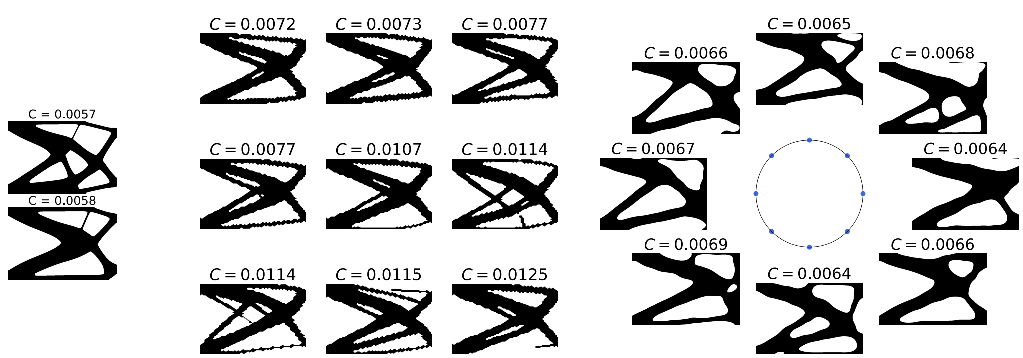
Diverse Topology Optimization using Modulated Neural Fields

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Topology optimization (TO) is a family of computational methods that derive near-optimal geometries from formal problem descriptions. Despite their success, established TO methods are limited to generating single solutions, restricting the exploration of alternative designs. To address this limitation, we introduce *Topology Optimization using Modulated Neural Fields* (TOM) – a data-free method that trains a neural network to generate structurally compliant shapes and explores diverse solutions through an explicit diversity constraint. The network is trained with a solver-in-the-loop, optimizing the material distribution in each iteration. The trained model produces diverse shapes that closely adhere to the design requirements. We validate TOM on established 2D and 3D TO benchmark problems. Our results show that TOM generates more diverse solutions than any previous method, all while maintaining near-optimality and without relying on a dataset. These findings open new avenues for engineering and design, offering enhanced flexibility and innovation in structural optimization. [1]



(a) Deflated barrier (b) TopoDiff (c) TOM (ours)

Figure 1: Solutions to the cantilever problem generated by three TO methods. (a) *Deflated barrier* – a classical TO method generating high-quality but few solutions. (b) *TopoDiff* – a data-driven diffusion model generating many, but low-quality solutions. (c) *TOM* – our modulated neural field trained with a solver-in-the-loop and a diversity constraint generating diverse near-optimal structures. Here, we use a circular modulation space to capture a smooth manifold of solutions. All methods minimize the structural compliance C, which measures the total displacement of the loaded shape.

[1]The code will be made public upon acceptance.

Submitted to the AI for Science workshop (NeurIPS 2025). Do not distribute.

1 Introduction

Topology optimization (TO) is a computational design technique to determine the optimal material distribution within a given design space under prescribed boundary conditions. A common objective in TO is the minimization of structural compliance, which measures the displacement under load. Due to the non-convex nature of TO problems, these methods generally provide near-optimal solutions, with no guarantees of converging to global optima [1].

Traditional TO methods are limited to producing a single design. However, generating multiple diverse solutions is important to balance performance with other considerations, such as manufacturability, cost, and aesthetics. Therefore, we propose **T**opology **O**ptimization using **M**odulated Neural Fields (TOM), an approach to generate diverse, near-optimal designs: A neural network parametrizes the shape representations and is trained to generate near-optimal solutions. The model is optimized using a solver-in-the-loop approach [43], where the neural network iteratively adjusts the design based on feedback from a physics-based solver.

To enhance the diversity of solutions, we introduce an explicit diversity constraint during training. TOM not only enables the generation of multiple, diverse designs but also leverages machine learning to explore the design space more efficiently than traditional TO methods [36].

Empirically, we validate our method on TO for linear elasticity problems in 2D and 3D. Our results demonstrate that TOM obtains more diverse solutions than prior work while being substantially faster and remaining near-optimal. This addresses a current limitation in TO and opens new avenues for automated engineering design.

Our main contributions are summarized as follows:

1. We introduce TOM, the first method for data-free, solver-in-the-loop neural network training, which generates diverse solutions adhering to structural requirements.
2. We introduce a novel diversity constraint variant for neural density fields based on the Chamfer discrepancy that ensures the generation of distinct and meaningful shapes and enhances the exploration of the designs.
3. Empirically, we demonstrate the efficacy and scalability of TOM on 2D and 3D problems, showcasing its ability to generate a variety of near-optimal designs. Our approach significantly outperforms existing methods in terms of solution diversity.

2 Background

2.1 Topology optimization

TO is a computational method developed in the late 1980s to determine optimal structural geometries from mathematical formulations [6]. TO iteratively updates a material distribution within a design domain under specified loading and boundary conditions. Due to the non-convex nature of most TO problems, convergence to a global minimum is not guaranteed. Instead, the goal is to achieve a near-optimal solution, where the objective closely approximates the global optimum. There are four prominent method families widely recognized in TO. In this work, we focus on *SIMP* and refer the reader to Yago et al. [47] for a more detailed introduction.

Solid isotropic material with penalization (SIMP) is a prominent TO method we adapt for TOM. SIMP operates on a mesh with mesh points $\mathbf{x}_i \in \mathcal{X}, i \in \{1, ..., N\}$ in the design region. The aim is to find a binary density function at each mesh point $\rho(\mathbf{x}_i) \in \{0, 1\}$, where $\rho(\mathbf{x}_i) = 0$ represents void and $\rho(\mathbf{x}_i) = 1$ represents solid material. To make this formulation differentiable, the material density ρ is relaxed to continuous values in $[0, 1]$. A common objective of SIMP is to minimize the compliance C, a measure of deformation under load. The SIMP objective is then formulated as a constrained optimization problem:

$$\begin{aligned} \text{min} : \quad & C(\rho) = \mathbf{u}^T \mathbf{K}_\rho \mathbf{u} \\ \text{s.t.} : \quad & V = \sum_{i=1}^{N} \rho_i v_i \leq V^* \\ & 0 \leq \rho_i \leq 1 \quad \forall i \in N \end{aligned} \tag{1}$$

where $\mathbf{u}$ is the displacement vector, $\mathbf{K}_\rho$ is the global stiffness matrix, V is the shape volume, and V^* is the target volume. The density field is optimized iteratively. In each iteration, a finite element (FEM) solver computes the compliance and provides gradients to update the density field ρ. To encourage binary densities, intermediate values are penalized by raising ρ to the power $p > 1$. Hence, the stiffness matrix is defined as $\mathbf{K}_\rho = \sum_{i=1}^{N} \rho_i^p \mathbf{K}_i$, where $\mathbf{K}_i$ describes the stiffness of solid cells and depends on material properties.

TO for multiple solutions. Generating multiple design alternatives for TO problems is crucial for many real-world engineering cases where single optima often don't exist. However, classical TO algorithms typically yield a single solution and do not ensure convergence to a global minimum. Papadopoulos et al. [32] introduce the deflated barrier (DB) method, an extension of the classical SIMP approach that can find multiple solutions. By employing a search strategy akin to depth-first search, DB identifies multiple solutions without relying on initial guess variations, thereby enhancing design diversity.

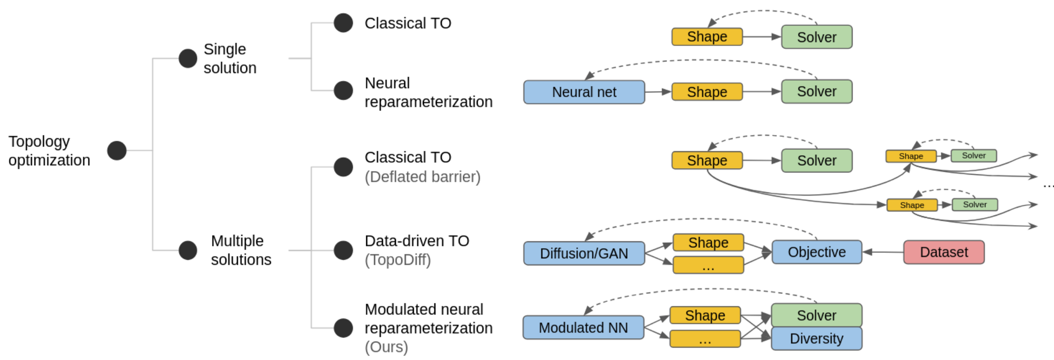


Figure 2: Taxonomy of classical and neural topology optimization methods. The dashed lines indicate iterative updates, such as gradient descent. TOM is the first data-free method to produce multiple shapes with a neural network.

