Towards Domain-Adaptive, Resolution-Free 3D Topology Optimization with Neural Implicit Fields

Amin Heyrani Nobari,[1*†] Lyle Regenwetter,[1*] Faez Ahmed[1]
[1]Massachusetts Institute of Technology, Cambridge, MA

Abstract

Topology optimization is a ubiquitous task in engineering design, involving the optimal distribution of material in a prescribed spatial domain. Recently, data-driven methods such as deep generative AI models have been proposed as an alternative to iterative optimization methods. However, existing data-driven approaches are often trained on datasets using fixed grid resolutions and domain shapes, reducing their applicability to different resolutions or different domain shapes. In this paper, we introduce two key innovations — a fast TO solver and a neural implicit field architecture to address these limitations. First, we introduce a fast, parallelizable, iterative GPU-based TO solver optimized for high-throughput dataset generation for 3D unstructured meshes. Our solver generated 122K optimized 3D topologies, an order of magnitude more than the largest existing public dataset. Second, we introduce a new resolution-free data-driven method for 3D topologies using neural fields, called NITO-3D. A single NITO-3D model trains and predicts for a variety of resolutions and aspect ratios. By also eliminating the need for computationally intensive physical field conditioning, NITO-3D offers a faster, more flexible alternative for 3D topology optimization. On average, NITO-3D generates topologies roughly 2000 times faster and with only 0.3% higher compliance than state-of-the-art iterative solvers. With 10 steps of iterative fine-tuning, NITO-3D is on average 15 times faster and generates topologies that are under 0.1% more compliant than SIMP's. We open-source all data and code associated with this work at https://github.com/ahnobari/NITO_Public.

1 Introduction

Topology optimization (TO) is a computational method used to determine the most efficient material distribution within a given design space to meet specific performance criteria under imposed constraints. TO has been explored for different types of objectives and constraints, however, the most common type of TO involves optimally distributing material within a domain to minimize compliance, which is often referred to as the minimum compliance problem

The Rise of Deep Generative Models in Engineering: The surge of recent advancements in deep generative AI models (DGMs) for vision and language [16, 8, 25, 44, 18, 9, 39] has motivated the use of deep learning in engineering optimization tasks, including TO. These advancements have already revolutionized our ability to handle multimodal and unstructured data, leading to novel generative approaches for such data [40, 34, 7, 41]. This growing prevalence of DGMs has paved the way for extensive research in engineering and design [43, 48, 26, 53] and now constitutes an important area of research in computational design and engineering.

Bridging Traditional and Deep Learning Approaches in TO: Given these advancements, the integration of machine learning, especially DGMs, into topology optimization, has been a focus of recent research, offering a new paradigm to tackle TO problems [43]. For conciseness, we will refer to such deep generative topology optimization models as "DGTOMs". DGTOMs, which are trained on optimized topologies and conditioned on various loads and boundary conditions, promise substantial time efficiency over traditional TO solvers like SIMP, providing a spectrum of near-optimal solutions [35, 32, 23, 22, 26]. This research has shown that the potential for time and cost reduction through DGTOMs is considerable. It is crucial to recognize that while DGTOMs and the integra-

*Equal contribution.
†Corresponding author: ahnobari@mit.edu

tion of deep learning into TO have shown promising advancements, there has been a range of feedback and critique from the research community

Challenges with DGTOMs: Despite their speed, current DGTOMs face challenges in precision, data dependency, and generalizability, which are discussed below. **Precision:** The most prevalent critique of DGTOMs has been their precision: They typically generate topologies that are less optimal than SIMP and are more likely to violate constraints like volume fraction. This limitation stems from their primary focus on density estimation, largely overlooking the physical aspects of the problems. Fortunately, recent DGTOMs have become significantly more precise, in part by integrating physics into the models in different ways, such as through physical field conditioning [32, 35, 26]. However, recent models

Data dependency: DGTOMs require significant amounts of data to train. Recent works [32, 22, 23, 35, 26] have trained on tens of thousands of optimized topologies, each of which was computed using an iterative TO optimizer, namely SIMP. This necessary step has historically prohibited researchers from applying DGTOMs to larger 2D TO problems or, particularly, 3D TO problems. The few DGTOM papers that have considered high-dimensional 2D or 3D topologies only consider a handful of boundary conditions in their datasets, possibly due to the limited data generation throughput

Generalizability: Since datasets are limited, and the space of possible TO problems is infinite, DGTOMs must learn to generalize to new problems. Whereas iterative TO is easily applied in a variety of domain shapes and resolutions, current DGMs are typically only able to perform well on test cases that have the same domain and resolution as their training data. This is often due to the choice of a representation scheme, where most existing DGTOM approaches rely on discretizing the physical domain into a grid, limiting their adaptability to varying resolutions or domain shapes [32, 22, 23, 35, 26]. Fundamentally, this dependence is ingrained in the convolutional neural network (CNN) architectures used by these models, which provide strong spatial learning capabilities, but restrict their generalizability. Attempts to generalize CNNs across domains have still required retraining on hundreds or thousands of new topologies to succeed in the new domain

3D DGTOMs: Both data dependency and generalizability concerns are significantly amplified in 3D TO compared to 2D. Data generation is orders of magnitude more costly, making data much more scarce. The combinatorial increase in boundary condition configurations and aspect ratios also makes the generalization of data-driven models much more difficult. In this work, we embrace the challenges of 3D TO to better address each of these prohibitive challenges with DGTOMs and provide a framework for 3D TO both for data generation and deep learning.

Addressing DGTOMs Challenges: In this paper, we take significant strides to address the data dependency and generalizability challenges with DGTOMs for challenging 3D topologies. To tackle the data dependency bottleneck, we introduce a new SIMP-based TO library in Python specialized for high-throughput data generation (up to 5x faster than older implementations such as Topy

- We introduce a new TO solver that leverages parallel computing to multiply dataset generation throughput. In under two days, our solver generated a dataset of 106K 3D topologies, an order of magnitude larger than any public 3D TO dataset.
- We release the largest public dataset of 3D topologies, featuring 210 topology configurations spanning numerous aspect ratios, resolutions, and boundary conditions.
- We introduce the first DGTOM for 3D topologies that can train and generate on multi-resolution, mixed aspect-ratio, and unstructured domains.
- We show that NITO-3D generates topologies nearly 2000x faster and with only 0.3% higher compliance than SIMP (median). With 10 steps of refinement, NITO's median topologies are only 0.08% more compliant and are still 15x faster to generate, compared to SIMP.

