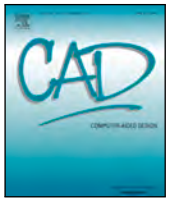
Computer-Aided Design 167 (2024) 103639

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad



IF-TONIR: Iteration-free Topology Optimization based on Implicit Neural Representations

Jiangbei Hu [a,b], Ying He [a,*], Baixin Xu [a], Shengfa Wang [b], Na Lei [b], Zhongxuan Luo [b]

[a] Nanyang Technological University, Singapore
[b] Dalian University of Technology, China

ARTICLE INFO

Keywords:
Topology optimization
Implicit neural representations
Variational Autoencoder
Persistent homology
Topological loss

ABSTRACT

Topology optimization holds great significance as a research topic in the field of mechanical engineering, aiming to design and optimize structures to achieve desired performance while adhering to specific constraints. However, its high computational complexity and iterative optimization process severely impact the efficiency, which presents substantial obstacles to its practical applications. To tackle this challenge, recent research is dedicated to the advancement of iteration-free topology optimization methods that leverage neural networks and deep learning, aiming to directly predict optimal structures through optimization problem configurations. In this paper, we propose IF-TONIR, a novel data-driven topology optimization method that utilizes implicit neural representations. Our approach employs signed distance fields to represent structures, offering compact and smooth representations that effectively eliminate the checkerboard phenomenon commonly observed in density-based methods. IF-TONIR leverages Conditional Variational Autoencoders, which use a CNN-based encoder and a MLP-based decoder to learn and reconstruct optimal structures. We employ the features extracted from physical information as conditions to guide the decoder in generating optimal structures that adhere to specific design domain shapes and boundary conditions. Furthermore, we propose the integration of a topological loss based on persistent homology to train the model. This loss function effectively penalizes the existence of structural disconnections in the reconstructed output, thereby enhancing the overall physical reliability of the generated structures. Various experiments have demonstrated that our iteration-free topology optimization method based on implicit representations can accurately identify regions of high strain energy and generate continuous structures with low compliance. The methods also holds the theoretical capability of outputting optimal structures at any desired resolution. Our code and dataset are available on https://github.com/jbHu67/IF-TONIR.git

1. Introduction

Topology Optimization (TO) is a computational design methodology aimed at determining the optimal distribution of materials within a design domain to achieve specific performance objectives while adhering to given constraints [1]. Over the past three decades, the advancement of computer technology and 3D printing has led to significant progress in topology optimization methods. These include the Solid Isotropic Material with Penalization (SIMP) approach [2], the Evolutionary Structural Optimization (ESO) approach [3], the Level Set Method (LSM) [4], and the more recent Moving Morphable Components/Voids (MMC/MMV) approach [5,6]. These methods differ in their structural representation, but they all iteratively seek the optimal material distribution based on physical response analysis and parameter gradient information. The solution procedure in topology optimization involves repetitive and computationally demanding finite element analysis to solve the physical equilibrium equations.This computational demand is the primary reason for the inefficiency of these methods, thereby hindering the practical application of topology optimization in real-world scenarios.

To mitigate the issue of inefficiency, significant efforts have been dedicated to exploring promising approaches that accelerate the topology optimization process [7]. These approaches leverage various techniques such as multi-grid solvers, model reduction, high-performance computing, and combinations thereof. In recent years, the field of mechanical engineering has witnessed the significant impact of Artificial Intelligence (AI). In particular, there has been a surge of interest in utilizing neural networks and deep learning techniques to

* Corresponding author.
E-mail addresses: jiangbei.hu@ntu.edu.sg (J. Hu), yhe@ntu.edu.sg (Y. He), baixin001@e.ntu.edu.sg (B. Xu), sfwang@dlut.edu.cn (S. Wang), nalei@dlut.edu.cn (N. Lei), zxluo@dlut.edu.cn (Z. Luo).

https://doi.org/10.1016/j.cad.2023.103639
Received 14 July 2023; Received in revised form 24 October 2023; Accepted 28 October 2023
Available online 4 November 2023

address topology optimization challenges [8–13]. Among these strategies, the *iteration-free* methods [14,15] has attracted considerable attention, which construct deep neural networks to directly predict the optimal structures from conditions and problem configurations. The potential to achieve near real-time topology optimization through this direct-design approach is highly appealing. Sigmund et al. highlighted several challenging issues associated with direct design methods in their review [13]. Firstly, constructing the training dataset using traditional topology optimization methods is computationally expensive. Secondly, the trained model struggles to generalize to unseen samples. Thirdly, commonly used loss functions, such as Mean Squared Error (MSE) or Binary-Cross-Entropy (BCE), lack the required sensitivity to detect and prevent structural disconnections. In this paper, we propose a novel iteration-free topology optimization framework called IF-TOINR, which aims to address the aforementioned issues. The architecture of IF-TONIR is illustrated in Fig. 1.

Implicit neural representation (INR) has recently garnered considerable attention [16–19]. INR utilizes neural networks to generate continuous parameterized representations of discrete signal fields, such as color, signed distance values, or density. The application of INR models in various challenging domains has demonstrated their robustness to changes in input data and their ability to generalize well to unseen scenarios. Taking inspiration from this, we choose to represent structures using continuous *signed distance fields* (SDFs). This representation approach, in contrast to discrete representations like density-based methods, offers a compact and smooth representation that effectively eliminates the checkerboard phenomenon. Moreover, the use of INR decouples the design process from spatial grids, theoretically enabling the acquisition of structures at any desired precision. Our IF-TOINR framework utilizes a Variational Autoencoder (VAE) to learn and reconstruct the SDFs of optimal structures from the training dataset. We employ a CNN-based encoder to embed the structures into a lower-dimensional latent space. Subsequently, the MLP-based decoder takes both the coordinates of a specific point within the design domain and a latent vector from the learned latent space as inputs. It then predicts the corresponding SDF value for that point, incorporating the information from both sources. IF-TONIR incorporates a subnetwork called *physics-net*, responsible for extracting features from various conditions, including loads, displacement constraints, as well as stress and strain fields. We employ the feature vector from the physics-net as the condition code to guide the VAE model in generating structures based on the given problem configurations. To enable the network to recognize the shape of the design domain, we convert the design domain into its SDF representation. Then, we append the SDF value of the design domain to each point as an additional property. These points coordinates along with their properties are fed into the decoder for outputting the SDF values of optimal structures. Ultimately, we obtain the optimal structures by extracting the zero-level set of their SDFs.

The choice of loss function plays a crucial role in training neural networks. However, classical loss functions like MSE or BCE only focus on local discrete signal errors, which may not adequately capture global structural features like connectivity. In this work, we introduce a topology-aware loss term based on persistent homology techniques [20] from computational topology. The topological loss can capture the error in topological features between the reconstructed structures and the ground truth (GT), such as the number of connected components and holes. Experimental results validate that the topological loss can enhance the reconstruction accuracy and reduce the occurrence of structural disconnections. Once the model is trained, we can directly infer the corresponding optimal structure by inputting relevant design domain shape, problem configurations and physical information.

We summarize our contributions as follows:

- We propose a novel iteration-free topology optimization method that outputs the optimal structures directly from various design domain shapes and boundary conditions.
- In addition to commonly used geometric losses, we incorporate a topological loss based on persistent homology to quantify structural continuity, which effectively detect and measure even tiny structural disconnections.
- IF-TONIR generates compact and smooth implicit representations of structures. It separates the design process from spatial grids, allowing for the generation of structures at any desired resolution.

2. Related work

2.1. Iteration-free topology optimization algorithms

Unlike traditional topology optimization techniques, iteration-free topology optimization methods leverage neural networks to directly predict the optimized material distribution based on the conditions and problem configurations, bypassing the need for iterative processes. The near real-time efficiency of iteration-free topology optimization methods has garnered significant attention from researchers, resulting in a surge of studies in recent years. The most commonly used approach is inspired by the traditional SIMP method, in which the design domain is partitioned into elements, and the material distribution is represented as element densities. Given this discrete representation, it is intuitive to utilize popular network architectures such as Convolutional Neural Networks (CNNs) to solve classification (determining whether an element is void or solid) or regression problems (predicting the element density value) [21,22]. In recent years, popular generative models in the field of AI, such as Variational Autoencoder models (VAEs) and Generative Adversarial Networks (GANs), have also been preliminarily explored in the field of topology optimization [23]. Guo et al. [24] utilize augmented VAE to embed the 2D structures into a latent space, thereby enhancing topology optimization capabilities. Sharpe et al. [25] introduced conditional GAN to address topology optimization problems without iterations, while Nie et al. further improved this method in their work called TopologyGAN [26], enhancing the generalization capabilities to various load configurations and volume constraints. As a new generative architecture, diffusion models have achieved greater success in the field of image generation compared to GANs. This accomplishment has also been demonstrated by Mazé et al. in the context of topology optimization [27]. They proposed the TopoDiff framework, which introduced explicit compliance and floating materials guidance to diffusion models, and achieved better performance in generating structures with low compliance and good feasibility.

Many of the aforementioned algorithms are inspired by network architectures in the field of computer vision. These approaches typically focus on fixed-resolution problems within a regular design domain. When dealing with new resolutions or shapes, retraining or super-resolution techniques are required. Moreover, the density-based methods are prone to the occurrence of the checkerboard phenomenon, where meaningless isolated solid elements appear in the generated structure. TopoDiff addresses the checkerboard issue by training a network to discriminate floating materials. However, it requires additional data preparation and training, which makes more computational resources and time-consuming. Many studies have effectively used machine learning techniques to solve topology optimization problems in complex and irregular design domains [28–30]. However, these methods are not implemented in an end-to-end manner. Therefore, in order to address these problems, we choose to utilize the signed distance fields to represent the design domains and structures.

2.2. Implicit neural representation

Implicit neural representation refers to utilize neural networks to parameterize information fields such as color, lighting, depth, and geometric features [31]. The success of the NeRF (Neural Radiance Fields) [17] has demonstrated the effectiveness and advantages of the

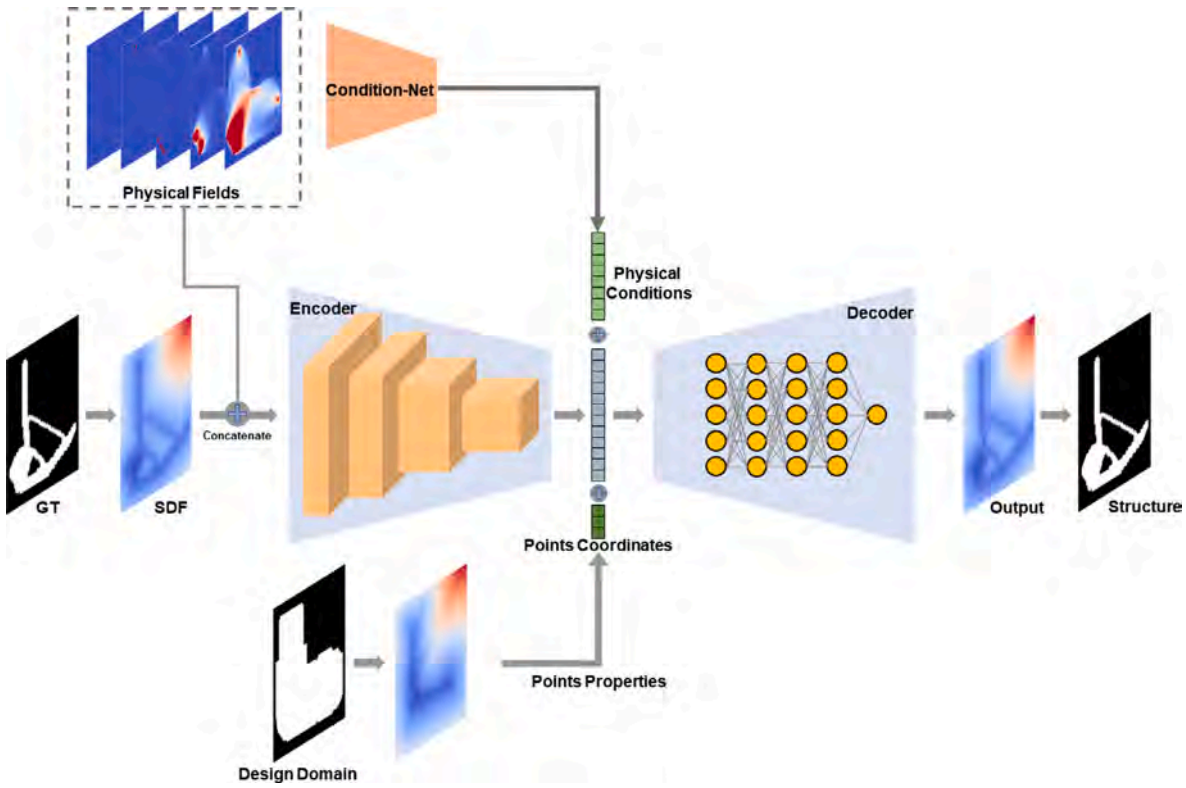


Fig. 1. The architecture of IF-TOINR network, consisting of a VAE to reconstruct optimal structures. Physical conditions extracted from physical fields, along with points properties, guide the network to generate structures that adhere to specific design domain shapes and boundary conditions.