2.2 Shape generation with neural networks

Neural fields offer a powerful framework for geometry processing, utilizing neural networks to model shapes implicitly. This approach enables high-quality and topologically flexible shape parameterizations [12]. The two prevalent methods for representing implicit shapes are signed distance functions (SDF) [33, 2] and density (or occupancy) [29] fields. We opt for the density representation due to its straightforward compatibility with SIMP optimization.
Given a $d_x \in \{2, 3\}$ dimensional domain $\mathcal{X} \subset \mathbb{R}^{d_x}$, a neural density field employs a neural network $f_\theta : \mathcal{X} \to (0, 1)$ with parameters θ to define the shape $\Omega := \{x \in \mathcal{X} | f_\theta(x) \geq \tau\}$ as the $\tau \in [0, 1]$ super-level-set of f_θ.

Conditional neural fields. While a neural density field represents a single shape, a *conditional* neural field represents a set of shapes with a single neural network [26]. Generally, one can condition on text, point clouds, or other modalities of interest [49]. In this work, we use a modulation code $\mathbf{z} \in \mathbb{R}^{d_z}$ as an additional input to the network. The resulting network $f_\theta(\mathbf{x}, \mathbf{z})$ parametrizes a set of shapes. There are different ways to incorporate the modulation vector into the network, such as input concatenation [33], hypernetworks [17], or attention [35]. In this work, we use input concatenation, as it is simple and fast to train.

Diversity constraint can be used to modify the TO problem formulation (1) to facilitate the discovery of multiple solutions. The diversity constraint introduced in geometry-informed neural networks (GINNs) [8] defines a diversity measure δ on the set of shapes $\{\Omega_i\}$ as

$$\delta(\{\Omega_i\}) = \left(\sum_j \left[\min_{k \neq j} d(\Omega_j, \Omega_k) \right]^{1/2} \right)^2 . \tag{2}$$

This measure builds upon a chosen dissimilarity function $d(\Omega_i, \Omega_j)$. Essentially, δ encourages diversity by maximizing the distance between each shape and its nearest neighbor.

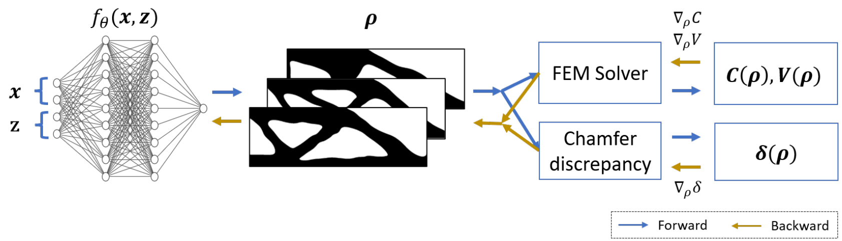


Figure 3: A single iteration of TOM trained with $M = 3$ shapes in parallel. In each iteration, the input to the network consists of the mesh vertices $\{\mathbf{x}_i\}_{i=1,\dots,N}$ and modulation vectors $\{\mathbf{z}\}_{j=1,\dots,M}$. The network outputs densities ρ_j at the vertices $\mathbf{x}_i$ for each shape. The densities are passed to the FEM solver, which computes the compliances C_j and volumes V_j, as well as their gradients $\nabla_\rho C_j$ and $\nabla_\rho V_j$. The diversity loss $\delta(\rho)$ and its gradient $\nabla_\rho \delta$ are based on the Chamfer discrepancy between the surface points of the shapes.

GINNs utilize a dissimilarity function on the shape boundary, but their approach is limited to signed distance functions (SDFs). In Section 3.2, we show how to adapt the Chamfer discrepancy as a dissimilarity function on density fields.

2.3 Topology optimization with neural networks

Figure 2 presents an overview of various TO methods that search for either single or multiple solutions, further categorized by their use of neural networks.

Neural reparameterization uses a neural network to represent the material distribution in a discretization-free manner. Existing work explores training a single shape by parametrizing the material *density* [40, 9, 10, 18] or *boundary* [13], which is optimized with a solver-in-the-loop. NITO [31] uses modulation of a neural field to introduce constraints on the boundary of a *single* shape. NTopo [48] uses a conditional neural field to generate individual solutions for different topology optimization problems, adapting to factors like target volume or force vector position. TOM, however, finds multiple solutions, not just one per setting.

Data-driven neural TO uses a dataset of generated solutions to train neural networks that can then generate diverse solutions. Most prior works use generative adversarial networks (GANs) [16, 15, 44, 30]. Recent work showed the superiority of diffusion models for data-driven TO [25]. Data-driven methods require a large amount of compute to generate a dataset and train a large generative model and aim to *amortize* these costs by fast inference on new, unseen problem settings.

3 Method

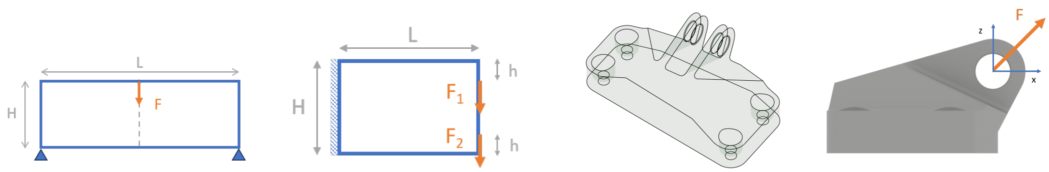
3.1 TO using modulated neural fields

Definitions. Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ be the domain of interest in which there is a shape $\Omega \subset \mathcal{X}$. Let $\mathcal{Z} \subset \mathbb{R}^{d_z}$ be a discrete or continuous modulation space, where each $\mathbf{z} \in \mathcal{Z}$ parametrizes a shape $\Omega_\mathbf{z}$. The possibly infinite set of all shapes is denoted by $\Omega_\mathcal{Z} = \{\Omega_\mathbf{z} | \mathbf{z} \in \mathcal{Z}\}$. The modulation vectors $\{\mathbf{z}_i\}$ are sampled according to a probability distribution $p(\mathcal{Z})$.
For density representations a shape $\Omega_\mathbf{z}$ is defined as the set of points with a density greater than the level $\tau \in [0, 1]$, formally $\Omega_\mathbf{z} = \{\mathbf{x} \in \mathcal{X} \mid \rho_\mathbf{z}(\mathbf{x}) > \tau\}$. While some prior work on occupancy networks [28] treats τ as a tunable hyperparameter, we follow Jia et al. [19] and fix the level at $\tau = 0.5$. We model the density $\rho_\mathbf{z}(\mathbf{x}) = f_\theta(\mathbf{x}, \mathbf{z})$, corresponding to the modulation vector $\mathbf{z}$ at a point $\mathbf{x}$ using a neural network f_θ, with θ being the learnable parameters.

TO using modulated neural fields. Our approach, akin to standard SIMP, formulates a constrained optimization problem aiming to minimize an objective function for multiple shapes simultaneously, while adhering to a volume constraint for each shape. Central to our method is the introduction of a diversity constraint, denoted as $\delta(\Omega_\mathcal{Z})$, which is defined over multiple shapes to reduce their similarity. This leads to the following constrained optimization problem:

$$
\begin{aligned}
\min: \quad & \mathbb{E}_{\mathbf{z}\sim p(\mathcal{Z})}\left[C(\rho_{\mathbf{z}})\right] \\
\text{s.t.}: \quad & V_{\mathbf{z}} = \int_{\mathcal{X}} \rho_{\mathbf{z}}(\mathbf{x}) d\mathbf{x} \leq V^* \\
& 0 \leq \rho_{\mathbf{z}}(\mathbf{x}) \leq 1 \quad \delta^* \leq \delta(\Omega_{\mathcal{Z}})
\end{aligned} \tag{3}
$$

where V^* and δ^* are target volumes and diversities, respectively. TOM updates the density fields of multiple shapes iteratively. In each iteration, the density distribution of each shape is computed and the resulting densities are passed to the FEM solver. The FEM solver calculates the compliance loss C and gradients ∇C. To accelerate the computation, parallelization of the FEM solver is employed across multiple CPU cores, with a separate process dedicated to each shape.
The diversity loss and its gradient are computed on the GPU using PyTorch [34], along with any optional geometric losses similar to GINNs.
We use the adaptive augmented Lagrangian method (ALM) [4] to automatically balance multiple loss terms. Additionally, we gradually increase the sharpness parameter β of the Heaviside filter (Equation 14), which acts as annealing. The complete TOM method is concisely presented in Algorithm 1.



(a) MBB (b) Cantilever (c) Jet engine bracket

Figure 4: Three benchmark problem definitions. (a) *MBB*: The boundary is fixed at the attachment points at the bottom left and right corners. The dotted line indicates the symmetry axis at $\frac{L}{2}$ where the force F is applied. (b) *Cantilever*: Forces F_1, F_2 are applied at distance h from the upper and lower boundaries. The left boundary is fixed. (c) *Jet engine bracket*. Left: Available design region with six cylindrical interfaces where the shape must attach. Right: A diagonal force is applied at the two central interfaces. The bracket is fixed at the four side interfaces.