2 Background & Related Works

This section delves into the background of topology optimization and neural implicit fields. We also provide an overview of existing DGTOMs.

2.1 Structural Topology Optimization

Topology optimization (TO) is a computational technique that determines the optimal material distribution within a given set of constraints, boundary conditions, and loads, often with the objective of minimizing compliance in structural scenarios (see Fig. 1)

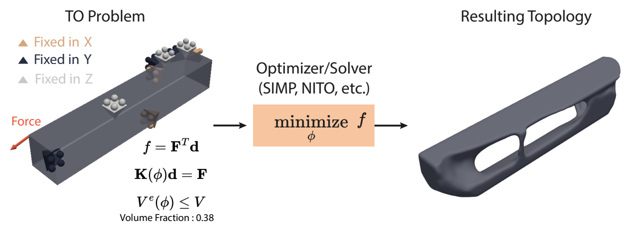

Figure 1: Overview of 3D Topology Optimization: This figure illustrates the essential components of TO, including the domain, boundary conditions, loads, and volume fraction. The objective of TO is to identify the optimal design variables, denoted as ϕ, that enhance prescribed performance objectives such as minimizing compliance f, while adhering to all specified constraints and maintaining static equilibrium

2.1.1 Deep Learning for Topology Optimization

While deep learning techniques have revolutionized the vision and language domains, their full potential in engineering applications, particularly in TO, is still being realized. These methods have been applied across a spectrum of engineering tasks: direct design [1, 2, 6, 31, 51], post-processing [24, 56], and acceleration [4, 28, 29, 49, 55] of optimization processes, sensitivity analysis [3, 37, 5, 45], super-resolution [20, 33, 57], and many others [15, 11, 12, 17]. Particularly relevant to this work are conditional deep generative models that perform the topology optimization task end-to-end. We discuss such DGMs in more detail in the following section. The evolving landscape of deep learning in TO is well-summarized in the critique by Woldseth et. al. [52] and the review by Shin et. al. [47], providing a comprehensive overview of current methodologies and their implications.

2.2 Deep Generative Models for Topology Optimization

Deep Generative Models have seen increasing utilization in design. DGMs have been leveraged for material and molecular discovery

The challenges of traditional optimization methods have spurred a surge in research leveraging DGMs for TO. Key to our study are deep generative topology optimization methods (DGTOMs) that offer an end-to-end solution, accepting constraints and boundary conditions to deliver near-optimal topologies aimed at compliance minimization. Many of these approaches have leveraged either generative adversarial networks (GANs) or diffusion models, though other types of DGMs have also been explored [21, 50]

2.2.1 GAN-based DGTOMs

The generative adversarial network was one of the first popular DGM frameworks used for TO [36, 46, 38, 42]. For instance, Yu et. al

2.2.2 Diffusion-based DGTOMs

Recent advancements have seen diffusion models outperform GANs in topology optimization [32, 22, 23]. Mazé & Ahmed

2.3 Topology Generation using Neural Fields

In this section, we introduce neural fields, discuss their application to TO, and present previous work using neural fields for TO.

2.3.1 Neural Fields

Neural fields are fields that are parameterized by a neural network. These networks typically take spatial coordinates $\mathbf{x} \in \mathbb{R}^n$ as input, then output field values $\Phi(\mathbf{x}) \in \mathbb{R}^m$, encapsulated as $\tilde{\Phi}(\mathbf{x}) = f_\Theta(\mathbf{x})$ where f_Θ is the neural function parameterized by Θ [54]. Neural fields have been applied across various domains including audio, images, videos, and 3D representations. Since TO can be regarded as the generation of an optimal density field across space, neural fields can be used to represent topologies. In fact, neural implicit representations have been directly optimized in a gradient-based approach to identify the optimal topology for any given problem [58, 27, 13, 11, 10]. These works demonstrate that implicit neural fields are capable representations for topologies

2.3.2 Neural Fields for TO

Recent studies have explored implicit neural fields for topology creation, with Hu et. al.

2.4 Datasets and Data-driven Solvers for 3D TO

3D topology optimization presents significant challenges over its 2D counterpart due to higher dimensionality and computational demands, making both iterative solving and dataset generation for DGMs notably more complex and costly. The vast constraint space further complicates DGMs' generalizability for 3D data. Limited data-driven methods have been explored for 3D TO; notable attempts include Behzadi & Ilies [6], who train a CNN, swapping components and fine-tuning the model to switch between the nine solution domains and resolutions tested. Keshavarzzadeh et. al. [30] train a deep disjunctive normal shape model for topologies. Finally, Dittmer et. al. [19] use equivariant neural networks to generate topologies. However, these models need to be retrained to solve any problem with a different resolution. These models provide a baseline for the size and diversity of existing 3D TO datasets. Statistics on the corresponding datasets used are included in Table 1, for easy comparison to our own.

3 Methodology

In this section, we go into the details of our approach and discuss some of the details of our solver and dataset. We then discuss NITO-3D, focusing on its resolution-free, domain-agnostic features.

3.1 Dataset Generation & Solver

As discussed, data-generation throughput is a critical limitation for DGTOMs. To address this, we introduce a fast iterative TO solver customized for dataset generation and release the largest public dataset of optimized 3D topologies. In this subsection, we discuss the features of our solver and dataset.

