Title	Topology optimization with text-guided stylization
Author(s)	Zhong, Shengze; Punpongsanon, Parinya; Iwai, Daisuke et al.
Citation	Structural and Multidisciplinary Optimization. 2023, 66(12), p. 256
Version Type	VoR
URL	https://hdl.handle.net/11094/93539
rights	This article is licensed under a Creative Commons Attribution 4.0 International License.
Note	

RESEARCH PAPER

Topology optimization with text-guided stylization

Shengze Zhong[1] · **Parinya Punpongsanon**[1] · **Daisuke Iwai**[1] · **Kosuke Sato**[1]

Abstract
We propose an approach for the generation of topology-optimized structures with text-guided appearance stylization. This methodology aims to enrich the concurrent design of a structure's physical functionality and aesthetic appearance. Users can effortlessly input descriptive text to govern the style of the structure. Our system employs a hash-encoded neural network as the implicit structure representation backbone, which serves as the foundation for the co-optimization of structural mechanical performance, style, and connectivity, to ensure full-color, high-quality 3D-printable solutions. We substantiate the effectiveness of our system through extensive comparisons, demonstrations, and a 3D-printing test.

Keywords Topology optimization · Stylization · Text-guidance · Neural field

1 Introduction

Topology optimization is a mathematical method that automatically designs structures with optimal performance under physical boundary conditions and constraints (Rozvany 2009). Its utilization has progressively extended into engineering domains like automotive engineering (Yang and Chahande 1995) and aerospace engineering (Zhu et al. 2016; Aage et al. 2017), where the demand for high-performance structures is paramount. With the advancement of topology optimization algorithms, users can now generate high-quality structures by manipulating a small number of parameters, such as material properties and manufacturing costs (Kazi et al. 2017; Chen et al. 2018; Ma et al. 2021). Consequently, even individuals with limited expertise can effectively wield optimization tools following brief training (Nobel-Jørgensen et al. 2016). This simplicity of structure design and reduced demand for user expertise have facilitated the wide-ranging application of topology optimization across domains, including garment design (Zhang and Kwok 2019) and the development of musical instruments (Yu et al. 2013; Li et al. 2016).

The advent of commercially available topology-optimized products has sparked an upsurge in the desire for visually appealing designs. The appearance of a product holds considerable sway over user preferences, with aesthetic and symbolic characteristics assuming dominant roles (Creusen and Schoormans 2005). However, designing structures that balance both performance and appearance remains a challenging task, even for seasoned designers. We refer to this task as stylized topology optimization.

Due to the difficulty in building a comprehensive and differentiable description of the structure style, previous investigations into stylized topology optimization have primarily relied on texture-based approaches to guide the visual aesthetics of the structure (Martínez et al. 2015; Hu et al. 2019; Navez et al. 2022). These methods locally apply geometric features to the structure that align with the user-provided texture, albeit at the expense of a holistic stylization perspective. Furthermore, they necessitate the laborious task of manually designing textures, thus compromising user convenience. On the other hand, Loos et al. creatively introduced a general evaluation criterion, unity-in-variety, to assess structural style (Loos et al. 2022). This approach has demonstrated its ability to improve user aesthetic preference while requiring manually adjusted structures.

We are motivated to address the aforementioned issues by exploring a comprehensive and differentiable structural stylization evaluation metric, for bringing enhanced stylization expressiveness. Inspired by the recent success of large-scale image-text neural networks in content generation (Radford

---

Responsible Editor: Jun Wu.

✉ Shengze Zhong
zhong@sens.sys.es.osaka-u.ac.jp

1   Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan

et al. 2021), we present a novel approach to topology optimization that incorporates text-guided stylization. By inputting the desired appearance's descriptive text alongside the physical boundary conditions, this method can generate mechanically optimized, full-color stylized, and 3D-printable structures.

