{NN}}^2|_{x=0}, \tag{44}$$

Figure 8 shows the problem when using finite element shape function interpolation for calculating the spatial derivatives in the DEM loss function. Let the points $x_0, x_1, ..., x_5$ be the set of 1D finite element nodes. Following the work in He et al. (2023), when evaluating L_{total}, $\mathrm{d}T_{\text{NN}}/\mathrm{d}x$ are interpolated on the integration points using finite element shape functions, mathematically represented as $\mathrm{d}T_{\text{interp}}/\mathrm{d}x = \sum_{i=1}^{5} \mathrm{d}\phi_i(x)/\mathrm{d}x T_{\text{NN}}(x_i)$, where $\phi_i(x)$ denotes the shape function, and $T_{\text{NN}}(x_i)$ denotes neural network output T_{NN} at node x_i. This approach results in regularization being confined only to the nodes, neglecting the regions in between and the true derivative of the neural network output $\mathrm{d}T_{\text{NN}}/\mathrm{d}x$ across the domain.

Figure 9 shows the problem of unbounded L_{total} with a static set of integration points. This time we let the points $x_0, x_1, ..., x_5$ be the set of integration points. With the shape of T_{NN} shown in the Fig. 9, since $T_{\text{NN}}(x_i) = 0$ for $i = 0, 2, ..., 5$ and $(\mathrm{d}T_{\text{NN}}/\mathrm{d}x|_{x_i})_{AD} = 0$ for $i = 0, ..., 5$, we have $L_{\text{total}} = (-2/5)T_{\text{NN}}|_{x=x_1}$. If $T_{\text{NN}}|_{x=x_1}$ goes to $+\infty$, L_{total} goes to $-\infty$. The L_{total} is not lower bounded. This occurs because the DEM model can have sudden value changes and high spatial gradient in the middle of two adjacent points, which aren't captured due to the lack of integration points in those regions. No matter how dense the integration points are sampled, as long as the training set is static, the unbounded issue of the loss function exists. Using a Dynamic integration point set can solve this issue since it allows sampling in these regions during updates.

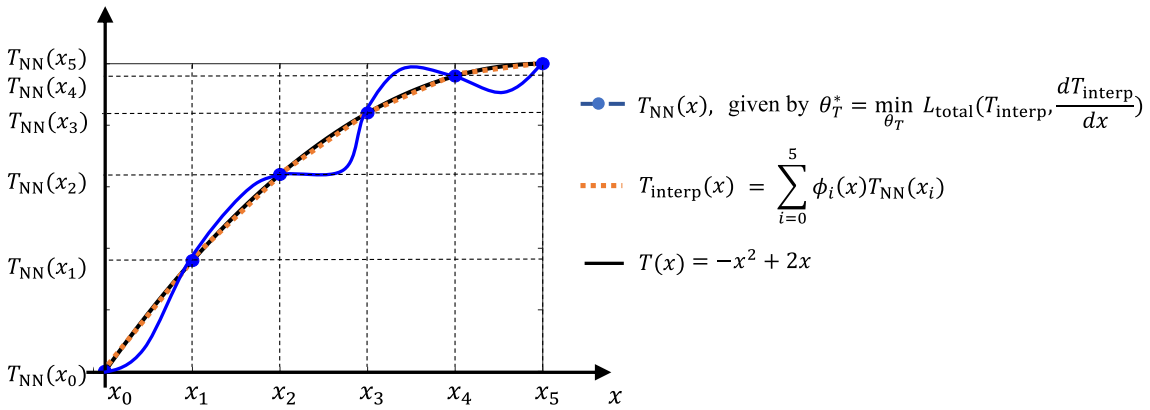


Fig. 8 Illustration of the problem for FEM discretization-based evaluation of ∇T_{NN}

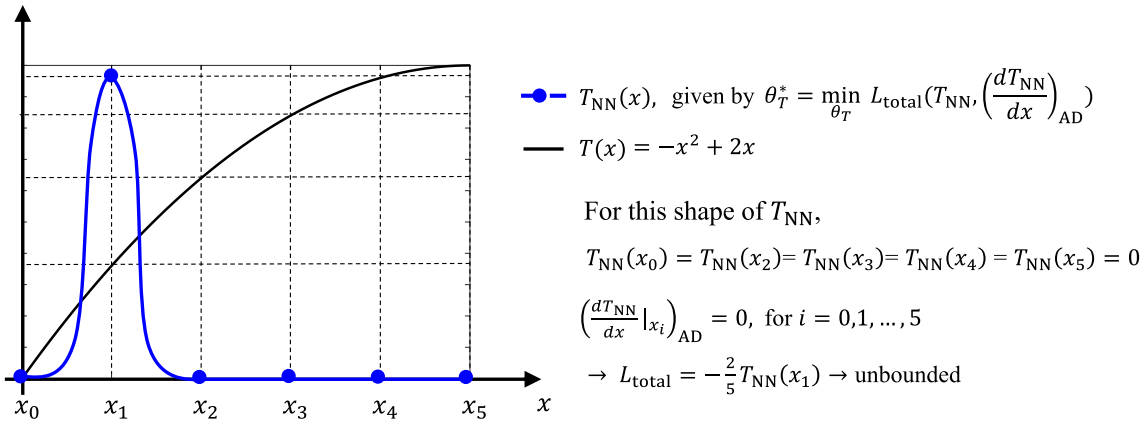


Fig. 9 Illustration of the unbounded loss function problem with a static set of integration points

4 Examples and discussion

This section presents the application of the DEM-based TO framework to three benchmark problems outlined in Sect. 2.

We utilize transfer learning between consecutive DEM-based TO iterations to reduce the training time. Specifically, for the first DEM-TO iteration, the DEM model parameters are initialized using the Glorot normal initializer and trained from scratch to solve the primal PDE and adjoint PDE. For all subsequent TO iterations, the model parameters of the DEM model learned in the previous TO iteration are used to initialize the model parameters for the current iteration. Since the TO designs are similarly in adjacent iterations, the corresponding solutions of the primal PDE and adjoint PDE are also similar. This parameter migration avoids the need to train from scratch, thereby helping to reduce the total training time.

All examples of DEM-TO were run using TensorFlow (v2.6.2) and the Adam optimizer on an NVIDIA RTX 3060 GPU with 32 GB RAM. FEM-TO is implemented based on Aage (2019). The DEM hyperparameters are listed in Table 1. These hyperparameters are optimized using Bayesian optimization for minimizing the error of solving the primal PDE of the linear elasticity problem, employing the package described in Bergstra et al. (2013). These hyperparameters are used in all examples shown in Sect. 4. The first example employs a single neural network for the scalar T, while the latter two train separate networks for each displacement vector component.

4.1 Application to 2D heat sink design problem

The domain dimensions are $(L, H) = (100, 50)$ with grid size 100×50, and we set the heat conductivity κ to 1 and $\kappa_{\min}$ to $1e^{-3}$. The density filter radius is $R = 4.2L/100$. The volume fraction for TO is set as $V_f = 0.5$.

For the TO problem defined in (15)–(20), the detailed derivation of the adjoint problem and the sensitivity can be found in Appendix C.1.

The adjoint problem given by the AVM is:

$$\boldsymbol{\nabla} \cdot (\kappa \boldsymbol{\nabla} \mathbf{v}) + s = 0, \quad \boldsymbol{x} \in \Omega, \tag{45}$$

$$-\kappa \boldsymbol{\nabla} \mathbf{v} \cdot \mathbf{n} = 0, \quad \boldsymbol{x} \in \Gamma_N, \tag{46}$$

$$\mathbf{v} = 0, \quad \boldsymbol{x} \in \Gamma_D. \tag{47}$$

The primal problem in (16)–(18) coincides with the adjoint PDE in (45)–(47). Thus only the solution of the direct state problem is needed for TO.

The sensitivity $\partial f_{\text{obj}}/\partial \tilde{\gamma}_i$ is:

$$\frac{\partial f_{\text{obj}}}{\partial \tilde{\gamma}_{ei}} = -\frac{1}{2}\left(\frac{\mathrm{d}\kappa}{\mathrm{d}\tilde{\gamma}}\right)_i \int_{\Omega_{ei}} \boldsymbol{\nabla} T \cdot \boldsymbol{\nabla} T \mathrm{d}\boldsymbol{x}. \tag{48}$$

4.1.1 DEM-TO vs. FEM-TO

For this example, both DEM-based TO and FEM-based TO are conducted with 100 TO iterations and with the same parameters for the MMA optimizer. Figure 10 shows the comparison between the DEM-TO result and FEM-TO result.

Figure 10a, b show that both DEM-based and FEM-based TO procedures can successfully minimize thermal compliance and satisfy the volume constraint. Comparing DEM-TO and FEM-TO designs at various iterations (Fig. 10c), one can note a similar progression and final designs, with the DEM-TO process manifesting branch features slightly sooner. Despite small differences in the final designs, the relative objective functions for DEM-TO and FEM-TO are closely matched at 0.192 and 0.190, respectively.

4.1.2 Effect of mesh size

We analyze the effect of changing TO mesh size on the quality of the optimal design given by the proposed DEM-TO framework. In Fig. 11, the three topologies are obtained with mesh size 100×50, 200×100 and 400×200 respectively. The neural network architectures for obtaining the three topologies are the same as in Table 1. For all three topologies, the filter size is 4.2 times the element sizes, the number of TO iterations is 500, and the number of training epochs is 10 for each TO iteration except the 1st TO iteration. At the first TO iteration, the number of DEM training epochs is 1000 for obtaining all three topologies. We observe that there are no obvious irregular branches for all of the three topologies. The results show that the proposed DEM-TO framework can generate fine structures.

