



# Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design

Hao Deng<sup>1</sup> · Albert C. To<sup>1</sup>

Received: 12 December 2019 / Accepted: 29 April 2020 / Published online: 26 May 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

This paper proposed a new topology optimization method based on geometry deep learning. The density distribution in design domain is described by deep neural networks. Compared to traditional density-based method, using geometry deep learning method to describe the density distribution function can guarantee the smoothness of the boundary and effectively overcome the checkerboard phenomenon. The design variables can be reduced phenomenally based on deep learning representation method. The numerical results for three different kernels including the Gaussian, Tansig, and Tribas are compared. The structural complexity can be directly controlled through the architectures of the neural networks, and minimum length is also controllable for the Gaussian kernel. Several 2-D and 3-D numerical examples are demonstrated in detail to demonstrate the effectiveness of proposed method from minimum compliance to stress-constrained problems.

**Keywords** Topology optimization · Deep learning · Geometry complexity · Stress-constrained

## 1 Introduction

Lightweight designs are desirable in many industrial applications and structural optimization is an effective way to achieve this. Topology optimization is an important tool to obtain the optimal design of engineering structures. Because of its importance in engineering designs, this subject has drawn great attention by academia for more than twenty years, and remarkable progress has been made since the pioneering work of Bendsoe and Kikuchi [1]. In recent years, many topology optimization methods have been proposed, such as the solid isotropic material with penalization (SIMP), the level-set method (LSM) [2], and the moving morphable components (MMC) [3], etc. Because of the simple material distribution description, the SIMP method gained much popularity for structural design in engineering. The SIMP method is a pixel-based method, where the material layout is fully described by pixels. Because of the pixel-based description, numerical problems are often occurred such as checkboards, staggered boundaries, and

mesh dependency [4]. Although pixel-based material layout has already achieved remarkable progress in recent years, there still exist several challenges such as controlling the structural complexity and ensuring manufacturability of an optimal design [5]. To resolve the above problems, several new geometry representation methods have been proposed in recent years. To control design complexity in an explicit geometrical way, a moving morphable component (MMC) approach was proposed by Guo et al. [3]. In Guo's work, all components are described by level set function and allowed to move, overlap and merge freely, where XFEM analysis based on a fixed mesh is carried out to solve physical problems. Based on the MMC approach, those researchers [6–8] further extended MMC to treat 3-D problems and solve more complex physical problem such as stress constraint and multi-material problems. Recently, Tortorelli et al. [9] proposed a geometry projection method for the continuum-based topology optimization made of discrete elements. This method is in the context of density-based topology optimization, and hence standard finite element analysis (FEA) and nonlinear programming algorithm can be applied, where a differentiable mapping from discrete element to density field is realized in this paper. Furthermore, Zhang et al. [10] developed this method to solve stress constraint problem, where the optimal design is based on optimizing an assembly of discrete geometric components such as bars or

✉ Albert C. To  
albertto@pitt.edu

<sup>1</sup> Department of Mechanical Engineering and Materials Science, University of Pittsburgh, Pittsburgh, PA 15261, USA

plates. Lately, Watts and Tortorelli [11] extended the geometric projection method to 3-D and design unit cell for lattice materials based on inverse homogenization, where a negative Poisson's ratio lattice material is achieved. White et al. [12] proposed a novel method to represent the density field with a truncated Fourier representation, where the number of decision variables are reduced significantly. Recently, Gao et al. [13] proposed an effective method to apply density distribution function, in combination with isogeometric topology optimization, to describe the material layout, where the smoothness and continuity of the optimal design are demonstrated in details. The methods mentioned above can be classified as dimension reduction method. From the mathematical view, solving these problems is equivalent to finding an appropriate density field representation methodology to take the place of traditional pixel-based method.

In recent years, introducing the deep learning method to resolve physical problem such as physics-informed deep learning method is a hot and advanced topic [14–23]. For topology optimization, some new methods are proposed to apply deep learning in design as described in Ref. [24–26]. In computer graphics, the 3D computer vision and robotic communities have come up with multiple approaches to represent 3D geometry for rendering and reconstruction. Fidelity, efficiency, and compression capabilities are the three key factors to balance when choosing among different representation methods. Recently, deep learning for 3D geometric representations draws great attentions from academy [27, 28]. In general, data-driven 3D representation learning approaches can be classified into three categories: point-based, voxel-based and mesh-based methods. For point-based method, a point cloud is a lightweight 3D representation which can closely relate to the geometric raw data, and hence is a natural choose for 3D geometric representation. PointNet [29], for example, apply the max-pool operations to extract and represent geometry. Mesh-based learning methods [30] using parameterization algorithm to represent 3D surface through morphing 2D planes. However, this mesh-based method is often sensitive to input mesh quality. Voxels, using 3D grids to describe volumes, are the most natural extension into 3D domain; however, voxel-based approaches cannot preserve fine shape details and their normal are not smooth. Furthermore, using voxel approach to represent geometry can generally handle low resolutions ( $128^3$  or below), and memory requirements of this method increase cubically. Recently, a new geometry representation called the deep signed distance function (DeepSDF) [31] is proposed to describe modeling shapes as the zero iso-surface boundaries, and a deep feed-forward network is trained to represent SDFs. In this way, the CAD surface is implicitly represented by zero level-set, and Marching Cubes [32] is applied to generate geometry model through raycasting of the surface mesh. The purpose of the

present work is proposing the deep representation learning (DRL) method, which incorporates geometry deep learning into existing density-based topology optimization method. The density field is described by a deep feed-forward neural network to ensure the enough smoothness and continuity of material layout.

Recently, Zhou et al. [33] proposed a so-called generalized discrete cosine transform (DCT) compression-based density method to achieve an efficient topology optimization, which does not need any additional filter for optimization and design variables can be phenomenally reduced. As described by Zhou et al. [33], the number of design variables is positive correlative to the computation time. However, the computation time is not linearly dependent on the number of design variables, because the time cost of FEA calculation is not reduced in this method. From this perspective, there is not much differences in computational time if the FEA analysis is time-consuming in that reducing the design variables mainly decreases the time for optimization solver to update. Similar methods based on Fourier transformation is also reported by White et al. [12], and a new method called a material-field series-expansion method [34] is also proposed recently. A dual mesh method proposed by White et al. [35] uses Bernstein polynomials on a coarse mesh to reduce the number of design variables and provide length scale control and allow for AMR. In fact, some other methods such as using IGA [13] to represent density distribution shares some similarity with respect to above methods. The major advantages of these methods come from a) capability of avoiding the checkerboard patterns and mesh dependency b) enough smoothness at boundary and dimension reduction c) Adaptive mesh refinement causes problems for traditional SIMP since the number of design variables is not constant. This is an advantage of the method proposed in this paper, the design has been decoupled from the analysis mesh. d) Effectively reduce design variables. One of the major reasons to reduce the design variables is to build surrogate model for physical problem for topology optimization, which is reported in recent literature [36]. As described in this paper, an effective non-gradient method is proposed using material-field series expansion to represent density field. Because the number of design variables can be reduced significantly, it is possible to build a surrogate model for physical problem (the cost of building surrogate model is dependent on the number of variables). This material-field series expansion method combined with kriging-based optimization method successfully achieve a non-gradient method for topology optimization from linear problems to nonlinear problems as shown in Ref [36]. The method proposed in our paper is a new density field representation method compared to material-field series expansion method, which can be potentially combined with kriging-based optimization algorithm to achieve non-gradient optimization. This point will be investigated

in the future. Above methods are indeed using function to describe the material distribution instead of pixel. From this point, the core novelty of this paper is to propose a generalized function description method for topology optimization design. As described in previous paragraph, the deep neural networks have ability to approximate any complex functions. Thus, we proposed a deep learning-based method to achieve a generalized description of density distribution, which is a generalization formulation of above methods.

The paper is organized as follows. In Sect. 2, we describe the deep learning algorithm for geometry representation. Section 3 describes the topology optimization formulation based on deep geometry representation. In Sect. 4, several typical numerical cases are presented to demonstrate the effectiveness of proposed algorithm, followed by conclusions in Sect. 5.

## 2 Geometry description based on deep representation learning

In computer-aided design, B-rep (boundary representation) [37] is a general way to represent shapes. In this way, a geometry is represented as collection of connected surface elements, which formulate the boundary between solid and void. Compared to B-rep representation method, implicit surface modeling method [38, 39] describes the geometry by implicit function, and the level set of the function represents boundary surface, whereas B-rep method usually consists of piecewise surface patches. Geometry description based on implicit surfaces provide a straightforward way (metamorphosis [40]) to fillet and round surfaces, which is a powerful tool to join two geometry with sufficient continuous. Moreover, implicit geometry modeling method is simpler to determine whether a point is inside, outside, or on a surface. This facilitates the construction of complex geometry such as lattice or porous media. Another advantage of implicit surface is memory requirement is far less than B-rep [41, 42], because the geometry is described by spatial continuous functions  $f(x, y, z)$ , where  $x, y, z$  are spatial coordinates and the zero-level set of  $f(x, y, z)$  represents the isosurface of geometry. A Stanford bunny [43], which serves as a computer graphics 3D test model, is shown in Fig. 1. For density-based topology optimization method, a geometry model is generally described by voxels. Voxels, which non-parametrically described geometry with 3D grids of values, are used commonly in density-based topology optimization. The voxel representation method suffers from huge computing costs and memory requirement [44], while it is difficult to obtain high-fidelity shapes using the voxel model because the rendered normal is not smooth.

Recently, modern representation learning techniques have been developed to automatically extract a set of features that



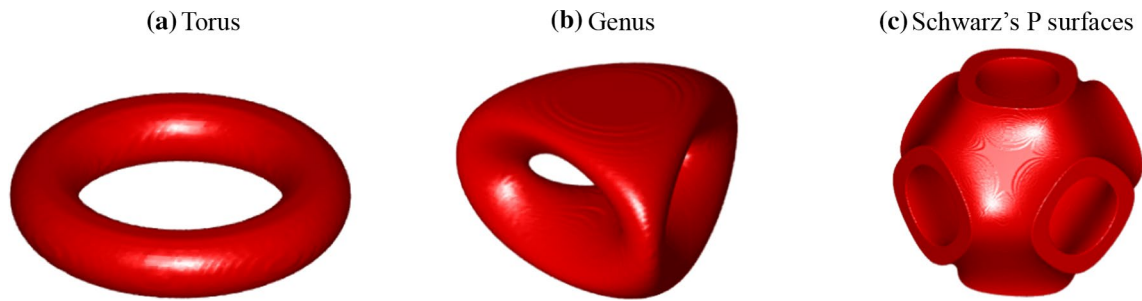
Fig. 1 Model for stanford bunny

compactly represent geometry without loss of fidelity. Several representation learning techniques are proposed such as Generative Adversial Networks [45], Auto-encoders [46], and Optimizing Latent Vectors [47]. In this paper, a new geometry representation method is employed to describe the design shape, while the feedforward neural networks are trained to represent the implicit surface. To demonstrate this idea in detail, a general implicit surface can be expressed as,

$$F(x, y, z) = 0 \quad (1)$$

The implicit surface is the set of spatial coordinates  $\{x, y, z\}$  that satisfy above equation, and the implicit surface is the level set of density field  $F(x, y, z)$ . Some typical implicit surfaces are shown as following (Fig. 2),

However, analytical expression for implicit geometry is limited and difficult to achieve free-form topology optimization. In this paper, a new implicit geometry representation method is proposed. Instead of applying analytical expression to describe a geometry, a deep feedforward neural network [49–51] is implemented here to substitute the implicit function  $F(x, y, z)$  in Eq. (1). Deep feedforward networks [49, 52, 53], also known as multilayer perceptions, are the foundation of most of the deep learning models such as convolutional neural networks (CNNs) [54]. The main goal of a feedforward network is to approximate a function. For example, a spatial function  $g = F(x, y, z)$  maps the 3D coordinate  $\{x, y, z\}$  to a value  $g$ . Similarly, a feedforward network defines a mapping function  $g_n = F_{networks}(x, y, z, \theta)$  from input coordinate to output  $g_n$ . Note that the parameters  $\theta$  need to be trained to achieve the best function approximation. In fact, deep networks can represent certain functions far more efficiently than shallow ones, and the fitting capability increase significantly with greater depth [55]. Assume that a Stanford bunny can be represented by a density field (voxel representation), which is described by an implicit function. To directly obtain the analytical implicit geometry expression for this Stanford bunny is difficult; however, deep neural networks can be employed to approximate the implicit



**Fig. 2** implicit surface **a** Torus  $(x^2 + y^2 + z^2 + R^2 - a^2)^2 - 4R^2(x^2 + y^2) = 0$ , **b** Genus  $2y(y^2 - 3x^2)(1 - z^2) + (x^2 + y^2)^2 - (9z^2 - 1)(1 - z^2) = 0$  and **c** Schwarz's P surfaces [48]  $(\cos(\pi x) + \cos(\pi y) + \cos(\pi z))^2 - t^2 = 0$

geometry function, which shares some similarity with Deep-SDF [56]. The objective here is to find a compact representation for the spatial density distribution of the Stanford bunny shown in Fig. 1. The Stanford bunny is represented by  $100 \times 100 \times 100$  voxels. The input of deep feedforward networks is spatial coordinates of a voxel, and the output is density value at the present coordinates. Thus, the number of training data is  $1 \times 10^6$ . The activation kernel is chosen as Tansig (Hyperbolic tangent sigmoid transfer function [57]) neuron. The optimization formulation for deep geometry representation can be written as,

$$\begin{cases} \text{Find : } \theta \\ \text{Min : } \sum_{i=1}^N F_{networks}(x_i, y_i, z_i, \theta) - D(x_i, y_i, z_i)_2 \end{cases} \quad (2)$$

where  $F_{networks}$  represent neural network, and the  $\theta$  is the parameters of network.  $D(x, y, z)$  represents density value of Stanford bunny at point  $(x, y, z)$  and operator  $\|\cdot\|_2$  denotes 2-norm.  $N$  is total number of voxels.