INR in geometric representation and rendering. Numerous work focused on describing geometric structures in the form of signed distance fields [16,32,33]. These methods employ neural networks to embed a class of geometric objects into a low-dimensional latent space. By concatenating the latent vector with spatial coordinates and passing it to the decoder, the signed distance value corresponding to each coordinate can be obtained. In this way, the geometric structure can be represented by a continuous function, which is convenient for rendering and other operations. Zheng et al. proposed the SDF-StyleGAN [34], which can generate novel 3D structures with controllable styles. Zhang et al. [35] and Lyu et al. [36] also have proposed shape generation models based on diffusion models and SDF implicit neural representation. These methods are capable of generating structures from point clouds, text, or images. Indeed, topology optimization can be seen as a problem of generating structures under the configuration of relevant physical constraints. Based on the implicit neural representation, explicit mesh or voxels can be avoided. Furthermore, once the network training is completed, it is theoretically possible to generate structures at any desired precision based on the obtained signed distance functions. Recently, Zhang et al. proposed TOINR [37] that utilizes the implicit neural representation to solve topology optimization problems. Their work extensively discusses the advantages of using neural networks for implicit parameterization of structures. While there may be similarities in the algorithms, our framework mainly focuses on developing data-driven method for generating optimized structures without iterative processes.

2.3. Persistent homology

Mathematically, features derived from geometric analysis can capture local structural information. However, they often become overwhelmed by excessive details, resulting in increased data complexity. Persistent homology (PH) provides an ideal way to bridge the gap between geometry and topology [38]. By analyzing the evolution of topological features, persistent homology allows for a more robust representation of structural information, capturing essential topological characteristics while reducing the impact of fine-scale geometric details. PH has showcased successful applications in various machine learning-related fields. It serves as a powerful tool for simplifying and extracting topological features from complex data [39,40]. Hofer et al. [41] introduced an approach that incorporates persistence diagrams into deep learning architecture by leveraging a differentiable projection function. Moor et al. [42] proposed a topological autoencoder that focuses on preserving the topological features of the input data in low dimensional representations. Gabrielsson et al. [43] introduced a topology layer for machine learning, which can enhance the reconstruction and segmentation performance of the network. Zhang et al. [44] illustrated in their work that the topological loss based on persistent homology can enforce the model to achieve certain desired topological property. Different complexes can be used to compute persistent homology for different data format. For example, Vietoris–Rips complex is used for point clouds [45,46], while cubical complex is used for images [47] or voxels [48]. In the field of topology optimization, Behzadi et al. [49] were the first to use the topological loss based on persistent homology to enhance the structural continuity. They computed the persistence diagrams of the structures using the Ripser complexes on the discrete material occupancy data predicted by network. Different from their work, we represent the structures using the signed distance fields. By using the SDFs as the filtration functions (Definition 3.2) to construct the complexes, the smooth and continuous implicit representation facilitates more stable training for the topological loss.

3. Preliminary

3.1. Topology optimization formulation

Topology optimization aims to optimize the material distribution within a designated domain to meet specific objectives and constraints.

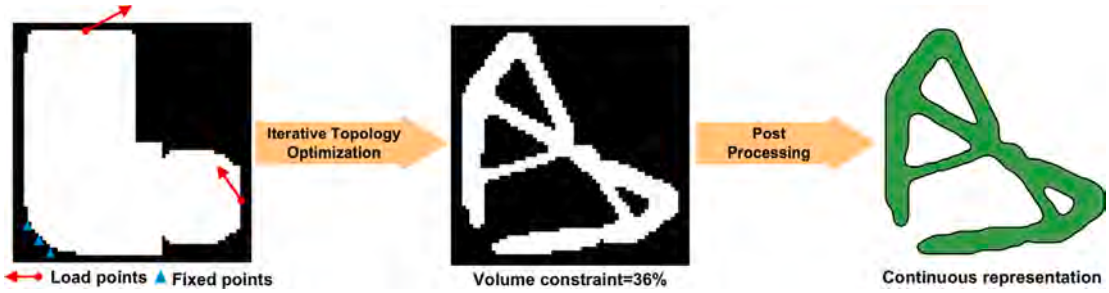


Fig. 2. The general process of traditional topology optimization. Problem configurations (right): The white region indicates the design domain. Loads and displacement constraints are applied on the boundary. A density-based representation of the optimized structure (middle): This representation is obtained through iterative optimization while adhering to a specific volume constraint. Continuous structural representation (left): For practical industrial manufacturing, post-processing is necessary to convert the obtained result into a continuous structural representation.

In this paper, we evaluate our proposed algorithm on the well-known minimum structural compliance problem. Notably, given a suitable dataset, our method can be readily extended to address topology optimization problems with different objectives and constraints, such as stress-related problems [50] or heat conduction [51] problems. Specifically, we consider the 2D minimum compliance problem under a volume constraint. As shown in Fig. 2(left), when specific loads are applied to a given domain along with additional boundary displacement constraints, it gives rise to the following problem formulation:

$$\min_{\Theta} \; C(\Theta) = \int_{\Omega_\Theta} \mathbf{f}\cdot\mathbf{u}\,\mathrm{d}\sigma + \int_{\Gamma_T} \mathbf{t}\cdot\mathbf{u}\,\mathrm{d}s, \tag{1a}$$

$$\text{s.t.:} \int_{\Omega_\Theta} \mathbb{E} : \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})\,\mathrm{d}\sigma = \int_{\Omega_\Theta} \mathbf{f}\cdot\mathbf{v}\,\mathrm{d}\sigma + \int_{\Gamma_T} \mathbf{t}\cdot\mathbf{v}\,\mathrm{d}s, \;\forall \mathbf{v}\in\mathcal{U}_{ad}, \tag{1b}$$

$$\mathbf{u} = \overline{\mathbf{u}}, \text{ on } \Gamma_u, \tag{1c}$$

$$\int_{\Omega_\Theta} 1\,\mathrm{d}\sigma \le \overline{V}, \tag{1d}$$

where C is the structural compliance, which measures the total structural strain energy. Lower values of C indicating greater stiffness in the structure. Ω_Θ is the region occupied by the structure, Θ are the optimization variables that control the geometry and topology of the structure. $\mathbf{f}$ is the external force, $\mathbf{t}$ is the traction defined on the Neumann boundary, $\mathbf{u}$ is the displacement field, $\mathbf{v}$ is the corresponding test functions defined on Ω_Θ, $\mathcal{U}_{ad}$ is the space of kinematically admissible displacement fields. ε is the second order linear strain tensor, $\mathbb{E}$ is a fourth order isotropic elasticity tensor that is decided by the Young's Modulus and the Poisson ratio, $\overline{\mathbf{u}}$ is the prescribed displacement on the Dirichlet boundary Γ_u, $\overline{V}$ is the given volume constraint. In the traditional SIMP method, Θ denotes the density of discrete elements. An iterative optimization algorithm is used to obtain the optimal density distribution, like shown in Fig. 2(middle). Subsequently, a series of post-processing operations are applied to obtain the continuous optimal structural representation, as shown in Fig. 2(right). In our algorithm, the continuous nature of SDFs allow for straightforward adjustment to the representation, enabling us to refine the structural features or introduce smoothing operations as desired.

3.2. Implicit representations of structures

Signed Distance Field (SDF) is a widely used representation for geometric models. It assigns a signed distance value to each point in space, indicating the distance between that point and the closest point on the boundary of the structure. The sign of the distance indicates whether the point is located in the interior or exterior of the structure, as depicted in Fig. 3(a). Within the region occupied by the structure, as shown in Fig. 3(b), we can define a continuous signed distance function $\phi_\Theta(\mathbf{x})$ that satisfies:

$$\phi_\Theta^{SDF}(\mathbf{x}) \begin{cases} > 0, & \text{if } \mathbf{x} \text{ is inside,} \\ = 0, & \text{if } \mathbf{x} \text{ is on the boundary,} \\ < 0, & \text{if } \mathbf{x} \text{ is outside.} \end{cases} \tag{2}$$

By extracting the zero level-set of ϕ_Θ, it becomes straightforward to obtain the boundary of the structure. In our algorithm, we utilize a neural network to fit the SDF of the structure, with the network parameters denoted as Θ. One major advantage of this implicit representation method is its decoupling from spatial grids. By inputting the coordinates of any point within the design domain, we can obtain the corresponding SDF value without relying on predefined spatial grids. Theoretically, this allows us to obtain the structure at any desired resolution (Fig. 3(c)). However, it is important to note that the complexity of the structure details depends on the expressive power of the neural network employed. More complex neural networks have the capability to capture intricate structural details. Given that our training dataset was derived from the SIMP method (Section 4.1), which produces density fields of structures, it is essential for us to establish a mutual conversion between the SDF and these density fields. We apply the classic distance transform method [52] to derive the SDF from the density field of the structure. Once we obtain the SDF of the structure, we can convert it into a density field as follows:

$$\phi^D(\phi^{SDF}(\mathbf{x})) = \begin{cases} 0, & \text{if } \phi^{SDF}(\mathbf{x}) > 0, \\ 1, & \text{if } \phi^{SDF}(\mathbf{x}) \le 0. \end{cases} \tag{3}$$

3.3. Persistent homology of cubical complex

Topological Data Analysis (TDA) is an innovative approach that combines topology, geometry, and data science to reveal meaningful patterns and structures within complex data sets. By focusing on the inherent shape and connectivity of the data, TDA provides valuable insights and a deeper understanding of relationships within the data. Simplicial complexes [53] are often used to discrete data space and represent the connection relationships between data points. We use cubical complexes to analyze the topological features of the implicit neural representations of structures. As the structures are represented using signed distance fields, the cubical complexes provide an effective framework for extracting and examining the underlying topological information encoded in these representations.

Definition 3.1 (*Cubical Complex*)**.** A cubical complex C is a finite collection of n-dimensional cubes of the form $[a_1, b_1]\times\cdots\times[a_n, b_n]$, where $a_i, b_i \in \mathbb{R}$ and $a_i \le b_i$, such that the following conditions are satisfied:

1. If a cube $c \in C$, then all faces of c are also in C.
2. If two cubes $c_1, c_2 \in C$ have a non-empty intersection, then their intersection is either a face of c_1 and c_2 or a cube in C.

Intuitively,in a cubical complex based on 2D data, the basic elements consist of points (0-dimensional cubes), line segments (1-dimensional cubes), and squares (2-dimensional cubes). Fig. 4(a) illustrates the binarized representation of the structure using discrete elements. In this representation, elements with density values equal 1 (solid elements), combine to form a cubical complex C_s. We can

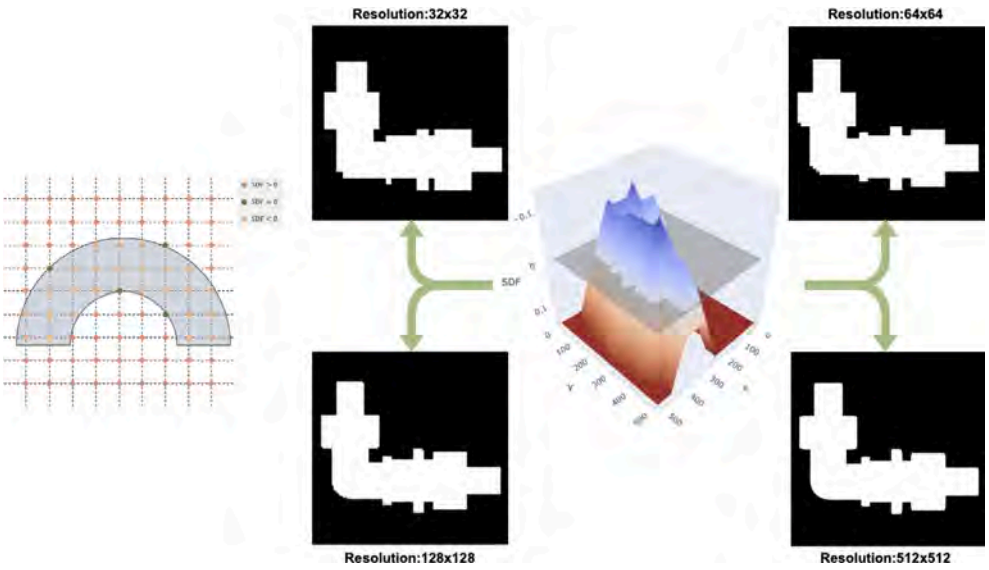


Fig. 3. SDF based structure representation. In SDF, points with SDF < 0 indicate the interior of the structure (color in white), points with SDF = 0 are on the boundary, and points with SDF > 0 are outside the structure (color in black). In theory, since SDF is a continuous function, the structure can be obtained at any desired resolution.

compute the topological features of C_s, namely the ***Betti numbers***. The Betti numbers reflect the number of connected components (0-dimensional Betti number β_0) and the number of holes (1-dimensional Betti number β_1) present in the structure. However, Betti numbers are discrete topological features, which are not differentiable and cannot be directly be used for network training. In order to solve this problem, we use the ***persistent homology*** method to extract the topological features of the structure.

