Proceedings of the ASME 2026 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2026
August 23-August 26, 2026, Houston, USA

DETC2026-188694

NEURAL TOPOLOGY OPTIMIZATION WITH MOVABLE NON-DESIGN FEATURES

Timothy Sheffield
Department of Mechanical Engineering
The Pennsylvania State University
University Park, Pennsylvania 16802

Yuewan Sun
Walker Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas 78712

Guha Manogharan
Department of Mechanical Engineering
The Pennsylvania State University
University Park, Pennsylvania 16802

Zhenghui Sha*
Walker Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

Optimizing both material layout and the placement of critical non-design features is essential for modern engineering structures, as it directly impacts mechanical performance, weight efficiency, and manufacturability. This work presents a neural network-based topology optimization framework that concurrently optimizes material density distributions and the placement of movable non-design features (e.g., a hole structure for embedded electronic sensor). Traditional integrated layout and topology optimization approaches rely on gradient-based methods that are susceptible to local minima and require complex sensitivity analyses for moving the non-design features. To address these limitations, a multi-layer perceptron is employed to simultaneously parameterize both the SIMP-like density field and the spatial coordinates of non-design features as differentiable network outputs. A finite difference augmented backpropagation scheme bridges the non-differentiable FEA solver with PyTorch, enabling end-to-end optimization via the Adam optimizer. The framework is validated on four structural benchmark problems, MBB beam, top-loaded cantilever, mid-loaded cantilever, and bottom-loaded cantilever, each paired with three non-design feature types: a hole, disk, and annulus. In all cases, the algorithm successfully converges the non-design feature from a centered initialization to a structurally advantageous location while optimizing the surrounding material distribution. The disk and annulus features exhibit robust convergence, while the hole feature shows sensitivity-related instabilities in void-dominated regions. Results demonstrate that the neural network-based framework is a viable and flexible strategy for integrating non-design feature layout and topology optimization.

Keywords: Topology optimization, Non-design feature placement, Neural networks, Finite element analysis (FEA)

1 INTRODUCTION

Topology optimization (TO) is a structural design tool that finds the optimal material placement for structural performance and material usage. Since the pioneering work by Bendsøe and Kikuchi [1], frameworks such as density, phase field, and evolutionary approaches have been developed, each having their own advantages [2]. Traditionally, TO algorithms use gradient-based optimization techniques to efficiently place material. As an example, Solid Isotropic Material with Penalization (SIMP) works by penalizing the densities between 0 and 1 within the elements

*Corresponding author: zsha@austin.utexas.edu

in a discretized mesh. Therefore, the gradient-based optimization solver pushes each element to either 0 (void) or 1 (solid), resulting in a very clear display for where to place material contingent upon compliance and material constraints.

A common addition to TO problems is including non-design features where some of the domain is forced to be either void or solid, independent of the optimization algorithm. This consideration allows for design constraints to be added into the design domain without interfering with the optimization algorithm. Often, these features are at a set location and cannot change throughout the optimization process, but fixing the location eliminates the possibility to place the features in the most optimal spot. As many transportation and aerospace applications require placement of components, such as batteries, microcontrollers, motors, electrical components, and interconnects, it is advantageous to discover the best possible placement for the component. For example, designing a drone will require lightweight structures but also ideal placement of components without severely impacting structural performance. With advancements in additive manufacturing, it is possible to embed these electrical components directly into the printing process, creating full electronic circuits [3–6]. Additionally, ceramic inserts have been embedded into structures using metal additive manufacturing, specifically employing a hybrid directed energy deposit process [7].

However, using traditional TO algorithms, the placement of the non-design feature requires a large penalty to computational time as another gradient-based optimization or additional constraints need to be solved. Also, gradient-based optimization algorithms may converge to a local minima rather than the global optimum. In many movable non-design feature works, the non-design feature remains in the vicinity of the initial placement, even if a different location may be a better placement [8–11].

In recent years, machine learning (ML) methods have been applied to TO. Of the machine learning methods, neural networks (NN) are some of the most common. Broadly, NN methods for TO are used to either accelerate traditional TO methods or directly execute the TO using a NN. Readers should refer to [12] for more information on using NN in TO. In the direct execution methods, the density of each element is implemented as individual neurons and iterated over multiple layers within the NN [13].

With the advantages of NN, the contribution of this work is the design of a new NN-based TO framework that directly solves a SIMP-like TO with movable non-design features. In doing so, we aim to improve upon traditional TO methods with movable non-design features by simultaneously considering the material distribution and non-design feature location in the objective function read back into the NN. This will reduce the risk of capturing a local minimum. Additionally, in using NN to directly solve TO, each element needs its own neuron [13], but adding a movable non-design feature only adds 2 neurons in the NN for a 2D formulation to complete TO problem which only marginally increases computational time.

2 RELATED WORK AND CONCEPTUAL BACKGROUND

2.1 Traditional Approaches in Topology Optimization

TO has evolved significantly since the homogenization method proposed by Bendsøe and Kikuchi [1]. The field is currently dominated by density-based methods, such as SIMP, and boundary-based methods such as the level-set method (LSM) [14]. While these methods are effective for optimizing material layouts in a fixed design domain, many engineering designs also contain non-design features, such as bolt holes, embedded sensors, electronic components, cooling channels, or locally preserved material regions. These regions can be prescribed as passive solid or passive void elements, but fixed passive regions do not solve the placement problem this study aims to solve and can lead to suboptimal layouts.

To address this issue, several integrated layout and topology optimization (ILTO) methods have been developed. Early work optimized the placement of predesigned rigid components using continuous location and orientation variables [15], and later studies considered multi-component systems using finite-circle constraints, density points, embedded meshing, and consistent material interpolation schemes [16]. Other density- and level-set-based formulations have optimized embedded movable holes, flexible void regions, movable components, and stress-driven movable holes [8, 10, 17, 18]. These methods provide rigorous formulations for coupled layout-topology design, but they typically require problem-specific geometric parameterizations, analytical sensitivity derivations, or additional constraints to manage component motion and interaction with the surrounding topology.

A closely related implementation strategy is feature mapping or geometry projection. Instead of remeshing as a component moves, these methods map high-level geometric features onto a fixed analysis grid through pseudo-density, ersatz-material, or immersed-boundary descriptions [11]. For example, Guest developed a projection-based approach for optimizing the layout of discrete objects [19], and Norato et al. introduced a differentiable geometry projection method for continuum topology optimization with discrete elements [20]. These approaches provide an effective analysis interface between explicit geometric features and fixed-grid FEA; however, the mapping operation by itself does not define how the feature variables should be explored jointly with the surrounding density field. As a result, the overall formulation often still relies on a separate optimizer, problem-specific parameter updates, and sensitivity information for the geometric variables. This distinction motivates a formulation in which the feature location and the material density are treated as coupled outputs of the same differentiable neural parameterization, rather than as two design descriptions connected only through a projection map.