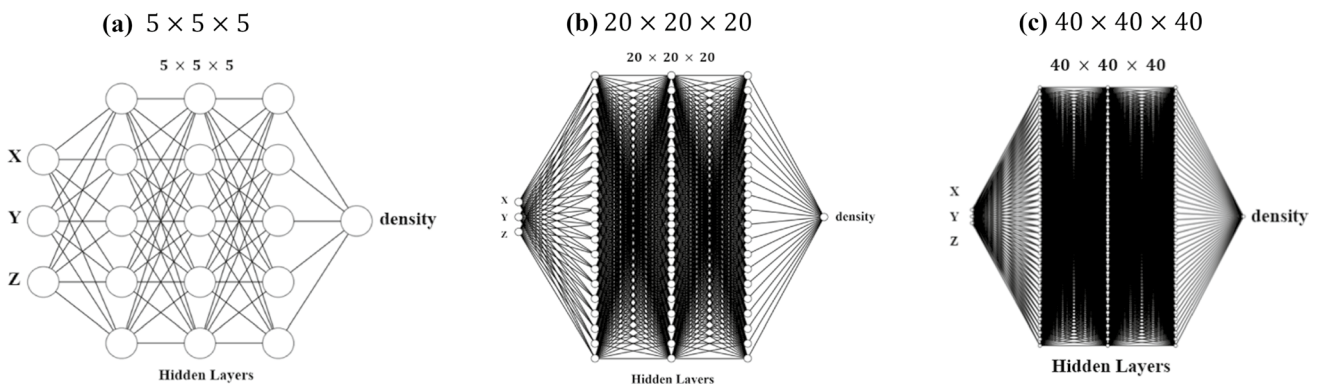
Figure 3 illustrates three feedforward networks with three hidden layers are chosen with different neurons in each layer for comparison. The Levenberg–Marquardt backpropagation

algorithm [58] is implemented to train the networks based on the objective function Eq. (2). The training results are presented in Fig. 4. Obviously, the network with  $5 \times 5 \times 5$  hidden layers is only able to represent coarse configuration, and lots of geometry details are missing. However, the network with  $40 \times 40 \times 40$  hidden layers is able to represent geometry with a high fidelity. Mean squared error (MSE) is applied to measure the error between the objective density field (Stanford bunny) with respect to density field represented by trained neural networks.

### 3 Topology optimization formulation based on deep representation learning

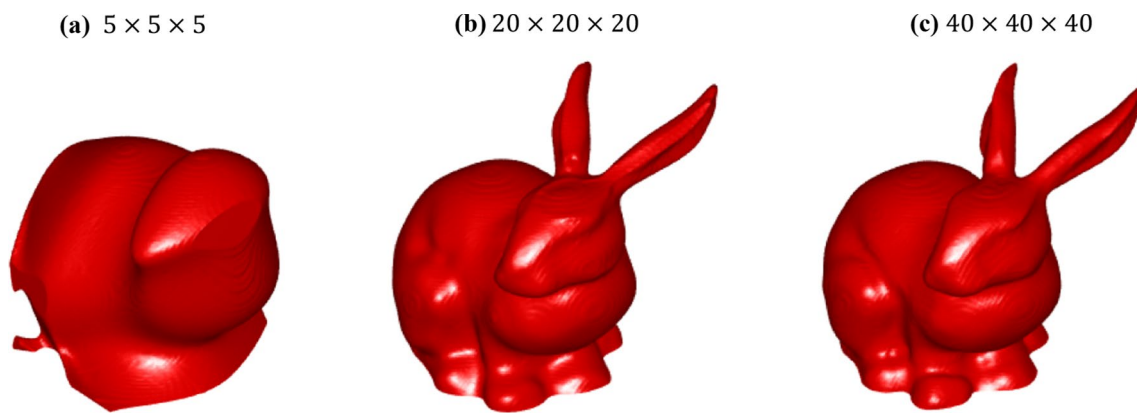
#### 3.1 Density field described by deep feedforward networks

For density-based method, the material distribution is transformed to spatial arrangement of finite elements. The finite element method (FEM) formulation is formulated by assembling the discrete elements with different density. For the well-established solid isotropic material with penalization



**Fig. 3** The architecture of feedforward networks **a**  $5 \times 5 \times 5$  (freedom : 81), **b**  $20 \times 20 \times 20$  (freedom : 921), **c**  $40 \times 40 \times 40$  (freedom : 3441)





**Fig. 4** Stanford bunny represented by deep feedforward neural networks **a**  $5 \times 5 \times 5$  (*freedom* : 81; *MSE* : 0.2075), **b**  $20 \times 20 \times 20$  (*freedom* : 921, *MSE* : 0.007952), **c**  $40 \times 40 \times 40$  (*freedom* : 3441, *MSE* : 0.001617)

(SIMP) approach, the spatial arrangement of density is represented by mesh, which results in optimized layout with staggered boundary (i.e. Lego effect). Thus, a substantial effort in post-processing is needed to generate a smooth CAD model, which may compromise geometric precision along the boundary. Since mesh are utilized to represent the structural topology, the number of design variables is usually quite large for three-dimensional design, and many mature optimization techniques are not applicable for large-scale problem [59]. To resolve the above issues, a new density representation method using deep feedforward network is described in this section. As described in Sect. 2, a complex geometry can be represented by a deep feedforward network with high fidelity, and smoothness of surface can be guaranteed. Thus, it is a natural choice to apply deep feedforward network to represent the density field in the design domain. A requirement should be satisfied to ensure a well-justified density field, i.e., the bounds of element densities are within [0, 1]. Like the formulation in Sect. 2, a density function in design domain is described by a deep feedforward network, and the input for the network are all the point coordinates. The output is density value at a given point. To ensure the output density is in the bounds [0, 1], a mapping function  $\mathcal{M}$  is applied as follows,

$$\mathcal{M} = \frac{(\tanh(\beta x) + 1)}{2} (\beta = 0.5) \tag{3}$$

Note that the parameter  $\beta$  is chosen as 0.5 in this paper. An example is presented here to demonstrate the functionality of mapping function  $\mathcal{M}$ . Consider a two-dimensional problem, the density field  $\phi(x, y)$  is described by a  $20 \times 20 \times 20$  feedforward network with Tansig activation function. Thus, the mathematical formulation of density field can be expressed as:

$$\phi(x, y) = \mathcal{M}(\mathbb{N}(x, y, \theta))(2Dproblem) \tag{4}$$

$$\phi(x, y, z) = \mathcal{M}(\mathbb{N}(x, y, z, \theta))(3Dproblem) \tag{5}$$

where  $\mathbb{N}$  denotes feedforward networks, and the  $\theta$  is parameter. The architecture of deep layered network composes many hidden layers. Denoting the output of hidden layers by  $\mathbf{h}^{(l)}(x)$ , a network with L hidden layers can be expressed as,

$$\mathbb{N}(x, y, z, \theta) = \mathbb{N}(\mathbf{a}^{(L+1)}(\mathbf{h}^{(L)}(\mathbf{a}^{(L)}(\dots \mathbf{h}^{(1)}(\mathbf{a}^{(1)}(x, y, z)))))) \tag{6}$$

where  $\mathbf{a}^{(l)}(x)$  is a linear operation, expressed as,

$$\mathbf{a}^{(l)}(x) = \mathbf{W}^{(l)}\mathbf{x} + \mathbf{b}^{(l)} \tag{7}$$

where  $\mathbf{W}^{(l)}$  is weight matrix and  $\mathbf{b}^{(l)}$  is bias vector for the  $l$ th layer. The weight matrix  $\mathbf{W}^{(l)}(l = 1, 2, \dots, L)$  and bias  $\mathbf{b}^{(l)}(l = 1, 2, \dots, L)$  can be combined into a single parameter  $\theta$ .  $\mathbf{h}^{(l)}(l = 1, 2, \dots, L)$  are hidden-layer activation functions (kernel functions).

### 3.2 Minimum compliance

In this section, the deep representation learning (DRL) is adopted to develop the topology optimization formulation of compliance minimization [60]. The density field is represented by a deep neural network in the design domain. Hence, the TO will iteratively optimize the density field through updating the parameters of the network in the design domain until the material layout has the best stiffness performance. Here, the weights of the feedforward network are defined as the design variables for evolving the density field in the design domain during the optimization. Thus, the optimization problem can be expressed as:

$$\begin{cases} \text{Find : } \theta \\ \text{Min : } C(u, \Phi) = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}(u)^T \mathbf{D}(\Phi(\theta)) \boldsymbol{\varepsilon}(u) d\Omega \\ \text{s.t : } \left\{ \begin{array}{l} \frac{1}{|\Omega|} \int_{\Omega} \Phi(\theta) d\Omega - V_{\text{prescribe}} \leq 0 \end{array} \right. \end{cases} \quad (8)$$

where the  $\theta$  are the parameters of the deep feedforward network, and  $C$  is the objective function defined by the structural compliance.  $\Phi$  is the density distribution in the design domain  $\Omega$ , and  $V_{\text{prescribe}}$  is the prescribed volume fraction. In the finite element model,  $u$  is the unknown displacement field,  $\boldsymbol{\varepsilon}$  is the strain, and  $\mathbf{D}$  is the elastic tensor matrix.

### 3.3 Minimum compliance with stress constraint

For the minimum compliance with stress constraint problem, the von Mises stress is always used for local stress measurement and as stress constraint in the optimization. However, constraining the local stress is numerically expensive in practice. Thus, a p-norm approach is implemented here to approximate the local stress constraint. In recent years, several modified methods have been proposed to accurately control the local stress [10, 61–68]. For simplicity, we apply a well-developed method to constrain the local von Mises stress as described in Ref. [69]. In this method, the p-norm measure  $\sigma_{PN}$  is adopted to formulate the constraint. Thus, the problem in Sect. 3.2 can be reformulated as:

$$\begin{cases} \text{Find : } \theta \\ \text{Min : } C(u, \Phi) = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}(u)^T \mathbf{D}(\Phi(\theta)) \boldsymbol{\varepsilon}(u) d\Omega \\ \text{s.t : } \left\{ \begin{array}{l} \frac{1}{|\Omega|} \int_{\Omega} \Phi(\theta) d\Omega - V_{\text{prescribe}} \leq 0 \\ \sigma_{PN} = \left( \sum_{e=1}^N (v_e \sigma_e^{vM})^p \right)^{\frac{1}{p}} \leq \overline{\sigma_{PN}} \left( \sum_{e=1}^N (v_e \sigma_e^{vM})^p \right)^{\frac{1}{p}} - \overline{\sigma_{PN}} < 0 \end{array} \right. \end{cases} \quad (9)$$

where  $p$  is the p-norm parameter,  $\sigma_e$  is element von Mises stress,  $\sigma_{PN}$  is p-norm measure, and  $\overline{\sigma_{PN}}$  is the global stress limit.  $v_e$  is element  $e$  solid volume. A good choice for  $p$  can make the algorithm perform well and provide an adequate approximation of the maximum stress value. In this paper,  $p = 10$  is applied in all stress-constrained numerical examples.

### 3.4 Design sensitivity analysis

For gradient-based optimization, the sensitivity analysis of the objective with respect to the design variables, i.e., weights of the feedforward network, are needed. To derive the sensitivity of the objective function, the chain rule will be employed. The adjoint method [70] can be used to obtain the sensitivity with respect to the density field  $\Phi$ :

$$\frac{\partial C}{\partial \Phi} = \lambda^T \frac{\partial \mathbf{K}}{\partial \Phi} \mathbf{u} \quad (10)$$

where  $\lambda$  is the adjoint vector computed from the adjoint equation  $\mathbf{K}\lambda = -\mathbf{f}$ , and  $\mathbf{K}$  is the assembled stiffness matrix, see Ref. [60]. Based on the chain rule, the sensitivity of objective  $C$  with respect to design variables  $w$  can be expressed as:

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial \Phi} \cdot \frac{\partial \Phi}{\partial w} \quad (11)$$

where the density field  $\Phi$  can be expressed as  $\mathcal{M}(\mathbb{N})$ . The sensitivity of  $\mathcal{M}(\mathbb{N})$  with respect to the network weights  $w$  can be readily obtained using the algorithmic differentiation (AD) technique [71, 72] implemented in the open-source software CasADi [73]. For sensitivity analysis of the p-norm stress, similar derivation can be achieved based on chain rule as follows:

$$\frac{\partial \sigma_{PN}}{\partial w} = \frac{\partial \sigma_{PN}}{\partial \Phi} \cdot \frac{\partial \Phi}{\partial w} \quad (12)$$

where the analytical sensitivity derivation based on the adjoint method of  $\frac{\partial \sigma_{PN}}{\partial \Phi}$  can be found in Ref. [74].