Persistent homology [54] is a powerful mathematical tool used to compute and analyze multi-dimensional topological features in nested simplicial complexes. It allows us to track the evolution of these topological features across all complexes in the superlevel set filtration, as defined in Definition 3.2. Depending on the research question, there are many ways to define filtrations. In our algorithm, we take the SDFs (Fig. 4(b))of the structures as the filtration function assigned to C_s. Then we can define a continuous feature metric, ***persistence***, to analyze the multi-dimensional topological features of the data, as Definition 3.3. We can encode all the information of birth values, death values and persistence of topological features in a ***persistence diagram*** (PD), as shown in Fig. 4(c). Persistence diagram serves as a multi-dimensional shape descriptor that captures all topological features of the structure.

Definition 3.2 (*Superlevel Set Filtration*)**.** Define a *filtration function* $f : C_s \to \mathbb{R}$, it assigns a real value to each element in C_s. The *superlevel set filtration* of f is a nested sequence of complexes $\emptyset \subseteq C_0 \subseteq C_1 \subseteq \cdots \subseteq C_n = C_s$, where $C_i = \{c \in C_s | f(c) \leq \alpha_i\}$, and $\alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_n$.

Definition 3.3 (*Persistence*)**.** For a topological feature β_k, its *persistence* is defined as the difference between the birth and death values of the feature, i.e., $\text{pers}(\beta_k) = |\alpha_i - \alpha_j|$, where birth value α_i and death value α_j correspond to the filtration value when the feature is created and destroyed, respectively. *Persistence pairs*, (α_i, α_j), also depict topological features. Notably, $(\alpha_i, +\infty)$ reflects the existence of a feature that is born at α_i and persists to the end of the filtration.

4. Algorithm

We employ the SIMP method to generate a 2D dataset, producing optimal structures (referred to as Ground truth, GT) that minimize structural compliance under specific conditions, including loads, displacement constraints, and volume constraints. The **key idea** of IF-TONIR is to utilize a Variational Auto-Encoder (VAE) for reconstructing the optimal structures. Through training, the network captures the structural features and encodes them into a low-dimensional latent space. Then we can generate the corresponding optimal structures from this latent space, taking into account boundary information and design domain shape as conditions. Specifically, we use a physics-net to extract features from the boundary conditions and physical fields. This feature vector serves as the condition code, guiding the network to generate structures based on the given problem configurations. By inputting the point coordinates along with the associated properties related to the design domain shapes into the decoder, the network can effectively recognize the design region where materials can be distributed.

4.1. Datasets

Generalizing a well-trained model to arbitrary design domains and constraint conditions can be challenging. Our concern lies in the fact that even for design domains with similar shapes that exhibit slight difference, a new iterative optimization process is often necessary in traditional topology optimization methods. To address this issue, we prepare a dataset where design domains with similar features are categorized into groups, as shown in Fig. 5(a). We train the IF-TONIR on this dataset consisting of optimal structures generated from this group of design domains. As a result, when dealing with problems that involve design domains with similar shapes, the network can directly predict the corresponding optimal structures. In addition, considering that in practical applications, industrial components with the same function often have similar load scenarios despite differences in size, it is not necessary to require the network to generalize to arbitrary load conditions. Therefore, we have set up six types of boundary conditions for each shape design domain (Fig. 5(b)). Each boundary condition includes different load directions and volume constraints. The potential application scenario of IF-TONIR is that for similar-shaped components requiring optimization, the network trained on previously optimal structures can directly generate the optimized solution without the need for iteration. The work [55] provides a compact and efficient Matlab implementation of SIMP, and we employ it to generate the corresponding optimal structures as the ground truth. The specific parameters used to generate the training data set are detailed in Table 1. Please refer to Appendix A a more detailed explanation of the parameters. We will release the data set to the research community, which will serve as a standardized benchmark for future research in the field of topology optimization.

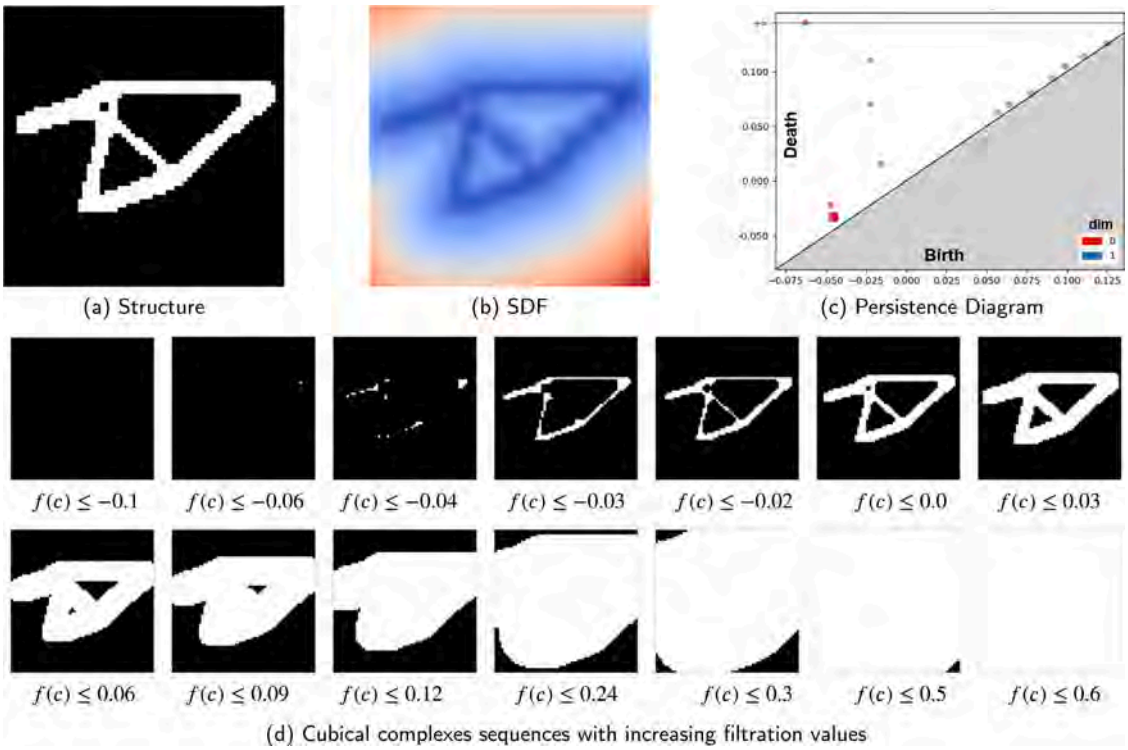


Fig. 4. Illustration of topological features using the persistent homology. (a) The original structure via density representation. (b) The SDF of the structure. (c) The corresponding persistence diagram. (d) The white elements combine to form the corresponding cubical complex at the current filtration value $f(c) \le \alpha_i$.

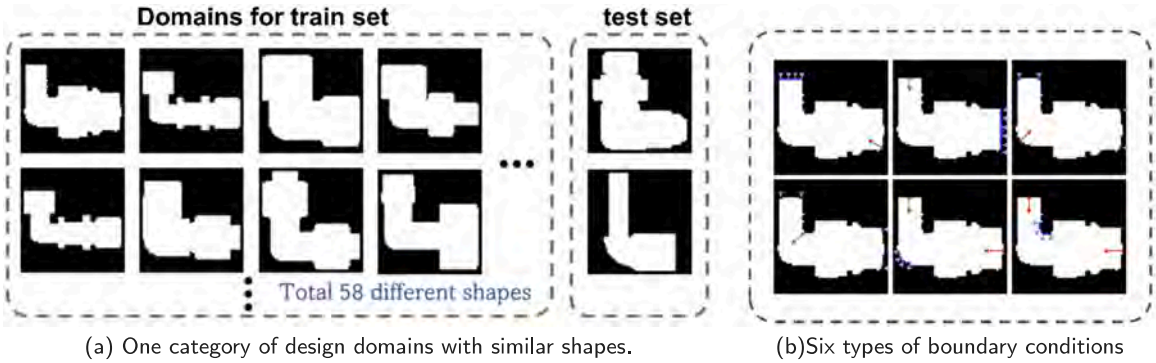


Fig. 5. Dataset preparation. We organized 60 design domains with similar shapes into a single group. From this group, we selected two design domains to generate test samples for evaluating the trained model. We defined six different types of boundary conditions on these design domains.

We observed a strong correlation between the physical response analysis results over the original design domain under the corresponding boundary conditions and the optimal structures. Therefore, in addition to the SDF representation and boundary conditions, we also calculated the stress field and strain energy field of the original structure under the loads and displacement constraints. This additional physical fields serve as supplementary information in the train process and have been verified to be effective in previous works such as TopologyGAN [26] and TopoDiff [27].

4.2. Network architecture

As depicted in Fig. 1, our network architecture mainly comprises two components: an optimal structure reconstruction network based on a Variational Auto-Encoder, and a physics-net responsible for encoding boundary conditions and physical fields as condition codes. For a more detailed explanation of the network architecture, please refer to Appendix B.

4.2.1. Input

In our dataset, the original 2D optimal structures are represented as binary images with a resolution of 64×64, in which elements with '1' (white) represent solid elements, elements with '0' (black) represent void elements. We compute the SDFs of the structures $\{\mathbf{s}_i^s\}_{i=1}^N$ as the input for the VAE network (Fig. 6(a)), where N is the total number of samples. Since VAE is a self-supervised learning method, the SDFs $\mathbf{s}_i^s$ also serve as the GT for the network regression outputs. Additionally, the SDFs of the design domains $\{\mathbf{s}_i^d\}_{i=1}^N$ are appended to the points as properties to identify the shape of the design domain (Fig. 6(b)). As illustrated in Fig. 6(c), we represent the loads and displacement constraints as 2D sparse matrices. In load matrices (both in the x-direction and y-direction), the value of each element is the force value. For the displacement constraints, we represent them using a matrix of the same size. Elements in the matrix with a value of 1 indicate that the corresponding element has a displacement constraint of 0 in both x and y directions, while elements with a value of 0 represent unconstrained elements. The supplementary physical information includes the stress

Table 1
Parameters for generating dataset.