Our research object can be succinctly viewed as a multi-objective optimization problem of an implicitly neural represented structure (Xie et al. 2022). To this end, we employ a hash-encoded neural network (Müller et al. 2022) to encode coordinates into color and density, effectively capturing both the topology and appearance of the structure. This approach offers a superior representation of structural high-frequency details and demonstrates a faster convergence rate when compared to methods relying on Fourier-featured positional encoding (Tancik et al. 2020; Sitzmann et al. 2020). Subsequently, we leverage a pre-trained image-text neural network called CLIP (Radford et al. 2021) to evaluate the latent feature similarity between the appearance image of the structure and the user's textual description. This evaluation guides the stylization process, facilitating effective control over the desired appearance. Furthermore, we conduct structure connectivity optimization based on connected component labeling (He et al. 2017) to ensure the structure can be 3D-printed in one piece. In the experiments, we analyzed the structural mechanical performance in benchmark tasks (Valdez et al. 2017), showcased various stylized structures, presented the structure stylization control achievable with this research, and concluded with a 3D-printing test.

In summary, we present a text-driven stylized topology optimization method. It employs more user-friendly text-based guidance for the appearance design of a diverse range of topology-optimized structures, and for the first time, takes into account the overall style of the structure, in full-color.

2 Related works

2.1 Topology optimization methods

In mechanical engineering, topology optimization is a method that maximizes structural performance by reallocating the spatial distribution of materials (Sigmund and Maute 2013). Its applications include a range of areas, such as enhancing the flexibility of structures (Bruggi and Duysinx 2012), adjusting the natural vibration frequency (Tsai and Cheng 2013), and optimizing heat conduction (Dbouk 2017). Based on the representation of structures, topology optimization methods can be categorized into explicit and implicit, with representative algorithms being solid isotropic material with penalization (SIMP) (Andreassen et al. 2011), bi-directional evolutionary structural optimization (BESO) (Huang and Xie 2009), and level-set-based methods (Wang

et al. 2003; Zhang et al. 2016). Recently, the advent of implicit neural representation (INR) in topology optimization (Chandrasekhar and Suresh 2021; Woldseth et al. 2022) has enabled novel functionalities, such as arbitrary resolution sampling or solution space generation (Zehnder et al. 2021; Zhong et al. 2022).

Regardless of the representation form of the structure, the primary flow of topology optimization involves projecting the structure onto a finite-element mesh for mechanical performance analysis and then back-propagating the gradient of mechanical performance to update the representation of the structure. This optimization process typically yields a flat solution space, wherein multiple local optima coexist for the given boundary conditions (Sigmund and Petersson 1998). Exploiting this characteristic, we are afforded an ample computational realm to introduce stylized design aspects to the structural appearance.

2.2 Stylized topology optimization

On this basis, previous works have explored various stylized topology optimization methods with texture guidance, in order to enhance the structural aesthetics. Martinez et al. (2015) proposed using exemplars as a guide to stylize 2D topology-optimized structures, endowing the structures with features resembling the exemplar. Their work entailed deriving the first derivative of the structural similarity to the exemplar and re-formulating the multi-objective optimization problem to balance mechanical performance and appearance. Subsequently, Hu et al. (2019) introduced a texture-guided generative structural design method that simultaneously generates a series of stylized structures based on textures. Navez et al. (2022) recently extended these efforts from 2D to 3D, with enhanced local stylization control. Additionally, Loos et al. (Loos et al. 2022) analyzed the aesthetics of topology-optimized structures using the principle of unity-in-variety in industrial design and proposed a simulation for improving the design. These studies have well demonstrated the potential and applications of stylizing topology-optimized structures.

However, there is still no widely accepted structural style evaluation metric in the field of topology optimization, which is due to the highly abstract and complex nature of structural aesthetics and styles. As a consequence, previous investigations have concentrated on specific structural characteristics, such as local geometric patterns, in an attempt to establish quantifiable style evaluation metrics. However, these approaches often sacrifice the overall expressiveness of stylization. Moreover, highly specialized style metrics lead to the lacking of stylization controllability. Furthermore, they have not fully considered the connectivity of 3D-printed structures. Our study addresses these limitations.

2.3 Text-guided generation

The challenge of stylized topology optimization lies in establishing a differentiable and objective evaluation criterion of structural appearance. Recently, data-driven methods show a feasible solution. For instance, Chen et al. proposed a neural network that bridges shape and its human-evaluated aesthetics. After training, the network prediction of shape aesthetics could be leveraged to guide the beautification of novel input shapes (Chen and Lau 2022).