3.2 Diversity

The diversity loss, defined in Equation 2, requires the definition of a pairwise dissimilarity measure between shapes. GINNs [8] use boundary dissimilarity, which is easily optimized for SDFs since the value at a point is the distance to the zero level set. However, this dissimilarity measure does not apply to our density field shape representation. Hence, we propose a boundary dissimilarity based on the chamfer discrepancy.

Diversity via differentiable chamfer discrepancy. To define the dissimilarity on the boundaries of a pair of shapes $(\partial\Omega_1, \partial\Omega_2)$, we use the one-sided chamfer discrepancy (CD):

$$
\text{CD}(\partial\Omega_1, \partial\Omega_2) = \frac{1}{|\partial\Omega_1|} \sum_{x \in \partial\Omega_1} \min_{\tilde{x} \in \partial\Omega_2} ||x - \tilde{x}||_2 \tag{4}
$$

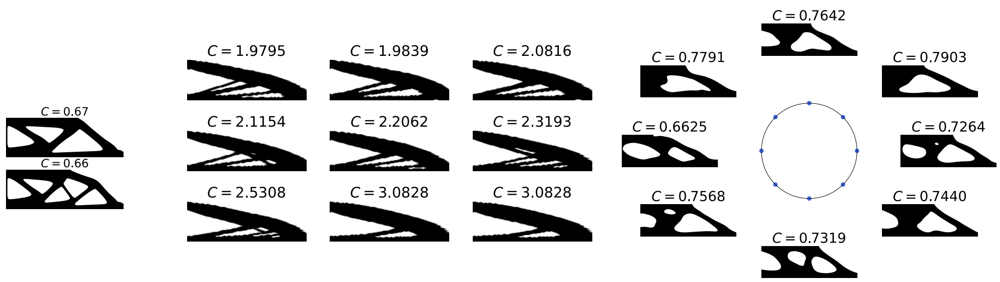
To use the CD as a loss, it must be differentiable w.r.t. the network parameters θ. However, the chamfer discrepancy $\text{CD}(\partial\Omega_1, \partial\Omega_2)$ depends only on the boundary points $x_i \in \partial\Omega$ which only depend on $f_\theta(x_i)$ implicitly. Akin to prior work [11, 7, 27], we apply the chain rule and use the level-set equation to derive

$$
\frac{\partial \text{CD}}{\partial \theta} = \frac{\partial \text{CD}}{\partial x} \frac{\partial x}{\partial y} \frac{\partial y}{\partial \theta} = \frac{\partial \text{CD}}{\partial x} \frac{\nabla_x f_\theta}{|\nabla_x f_\theta|^2} \frac{\partial y}{\partial \theta} \tag{5}
$$

where $y = \rho$ is the density field in our case. We detail this derivation in Appendix C.3. Finding surface points on density fields is substantially harder than for SDFs, as there is no reliable gradient information to exploit for root finding. Therefore, we employ a robust algorithm detailed in Algorithm 2.

Geometric constraints We utilize geometric constraints similar to GINNs, leveraging the continuous field representation to learn finer details, particularly at interfaces. Further details are provided in Table 5 and Appendix B.

4 Experiments



(a) Deflated barrier (b) TopoDiff (c) TOM (ours)

Figure 5: Solutions to the *MBB beam* problem generated by three TO methods.

We solve three standard TO benchmark problems comparing TOM to established classical and data-driven methods. For all our TOM experiments, we employ the WIRE architecture [37], which uses wavelets as activation functions. This imposes an inductive bias towards high frequencies, while being more localized than, e.g., a sine activation [39]. We use a 2-dimensional modulation space and sample vectors uniformly on a circle with radius r. We denote a circle centered at the origin with radius r as the 1-sphere $\mathcal{S}^1(r) = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| = r\}$. This 1D manifold embedded in the 2D modulation space assures that modulation vectors are sufficiently far apart from each other to avoid mode collapse. The radius controls the initial diversity of the shapes and is an important hyperparameter to tune (see also Appendix A.6). At each iteration, the number of shapes M is either 9 or 25, depending on the problem. For further experimental details, see Appendix A.

4.1 Problem definitions

We apply our method to common linear elasticity problems [38, 19] in two and three dimensions, namely the Messerschmitt-Bölkow-Blohm (MBB) beam, the cantilever beam, and the jet engine bracket. Due to the symmetry of the MBB beam, we follow Papadopoulos et al. [32] and optimize only the right half. More details and descriptions are provided in Figure 4 and Appendix D.1.

4.2 Baselines

The experiments compare TOM with state-of-the-art classical and data-driven approaches (see Figure 2). As previously introduced , the standard method for single shape TO is SIMP [6], for which we use the FeniTop implementation [19] – a well-documented Python wrapper of the standard open-source FenicsX FEM solver [3].

Deflated barrier (DB) method [32] is the state-of-the-art method for finding multiple solutions to TO problems. DB is a sequential algorithm that cannot perform multiple solver steps in parallel. For the jet engine bracket, a comparison to DB is omitted, given DB's slow performance, complexity, and hyperparameter sensitivity.

TopoDiff [25] is a state-of-the-art data-driven TO solver. It is a diffusion model trained on 33,000 TO problem-solution pairs, discretized as 64×64 images. Similar to DB, we could not apply TopoDiff to the 3D benchmark problem, since neither a pretrained model nor a sufficiently large 3D TO dataset is publicly available.

4.3 Metrics

Quality. To evaluate structural characteristics, we report compliance C and volume V. We compute them using the open-source numerical library FenicsX [3]. For each problem, we use the same high-resolution mesh across all methods to compute the evaluation metrics. This minimizes discretization errors and prevents numerical artifacts due to the mesh dependency of the solver. Following Mazé and Ahmed [25], we also compute the *load violation* LV : $\Omega \mapsto \{0, 1\}$ for each solution Ω. LV $= 1$ only when the shape lacks material where the load is applied, resulting in high compliance.

Diversity. As a diversity measure, we use the Hill numbers $D_q(S), q \in \mathbb{R}$ over a set S as introduced by Leinster [24]. For $q = 2$, the Hill number corresponds to the expected dissimilarity if two elements of a set are sampled with replacement. As a dissimilarity metric we choose the Wasserstein distance, as it is a mathematically well-defined distance between distributions. In particular, we use the sliced-1 Wasserstein distance approximation, a computationally cheap implementation [14]. Hence, computing the Hill number $D_2(\Omega_{\mathcal{Z}})$ corresponds to computing the expected Wasserstein distance:

$$\mathbb{E}[W_1] := \mathbb{E}_{\mathbf{z}_i, \mathbf{z}_j \sim p(\mathcal{Z})} \left[W_1(\Omega_{\mathbf{z}_i}, \Omega_{\mathbf{z}_j}) \right]. \tag{6}$$

Computational cost. We report the wall clock time in Table 1, to give a sense of the practicality of running each method. Additionally, we further characterize the computational cost of different methods in Table 2.

4.4 Results

Table 1: Results on the three benchmark problems. C is the compliance. LVR is the ratio of load-violating solutions. As these always result in high compliance, results marked with a * report statistics after filtering such solutions. V is the mean ($\pm$ standard deviation) of the volumes as a fraction of the available volume. $E(W_1)$ quantifies the diversity as the expected Wasserstein-1 distance between solutions. N is the number of solutions for methods that produce a finite number of samples. TopoDiff and TOM produce a continuous distribution that can be sampled. Last is the wall-clock time to run the methods. Table 2 contains further computational characterization.

Problem	**Method**	$C \downarrow$	$\min(C)$	$\max(C)$	LVR[%] $\downarrow$	V[%]	$\mathbb{E}(W_1) \uparrow$	$N \uparrow$	Time[min] $\downarrow$
MBB						53.5	$\times 10^{-2}$		
	FeniTop	0.68			0	53.49	0	1	2
	DB	0.67 ± 0.01	0.66	0.67	0	53.49 ± 0.00	1.36	2	55.5
	TopoDiff	2.29 ± 0.37	1.79	3.43	0	51.83 ± 0.59	2.14		
	TOM	0.75 ± 0.03	0.66	0.83	0	55.08 ± 1.04	4.23		6.1
Cantilever			$\times 10^{-2}$			50.00	$\times 10^{-2}$		
	FeniTop	0.59			0	49.91	0	1	4
	DB	0.58 ± 0.01	0.57	0.58	0	49.98 ± 0.00	1.01	2	123
	TopoDiff*	1.19 ± 0.28	0.72	1.95	87	50.23 ± 0.70	2.15		
	TOM*	0.69 ± 0.13	0.64	1.27	8	49.21 ± 0.48	2.42		29
Bracket			$\times 10^{-3}$			7.00	$\times 10^{-2}$		
	FeniTop	0.99			0	7.16	0	1	187
	TOM	1.31 ± 0.01	1.07	1.54	0	6.86 ± 0.09	0.15		47

The quantitative results of our experiments are summarized in Table 1. Qualitatively, we show different results in Figures 1, 5, and 6. Further quantitative and qualitative results are shown in Appendix A.6.