Parameters	Settings
volume constraint	[0.3 : 0.02 : 0.5]
load direction	[0 : $\pi/6$: π]
Resolution	64×64
SIMP penalty	{50, 3, 25, 0.25}
SIMP filter radius	10
Young modulus	1.0
Poisson ratio	0.2

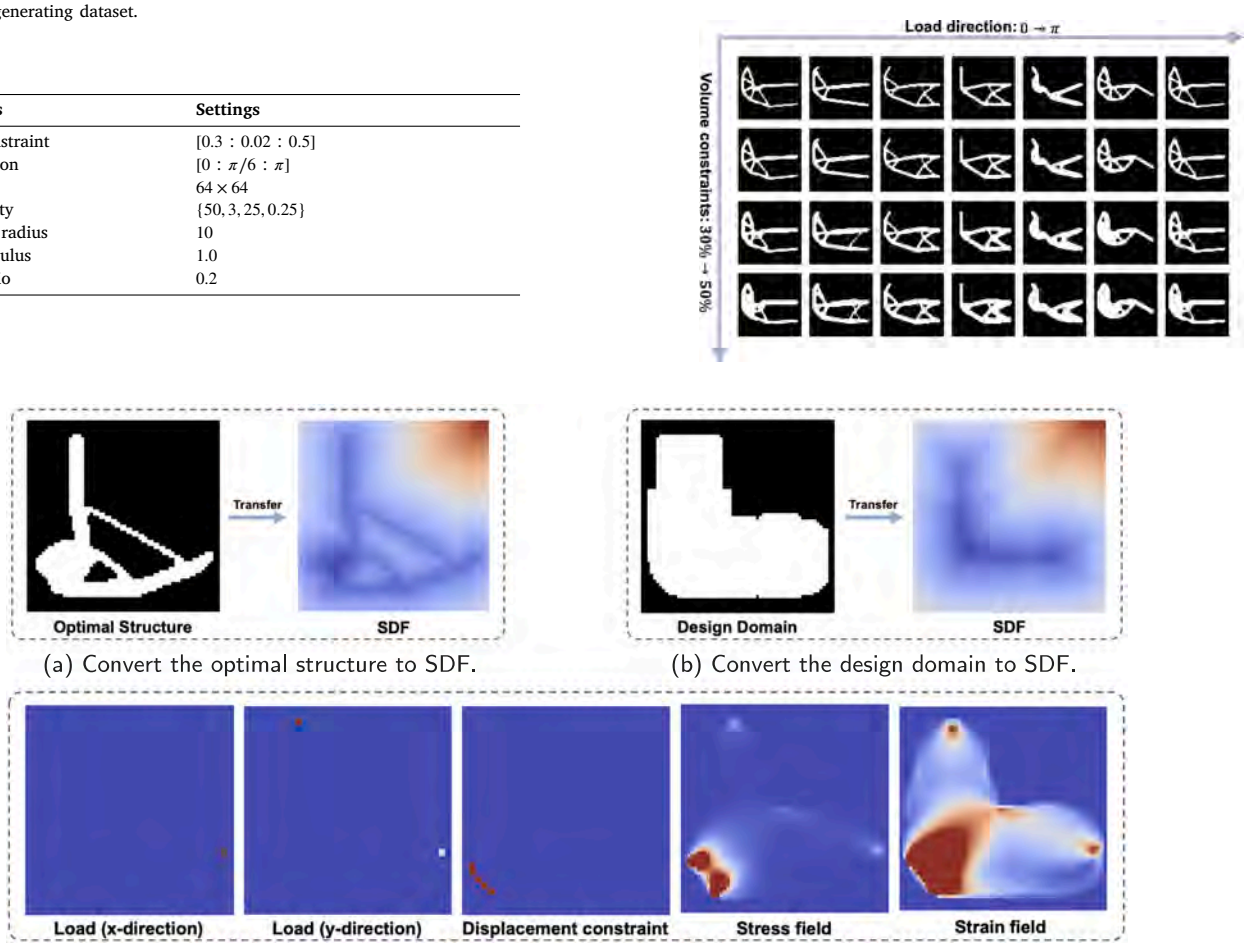


Fig. 6. Input Illustration.

field and strain field of the original structure under specific loads and displacement constraints, which we calculated using SolidsPy [56]. By integrating these five fields, we obtain the physical information samples $\{\mathbf{s}_i^{phy}\}_{i=1}^N$

4.2.2. VAE for structure reconstruction

One of the main objectives of IF-TONIR is to perform prior learning and implicit reconstruction of input optimal structures. To achieve this, we adopt the Variational Auto-Encoder (VAE) network, which is a widely used generative model. The VAE network consists of an encoder and a decoder. In our approach, we utilize the ResNet18 architecture as the encoder. ResNet18 [57] is a well-known convolutional neural network architecture that has shown effectiveness in various image-related tasks. The encoder processes the input SDFs of optimal structures through a series of 2D convolutional layers, extracting features and mapping them to a low-dimensional latent space. Additionally, we also input the condition information $\{\mathbf{s}_i^{phy}\}_{i=1}^N$ into the encoder. The purpose of the encoder ϕ^E is to learn the distribution of a latent code $\mathbf{z}_i$ that can reconstruct the input data $\mathbf{s}_i^s$. Typically, the latent code $\mathbf{z}_i$ is assumed to follow a Gaussian distribution,

$$q_{\phi^E}(\mathbf{z}_i|\mathbf{s}_i^s, \mathbf{s}_i^{phy}) = \mathcal{N}(\mu_{\phi^E}(\mathbf{s}_i^s, \mathbf{s}_i^{phy}), \sigma_{\phi^E}(\mathbf{s}_i^s, \mathbf{s}_i^{phy})), \tag{4}$$

where $\mu_{\phi^E}(\mathbf{s}_i^s)$ and $\sigma_{\phi^E}(\mathbf{s}_i^s)$ are the mean and variance of the latent code, parameterized by the encoder ϕ^E. Then we employ the reparameterization trick [58] to formulate the latent code $\mathbf{z}_i$ as:

$$\mathbf{z}_i = \mu_{\phi^E}(\mathbf{s}_i^s, \mathbf{s}_i^{phy}) + \sigma_{\phi^E}(\mathbf{s}_i^s, \mathbf{s}_i^{phy}) \odot \epsilon, \tag{5}$$

where ϵ is a random variable sampled from a standard Gaussian distribution, $\odot$ is the element-wise product. Following the encoding, we employ a decoder ϕ^D based on a Multi-Layer Perceptron (MLP) to decode the latent code $\mathbf{z}_i$ and reconstruct the SDF of the optimal structure. We sample points within the design domain and feed the coordinates of these points into the decoder. The decoder then returns the signed distance value of each point. In this way, the decoder effectively serves as a signed distance function for the structure. In our training process, we uniformly sample 64×64 points $\{\mathbf{p}_k\}_{k=1}^M$ in the design domain. We train the network by minimizing the reconstruction error between the output SDFs of the decoder $\hat{\mathbf{s}}_i^s$ and the GT input. The specific configuration of the loss functions can be found in Section 4.3.

4.2.3. Conditional information

In order to enable the network to generate structures under various design domain shapes and boundary conditions, it is essential to incorporate conditional information into the network. This is achieved by introducing the prior probability distribution $p(\mathbf{s}_i^s|\mathbf{c})$, where $\mathbf{c}$ represents the conditional information. The conditional information encompasses aspects such as the design domain shapes, loads, displacement constraints, and volume constraint.

Physical conditions. As described in Section 4.2.1, the boundary settings consists of five 2D tensors, namely the load matrices in the x and y directions, the displacement constraint matrix, the stress field and the strain field. We employ a shallow CNN net to effectively represent and encode these boundary information as conditions. The CNN-net takes in

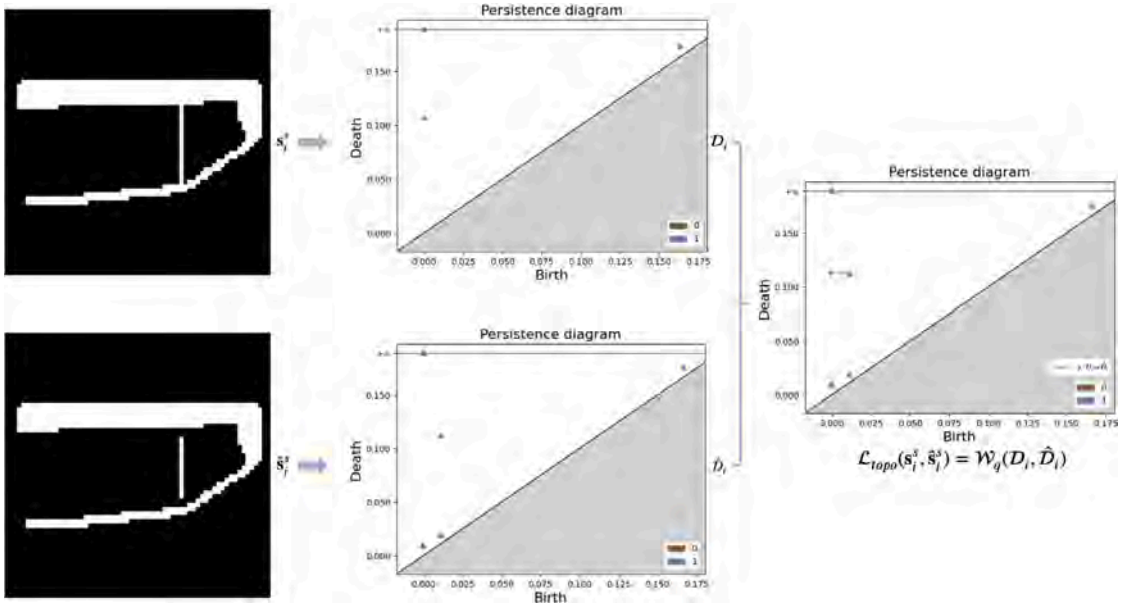


Fig. 7. Illustration of the topological loss $\mathcal{L}_{topo}$: Wasserstein distance between persistence diagrams of output and GT structures.

the boundary tensors $\mathbf{s}_I^{phy}$ as input and processes them through a series of convolutional layers, capturing the relevant features and patterns. This allows the physics-net to generate a latent code representation $\mathbf{c}_i^{phy}$ that encapsulates the essential information of the boundary conditions. By encoding the boundary conditions in a latent code, we enable the network to effectively utilize this information during the generation process, ensuring that the generated structures align with the specified boundary conditions.

Points properties. Since the decoder outputs SDF value based on the coordinates of the input point $\mathbf{p}_k = (x_k, y_k) \in \Omega$ within the design domain, we can include shape information and volume constraint as additional properties appended to the point. The property $\mathbf{s}_i^d(\mathbf{p}_k)$ helps the network recognize the shape of the design domain, thereby establishing the boundary of material distribution. Attaching the volume constraint v_i as property to each point can be regarded as representing the probability density distribution of material presence or absence at that point. Therefore, we input vector $\mathbf{c}_k^{pro} = (x_k, y_k, \mathbf{s}_i^d(\mathbf{p}), v_i)$ associated with each point into the decoder. This allows the network to consider the design domain shape and volume constraint during the generation of the SDF values. Rahaman et al. [59] demonstrated that deep neural networks tend to learn lower frequency functions, which may limit their ability to accurately fit data with high frequency variation. Taking inspiration from NeRF [17], we map the points properties $t \in \mathbb{R}$ into a higher-dimensional space $\mathbb{R}^{2L}$ using an encoding function $\gamma(t)$:

$$\gamma(t) = (\sin(2^0\pi t), \cos(2^0\pi t), \sin(2^1\pi t), \cos(2^1\pi t), \dots, \sin(2^{L-1}\pi t), \cos(2^{L-1}\pi t)), \tag{6}$$

where L is a hyperparameter that determines the frequency of the encoding function. In our experiments, we set $L = 4$ for encoding the coordinates of the input point $\mathbf{p}_k = (x_k, y_k)$, $L = 6$ for encoding $\mathbf{s}_i^d(\mathbf{p}_k)$, and $L = 10$ for encoding the volume constraints. Additionally, you can add loads and displacement constraints as properties to the points, which further enhance the ability of the network to identify the boundary conditions.

In summary, we concatenate the latent code $\mathbf{z}_i$ of the optimal structure, the condition code $\mathbf{c}_i^{phy}$, and the points properties $\mathbf{c}_k^{pro}$. This combined vector is then input to the decoder for structure generation. By incorporating all these components, the network is able to leverage the latent space, condition information, and point properties to generate optimal structures that satisfy the given design domain shapes and conditions.

4.3. Losses and metrics

Similar to classical VAEs, IF-TONIR utilizes a loss function during the training process that consists of the reconstruction loss and the KL divergence loss. However, in order to enhance the accuracy of structure reconstruction, we introduce a topological loss in addition to the conventional geometric losses. This topological loss is designed to ensure that the generated structures possess the same topology as the GT structures.

4.3.1. Geometric losses

The geometric losses measure the regression error between the SDF values $\hat{\mathbf{s}}_i^s$ predicted by the network and the GT SDF values $\mathbf{s}_i^s$ of structures. Notably, the shape of a structure is primarily determined by its boundary, while the values of off-boundary points are less important. To prioritize the accuracy of points near the boundary, we employ the clamped L_1-loss function [60] defined in Eq. (7). This allows IF-TONIR to effectively capture the structural details and preserving the desired shape characteristics.

$$\mathcal{L}_{clamp}(\hat{\mathbf{s}}_i^s, \mathbf{s}_i^s) = \frac{1}{M}\sum_{k=1}^{M} \mid \chi(\hat{\mathbf{s}}_i^s(\mathbf{p}_k), \tau) - \chi(\mathbf{s}_i^s(\mathbf{p}_k), \tau) \mid, \tag{7}$$

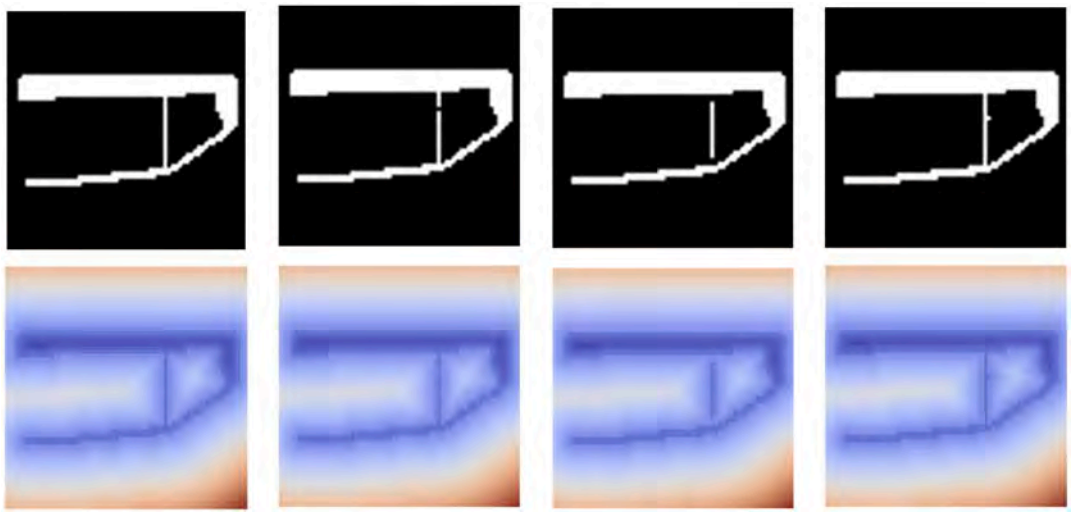
where M is the total number of sample points in domain, $\chi(x, \tau) = \min(\tau, \max(-\tau, x))$.