### 3.5 The relationship between geometry complexity with respect to architecture of neural networks

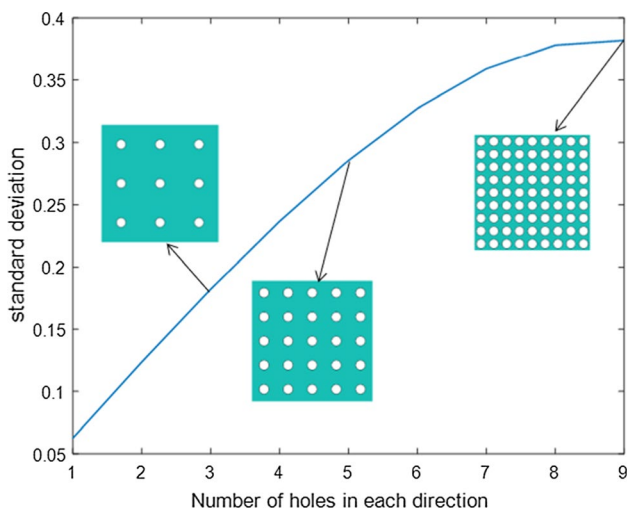
The architecture of neural networks is close related to the fitting ability, and how to design a deep neural network to satisfy a certain requirement is a hot topic in recent years. To investigate the relationship between the geometry complexity with respect to architecture of neural networks, several numerical experiments are conducted in this section. To simplify the problem, the geometry complexity is measured using standard deviation of density field as follows,

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\phi_i - \bar{\phi})^2} \quad (13)$$

where  $N$  is the total number of density points, and  $\bar{\phi}$  is mean value of density field, which can be expressed as,

$$\bar{\phi} = \frac{\sum_{i=1}^N \phi_i}{N} \quad (14)$$

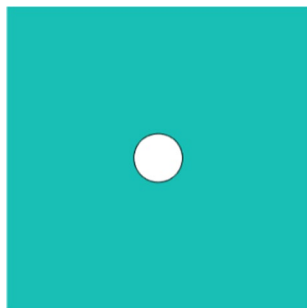
The relationship between the number of holes with respect to standard deviation can be found in Fig. 5. For number of holes less than  $9 \times 9$ , the density standard deviation increases with increasing number of holes, which means that the geometry complexity increases with more holes. The mean squared error (MSE) loss can be defined as,



**Fig. 5** The relationship between standard deviation with respect to number of holes in each direction

$$MSE = \frac{1}{N} \sum_{i=1}^N |\phi_i - \hat{\phi}_i|^2 \tag{15}$$

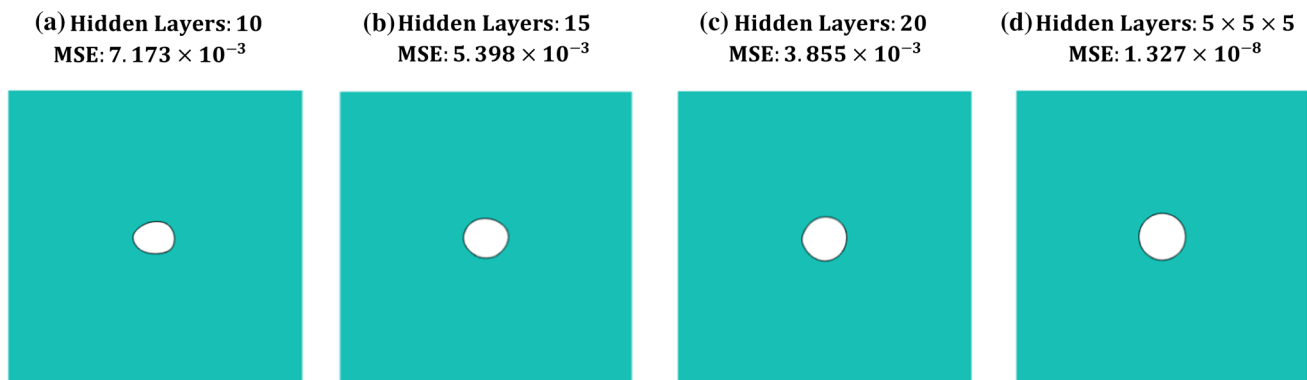
where the  $\hat{\phi}_i$  denotes the target density value, and  $\phi_i$  is density value computed from neural networks. *MSE* defines the



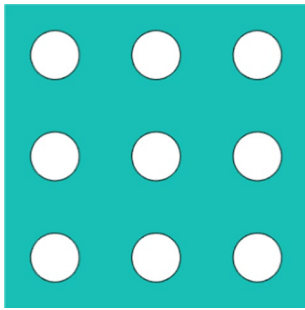
**Fig. 6** One hole in each direction

error between target density field and density field obtained from networks.

To verify that the geometry fitting ability is increasing with larger neural networks, several numerical experiments are conducted here. We apply different neural networks architecture to approximate objective geometry with different complexity. Figure 6 shows a square with one hole, which works as target density field. Four different neural networks are implemented to fitting this simple geometry. The fitting results (level set function,  $\phi = 0.5$ ) are shown in Fig. 7. Obviously, for one hidden layer, the MSE value decreases with increasing the number of neurons. Compared with shallow neural networks (one hidden layer), the networks with three hidden layers show better fitting ability with  $MSE = 1.327 \times 10^{-8}$ . For squares with three or five holes (Figs. 8 and 10), the fitting results are plotted in Figs. 9 and 11. The numerical results show that networks with more hidden layers and neurons have more powerful geometry fitting ability with lower MSE values. Another finding is that the networks can capture the major geometry feature once the MSE value is less than  $1 \times 10^{-2}$  from numerical experiments. For example, comparing Fig. 11d, g, networks with one hidden layer (15 neurons) cannot capture the major geometry features with  $MSE = 4.727 \times 10^{-2}$ , while networks with three hidden layers  $MSE = 9.638 \times 10^{-3}$  is able to capture major geometry features with some small defects at the boundary. The same conclusion can be drawn from Fig. 12. As shown in Fig. 12, we compare the fitting ability of two different networks. The horizontal axis represents number of holes, and the vertical axis denotes mean squared error (MSE). Apparently, the larger networks with more neurons in each layer have better fitting ability with lower MSE values. For example, if we use  $1 \times 10^{-2}$  as MSE threshold to work as a standard to determine whether fitting results is sufficient to approximate the target geometry, it is evident to observe that networks with 20 neurons in each layer can fit up to the square with  $9 \times 9$  holes. However, the smaller networks with 10 neurons has lower fitting ability, which



**Fig. 7** Geometric fitting results with different networks



**Fig. 8** Three holes in each direction

can only fit up to the square with  $4 \times 4$  holes. As shown in Fig. 5, standard deviation of density field is an approach to measure the geometry complexity. To further examine the geometry complexity with respect to networks architecture, the relationship between standard deviation and networks (three hidden layers) is plotted in Fig. 13. If we choose the  $MSE = 1 \times 10^{-2}$  as the threshold, the max standard deviation value for the networks with 5 neurons in each layer is 0.16, while max standard deviation value can reach 0.38 for the networks with 20 neurons. Therefore, standard deviation can work as a complexity measurement when designing networks. Meanwhile, the Fig. 13 can also work as a guidance when choosing the size of the neural networks to achieve

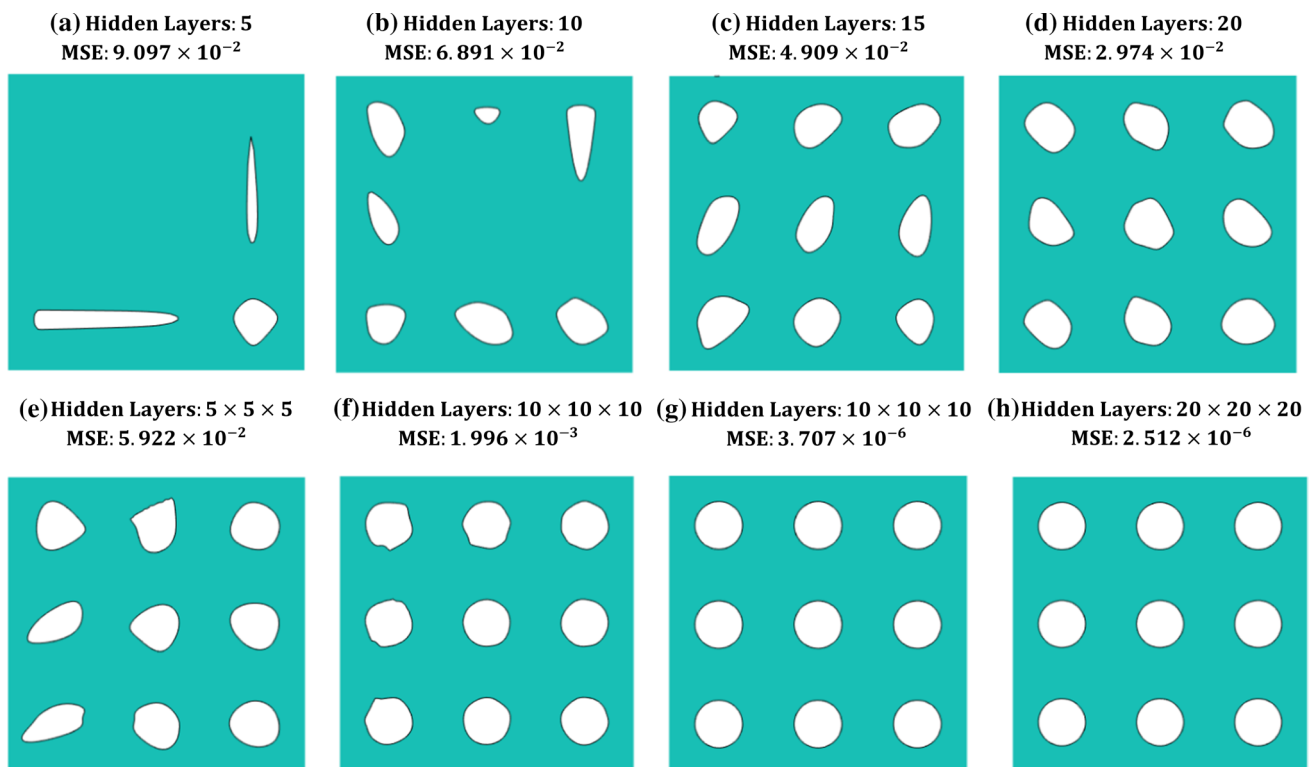
a certain geometry complexity. More works will be done to incorporate the standard deviation of density field as a complexity constraint in optimization in the future.

## 4 Numerical examples

In this section, several 2D and 3D numerical examples are demonstrated in detail to present the effectiveness of the proposed method. The classic MBB beam is first investigated to illustrate the benefits of the proposed DRL method. The box constraints are chosen as  $[-10, 10]$  for both weights and biases during optimization.

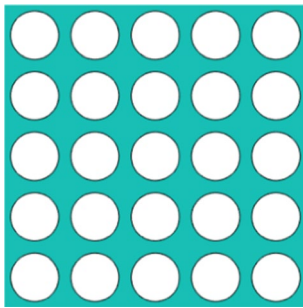
### 4.1 Compliance optimization for MBB design

The MBB-beam [75] is a popular test and benchmark problem in topology optimization. The symmetry is used for design, and the right half of the beam is modelled. The design of the MBB beam with the loading and boundary conditions is illustrated in Fig. 14. The design domain is uniformly meshed by  $300 \times 100$  elements. The prescribed volume fraction is set as 30%. The elastic constants are chosen as follows: elastic modulus  $E = 1$  and Poisson's ratio  $\mu = 0.3$ . The initial weights of the network are computed to generate uniformed density distribution in design domain.



**Fig. 9** Geometric fitting results with different networks ( $3 \times 3$  holes)

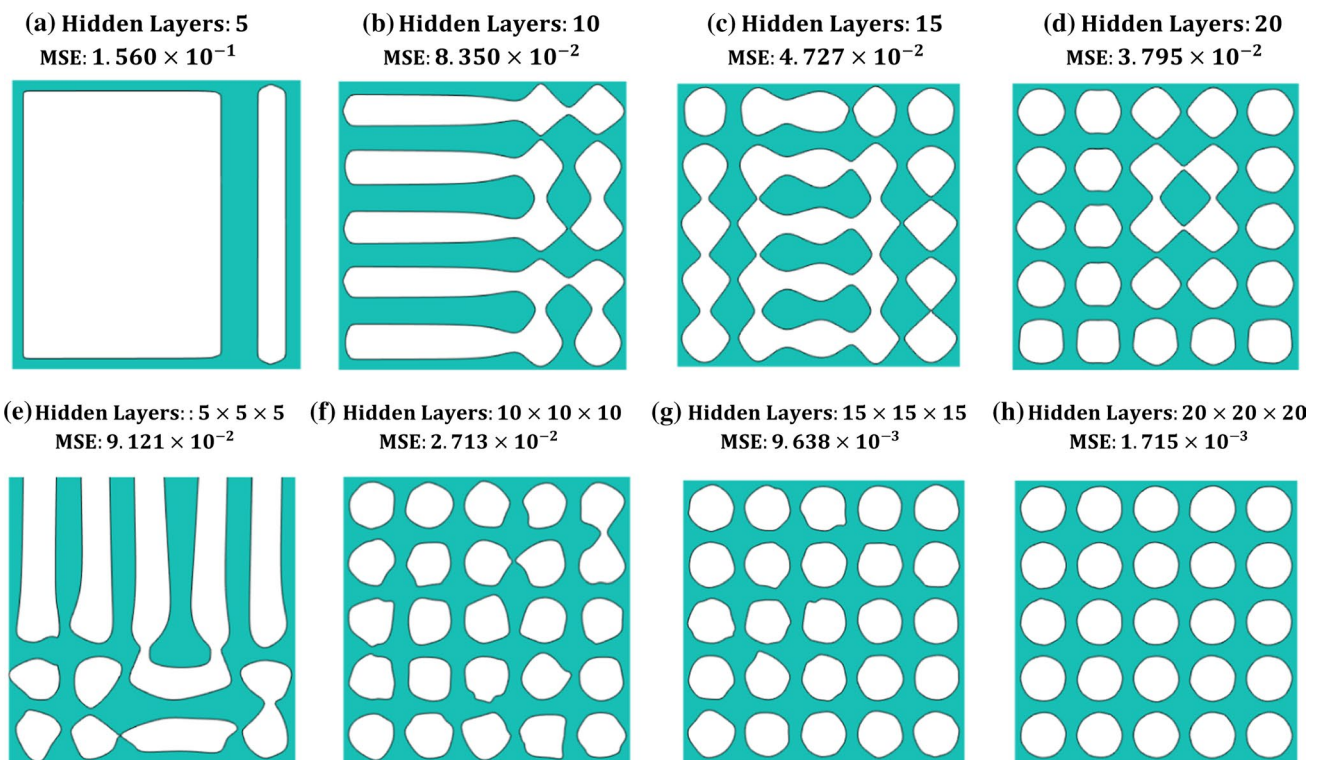




**Fig. 10** Five holes in each direction

For comparison, different network architectures are generated as shown in Fig. 15. The activation function is chosen as Tansig (Hyperbolic tangent sigmoid), and networks are fully connected. For the 2D problem, the inputs are spatial coordinates  $\{x, y\}$ , and output is density field. For networks with  $20 \times 20 \times 20$  hidden layers, the evolution of density field is presented in Fig. 20, and optimized design is plotted in Fig. 21. For shallow neural networks with only one layer, the evolution of density field and optimal design is plotted in Figs. 16 and 17. For  $20 \times 20$  hidden layers, the evolution history and optimal design are demonstrated in Figs. 18 and 19. It can readily be found that the different architectures result in different optimal topologies, and