3.1.1 Solver

Many iterative TO solvers are publicly available

Given the constraints of current TO libraries, we developed a new TO library. In doing so, we carefully considered the existing state-of-the-art in iterative TO

Finite Element Analysis (FEA) For Linear Elasticity: Iterative TO solvers rely on Finite Element Analysis (FEA) solvers to solve the linear elasticity equation for different problems. The first step in FEA is to discretize a physical domain, which is done by making meshes that describe a given domain in discrete volumetric or surface meshes for 3D and 2D respectively. Commonly, structured meshes comprising square quadrilaterals in 2D and voxel hexahedrals in 3D are employed, as noted in most publicly available codes

While structured meshes benefit from computational efficiencies due to their regularity, our goal to generate diverse datasets necessitates accommodating arbitrary, unstructured meshes, making stiffness matrix assembly more challenging. To streamline this process for unstructured meshes, we have developed a method to vectorize the assembly of the stiffness matrix. We precompute sparse assembly kernels that map the stiffness contributions of each element to the overall matrix based on element densities, creating a sparse kernel matrix sized $N_{non-zero}$ by $N_{elements}$. This approach allows for efficient matrix multiplication that aggregates these contributions into the comprehensive stiffness matrix K, all while maintaining sparse matrix efficiency. We apply a similar strategy for other density-dependent calculations, like adjoint gradient updates, enhancing overall optimization speed. Detailed explanations and kernel construction are documented in our code.

After assembling the stiffness matrices, the subsequent task involves solving the resultant large sparse linear system. We primarily employ a direct sparse solver executing Cholesky decomposition, capitalizing on the guaranteed symmetric positive definite nature of stiffness matrices. This method proves significantly more efficient than LU decomposition utilized by earlier Python solvers, such as Topy. For exceedingly large systems where Cholesky decomposition's performance diminishes, we switch to the conjugate gradient (CG) method. Notably, our solver introduces, for the first time to our knowledge, GPU acceleration for the CG method, achieving substantial speed enhancements for large-scale problems. These advancements are integrated into our solver, accessible through a user-friendly Python API, facilitating TO applications on both structured and unstructured meshes across large domains. This solver is made publicly available with a simple Python API which allows for easy application of TO for both structured and unstructured meshes and for very large domains, which we release as part of our solver.

3.1.2 Randomization Process and Dataset

Besides the aforementioned accelerated solver, we also include code for random parallelized TO data generation on both CPU and GPU This is done by a configuration randomizer that creates randomized load cases, boundary configurations, and resolutions for TO problems. In creating our dataset, we select a discrete set of 210 resolution, domain, and boundary condition combinations, while leaving the load position and direction random across the entire domain surface. More details are included in Appendix **??**. Some samples from the dataset are displayed in Fig. 2, which shows several of the dataset's different domain shapes and boundary condition combinations. In total, the dataset includes 106,425 3D topologies with different domain shapes, resolutions, and boundary conditions.

Table 1: Comparison to existing datasets of optimized 3D topologies and several recent public 2D datasets. We consider the number of topologies included, the minimum and maximum number of elements per topology, the total elements across all topologies, the number of unique support configurations, and the number of unique aspect ratios in the dataset. Our dataset features an order of magnitude more topologies, configurations and total elements than existing 3D TO datasets.

	Number of Topologies	Minimum Elements	Maximum Elements	Total Elements	Support Configs	Aspect Ratios	Domain Dimensionality
Topodiff	33K	4.1K	4.1K	122M	42	1	2D
TopologyGAN	49K	8.2K	8.2K	402M	42	1	2D
DOM	60K	66K	66K	3.9B	42	1	2D
Keshavarzzadeh et. al. [30]	2.0K	0.4K	3.2K	3.6M	1	1	3D
Behzadi & Ilies [6]	2.2K	8.0K	**128K**	75M	9	5	3D
Dittmer et. al. [19]	9.8K	6.1K	32K	315M	11	2	3D
NITO-3D (ours)	**122K**	**32K**	48K	**4.3B**	**210**	**7**	3D

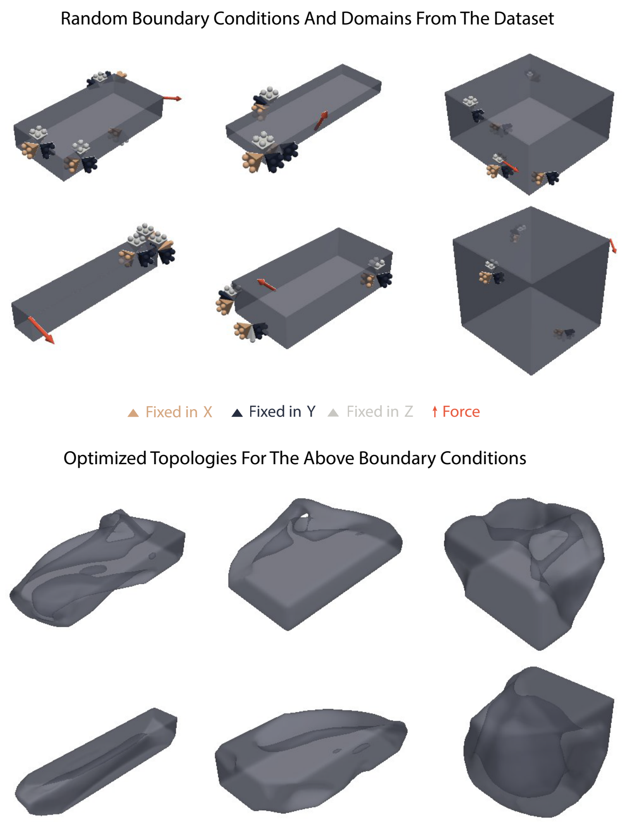

Figure 2: Random samples from the dataset demonstrating the diversity of different domain shapes and varying boundary conditions and loads that are included in the dataset. Aside from the domain shapes, different resolutions also apply to these domains which may not be visible here. The boundary conditions are displayed above and the corresponding SIMP solution is displayed below it.