TOM has high quality and diversity. Across all experiments, TOM finds diverse solutions with a near-optimal compliance. DB has better compliance but a substantially lower diversity with a low number of solutions ($N = 2$). TopoDiff solutions are similarly diverse, but have poor compliance. For the cantilever, we filtered LVR $= 87\%$ of TopoDiff outputs due to load violations, whereas for TOM 8%. Surprisingly, the high LVR of TopoDiff contradicts the LVR $= 0\%$ reported by Mazé and Ahmed [25]. We attribute this to the cantilever being out of distribution of TopoDiff's training dataset.

TOM outperforms TopoDiff. At a first glance, the TopoDiff results in Figure 1 (b) may appear visually appealing – the shapes exhibit thin features and horizontal symmetry. However, the quantitative examination reveals that TopoDiff produces shapes with poor compliance and high LVR, as discussed further in Appendix A.5. In contrast, TOM achieves lower compliance, lower LVR, and greater diversity. It is also data-free, eliminating the need for an expensive pretraining dataset. Furthermore, TopoDiff requires around 9 seconds for inference. Although training time is not reported, it is expected to be significantly higher due to the complex architecture, conditioning and data, as well as training across problem instances, limiting the direct comparability of speed.

TopoDiff does not amortize. Despite being trained on 30,000 designs, TopoDiff fails to generate compliant shapes on standard TO benchmark problems. These results align with prior work Woldseth et al. [46], finding large shortcomings of purely data-driven methods.

TOM is more diverse and faster than DB. While the classical DB demonstrates higher compliance, TOM consistently outperforms DB in terms of diversity and wall-clock time, albeit with an increased number of solver steps. This performance advantage is primarily due to TOM's parallel nature, contrasting with the fundamentally sequential algorithm of DB. Additionally, the slower wall-clock time of DB can be partially attributed to its public implementation, which involves mesh refinement during training, thereby increasing the runtime for each solver step.

TOM learns a continuous space of shapes. Due to its modulated neural field architecture, TOM enables exploring different near-optimal solutions by moving continuously in the modulation space.

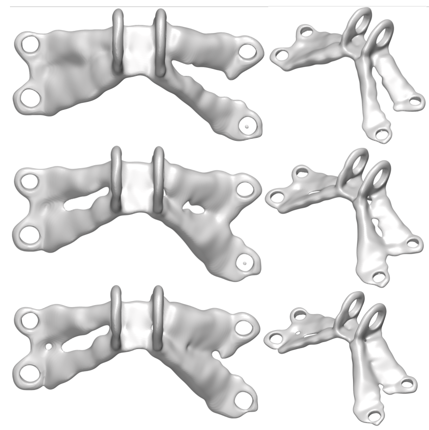


Figure 6: Three different jet engine bracket designs generated by TOM. Left: top view. Right: isometric view. Notably, all three generated designs show a similar low compliance as the single solution by the FeniTop baseline. These explored solutions can be further refined and post-processed by a classical pipeline.

TOM has surface undulations. Qualitatively, we observe, e.g., in Figure 6, that shapes produced by TOM are wavier than DB or TopoDiff and can contain floaters (small disconnected components). Undulations and floaters are a common problem in neural TO approaches [25, 30] and can be corrected with simple post-processing [41], which we demonstrate in Appendix D.3.

Ablations. We perform ablations to distill the impact of the diversity constraint, the radius of the modulation space, the SIMP penalty p, and annealing of the Heaviside filter. We find that all these components are necessary for TOM to work. Additionally, we show results for two alternative diversity metrics and demonstrate that they lead to similar conclusions. Furthermore, we ablate the uniform sampling on the circle with fixed modulation vectors. We find that this variant also leads to satisfactory results. The details are provided in Appendix A.6.

5 Conclusion

This paper presents topology optimization using modulated neural fields (TOM), a novel approach that addresses a limitation in traditional TO methods. By leveraging neural networks to parameterize shapes and enforcing solution diversity as an explicit constraint, TOM enables the exploration of multiple near-optimal designs. This is crucial for industrial applications, where manufacturing or aesthetic constraints often necessitate the selection of alternative designs. Our empirical results demonstrate that TOM is both effective and scalable, generating more diverse solutions than prior methods while adhering to mechanical requirements. This work opens new avenues for generative design approaches that do not rely on large datasets, addressing current limitations in engineering and design.

Limitations and future work. While TOM shows notable progress, several areas warrant further investigation to fully realize its potential. Future research could explore extending the one-dimensional modulation manifold into higher dimensions, allowing multi-axis design exploration. Notably, the mitigation of floaters and undulations requires more in-depth investigation. Additionally, more research is needed to explore different dissimilarity metrics for the diversity loss, as this could further enhance the variety of generated designs. Improving sample efficiency is also crucial for practical applications, as the current method may require significant computational resources. Promising directions include using second-order optimizers [20]. Lastly, future work could train TOM on coarse designs and subsequently refine them with classical TO methods, combining the strengths of both approaches to achieve even better results.

References

[1] Grégoire Allaire, Charles Dapogny, and François Jouve. Shape and topology optimization. In *Handbook of numerical analysis*, volume 22, pages 1–132. Elsevier, 2021.

[2] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. DOLFINx: the next generation FEniCS problem solving environment. *preprint*, 2023. doi: 10.5281/zenodo.10447666.

[4] Shamsulhaq Basir and Inanc Senocak. An adaptive augmented lagrangian method for training physics and equality constrained artificial neural networks. *arXiv preprint arXiv:2306.04904*, 2023.

[5] Shamsulhaq Basir and Inanc Senocak. An adaptive augmented lagrangian method for training physics and equality constrained artificial neural networks. *arXiv preprint arXiv:2306.04904*, 2023.

[6] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988. ISSN 0045-7825. doi: https://doi.org/10.1016/0045-7825(88)90086-2.

[7] Arturs Berzins, Moritz Ibing, and Leif Kobbelt. Neural implicit shape editing using boundary sensitivity. In *The Eleventh International Conference on Learning Representations*. OpenReview.net, 2023.

[8] Arturs Berzins, Andreas Radler, Eric Volkmann, Sebastian Sanokowski, Sepp Hochreiter, and Johannes Brandstetter. Geometry-informed neural networks. *arXiv preprint arXiv:2402.14009*, 2024.

[9] Mariusz Bujny, Nikola Aulig, Markus Olhofer, and Fabian Duddeck. Learning-based topology variation in evolutionary level set topology optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, page 825–832, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356183. doi: 10.1145/3205455.3205528.

[10] Aaditya Chandrasekhar and Krishnan Suresh. TOuNN: Topology optimization using neural networks. *Structural and Multidisciplinary Optimization*, 63(3):1135–1149, Mar 2021. ISSN 1615-1488.

[11] Siqi Chen, Guillaume Charpiat, and Richard J Radke. Converting level set gradients to shape gradients. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V 11*, pages 715–728. Springer, 2010.

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[13] Hao Deng and Albert C To. A parametric level set method for topology optimization based on deep neural network. *Journal of Mechanical Design*, 143(9):091702, 2021.

[14] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[15] Michael Gillhofer, Hubert Ramsauer, Johannes Brandstetter, Bernhard Schäfl, and Sepp Hochreiter. A gan based solver of black-box inverse problems. In *NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks*, 2019.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[17] David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, 2017.

[18] Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. Neural reparameterization improves structural optimization. *arXiv preprint arXiv:1909.04240*, 2019.

[19] Yingqi Jia, Chao Wang, and Xiaojia Shelly Zhang. Fenitop: a simple fenicsx implementation for 2d and 3d topology optimization supporting parallel computing. *Structural and Multidisciplinary Optimization*, 67(8):140, 2024.

[20] Anas Jnini, Flavio Vella, and Marius Zeinhofer. Gauss-newton natural gradient descent for physics-informed computational fluid dynamics. *arXiv preprint arXiv:2402.10680*, 2024.

[21] Kaspar Kiis, Jared Wolfe, Gregg Wilson, David Abbott, and William Carter. Ge jet engine bracket challenge. `https://grabcad.com/challenges/ge-jet-engine-bracket-challenge`, 2013. Accessed: 2024-05-22.

[22] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[23] Boyan Stefanov Lazarov and Ole Sigmund. Filters in topology optimization based on helmholtz-type differential equations. *International journal for numerical methods in engineering*, 86(6): 765–781, 2011.

[24] Tom Leinster. *Entropy and diversity: the axiomatic approach*. Cambridge university press, 2021.