shallow network can generate simple optimal layout with less geometric complexity. As shown in the density evolutionary progress, the density field is smoother and less small sharp features are found using shallow neural networks. This can be easily explained in that networks with more hidden layers present better fitting ability, which leads to more complex geometric topology. The optimal compliance for different designs is 454.75 (Hidden Layers: 20), 391.82 (Hidden Layers:  $20 \times 20$ ), and 336.45 (Hidden Layers:  $20 \times 20 \times 20$ ) respectively. The number of design variables for different architectures are 81, 501 and 921. Compared with voxel-based density method, the design variables reduce significantly. At present, the number of neurons is chosen based on experience and fully connected networks are used in density representation. However, the regularization method [76] of neural networks may be implemented to prune the topology structure. The regularization method is a technique that makes modification to the connectivity of networks such that the model generalizes better and reduces overfitting. In regularization, the coefficients of weights are penalized through modifying the objective function so that a sparse optimal result of weights is obtained. In such manner, some neurons in the network will be dropped so that the network has better compact representation. In this paper, we will not discuss the regularization of neural networks, and more research will be devoted to regularization in the future.



**Fig. 11** Geometric fitting results with different networks (5 × 5 holes)

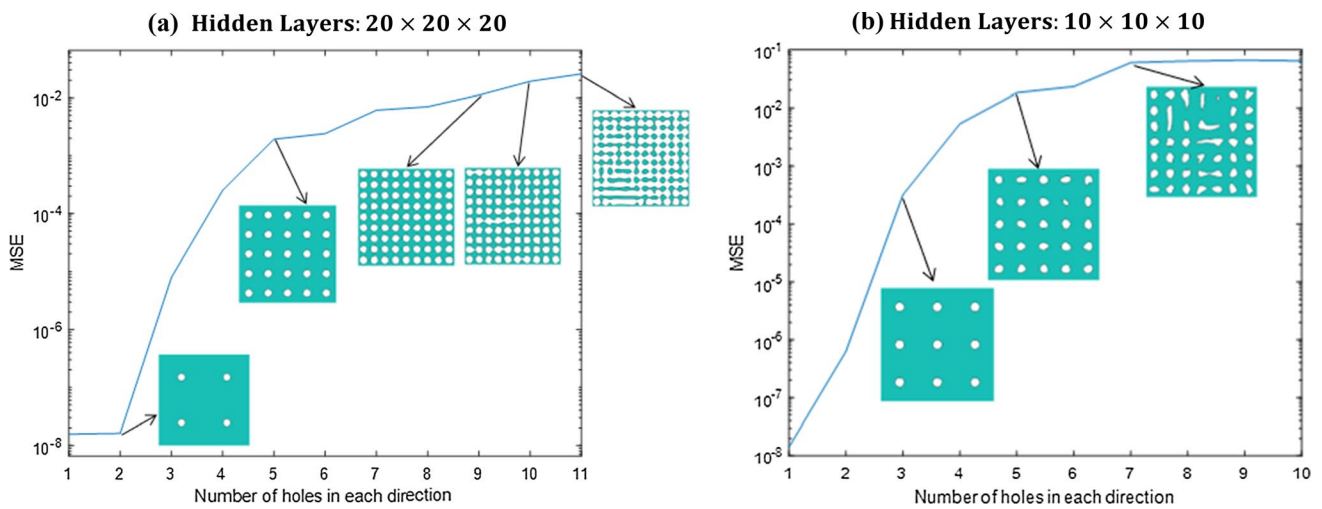


Fig. 12 Geometric fitting results with different networks (5 × 5 holes)

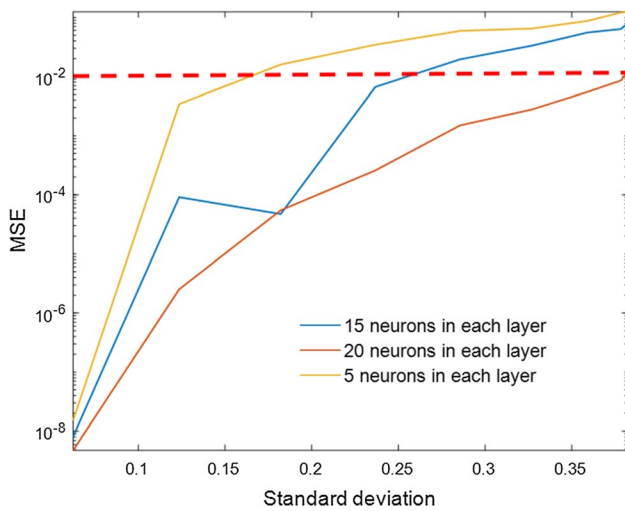


Fig. 13 Geometric fitting ability (Three Hidden Layers)

In this part, different neuron activation functions are implemented to examine the effect of activation function on the optimal design. Three activation functions implemented in this paper are Tansig (Hyperbolic tangent sigmoid transfer function), Gaussian [77], and Tribas (Triangular basis

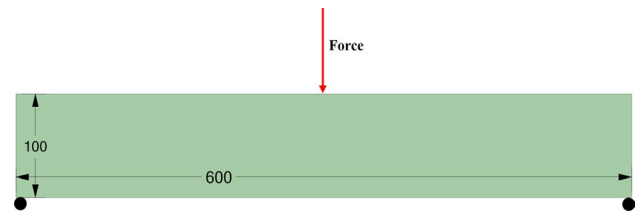


Fig. 14 MBB beam example

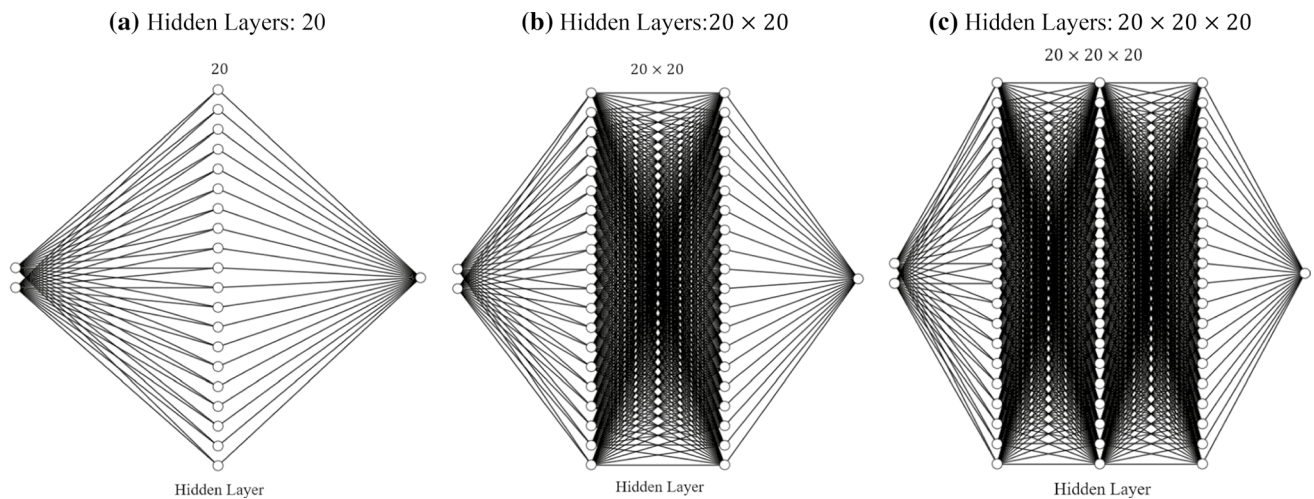
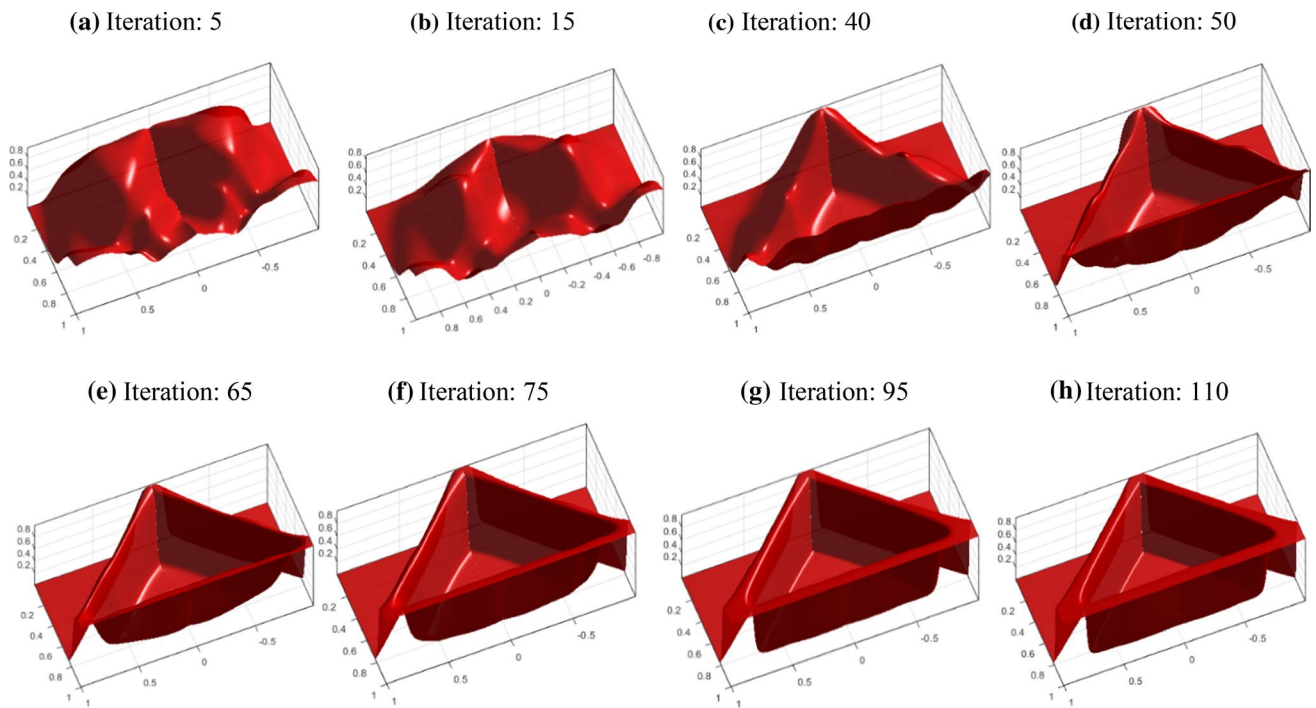


Fig. 15 Architectures of feedforward networks

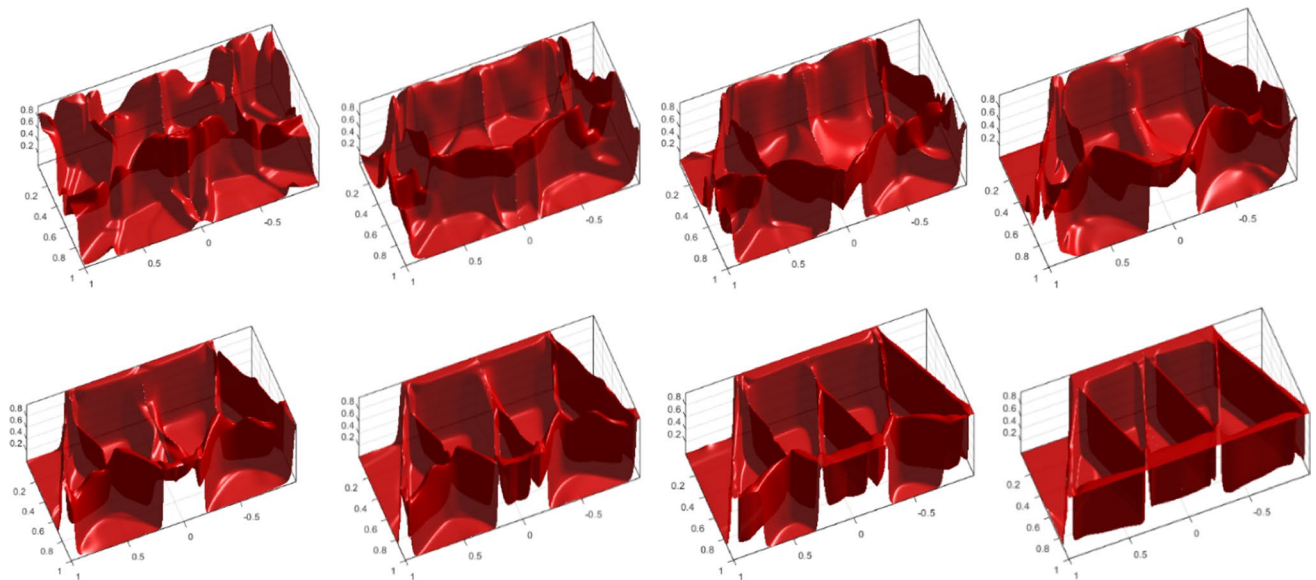


**Fig. 16** Evolution of density field (hidden layers structure: 20)



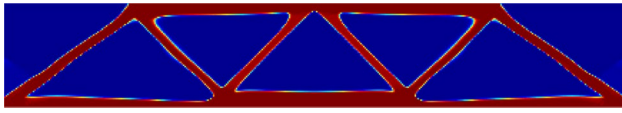
**Fig. 17** Optimal design (hidden layers structure: 20 compliance: 454.75)

function [78]). The mathematical properties of the three typical activation functions are plotted in Fig. 22. Note that the Tribas function is made of piece-wise linear function so that the spatial density distribution is piece-wise smooth as shown in Fig. 23. The optimal design obtained using Tribas is simpler geometrically compared to the Tansig kernel, the