To enforce the volume constraints on the output structures, we incorporate a volume loss term that quantifies the discrepancy between the predicted structure and the target structure in terms of their volumes. We calculate the volume of a structure from its SDF representation using the following integral equation [61]:

$$V(\mathbf{s}) = \int_{\Omega} H(-\mathbf{s}(\mathbf{p}))\, \mathrm{d}A, \tag{8}$$

where Ω is the entire design domain, $\mathrm{d}A$ is the area element, $H(\cdot)$ is the Heaviside function,

$$H(x) = \begin{cases} 1, & \text{if } x < -\eta, \\ \frac{3}{4}(\frac{x}{\eta} - \frac{x^3}{3\eta^3}) + \frac{1}{2}, & \text{if } -\eta \le x \le \eta, \\ 0, & \text{if } x > \eta. \end{cases} \tag{9}$$



Measures	Structure a	Structure b	Structure c	Structure d
$\mathcal{L}_{geo}$ $(vs.\ a)$	0	0.00024	0.00098	0.00024
β_0	1	1	2	1
β_1	1	0	0	1
$\mathcal{L}_{topo}(vs.\ a)$	0	0.03125	0.03886	0.00937
$\mathcal{E}_{comp}(vs.\ a)$	-	21.6%	69.7%	0.5%

Fig. 8. Comparison of the topological loss and geometric loss. We take structure (a) as the benchmark original structure. Structure (b) represents a configuration missing one solid element, while structure (c) is a configuration missing two solid elements. Conversely, structure (d) represents a configuration with an additional solid element. The table below shows the measure of geometric loss, topological loss and compliance error for structures (b-d) relative to structure (a).

Then we define the volume loss as follows:

$$\mathcal{L}_{vol}(\hat{\mathbf{s}}_i^s, \mathbf{s}_i^s) = \mid V(\hat{\mathbf{s}}_i^s) - V(\mathbf{s}_i^s) \mid / V(\mathbf{s}_i^s). \tag{10}$$

The geometric loss is the sum of the clamped L_1-loss, MSE loss and volume loss:

$$\mathcal{L}_{geo}(\hat{\mathbf{s}}_i^s, \mathbf{s}_i^s) = \mathcal{L}_{clamp}(\hat{\mathbf{s}}_i^s, \mathbf{s}_i^s) + \mathcal{L}_{vol}(\hat{\mathbf{s}}_i^s, \mathbf{s}_i^s). \tag{11}$$

Experimentally, we set $\tau = 0.05, \eta = 0.02$ in our experiments.

4.3.2. Topological loss

Geometric losses focus on local information, while topological loss captures global structural features. As shown in Fig. 8(b), small structural defects can lead to disconnections in the structure, causing significant loss in structural compliance. However, these errors may only lead to small values in geometric losses. By incorporating topological loss, IF-TONIR can effectively detect and measure tiny structural disconnections.

To obtain differentiable topological features of the structure s, we utilize persistent homology analysis as described in Section 3.3 to derived its corresponding persistence diagram $\mathcal{D}$. As shown in Fig. 7, we define the topological loss $\mathcal{L}_{topo}$ as the *Wasserstein distance* between the persistence diagrams $\mathcal{D}_i$ and $\hat{\mathcal{D}}_i$ obtained from the GT and the network prediction structures, respectively:

$$\mathcal{L}_{topo}(\mathbf{s}_i^s, \hat{\mathbf{s}}_i^s) = \mathcal{W}_q(\mathcal{D}_i, \hat{\mathcal{D}}_i) = \left(\inf_{\gamma : \mathcal{D}_i \to \hat{\mathcal{D}}_i} \sum_{p \in \mathcal{D}_i} \|p, \gamma(p)\|^q \right)^{1/q}, \tag{12}$$

where $\|\cdot\|$ is the L_2 norm, γ is a bijection between $\mathcal{D}_i$ and $\hat{\mathcal{D}}_i$, and q is a hyperparameter that controls the sensitivity of the topological loss. In our experiments, we set $q = 2$. We calculate the topological loss by using the *GUDHI* library [62]. To enhance efficiency, we employed parallel processing for acceleration. As shown in Fig. 8, the topological loss proves to be effective in capturing structural defects and disconnections compared to the geometric loss. Structure (a) is a single connected ($\beta_0 = 1$) structure with one hole ($\beta_1 = 1$). Compared to structure (a), structure (b) is disconnected due to the absence of one solid element, resulting in the elimination of the hole ($\beta_1 = 0$) while retaining the single connected component ($\beta_0 = 1$). Despite the fact that the structural features have undergone significant changes (high structural compliance error $\mathcal{E}_{comp}$), the absence of one element still results in low geometric loss, while the topological loss exhibits a significant response. Because of the lack of two solid elements, structure (c) consists of two disconnected components ($\beta_0 = 2$). Similarly, the topological loss reveals a greater response than the relatively modest geometric loss. For comparison, we set up structure (d) with an additional solid element whose topological features remain consistent with the structure (a). For structure (d), the geometric loss, topological loss, and compliance error all maintain a relatively low level. Therefore, we employ the following reconstruction loss to train the deep neural network, enabling the network to reconstruct the structures more accurately.

$$\mathcal{L}_{recon}(\mathbf{s}_i^s, \hat{\mathbf{s}}_i^s) = \lambda_1 \mathcal{L}_{geo}(\mathbf{s}_i^s, \hat{\mathbf{s}}_i^s) + \lambda_2 \mathcal{L}_{topo}(\mathbf{s}_i^s, \hat{\mathbf{s}}_i^s), \tag{13}$$

where we set the weights $\lambda_1 = 0.7, \lambda_2 = 0.3$ in our experiments. The application of topological loss to some extent ensures the solid of the output structures.

In addition to the reconstruction loss, VAEs also need the KL divergence loss to constrain the learned latent space distribution to match a predefined prior distribution [58]. It encourages the learned distribution to match the prior distribution, promoting the model to capture the essential features of the data. The KL divergence loss is defined as:

$$\mathcal{L}_{\mathrm{KL}} = -\frac{1}{2} \sum_{i=1}^{N} (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \tag{14}$$

where μ_i and σ_i represent the mean and standard deviation of the learned latent distribution for the ith data sample, and N is the total

number of data samples. Minimizing the KL divergence loss encourages the latent space to be close to a multivariate Gaussian distribution with zero mean and unit variance, ensuring better sample generation and latent space interpolation properties. Thereby, we use the total loss $\mathcal{L}_{total} = \mathcal{L}_{recon} + \mathcal{L}_{KL}$ to train the IF-TONIR.

4.3.3. Evaluation metrics

In addition to the loss functions stated above, we also use the following accuracy metrics to assess the performance of IF-TONIR.

Reconstruction accuracy. To evaluate the accuracy of the network in structural reconstruction, we use the Intersection over Union (IoU) metric defined as follows to quantify the difference between the GT structure and the network output, which is commonly used in object detection [63],

$$\mathcal{E}_i^{IoU} = \frac{\| \Omega_{\hat{\mathbf{s}}_i^s} \cap \Omega_{\mathbf{s}_i^s} \|}{\| \Omega_{\hat{\mathbf{s}}_i^s} \cup \Omega_{\mathbf{s}_i^s} \|}, \tag{15}$$

where $\Omega_{\hat{\mathbf{s}}_i^s}$ and $\Omega_{\mathbf{s}_i^s}$ are the regions occupied by $\hat{\mathbf{s}}_i^s \le 0$ and $\mathbf{s}_i^s \le 0$, respectively. $\|\cdot\|$ denotes the area of the region.

Physical accuracy. In order to assess the validity of the objective and volume constraint of the original optimization problem, we further analyze the error of the relative volume and structural compliance,

$$\mathcal{E}_i^{vol} = \frac{|V(\hat{\mathbf{s}}_i^s) - V(\mathbf{s}_i^s)|}{V(\mathbf{s}_i^s)}, \tag{16}$$

$$\mathcal{E}_i^{comp} = \frac{|C(\hat{\mathbf{s}}_i^s) - C(\mathbf{s}_i^s)|}{C(\mathbf{s}_i^s)} \tag{17}$$

where $V(\mathbf{s}_i^s) = \int_{\Omega_{\mathbf{s}_i^s}} 1 \mathrm{d}A$, $C(\mathbf{s}_i^s)$ is defined in Eq. (1).

4.4. Training and generating

Our IF-TONIR algorithm demonstrates that, once the neural network has been sufficiently trained and converged, we can directly generate the optimal structure with low compliance under volume constraints through an iteration-free method. We outline the training procedure of IF-TONIR in Algorithm 1. Based on VAE network, we embed the SDFs of optimized structures into low dimensional latent space. By sampling from the learned latent space, we can generate new structures. To guide the generation process, we attach design domain shapes, loads, and displacement constraints to spatial coordinates as properties, along with the conditional information obtained from the physics-net. Then we can get the optimized structures using the trained decoder. The detailed generation process is shown in Algorithm 2.

Algorithm 1: Training of IF-TONIR.

Input: GT SDFs of structures: $\{\mathbf{s}_i^s\}_{i=1}^N$; Design domains: $\{\mathbf{s}_i^d\}_{i=1}^N$; Physical fields: $\{\mathbf{s}_i^{phy}\}_{i=1}^N$; Volume constraints: $\{v_i\}_{i=1}^N$; Sample points: $\{\mathbf{p}_k = (x_k, y_k)\}_{k=1}^M$

Output: The optimal network parameters $\Theta^* = \{\Theta_{vae}^*, \Theta_{phy}^*\}$.

Initialize the network $\Phi_{vae}^E, \Phi_{vae}^D, \Phi_{phy}, \lambda_1, \lambda_2$;
while not converged **do**
  Divide the dataset into M batches $\{\mathcal{T}_i\}_{i=1}^M$;
  foreach *batch* $\mathcal{T}_i$ **do**
    Initialize $\mathcal{L}_{geo}^i = 0, \mathcal{L}_{topo}^i = 0$;
    foreach *sample* $\mathbf{s}_j$ *in* $\mathcal{T}_i$ **do**
      Compute the structural latent code: $\mathbf{z}_j = \Phi_{vae}^E(\mathbf{s}_j^s)$;
      Compute points properties vector: $\mathbf{c}_k^{prop} = [\gamma(x_k), \gamma(y_k), \gamma(\mathbf{s}_i^d), \gamma(v_i)]$;
      Compute the condition code: $\mathbf{c}_j^{phy} = \Phi_{phy}(\mathbf{s}_j^{phy})$;
      Concatenate the latent codes: $\mathbf{l}_j = [\mathbf{z}_j, \mathbf{c}_k^{prop}, \mathbf{c}_j^{phy}]$;
      Compute the reconstructed SDF: $\hat{\mathbf{s}}_j^s = \Phi_{vae}^D(\mathbf{l}_j)$;
      Evaluate the geometric loss: $\mathcal{L}_{geo}^i = \mathcal{L}_{geo}^i + (\mathcal{L}_{clamp}(\hat{\mathbf{s}}_j^s, \mathbf{s}_j^s) + 0.1 * \mathcal{L}_{mse}(\hat{\mathbf{s}}_j^s, \mathbf{s}_j^s))$;
      Compute the PDs through persistent homology: $\mathcal{D}_j, \hat{\mathcal{D}}_j$
      Evaluate the topological loss: $\mathcal{L}_{topo}^i = \mathcal{L}_{topo}^i + \mathcal{W}_q(\hat{\mathcal{D}}_j, \mathcal{D}_j)$;
      Evaluate the reconstruction loss: $\mathcal{L}_{recon}^i = \lambda_1 \mathcal{L}_{geo}^i + \lambda_2 \mathcal{L}_{topo}^i$;
    Evaluate the total loss function: $\mathcal{L}_{total}^i = \mathcal{L}_{recon}^i + \mathcal{L}_{KL}$;
    Update Θ by $\frac{\partial \mathcal{L}_{total}^i}{\partial \Theta}$;

Algorithm 2: Generating optimal structures via IF-TONIR.