[25] François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[26] I. Mehta, M. Gharbi, C. Barnes, E. Shechtman, R. Ramamoorthi, and M. Chandraker. Modulated periodic activations for generalizable local functional representations. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14194–14203, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society.

[27] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A level set theory for neural implicit evolution under explicit flows. In *European Conference on Computer Vision*, pages 711–729. Springer, 2022.

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[30] Zhenguo Nie, Tong Lin, Haoliang Jiang, and Levent Burak Kara. Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *Journal of Mechanical Design*, 143(3):031715, 2021.

[31] Amin Heyrani Nobari, Giorgio Giannone, Lyle Regenwetter, and Faez Ahmed. Nito: Neural implicit fields for resolution-free topology optimization. *arXiv preprint arXiv:2402.05073*, 2024.

[32] Ioannis PA Papadopoulos, Patrick E Farrell, and Thomas M Surowiec. Computing multiple solutions of topology optimization problems. *SIAM Journal on Scientific Computing*, 43(3): A1555–A1582, 2021.

[33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[35] Daniel Rebain, Mark J. Matthews, Kwang Moo Yi, Gopal Sharma, Dmitry Lagun, and Andrea Tagliasacchi. Attention beats concatenation for conditioning neural fields. *Trans. Mach. Learn. Res.*, 2023, 2022.

[36] Suryanarayanan Manoj Sanu, Alejandro M Aragon, and Miguel A Bessa. Neural topology optimization: the good, the bad, and the ugly. *arXiv preprint arXiv:2407.13954*, 2024.

[37] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[38] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, February 2007. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-006-0087-x. Number: 4-5.

[39] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[40] Ivan Sosnovik and Ivan Oseledets. Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4):215–223, 2019.

[41] Subodh C Subedi, Chaman Singh Verma, and Krishnan Suresh. A review of methods for the geometric post-processing of topology optimized models. *Journal of Computing and Information Science in Engineering*, 20(6):060801, 2020.

[42] Damien Teney, Armand Mihai Nicolicioiu, Valentin Hartmann, and Ehsan Abbasnejad. Neural redshift: Random networks are not random functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4786–4796, 2024.

[43] Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *Advances in Neural Information Processing Systems*, 2020.

[44] Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. TopoGAN: A topology-aware generative adversarial network. In *Proceedings of European Conference on Computer Vision*, 2020.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

[46] Rebekka V Woldseth, Niels Aage, J Andreas Bærentzen, and Ole Sigmund. On the use of artificial neural networks in topology optimisation. *Structural and Multidisciplinary Optimization*, 65(10):294, 2022.

[47] Daniel Yago, Juan Cante, Oriol Lloberas-Valls, and Javier Oliver. Topology optimization methods for 3d structural problems: a comparative study. *Archives of Computational Methods in Engineering*, 29(3):1525–1567, 2022.

[48] Jonas Zehnder, Yue Li, Stelian Coros, and Bernhard Thomaszewski. NTopo: Mesh-free topology optimization using implicit neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10368–10381. Curran Associates, Inc., 2021.

[49] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

A Implementation and experimental details

A.1 Algorithmic description

The TOM algorithm is given in Algorithm 1. A detailed textual description is provided in the main paper.

A.2 Additional results

Table 2 contains additional characterization of the main experimental results in terms of computational resources.

A.3 Model hyperparameters

The most important hyperparameters are summarized in Table 3. The hyperparameters were found by manual search with compliance and diversity as targets. The search was additionally guided by visual inspection and interpretation of the produced shapes.

Algorithm 1 TOM

Input: parameterized density f_θ, vector of mesh points $\mathbf{x_i} \in \mathbb{R}^{d_x}$, β and annealing factor Δ_β, number of shapes per iteration k, learning rate scheduler $\gamma(t)$, iterations T

for $t = 1$ **to** T **do**

- $\mathbf{z_j} \sim p(\mathcal{Z})$ {sample modulation vectors}
- $\tilde{\rho}_{z_j} \leftarrow f_\theta(\mathbf{x_i}, \mathbf{z_j})$ {net forward all shapes j}
- $\rho_j \leftarrow H(\tilde{\rho}_{z_j}, \beta)$ {Heaviside contrast filter}
- $C, V, \frac{\partial C}{\partial \rho}, \frac{\partial V}{\partial \rho} \leftarrow \text{FEM}(\rho_j)$ {solver step}
- $\frac{\partial \rho}{\partial \theta} \leftarrow \text{backward}(f_\theta, \rho_j)$ {autodiff backward}
- $\nabla_\theta C \leftarrow \frac{\partial C}{\partial \rho} \frac{\partial \rho}{\partial \theta}$ {compliance gradient}
- $\nabla_\theta V \leftarrow \lambda_V \frac{\partial C}{\partial \rho} \frac{\partial \rho}{\partial \theta}$ {volume gradient}
- $L_\text{G} \leftarrow \lambda_\text{interface} L_\text{interface} + \ldots$ {GINN constraints}
- $\Delta\theta \leftarrow \text{ALM}\left(\nabla_\theta C, \nabla_\theta V, \nabla_\theta L_\text{G}\right)$ {ALM}
- $\theta \leftarrow \theta - \gamma(t)\Delta\theta$ {parameter update}
- $\beta \leftarrow \beta \Delta_\beta$ {annealing}

end for

Table 2: Additional characterization of the main experimental results reported in Table 1, including iterations needed to converge. Each iteration consists of several FEM solver calls, which are executed in parallel for TOM, resulting in an overall faster training than the sequential DB. Lastly, we report the mesh resolution used by the FEM solver. For DB, it changes over training due to the adaptive mesh refinement, which is indicated by the initial grid $\rightarrow$ final refined mesh size.

Problem	Method	Iterations	Total solver calls	FEM resolution
MBB				
	FeniTop	400	400	180×60
	DB		3190	$50 \times 150 \rightarrow 27174$
	TopoDiff			64×64
	TOM	400	10000	180×60
Cantilever				
	FeniTop	400	400	150×100
	DB		4922	$25 \times 37 \rightarrow 52024$
	TopoDiff*			64×64
	TOM*	1000	25000	150×100
Bracket				
	FeniTop	400	400	$104 \times 172 \times 60$
	TOM	1000	9000	$26 \times 43 \times 15$

Compute. We report additional information on the experiments and their implementation. We run all experiments on a single GPU (NVIDIA Titan V12) on a node with a Xeon Gold 6150 CPU (36 Cores, 2.70GHz) and 384GB RAM. The maximum GPU memory requirements are less than 2GB for all experiments.

Neural network. For the model to effectively learn high-frequency features, it is important to use a neural network represenation with a high frequency bias [39, 42]. Hence, all models were trained using the real, 1D variant of the WIRE architecture [37]. WIRE allows to adjust the frequency bias by setting the hyperparameters ω_0^1 and s_0. For this architecture, each layer consists of 2 Multi-layer perceptron (MLPs), one has a periodic activation function $cos(\omega_0 x)$, the other with a gaussian $e^{(s_0 x)^2}$. The post-activations are then multiplied element-wise.

Optimization. The overall loss for a data point x can be expressed as $L(x) = o(x) + \sum_i \lambda_i c_i(x)$, where $o(\cdot)$ represents the objective function, $c_i(\cdot)$ denotes the constraints, and λ_i are the balancing coefficients for these constraints. We employ the augmented Lagrangian method (ALM) [5] to dynamically balance the various constraints throughout the training process. Intuitively, the better a constraint c_i is satisfied, the smaller the corresponding λ_i. We designate the compliance loss as the objective function, which consequently remains unaffected by the balancing mechanism of ALM. Crucially, for ALM to function optimally, we scale all loss terms by a dedicated factor to ensure they are approximately on the same magnitude. This adjustment is analogous to harmonizing the different "units" of the losses (e.g., volume loss versus diversity loss).

Delayed start of diversity loss for jet engine bracket. For the jet engine bracket, we start the chamfer discrepancy loss at an iteration where the shapes already have reasonable formed (see "start diversity" in Table 3). Empirically, this is needed to not destroy the shapes early on. This could have several potential reasons, including a low target volume of only 0.07 or the fact that 3D shape optimization might be fundamentally harder. We leave further investigations to future work.

Table 3: TOM hyperparameters for different experiments.

	MBB beam	**Cantilever**	**JEB**
Hidden layers	32x3	32x3	64x3
ω_0^1 for WIRE	10	9	18
s_0 for WIRE 10	10	6	
Learning rate	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	10^{-3}
Decay rate	400	200	
radius r	1.2	0.6	0.02
SIMP penalty p	3	3	1.5
β annealing	$[0, 400]$	$[0, 400]$	$[0, 400]$
minimal diversity δ^*	0.3	0.4	0.13
# iterations	400	1,000	1,000
# shapes per batch	25	25	9
Compliance scale	1	1	1,000
Volume scale	1	1	10
Chamfer diversity scale	1	10	10
Interface scale			2,000
Interface normal scale			1
Design region scale			1
Start diversity			350

A.4 Reference solutions

We provide reference solutions to the problem settings generated by the standard TO. We use the implementation of the SIMP method provided by the FeniTop library [19] as the classical TO baseline. We also generate single solutions using TOM without a diversity constraint or modulation variable as input. These single shape training runs showcase the baseline capability of TOM.
The reference solutions for MBB beam problem are shown in Figure 7, for the cantilever beam problem in Figure 8, and for the jet engine bracket problem in Figure 9.