Figure 12 shows the effect of changing mesh size on the computational time for both DEM-TO and FEM-TO for the heat sink design problem. Here we present the total wall-clock time for DEM-TO and FEM-TO. FEM-TO computation is based on CPU and the computation of DEM-TO is based on GPU. For DEM-TO, this includes parameter initialization, DEM model training to solve the Primal and Adjoint PDEs, design sensitivity calculation, and density filtering. For FEM-TO, this includes parameter initialization, solving the Primal and Adjoint PDEs using FEM and a direct solver, sensitivity calculation, and density filtering. The main difference in

Table 1 Hyperparameters of the DEM model

Layers	Neurons	Activation function	N_Ω	N_{batch}	N_{epoch}	Learning rate	σ_{FFE}
4	86	Swish	40,000	2000	200	$1e^{-3}$	1.32

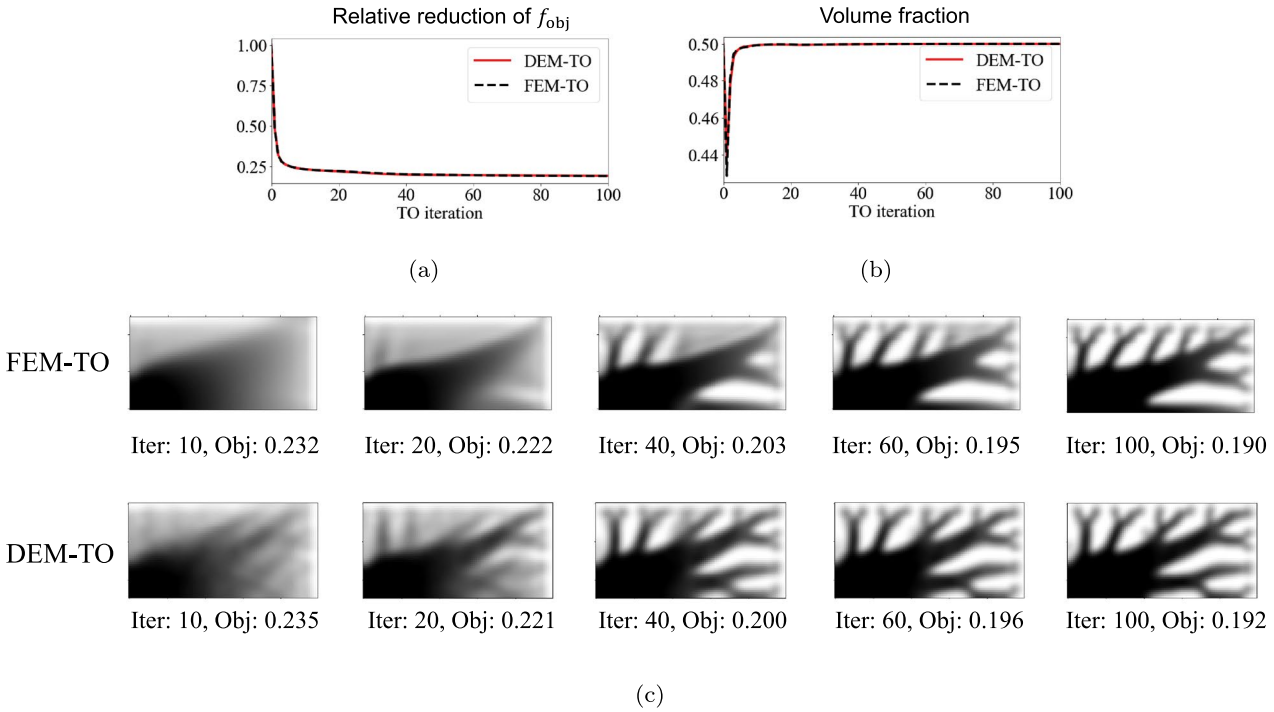


Fig. 10 Comparison of DEM-TO and FEM-TO at various TO iterations for minimal thermal compliance design. **a** The history of the relative reduction of thermal compliance during DEM-based and FEM-based TO. **b** The history of the volume fraction during DEM-based and FEM-based TO. **c** The designs given by FEM-based and DEM-based TO at different TO iterations

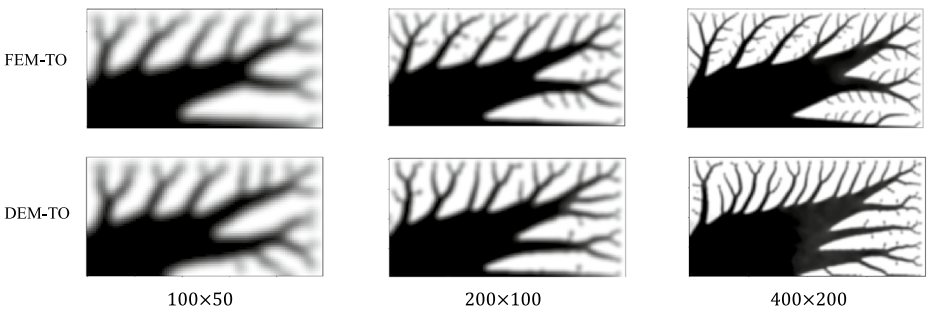


Fig. 11 DEM-TO optimal topologies under various mesh sizes

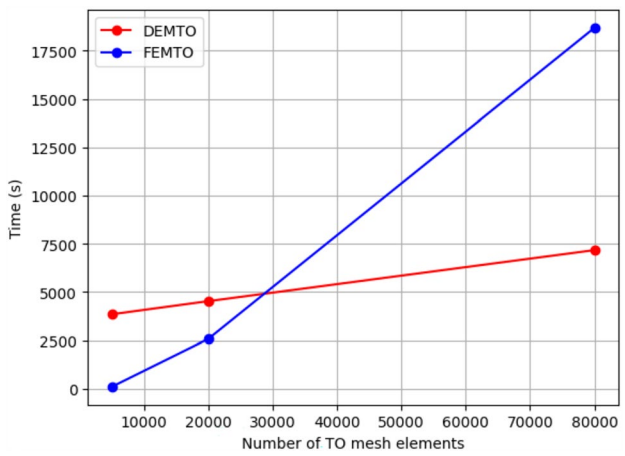


Fig. 12 Computation time of DEM-TO and FEM-TO

time between the two methods is due to the difference in time required for DEM model training versus the direct solver. With the settings used in this paper, we observe that the total computation time of FEM-TO increases faster than that of DEM-TO as the number of TO elements increases. For the mesh size 400×200, the computation time of DEM-TO is 7171.43 s, shorter than the computation time of FEM-TO, which is 18692.31 s. One should note that the purpose of presenting the computation time is not to strictly demonstrate that DEM-TO is more efficient than FEM-TO, as this is not the main focus of this paper, but rather to provide readers with a concrete sense of the time required for our method. The computational time for DEM-TO and FEM-TO can depend on many other factors, such as whether the FEM method uses a direct solver or an iterative solver, whether parallel acceleration is used,

the model and batch size used in DEM training, as well as the programming language, operating system, and processing power. The discussion about computational time here is limited to the hardwares and parameter settings used in this work. A fair comparison of the efficiency of DEM-TO and FEM-TO methods requires a more rigorous discussion of these factors.

4.2 Application to compliant displacement inverter design problem

The domain dimensions are $L = 120$ with gird size 120×60. We set the Young's modulus E to 1 and set $E_{\min}$ to $1e^{-3}$. The Poisson ratio ν is set to be 0.3. The spring stiffness coefficients are set to $k_{\text{in}} = 1, k_{\text{out}} = 0.001$. The density filter radius is $R = 4.2L/120$. The volume fraction is set as $V_f = 0.3$.

The details of deriving the adjoint problem and sensitivity expression are shown in Appendix C.2.

The adjoint problem given by AVM is:

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}^{\mathbf{v}} = 0, \quad \text{for} \quad \boldsymbol{x} \in \Omega, \tag{49}$$

$$\mathbf{v} \cdot \boldsymbol{e}_y = 0, \quad \boldsymbol{\sigma}^{\mathbf{v}} \cdot \mathbf{n} \cdot \boldsymbol{e}_x = 0, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\text{tp}}, \tag{50}$$

$$\mathbf{v} = 0, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\text{lb}}, \tag{51}$$

$$\boldsymbol{\sigma}^{\mathbf{v}} \cdot \mathbf{n} = \delta_{\text{Direc}}(\mathbf{x} - \mathbf{x}_{\text{out}})[-k_{\text{out}}(\mathbf{v} \cdot \boldsymbol{e}_x) + 1]\boldsymbol{e}_x, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\text{out}}, \tag{52}$$

$$\boldsymbol{\sigma}^{\mathbf{v}} \cdot \mathbf{n} = \delta_{\text{Direc}}(\boldsymbol{x} - \boldsymbol{x}_{\text{in}})[-k_{\text{in}}(\mathbf{v} \cdot \boldsymbol{e}_x)]\boldsymbol{e}_x, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\text{in}}, \tag{53}$$

$$\boldsymbol{\sigma}^{\mathbf{v}} \cdot \mathbf{n} = 0, \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\text{rm}}. \tag{54}$$

The sensitivity is:

$$\frac{\partial f_{\text{obj}}}{\partial \tilde{\gamma}_i} = -\frac{\mathrm{d}E_i}{\mathrm{d}\tilde{\gamma}_{ei}} \int_{\Omega_{ei}} \hat{\boldsymbol{\sigma}} : \boldsymbol{\nabla}\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{55}$$

To calculate the sensitivity, one needs to solve both the direct state PDE and adjoint PDE.

The DEM can be naturally applied to the primal PDE in this example since the linear elasticity problem naturally has the minimal functional statement, which is the principle of minimal potential energy. The system's potential energy Π is expressed as:

$$\Pi(\mathbf{u}) = W_{\text{in}}(\mathbf{u}) - W_{\text{ex}}(\mathbf{u}), \tag{56}$$

$$W_{\text{in}}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon}\, \mathrm{d}\mathbf{x}, \tag{57}$$

$$W_{\text{ex}}(\mathbf{u}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{u}\, \mathrm{d}\mathbf{x} + \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{u}\, \mathrm{d}s, \tag{58}$$

where Π, W_{in}, and W_{ex} denote the total potential energy, internal strain energy, and work done by external forces, respectively, with $\mathbf{f}$ as the body force and $\mathbf{t}$ as the traction force on the Neumann boundaries.

For the adjoint PDE in (49)–(54), the minimal potential energy formulation is also applicable. The expression for the potential energy is the same as in (56)–(58) with different boundary conditions.

4.2.1 DEM-TO result vs. FEM-TO result

In Fig. 13, we compare the DEM-TO and FEM-TO designs at the TO iteration 400. The designs given by DEM-TO and FEM-TO have discrepancies, and the design performances are comparable but slightly different. The final objective function values at the 400th TO iteration are -1.340 and -1.412 for DEM-TO and FEM-TO respectively. One can notice in panel (a) and panel (b) that both DEM-TO and FEM-TO can minimize the objective function and achieve optimal TO designs while satisfying the volume constraint.

4.2.2 Solution error of DEM models

Figure 14 shows the relative error of the DEM model solutions at various TO iterations. Here the relative root mean squared error (Relative RMSE) is used. The expression for the Relative RMSE between two vector A and B is:

$$\text{Relative RMSE}(A, B) = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_i - B_i)^2}}{B_{\max} - B_{\min}}, \tag{59}$$

in which N represents the number of entries in both A and B. In Fig. 14, panel (a) and panel (b) show the relative RMSE for the DEM primal solution and the DEM adjoint solution respectively. In both panel (a) and panel (b), the light red curves show the original data of relative errors, and the blue curves show the window moving average to highlight the main tendency. One can observe the decreasing tendencies of DEM solution errors for both the primal PDE solution and the adjoint PDE solution. Figure 15 shows the comparison between the DEM solutions and the FEM solutions for the direct state PDE and the adjoint PDE at TO iteration 400. One can observe that at the end of DEM-TO, the solutions given by DEM and FEM are very close. The relative RMSEs for the primal PDE solution and the adjoint PDE solution are $9.53 \times e^{-3}$ and $1.09 \times e^{-2}$ respectively.