Input: Design domain: $\mathbf{s}^d$; Boundary conditions: Loads and displacement constraints; Volume constraints: v; Sample points within the design domain: $\{\mathbf{p}_k\}_{k=1}^M$; Optimal network parameters: $\Theta^* = \{\Theta_{vae}^*, \Theta_{phy}^*\}$.

Output: The optimal structure $\mathbf{s}^*$ with the minimum compliance under the given volume constraint v

Step 1: Sample the shape latent code from standard Gaussian distribution $\mathbf{z} \sim \mathcal{N}(0, 1)$;
Step 2: Compute points properties vector: $\mathbf{c}_k^{prop} = [\gamma(x_k), \gamma(y_k), \gamma(\mathbf{s}^d), \gamma(v)]$;
Step 3: Compute the physical fields $\mathbf{s}^{phy}$ according to the boundary conditions;
Step 4: Compute the physical latent code: $\mathbf{c}^{phy} = \Phi_{phy}(\mathbf{s}^{phy}; \Theta_{phy}^*)$;
Step 5: Compute the volume constraint code: $\gamma(v)$;
Step 6: Concatenate the latent codes: $\mathbf{l} = [\mathbf{z}, \mathbf{c}_k^{prop}, \mathbf{c}^{phy}]$;
Step 7: Generate the SDF of optimal structure: $\mathbf{s}_s^* = \Phi_{vae}^D(\mathbf{l}; \Theta_{vae}^*)$;
Step 8: Extract the structure from the SDF: $\mathbf{s}^*$;

5. Experiments

5.1. Hyperparameters and convergence

Our training dataset on the design domains shown in Fig. 5 contains 26796 samples. For every epoch, we randomly selected 21437 samples (80%) for training and 5359 samples (20%) for validation. We evaluate the performance of the trained network on the test dataset with 924 samples on the unseen design domains. We train our model according to Algorithm 1 on a machine with one Nvidia A40 GPU. We employ the Adam optimizer with an initial learning rate of 0.001. We reduce the learning rate by a factor of 0.5 every 20 epochs. It is very time-consuming to take structural compliance as a loss or accuracy metric during training process. Although TopoDiff [27] introduced a regression model to predict the compliance of the generated structures, it requires additional data preparation and network training. Therefore, we monitor the training process using the IoU metric. Incorporating physical quantities supervision will be part of our future work. Fig. 9 illustrates the stable convergence of the loss and IoU accuracy during the training. It can be observed that the generated structures from the network gradually become more reasonable and robust as the training progresses. It requires around 150 epochs to yield a prediction result with IoU accuracy about 94%.

5.2. Evaluation

After completing the training, we evaluate the accuracy of the generated structures in terms of structural compliance and volume. As shown in Fig. 11, IF-TONIR successfully predicts structures that closely resemble the GT samples from the training dataset. It accurately identifies the shapes of the design domains and the locations of the applied loads. On the entire training dataset, IF-TONIR achieves an average compliance error of 6.90% and an average volume error of 1.32%. To demonstrate the effectiveness of IF-TONIR, we evaluate its performance

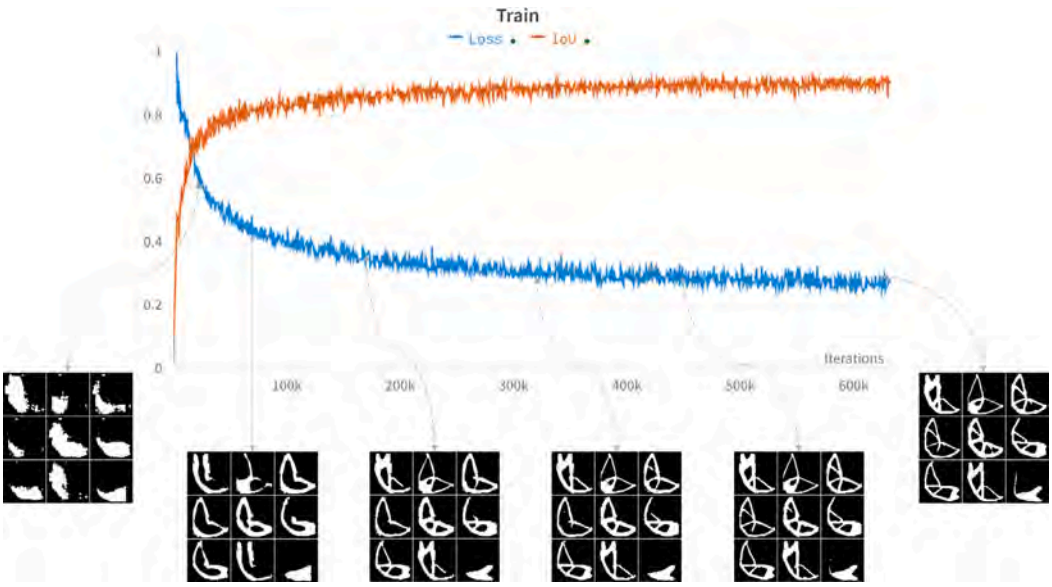
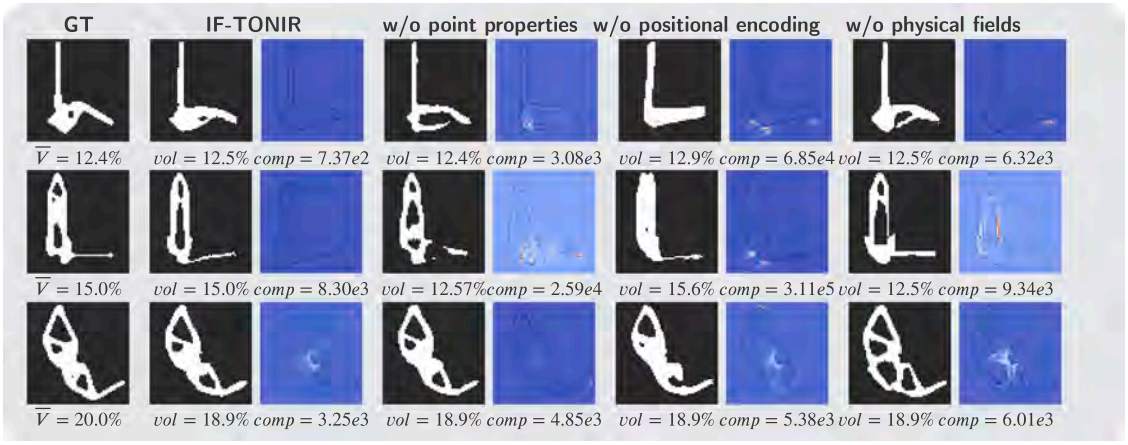


Fig. 9. Training process of IF-TONIR.



Metrics	IF-TONIR	w/o point properties	w/o positional encoding	w/o physical fields
Average ϵ^{comp}	54.2%	296.9%	114.8%	208.4%
Average ϵ^{vol}	4.02%	5.85%	5.33%	5.20%

Fig. 10. Comparative results illustrating the importance of local network architecture. Columns 2–3 show the optimized structure and corresponding strain energy field of the complete network framework; Columns 4–5 depict the results without point properties; Columns 6–7 present the results without positional encoding; Columns 8–9 display the results after removing stress and strain energy fields from the physical information.

on the test dataset, shown in Fig. 12. The average compliance error on the test dataset is 54.2%, while the average volume error is 4.02%. It is worth noting that the test dataset includes two design domain shapes that were not observed during the training process. Therefore, the boundary condition positions in the test dataset are totally different from those in the training dataset. To the best of out knowledge, our algorithm is the first to consider the design domain shape as part of generalization capabilities.

5.3. Ablation studies

Topological loss. The structural discontinuity often leads to instability or high compliance, Fig. 13 illustrates the effectiveness of the topological loss in improving the structural continuity of the generated structures. Because we utilize the SDFs of structures as the filtration value for computing persistent homology, the resulting PD can effectively capture the geometric and topological features of the data. The continuous filtration value fields make the training process more stable. It allows the structures generated by the network to closely resemble the GT samples.

Points properties. To determine the SDF value at a specific point, we utilize design shape information and volume constraints as point properties, in conjunction with coordinates. These point properties can also include load locations and displacement constraints. Our approach differs from previous methods as we incorporate volume constraints as part of the point properties in the decoder, instead of duplicating them to match the input tensor size. Control experiments conducted without point properties show their crucial role, as illustrated in Fig. 10. We enhance the impact of point properties in our algorithm by applying high-frequency transformations to their values, specifically position encoding as Eq. (6). Ablation experiments conducted to verify the importance of position encoding show that the position encoding layer can help the network predict more accurate volumes and force application

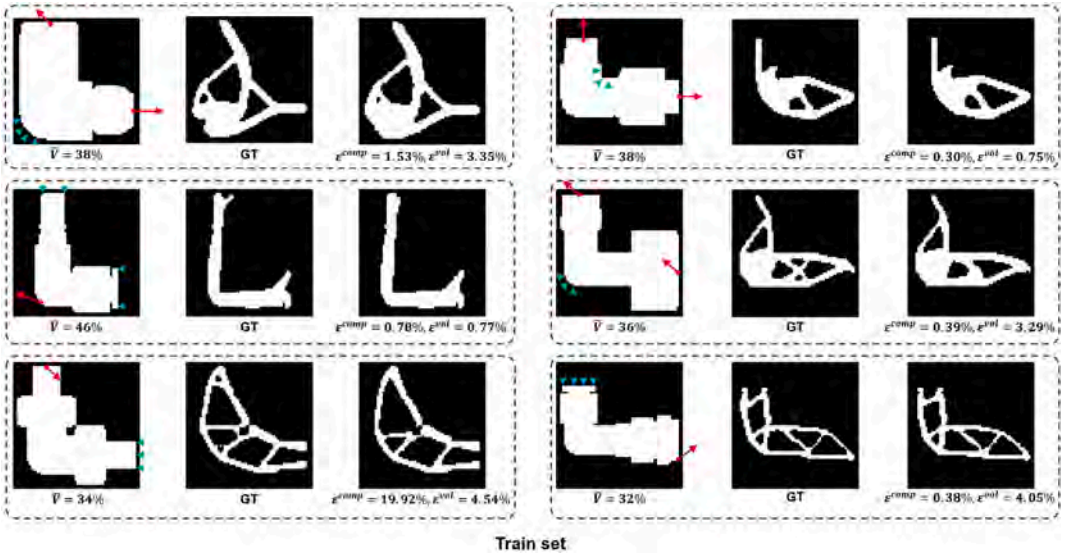


Fig. 11. Evaluation results on train dataset. We compare the generated structures with their corresponding GT samples in term of compliance and volume.

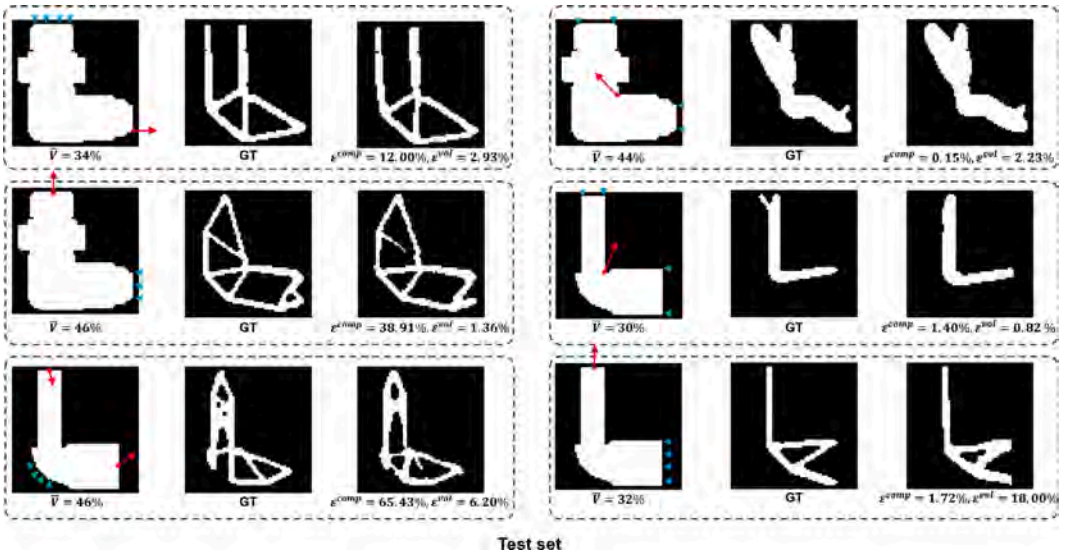


Fig. 12. Evaluation results on test dataset.