**Fig. 18** Evolution of density field (hidden layer structure: 20 × 20)





**Fig. 19** Optimal design (hidden layer structure:  $20 \times 20$  compliance: 391.82)

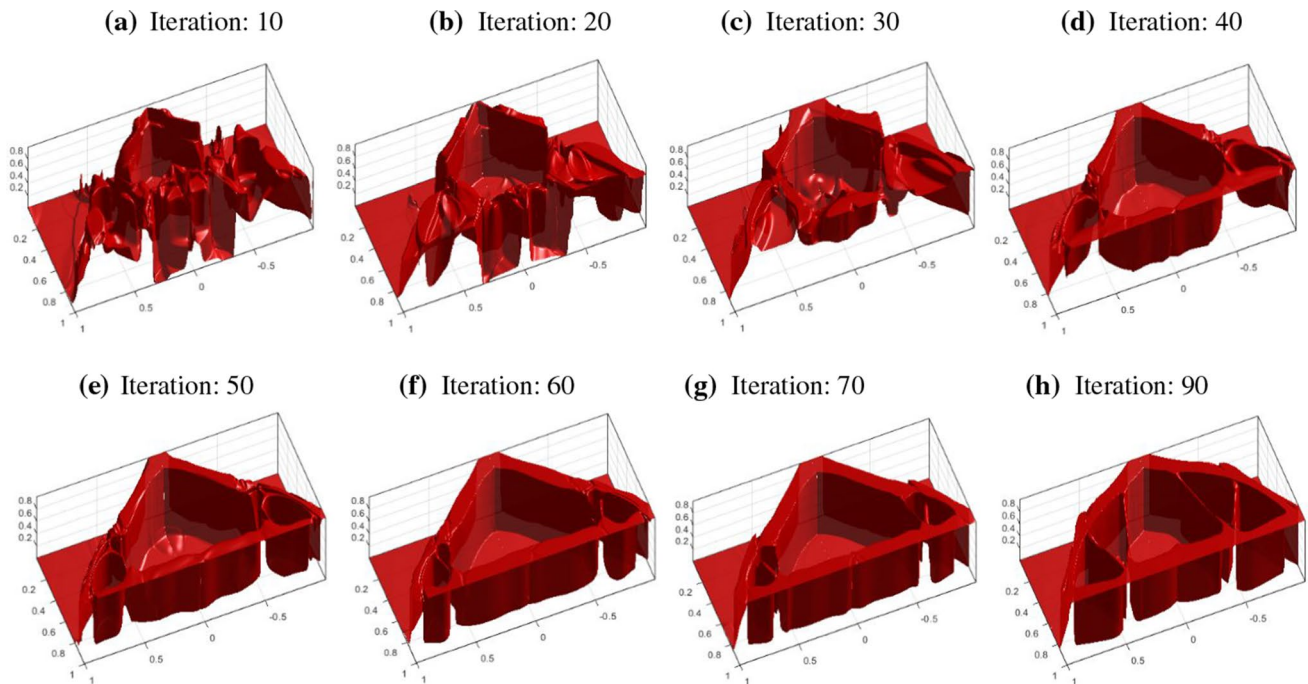
main reason lies in that the nonlinearity of Tansig is higher than Tribas (Fig. 24).

The Gaussian kernel, which is widely used in probability theory, shows excellent fitting capacity for highly nonlinear problems. The graph of Gaussian kernel is a characteristic symmetric “bell curve” shape, and the width of the “bell” is controlled by a parameter called the standard deviation [79]. The Gaussian kernel is continuous and infinitely differentiable, which is a significant difference from Tribas. Using the Gaussian kernel, the minimum feature of optimal design can be controlled through kernel width as shown in Fig. 25. In the Fig. 25, four different values of kernel width are chosen to generate four optimized designs. The minimum length of optimal design increases after increasing kernel width from 0.25 to 2. The evolution history for four different designs are plotted in Figs. 26, 27, 28 and 29. Evidently, the networks with smaller kernel width has more detailed feature as shown in Fig. 29. However, for large kernel width, the detailed feature or noise cannot be found in optimization as shown in Fig. 26. This can be explained based on image processing theory. In image processing, an image can be

blurred by a Gaussian function, which is known as Gaussian smoothing. The Gaussian function can be applied to reduce image noise and reduce detail. Mathematically, a Gaussian smoothing is a low pass filter, which has the effect of reducing the high-frequency components of function. This can be proven based on Weierstrass transform [80], and the kernel width can directly control property of low pass filter. Thus, the geometry details can be effectively controlled through kernel width. Strict mathematical proven will be done in the future to verify our numerical results. In fact, several effective minimum feature control methods are proposed in recently years based on conventional density-based method [5, 81–93]. Compared to these methods, our method provides an alternative way to control the minimum feature based on deep representation learning.

#### 4.2 Stress constrained optimization for two-dimensional L-bracket design

To further verify the effectiveness of the proposed method, the compliance minimization with stress constraint problem is considered in this section. The L-bracket is modeled by a  $100 \times 100$  finite element mesh with  $50 \times 50$  section removed as shown in Fig. 30. The boundary condition and force are demonstrated in Fig. 30. A vertical load  $F = 4$  is applied uniformly on four nodes, and element size is unity in this numerical example. The elastic constants are chosen as follows: elastic modulus  $E = 1$  and Poisson’s ratio  $\mu = 0.3$ . The p-norm value for this numerical



**Fig. 20** Evolution of density field (hidden layers:  $20 \times 20 \times 20$ )



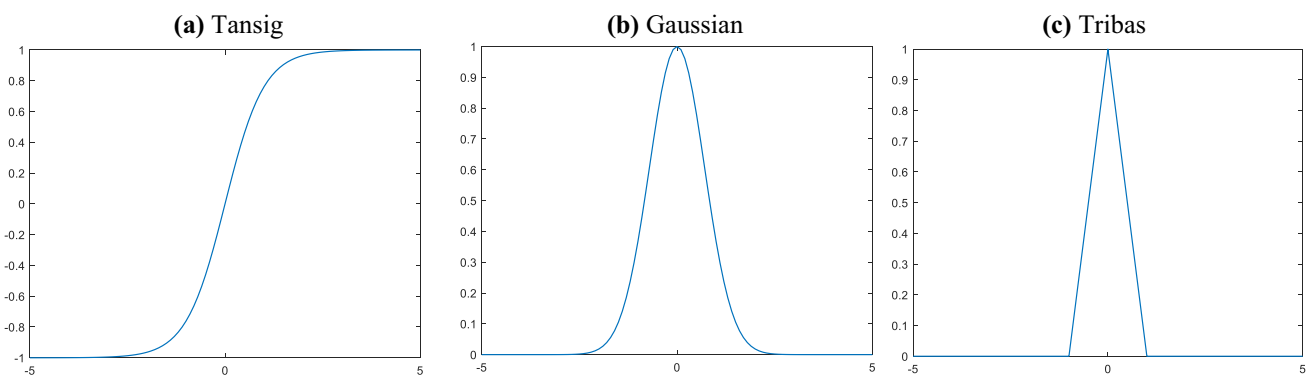
**Fig. 21** Optimal design (hidden layers structure:  $20 \times 20 \times 20$  compliance: 336.45)



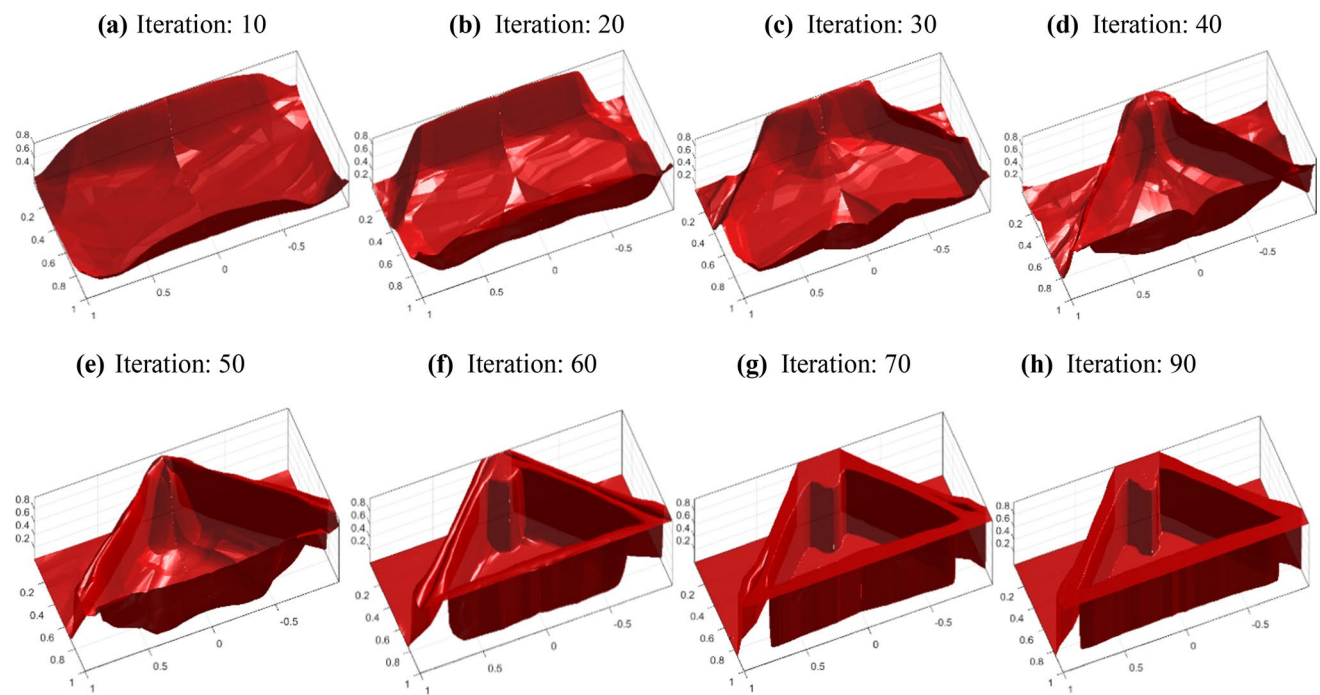
**Fig. 24** Optimal design (hidden layers:  $20 \times 20 \times 20$ , Tribas, Compliance: 445.77)

example is chosen as  $p = 10$ . The volume fraction is chosen as 0.3, and stress constraint (SC) in the p-norm is set to  $\sigma_{pnorm} < 2(SC : \sigma_{pnorm} - 2 < 0)$ . The neural network with three hidden layers of size  $20 \times 20 \times 20$  is implemented to represent the density distribution, and the Gaussian kernel is chosen as the activation function. Considering that the stress constraint optimization is highly nonlinear, a small moving

limit of 0.005 in the MMA algorithm [94] is employed in the optimization. At the beginning of the optimization, the stress concentration occurs at the sharp corner, and the sensitivity at this region is negative so that the material in this area tends to be removed. The final optimal result is plotted in Fig. 31. Note that round corners are generated to