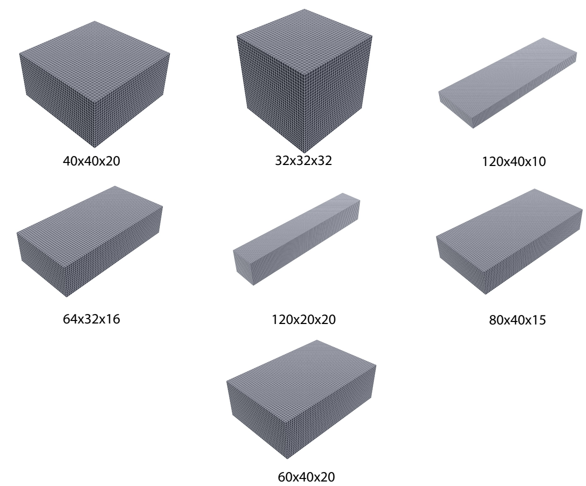

Figure 3: The 7 different domain shapes and resolutions used in the dataset. This shows that our dataset covers both different shapes and different resolutions.

3.2 Introduction to Hybrid 3D Neural Implicit Topology Optimization

In this paper, we introduce a hybrid scheme to accelerate topology optimization by combining deep learning with optimization, which retains the precision of optimization schemes while accelerating optimization using deep learning. Our framework is designed to expedite topology generation by removing the need for iterative sampling, achieving linear scalability with the number of sampled points. In this section, we detail how we achieve this by describing different aspects of our proposed model.

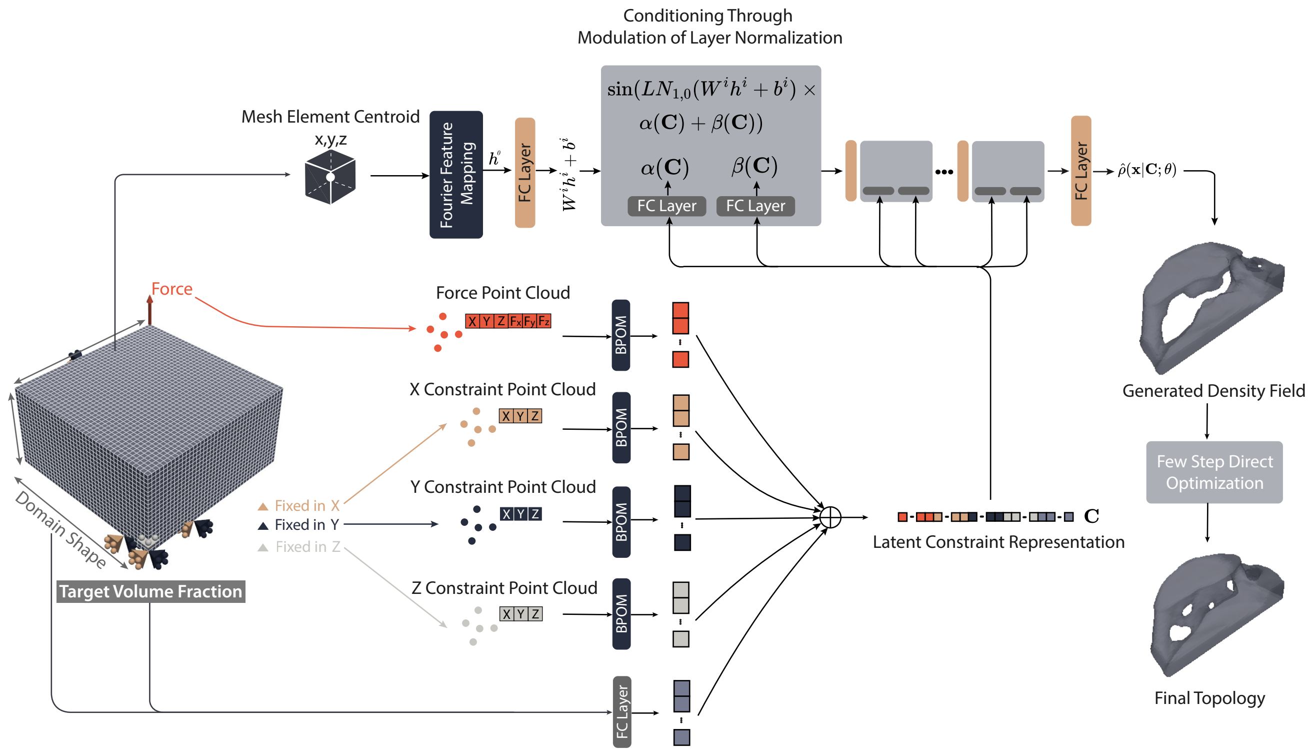

Figure 4: The NITO-3D framework for generalizable topology optimization through deep optimization. BPOM is used to process point cloud representations of the boundary conditions and a neural field is guided by these representations by modulating layer normalization based on the latent representation of the constraints. In the end, the resulting density field is further refined through a few steps of direct optimization.

3.2.1 Implicit Neural Representation For Learning Material Density

The core objective of our framework is to learn the material distribution within a domain to minimize the mechanical compliance of the resulting structure. Following

3.2.2 Beyond Physical Fields: Latent Constraint Representation

One of the key contributions of NITO-3D compared to recent works

Conditioning on Physics Fields: Previous research [14, 32, 35, 14] predominantly employs physical fields like stress and strain energy derived from simulations to represent boundary conditions into problems. This method is often deemed essential for high-performance topology generation using conditioned generative models. However, this reliance on field-based conditioning restricts the models to a specific resolution and domain due to the dependency on CNNs, impacting their generalizability. It is also time-consuming, effectively requiring a finite element simulation for every topology generated. The prevalent use

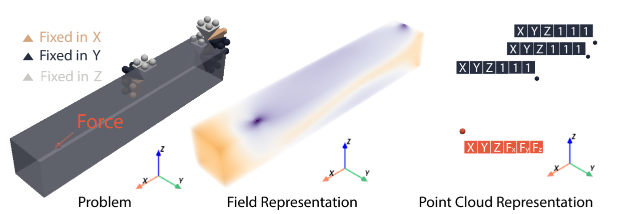

Figure 5: Comparison of field-based representations, given a TO problem (left), such as stress fields (middle), and point-cloud-based (right). Unlike the iterative FEA method, point clouds provide a more generalizable and memory-efficient representation of the boundary conditions.

of field-based conditioning is arguably due to limitations in existing conditioning mechanisms, where conditioning is typically applied only at the initial layers. NITO-3D leverages a simplified model to represent and integrate conditions, enhancing generality and applicability.