The reason for the higher DEM solutions error possibly lies in two aspects: First, more sufficient training in the later stage of TO. In the process of DEM-TO, we utilize transfer learning. The training for each TO iteration is initialized with the neural network parameters obtained from the end stage of the preceding TO iteration. Thus, with more TO iterations,

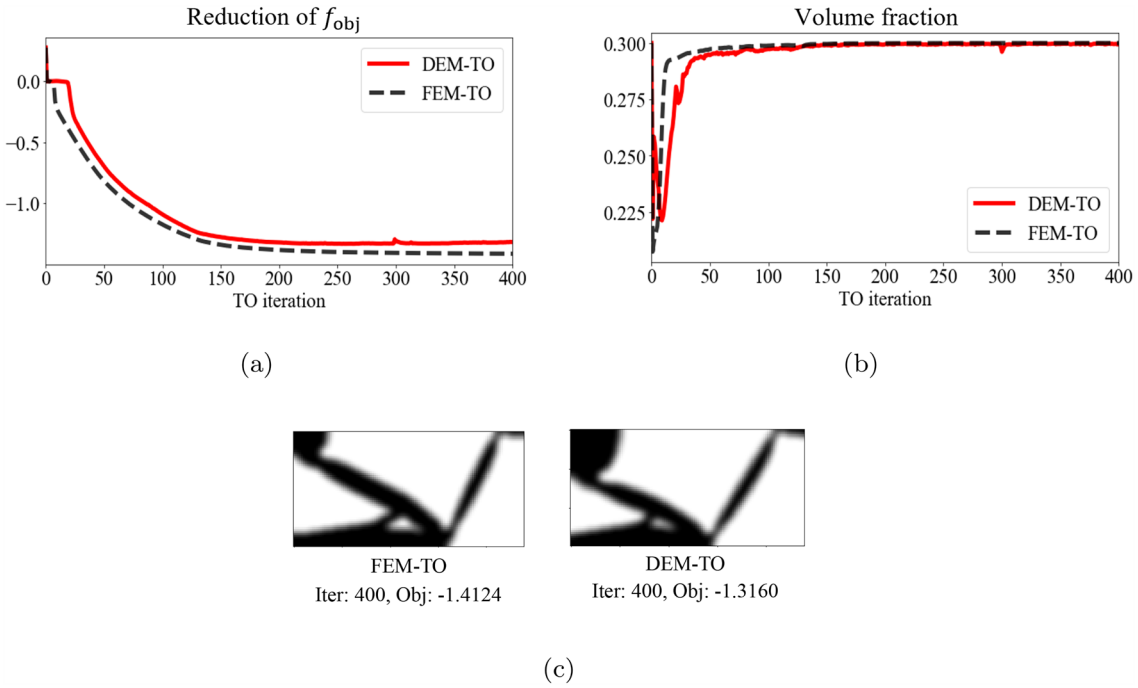


Fig. 13 Comparison of DEM-TO and FEM-TO for compliant mechanism displacement inverter design. **a** The history of the objective function during DEM-based and FEM-based TO. **b** The history of the volume fraction during DEM-based and FEM-based TO. **c** The designs given by FEM-based and DEM-based TO at the TO iteration 400

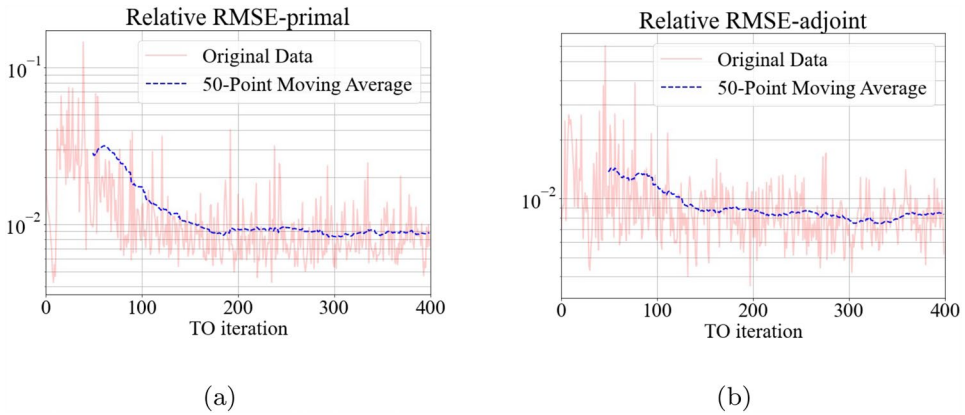


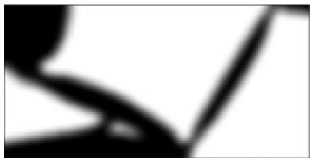
Fig. 14 Relative errors of DEM for **a** primal equation solutions and **b** adjoint equation solutions at various TO iterations for compliant mechanism displacement inverter design

the training of the neural network is more sufficient, which can make a decreasing trend of prediction errors. Second, less design change in the later stage of TO. Figure 16 shows the dissimilarity of the updated design with the old design at various TO iterations. Here the definition of the design dissimilarity is given by

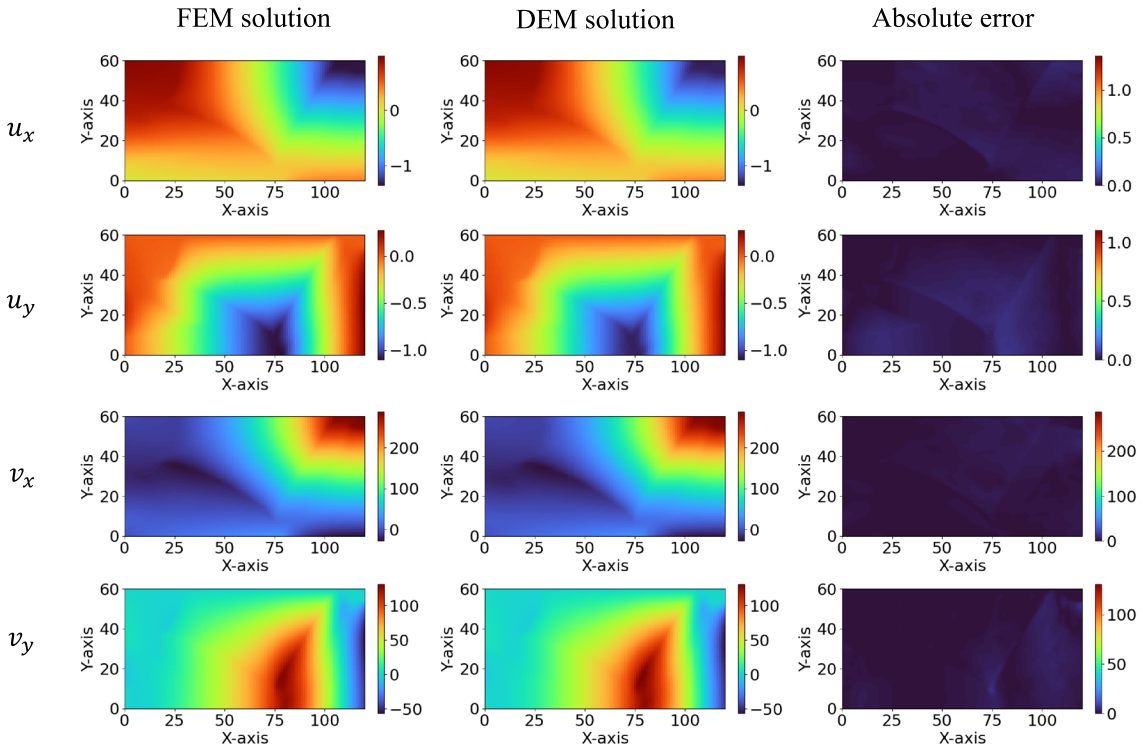
$$\begin{aligned}&\text{Dissimilarity}(\gamma_{\text{old}}, \gamma_{\text{updated}})\\ &= 1 - \text{Cosine Similarity}(\gamma_{\text{old}}, \gamma_{\text{updated}}),\end{aligned} \tag{60}$$

$$\begin{aligned}&\text{Cosine Similarity}(\gamma_{\text{old}}, \gamma_{\text{updated}})\\ &= \frac{\sum_{i=1}^{n} \gamma_{\text{old}_i} \times \gamma_{\text{updated}_i}}{\sqrt{\sum_{i=1}^{n} \gamma_{\text{old}_i}^2} \times \sqrt{\sum_{i=1}^{n} \gamma_{\text{updated}_i}^2}}.\end{aligned} \tag{61}$$

A dissimilarity of 0 implies that the two adjacent designs are perfectly aligned, and a dissimilarity of 1 indicates that the two adjacent design vectors are orthogonal, which means no similarity. From Fig. 16 one can observe a decreasing trend of design dissimilarity, which means the design changes less in the later TO stage. At a DEM-TO iteration, if the updated design is similar to the old design, the initialization of the neural network parameters can be closer to the target values,



(a) DEM-TO design



(b) Solutions for the primal PDE and adjoint PDE

Fig. 15 DEM solutions vs. FEM solutions at TO iteration 400 (Primal solution Relatie RMSE = $9.53 \times e^{-3}$, adjoint solution Relatie RMSE = $1.09 \times e^{-2}$)

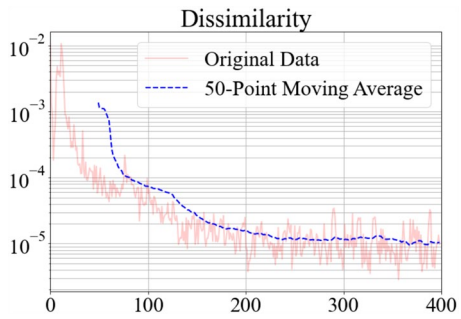


Fig. 16 Dissimilarity of design at various DEM-TO iterations

thus the training at the iteration can be more sufficient and the DEM solutions can be more accurate.

For the oscillation of the DEM solution errors shown in Fig. 14, the reason possibly lies in two aspects: First, the oscillation of design similarity. One can also observe an oscillation in Fig. 16, which means the similarity of adjacent designs oscillates. Second, the oscillation of the DEM solution error may related to the dynamic sampling of the training set. In each DEM training, to avoid the problem of the unbounded loss function, we randomly resample the training points, which may add variability to the training process. This may also affect the DEM solution at various TO iterations, contributing to the oscillation of the DEM solution errors.

4.3 Application to compliant gripper design problem with passive elements

The sensitivity expression of this example is the same as in 4.2. The solution to the adjoint PDE is equivalent to the equilibrium solution of the elastic system depicted in Fig. 5b, the derivation of the adjoint equation is not discussed again.

The domain size, grid resolution, filter radius, volume fraction, and material properties in the complaint inverter example are also used here. The stiffness coefficients of the springs are $k_{\text{in}} = 1, k_{\text{out}} = 0.005$ for the input port and output port respectively.

4.3.1 DEM-TO result vs. FEM-TO result

In Fig. 17 shows the objective function history, volume fraction history, and the optimal TO designs for DEM-TO and FEM-TO. The final objective functions values are $f_{\text{obj}} = -0.6063$ and $f_{\text{obj}} = -0.6200$ for DEM-TO and FEM-TO respectively, which is comparable. Both DEM-TO and FEM-TO minimize the objective function and achieve the optimal designs while satisfying the volume constraint.

4.3.2 Effect of changing N_{epoch} on DEM solution accuracy and DEM-TO efficiency

In this section, we demonstrate that reducing the maximum number of training epochs for each DEM training (N_{epoch}) leads to an increase in the overall solution error of the DEM in the DEM-TO process. However, this reduction enhances the efficiency of the DEM-TO process. With smaller N_{epoch}, a comparable level of design performance can be achieved with fewer total DEM training epochs. Figure 18 shows the DEM solution relative RMSE for the primal PDE and the adjoint PDE for $N_{\text{epoch}} = 200$, $N_{\text{epoch}} = 100$ and $N_{\text{epoch}} = 10$. One can observe that when N_{epoch} decreases, which means the DEM training is less sufficient at each TO iteration, the overall error of DEM solutions increases.