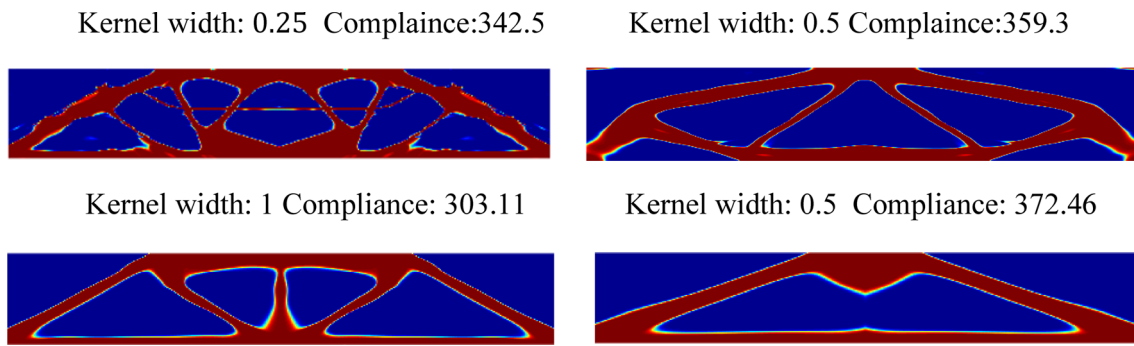


**Fig. 22** Graph of activation function **a** Tansig, **b** Gaussian, **c** Tribas

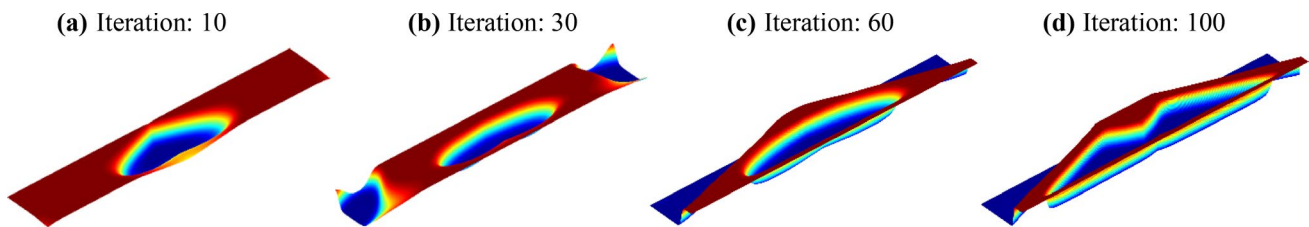


**Fig. 23** Evolution of density field (Kernel: Tribas, hidden layers:  $20 \times 20 \times 20$ )

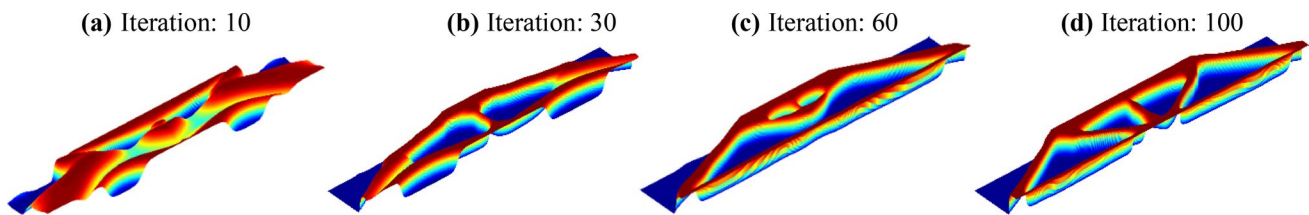




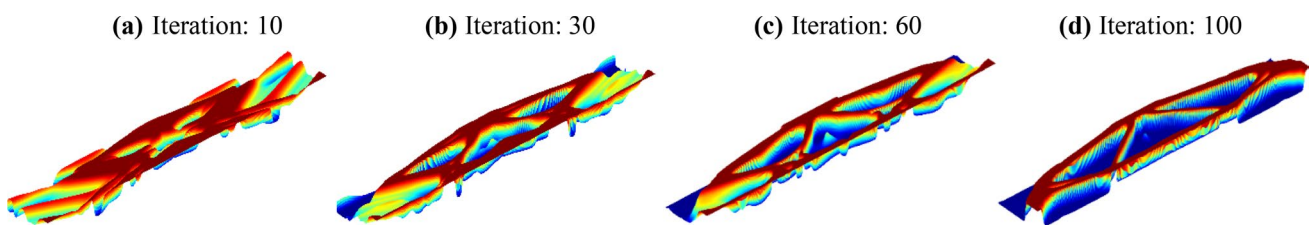
**Fig. 25** Optimal design with different kernel width (hidden layers:  $20 \times 20 \times 20$ , Gaussian kernel)



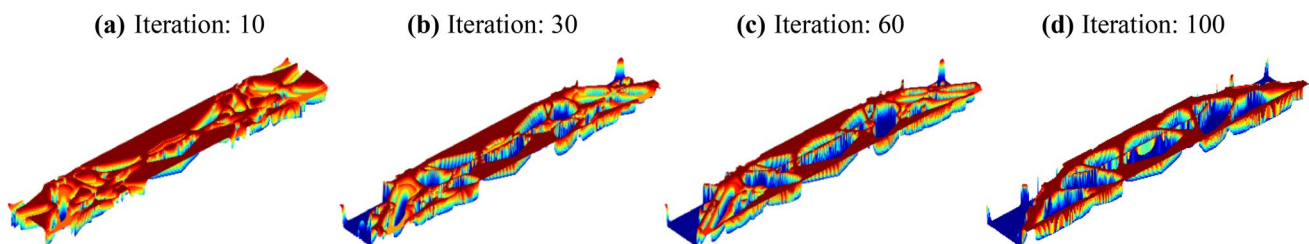
**Fig. 26** Evolution history with kernel width 2 (hidden layers:  $20 \times 20 \times 20$ , Gaussian kernel)



**Fig. 27** Evolution history with kernel width 1 (hidden layers:  $20 \times 20 \times 20$ , Gaussian kernel)



**Fig. 28** Evolution history with kernel width 0.5 (hidden layers:  $20 \times 20 \times 20$ , Gaussian kernel)



**Fig. 29** Evolution history with kernel width 0.25 (hidden layers:  $20 \times 20 \times 20$ , Gaussian kernel)

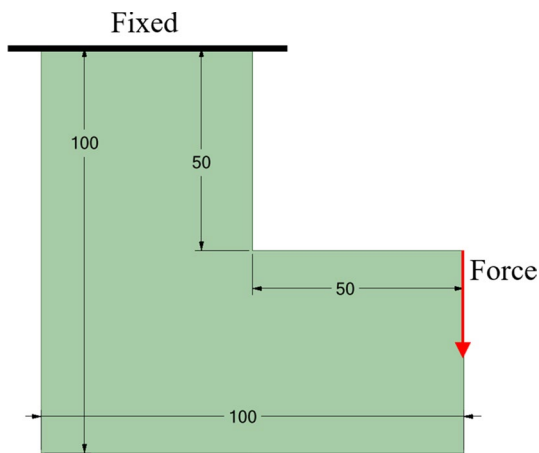


Fig. 30 2D L-bracket example

reduce stress concentration, and the optimized material layout boundary becomes smooth. The evolutionary history of density distribution is shown in Fig. 32. The stress contour and distribution for the final optimal design is presented in Fig. 33. The stress distribution of optimal design is uniform and smooth, and the maximum stress are in the region near

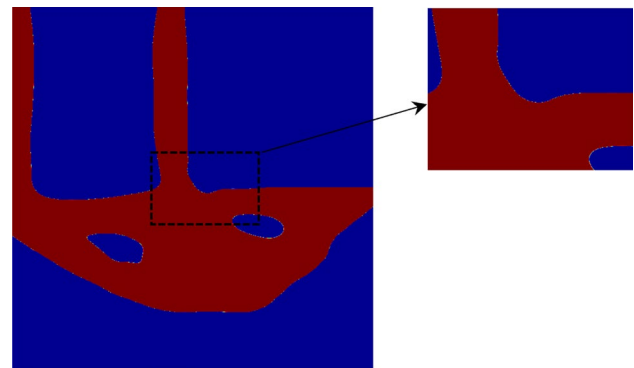


Fig. 31 Optimal design for stress constrained optimization

loading points as plotted in Fig. 33. The convergence history is shown in Fig. 34. Note that after optimization, the stress constraint is satisfied, which the compliance decreases significantly to around 1/3 of initial design. Because of the local stress singularity, the local oscillation of convergence curve can be observed in Fig. 34.

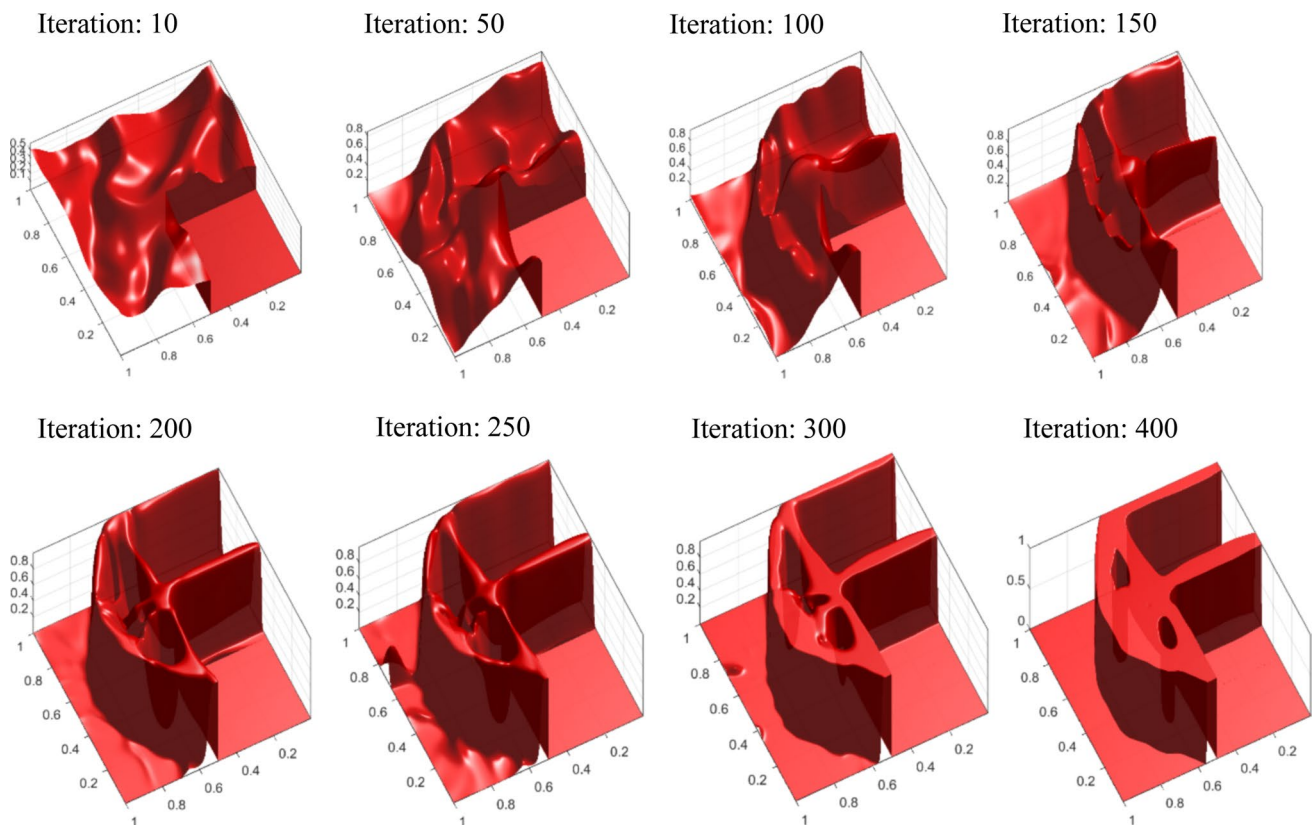
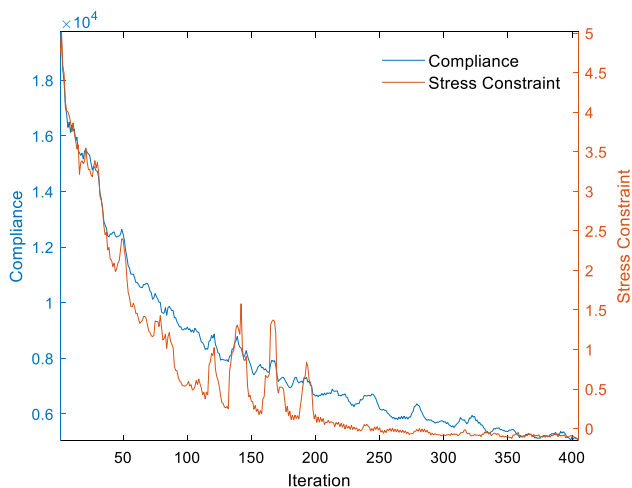
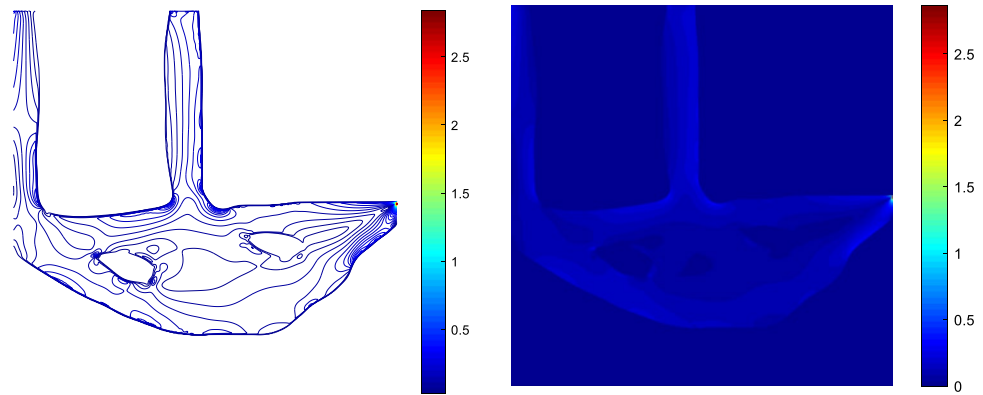


Fig. 32 Evolution of density field (Kernel: Gaussian, hidden layers:  $20 \times 20 \times 20$ )

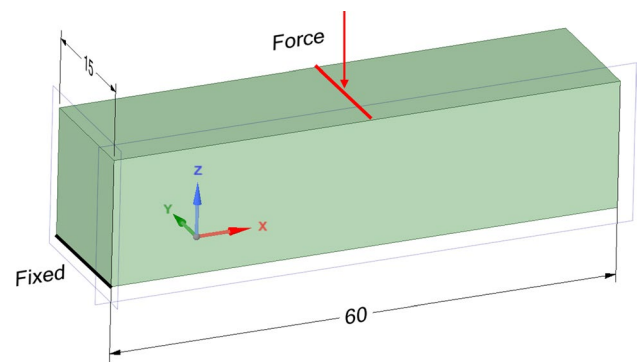
**Fig. 33** Stress distribution of optimal design (Kernel: Gaussian, hidden layers:  $20 \times 20 \times 20$ )



**Fig. 34** Convergence history

### 4.3 Compliance optimization for three dimensional MBB design

In this section, a three-dimensional MBB example is presented for compliance optimization. The MBB is modeled by a  $600 \times 150 \times 150$  hexahedral mesh, and the dimension of the design is demonstrated in Fig. 35. A uniform line force  $F = 1$  is applied on the mid-top of the rectangle domain. The neural network with three hidden layers of size  $20 \times 20 \times 20$  is implemented to represent the density field in the design domain. Note that three inputs are needed to represent coordinates  $x$ ,  $y$  and  $z$ . In the actual FEM analysis, only half of the design domain is modeled considering the geometric symmetry. The elastic constants are chosen as follows: elastic modulus  $E = 1$  and Poisson's ratio  $\mu = 0.3$ . The first numerical result is obtained using the Tansig kernel. The optimization converges after 60 iterations and the density evolution history is plotted in Fig. 36. To make a comparison, a Gaussian kernel is also employed, and the