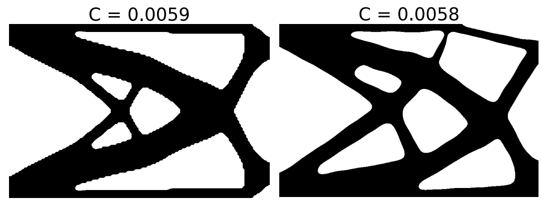
(a) Single solution generated with FeniTop (b) Single solution generated with TOM

Figure 7: Reference solutions for the MBB beam problem

A.5 Load violations in TopoDiff

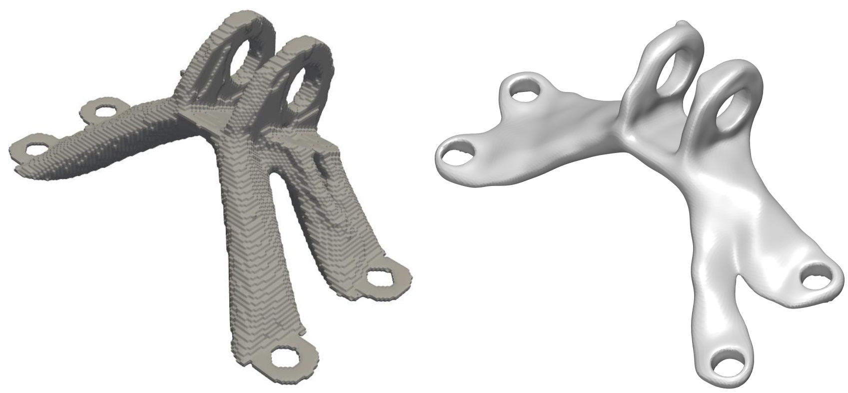
[25] provide a detailed evaluation of TopoDiff, including a test data set (called *level 2 test data*) with unseen boundary conditions. They report that TopoDiff successfully generalizes to unseen boundary conditions and produces shapes which fit these new, unseen boundaries with 100% accuracy (0% load violations, Table 1 in [25]).

In the case of the 2D cantilever problem, we find that 87.2% of the shapes generated by TopoDiff contain load violations. Our FEM solver regularizes the solve step by placing so-called ersatz material



(a) Single solution generated with FeniTop (b) Single solution generated with TOM

Figure 8: Reference solutions for the Cantilever problem



(a) Single solution generated with FeniTop (b) Single solution generated with TOM

Figure 9: Reference solutions for the jet engine bracket (JEB) problem

with a very low (but non-zero) density at void mesh elements. This prevents NaNs or undefined behavior by the solver when encountering violated boundary conditions, such as no material at loading points (load violations), see Figure 10b.

This results in a large jump in the compliance values, which can be seen in Figure 10a. Note that these values are, however, unphysical and an artifact of the solver using ersatz-material instead of void cells. Physically, the compliance is simply undefined as a force cannot be applied when no material is present.

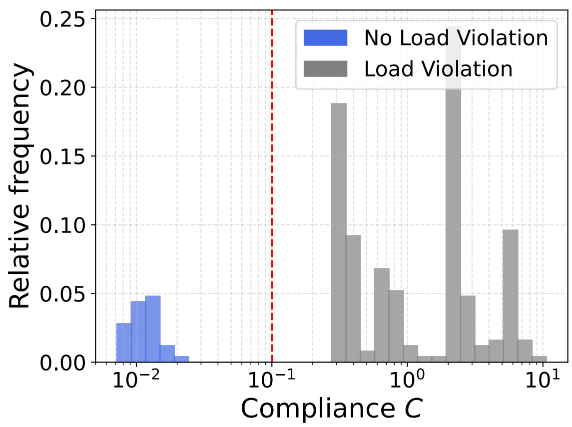
We filter out all shapes containing load violations in the results section to allow for a better comparison and to avoid reporting compliances that are mere solver artifacts and not physically meaningful.

A.6 Ablations

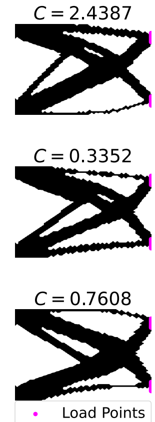
To better understand the impact of different parts of TOM, we perform ablations and summarize the results in Table 4.

Diversity constraint. We ablate the diversity loss term. While some diversity remains in the 2D case due to the network initialization randomness, we observe complete mode collapse in 3D, illustrated in Figure 13.

Radius r of modulation space. We highlight the choice of radius r by ablating it with a radius $r' = \frac{1}{10}r$. The resulting shapes are less diverse (c.f. Table 4) despite applying the diversity constraint during training.
The importance of a network's frequency bias has been highlighted by several works [39, 37]. Importantly, depending on the problem, there might be a different frequency bias necessary for the coordinates $\mathbf{x}$ and the modulation vectors $\mathbf{z}$. For the WIRE architecture we use in our experiments, the frequency bias of the modulation is implicitly controlled by the radius r. The greater r, the greater the diversity at initialization and convergence as suggested by Teney et al. [42].



(a) Distribution of compliance C of TopoDiff shapes. The large jump in compliance above 10^{-1} is explained by forces being applied to the ersatz material.



(b) Load violations of TopoDiff shapes. Shapes do not attach to all load points, leading to artifacts in the computed compliance C.

Table 4: Results for ablations on the three TO problems.

Problem	Method	C	$\min(C)$	$\max(C)$	V[%]	LVR[%]	$\mathbb{E}(W1)$	$\mathbb{E}(d_H)$	$\mathbb{E}(d_{\text{SSIM}})$
MBB Beam									
	Base	0.75 ± 0.03	0.66	0.83	55.08 ± 1.04	0.0	0.042	186.22	0.1096
	No diversity constraint	0.76 ± 0.02	0.73	0.84	54.62 ± 1.07	0.0	0.042	170.10	0.1124
	Radius 1/10	0.85 ± 0.23	0.65	1.43	58.55 ± 2.63	0.0	0.040	93.29	0.1075
	Penalty $p = 1.5$	1.56 ± 1.31	0.68	5.54	58.17 ± 2.93	0.0	0.043	169.28	0.1299
	Annealing	0.89 ± 0.25	0.67	1.66	59.75 ± 1.70	0.0	0.028	221.58	0.0917
	Fixed modulation	0.71 ± 0.01	0.70	0.76	53.49 ± 0.32	0.0	0.022	197.55	0.0784
Cantilever									
	Base	0.69 ± 0.13	0.64	1.28	49.21 ± 0.48	8.0	0.024	85.33	0.1799
	No diversity constraint	0.90 ± 0.88	0.62	4.95	49.63 ± 0.61	8.0	0.023	89.25	0.1791
	Radius 1/10	0.92 ± 0.07	0.79	1.01	49.43 ± 2.08	0.0	0.032	84.02	0.1186
	Penalty $p = 1.5$	1.19 ± 1.21	0.54	4.97	50.55 ± 1.15	12.0	0.019	69.14	0.1565
	Annealing	0.74 ± 0.22	0.61	1.35	49.53 ± 0.33	12.0	0.027	84.42	0.1767
	Fixed modulation	0.82 ± 0.26	0.62	1.25	48.87 ± 0.28	4.0	0.026	93.93	0.1848
Simjeb									
	Base	0.13 ± 0.01	0.11	0.15	6.86 ± 0.09	0.0	0.027	9.01	0.0782
	No diversity constraint	0.11 ± 0.01	0.10	0.14	7.51 ± 0.06	0.0	0.006	8.2	0.0687
	Radius 1/10	0.13 ± 0.01	0.12	0.15	7.35 ± 0.02	0.0	0.002	5.54	0.0093
	Penalty $p = 1.5$	0.09 ± 0.00	0.09	0.10	7.09 ± 0.01	0.0	0.010	11.33	0.0443
	Annealing	0.60 ± 0.19	0.40	0.85	6.21 ± 0.05	66.7	0.041	9.47	0.1024
	Fixed modulation	0.08 ± 0.01	0.07	0.09	7.64 ± 0.04	0.0	0.034	11.01	0.0970

Sensitivity to penalty p. As noted by prior work [36], neural reparemeterization is sensitive to the penalty parameter p. For SIMP, the density is penalized with an exponent p. We ablate this parameter by setting the 2D penalty from $p = 3$ to 1.5 and for 3D from $p = 1.5$ to 3. Our experiments confirm that TOM is sensitive to the choice of p, as it sharpens the loss landscape. This becomes more apparent when looking at the derivative of the stiffness of a mesh element $\frac{\partial \mathbf{K}_i(\rho_i)}{\partial \rho_i} = p\rho_i^{p-1}\mathbf{K}_i$. E.g., for $p = 3$ the gradient is quadratically scaled by the current density value. This implies that the higher p, the harder it is to escape local minima. We do not observe a large performance impact in the 2D experiments. However, in 3D, we observe that $p = 3$ instead of $p = 1.5$ leads to convergence to an undesired local minimum, illustrated in Figure 12a.