Constraints as Point Clouds: Using the 'Boundary Point Order-invariant MLP' (BPOM) method from

Order-Invariant Aggregation Given the variable size of point clouds, we employ order-invariant pooling, as in

3.2.3 Building Blocks of Neural Implicit Fields

In this section, we will delve into some key aspects of our framework's implementation. Our implementation utilizes neural fields constructed from basic multilayer perceptrons (MLPs). Among the various implicit neural field models, our approach specifically adopts SIREN layers, as introduced by Sitzmann et al.

Advanced Conditioning Techniques for TO: Neural fields are capable of adapting to new scenarios through conditioning on a latent vector $\mathbf{C}$. This vector represents the unique attributes of a TO problem, including boundary conditions and volume ratio, or forces and domain shape. The concept of Feature-wise Linear Modulation (FiLM) proposed by Perez et al.

Synthesizing Components for Topology Optimization: Putting this all together this framework can be described as:

$$f_\theta(\mathbf{x}, \mathbf{C}) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(0)}(\mathbf{x}, \mathbf{C})) \quad (1)$$

where $f^{(i)}$ for $i \in \{1, 2, ..., L-1\}$ indicates the function applied at each layer of the neural field except the first and last layer. Each layer takes a hidden input $h^{(i)}$ and sends it through a fully connected (FC) layer and normalization with modulation based on the condition vector, which in this case is the latent constraint vector $\mathbf{C}$:

$$f^i(h^i, \mathbf{C}) = \sin(LN_{1,0}(W^i h^i + b^i) \times \alpha(\mathbf{C}) + \beta(\mathbf{C})), \quad (2)$$

where $LN_{1,0}$ is layer norm with scale=1 and shift=0 and α and β are FC layers that use the condition $\mathbf{C}$ to determine the feature-wise scale and shift for the normalization. In the first layer, the input coordinates are transformed by Fourier feature mapping before being passed to the first FC layer in the neural field. The final layer lacks the conditioning modulation on layer normalization and is activated by a *sigmoid* function rather than the sin activation used in other layers. These neural fields can easily be adapted to any domain shape or resolution so long as it is specified in the latent representation $\mathbf{C}$. This stems from the fact that the input to the neural field is coordinates, which allows for sampling arbitrarily in space making this kind of model well-suited for generalization to different domains.

Perfecting Generated Topologies with Few-step Refinement: While generative models and deep learning strategies have shown promise in TO, the quality of topologies they produce still lags behind that of traditional optimization baselines. Giannone et. al.

4 Experiments & Results

In this section, we conduct various experiments to demonstrate, quantify, and compare the capabilities of NITO-3D to SIMP. As existing CNN-based methods do not generalize to multiple domain shapes, unstructured meshes, and different mesh resolutions, we solely focus on quantifying the performance of our method by comparing it against the SIMP optimization method.

With the experiments in this section, we provide compelling evidence that:

1. NITO-3D is scalable, resolution-free, and compact, with a smaller number of parameters than even state-of-the-art 2D models, yet it is capable of performing very well and on par with SIMP.
2. NITO-3D is faster than most 2D state-of-the-art models, despite operating on 3D data, and is capable of accelerating the entire TO process in 3D by an order of magnitude in comparison to conventional iterative optimization schemes such as SIMP.
3. NITO-3D integrates the speed of deep learning models with the accuracy and dependability of optimization methods, establishing a robust 'deep optimization' framework. This approach is a noteworthy avenue for the widespread implementation and application of deep learning techniques in the field of engineering design.

4.1 Experimental Details

We first establish some of the details of the experiments we run to clarify what we measure and how the measurements are performed.

Topology Optimization Dataset: We use our new dataset of optimized 3D topologies for the training and testing of our model. 2,000 samples are held out of the training split for testing.

Evaluation Metrics: We evaluate the models in terms of performance (i.e., minimum compliance), constraint satisfaction, and inference time. Initially, to evaluate the effectiveness of the models in minimizing compliance, we calculate the compliance error (CE) by determining the difference between the compliance of a produced sample and the compliance of the SIMP-optimized solution for the same issue. Additionally, we assess the volume fraction error (VFE), which represents the absolute discrepancy between the actual volume fraction of the generated topology and the target volume fraction designated for the problem. Beyond these fundamental performance indicators, we also measure inference time and compare it against the speed of the SIMP optimizer for context.

Setup: We train NITO-3D for 50 epochs with a uniform sampling of points in space. The batch size used to train NITO-3D is 32, with 2048 points sampled for each item in the batch. The optimizer we use for training is AdamW, with a decaying learning rate on the cosine schedule, which starts at a learning rate of 10^{-4} and decays by stepping at the end of each epoch to the minimum learning rate of 10^{-5}. Since the model in this work is not probabilistic and would not yield different results for a given boundary conditions, we only asses the performance of the trained model on one sample for each boundary condition. Similar to prior work, when reporting results, we remove outlier samples

4.2 Qualitative Results & Discussion of Generalizability

Here we visualize some of the results of our experiments and discuss some notable observations. Figure 6 shows a few examples of topologies generated by NITO-3D with and without additional SIMP optimization steps. In most cases, NITO-3D produces topologies that are very close to the ground truth topology. However, we see that NITO-3D sometimes struggles to cleanly generate intricate features that are present in the ground truth topologies. This highlights the value of adding a very small number of steps of direct optimization. Even with 5 steps of SIMP on top of NITO-3D, the resulting topologies quickly converge to detailed topologies, and with 10 steps of direct optimization, the results start to look even better. This observation is also supported by our quantitative results (discussed later), where the median compliance error of NITO-3D reduces from 0.32% to 0.11% in five steps and 0.077% in ten steps.