Moving morphable component (MMC) represents another important branch of explicit geometric topology optimization.

In MMC, structural components are described by a small set of geometric variables and topology description functions [21], with later variants improving the component geometry, for example through curved skeletons [22]. Guo's group has also connected MMC with machine learning. For example, Lei et al. trained machine-learning models for real-time prediction under the MMC framework [23], and their later work used deep learning to predict MMC design variables directly [24]. These approaches benefit from compact geometric descriptions, but they depend on predefined component forms or training data for MMC variables. In contrast, the present work uses the neural network as the optimization-side parameterization that outputs both a SIMP-like density field and the coordinates of a movable circular non-design feature.

2.2 Neural Networks in Topology Optimization

The integration of ML into structural mechanics has opened new avenues for accelerating or reparameterizing TO. Early applications of NNs often treated TO as an image-to-image translation task. For instance, Yu et al. [25] utilized conditional generative adversarial networks (cGANs) to predict optimized topologies directly from boundary conditions, achieving near-instantaneous results after training. Similarly, Li et al. [26] demonstrated the use of deep CNNs to map initial strain fields to final optimized structures. These approaches can be computationally attractive, but they generally require representative training datasets and may be tied to the discretization and boundary-condition classes used during training.

More recently, neural reparameterization has been explored as an optimization strategy rather than only a data-driven prediction strategy. Instead of optimizing a discrete grid of independent design variables, the density field is represented as the output of a neural network whose weights are optimized. Hoyer et al. [27] showed that this change of parameterization can alter optimization behavior and promote smoother layouts. Chandrasekhar and Suresh proposed TOuNN, in which a neural network directly represents the density field for topology optimization [13]. Neural implicit representations have also been used to define structures through signed distance functions, allowing mesh-free geometric descriptions and resolution-independent querying [28]. These studies motivate the use of NNs as differentiable design representations, but most do not explicitly address movable non-design regions whose geometric parameters must be coupled to the material interpolation used in FEA.

2.3 Co-optimization of Topology and Discrete Features

Despite the progress in NN-based TO, many existing models assume a static design domain or predict only the optimized material distribution. When a non-design feature such as a circular hole, disk, or annulus must change its location, the optimization must connect the feature coordinates to the density field that enters the material interpolation and FEA. Conventional CNN prediction models do not naturally provide this connection because the coordinates of the feature are not part of the differentiable design representation. Some hybrid approaches have attempted to use reinforcement learning (RL) for topology generation or placement decisions, but these approaches often require many environment interactions and repeated FEA calls during training [29].

Our work bridges this gap by introducing a unified neural framework in which the same network outputs both the element-wise density variables and the coordinates of the movable non-design feature. The feature is then mapped to the analysis mesh using a smooth circular indicator function, yielding an effective physical density field before SIMP interpolation. In this way, the proposed formulation directly addresses both parts of the coupled problem: where the circular feature should be placed and how the surrounding material should be distributed. The method should therefore be interpreted as a neural reparameterization and fixed-grid feature-mapping strategy for ILTO, rather than as a guarantee of global optimality.

3 Methodology

3.1 Overview of the Proposed Framework

The primary objective of this work is to develop a unified optimization framework that concurrently determines the optimal material distribution (topology) and the optimal positioning of non-design features, such as circular voids. Unlike traditional approaches that often optimize these two aspects in decoupled stages, our framework utilizes NN as a global reparameterizer for both the density field and the discrete geometric coordinates of the features, as illustrated in Fig. 1.

The proposed hybrid computational loop consists of three core modules, as illustrated in the following logic:

- **Neural Module:** Instead of directly optimizing individual element densities as independent variables, we employ a deep neural network G_θ, parameterized by weights θ. The network maps a latent input vector to a high-dimensional output, which is partitioned into two distinct design components: a pseudo-density vector $\rho \in [0,1]^{n_e}$ representing the structural topology, and a set of spatial coordinates (x_h, y_h) defining the center of the non-design feature in the physical domain.
- **Differentiable Masking and FEA Evaluation:** To integrate discrete geometric features into the continuous topology, a smooth indicator function (masking) is applied. This ensures that the hole can "move" across the domain in a differentiable manner. The resulting effective density field, ρ_{eff}, which accounts for both the generated topology and the hole placement, is then passed to a Finite Element Anal-

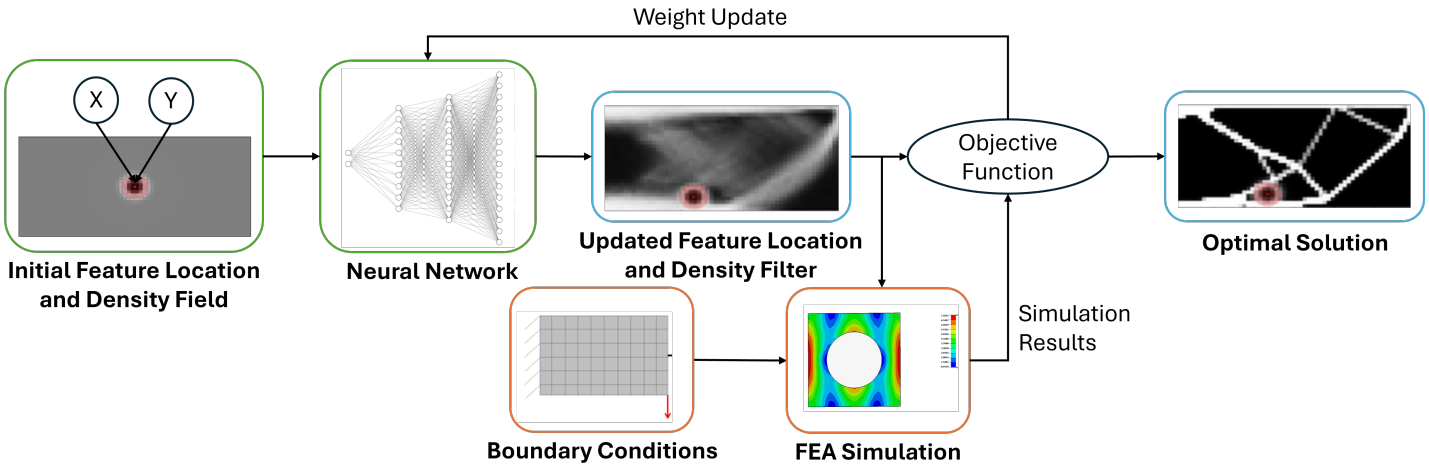

FIGURE 1: The overview of framework.

ysis (FEA) solver. The solver evaluates the structural performance, specifically the compliance $\mathscr{C}$ and the total volume fraction $\mathscr{V}$.