Annealing is necessary for good convergence. We find that TOM requires annealing to achieve good convergence. Figure 12b demonstrates a failed run without β scheduling the Heaviside function (Equation 14). The optimization fails to converge to a useful compliance value and does not fulfill the desired interface constraints.

Fixed modulation vectors. TOM uses a continuous modulation space, and in our current implementation, samples modulation vectors from a circle. An interesting alternative is to keep the modulation

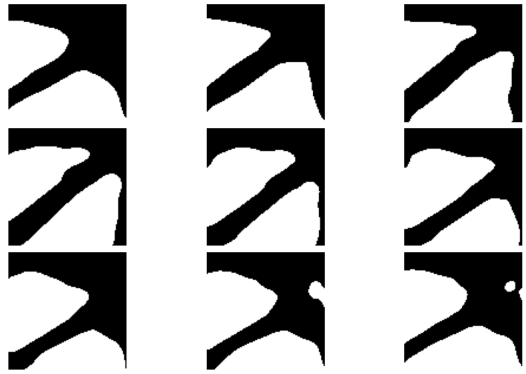


Figure 11: The cantilever ablation of the radius leads to mode collapse qualitatively.

vectors fixed throughout training. As shown in Table 4, the results of this variant also produce shapes with satisfactory compliance and diversity.

Alternative diversity metrics We measure diversity via the Hill number $D_q(S)$ over the set S, where we choose $q = 2$ as it is interpretable as the expected dissimilarity between two elements drawn from S. The diversity can be computed based on an arbitrary pairwise dissimilarity function. As a measure of dissimilarity, we chose the Wasserstein distance because we consider material distributions, and the Wasserstein distance is a distance between distributions. Also, it is a proper metric distance satisfying the four axiomatic properties.
In addition to the Wasserstein-1 distance, we performed additional evaluations of the diversity based on two other dissimilarity functions: the Hausdorff distance (d_H) and structural dissimilarity (d_{SSIM}) [45]. The results in Table 4 show similar results across all dissimilarity functions. Note that the Hausdorff distance is particularly sensitive to outliers, making it less suitable for robust measurements. Additionally, we observe that the proposed diversity metrics do not fully capture our intuition about the variety of shapes. For instance, the radius-ablation variant of the cantilever exhibits a higher diversity score than the base model. However, the qualitative evaluation in Figure 11 reveals mode collapse. These shapes also exhibit poor compliance. Sanu et al. [36] demonstrated that neural networks can more effectively explore the solution space of topology optimization (TO) problems. We therefore hypothesize that inducing higher diversity, either at initialization or as a constraint, may promote further exploration.

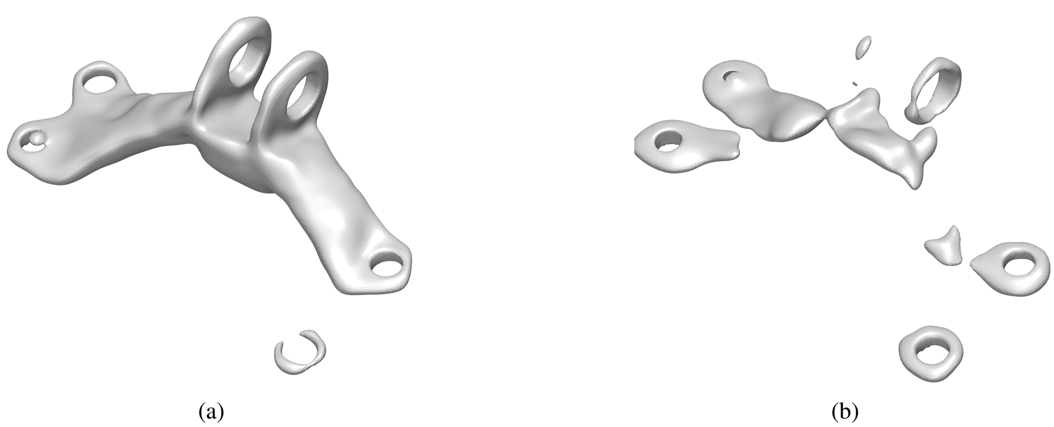


Figure 12: Ablations for penalty and β annealing. (a) Jet engine bracket trained with penalty $p = 3$. (b) Jet engine bracket trained without annealing.

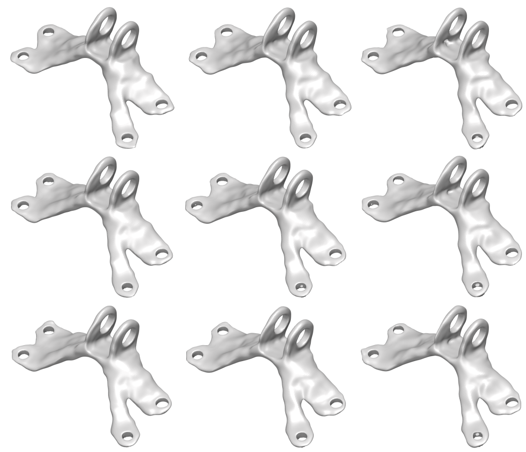


Figure 13: JEB: Training in 3D without diversity constraint results in mode collapse.

Table 5: Geometric constraints are derived from GINNs. The shape Ω and its boundary $\partial\Omega$ are implicitly defined by the level set τ of the function f. The shape must reside within the *design region* $\mathcal{E} \subseteq \mathcal{X}$ and adhere to the *interface* $\mathcal{I} \subset \mathcal{E}$ with a specified *normal* $\bar{n}(x)$.

	Set constraint $c_i(\Omega)$	Function constraint	Constraint violation $c_i(f)$
Design region	$\Omega \subset \mathcal{E}$	$f(x) < \tau \; \forall x \notin \mathcal{E}$	$\int_{\mathcal{X} \setminus \mathcal{E}} [\max(0, f(x) - \tau)]^2 \, dx$
Interface	$\partial\Omega \supset \mathcal{I}$	$f(x) = \tau \; \forall x \in \mathcal{I}$	$\int_{\mathcal{I}} [f(x) - \tau]^2 \, dx$
Prescribed normal	$n(x) = \bar{n}(x) \; \forall x \in \mathcal{I}$	$\frac{\nabla f(x)}{\|\|\nabla f(x)\|\|} = \bar{n}(x) \; \forall x \in \mathcal{I}$	$\int_{\mathcal{I}} \left[\frac{\nabla f(x)}{\|\|\nabla f(x)\|\|} - \bar{n}(x) \right]^2 dx$

B Geometric constraints

We formulate geometric constraints analogously to GINNs in Table 5 for density representations. There are two important differences when changing the shape representation from a signed distance function (SDF) to a density function.

First, the level set $\partial\Omega_\tau$ changes from $\partial\Omega_\tau = \{x \in \mathbb{R}^{d_x} | f_\theta = 0\}$ to $\partial\Omega_\tau = \{x \in \mathbb{R}^{d_x} | f_\theta = 0.5\}$. Second, for a SDF the shape Ω_{tau} is defined as the sub-level set $\Omega_\tau = \{x \in \mathbb{R}^{d_x} | f_\theta \leq 0\}$, whereas for a density it is the super-level set $\Omega_\tau = \{x \in \mathbb{R}^{d_x} | f_\theta \geq 0\}$.

C Diversity

C.1 Diversity on the volume

As noted by Berzins et al. [8], one can define a dissimilarity loss as the L^p function distance

$$d(\Omega_i, \Omega_j) = \sqrt[p]{\int_{\mathcal{X}} (f_i(x) - f_j(x))^p dx} \tag{7}$$

We choose L^1, to not overemphasize large differences in function values. Additionally, we show in the next paragraph that for L^1 and for the extreme case where $f(x) \in \{0, 1\}$ this is equal to the Union minus Intersection of the shapes:

$$\int_{\mathcal{X}} |f_i(x) - f_j(x)| dx = \text{Vol}(\Omega_i \cup \Omega_j) - \text{Vol}(\Omega_i \cap \Omega_j) \tag{8}$$

C.2 L^1 distance on neural fields resembles Union minus Intersection

To derive that the distance metric

$$d(\Omega_i, \Omega_j) = \sqrt[p]{\int_{\mathcal{X}} (f_i(x) - f_j(x))^p dx} \tag{9}$$

for $p = 1$ and $f_i, f_j \in \{0, 1\}$ corresponds to the union minus the intersection of the shapes, we consider the following cases:

$f_i(x)$	$f_j(x)$	$\lvert f_i(x) - f_j(x)\rvert$
1	1	0
1	0	1
0	1	1
0	0	0

Table 6: Function distance if the function only has binary values.