When N_{epoch} is smaller, the total number of DEM training epochs decreases for the same number of TO iterations. For example, for $N_{\text{epoch}} = 200$ the total number of DEM training epochs at TO iteration 400 equals $200 \times 400 \times 2$ =160,000, and for $N_{\text{epoch}} = 10$ the total number of DEM training epochs at TO iteration 400 equals $400 \times 10 \times 2 = 8000$. Figure 19 shows the reduction of the objective function with respect to the total number of DEM training epochs for $N_{\text{epoch}} = 200$, $N_{\text{epoch}} = 100$ and $N_{\text{epoch}} = 10$. It can be observed that to attain an equivalent reduction in f_{obj}, fewer total training epochs are required when N_{epoch} is set to a smaller value, such as 10. With the same number of total training epochs,

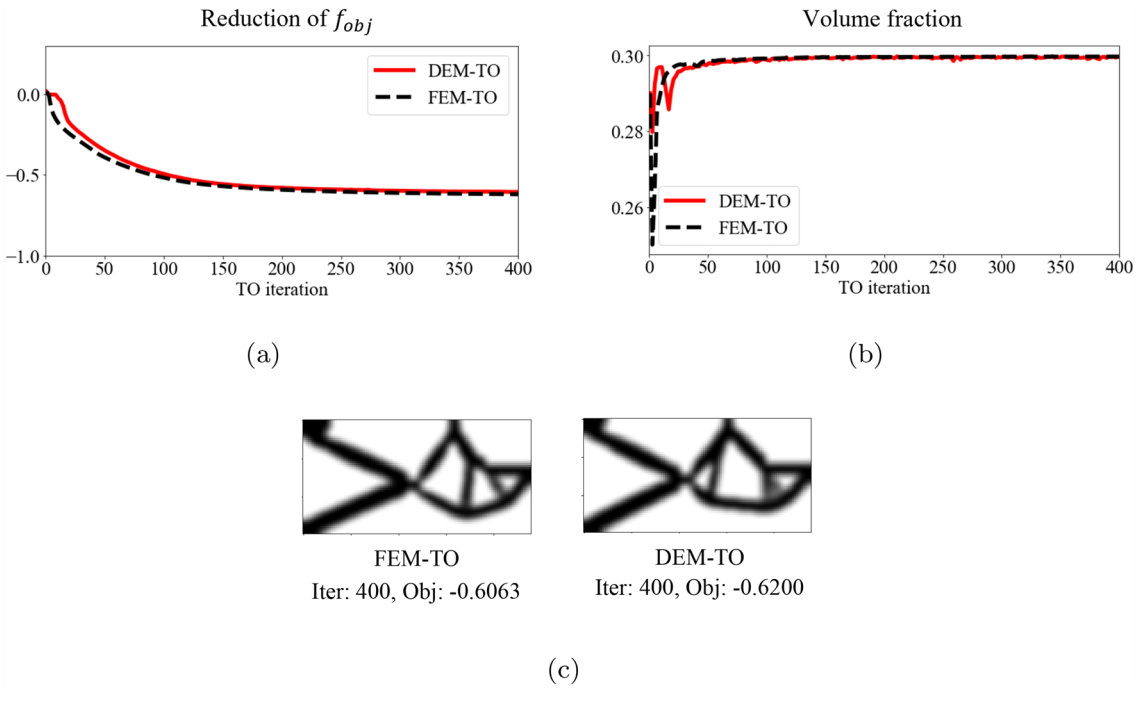
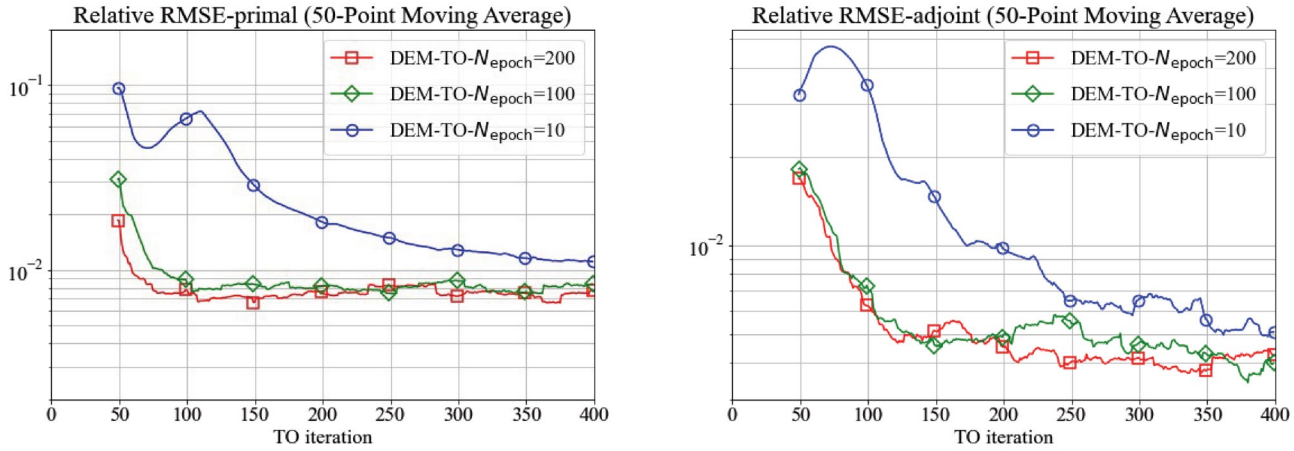


Fig. 17 Comparison of DEM-TO and FEM-TO for compliant mechanism gripper design. **a** The history of the objective function during DEM-based and FEM-based TO. **b** The history of the volume fraction during DEM-based TO and FEM-based TO. **c** The designs given by FEM-based and DEM-based TO at iteration 400



(a) Relative RMSE of DEM solutions - primal PDE

(b) Relative RMSE of DEM solutions - adjoint PDE

Fig. 18 Effect of changing N_{epoch} for DEM-TO on DEM accuracy

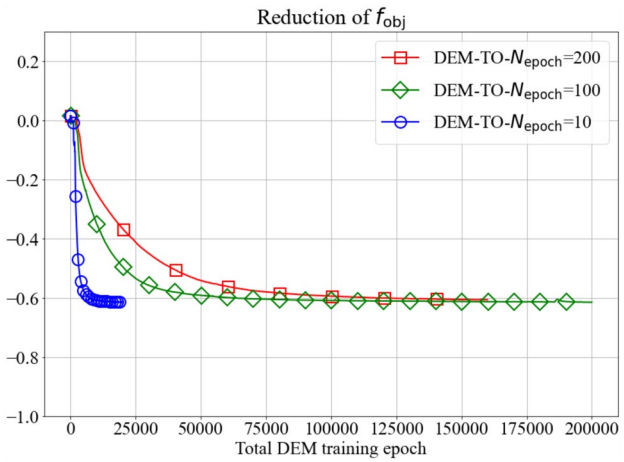


Fig. 19 Effect of changing N_{epoch} on DEM-TO efficiency

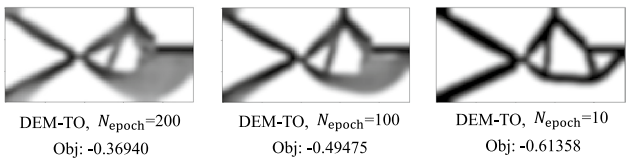


Fig. 20 Comparison of compliant gripper designs obtained from DEM-TO at various choices of N_{epoch}, the total number of training epochs is 20,000 for all three designs

the design performance given by the case $N_{epoch} = 10$ is the best.

In the proposed approach, we do not apply an early stop criterion to DEM training but control the stop of DEM training by setting the maximum number of epochs for each DEM model training. The reason is, that we observe that in the early stages of DEM-TO optimization iterations, the accuracy of solving the primal and adjoint PDEs does not significantly affect the optimality of the final design. Therefore, it is unnecessary for DEM training to fully converge during the early optimization iterations. This is demonstrated in Figs. 18 and 19 of the manuscript. Figure 18 shows that with only 10 epochs of DEM training per iteration, the error in solving the primal and adjoint PDEs is larger in the early optimization iterations compared to 100 and 200 epochs. However, Fig. 19 shows that the final design objective values achieved by DEM-TO are comparable in all three cases.

Figure 20 presents the compliant gripper designs obtained from DEM-TO for $N_{epoch} = 200$, $N_{epoch} = 100$ and $N_{epoch} = 10$. All of those designs are given after 20,000 total training epochs. One can observe that with the same number of total epoches, the DEM-TO design obtained from $N_{epoch} = 10$ is better than DEM-TO design obtained from $N_{epoch} = 100$ and a better than DEM-TO design obtained from $N_{epoch} = 200$.

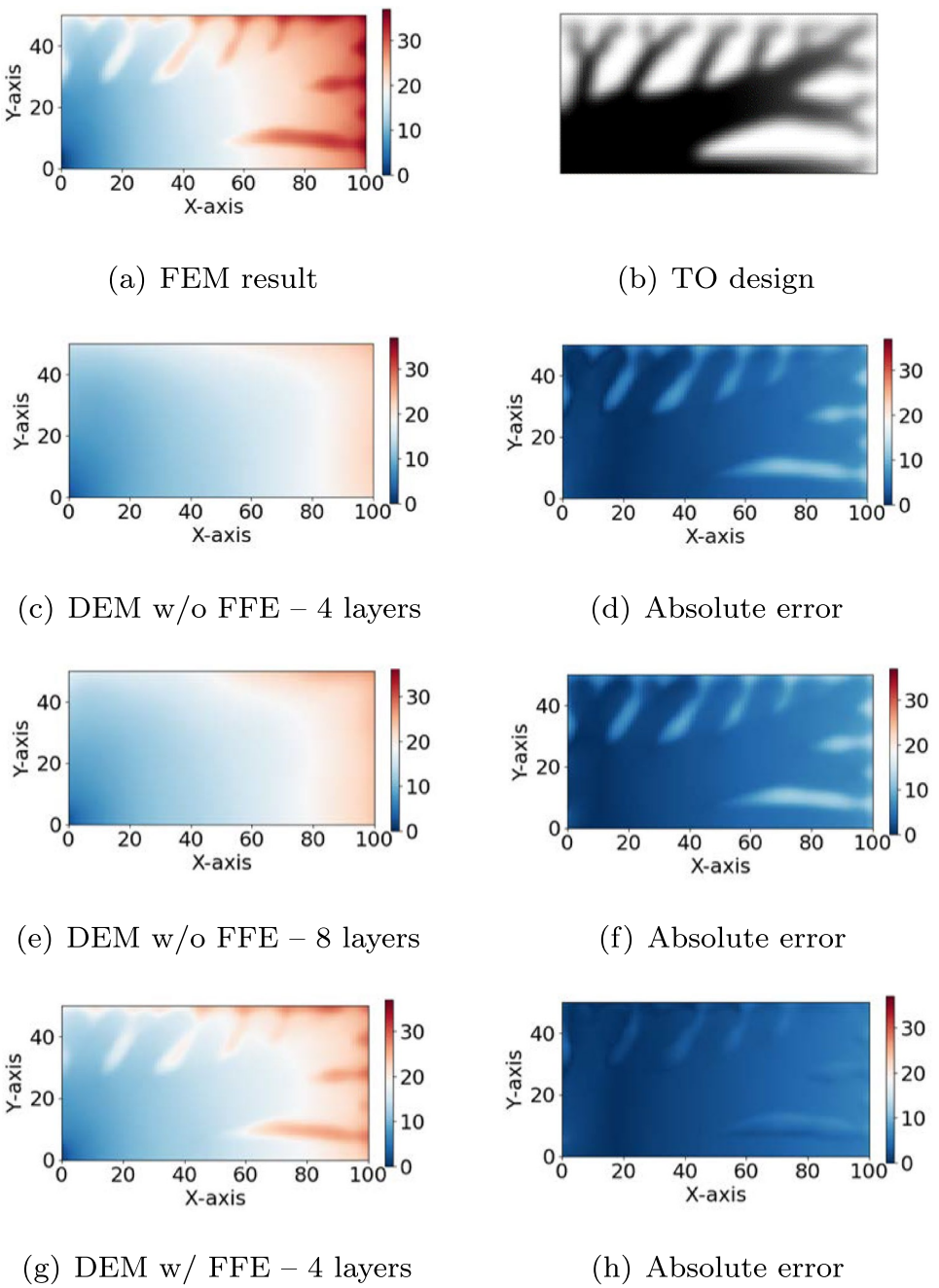
In summary, a smaller N_{epoch} improves DEM-TO efficiency, as fewer total training epochs are needed for equivalent design performance. However, this can result in higher overall errors in DEM solutions, especially in the early stages of DEM-TO. Thus, if the primary need is the final DEM-TO design, with less emphasis on early-stage DEM solution accuracy, a smaller N_{epoch} is a beneficial choice for greater efficiency.