The proposed text-guided structure stylization method is further inspired by the recently prominent text-guided AI drawing and modeling (Frans et al. 2021; Rombach et al. 2022; Jain et al. 2022; Poole et al. 2022), where text, one of the most common and expressive mediums, is utilized to guide the stylization and creation. This technique generally relies on a neural network trained on huge datasets of image-text pairs, to create a multi-modal relationship in between. Then the network estimates the cross-modal similarity between the generated object (often rendered as an image) and the user-input text description and performs optimization. For example, Kevin et al. utilized CLIP guidance to generate drawings based on text input (Frans et al. 2021). Michel et al. proposed a Text2Mesh system that optimizes the position and color of mesh vertices through CLIP guidance, thereby generating 3D objects that conform to the textual description (Michel et al. 2022). Though previous researches show prominent simplicity and expressiveness in shape creation, those involving physical properties (e.g., mechanical performance, connectivity of 3D-printed structure) have not been fully explored.

To address this disparity, we present a novel topology optimization method enriched with text-guided stylization. Our approach enables the generation of visually captivating and structurally robust designs without the need for arduous geometric editing. It also featured much stronger convergence than prevalent Fourier-featured methods (Chandrasekhar and Suresh 2021), and further ensured 3D-printing capability through our introduction of connectivity constraints. Through comprehensive experiments, we explore the trade-off between structural mechanical performance and aesthetics and demonstrated the structure stylization expressiveness and controllability in various applications.

3 Proposed method

In this study, we present a problem formulation as a multi-objective optimization scenario. Initially, we adopt a hash-encoded neural network as a means to implicitly represent a structure S. Subsequently, we concurrently assess the mechanical performance, aesthetic style, and connectivity aspects of S. Finally, we update the structure S utilizing the

gradients associated with these three objectives. The complete optimization workflow is shown in Fig. 1.

In the three subsections of Sect. 3, we provide a sequential account of the methodologies employed to compute each objective function as outlined below:

- Sec. 3.1: Compute the mechanical performance $\mathcal{L}_{\mathrm{mech}}$ of the structure through density-based topology optimization.
- Sec. 3.2: Compute the aesthetic style $\mathcal{L}_{\mathrm{sem}}$ of the structure by the image-text neural network CLIP.
- Sec. 3.3: Compute the connectivity $\mathcal{L}_{\mathrm{conn}}$ of the structure by applying the Connected Component Labeling algorithm.

The integration of the three objects is accomplished through the utilization of the penalty method, a technique that converts a constrained optimization problem into an unconstrained form, as shown in Eq. 1. Here, the penalty factors α and β are assigned to govern the semantic and connectivity losses of the structure, respectively.

$$\mathcal{L} = \mathcal{L}_{mech} + \alpha \mathcal{L}_{\mathrm{sem}} + \beta \mathcal{L}_{\mathrm{conn}} \tag{1}$$
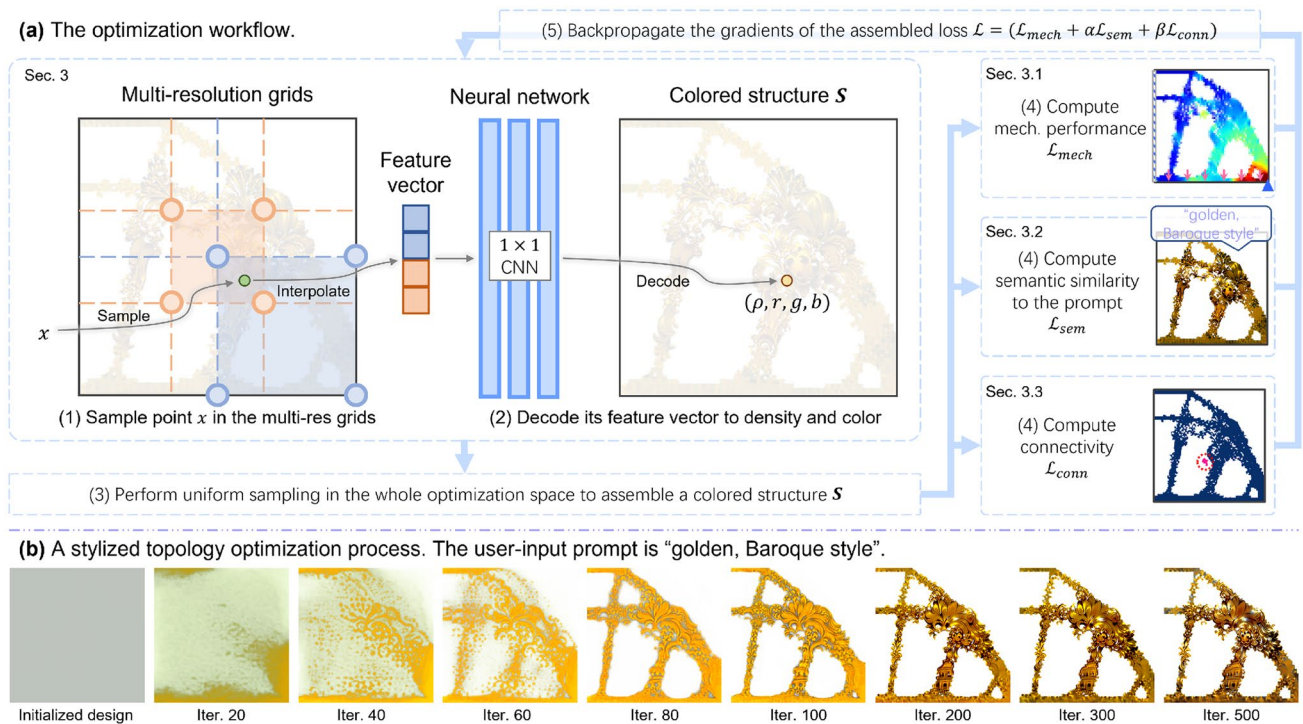
We employ the Adam optimizer (Kingma and Ba 2014) with a decreasing learning rate to ensure convergence. The gradients of the loss function $\mathcal{L}$ are back-propagated to the hash-encoded network, encompassing both the hashed grid features and the network's weights, as depicted in Fig. 1. This completes a single iteration of the structure optimization.