- **Hybrid Gradient Backpropagation:** A significant challenge in this integration is that the FEA solver is treated as a "black-box" and is non-differentiable within the standard autograd environment. To bridge this gap, we implement a Finite Difference (FD) augmented backpropagation scheme. While the objective function $\mathscr{L} = \mathscr{C} + \beta\mathscr{V}$ is evaluated via FEA, its sensitivities with respect to the NN outputs are computed using finite difference approximations. These numerical gradients are then injected into the PyTorch autograd engine, allowing the network weights θ to be updated via the Adam optimizer through the chain rule:

$$\frac{\partial \mathscr{L}}{\partial \theta} = \frac{\partial \mathscr{L}}{\partial out} \cdot \frac{\partial out}{\partial \theta} \tag{1}$$

where $out = \{\rho, x_h, y_h\}$ represents the coupled output of the network.

By parameterizing the hole coordinates as direct network outputs, the model leverages the NN's ability to navigate complex, non-convex design spaces, potentially identifying superior global layouts that traditional gradient-based moving boundary methods might fail to capture due to local minima.

3.2 Design Domain and Mesh

Often, continuum TO problems are defined through a fixed design domain, Ω, that does not change between each iteration. For SIMP approaches, the fixed design domain allows for a fixed, Eulerian mesh. Using QUAD4 elements, the R^2 space Ω is discretized, as seen in Fig. 2.

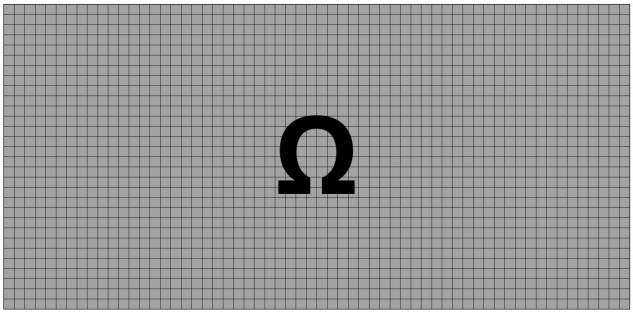

FIGURE 2: Design domain with QUAD4 elements.

3.3 FEA

For a compliance-based TO, the compliance is calculated at each iteration. FEA is a reliable methodology for accomplishing this task. The mesh is extended into a density field, ρ, where each element in the mesh is given ρ_e. First, an isotropic linear elasticity in 2D with the fourth-order constitutive tensor is adopted, in Eq. 2:

$$C_{abcd} = \mu\left(\delta_{ac}\delta_{bd} + \delta_{ad}\delta_{bc}\right) + \lambda\,\delta_{ab}\delta_{cd} \quad a,b,c,d \in \{1,2\} \tag{2}$$

where δ_{ij} is the Kronecker delta and the Lamé parameters are chosen as

$$\lambda = \frac{\nu E_0}{1-\nu^2} \qquad \mu = \frac{E_0}{2(1+\nu)} \tag{3}$$

which correspond to the common 2D FEA calculations [30]. ν is the Poisson ratio of the material and E_0 is the elastic modulus of the material. The constitutive tensor is computed outside of the stiffness calculation loop for computational efficiency.

Using the constitutive tensor, the stiffness at each element is calculated. For QUAD4 elements, each element's stiffness is calculated via Eq. 4:

$$K_e = \int_{\Omega_e} B^\top C B \, \mathrm{d}\Omega \approx \sum_{m=1}^{4} B(\xi_m, \eta_m)^\top C B(\xi_m, \eta_m) J^{(m)} \alpha^{(m)} \quad (4)$$

with **B** as the strain–displacement matrix assembled from the gauss shape functions and J is the Jacobian of the shape functions. m is the Gauss point where there are 4 in QUAD4 elements [30].

Using the SIMP method, each of the elements are scaled with respect to the density of the element.

$$\tilde{K}_e(\rho_e) = \frac{E_{\min} + \rho_e^p (E_0 - E_{\min})}{E_0} K_e \quad (5)$$

where $\tilde{K}_e$ is the scaled elastic element matrix is computed with E_0 [31] and p is the SIMP penalty parameter. $E_{\min}$ is calculated by

$$E_{\min} = E_0 \times 10^{-6} \quad (6)$$

The unknown displacement Degrees of Freedom (DOF) indexed by u and the prescribed displacement DOFs by p. Prescribed displacement DOFs are nodes that are defined by the boundary conditions, and thus, are not influenced by the FEA process. After a consistent reordering, the global system reads

$$\begin{bmatrix} K_{uu} & K_{up} \\ K_{pu} & K_{pp} \end{bmatrix} \begin{bmatrix} U_u \\ U_p \end{bmatrix} = \begin{bmatrix} F_u \\ F_p \end{bmatrix}. \quad (7)$$

To solve for the unknown displacement values, the prescribed displacement DOFs for both forces and displacements are removed. The elimination of prescribed displacement DOFs is accomplished by implementing Eq. 8 and Eq. 9.

$$F_{\text{eff}} = F_u - K_{up} U_p \quad (8)$$

$$U_u = K_{uu}^{-1} F_{\text{eff}}. \quad (9)$$

The compliance of the total part is then calculated through Eq. 10 by summing the compliance at each element.

$$c(\rho) = F_u^\top U_u = \sum_{e=1}^{n_{\text{el}}} u_e^\top \tilde{K}_e(\rho_e) u_e \quad (10)$$

where u_e is the calculated displacement of each element of the mesh. For more information on the FEA, the reader is encouraged to reference the textbooks [30, 31].

3.4 Neural Network Module

Instead of directly optimizing pixel-wise density variables, we introduce a neural network module that generates the structural design from a low-dimensional latent representation. Specifically, a Multi-Layer Perceptron (MLP) is used as a continuous mapping that outputs both the material distribution and the geometric parameters of the design.

Due to the smooth functional mapping of the neural network, the generated structures exhibit improved spatial coherence without requiring additional filtering or regularization techniques. Furthermore, the network is designed to explicitly output both the density distribution and the spatial coordinates of geometric features, including holes and movable non-design regions. By treating the locations of these constrained regions as learnable variables rather than fixed inputs, the proposed formulation enables material topology and geometric configuration to be optimized simultaneously. This results in a more flexible design process in which structural layout and geometric constraints evolve jointly in an end-to-end manner.

The network architecture, denoted as G_θ, consists of a series of fully connected layers with ReLU activation functions. Given a fixed latent input **z**, the network produces an output that is branched into two distinct design components:

1. **Density Field Generation:** One branch of the network output is mapped through a Sigmoid activation function to generate the element-wise pseudo-density field $\rho \in [0,1]^{n_e}$. By representing the density field as a neural output, we ensure that the generated structure maintains a level of spatial coherence and inherent smoothness. This reparameterization transforms the optimization problem from the discrete pixel space to the continuous weight space of the network, which has been shown to improve convergence and avoid some checkerboard artifacts common in traditional SIMP methods.