Importantly, we see NITO generating topologies in different domain shapes and resolutions without re-

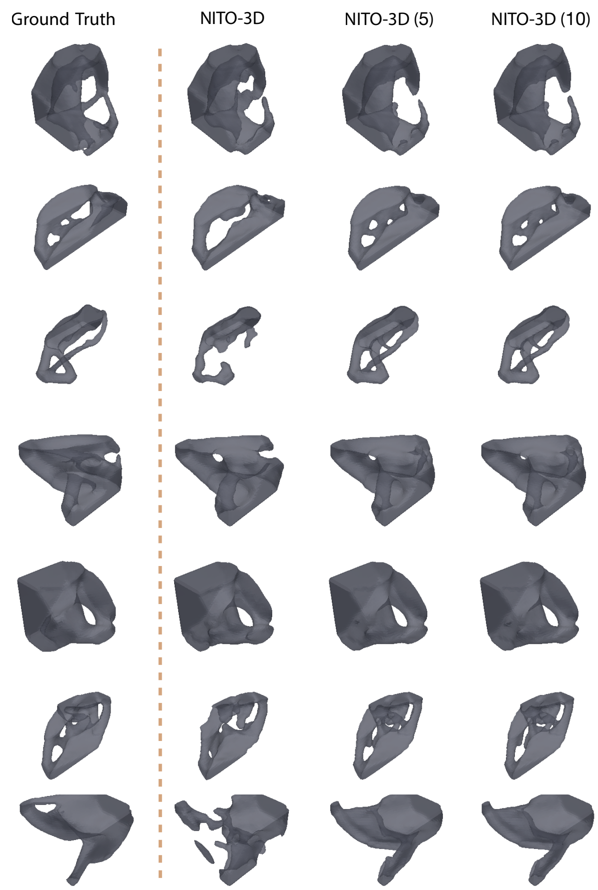

Figure 6: Qualitative visualization of NITO-3D generated topologies with and without a few steps of direct optimization. Column 1: ground truth obtained using our SIMP optimizer. Column 2: The topology produced by the neural field with BPOM without direct optimization. Columns 3 & 4: the NITO-3D framework output leveraging 5 and 10 steps of direct optimization. We see that NITO-3D is an effective deep optimization framework that could enable accelerated topology optimization in a generalizable fashion. Note that the raw output of the neural field includes continuous values that are not easy to visualize, which is why a few steps of optimization are effective

training. As discussed, most prior works focus on one domain shape at a time and require retraining on new domains. We demonstrate that generalizable frameworks like NITO-3D can cover different domain shapes and resolutions simultaneously without any significant loss of performance. Notably, NITO-3D has also learned to generate near-optimal topologies without the need for physical fields as input for describing boundary conditions, showing that BPOM is an effective strategy for conditioning deep learning models on sparse boundary conditions. This is also a critical step for generalizability, avoiding fixed-domain non-sparse conditioning strategies. These results showcase NITO-3D's promise as a generalizable and foundational framework for topology optimization.

4.3 Performance

In Table 2, we present the performance metrics and constraint satisfaction outcomes. The table reveals that our TO framework, which leverages neural fields, achieves comparable results to SIMP in most scenarios, even without implementing the direct optimization step. This is highlighted by the median compliance error of 0.32%, marked in green in Table 2. However, it's noted that in some instances, solely using neural fields results in significant deviations, pushing the average compliance error to 5.95%, much higher than the median. Yet, incorporating a direct optimization step significantly enhances the neural field's density predictions. Unlike the binary outcomes seen in Figure 6, the neural field actually produces a probability map that can be finely tuned through direct optimization, avoiding the need for binary thresholding. This nuanced approach allows for rapid optimization convergence, particularly in areas of uncertainty predicted by the neural field, as shown by the reduced compliance error to 0.52% and volume fraction error to below 1% with just five steps of direct optimization in Table 2. This capability outpaces approaches that rely on more deterministic starting points

Table 2: Quantitative evaluations of NITO-3D with and without direct optimization. This table shows that NITO's performance is very close to SIMP with much less time devoted to optimization. Even without direct optimization, NITO-3D performs very well on the test data with an average compliance error of 5.96%. Note that the median value for compliance error is far lower, which is because a handful of outliers skew the mean while in most cases vanilla neural fields would perhaps be close enough to SIMP.

Model	CE % Mean	CE % Med	VFE % Mean	VFE % Med
NITO-3D	5.95	0.32	4.38	2.40
NITO-3D (5) (ours)	0.52	0.11	0.96	0.81
NITO-3D (10)(ours)	0.31	0.077	0.67	0.54

Another notable observation we encountered in the quantitative data is that in some cases, NITO-3D outperforms SIMP. In the 2,000 test samples that we used in our experiments, NITO-3D with 10 steps of optimization outperformed the ground truth in 86 of the cases, about 4%. This is something that other works

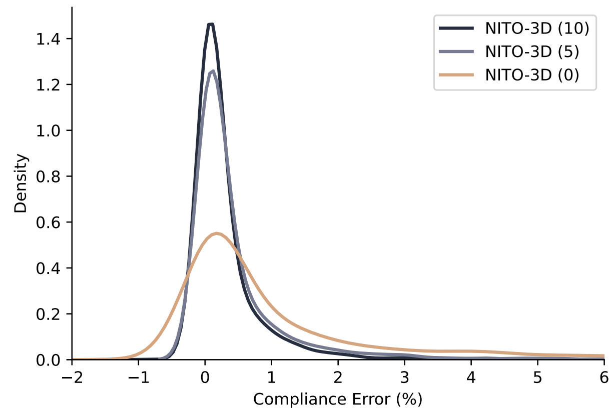

Figure 7: Kernel density estimate plot of compliance error distributions for NITO-3D with 0, 5, and 10 steps of optimization. With more optimization, NITO-3D generates compliances with near-zero compliance error. Interestingly, NITO-3D without optimization yields more designs with lower compliance than SIMP (negative compliance error). This phenomenon is explained by the higher volume fraction error, causing some of these generated design to use more material than SIMP.

have also observed in some cases

4.4 Inference Speed & Scalability

One of the key benefits of deep learning for TO is speed. We therefore measure inference time and speed for different configurations of NITO-3D in Table 3. We measure inference times and compare them against the inference time for performing the full optimization using SIMP. We see that the neural field alone only takes an average of 0.124s to compute topologies, which is an impressive **1900x improvement** in inference, this is also faster than most CNN-based methods in 2D

NITO-3D has several features that allow for natural parallelism and low memory use. NITO trains by sampling batches of points from a field. This enables easy parallelism, accelerating training on modern GPUs. Notably, it also avoids problematic memory scaling trends seen in convolution-based models. To train on larger and larger dimensionalities, CNNs require more memory

5 Conclusion & Future Works

We propose NITO-3D, a 3D Neural Implicit Topology Optimization framework, marking a departure

Table 3: Average inference time measured for SIMP and NITO-3D in different configurations. Here we see that NITO-3D without direct optimization is multiple orders of magnitude faster than SIMP, while even with 10 steps of direct optimization, NITO-3D is 93% faster than SIMP. Note that times are averaged for test samples from the dataset which have different element counts. SIMP time is calculated for 150 steps of optimization, the rough average iteration count to convergence in generating the dataset. These times are measured using an RTX 4090 GPU and i9-13900K CPU.