We contemplate the selection of structural representation methods from the following perspectives. Firstly, with regard to the explicit and implicit depiction of the structure, we have opted for the latter in order to acquire a more adaptable design space. The utilization of implicit representation allows for the interpolation of the structure to higher resolutions. Moreover, by employing different resolutions for computing the objective functions (e.g., conducting FEM analysis at low resolution while optimizing style at high-resolution), we can enhance computational efficiency.

Secondly, among the various implicit representation methods, we have selected neural networks to approximate the implicit representation of the structure. This choice enables us to achieve a higher degree of structural expressiveness. Conversely, alternative implicit methods (Wein et al. 2020) like level sets (Wang et al. 2003) or moving morphable components (Zhang et al. 2016) tend to simplify the structural representation by assuming the structure is composed of basic elements. Such simplifications conflict with our objective of ensuring a rich and expressive representation of the structure's appearance.

Lastly, in our approach of utilizing neural networks to implicitly represent the structure, we have adopted a hybrid

(a) The optimization workflow.

(5) Backpropagate the gradients of the assembled loss $\mathcal{L} = (\mathcal{L}_{mech} + \alpha\mathcal{L}_{sem} + \beta\mathcal{L}_{conn})$

(1) Sample point x in the multi-res grids   (2) Decode its feature vector to density and color

(3) Perform uniform sampling in the whole optimization space to assemble a colored structure S

Sec. 3.1 (4) Compute mech. performance $\mathcal{L}_{mech}$

Sec. 3.2 (4) Compute semantic similarity to the prompt $\mathcal{L}_{sem}$

Sec. 3.3 (4) Compute connectivity $\mathcal{L}_{conn}$

(b) A stylized topology optimization process. The user-input prompt is "golden, Baroque style".

Initialized design | Iter. 20 | Iter. 40 | Iter. 60 | Iter. 80 | Iter. 100 | Iter. 200 | Iter. 300 | Iter. 500