2. **Optimization of Movable Non-Design Region Position:** Simultaneously, the network outputs the coordinates of the non-design feature (e.g., a circular hole). To restrict the movement of this feature within the physical design domain, the coordinate outputs are passed through a Tanh activation function followed by a linear scaling:

$$\begin{cases} x_h = \frac{1}{2}(\tanh(\mathbf{y}_x)+1)(x_{max}-x_{min})+x_{min} \\ y_h = \frac{1}{2}(\tanh(\mathbf{y}_y)+1)(y_{max}-y_{min})+y_{min} \end{cases} \tag{11}$$

where $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ define the physical boundaries of the domain. To ensure the feature remains fully contained within the structure, a clipping operation is further applied such that $(x_h, y_h) \in [x_{min}+r, x_{max}-r] \times [y_{min}+r, y_{max}-r]$, where r is the feature radius.

To ensure a robust starting point for the co-optimization process, we employ a target-driven bias initialization for the final layer of the network. By shifting the initial logits such that the mapping outputs zero in the normalized space, the feature is positioned precisely at the domain's geometric center and the density field is initialized to a uniform value of 0.5. This strategy serves as a form of optimization warm-start, effectively preventing the gradient-based updates from becoming trapped in localized boundary minima during the critical early stages of training.

3.5 Density Filter

To avoid checkerboard patterns seen in discretized mesh TO algorithms, a mesh independent density filter is implemented. The density filter acts by considering the physical density of an element by a weighted average of surrounding elements as defined by a radius [32, 33]. The filtered density, $\tilde{\rho}$, for each element is calculated by

$$\tilde{\rho}_e = \frac{\sum_{j \in \mathcal{N}_e} w(x_j, y_j)\rho_j}{\sum_{j \in \mathcal{N}_e} w(x_j, y_j)} \tag{12}$$

where $\mathcal{N}_e$ represents the surrounding elements within the filtering domain, ρ_j is the density of the element surrounding the filtered element, and $w(x_j, y_j)$ is the 2D weighting function defined as

$$w(x_j, y_j) = R - \sqrt{(x_j - x_e)^2 + (y_j - x_e)^2} \tag{13}$$

with R being the filter radius, x_e and y_e are the coordinates of the element being filtered, and x_j and y_j are the coordinates of the neighboring elements. To normalize the density output, a threshold projection is applied. The projected physical density, $\bar{\rho}_e$, is calculated via

$$\bar{\rho}_e = \frac{\tanh(\beta\eta) + \tanh\big(\beta(\tilde{\rho}_e - \eta)\big)}{\tanh(\beta\eta) + \tanh\big(\beta(1-\eta)\big)} \tag{14}$$

where β is the projection parameter and η is a threshold value [34].

3.6 Objective Function

Traditionally, optimization in TO requires a constrained optimization problem with robust and efficient optimization algorithms [31, 35]. However, many NN optimization packages, such as the Adam optimization [36], minimize an unconstrained optimization problem. Thus, constrained optimization problems defined in most SIMP TO is transitioned into the unconstrained optimization problem

$$\min\ c(\rho) + \alpha\left(\frac{\sum_{e=1}^{n_{el}} \bar{\rho}_e}{n_{el}}\right) \tag{15}$$

where $c(\bar{\rho}_e)$ is the total compliance from Eq. 10 and α is a penalty term to balance the importance of compliance with material usage. Reducing α puts a larger emphasis on minimizing compliance. As the compliance formulation is based on an the unnormalized density, ρ, α should account for the units of compliance in relation to the normalized formulation of $\bar{\rho}_e$.

4 EXPERIMENTS

Experiments were run on Messerschmitt–Bölkow–Blohm (MBB) beams, top-loaded cantilevers, mid-loaded cantilevers, and bottom-loaded cantilevers as seen in Fig. 3. Each of these experiments were selected for their asymmetric boundary conditions. Additionally, the movable non-design features of a *hole* (circular void region), *disk* (circular material region), and *annulus* (circular void region surrounded by a material boundary region) were implemented. Fig. 4 displays the non-design features on the MBB beam before any optimization. The initial placement of the features is at the center of the domain.

Unless otherwise specified, the following parameters were kept fixed in all experiments:

- Material properties: E_0 = 193E9, $\nu = \frac{1}{3}$
- Boundary Conditions: F = 1E7
- The grid was 60x30 elements for all simulations. The physical dimensions are in a ratio of 2x1.

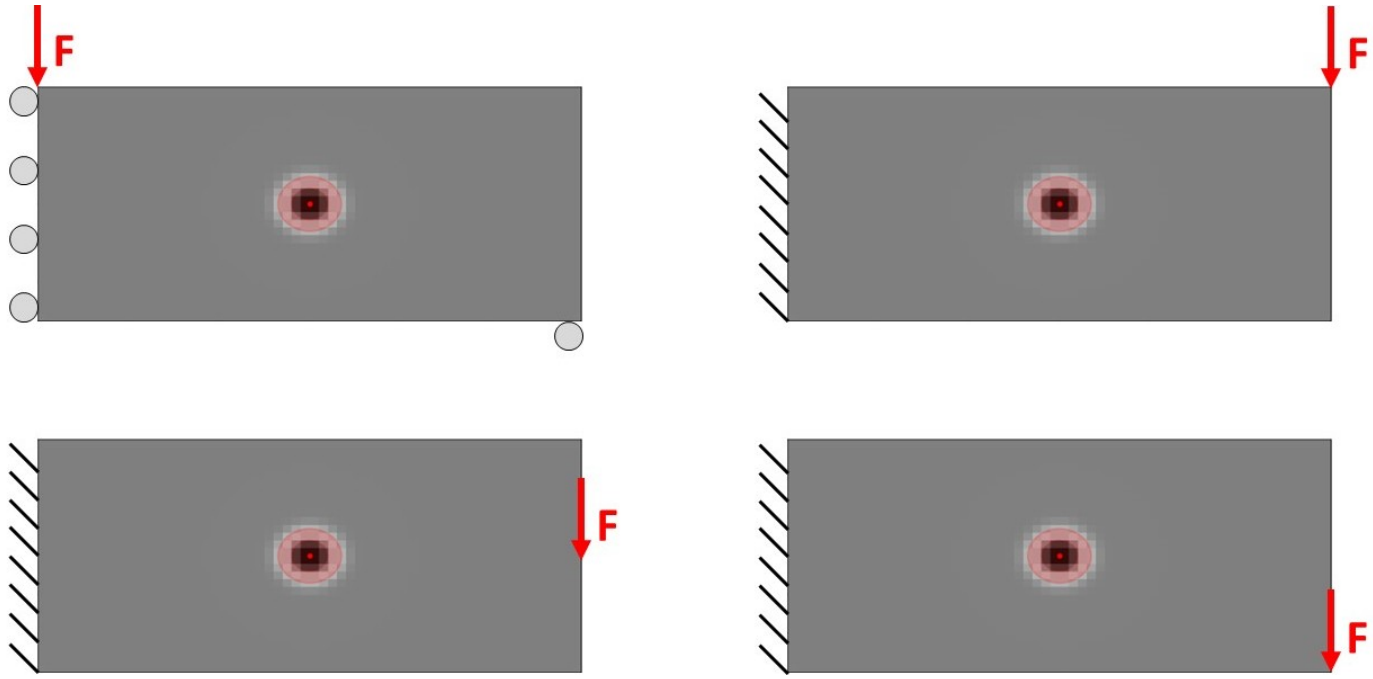

FIGURE 3: The problems solved are the (top left) MBB beam, (top right) top-loaded cantilever, (bottom left) mid-loaded cantilever, and the bottom-loaded cantilever.