4.4 Ablation studies

4.4.1 The effect of using Fourier Feature Embedding

With the 2D heat conduction problem as an example, the advantages of incorporating Fourier Feature Embedding are

demonstrated. Figure 21 illustrates a comparative analysis of the Deep Energy Method's (DEM) prediction accuracy for a topology-optimized design, examining the influence of Fourier Feature Embedding (FFE). Panel (a) displays the result obtained through the Finite Element Method (FEM), serving as a reference. Panel (b) showcases the optimized design achieved through topology optimization (TO). Panels (c) to panel (h) showcase the DEM prediction results for three different neural network architectures. Panels (c) and (e) depict the DEM predictions without the application of FFE, with four and eight layers respectively. Panels (g) show the DEM prediction utilizing FFE with four layers. The number of training epochs is 1000 for all the three cases. All the other DEM parameters for the three cases are shown in the following Table 1. One can observe that the DEM prediction with Fourier Feature Embedding (FFE) successfully captures fine features near the branches in the TO design, while without FFE, the prediction fails to accurately represent these details, despite an increased number of hidden layers in the neural network. Table 2 quantitatively shows the increase in DEM prediction accuracy for the three cases. One can observe that the using of FFE more effectively reduces the relative



(a) FEM result (b) TO design

(c) DEM w/o FFE – 4 layers (d) Absolute error

(e) DEM w/o FFE – 8 layers (f) Absolute error

(g) DEM w/ FFE – 4 layers (h) Absolute error

Fig. 21 Comparison of DEM prediction accuracy between w/ and w/o Fourier Feature Embedding

RMSE than increasing the number of hidden layers in those three cases.

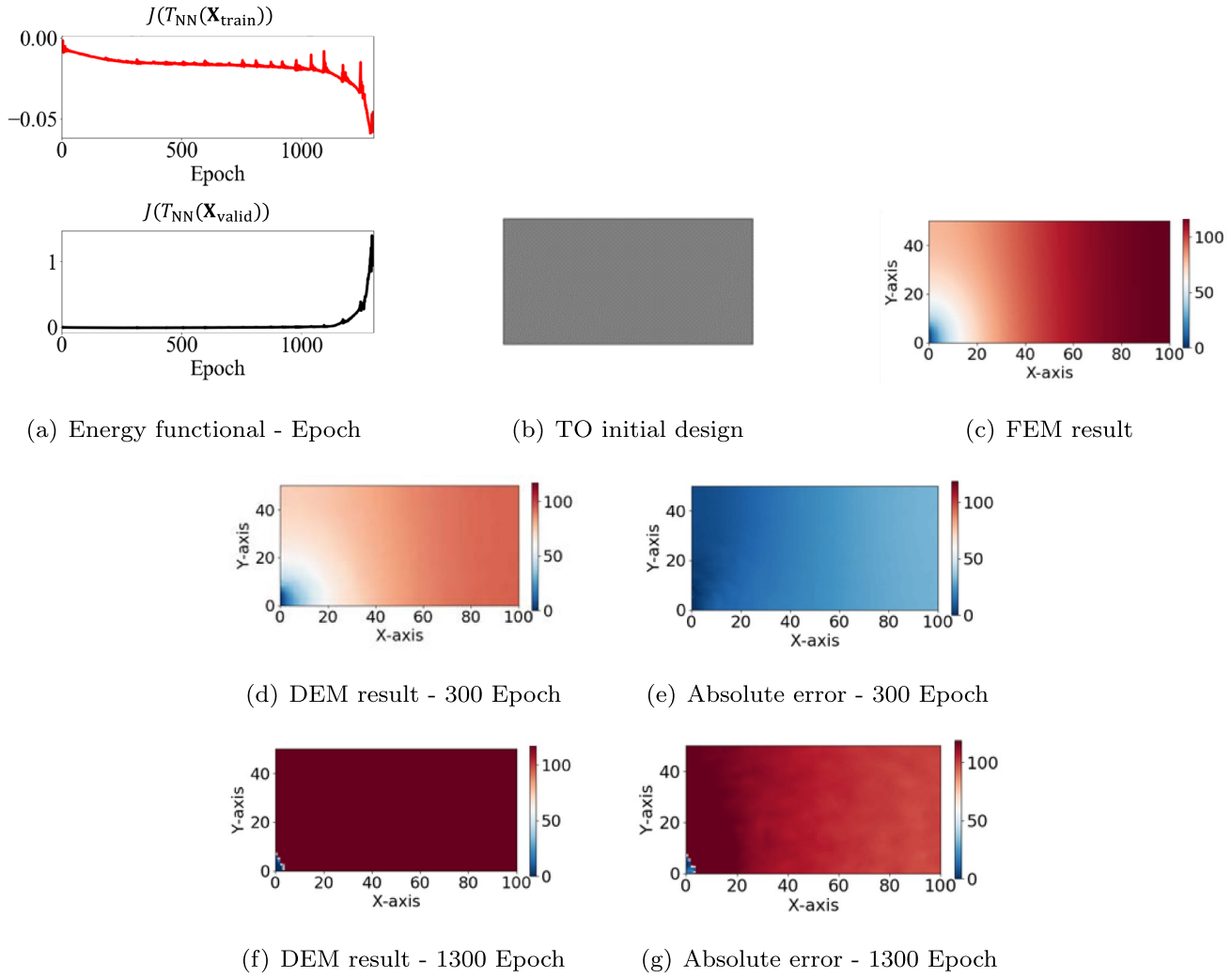
4.4.2 Unbounded loss function given by static training set

The problem of the unbounded loss function given by a static training set in DEM is shown in Fig. 22. The DEM is trained for solving the equilibrium temperature given the TO design in 22c. In panel (a), the red curve shows the thermal energy functional history calculated on a static training set. The black curve shows the thermal energy functional calculated on a validation set, which is also randomly sampled but different from the training set. One can observe the training set energy functional is unbounded, which is continuously decreasing from epoch 1 to epoch 1300, whereas the validation set energy functional decreased from epoch 1 to epoch 300 and began to increase from epoch 300. Moreover, panel (d) and panel (f) depict the DEM prediction result at epoch 300 and epoch 1300 respectively, one can observe that at epoch 1300, the prediction error can be very high. The issue arises from utilizing a static training set, which results in the neural network not being regularized in areas outside of the training points.

In this study, to avoid this problem, we randomly resample the training set in every 10 epochs. Figure 23 shows the effectiveness of resampling in avoiding this problem.

Table 2 Comparison of DEM prediction accuracy between w/ and w/o Fourier Feature Embedding

Network	N_l	N_θ	Relative RMSE (%)
FDNN w/o FFE	4	22,791	14.37
	8	52,719	12.06
FDNN w/ FFE	4	30,101	8.24

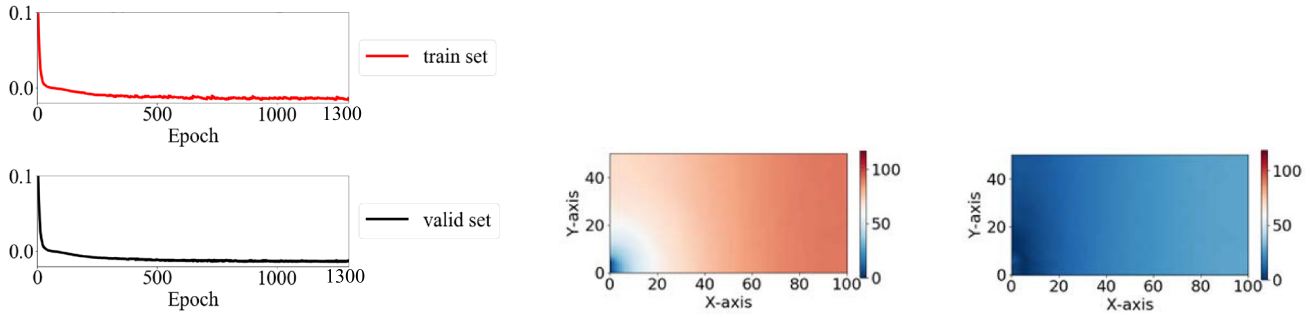


(a) Energy functional - Epoch (b) TO initial design (c) FEM result

(d) DEM result - 300 Epoch (e) Absolute error - 300 Epoch

(f) DEM result - 1300 Epoch (g) Absolute error - 1300 Epoch

Fig. 22 Unbounded loss function in DEM training when using static training set. (Colour figure online)



(a) Energy functional - Epoch (b) DEM result - 1300 Epoch (c) Absolute error

Fig. 23 Effectiveness of resampling on avoiding unbounded loss function

5 Conclusions

We have proposed a DEM-based TO approach using continuous TO formulation, which is applicable to both self-adjoint and non-self-adjoint problems and free from finite element analysis. We apply the proposed DEM-TO framework to a 2D heat sink design problem (self-adjoint case) and two compliant mechanism design problems (non-self-adjoint cases).

Our results validate the DEM-TO framework as a competitive alternative to traditional FEM-based approaches, demonstrating its ability to produce comparable design performances. This study also highlights the importance of using Fourier Feature Embedding to efficiently improve the DEM solution accuracy and the importance of using a dynamic training set to avoid the problem of unbounded DEM loss function. Additionally, we have observed a decreasing trend in the DEM solution error during the DEM-TO process, which can be ascribed to to design convergence and transfer learning effect. Moreover, the effect of changing N_{epoch} on DEM solution accuracy and DEM-TO efficiency is discussed. We demonstrate that a smaller N_{epoch} can lead to higher DEM solutions error in the early stage of DEM-TO, but it can drastically improve the computational efficiency of the proposed DEM-TO framework. On the other hand, a larger N_{epoch} may reduce computational efficiency but allow for more accurate state analysis solutions throughout the entire DEM-TO process at each design update iteration.