Fig. 1 The system overview. **a** The overall optimization workflow. First, multi-resolution hashed grids are used to store and interpolate the coordinate-dependent features of a structure S. Second, these features are decoded to the density ρ and colors r, g, b using a tiny neural network. Third, through uniform sampling in the grids, a colored structure S is obtained. Fourth, we compute the structure's mechanical performance, semantic similarity to the user-input stylization prompt, and connectivity. Finally, these optimization objects are assembled as a single loss $\mathcal{L}$, and its gradients are backpropagated to both the grids and neural network to update the structure representation. Therefore, the three optimization objects are simultaneously optimized during each iteration. **b** The optimization process of a Bridge structure with a stylization prompt "*golden, Baroque style*". The optimization starts from a randomly initialized structure, and converges to a stable topology taking around 100 iterations

representation technique, namely the *instant neural graphics primitives* (Müller et al. 2022). Specifically, we store the spatial features of the structure in multi-resolution hashed grids, which are subsequently decoded into color and density using a neural network. Within this framework, the neural network is exclusively responsible for feature decoding and does not need to store the structural features within its network weights, as seen in previous works (Chandrasekhar and Suresh 2021). Therefore, we can employ a compact neural network for this purpose. This approach significantly enhances the convergence speed of the optimization process compared to previous studies and effectively preserves high-frequency details in the structure.

In this framework, the multi-resolution hashed grids encode the input coordinate x into a feature vector h_x, while the neural network Φ decodes h_x into the structural parameters S_x, as Eq. 2.

$$S_x = \Phi(h_x) \tag{2}$$

The multi-resolution hashed grids consist of L layers of two-dimensional grids, each with a resolution of N_l, as shown

in Eq. 3. Here, l is the layer index, $N_{\min}$ and $N_{\max}$ are the coarsest and finest layer resolution, respectively. Within each layer, the encoded coordinate x is looked up from an independent hash table θ_l with T entries and F dimensions. The multi-resolution hashed grids encompass a total of $L \times T \times F$ parameters.
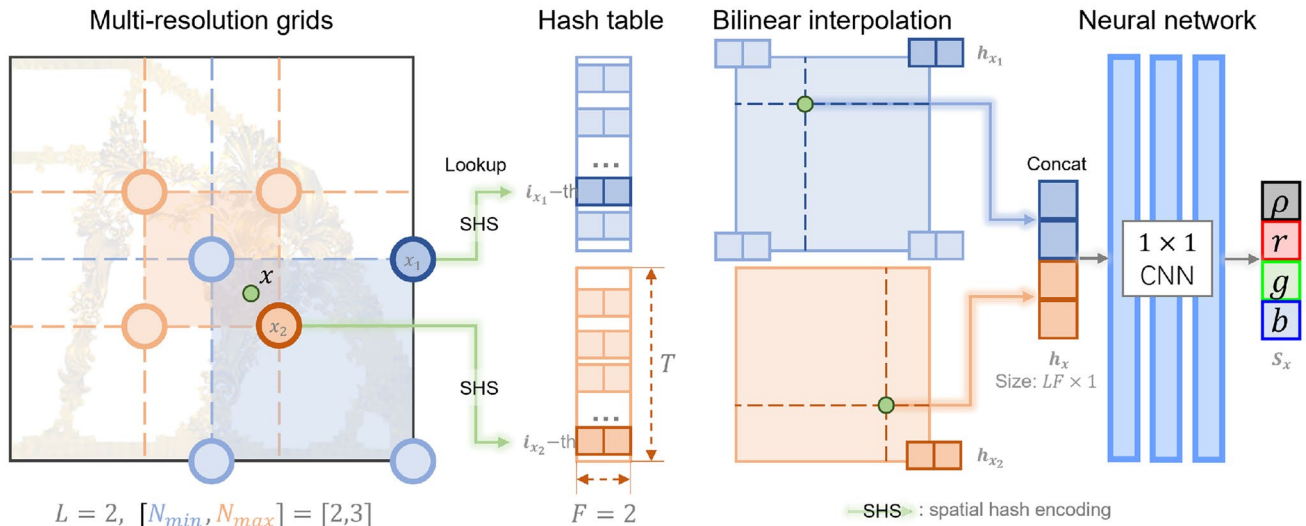
$$N_l = \lfloor N_{\min} \cdot b^l \rfloor$$
$$b = \exp\left(\frac{\ln N_{\max} - \ln N_{\min}}{L-1}\right) \tag{3}$$

Figure 2 depicts the encoding process from the input coordinate x to output S_x in a multi-resolution grid, with its two layers illustrated in blue and orange. Consider the encoding process within the lth layer of the grids: firstly, we find the four corner vertices coordinates x_l around the input point coordinate x. Next, x_l are scaled by N_l and rounded down, then subjected to spatial hash encoding function (Hamming 1952) to obtain integer indices as shown in Eq. 4.