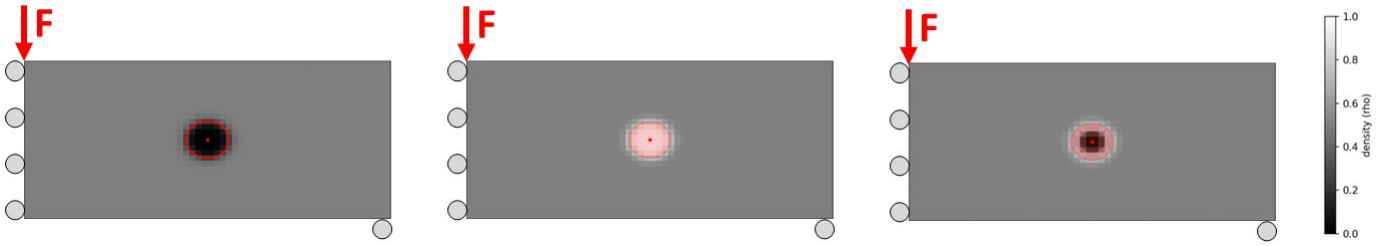

FIGURE 4: The (left) *hole*, (middle) *disk*, and (right) *annulus* on the initial setup for the MBB beam.

- The learning rate for the Adam optimizer was 0.001.
- The α parameter was set to 1E6.
- The filter radius was set to $1.5 \times \min(h_i)$ where h_i is the slide length of the elements.
- The p parameter was set to 3 [37].
- β was initially set at 2 but increased 1 every 50 iterations. η was set to 0.5.
- The diameter of the *hole* and *disk* were 7 elements. The annulus had an outer diameter of 7 elements with a inner void region with a diameter of 4 elements.
- A termination criterion of $\Delta(c(\rho) + \alpha \left(\frac{\sum_{e=1}^{n_{el}} \tilde{\rho}_e}{n_{el}} \right)) < 100$ was established between two consecutive iterations.

5 RESULTS AND DISCUSSION

The experiments are run until completion for the MBB beam, top-cantilever, mid-cantilever, and bottom cantilever. The final converged geometry for each problem and each feature is shown in Fig. 5. Each of the final feature locations have moved from its initial placement at the center to its optimal location. Figs. 6 - 9 show the convergence of the minimization function where it is normalized against the first iteration, as well as the feature path to its final location on the 60x30 element grid.

Overall, the TO algorithm optimizes the material placement as well as the location of the non-design feature. Generally, the minimization functions for all experiments continuously decrease until convergence and the non-design features find a path to the best placement.

One area of interest is the spikes seen in the minimization

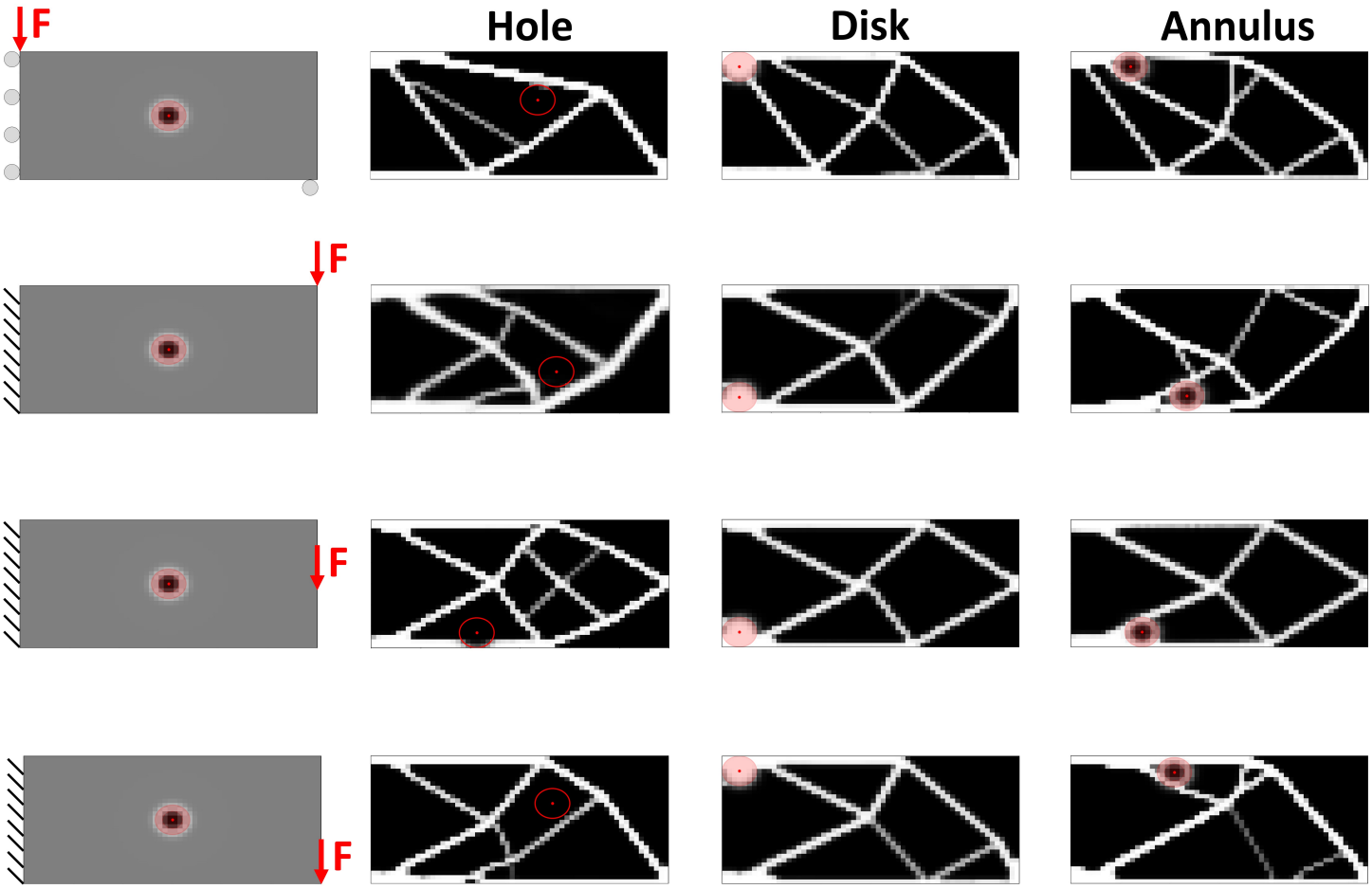

FIGURE 5: Converged material density distributions with non-design feature locations for all problem types and feature combinations.