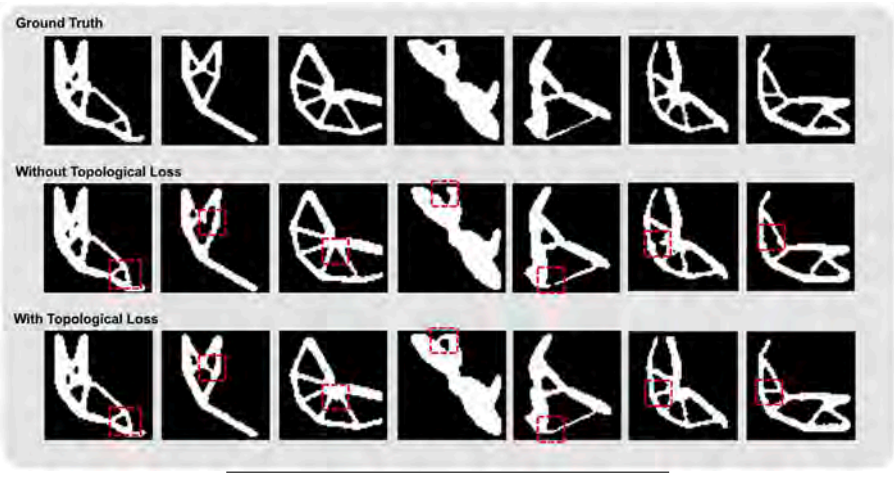
positions. The use of position encoding results in an average compliance that is almost one-sixth of that without position encoding, indicating a higher level of structural strength. This is demonstrated in several examples shown in Fig. 10.

Physical fields. The physical field information on the original design domain has been previously demonstrated to play a positive role in the prediction of optimal structures. To illustrate its significance to our algorithm, as shown in Fig. 10, we present the adverse effects after removing the stress and strain energy fields from the physical information.

5.4. Comparison

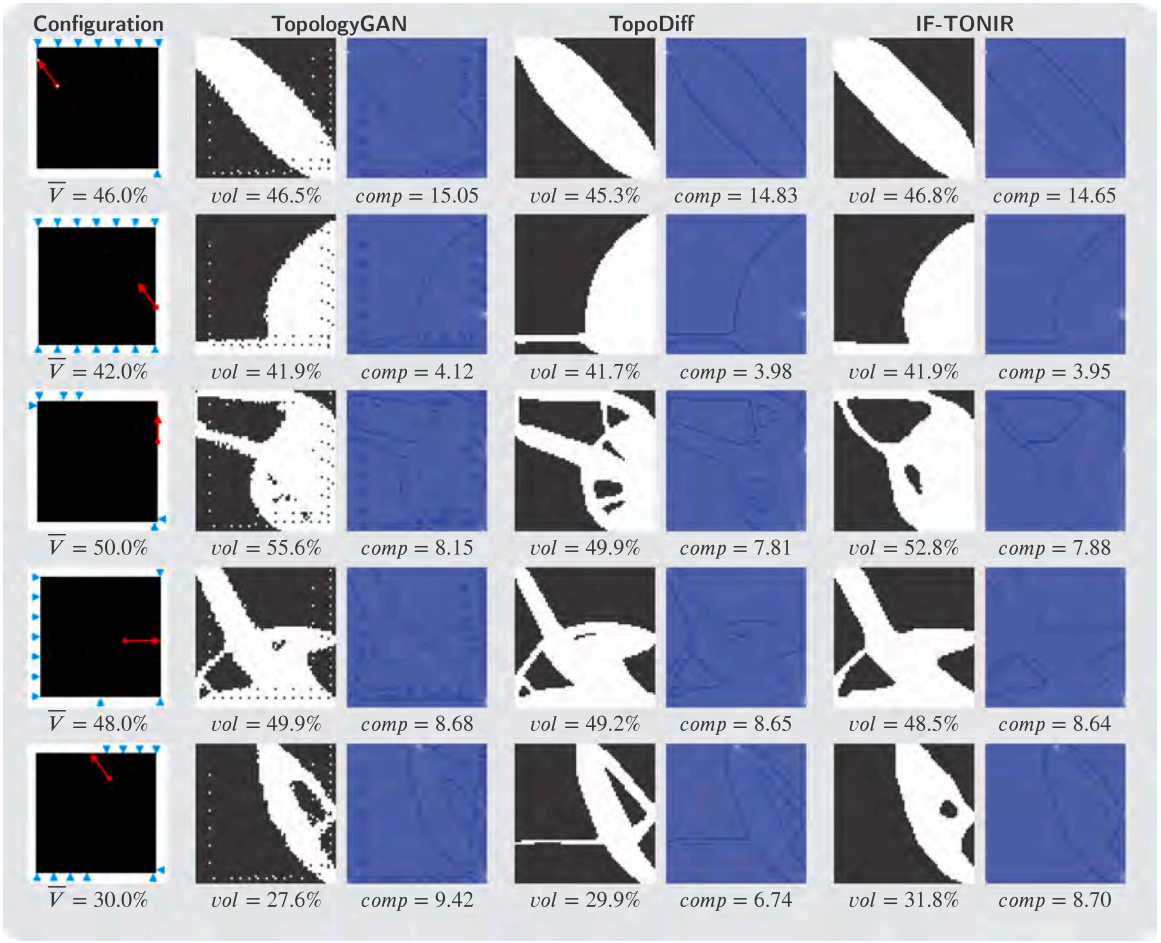
We conduct a comparative analysis of IF-TONIR with other generative models, namely TopologyGAN [26] and TopoDiff [27]. Notably, these previous work did not consider the design domain shape. To ensure a fair comparison, we evaluated all three models on the same dataset provided by TopoDiff. As the structures in this dataset were generated on a fixed regular design domain, we only need to remove the domain-related information from IF-TONIR to train and evaluate it on the provided dataset. Fig. 14 demonstrates the comparison results between IF-TONIR, TopologyGAN, and TopoDiff.

As one of the most popular generative models, diffusion models have been proven to have better capability in capturing fine details in image or shape generation, as we see in Fig. 14. Additionally, TopoDiff framework incorporated three sub-networks to enhance the strength and continuity of the generated structures. Therefore, from the table presented in Fig. 14, it can be observed that TopoDiff currently achieve the best performance. However, it should be noted that TopoDiff has the highest complexity among the compared models and requires more inference time for structure generation. TopoDiff reported in their work that it takes 21.59 s to generate one structure, while IF-TONIR only requires 0.09 s (on a CPU). Moreover, extending diffusion models to the 3D domain poses challenges in terms of computational efficiency and memory utilization. In contrast, IF-TONIR achieves better results than GAN models with a lighter network architecture. It is well known that train a GAN model is a difficult task. Our framework also demonstrates the ability to generalize across design domains with different shapes, which was not considered in previous works. However, similar to other



Experiments	ϵ^{comp} (Train)	ϵ^{vol} (Train)	ϵ^{comp} (Test)	ϵ^{vol} (Test)
w/o $\mathcal{L}_{topo}$	10.21%	1.90%	61.17%	4.64%
w/ $\mathcal{L}_{topo}$	6.96%	1.72%	54.64%	4.28%

Fig. 13. Comparison of network output structures without and with topological loss.



Metrics	TopologyGAN	TopoDiff	IF-TONIR
Average ϵ^{comp}	48.51%	4.39%	13.17%
Average ϵ^{vol}	11.87%	2.56%	2.55%

Fig. 14. We compare the test results of IF-TONIR with TopologyGAN and TopoDiff. Each sample displays the structure output by the corresponding method and its strain energy field. Our method accurately identifies regions with concentrated strain and effectively avoids checkerboard phenomenon. It outperforms TopologyGAN in terms of structural compliance using a lighter network architecture. And IF-TONIR effectively eliminates the checkerboard phenomenon.

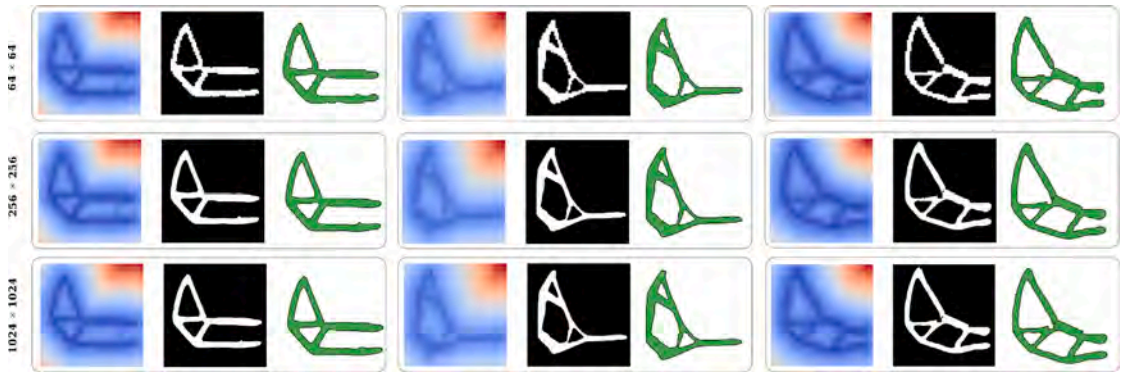


Fig. 15. TONIR has the ability to generate structures at any desired resolution by increasing the number of sampled points within the design domain.

VAE models, IF-TONIR still has limitations in generating structures with fine details, which is an area of focus for future work.

Resolution. TopologyGAN and TopoDiff require the input and output resolutions to be consistent, which poses challenges when generating high-resolution structures. This requires either preparing a new dataset with high-resolution samples or employing techniques such as super-resolution. However, due to the utilization of implicit neural representation in IF-TONIR, we have the flexibility to obtain structure at any desired resolution in theory. As shown in Fig. 15, the trained network allows us to generate structures with higher resolution by increasing the number of sampled points within the design domain. Remarkably, IF-TONIR can generate structures with a resolution of 1024 on a CPU in just around 10 s. Compared to traditional SIMP methods, our method eliminates the need for post-processing to obtain a smooth and continuous structural representation. The point we need to clarify is that the resolution here refers to the resolution of the final structural representation, not the resolution of the structural features. Enhancing the resolution of structural representation does not add more structural details.

6. Conclusions

In this paper, we propose IF-TONIR, a novel iteration-free topology optimization. Our algorithm utilizes implicit neural representation to directly generate optimized structures from given conditions and problem configurations. Through experiments conducted on the test dataset, IF-TONIR demonstrates its effectiveness in generating optimized structures for various similar-shaped design domains, achieving low structural compliance. To the best of our knowledge, this is the first end-to-end work that generalizes the topology optimization process to different design domain shapes. The utilization of implicit neural representation decouples the network from spatial grids, enabling structure generation at any desired resolution theoretically. Besides, this smooth and compact representation method effectively avoids the checkerboard phenomenon.

There are several directions that need further research in future work for IF-TONIR. Firstly, in this paper, we only focus on the 2D problem. Upon successfully addressing the demanding challenges of efficiency and memory storage, IF-TONIR has the potential to expand its application to solve more practical problems in the 3D domain. In particular, the computation of the topological loss used in our paper is very time-consuming for 3D problems. Secondly, although IF-TONIR achieved better performance compared to GANs using a lightweight network architecture, the VAEs used in our approach still have limitations in reconstructing fine structural details. Future work could explore the use of improved generative models to enhance the reconstruction capabilities of IF-TONIR.

Table 2
CNN network architecture for extracting physical fields features.

CNN				
Layer	Kernel	Stride	Padding	Channel
2D Conv Batch Norm LeakyReLU	3 × 3	1	1	5 → 64
Max Pool	2	1	1	64 → 64
2D Conv Batch Norm LeakyReLU	3 × 3	1	1	64 → 64
Max Pool	2	1	1	64 → 64
2D Conv Batch Norm LeakyReLU	3 × 3	1	1	64 → 32
Max Pool	2	1	1	32 → 32
2D Conv Batch Norm LeakyReLU	3 × 3	1	1	32 → 32
Max Pool	2	1	1	32 → 32
2D Conv Batch Norm LeakyReLU	3 × 3	1	1	32 → 32
Max Pool	2	1	1	32 → 32
Output: 1 × 128				

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge the support provided by the Ministry of Education, Singapore, under its Academic Research Fund Grants (MOE-T2EP20220-0005 & RT19/22), the National Key R&D Program of China under Grant (2021YFA1003003), LiaoNing Revitalization Talents Program (2022RG04), Fundamental Research Funds for the Central Universities (DUT22QN212).