	SIMP (150)	NITO-3D (0)	NITO-3D (5)	NITO-3D (10)
Inf. Time (s)	239.14	0.124	8.10	16.06
Acceleration	0%	99.95%	96.61%	93.28%

from traditional neural TO methods as a pure resolution- and domain-agnostic approach. Our proposed Boundary Point Order-Invariant MLP (BPOM) sidesteps the complexities that CNNs face, enabling NITO to adapt to various resolutions and domain shapes without retraining. It also has a smaller parameter footprint than 2D models. The ability of NITO-3D to scale and generalize provides a strong basis for future models in topology optimization and other areas involving physics, solving high-dimensional problems that were not possible with CNN-based methods.

We also introduce an efficient Python solver for rapid data generation and a comprehensive dataset of 122K optimized 3D topologies to aid future TO model development. While NITO advances TO, its deterministic nature limits design diversity and its performance on new problem types. Future research must focus on this aspect, possibly by incorporating advancements in probabilistic approaches for neural implicit fields

Acknowledgments

We acknowledge the help and guidance provided by Giorgio Giannone in developing the methodology. We would also like to thank Professor Josephine V. Carstensen for her course on topology optimization which provided the necessary background and guidance to develop the solver used in this work for data generation. The authors also acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing high-power computing resources that allowed for the timely creation of our dataset. Finally, we also thank MathWorks for supporting Amin Heyrani Nobari's studies while working on this project

References

[1] D. W. Abueidda, S. Koric, and N. A. Sobh. Topology optimization of 2d structures with nonlinearities using deep learning. *Computers & Structures*, 237:106283, 9 2020.

[2] G. C. Ates and R. M. Gorguluarslan. Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. *Structural and Multidisciplinary Optimization*, 63(4):1927–1950, 4 2021.

[3] N. Aulig and M. Olhofer. Evolutionary generation of neural network update signals for the topology optimization of structures. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pages 213–214, New York, NY, USA, 7 2013. ACM.

[4] S. Banga, H. Gehani, S. Bhilare, S. Patel, and L. Kara. 3d topology optimization using convolutional neural networks. *Preprint*, 8 2018.

[5] S. Barmada, N. Fontana, A. Formisano, D. Thomopulos, and M. Tucci. A deep learning surrogate model for topology optimization. *IEEE Transactions on Magnetics*, 57(6):1–4, 6 2021.

[6] M. M. Behzadi and H. T. Ilieş. Real-time topology optimization in 3d via deep transfer learning. *Computer-Aided Design*, 135:103014, 6 2021.

[7] A. Blattmann, R. Rombach, K. Oktay, and B. Ommer. Retrieval-augmented diffusion models. *arXiv preprint arXiv:2204.11824*, 2022.

[8] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[10] A. Chandrasekhar, A. Mirzendehdel, M. Behandish, and K. Suresh. Frc-tounn: Topology optimization of continuous fiber reinforced composites using neural network. *Computer-Aided Design*, 156:103449, 2023.

[11] A. Chandrasekhar, S. Sridhara, and K. Suresh. Gm-tounn: Graded multiscale topology optimization using neural networks. *arXiv preprint arXiv:2204.06682*, 2022.

Table 4: In this table, we summarize some of the latest works on TO using deep learning. Domain adaptability refers to an approach's ability to be trained on multiple domain shapes simultaneously. Resolution-free refers to an approach's ability to be trained and inferred at multiple resolutions at the same time. Scalable training refers to the fact that the training does not require processing information on an entire domain, enabling training on very large domains (e.g., the model does not need physical fields for training and does not need to generate the solution during training). We see that the best-performing convolution-based methods are not generalizable while requiring significantly more parameters for 2D than NITO-3D needs for 3D. † IF-TONIR has no public code and its model size is unknown. ‡ IF-TONIR's training is resolution-dependent due to field calculations. During inference, different topologies can be sampled, but the provided physical field must be calculated at the same resolution as the training data.

Model	Domain Dimensionality	Parameter Count (M)	Domain Adaptable	Resolution Free	Scalable Training	Base Architecture
TopoDiff	2D	121	✗	✗	✗	Convolution
TopoDiff-Guided	2D	239	✗	✗	✗	Convolution
DOM	2D	121	✗	✗	✗	Convolution
TopologyGAN	2D	~ 300	✗	✗	✗	Convolution
IF-TONIR (CNN Based Encoder/Conditioning)	2D	N/A†	✓	✗‡	✗	Neural Field
NITO	2D	**22**	✓	✓	✓	Neural Field
NITO-3D	3D	**72**	✓	✓	✓	Neural Field

[12] A. Chandrasekhar and K. Suresh. Multi-material topology optimization using neural networks. *Computer-Aided Design*, 136:103017, 7 2021.

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

[14] H. Chen, A. Joglekar, and L. Burak Kara. Topology Optimization Using Neural Networks With Conditioning Field Initialization for Improved Efficiency. *Journal of Mechanical Design*, 146(6):061702, 12 2023.

[15] H. Chen, K. S. Whitefoot, and L. B. Kara. Concurrent build direction, part segmentation, and topology optimization for additive manufacturing using neural networks. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 86229, page V03AT03A029. American Society of Mechanical Engineers, 2022.