function for the *hole* non-design feature TO. Looking at the iteration directly before the spike and at the spike, the *hole* approaches a thin material section, reducing the material width and significantly increasing the compliance. The *disk* and *annulus* are more robust and spikes are not seen in the minimization function. Additionally, the location of the *disk* and *annulus* smoothly find its ideal location with small movements in later iterations as the density of elements change while sometimes, the location of the *hole* moves in the opposite direction of the optimal location. The neural network is unable to understand small changes in the location of the hole when the location of the *hole* does not interact with material. Therefore, the minimization function may decrease with a change in density, but the location of the *hole* changes in that iteration, giving the NN false signals. Eventually, the *hole* is moved so that it interacts with material, increasing the compliance. The NN corrects the course of the *hole*, reducing the minimization function back to the value before the spike. As this phenomenon is not a concern in the *disk* and *annulus* cases, these non-design features produce more robust results.

6 LIMITATIONS AND FUTURE WORK

One limitation is that the density of some elements does not converge to either 0 or 1. As seen in Fig. 5, some elements are not black or white, suggesting an intermediate density. Possibly a simple change to α or the convergence criteria may produce results where most elements converged to a density of either 0 or 1, but further investigation is necessary.

This work required a density filter to remove checkerboard patterns throughout the TO solution. However, in other NN-based TO algorithms, such as [13], a density filter is not incorporated and no checkerboard pattern was produced. By requiring the density filter, the computational time required per iteration increased, and more iterations were necessary. The combined ef-

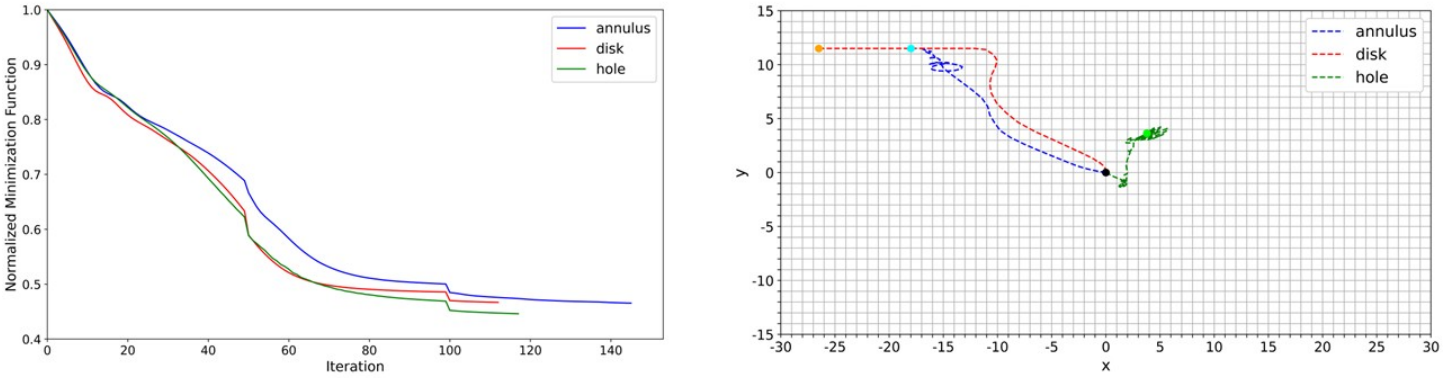

FIGURE 6: The results for the MBB experiments showing the (left) value of normalized minimization function at each iteration and (right) the path of the movable non-design features until convergence on the 60x30 element grid.

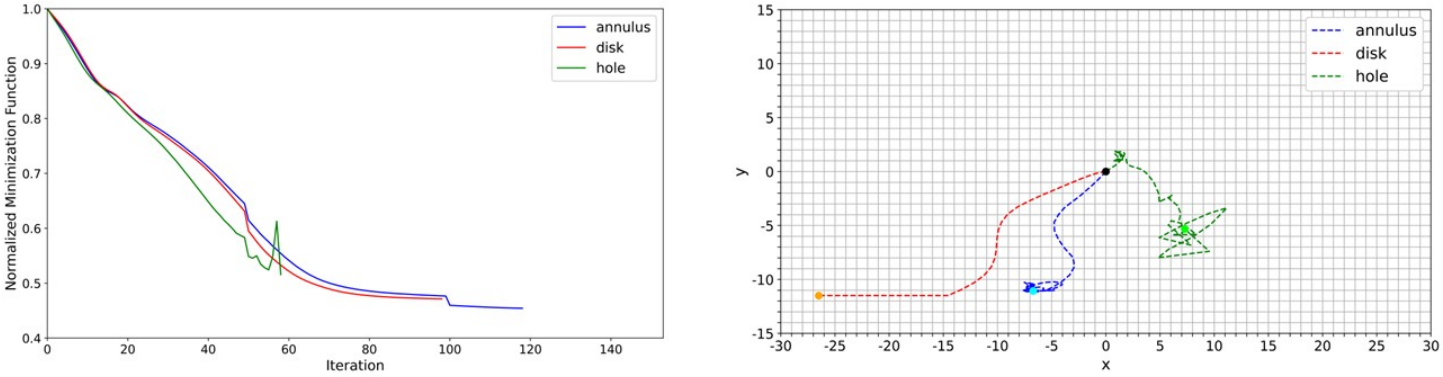

FIGURE 7: The results for the top-loaded cantilever experiments showing the (left) value of normalized minimization function at each iteration and (right) the path of the movable non-design features until convergence on the 60x30 element grid.

fect led to a significant increase in the computational time before convergence. The large computational time limits the utility of the algorithm, especially for 3D design domains.

While the non-design features reach many areas of the design domain, more robust methods need to be developed to ensure that the location of feature converges to the global minimum rather than a local minimum. More experimentation is needed to ensure that the global minimum is found in each TO solution. Possible testing could include first optimizing the location for the non-design feature and then optimizing the density of elements.

For future work, this framework can be extended in many directions to increase performance and add more capabilities. Here, only a single movable non-design feature is presented, but engineering systems often have more than one component to be embedded. Introducing two or more movable non-design spaces would enable this capability. The feature options could also be extended to more than symmetrical, circular non-design spaces and include an angle design variable such that the orientation of the non-design space is considered and optimized. Similarly, adding analysis for diverse examples with differing boundary conditions, design domains, and objective functions would extend the capabilities of this work.