In our paper, we briefly discuss the computational time for DEM-TO and FEM-TO based on the hardware and settings used. However, the computational time for these methods depends on various factors, such as whether FEM uses parallel acceleration and GPU, the model used for DEM training, and batch size. A further possible future development consists in exploring a rigorous comparison of the efficiency of DEM-TO and FEM-TO methods, considering these factors. Additionally, in the work done by Nguyen-Thanh et al. (2020), it is practically shown that different integration strategies may lead to varying accuracy of DEM solutions. An interesting future work is exploring the impact of using Simpson's rule, the trapezoidal rule or Gaussian quadrature on the accuracy of DEM solutions with dynamic integration point sets in future work.

Appendix A: steps of deriving the minimal functional formulation for solving the 2D heat conduction problem

For the heat conduction problem (16)–(18), the way of deriving its equivalent minimal functional problem is illustrated as follows:

- First, we derive the weak formulation of the heat conduction equation. We multiply a test function v on both sides of the Eq. (16) and integral both sides over the domain Ω. The weak formulation reads: Find $T \in X \equiv \{w \in H^1(\Omega) | \quad w|_{\Gamma_D} = 0\}$ such that:

$$a(T, v) = f(v), \quad \forall v \in X, \tag{A.1}$$

$$a(T, v) = \int_{\Omega} \kappa(\nabla T \cdot \nabla v) \mathrm{d}\boldsymbol{x}, \tag{A.2}$$

$$f(v) = \int_{\Omega} sv \mathrm{d}\boldsymbol{x}. \tag{A.3}$$

- Then we derive an equivalent functional minimization problem utilizing the weak formulation. For a weak formulation of an elliptic equation ($a(T, v) = f(v), \quad \forall v \in X$), the following proposition holds: If the bilinear form $a(u, v)$ is symmetric and all hypotheses in the Lax-Milgram theorem (Evans 2022;

Le Dret and Lucquin 2016) are satisfied, the unique solution of the weak formulation is the unique solution of:

$$\min_{w\in X} J(w) \quad \text{with} \quad J(w)=\frac{1}{2}a(w,w)-l(w). \tag{A.4}$$

The proposition is true for the weak formulation in (A.1), for which the proof is shown in Appendix B. Thus the solution of weak formulation Eq. (A.1) is the solution of:

$$\min_{T} \quad J(T)=\frac{1}{2}\int_\Omega \kappa|\boldsymbol{\nabla} T|^2\mathrm{d}\boldsymbol{x}-\int_\Omega sT\mathrm{d}\boldsymbol{x}, \tag{A.5}$$

$$\text{s.t.} \quad T=0, \quad \text{for} \quad \boldsymbol{x}\in\Gamma_D. \tag{A.6}$$

Appendix B: proof of Lax–Milgram theorem for the heat conduction problem

For the 2D heat conduction problem, the weak form is

$$a(T,v)=f(v), \quad \forall v\in X\equiv\{w\in H^1(\Omega)| \quad w|_{\Gamma_D}=0\}, \tag{B.1}$$

$$a(T,v)=\int_\Omega \kappa(\boldsymbol{\nabla} T\cdot\boldsymbol{\nabla} v)\mathrm{d}\boldsymbol{x}, \tag{B.2}$$

$$f(v)=\int_\Omega sv\mathrm{d}\boldsymbol{x}-\int_{\Gamma_N} qv\mathrm{d}\boldsymbol{s}. \tag{B.3}$$

Given

$$s=0.001, \tag{B.4}$$

$$q=0, \tag{B.5}$$

$$T_d=0, \tag{B.6}$$

the $f(v)$ becomes

$$f(v)=\int_\Omega sv\mathrm{d}\boldsymbol{x}. \tag{B.7}$$

For the heat conductivity κ, we have $0<\kappa_{\min}\le\kappa\le\kappa_0$.

- By applying Cauchy–Schwarz inequality and Poincaré inequality, we have

$$|a(T,v)|=\left|\int_\Omega \kappa(\boldsymbol{\nabla} T\cdot\boldsymbol{\nabla} v)\mathrm{d}\boldsymbol{x}\right| \tag{B.8}$$

$$\le\int_\Omega |\kappa||\boldsymbol{\nabla} T||\boldsymbol{\nabla} v|\mathrm{d}\boldsymbol{x} \tag{B.9}$$

$$\le\kappa_0\int_\Omega |\boldsymbol{\nabla} T||\boldsymbol{\nabla} v|\mathrm{d}\boldsymbol{x} \tag{B.10}$$

$$\le\kappa_0\|\boldsymbol{\nabla} T\|_{L^2(\Omega)}\|\boldsymbol{\nabla} v\|_{L^2(\Omega)} \tag{B.11}$$

$$\le\kappa_0\|T\|_{H^1}\|v\|_{H^1} \tag{B.12}$$

The bilinear form a is continuous.

- Also, we have

$$a(T,v)=\int_\Omega \kappa|\boldsymbol{\nabla} T|^2\mathrm{d}\boldsymbol{x} \tag{B.13}$$

$$\ge\kappa_{\min}\int_\Omega |\boldsymbol{\nabla} T|^2\mathrm{d}\boldsymbol{x} \tag{B.14}$$

$$=\kappa_{\min}\|\boldsymbol{\nabla} T\|^2_{L^2(\Omega)}. \tag{B.15}$$

By Poincaré inequality we have

$$\|T\|_{L^2(\Omega)}\le C\|\boldsymbol{\nabla} T\|_{L^2(\Omega)}, \tag{B.16}$$

which give us

$$a(T,v)\ge\alpha\|T\|^2_{H^1(\Omega)}. \tag{B.17}$$

Here α is taken to be the minimum of $\kappa_{\min}$ and $k_{\min}/C^2$. The bilinear form a is V-elliptic.

- For the linear form $f(v)$, we have

$$|f(v)|=\int_\Omega sv\mathrm{d}\boldsymbol{x}\le s\int_\Omega |v|\mathrm{d}\boldsymbol{x}=s\|v\|_{L^2(\Omega)}. \tag{B.18}$$

The linear form f is continuous.

Thus the conditions in the Lax–Milgram theorem are met.

Appendix C: derivation of continuous adjoint sensitivity analysis

Appendix C.1: 2D heat conduction problem

For the TO problem defined in (15)–(20), the continuous adjoint method is employed to determine the sensitivity of the objective function with respect to the physical density variables $\frac{\partial f_{\text{obj}}}{\partial\tilde{\gamma}}$, we use the continuous adjoint method. The augmented objective function is given by:

$$\begin{aligned} L=&\frac{1}{2}\int_\Omega \kappa|\boldsymbol{\nabla} T|^2\mathrm{d}\boldsymbol{x}\\ &+\int_\Omega \mathbf{v}(\boldsymbol{\nabla}\cdot(\kappa\boldsymbol{\nabla} T)+s)\mathrm{d}\boldsymbol{x}. \end{aligned} \tag{C.1}$$

By applying integration by parts, divergence theorem, and rearranging terms, we have:

$$L = \frac{1}{2}\int_\Omega \kappa|\nabla T|^2 \mathrm{d}\boldsymbol{x} + \int_\Gamma \mathbf{v}\kappa\nabla T\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} - \int_\Omega \kappa\nabla T\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} + \int_\Omega \mathbf{v}s\mathrm{d}\boldsymbol{x}. \tag{C.2}$$

The first-order variation of the augmented Lagrange function L with respect to the variation of physical density variable $\tilde{\gamma}$ is expressed as:

$$\delta L = \frac{1}{2}\int_\Omega \delta[\kappa(\nabla T\cdot\nabla T)]\mathrm{d}\boldsymbol{x} + \int_\Gamma \mathbf{v}\delta(\kappa\nabla T)\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} - \int_\Omega \delta(\kappa\nabla T)\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} + \int_\Omega \mathbf{v}\delta s\mathrm{d}\boldsymbol{x}. \tag{C.3}$$

By substituting

$$\frac{1}{2}\int_\Omega \delta[\kappa(\nabla T\cdot\nabla T)]\mathrm{d}\boldsymbol{x} = \frac{1}{2}\int_\Omega \delta\kappa\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x} + \int_\Omega s\delta T\mathrm{d}\boldsymbol{x}, \tag{C.4}$$

$$-\int_\Omega \delta(\kappa\nabla T)\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} = -\int_\Omega \delta\kappa\nabla T\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} - \int_\Gamma \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s} + \int_\Omega \nabla\cdot(\kappa\nabla\mathbf{v})\delta T\mathrm{d}\boldsymbol{x}, \tag{C.5}$$

we have

$$\delta L = \frac{1}{2}\int_\Omega \delta\kappa\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x} + \int_\Omega s\delta T\mathrm{d}\boldsymbol{x} \tag{C.6}$$

$$+\int_\Gamma \mathbf{v}\delta(\kappa\nabla T\cdot\mathbf{n})\mathrm{d}\boldsymbol{s} \tag{C.7}$$

$$-\int_\Omega \delta\kappa\nabla T\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} - \int_\Gamma \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s} + \int_\Omega \nabla\cdot(\kappa\nabla\mathbf{v})\delta T\mathrm{d}\boldsymbol{x} \tag{C.8}$$

$$+\int_\Omega \mathbf{v}\delta s\mathrm{d}\boldsymbol{x}. \tag{C.9}$$

For the boundary integral terms, we have

$$\int_\Gamma \mathbf{v}\delta(\kappa\nabla T\cdot n)\mathrm{d}\boldsymbol{s} - \int_\Gamma \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s} \tag{C.10}$$

$$= \int_{\Gamma_D} \mathbf{v}\delta\kappa\nabla T\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} + \int_{\Gamma_D} \mathbf{v}[\kappa\nabla(\delta T)\cdot\mathbf{n}]\mathrm{d}\boldsymbol{s} + \int_{\Gamma_N} \mathbf{v}\delta(\kappa\nabla T\cdot\mathbf{n})\mathrm{d}\boldsymbol{s} \tag{C.11}$$

$$-\int_{\Gamma_D} \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s} - \int_{\Gamma_N} \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s}. \tag{C.12}$$

Since $\delta(\kappa\nabla T\cdot\mathbf{n}) = 0$ on Γ_N, the term $\int_{\Gamma_N} \mathbf{v}\delta(\kappa\nabla T\cdot\mathbf{n})\mathrm{d}\boldsymbol{s}$ vanishes. Since $\delta T = 0$ on Γ_D, the term $-\int_{\Gamma_D} \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s}$ vanishes. Since $\delta s = 0$ on Ω, the term $\int_\Omega \mathbf{v}\delta s\mathrm{d}\boldsymbol{x}$ vanishes. Then we can reformulate δL as:

$$\delta L = \frac{1}{2}\int_\Omega \delta\kappa\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x} - \int_\Omega \delta\kappa\nabla T\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} + \int_{\Gamma_D} \mathbf{v}\delta\kappa\nabla T\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} \tag{C.13}$$

$$+\int_\Omega [\nabla\cdot(\kappa\nabla\mathbf{v}) + s]\delta T\mathrm{d}\boldsymbol{x} + \int_{\Gamma_D} \mathbf{v}[\kappa\nabla(\delta T)\cdot\mathbf{n}]\mathrm{d}\boldsymbol{s} - \int_{\Gamma_N} \kappa\nabla\mathbf{v}\cdot\mathbf{n}\delta T\mathrm{d}\boldsymbol{s}. \tag{C.14}$$

We can let

$$\nabla\cdot(\kappa\nabla\mathbf{v}) + s = 0, \quad \boldsymbol{x}\in\Omega, \tag{C.15}$$

$$-\kappa\nabla\mathbf{v}\cdot\mathbf{n} = 0, \quad \boldsymbol{x}\in\Gamma_N, \tag{C.16}$$

$$\mathbf{v} = 0, \quad \boldsymbol{x}\in\Gamma_D. \tag{C.17}$$

Then the terms in (C.14) can be eliminated. The equations in (C.15)–(C.17) are the continuous adjoint equations for the aforementioned minimal thermal compliance problem, and we call the governing equations in (16)–(18) the primal equations for the minimal thermal compliance problem.