[16] R. Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

[17] H. Deng and A. C. To. A parametric level set method for topology optimization based on deep neural network (dnn). *Preprint*, 1 2021.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] S. Dittmer, D. Erzmann, H. Harms, and P. Maass. Selto: Sample-efficient learned topology optimization. *arXiv preprint arXiv:2209.05098*, 2022.

[20] M. O. Elingaard, N. Aage, J. A. Bærentzen, and O. Sigmund. De-homogenization using convolutional neural networks. *Computer Methods in Applied Mechanics and Engineering*, 388:114197, 1 2022.

[21] K. Fujita, K. Minowa, Y. Nomaguchi, S. Yamasaki, and K. Yaji. Design concept generation with variational deep embedding over comprehensive optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21*, Virtual, Online, Aug 2021. ASME.

[22] G. Giannone and F. Ahmed. Diffusing the optimal topology: A generative optimization approach. *arXiv preprint arXiv:2303.09760*, 2023.

[23] G. Giannone, A. Srivastava, O. Winther, and F. Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. *arXiv preprint arXiv:2305.18470*, 2023.

[24] N. Hertlein, P. R. Buskohl, A. Gillman, K. Vemaganti, and S. Anand. Generative adversarial network for early-stage design flexibility in topology

optimization for additive manufacturing. *Journal of Manufacturing Systems*, 59:675–685, 4 2021.

[25] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[26] J. Hu, Y. He, B. Xu, S. Wang, N. Lei, and Z. Luo. If-tonir: Iteration-free topology optimization based on implicit neural representations. *Computer-Aided Design*, 167:103639, 2024.

[27] A. Joglekar, H. Chen, and L. B. Kara. Dmf-tonn: Direct mesh-free topology optimization using neural networks. *Engineering with Computers*, Oct 2023.

[28] Y. Joo, Y. Yu, and I. G. Jang. Unit module-based convergence acceleration for topology optimization using the spatiotemporal deep neural network. *IEEE Access*, 9:149766–149779, 2021.

[29] N. A. Kallioras, G. Kazakis, and N. D. Lagaros. Accelerated topology optimization by means of deep learning. *Structural and Multidisciplinary Optimization*, 62(3):1185–1212, 9 2020.

[30] V. Keshavarzzadeh, M. Alirezaei, T. Tasdizen, and R. M. Kirby. Image-based multiresolution topology optimization using deep disjunctive normal shape model. *Computer-Aided Design*, 130:102947, 2021.

[31] F. Ma and Z. Zeng. High-risk prediction localization: evaluating the reliability of black box models for topology optimization. *Structural and Multidisciplinary Optimization*, 62(6):3053–3069, 12 2020.

[32] F. Mazé and F. Ahmed. Diffusion models beat gans on topology optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Washington, DC, 2023.

[33] N. Napier, S.-A. Sriraman, H. T. Tran, and K. A. James. An artificial neural network approach for generating high-resolution designs from low-resolution input in topology optimization. *Journal of Mechanical Design*, 142(1), 1 2020.

[34] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[35] Z. Nie, T. Lin, H. Jiang, and L. B. Kara. Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *Journal of Mechanical Design*, 143(3), 2021.

[36] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang. Deep generative design: Integration of topology optimization and generative models. *Journal of Mechanical Design*, 141(11), 09 2019. 111405.

[37] M. Olhofer, E. Oñate, J. Oliver, A. Huerta, and N. Aulig. Topology optimization by predicting sensitivities based on local state features. Technical report, 2014.

[38] C. Parrott, D. Abueidda, and K. A. James. Multi-head self-attention gans for multiphysics topology optimization. In *AIAA AVIATION 2022 Forum*, page 3726, 2022.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[40] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[41] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[42] S. Rawat and M. H. H. Shen. A novel topology design approach using an integrated deep learning network architecture. *Preprint*, 8 2018.

[43] L. Regenwetter, A. H. Nobari, and F. Ahmed. Deep generative models in engineering design: A review. *Journal of Mechanical Design*, 144(7):071704, 2022.

[44] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2021.

[45] H. Sasaki and H. Igarashi. Topology optimization accelerated by deep learning. *IEEE Transactions on Magnetics*, 55(6):1–5, 6 2019.

[46] C. Sharpe and C. C. Seepersad. Topology design with conditional generative adversarial networks. Volume 2A: 45th Design Automation Conference, 08 2019. V02AT03A062.

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

[48] B. Song, R. Zhou, and F. Ahmed. Multimodal machine learning in engineering design: A review and future directions. *arXiv preprint arXiv:2302.10909*, 2023.

[49] I. Sosnovik and I. Oseledets. Neural networks for topology optimization. *Preprint*, 9 2017.

[50] H. Sun and L. Ma. Generative design by using exploration approaches of reinforcement learning in density-based structural topology optimization. *Designs*, 4(2):10, 5 2020.

[51] E. Ulu, R. Zhang, and L. B. Kara. A data-driven investigation and estimation of optimal topologies under variable loading configurations. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 4(2):61–72, 3 2016.

[52] R. V. Woldseth, N. Aage, J. A. Bærentzen, and O. Sigmund. On the use of artificial neural networks in topology optimisation. *Structural and Multidisciplinary Optimization*, 65(10):294, 2022.

[53] R. Wu, C. Xiao, and C. Zheng. Deepcad: A deep generative network for computer-aided design models, 2021.

[54] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. Neural fields in visual computing and beyond, 2022.

[55] L. Xue, J. Liu, G. Wen, and H. Wang. Efficient, high-resolution topology optimization method based on convolutional neural networks. *Frontiers of Mechanical Engineering*, 16(1):80–96, 3 2021.

[56] A. Yildiz, N. Öztürk, N. Kaya, and F. Öztürk. Integrated optimal topology design and shape optimization using neural networks. *Structural and Multidisciplinary Optimization*, 25(4):251–260, 10 2003.

[57] S. Yoo, S. Lee, S. Kim, K. H. Hwang, J. H. Park, and N. Kang. Integrating deep learning into cad/cae system: generative design and evaluation of 3d conceptual wheel. *Structural and Multidisciplinary Optimization*, 64(4):2725–2747, 10 2021.

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