Another extension of this work is to add constraints. Many traditional TO formulations use constrained optimization techniques where more constraints can be added within that framework. However, the unconstrained objective function of Eq. 15 would need to be expanded to enable improved design or consider manufacturing capabilities. For example, to add overhang constraints for additive manufacturing, constrained optimization problems implement this overhang constraint within the constraints section of the constrained optimization problem [38, 39]. For the unconstrained optimization problem, considering constraints would require an extension to the optimization formulation.

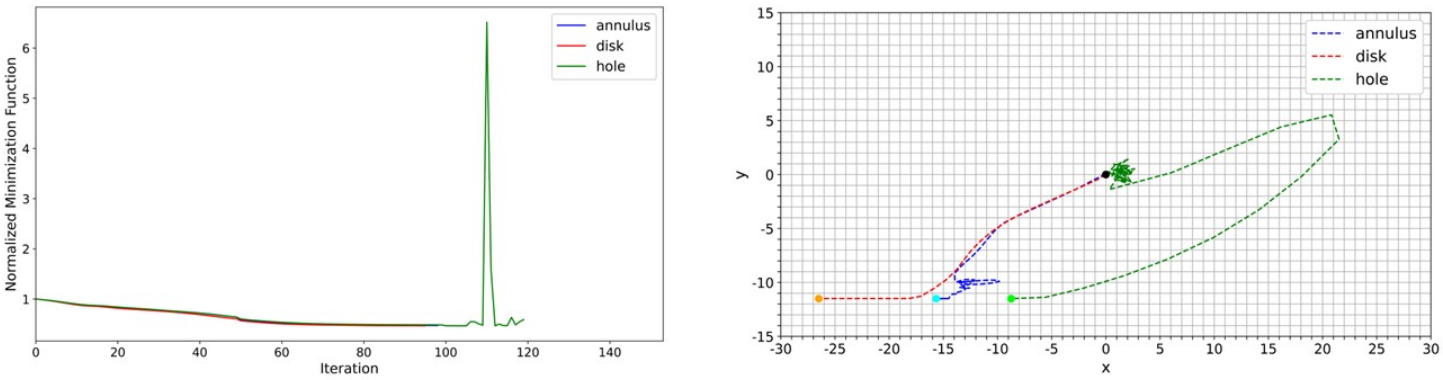

FIGURE 8: The results for the mid-loaded cantilever experiments showing the (left) value of normalized minimization function at each iteration and (right) the path of the movable non-design features until convergence on the 60x30 element grid.

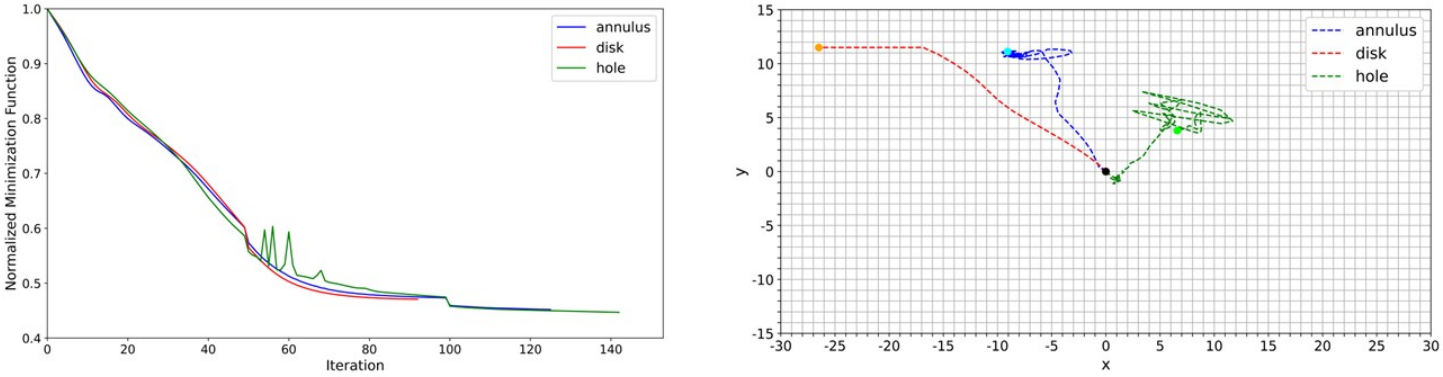

FIGURE 9: The results for the bottom-loaded cantilever experiments showing the (left) value of normalized minimization function at each iteration and (right) the path of the movable non-design features until convergence on the 60x30 element grid.

7 CONCLUSION

This work presents a NN-based TO framework for material density distributions and movable non-design feature placement. By parameterizing both the material density field and the spatial coordinates of non-design features as outputs of a single MLP, the proposed approach eliminates the need for decoupled or sequentially staged optimization.

The framework was validated across four structural benchmark problems, MBB beam, top-loaded cantilever, mid-loaded cantilever, and bottom-loaded cantilever, each paired with three non-design feature types: a *hole*, *disk*, and *annulus*. In all cases, the algorithm successfully optimized the material topology and feature placement, converging the non-design feature from a centered initialization to a structurally advantageous location. The *disk* and *annulus* features demonstrated particularly robust convergence behavior, while the *hole* feature exhibited sensitivity-related instabilities when traversing low-density regions, a limitation attributed to weak gradient signals in void-dominated areas.

Overall, this work demonstrates that neural reparameterization to include the location of the non-design feature is a viable and flexible strategy for integrated layout and topology optimization, offering a promising alternative to traditional TO with movable non-design spaces that are prone to local minima entrapment.

ACKNOWLEDGMENTS

This research was funded by the National Science Foundation (NSF) Foundational Research in Robotics Award number 2334881.

REFERENCES

[1] Bendsøe, M. P., and Kikuchi, N., 1988. "Generating optimal topologies in structural design using a homogenization method". *Computer methods in applied mechanics and engineering,* ***71***(2), pp. 197–224.

[2] Sigmund, O., and Maute, K., 2013. "Topology optimization approaches: A comparative review". *Structural and multidisciplinary optimization,* ***48***(6), pp. 1031–1055.

[3] Espalin, D., Muse, D. W., MacDonald, E., and Wicker, R. B., 2014. "3d printing multifunctionality: structures with electronics". *The International Journal of Advanced Manufacturing Technology,* ***72***(5), pp. 963–978.

[4] Macdonald, E., Salas, R., Espalin, D., Perez, M., Aguilera, E., Muse, D., and Wicker, R. B., 2014. "3d printing for the rapid prototyping of structural electronics". *IEEE access,* ***2***, pp. 234–242.

[5] Williams, M., Jacob, A., and Manogharan, G., 2025. "Continuous 5-axis routing of syringe deposited conductive traces over topology optimized structures". *Manufacturing Letters,* ***44***, pp. 540–551.