Figure 3 shows the design domain and boundary conditions in the 2D heat conduction problem, Fig. 3a is the original boundary conditions, Fig. 3b shows the reduction of the numerical domain by applying a symmetry treatment. In this study, for this 2D heat conduction state analysis, we consider the boundary conditions as

$$s = 0.001, \tag{C.18}$$

$$q = 0, \tag{C.19}$$

$$T_d = 0. \tag{C.20}$$

We can notice that with $q = 0$ and $T_d = 0$ for (16)–(18), the primal equations and the adjoint equations share the same boundary conditions and governing equations, which means $\mathbf{v} = T$. In this case, we call the aforementioned topology optimization problem self-adjoint. The sensitivity expression in (C.13) can be reduced to:

$$\delta L = -\frac{1}{2}\int_\Omega \delta\kappa\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x}. \tag{C.21}$$

Since $\delta\kappa = \frac{\mathrm{d}\kappa}{\mathrm{d}\gamma}\delta\tilde{\gamma}$, the (C.21) becomes

$$\delta L = -\frac{1}{2}\int_{\Omega}\frac{\mathrm{d}\kappa}{\mathrm{d}\tilde{\gamma}}\delta\tilde{\gamma}\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x}. \tag{C.22}$$

We can relate the variational form of the sensitivity with the partial derivative $\frac{\partial L}{\partial\tilde{\gamma}_i}$:

$$\frac{\partial f_{\mathrm{obj}}}{\partial\tilde{\gamma}_i} = -\frac{1}{2}\frac{\mathrm{d}\kappa_i}{\mathrm{d}\tilde{\gamma}_i}\int_{\Omega_i}\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x}, \tag{C.23}$$

in which κ_i denotes the heat conductivity value in the ith element, Ω_i denotes the region in the i element.

Conversely, if the primal equations and the adjoint equations do not share the same boundary conditions and governing equations, we need to solve primal equations and adjoint equations separately. The sensitivity expression in (C.13) is:

$$\delta L = \frac{1}{2}\int_{\Omega}\delta\kappa\nabla T\cdot\nabla T\mathrm{d}\boldsymbol{x} - \int_{\Omega}\delta\kappa\nabla T\cdot\nabla\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{C.24}$$

Appendix C.2: compliant mechanism design problems

For the two compliant mechanism design problems, to obtain the sensitivity of the objective function with respect to the physical density variables $\tilde{\gamma}$, we employ the continuous adjoint method. The augmented objective function is formulated as below:

$$L = -u_{\mathrm{out}} + \int_{\Omega}\mathbf{v}\cdot(\nabla\cdot\boldsymbol{\sigma}+\boldsymbol{b})\mathrm{d}\boldsymbol{x}. \tag{C.25}$$

The variation of the augmented Lagrangian function with respect to the variation of the physical design variable $\tilde{\gamma}$ can be expressed as:

$$\delta L = -\delta u_{\mathrm{out}} + \int_{\Omega}\mathbf{v}\cdot\nabla\cdot(\delta\boldsymbol{\sigma})\mathrm{d}\boldsymbol{x} + \int_{\Omega}\mathbf{v}\cdot\delta\mathbf{b}\mathrm{d}\boldsymbol{x} \tag{C.26}$$

$$= -\delta u_{\mathrm{out}} + \int_{\Omega}\mathbf{v}\cdot\nabla\cdot(\delta\boldsymbol{\sigma})\mathrm{d}\boldsymbol{x}. \tag{C.27}$$

Since $\mathbf{b}=0$, we have $\delta\mathbf{b}=0$, allowing us to omit the last term. Applying integration by parts and the divergence theorem to the second term, we obtain:

$$\int_{\Omega}\mathbf{v}\cdot\nabla\cdot(\delta\boldsymbol{\sigma})\mathrm{d}\boldsymbol{x} = -\int_{\Omega}\delta\boldsymbol{\sigma}:\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} + \int_{\Gamma}\mathbf{v}\cdot\delta\boldsymbol{\sigma}\cdot\mathbf{n}\mathrm{d}s \tag{C.28}$$

$$= -\int_{\Omega}\left(\delta\boldsymbol{\sigma}^{\mathbf{I}}+\delta\boldsymbol{\sigma}^{\mathbf{II}}\right):\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} + \int_{\Gamma}\mathbf{v}\cdot\delta\boldsymbol{\sigma}\cdot\mathbf{n}\mathrm{d}s. \tag{C.29}$$

where the $\delta\boldsymbol{\sigma}^{\mathbf{I}}$ and $\delta\boldsymbol{\sigma}^{\mathbf{II}}$ are defined as

$$\delta\boldsymbol{\sigma}^{\mathbf{I}}_{ij} = \mu\left(\frac{\partial\delta u_i}{\partial x_j}+\frac{\partial\delta u_j}{\partial x_i}\right)+\lambda\delta_{ij}\frac{\partial\delta u_m}{\partial x_m} \tag{C.30}$$

$$\delta\boldsymbol{\sigma}^{\mathbf{II}}_{ij} = \delta\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\delta\lambda\delta_{ij}\frac{\partial u_m}{\partial x_m}. \tag{C.31}$$

By applying integration by parts and the Gaussian divergence theorem, we arrive at:

$$-\int_{\Omega}\delta\boldsymbol{\sigma}^{\mathbf{I}}:\nabla\mathbf{v}\mathrm{d}\boldsymbol{x} = \int_{\Omega}\delta\mathbf{u}\cdot(\nabla\cdot\boldsymbol{\sigma}^{\mathbf{v}})\mathrm{d}\boldsymbol{x} - \int_{\Gamma}\delta\mathbf{u}\cdot\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}\mathrm{d}s, \tag{C.32}$$

where $\boldsymbol{\sigma}^{\mathbf{v}}$ is defined as:

$$\boldsymbol{\sigma}^{\mathbf{v}}_{ij} = \mu\left(\frac{\partial\mathbf{v}_i}{\partial x_j}+\frac{\partial\mathbf{v}_j}{\partial x_i}\right)+\lambda\delta_{ij}\frac{\partial\mathbf{v}_m}{\partial x_m}. \tag{C.33}$$

By substitutions and rearrangement, we get the expression for the variation of the augmented Lagrangian function:

$$\delta L = -\delta u_{\mathrm{out}} + \int_{\Omega}\delta\mathbf{u}\cdot(\nabla\cdot\boldsymbol{\sigma}^{\mathbf{v}})\mathrm{d}\boldsymbol{x} + \int_{\Gamma}\mathbf{v}\cdot\delta\boldsymbol{\sigma}\cdot\mathbf{n}\mathrm{d}s - \int_{\Gamma}\delta\mathbf{u}\cdot\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}\mathrm{d}s - \int_{\Omega}\delta\boldsymbol{\sigma}^{\mathbf{II}}:\nabla\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{C.34}$$

We can express $\delta\boldsymbol{\sigma}^{\mathbf{II}}$ as:

$$\begin{aligned}\delta\boldsymbol{\sigma}^{\mathbf{II}} =& \frac{\mathrm{d}E}{\mathrm{d}\tilde{\gamma}}\delta\tilde{\gamma}\left[\left(\frac{1}{2(1+\nu)}\right)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\right.\\ &\left.+\frac{\mu}{(1+\nu)(1-2\nu)}\delta_{ij}\frac{\partial u_m}{\partial x_m}\right]\\ =& \frac{\mathrm{d}E}{\mathrm{d}\tilde{\gamma}}\delta\tilde{\gamma}\hat{\boldsymbol{\sigma}},\end{aligned} \tag{C.35}$$

in which $\hat{\boldsymbol{\sigma}}$ represents the dimensionless stress tensor. Then we can rewrite the variation of the augmented Lagrange function as:

$$\delta L = -\delta u_{\mathrm{out}} + \int_{\Omega}\delta\mathbf{u}\cdot(\nabla\cdot\boldsymbol{\sigma}^{\mathbf{v}})\mathrm{d}\boldsymbol{x} \tag{C.36}$$

$$+\int_{\Gamma}\mathbf{v}\cdot\delta\boldsymbol{\sigma}\cdot\mathbf{n}\mathrm{d}s \tag{C.37}$$

$$-\int_{\Gamma}\delta\mathbf{u}\cdot\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}\mathrm{d}s \tag{C.38}$$

$$-\int_{\Omega}\frac{\mathrm{d}E}{\mathrm{d}\gamma}\delta\gamma\hat{\boldsymbol{\sigma}}:\boldsymbol{\nabla}\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{C.39}$$

The strong form of the adjoint equation is given by eliminating the term that depends on $\delta\mathbf{u}$ for $\boldsymbol{x}\in\Omega$, which is

$$\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}^{\mathbf{v}}=0, \tag{C.40}$$

The boundary conditions for the adjoint problem are derived by eliminating the boundary integral terms dependent on $\delta\mathbf{u}$. Here we take the compliant inverter design problem as an example. The boundary conditions of the primal equations for the complaint inverter design problems are shown in Fig. 4a. Mathematically, the boundary conditions are:

$$\mathbf{u}\cdot\boldsymbol{e}_y=0,\quad \boldsymbol{\sigma}\cdot\mathbf{n}\cdot\boldsymbol{e}_x=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{tp}}, \tag{C.41}$$

$$\mathbf{u}=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{lb}}, \tag{C.42}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n}=\delta_{\text{Direc}}(\boldsymbol{x}-\boldsymbol{x}_{\text{out}})[-k_{\text{out}}(\mathbf{u}\cdot\boldsymbol{e}_x)]\boldsymbol{e}_x,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{out}}, \tag{C.43}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n}=\delta_{\text{Direc}}(\boldsymbol{x}-\boldsymbol{x}_{\text{in}})(-k_{\text{in}}\mathbf{u}\cdot\boldsymbol{e}_x+f_{\text{in}})\boldsymbol{e}_x,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{in}}, \tag{C.44}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n}=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{rm}}. \tag{C.45}$$

From the variations of the boundary conditions, we have

$$\delta\mathbf{u}\cdot\boldsymbol{e}_y=0,\quad \delta\boldsymbol{\sigma}\cdot\mathbf{n}\cdot\boldsymbol{e}_x=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{tp}, \tag{C.46}$$

$$\delta\mathbf{u}=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{lb}}, \tag{C.47}$$

$$\delta\boldsymbol{\sigma}\cdot\mathbf{n}=\delta_{\text{Direc}}(\boldsymbol{x}-\boldsymbol{x}_{\text{out}})[-k_{\text{out}}(\delta\mathbf{u}\cdot\boldsymbol{e}_x)]\boldsymbol{e}_x,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{out}}, \tag{C.48}$$

$$\delta\boldsymbol{\sigma}\cdot\mathbf{n}=\delta_{\text{Direc}}(\boldsymbol{x}-\boldsymbol{x}_{\text{in}})(-k_{\text{in}}\delta\mathbf{u}\cdot\boldsymbol{e}_x)\boldsymbol{e}_x,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{in}}, \tag{C.49}$$

$$\delta\boldsymbol{\sigma}\cdot\mathbf{n}=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{rm}}. \tag{C.50}$$