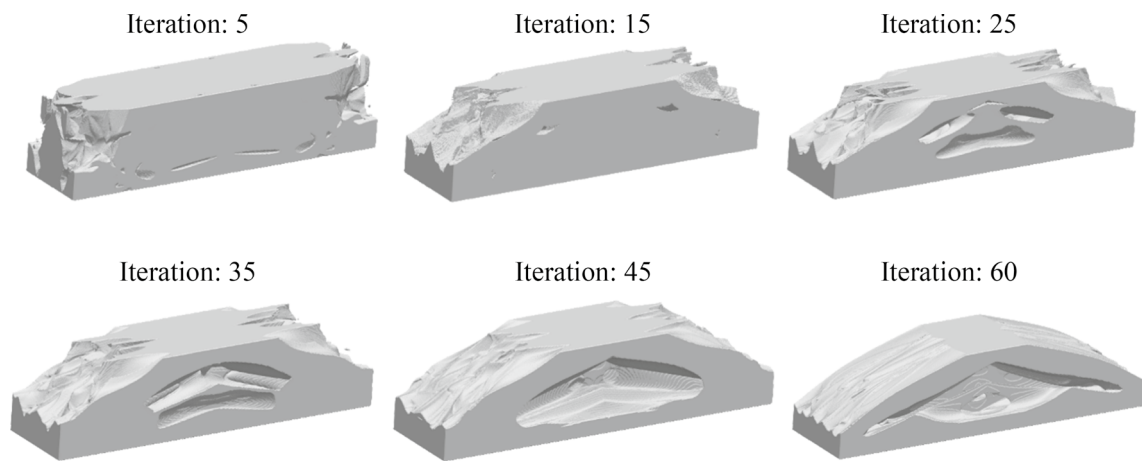


**Fig. 35** 3D MBB example

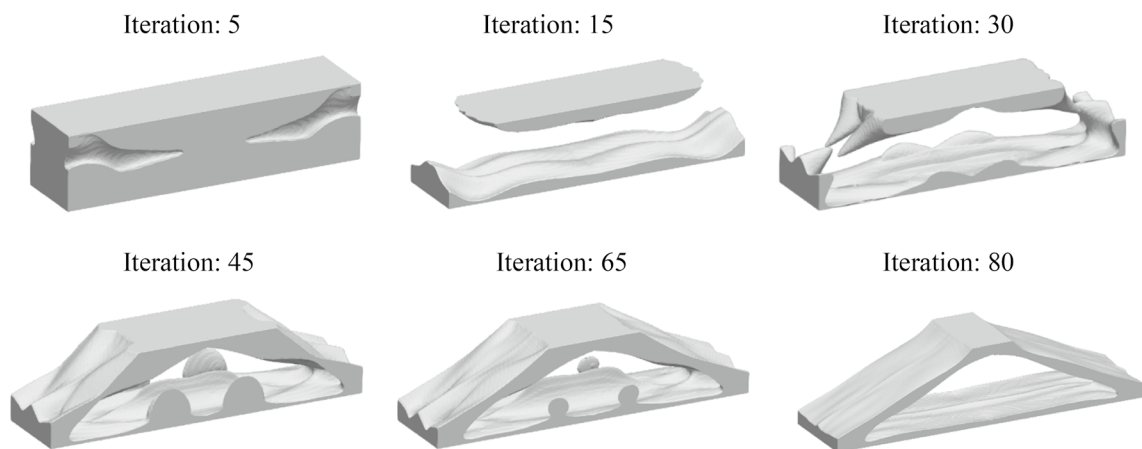
optimization progress is demonstrated in Fig. 37. The optimization converges after 80 iterations.

### 4.4 Stress constrained optimization for three-dimensional L-bracket design

To test the proposed algorithm in a three-dimensional case, a three-dimensional L-bracket example is plotted in Fig. 38. The dimensions of the L-bracket, boundary and loading conditions are found in Fig. 38. A distributed edge force  $F = 3$  is applied to the finite element model. The design domain is meshed with  $100 \times 100 \times 40$  uniform trilinear hexahedral elements with element size of  $h = 1$ . The material properties are the same as in the previous example. The p-norm stress constraint is set to be  $SC : \sigma_{pnorm} - 5 < 0$ , and volume fraction constraint is chosen as 0.3. Due to the presence of a sharp corner in the initial design, the stress is expected to concentrate at the corner with a high value. The neural network with three hidden layers of size  $20 \times 20 \times 20$  and Tansig kernel as the activation function is implemented to implicitly represent the density field. The moving limit of MMA algorithm is chosen as 0.005. The optimization converges after 120 iterations, and optimized density field is



**Fig. 36** Evolution of density field for MBB design (Tansig kernel)



**Fig. 37** Evolution of density field for MBB design (Gaussian kernel)

presented in Fig. 39. The final optimization result is transformed into a CAD model as shown in Fig. 42, where the stress distribution is plotted in Fig. 40. Apparently, the sharp corner disappears after optimization, and stress distribution tends to be uniform in the optimal structure. To validate the design, the commercial software ANSYS is applied to implement stress analysis. The tetrahedron mesh with 10 nodes is used to discretize the design (Total mesh number: 142785), the discretized finite element model and stress contour is plotted in Fig. 43. Stress optimization is a highly nonlinear optimization problem due to its local effects. This example successfully demonstrates that the deep learning method can represent complex geometry by generating effective optimal layout with significant decrease of design variables, i.e., from the original number of design variables of 400,000 to 941. The deep neural networks demonstrate excellent data compression ability. There are no small intricate features are found in optimal design, and the final optimal design

are represented in an implicit way. Meanwhile, no staggered phenomenon on the surface occurs due to the implicit representation method. The convergence history is plotted in Fig. 41.

## 5 Conclusion

In this paper, a density field representation algorithm based on deep learning is proposed to generate optimal design for compliance and stress constrained problems. The main conclusions are as follows,

- (a) The density field is represented by a neural network so that the design variables are reduced phenomenally compared to the conventional voxel-based optimization method.

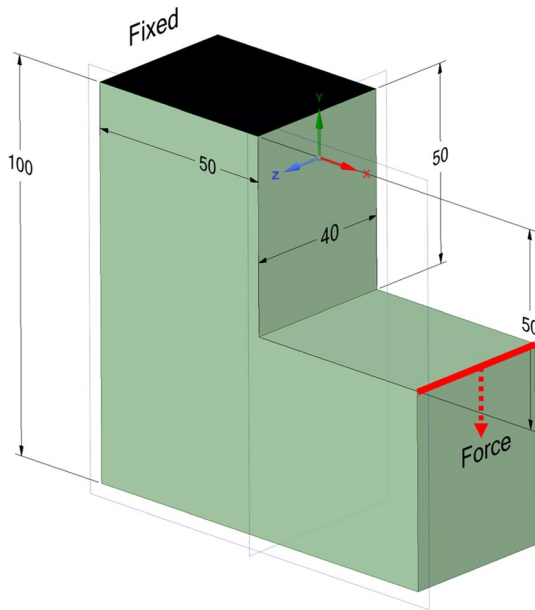


Fig. 38 3D L-bracket example

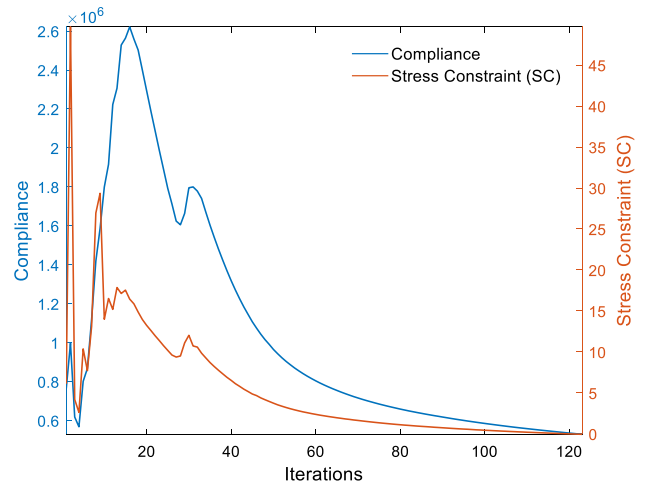


Fig. 41 Convergence history

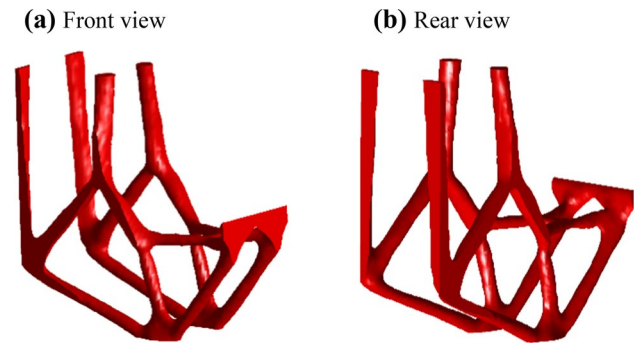


Fig. 39 Optimized material layout (Kernel: Tansig, hidden layers:  $20 \times 20 \times 20$ ) a front view, b rear view

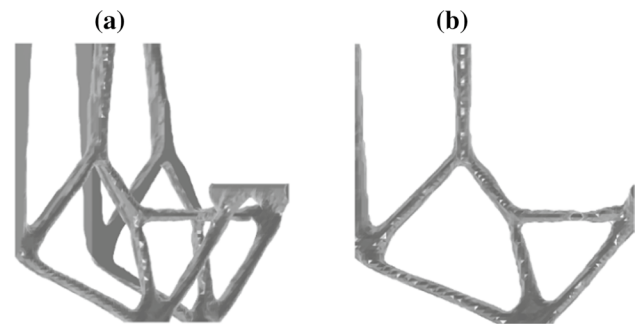
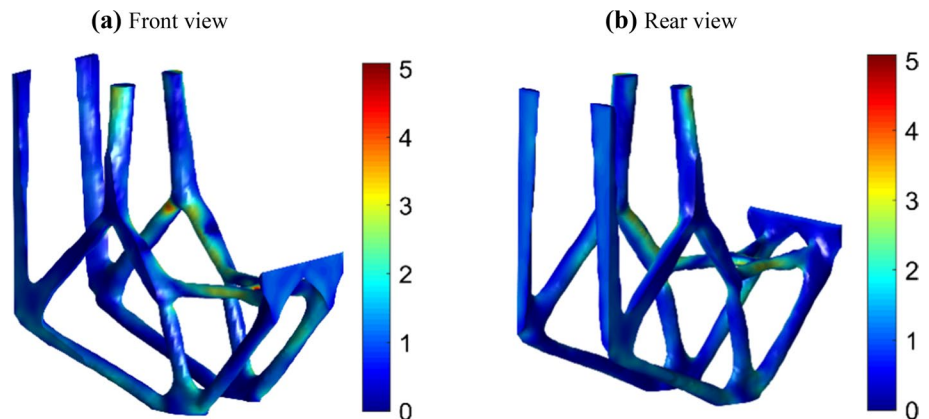


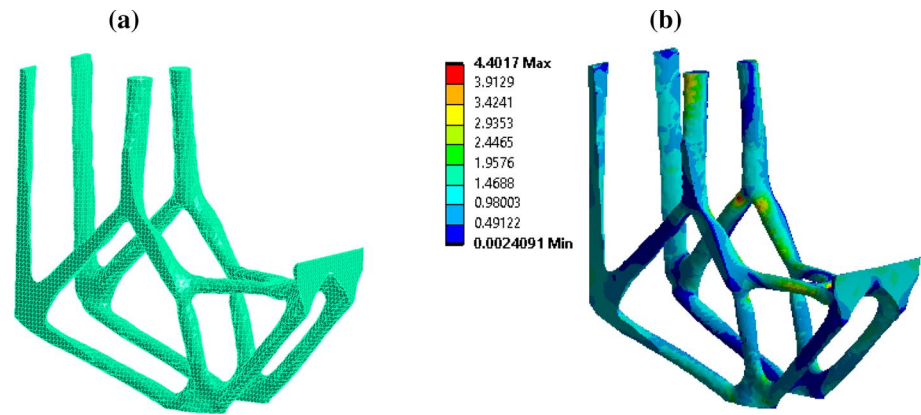
Fig. 42 CAD model a front view, b rear view

(b) Different kernel functions influence the optimized design, and the geometry complexity is directly related to the topology of neural networks. The simple optimal geometry is obtained with shallow neural networks.

Fig. 40 Stress distribution a front view, b rear view





**Fig. 43** a Tetrahedron mesh, b stress distribution

- (c) No filtering technique is needed in the proposed algorithm, and optimal designs are free from chessboard pattern [95].
- (d) Because the topology is represented in an implicit way, there is no staggered boundary found in the final design.

From the future perspective, the method proposed in this paper open a new opportunity to achieve a combination of deep learning with topology optimization in a geometric way. In fact, deep neural networks are only one of the deep learning models. In recently years, more powerful and sophisticated deep learning model are proposed (e.g., generative adversarial network [45] and convolutional neural network [96]). The method proposed in this paper is a real “marriage” between deep learning and topology optimization. Future work will focus on applying more deep learning models to represent density field such as CNN and GAN. Meanwhile, future directions include employing the classification method (e.g. decision tree algorithm [97], random forest [98]) for geometry representation to directly generate 0–1 solution in the design domain.