[6] Vath, C., Masum Billah, K., and Manogharan, G., 2023. "Development of multimaterial additive manufacturing systems for embedded electronics".

[7] Feldhausen, T., Yelamanchi, B., Gomez, A., Du Plessis, A., Heinrich, L., Saleeby, K., Fillingim, K., Post, B., Love, L., Cortes, P., et al., 2023. "Embedding ceramic components in metal structures with hybrid directed energy deposition". *The International Journal of Advanced Manufacturing Technology,* ***125***(9), pp. 4425–4433.

[8] Wang, X., Hu, P., and Kang, Z., 2020. "Layout optimization of continuum structures embedded with movable components and holes simultaneously". *Structural and Multidisciplinary Optimization,* ***61***(2), pp. 555–573.

[9] Li, Z., Wang, L., and Luo, Z., 2022. "A feature-driven robust topology optimization strategy considering movable non-design domain and complex uncertainty". *Computer Methods in Applied Mechanics and Engineering,* ***401***, p. 115658.

[10] Wang, X., Liu, H., Kang, Z., Long, K., and Meng, Z., 2021. "Topology optimization for minimum stress design with embedded movable holes". *Computers & structures,* ***244***, p. 106455.

[11] Wein, F., Dunning, P. D., and Norato, J. A., 2020. "A review on feature-mapping methods for structural optimization". *Structural and Multidisciplinary Optimization,* ***62***(4), pp. 1597–1638.

[12] Shin, S., Shin, D., and Kang, N., 2023. "Topology optimization via machine learning and deep learning: a review". *Journal of Computational Design and Engineering,* ***10***(4), pp. 1736–1766.

[13] Chandrasekhar, A., and Suresh, K., 2021. "Tounn: topology optimization using neural networks". *Structural and Multidisciplinary Optimization,* ***63***(3), pp. 1135–1149.

[14] Wang, M. Y., Wang, X., and Guo, D., 2003. "A level set method for structural topology optimization". *Computer Methods in Applied Mechanics and Engineering,* ***192***(1), pp. 227–246.

[15] Qian, Z., and Ananthasuresh, G. K., 2004. "Optimal embedding of rigid objects in the topology design of structures". *Mechanics Based Design of Structures and Machines,* ***32***(2), pp. 165–193.

[16] Zhu, J., Zhang, W., and Beckers, P., 2009. "Integrated layout design of multi-component system". *International Journal for Numerical Methods in Engineering,* ***78***(6), pp. 631–651.

[17] Kang, Z., and Wang, Y., 2013. "Integrated topology optimization with embedded movable holes based on combined description by material density and level sets". *Computer Methods in Applied Mechanics and Engineering,* ***255***, pp. 1–13.

[18] Clausen, A., Aage, N., and Sigmund, O., 2014. "Topology optimization with flexible void area". *Structural and Multidisciplinary Optimization,* ***50***(6), pp. 927–943.

[19] Guest, J. K., 2015. "Optimizing the layout of discrete objects in structures and materials: A projection-based topology optimization approach". *Computer Methods in Applied Mechanics and Engineering,* ***283***, pp. 330–351.

[20] Norato, J. A., Bell, B. K., and Tortorelli, D. A., 2015. "A geometry projection method for continuum-based topology optimization with discrete elements". *Computer Methods in Applied Mechanics and Engineering,* ***293***, pp. 306–327.

[21] Guo, X., Zhang, W., and Zhong, W., 2014. "Doing topology optimization explicitly and geometrically—a new moving morphable components based framework". *Journal of Applied Mechanics,* ***81***(8), 05, p. 081009.

[22] Guo, X., Zhang, W., Zhang, J., and Yuan, J., 2016. "Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons". *Computer Methods in Applied Mechanics and Engineering,* ***310***, pp. 711–748.

[23] Lei, X., Liu, C., Du, Z., Zhang, W., and Guo, X., 2019. "Machine learning-driven real-time topology optimization under moving morphable component-based framework". *Journal of Applied Mechanics,* ***86***(1), p. 011004.

[24] Rochefort-Beaudoin, T., Vadean, A., Gamache, J.-F., and Achiche, S., 2023. "Supervised deep learning for the moving morphable components topology optimization framework". *Engineering Applications of Artificial Intelligence,* ***123***, p. 106436.

[25] Yu, Y., Hur, T., Jung, J., and Jang, I. G., 2018. "Deep learning for determining a near-optimal topological design without any iteration". *Structural and Multidisciplinary Optimization,* ***59***(3), Oct., p. 787–799.

[26] Li, B., Huang, C., Li, X., Zheng, S., and Hong, J., 2019.

"Non-iterative structural topology optimization using deep learning". *Computer-Aided Design,* ***115***, pp. 172–180.

[27] Hoyer, S., Sohl-Dickstein, J., and Greydanus, S., 2019. Neural reparameterization improves structural optimization.

[28] Zehnder, J., Li, Y., Coros, S., and Thomaszewski, B., 2021. Ntopo: Mesh-free topology optimization using implicit neural representations.

[29] Brown, N. K., Garland, A. P., Fadel, G. M., and Li, G., 2022. ""deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains"". *Materials Design,* ***218***, p. 110672.

[30] Hughes, T. J., 2000. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation.

[31] Bendsoe, M. P., and Sigmund, O., 2003. *Topology Optimization: Theory, Methods, and Applications*. Springer Science & Business Media.

[32] Bruns, T. E., and Tortorelli, D. A., 2001. "Topology optimization of non-linear elastic structures and compliant mechanisms". *Computer methods in applied mechanics and engineering,* ***190***(26-27), pp. 3443–3459.

[33] Bourdin, B., 2001. "Filters in topology optimization". *International journal for numerical methods in engineering,* ***50***(9), pp. 2143–2158.

[34] Wang, F., Lazarov, B. S., and Sigmund, O., 2011. "On projection methods, convergence and robust formulations in topology optimization". *Structural and multidisciplinary optimization,* ***43***(6), pp. 767–784.

[35] Svanberg, K., 1987. "The method of moving asymptotes—a new method for structural optimization". *International journal for numerical methods in engineering,* ***24***(2), pp. 359–373.

[36] Kingma, D. P., and Ba, J., 2014. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.

[37] Bendsøe, M. P., and Sigmund, O., 1999. "Material interpolation schemes in topology optimization". *Archive of applied mechanics,* ***69***(9), pp. 635–654.

[38] Gaynor, A. T., Meisel, N. A., Williams, C. B., and Guest, J. K., 2014. "Topology optimization for additive manufacturing: considering maximum overhang constraint". In 15th AIAA/ISSMO multidisciplinary analysis and optimization conference, p. 2036.

[39] Gaynor, A. T., and Guest, J. K., 2016. "Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design". *Structural and Multidisciplinary Optimization,* ***54***(5), pp. 1157–1172.