For the variation of the objective function $-\mathbf{u}_{\text{out}}$, we have

$$-\delta\mathbf{u}_{\text{out}}=\int_{\Gamma_{\text{out}}}\delta\boldsymbol{u}\cdot(\delta_{\text{Direc}}(\boldsymbol{x}-\boldsymbol{x}_{\text{out}})\boldsymbol{e}_x)\mathrm{d}\boldsymbol{s} \tag{C.51}$$

By expanding the two boundary integral terms (C.37) and (C.38), and substituting Eqs. (C.46)–(C.51), we can rewrite δL as

$$\delta L=\int_{\Omega}\delta\mathbf{u}\cdot(\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}^{\mathbf{v}})\mathrm{d}\boldsymbol{x} \tag{C.52}$$

$$+\int_{\Gamma_{\text{tp}}}\mathbf{v}\cdot[(\delta\boldsymbol{\sigma}\cdot\mathbf{n}\cdot\boldsymbol{e}_y)\boldsymbol{e}_y]+(\delta\mathbf{u}\cdot\boldsymbol{e}_x)\boldsymbol{e}_x\cdot\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} \tag{C.53}$$

$$+\int_{\Gamma_{\text{lb}}}\mathbf{v}\cdot\delta\boldsymbol{\sigma}\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} \tag{C.54}$$

$$-\int_{\Gamma_{\text{rm}}}\delta\mathbf{u}\cdot\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}\mathrm{d}\boldsymbol{s} \tag{C.55}$$

$$+\int_{\Gamma_{\text{out}}}\delta\mathbf{u}\cdot[\delta_{\text{Direc}}(\mathbf{x}-\mathbf{x}_{\text{out}})[-k_{\text{out}}(\mathbf{v}\cdot\boldsymbol{e}_x)]\boldsymbol{e}_x+\delta_{\text{Direc}}(\mathbf{x}-\mathbf{x}_{\text{out}})\boldsymbol{e}_x-\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}]\mathrm{d}\boldsymbol{s} \tag{C.56}$$

$$+\int_{\Gamma_{\text{in}}}\delta\mathbf{u}\cdot[\delta_{\text{Direc}}(\mathbf{x}-\mathbf{x}_{\text{in}})[-k_{\text{in}}(\mathbf{v}\cdot\boldsymbol{e}_x)]\boldsymbol{e}_x-\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}]\mathrm{d}\boldsymbol{s} \tag{C.57}$$

$$-\int_{\Omega}\frac{\mathrm{d}E}{\mathrm{d}\tilde{\gamma}}\delta\tilde{\gamma}\hat{\boldsymbol{\sigma}}:\boldsymbol{\nabla}\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{C.58}$$

In order to eliminate the boundary terms that depend on $\delta\mathbf{u}$, we need:

$$\mathbf{v}\cdot\boldsymbol{e}_y=0,\quad \boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}\cdot\boldsymbol{e}_x=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{tp}}, \tag{C.59}$$

$$\mathbf{v}=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{lb}}, \tag{C.60}$$

$$\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}=\delta_{\text{Direc}}(\mathbf{x}-\mathbf{x}_{\text{out}})[-k_{\text{out}}(\mathbf{v}\cdot\boldsymbol{e}_x)+1]\boldsymbol{e}_x,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{out}}, \tag{C.61}$$

$$\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}=\delta_{\text{Direc}}(\boldsymbol{x}-\boldsymbol{x}_{\text{in}})[-k_{\text{in}}(\mathbf{v}\cdot\boldsymbol{e}_x)]\boldsymbol{e}_x,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{in}}, \tag{C.62}$$

$$\boldsymbol{\sigma}^{\mathbf{v}}\cdot\mathbf{n}=0,\quad \text{for}\quad \boldsymbol{x}\in\Gamma_{\text{rm}}, \tag{C.63}$$

that are the boundary conditions for the adjoint problem. The visualizations of the adjoint problem boundary conditions for the compliant inverter are shown in Fig. 4b. Similarly, we can derive the adjoint problem boundary conditions for the complaint gripper design problem, which is visualized in Fig. 5b.

After term elimination, the remaining terms express the sensitivity of the objective function with respect to the physical density variable $\tilde{\gamma}$:

$$\delta f_{\text{obj}}=\delta L=-\int_{\Omega}\frac{\mathrm{d}E}{\mathrm{d}\tilde{\gamma}}\delta\tilde{\gamma}\hat{\boldsymbol{\sigma}}:\boldsymbol{\nabla}\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{C.64}$$

We can relate the variational form expression (C.64) to the partial derivative form $\frac{\partial J}{\partial\tilde{\gamma}_i}$ as

$$\frac{\partial f_{\text{obj}}}{\partial\tilde{\gamma}_i}=-\frac{\mathrm{d}E_i}{\mathrm{d}\tilde{\gamma}_i}\int_{\Omega_i}\hat{\boldsymbol{\sigma}}:\boldsymbol{\nabla}\mathbf{v}\mathrm{d}\boldsymbol{x}. \tag{C.65}$$

Acknowledgements The support from US National Science Foundation Grant 2219931 is greatly appreciated. FM acknowledges the support of the research fund *FAR 2023 DIP - Fondo di Ateneo per*

la Ricerca per il finanziamento di progetti di ricerca dipartimentali (Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia).

Author contributions Xueqi Zhao: conceptualization, methodology, formal analysis, software, validation, data curation, investigation, writing—original draft, writing—review & editing. Francesco Mezzadri: conceptualization, methodology, writing—review & editing. Tianye Wang: conceptualization, methodology, writing—review & editing. Xiaoping Qian: conceptualization, methodology, formal analysis, project administration, writing—review & editing, supervision, funding acquisition.

Declarations

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

Replication of results The source codes related to this study are available in the GitHub repository: https://github.com/xzhao399/DEM_TO.git. All the datasets generated in this work are available upon reasonable request to the corresponding author.

References

Aage N (2019) Topology optimization codes written in Python. https://www.topopt.mek.dtu.dk/apps-and-software/topology-optimization-codes-written-in-python

Bendsoe MP, Sigmund O (2003) Topology optimization: theory, methods, and applications. Springer Science & Business Media, Berlin

Bergstra J, Yamins D, Cox DD (2013) Hyperopt: a Python library for optimizing the hyperparameters of machine learning algorithms. In: Proceedings of the 12th Python in science conference, vol 13. CiteSeer, p 20

Brunton SL, Kutz JN (2022) Data-driven science and engineering: machine learning, dynamical systems, and control. Cambridge University Press, Cambridge

Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE (2021) Physics-informed neural networks for heat transfer problems. J Heat Transf 143(6):060801

Chandrasekhar A, Suresh K (2021) TOuNN: topology optimization using neural networks. Struct Multidisc Optim 63:1135–1149

Costabal FS, Pezzuto S, Perdikaris P (2024) δ-PINNs: physics-informed neural networks on complex geometries. Eng Appl Artif Intell 127:107324

Eschenauer HA, Olhoff N (2001) Topology optimization of continuum structures: a review. Appl Mech Rev 54(4):331–390

Evans LC (2022) Partial differential equations, vol 19. American Mathematical Society, Providence

Frank M, Drikakis D, Charissis V (2020) Machine-learning methods for computational science and engineering. Computation 8(1):15

Fuhg JN, Bouklas N (2022) The mixed deep energy method for resolving concentration features in finite strain hyperelasticity. J Comput Phys 451:110839

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. Comput Methods Appl Mech Eng 379:113741

He Q, Barajas-Solano D, Tartakovsky G, Tartakovsky AM (2020) Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. Adv Water Resour 141:103610

He J, Chadha C, Kushwaha S, Koric S, Abueidda D, Jasiuk I (2023) Deep energy method in topology optimization applications. Acta Mech 234(4):1365–1379

Jagtap AD, Karniadakis GE (2021) Extended physics-informed neural networks (XPINNs): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. In: AAAI spring symposium: MLPS, vol 10

Jeong H, Bai J, Batuwatta-Gamage CP, Rathnayaka C, Zhou Y, Gu Y (2023a) A physics-informed neural network-based topology optimization (PINNTO) framework for structural optimization. Eng Struct 278:115484

Jeong H, Batuwatta-Gamage C, Bai J, Xie YM, Rathnayaka C, Zhou Y, Gu Y (2023b) A complete physics-informed neural network-based framework for structural topology optimization. Comput Methods Appl Mech Eng 417:116401

Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. Nat Rev Phys 3(6):422–440

Kharazmi E, Zhang Z, Karniadakis GE (2019) Variational physics-informed neural networks for solving partial differential equations. arXiv Preprint. arXiv:1912.00873

Kharazmi E, Zhang Z, Karniadakis GE (2021) hp-VPINNs: variational physics-informed neural networks with domain decomposition. Comput Methods Appl Mech Eng 374:113547

Le Dret H, Lucquin B (2016) Partial differential equations: modeling, analysis and numerical approximation, vol 168. Springer, Cham

Li W, Bazant M, Zhu J (2021) A physics-guided neural network framework for elastic plates: comparison of governing equations-based and energy-based approaches. Comput Methods Appl Mech Eng 383:113933

Lu L, Pestourie R, Yao W, Wang Z, Verdugo F, Johnson SG (2021) Physics-informed neural networks with hard constraints for inverse design. SIAM J Sci Comput 43(6):B1105–B1132

Nguyen-Thanh VM, Zhuang X, Rabczuk T (2020) A deep energy method for finite deformation hyperelasticity. Eur J Mech A 80:103874

Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707

Rukundo O, Cao H (2012) Nearest neighbor value interpolation. arXiv Preprint. arXiv:1211.1768

Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, Zhuang X, Rabczuk T (2020) 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 362:112790

Shin S, Shin D, Kang N (2023) Topology optimization via machine learning and deep learning: a review. J Comput Des Eng 10(4):1736–1766

Tancik M, Srinivasan P, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ramamoorthi R, Barron J, Ng R (2020) Fourier features let networks learn high frequency functions in low dimensional domains. Adv Neural Inf Process Syst 33:7537–7547

Wang S, Wang H, Perdikaris P (2021) Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Sci Adv 7(40):eabi8605

Woldseth RV, Aage N, Bærentzen JA, Sigmund O (2022) On the use of artificial neural networks in topology optimisation. Struct Multidisc Optim 65(10):294

Xiang Z, Peng W, Zhou W, Yao W (2022) Hybrid finite difference with the physics-informed neural network for solving PDE in complex geometries. arXiv Preprint. arXiv:2202.07926

Yu B (2018) The Deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. Commun Math Stat 6(1):1–12

Zehnder J, Li Y, Coros S, Thomaszewski B (2021) NTopo: mesh-free topology optimization using implicit neural representations. Adv Neural Inf Process Syst 34:10368–10381