**Acknowledgements** The authors would like to acknowledge the support from National Science Foundation (CMMI-1634261).

## References

- Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71(2):197–224
- Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192(1–2):227–246
- Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. *J Appl Mech* 81(8):081009
- Sigmund O, Bondsgc M (2003) Topology optimization. State-of-the-art future perspectives. Technical University of Denmark, Copenhagen
- Lazarov BS, Wang F, Sigmund O (2016) Length scale and manufacturability in density-based topology optimization. *Arch Appl Mech* 86(1–2):189–218
- Zhang W, Li D, Zhou J, Du Z, Li B, Guo X (2018) A moving morphable void (MMV)-based explicit approach for topology optimization considering stress constraints. *Comput Methods Appl Mech Eng* 334:381–413
- Zhang W et al (2018) Topology optimization with multiple materials via moving morphable component (MMC) method. *Int J Numer Methods Eng* 113(11):1653–1675
- Zhang W et al (2017) Explicit three dimensional topology optimization via moving morphable void (MMV) approach. *Comput Methods Appl Mech Eng* 322:590–614
- Norato J, Bell B, Tortorelli D (2015) A geometry projection method for continuum-based topology optimization with discrete elements. *Comput Methods Appl Mech Eng* 293:306–327
- Zhang S, Gain AL, Norato JA (2017) Stress-based topology optimization with discrete geometric components. *Comput Methods Appl Mech Eng* 325:1–21
- Watts S, Tortorelli DA (2017) A geometric projection method for designing three-dimensional open lattices with inverse homogenization. *Int J Numer Methods Eng* 112(11):1564–1588
- White DA, Stowell ML, Tortorelli DA (2018) Topological optimization of structures using Fourier representations. *Struct Multi-discipl Optim* 58(3):1205–1220
- Gao J, Gao L, Luo Z, Li P (2019) Isogeometric topology optimization for continuum structures using density distribution function. *Int J Numer Methods Eng* 119(10):991–1017
- Gulian M, Raissi M, Perdikaris P, Karniadakis G (2019) Machine learning of space-fractional differential equations. *SIAM J Sci Comput* 41(4):A2485–A2509
- 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
- Raissi M, Karniadakis GE (2018) Hidden physics models: machine learning of nonlinear partial differential equations. *J Comput Phys* 357:125–141
- Raissi M, Wang Z, Triantafyllou MS, Karniadakis GE (2019) Deep learning of vortex-induced vibrations. *J Fluid Mech* 861:119–137
- Raissi M, Perdikaris P, Karniadakis GE (2017) Machine learning of linear differential equations using Gaussian processes. *J Comput Phys* 348:683–693
- Alber M et al. (2019) Multiscale modeling meets machine learning: what can we learn? arXiv:11958

20. Mao Z, Jagtap AD, Karniadakis GE (2020) Physics-informed neural networks for highspeed flows. *Comput Methods Appl Mech Eng* 360:112789
21. Zhang D, Lu L, Guo L, Karniadakis GE (2019) Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *J Comput Phys* 397:108850
22. Meng X, Li Z, Zhang D, Karniadakis GE (2019) PPINN: para-real physics-informed neural network for time-dependent PDEs. arXiv:10145
23. Yang L et al. (2019) Highly-scalable, physics-informed GANs for learning solutions of stochastic PDEs. arXiv:13444
24. Lei X, Liu C, Du Z, Zhang W, Guo X (2019) Machine learning-driven real-time topology optimization under moving morphable component-based framework. *J Appl Mech* 86(1):011004
25. Oh S, Jung Y, Kim S, Lee I, Kang N (2019) Deep generative design: integration of topology optimization and generative models. *J Mech Des* 141(11):111405
26. White DA, Arrighi WJ, Kudo J, Watts SE (2019) Multiscale topology optimization using neural network surrogate models. *Comput Methods Appl Mech Eng* 346:1118–1135
27. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal* 35(8):1798–1828
28. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
29. Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3D classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 652–660
30. Litany O, Bronstein A, Bronstein M, Makadia A (2018) Deformable shape completion with graph convolutional autoencoders. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1886–1895
31. Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S (2019) DeepSDF: learning continuous signed distance functions for shape representation. arXiv:05103
32. Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3D surface construction algorithm. *ACM Siggraph Comput Graph* 21(4):163–169
33. Zhou P, Du J, Lü Z (2018) A generalized DCT compression based density method for topology optimization of 2D and 3D continua. *Comput Methods Appl Mech Eng* 334:1–21
34. Luo Y, Bao J (2019) A material-field series-expansion method for topology optimization of continuum structures. *Comput Struct* 225:106122
35. White DA, Choi Y, Kudo J (2019) A dual mesh method with adaptivity for stress-constrained topology optimization. *Struct Multidiscipl Optim* 61:749–762
36. Luo Y, Xing J, Kang Z (2020) Topology optimization using material-field series expansion and Kriging-based algorithm: an effective non-gradient method. *Comput Methods Appl Mech Eng* 364:112966
37. Benkő P, Martin RR, Várady T (2001) Algorithms for reverse engineering boundary representation models. *Comput Aided Des* 33(11):839–851
38. Hart JC (1998) Morse theory for implicit surface modeling. In: Hart JC (ed) *Mathematical visualization*. Springer, Berlin, pp 257–268
39. Gomes A, Voiculescu I, Jorge J, Wyvill B, Galbraith C (2009) *Implicit curves and surfaces: mathematics, data structures and algorithms*. Springer, Berlin
40. Ucier T (1992) Feature-based image metamorphosis. *Comput Graph* 26:2
41. Li Q, Hong Q, Qi Q, Ma X, Han X, Tian J (2018) Towards additive manufacturing oriented geometric modeling using implicit functions. *Vis Comput Ind Biomed Art* 1(1):1–16
42. Yoo DJ (2011) Porous scaffold design using the distance field and triply periodic minimal surface models. *Biomaterials* 32(31):7741–7754
43. Turk G, Levoy M (1994) Zippered polygon meshes from range images. In: *Proceedings of the 21st annual conference on computer graphics and interactive techniques*, pp 311–318
44. Tripathi Y, Shukla M, Bhatt AD (2019) Implicit-function-based design and additive manufacturing of triply periodic minimal surfaces scaffolds for bone tissue engineering. *J Mater Eng Perform* 28(12):7445–7451
45. Goodfellow I et al. (2014) Generative adversarial nets. In: *Advances in neural information processing systems*, pp 2672–2680
46. Dai A, Ruizhongtai Qi C, Nießner M (2017) Shape completion using 3D-encoder-predictor cnns and shape synthesis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 5868–5877
47. Tan SM, Michael L (1995) Reducing data dimensionality through optimizing neural network inputs. *AIChE J* 41(6):1471–1480
48. Schoen AH (1970) Infinite periodic minimal surfaces without self-intersections. National Aeronautics and Space Administration
49. Eldan R, Shamir O (2016) The power of depth for feedforward neural networks. In: *Conference on learning theory*, pp 907–940
50. Lin HW, Tegmark M, Rolnick D (2017) Why does deep and cheap learning work so well? *J Stat Phys* 168(6):1223–1247
51. Liang S, Srikant R (2016) Why deep neural networks for function approximation? arXiv preprint arXiv:1610.04161
52. Telgarsky M (2015) Representation benefits of deep feedforward networks. arXiv:08101
53. Mhaskar HN, Poggio T (2016) Deep vs. shallow networks: an approximation theory perspective. *Anal Appl* 14(06):829–848
54. Matsugu M, Mori K, Mitari Y, Kaneda YJNN (2003) Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Netw* 16(5–6):555–559
55. Mhaskar H, Liao Q, Poggio T (2016) Learning functions: when is deep better than shallow. arXiv:00988
56. Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S (2019) DeepSDF: learning continuous signed distance functions for shape representation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 165–174
57. Vogl TP, Mangis J, Rigler A, Zink W, Alkon D (1988) Accelerating the convergence of the back-propagation method. *Biol Cybern* 59(4–5):257–263
58. Pujol J (2007) The solution of nonlinear inverse problems and the Levenberg–Marquardt method. *Geophysics* 72(4):W1–W16
59. Biegler LT, Conn AR, Coleman TF, Santosa FN (1997) *Large-scale optimization with applications: optimal design and control*. Springer, Berlin
60. Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. *Struct Multidiscipl Optim* 43(1):1–16
61. Lee E, James KA, Martins JR (2012) Stress-constrained topology optimization with design-dependent loading. *Struct Multidiscipl Optim* 46(5):647–661
62. Picelli R, Townsend S, Brampton C, Norato J, Kim H (2018) Stress-based shape and topology optimization with the level set method. *Comput Methods Appl Mech Eng* 329:1–23
63. Kiyono C, Vatanabe S, Silva E, Reddy J (2016) A new multi-p-norm formulation approach for stress-based topology optimization design. *Compos Struct* 156:10–19
64. Lian H, Christiansen AN, Tortorelli DA, Sigmund O, Aage N (2017) Combined shape and topology optimization for minimization of maximal von Mises stress. *Struct Multidiscipl Optim* 55(5):1541–1557

65. Zhou M, Sigmund O (2017) On fully stressed design and p-norm measures in structural optimization. *Struct Multidiscipl Optim* 56(3):731–736
66. Cai S, Zhang W (2015) Stress constrained topology optimization with free-form design domains. *Comput Methods Appl Mech Eng* 289:267–290
67. Xia L, Zhang L, Xia Q, Shi T (2018) Stress-based topology optimization using bi-directional evolutionary structural optimization method. *Comput Methods Appl Mech Eng* 333:356–370
68. Wang MY, Li L (2013) Shape equilibrium constraint: a strategy for stress-constrained structural topology optimization. *Struct Multidiscipl Optim* 47(3):335–352
69. Le C, Norato J, Bruns T, Ha C, Tortorelli D, Optimization M (2010) Stress-based topology optimization for continua. *Struct Multidiscipl Optim* 41(4):605–620
70. Bradley AM (2013) *PDE-constrained optimization and the adjoint method*. Technical Report. Stanford University. [https://cs.stanford.edu/~ambrad/adjoint\\_tutorial.pdf](https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf)
71. Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2018) Automatic differentiation in machine learning: a survey. *J Mach Learn Res* 18(153):5595–5637
72. Bartholomew-Biggs M, Brown S, Christianson B, Dixon L, Mathematics A (2000) Automatic differentiation of algorithms. *J Comput* 124(1–2):171–190
73. Andersson JA, Gillis J, Horn G, Rawlings JB, Diehl M (2019) CasADi: a software framework for nonlinear optimization and optimal control. *J Math Program Comput* 11(1):1–36
74. Holmberg E, Torstenfelt B, Klarbring A, Optimization M (2013) Stress constrained topology optimization. *Struct Multidiscipl Optim* 48(1):33–47
75. Rahmatalla S, Swan C (2004) A Q4/Q4 continuum structural topology optimization implementation. *Struct Multidiscipl Optim* 27(1–2):130–135
76. Girosi F, Jones M, Poggio T (1995) Regularization theory and neural networks architectures. *Neural Comput* 7(2):219–269
77. Baudat G, Anouar F (2001) Kernel-based methods and function approximation. In: *IJCNN'01. International joint conference on neural networks. Proceedings (Cat. No. 01CH37222)*, vol. 2. IEEE, pp 1244–1249
78. Elleuch K, Chaari A (2011) Modeling and identification of hammerstein system by using triangular basis functions. *Int J Electr Comput Eng* 1:1
79. Wand MP, Jones MC (1994) *Kernel smoothing*. Chapman and Hall, London
80. Yakubovich S, Zayed AI (1997) *Handbook of function and generalized function transformations*. Academic Press, New York
81. Carstensen JV, Guest JK (2018) Projection-based two-phase minimum and maximum length scale control in topology optimization. *Struct Multidiscipl Optim* 58(5):1845–1860
82. Carstensen JV, Guest JK (2014) New projection methods for two-phase minimum and maximum length scale control in topology optimization. In: *15th AIAA/ISSMO multidisciplinary analysis and optimization conference*, p 2297
83. Guest JK, Smith Genut LC (2010) Reducing dimensionality in topology optimization using adaptive design variable fields. *Int J Numer Methods Eng* 81(8):1019–1045
84. Guest JK (2009) Topology optimization with multiple phase projection. *Comput Methods Appl Mech Eng* 199(1–4):123–135
85. Guest JK (2009) Imposing maximum length scale in topology optimization. *Struct Multidiscipl Optim* 37(5):463–473
86. Guest J, Prevost J (2006) A penalty function for enforcing maximum length scale criterion in topology optimization. In: *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, p 6938
87. Guest JK, Prévost JH, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int J Numer Methods Eng* 61(2):238–254
88. Lazarov BS, Wang F (2017) Maximum length scale in density based topology optimization. *Comput Methods Appl Mech Eng* 318:826–844
89. Zhou M, Lazarov BS, Wang F, Sigmund O (2015) Minimum length scale in topology optimization by geometric constraints. *Comput Methods Appl Mech Eng* 293:266–282
90. Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. *Struct Multidiscipl Optim* 43(6):767–784
91. Sigmund O (2007) Morphology-based black and white filters for topology optimization. *Struct Multidiscipl Optim* 33(4–5):401–424
92. Sigmund O (2009) Manufacturing tolerant topology optimization. *Acta Mech Sin* 25(2):227–239
93. Lazarov BS, Sigmund O (2011) Filters in topology optimization based on Helmholtz-type differential equations. *Int J Numer Methods Eng* 86(6):765–781
94. Svanberg K (2007) MMA and GCMMA—two methods for nonlinear optimization. *Optim Syst Theory* 1:1–15
95. Rozvany GI (2009) A critical review of established methods of structural topology optimization. *Struct Multidiscipl Optim* 37(3):217–237
96. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
97. Freund Y, Mason L (1999) The alternating decision tree learning algorithm. In: *icml*, vol 99, pp 124–133
98. Ho TK (1995) Random decision forests. In: *Proceedings of 3rd international conference on document analysis and recognition*, vol 1. IEEE, pp 278–282

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