$$i_{x_l} = (x_l \bigoplus \pi) \bmod T, \tag{4}$$

Multi-resolution grids   Hash table   Bilinear interpolation   Neural network

$$L = 2, \ [N_{min}, N_{max}] = [2,3] \qquad\qquad F = 2$$

SHS : spatial hash encoding

Fig. 2 The Multi-resolution hash encoding process, where an input coordinate x is encoded to a four-dimensional output S_x. Initially, we encode the corner vertices of the element containing x, and employ the encoded values as indices to look-up in the hash table, thus obtaining the feature vectors of the corner vertices. Subsequently, we obtain the feature vector at x via bilinear interpolation of the feature vectors of corner vertices. Finally, we concatenate the feature vectors of x at L layers and feedforward through a neural network to attain the output S_x

Where $\bigoplus$ is a bit-wise XOR operation, and $\pi = [1, 2654435761]$ are large prime numbers for better cache coherence.

Secondly, we perform a look-up in the lth layer hash table θ_l with the indices i_{x_l}, to obtain the corresponding F-dimensional feature vectors h_{x_l} of the corner vertices, as shown in Eq. 5. [ ] denotes indexing.

$$h_{x_l} = \theta_l[i_{x_l}] \tag{5}$$

Third, we bilinear-interpolate the feature vectors of corner vertices back to the input coordinate x, as h_x.

Fourth, we repeat the above steps for all L layers, then concatenate the feature vectors at all layers into an $LF \times 1$ vector. A neural network, Φ, finally decodes it into the output S_x. We utilize a two-layer convolutional neural network (CNN) with a kernel size of 1×1 to reduce network parameters.

Upon confirming the sizes of the multi-resolution grids and the network, the count of design variables remains consistent during computations at any resolution. This constancy stems from the fact that, for any given sampling position in each layer of the hashed grids, the feature value is obtained through interpolation from the neighboring grids. In the case of 2D, this entails bilinear interpolation using the surrounding four grid feature values, while in 3D, trilinear interpolation employs the surrounding eight grid feature values. As a result, the network serves solely for decoding purposes and exerts negligible influence on the structural representation, so we can maintain a constant size for the

convolutional neural network (CNN) throughout computations across different resolutions. Ultimately, the number of design variables exhibits a linear relationship with the size of each layer T in the multi-resolution grids and positively correlates with the number of layers L. In other words, L, T, F determine the number of design variables besides the 1×1 CNN network. Within the specified parameter selection outlined in Table 1, the hash-encoded network necessitates a storage capacity of 67.1MB per structure.

The core advantage of this methodology lies in its low resource consumption and fast convergence, which is attributed to its usage of a tiny neural network. Methods like the Fourier-featured network (Tancik et al. 2020; Sitzmann et al. 2020), which is widely employed in implicit neural representation topology optimization, necessitates a much larger neural network, such as a multi-layer perceptron, to store structural information, resulting in higher memory usage, larger storage, computational burden, and difficulties in convergence. In practice, the use of multi-resolution hashed grids has been shown to reduce the iterations from hours using the Fourier-featured network to seconds in the applications like gigapixel image fitting.

3.1 Topology optimization

A typical procedure for topology optimization entails conducting gradient-descent on the mechanical performance of a structure to iteratively refine its volumetric representation (Sigmund 2001). In our study, we introduce an additional pre-processing step involving the application of average pooling

(AP) to the structural density. Then, a conventional topology optimization process is performed using the finite-element method (FEA). The overall workflow is shown in Fig. 3.

The purpose of the average pooling is to provide more optimization space for structural stylization and alleviate the computational burden of obtaining high-resolution solutions. Specifically, following the principles of multi-resolution topology optimization, we perform topology optimization on the down-sampled grid S_{ap} obtained through pooling, while conducting style optimization on the original structure S output by the network. This approach relaxes the constraints imposed by the gradients of mechanical performance on structural details. We empirically set the kernel size and stride of the average pooling to 4×4.

In this study, compliance minimization (Bruggi and Duysinx 2012) was adopted as the objective of the topology optimization, with the aim of achieving optimal rigidity while minimizing the weight of the structure, as shown in Eq. 6.

$$\operatorname*{argmin}_{S} C(S)$$
$$s.t. \quad V(S)/V_0 \leq \delta \quad , \tag{6}$$

Wherein, C is the compliance of the structure, which reflects its deformation energy under external forces; V is the volume of the structure; V_0 is the volume of the entire optimization space, i.e., the volume of all-filled grids; and $\delta \in (0, 1)$ denotes the user-specified objective volume fraction.