From the table, we observe that the integrand $|f_i(x) - f_j(x)|$ is 1 when x belongs to one shape but not the other, and 0 when x belongs to both or neither. Thus, the integral $\int_{\mathcal{X}} |f_i(x) - f_j(x)|dx$ sums the volumes where x is in one shape but not the other, which is precisely the volume of the union of Ω_i and Ω_j minus the volume of their intersection. It follows that

$$d(\Omega_i, \Omega_j) = \int_{\mathcal{X}} |f_i(x) - f_j(x)|dx = \text{Vol}(\Omega_i \cup \Omega_j) - \text{Vol}(\Omega_i \cap \Omega_j) \tag{10}$$

.

C.3 Diversity on the boundary via differentiable chamfer discrepancy

We continue from Equation 5:

$$\begin{aligned} \frac{\partial L}{\partial \theta} &= \frac{\partial L}{\partial x} \frac{\partial x}{\partial y} \frac{\partial y}{\partial \theta} \\ &= \frac{\partial L}{\partial x} \frac{\nabla_x f_\theta}{|\nabla_x f_\theta|^2} \frac{\partial y}{\partial \theta} \end{aligned} \tag{11}$$

The center term $\frac{\nabla_x f_\theta}{|\nabla_x f_\theta|^2}$ and the last term $\frac{\partial y}{\partial \theta}$ can be obtained via automatic differentiation. For the first term, we derive $\frac{\partial L}{\partial x}$, where L is the one-sided chamfer discrepancy $\text{CD}(\partial\Omega_1, \partial\Omega_2)$.

$$\frac{\partial}{\partial x} \text{CD}(\partial\Omega_1, \partial\Omega_2) = \frac{\partial}{\partial x} \frac{1}{|\partial\Omega_1|} \sum_{x \in \partial\Omega_1} \min_{y \in \partial\Omega_2} ||x - y||_2 \tag{12}$$

$$= \frac{1}{|\partial\Omega_1|} \min_{y \in \partial\Omega_2} \frac{x - y}{\|x - y\|} \tag{13}$$

This completes the terms in the chain rule.

C.4 Finding surface points

We detail the algorithm to locate boundary points of implicit shapes defined by a neural density field, Algorithm 2.
On a high level, the algorithm first identifies points inside the boundary where neighboring points lie on opposite sides of the level set. Subsequently, it employs binary search to refine these boundary points. The process involves evaluating the neural network to determine the signed distance or density values, which are then used to iteratively narrow down the boundary points. We find that 10 binary steps suffice to reach the boundary sufficiently close.

For the 3D jet engine bracket, we additionally follow Berzins et al. [8] and exclude surface points which are within some ϵ region of the interface.

Algorithm 2 Find Boundary Points with Binary Search

Input: regular point grid x_i, level set l, neural density field f_θ, binary search steps T
Find points $x_i^{\text{in}} \in x_i$ for which $f(x_i^{\text{in}}) > l$
Find neighbor points $(x_i^{\text{out}})_i \in x_i$ to x_i^{in} $f(x_{\text{out}}) < l$ { for 2D/3D we check 4/6 neighbors respectively}
Refine pairs $(x_{\text{in}}, x_{\text{out}})_i$ via T binary search steps
Return $((x_{\text{in}} + x_{\text{out}})/2)_i$

D Topology optimization

D.1 Problem definitions

Messerschmitt-Bölkow-Blohm beam (2D). The MBB beam is a common benchmark problem in TO and is depicted in Figure 4a. The problem describes a beam fixed on the lower right and left edges with a vertical force F applied at the center. As the problem is symmetric around $x = L/2$, we follow Papadopoulos et al. [32] and only optimize the right half.
The concrete dimensions of the beam are $H = 1, L = 6$, and the force points downwards with $F = 1$.

Cantilever beam (2D). The cantilever beam is illustrated in Figure 4b. The problem describes a beam fixed on the left-hand side, and two forces F_1, F_2 are applied on the right-hand side.
The dimensions are $H = 1, L = 1.5, h = 0.1$ and the forces $F_1 = F_2 = 0.5$.

Jet engine bracket (3D). We apply TOM to a challenging 3D task, namely the optimization of a jet engine bracket as defined by Kiis et al. [21]. The design region for this problem is enclosed by a freeform surface mesh (see Figure 4c). The load case is depicted in Figure 4c in which a diagonal force pulls in positive x and positive z direction.

D.2 Smoothing and contrast filtering

Filtering is a general concept in topology optimization that aims to reduce artifacts and improve convergence. *Helmholtz PDE filtering* is a smoothing filter similar to Gaussian blurring, but easier to integrate with existing finite element solvers. By solving a Helmholtz PDE, the material density ρ is smoothened to prevent checkerboard patterns, which is typical for TO [23]. *Heaviside filtering* is a type of contrast filter, which enhances the distinction between solid and void regions. The Heaviside filter function, as defined in Equation 14, equals the sigmoid function up to scaling of the input. β is a parameter controlling the sharpness of the transform, similar to the inverse temperature of a classical softmax (higher beta means a closer approximation to a true Heaviside step function). Note that in contrast to the Helmholtz PDE filter, the Heaviside filter is not volume preserving. Therefore, the volume constraint has to be applied to the modified output.

$$H(x, \beta) = 0.5 + \frac{\tanh(\beta(x - 0.5))}{2\tanh(0.5\beta)} \tag{14}$$

Annealing is employed to make the continuous relaxation closer to the underlying discrete problem [22], enhancing the effectiveness of gradient-based optimization methods. Annealing gradually adjusts the sharpness of a function during the optimization. For TO, this is often done by gradually increasing the penalty p or by scheduling a sharpness filter. A common choice is the Heaviside filter as defined in Equation 14.

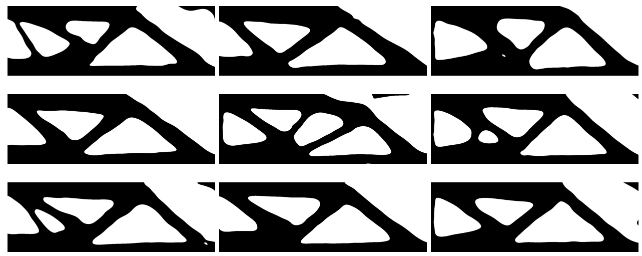
D.3 Post-processing

In general, the post-processing of TO solutions heavily depends on the downstream task and the specific manufacturing requirements, e.g., additive vs subtractive manufacturing. Here, we demonstrate two simple methods to post-process TOM solutions:

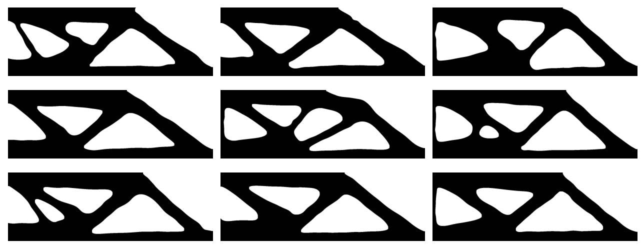
Method A is a simple yet standard image-based approach that removes disconnected components (floaters) and performs morphological closing to remove small artifacts (holes).

Method B is a fine-tuning approach that performs a few iterations of classical TO updates (5% of a full run, learning rate reduced by $\times 10$ compared to the standard FeniTop settings) on the TOM output. This approach also removes the artifacts. This illustrates how TOM extends, not competes, with classical TO.

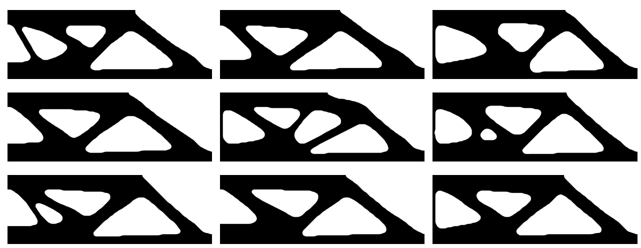
We depict a version of an MBB beam with many floaters and its post-processed version in Figure 14.



(a) Output of TOM model.



(b) Method **A**: Simple post-processing using OpenCV to remove artifacts.



(c) Method **B**: Finetune by running a few FeniTop iterations to convergence (5% of total iterations).

Figure 14: **Post-processing:** Showcasing 2 possible post-processing steps to remove artifacts from the TOM solutions.