Appendix A. Parameters for data set

When preparing the data set, we fix the material properties for each shape in a class of design domains. In our experimental settings, we set the Young modulus and Poisson ratio to 1.0 and 0.2, respectively. We

Table 3
Decoder based on MLP for generating optimized structures from latent space, where dim_z, dim_c, dim_p are the dimension of latent code, condition, points properties, respectively.

MLP-based decoder			
Layer	Input dim	Output dim	Dropout
Fully Connected	$dim_z + dim_c + dim_p$	512	0.0
Fully Connected	512	512	0.0
Batch Norm			
Fully Connected	512	512	0.1
Batch Norm			
Fully Connected	512	512	0.0
Batch Norm			
Fully Connected	512	512	0.1
Batch Norm			
Fully Connected	512+dim_p	512	0.0
Batch Norm			
Fully Connected	512	512	0.0
Batch Norm			
Fully Connected	512	512	0.1
Batch Norm			
Fully Connected	512+dim_p	512	0.0
Batch Norm			
Fully Connected	512	1	0.0
tanh			

take the volume constraint and the direction of force of each optimization problem as variables. We then use the conventional method [55] to iteratively obtain the optimization results, which serve as the ground truth for training. In this work, a continuous gradient penalty scheme is provided to reduce the checkerboard phenomenon in the SIMP method, allowing for optimization structures with clear boundaries. Moreover, compared to a fixed penalty coefficient, it can yield structures with lower compliance. Therefore, we also adopted the SIMP penalty setting parameters $\{50, 3, 25, 0.25\}$ provided in the article to generate optimized structures for our optimization problems.

Appendix B. Network architecture

Here we list the network architecture details for IF-TONIR. We employ a shallow CNN net (as shown in Table 2 to extract features from boundary settings and physical fields as conditions. For VAE, we use ResNet18 [64] as the encoder and an MLP as the decoder (as shown in Table 3).

References

[1] Sigmund O, Maute K. Topology optimization approaches: A comparative review. Struct Multidiscip Optim 2013;48(6):1031–55.
[2] Bendsøe MP, Sigmund O. Material interpolation schemes in topology optimization. Arch Appl Mech 1999;69:635–54.
[3] Xie YM, Steven GP. A simple evolutionary procedure for structural optimization. Comput Struct 1993;49(5):885–96.
[4] Wang MY, Wang X, Guo D. A level set method for structural topology optimization. Comput Methods Appl Mech Engrg 2003;192(1–2):227–46.
[5] Guo X, Zhang W, Zhong W. Doing topology optimization explicitly and geometrically—A new moving morphable components based framework. J Appl Mech 2014;81(8).
[6] Zhang W, Chen J, Zhu X, Zhou J, Xue D, Lei X, et al. Explicit three dimensional topology optimization via moving morphable void (MMV) approach. Comput Methods Appl Mech Engrg 2017;322:590–614.
[7] Mukherjee S, Lu D, Raghavan B, Breitkopf P, Dutta S, Xiao M, et al. Accelerating large-scale topology optimization: State-of-the-art and challenges. Arch Comput Methods Eng 2021;28(7):4549–71.
[8] Cang R, Yao H, Ren Y. One-shot generation of near-optimal topology through theory-driven machine learning. Comput Aided Des 2019;109:12–21.
[9] Kollmann HT, Abueidda DW, Koric S, Guleryuz E, Sobh NA. Deep learning for topology optimization of 2D metamaterials. Mater Des 2020;196:109098.
[10] Nie Z, Jung S, Kara LB, Whitefoot KS. Optimization of part consolidation for minimum production costs and time using additive manufacturing. J Mech Des 2020;142(7):072001.
[11] Chandrasekhar A, Suresh K. TOuNN: Topology optimization using neural networks. Struct Multidiscip Optim 2021;63(3):1135–49.
[12] Hoang V-N, Nguyen N-L, Tran DQ, Vu Q-V, Nguyen-Xuan H. Data-driven geometry-based topology optimization. Struct Multidiscip Optim 2022;65(2):69.
[13] Woldseth RV, Aage N, Bærentzen JA, Sigmund O. On the use of artificial neural networks in topology optimisation. Struct Multidiscip Optim 2022;65(10):294.
[14] Li B, Huang C, Li X, Zheng S, Hong J. Non-iterative structural topology optimization using deep learning. Comput Aided Des 2019;115:172–80.
[15] Behzadi MM, Ilieş HT. Real-time topology optimization in 3D via deep transfer learning. Comput Aided Des 2021;135:103014.
[16] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 165–74.
[17] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV. 2020.
[18] Xie Y, Takikawa T, Saito S, Litany O, Yan S, Khan N, et al. Neural fields in visual computing and beyond. In: Computer graphics forum, Vol. 41. (2):Wiley Online Library; 2022, p. 641–76.
[19] Xu B, Zhang J, Lin K-Y, Qian C, He Y. Deformable model-driven neural rendering for high-fidelity 3D reconstruction of human heads under low-view settings. 2023, arXiv:2303.13855.
[20] Edelsbrunner H, Harer J, et al. Persistent homology-A survey. Contemp Math 2008;453(26):257–82.
[21] Yu Y, Hur T, Jung J, Jang IG. Deep learning for determining a near-optimal topological design without any iteration. Struct Multidiscip Optim 2019;59(3):787–99.
[22] Chandrasekhar A, Suresh K. TOuNN: Topology optimization using neural networks. Struct Multidiscip Optim 2021;63:1135–49.
[23] Regenwetter L, Nobari AH, Ahmed F. Deep generative models in engineering design: A review. J Mech Des 2022;144(7):071704.
[24] Guo T, Lohan DJ, Cang R, Ren MY, Allison JT. An indirect design representation for topology optimization using variational autoencoder and style transfer. In: 2018 AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference. 2018, p. 0804.
[25] Sharpe C, Seepersad CC. Topology design with conditional generative adversarial networks. In: International design engineering technical conferences and computers and information in engineering conference, Vol. 59186. American Society of Mechanical Engineers; 2019, V02AT03A062.
[26] Nie Z, Lin T, Jiang H, Kara LB. TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain. J Mech Des 2021;143(3):1–12.
[27] Mazé F, Ahmed F. Diffusion models beat GANs on topology optimization. In: Proceedings of the AAAI conference on artificial intelligence. 2023.
[28] Jang S, Yoo S, Kang N. Generative design by reinforcement learning: enhancing the diversity of topology optimization designs. Comput Aided Des 2022;146:103225.
[29] Asanuma J, Doi S, Igarashi H. Transfer learning through deep learning: Application to topology optimization of electric motor. IEEE Trans Magn 2020;56(3):1–4.
[30] Kallioras NA, Kazakis G, Lagaros ND. Accelerated topology optimization by means of deep learning. Struct Multidiscip Optim 2020;62(3):1185–212.
[31] Xie Y, Takikawa T, Saito S, Litany O, Yan S, Khan N, et al. Neural fields in visual computing and beyond. In: Computer graphics forum, Vol. 41. (2):Wiley Online Library; 2022, p. 641–76.
[32] Chen Z, Zhang H. Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 5939–48.
[33] Sitzmann V, Chan E, Tucker R, Snavely N, Wetzstein G. Metasdf: Meta-learning signed distance functions. Adv Neural Inf Process Syst 2020;33:10136–47.
[34] Zheng X, Liu Y, Wang P, Tong X. SDF-stylegan: Implicit SDF-based stylegan for 3D shape generation. In: Computer graphics forum, Vol. 41. (5):Wiley Online Library; 2022, p. 52–63.
[35] Zhang B, Tang J, Niessner M, Wonka P. 3Dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. 2023, arXiv preprint arXiv:2301.11445.
[36] Lyu Z, Wang J, An Y, Zhang Y, Lin D, Dai B. Controllable mesh generation through sparse latent point diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 271–80.
[37] Zhang Z, Yao W, Li Y, Zhou W, Chen X. Topology optimization via implicit neural representations. Comput Methods Appl Mech Engrg 2023;411:116052.
[38] Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA. A roadmap for the computation of persistent homology. EPJ Data Sci 2017;6:1–38.
[39] Pun CS, Lee SX, Xia K. Persistent-homology-based machine learning: A survey and a comparative study. Artif Intell Rev 2022;55(7):5169–213.
[40] Hensel F, Moor M, Rieck B. A survey of topological machine learning methods. Front Artif Intell 2021;4:681108.
[41] Hofer C, Kwitt R, Niethammer M, Uhl A. Deep learning with topological signatures. In: Advances in neural information processing systems, Vol. 30. 2017.

[42] Moor M, Horn M, Rieck B, Borgwardt K. Topological autoencoders. In: International conference on machine learning. PMLR; 2020, p. 7045–54.
[43] Gabrielsson RB, Nelson BJ, Dwaraknath A, Skraba P. A topology layer for machine learning. In: International conference on artificial intelligence and statistics. PMLR; 2020, p. 1553–63.
[44] Zhang Y, Yao J, Wang Y, Chen C. On the convergence of optimizing persistent-homology-based losses. 2022, arXiv preprint arXiv:2206.02946.
[45] Liu W, Guo H, Zhang W, Zang Y, Wang C, Li J. TopoSeg: Topology-aware segmentation for point clouds. IJCAI; 2022.
[46] de Surrel T, Hensel F, Carrière M, Lacombe T, Ike Y, Kurihara H, et al. Ripsnet: A general architecture for fast and robust estimation of the persistent homology of point clouds. In: Topological, algebraic and geometric learning workshops 2022. PMLR; 2022, p. 96–106.
[47] Byrne N, Clough JR, Montana G, King AP. A persistent homology-based topological loss function for multi-class CNN segmentation of cardiac MRI. In: Statistical atlases and computational models of the heart. M&Ms and EMIDEC challenges: 11th international workshop, STACOM 2020, held in conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised selected papers 11. Springer; 2021, p. 3–13.
[48] Waibel DJ, Atwell S, Meier M, Marr C, Rieck B. Capturing shape information with multi-scale topological loss terms for 3d reconstruction. In: Medical image computing and computer assisted intervention–MICCAI 2022: 25th International conference, Singapore, September 18–22, 2022, Proceedings, Part IV. Springer; 2022, p. 150–9.
[49] Behzadi MM, Ilieş HT. Gantl: Toward practical and real-time topology optimization with conditional generative adversarial networks and transfer learning. J Mech Des 2022;144(2).
[50] Le C, Norato J, Bruns T, Ha C, Tortorelli D. Stress-based topology optimization for continua. Struct Multidiscip Optim 2010;41:605–20.
[51] Takezawa A, Yoon GH, Jeong SH, Kobashi M, Kitamura M. Structural topology optimization with strength and heat conduction constraints. Comput Methods Appl Mech Engrg 2014;276:341–61.
[52] Maurer CR, Qi R, Raghavan V. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. IEEE Trans Pattern Anal Mach Intell 2003;25(2):265–70.
[53] Rotman JJ. An introduction to algebraic topology, Vol. 119. Springer Science & Business Media; 2013.
[54] Edelsbrunner H, Letscher D, Zomorodian A. Topological persistence and simplification. In: Proceedings 41st annual symposium on foundations of computer science. IEEE; 2000, p. 454–63.
[55] Ferrari F, Sigmund O. A new generation 99 line matlab code for compliance topology optimization and its extension to 3D. Struct Multidiscip Optim 2020;62:2211–28.
[56] Guarín-Zapata N, Gómez J. Solidspy: 2D-finite element analysis with python. 2020, URL https://github.com/AppliedMechanics-EAFIT/SolidsPy.
[57] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, p. 770–8.
[58] Kingma DP, Welling M, et al. An introduction to variational autoencoders. Found Trends® Mach Learn 2019;12(4):307–92.
[59] Rahaman N, Baratin A, Arpit D, Draxler F, Lin M, Hamprecht F, et al. On the spectral bias of neural networks. In: International conference on machine learning. PMLR; 2019, p. 5301–10.
[60] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 165–74.
[61] Van Dijk N, Langelaar M, Van Keulen F. Explicit level-set-based topology optimization using an exact heaviside function and consistent sensitivity analysis. Internat J Numer Methods Engrg 2012;91(1):67–97.
[62] Maria C, Boissonnat J-D, Glisse M, Yvinec M. The gudhi library: Simplicial complexes and persistent homology. In: Mathematical software–ICMS 2014: 4th International congress, Seoul, South Korea, August 5-9, 2014. Proceedings 4. Springer; 2014, p. 167–74.
[63] Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S. Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 658–66.
[64] Wightman R. Pytorch image models. 2019, http://dx.doi.org/10.5281/zenodo.4414861, https://github.com/rwightman/pytorch-image-models.