We employ an L2 loss to enforce a volume constraint on the structure and convert the constraint optimization problem into a single-objective optimization problem through the use of a penalization method, as demonstrated in Eq. 7, where γ is a fixed penalization factor.

$$\mathcal{L}_{mech} = C + \gamma (V/V_0 - \delta)^2 \tag{7}$$

The structure volume can be obtained by summing its density values ρ over all elements, and its compliance C can be calculated with the SIMP topological optimization method

(Andreassen et al. 2011) and finite-element analysis (Rao 2017). The procedure is briefly outlined as follows.

Firstly, the pooled structure S_{ap} is constructed as a finite-element mesh comprising of rectangular elements, and the structure's stiffness matrix K and element stiffness matrix K_e are formulated based on the material elastic properties. They indicate the deformation of the structure under external loads.

Secondly, we solve the structure's deformation U under the external force F, according to the generalized Hook's law $KU = F$.

Finally, the compliance C is calculated as $C = \rho U^T K U$, thus completing the calculation of $\mathcal{L}_{mech}$. Here ρ denotes the $h \times w \times 1$ density channel of the structure S_{ap}. Readers may refer to established research (Andreassen et al. 2011) for a detailed derivation of the topology optimization for compliance minimization task.

Upon establishing the value of $\mathcal{L}_{mech}$, we opt to utilize the Adam optimizer instead of the optimality criteria method (OC) (Sigmund 2001) or the method of moving asymptotes (MMA) (Rojas-Labanda and Stolpe 2015), which are mathematical programming methods. The rationale behind this decision is: MMA, OC, and similar optimization methods are specifically tailored for topology optimization applications. They have demonstrated their effectiveness in enforcing tight constraints (e.g., volume constraint) during the topology optimization process. However, when the optimization objective involves a neural network, we favor the adoption of more versatile gradient-descent optimizers, which facilitate stable convergence. However, general neural network optimizer has significant limitations in that it's hard to reach the set constraints, and users have to manually decide the trade-off between optimization objects (e.g., in topology optimization cases, the optimized structures always have a larger volume than the objective volume fraction δ.)

Fig. 3 Structure topology optimization. Within, we perform an average pooling (AP) on the structure S and then conduct a finite-element analysis (FEA)
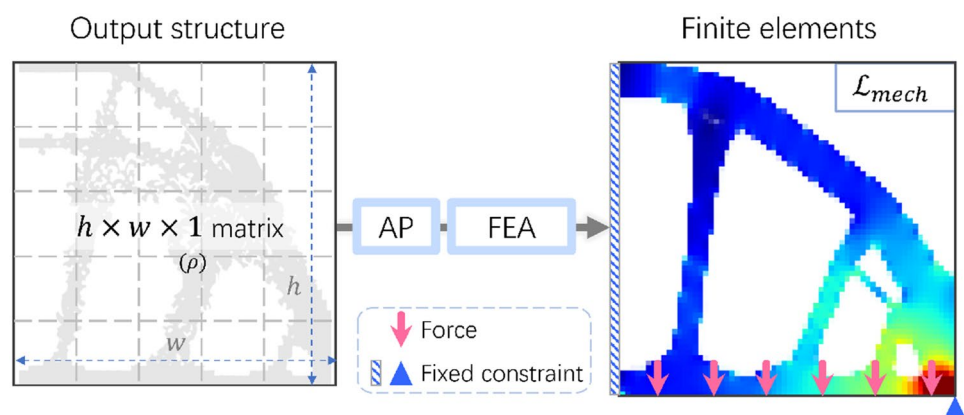
3.2 Style optimization

In order to stylize a structure based on a textual description, the CLIP model (Radford et al. 2021) is introduced. This is a neural network trained on a large corpus of text-image pairs. It converts images and texts into latent codes through corresponding encoders and learns the text-image matching relationship by minimizing the similarity between the latent features of images and texts. After training, it can be utilized for tasks such as image labeling (Zhou et al. 2022), image highlighting (Decatur et al. 2022), and text-to-image synthesis (Frans et al. 2021). In this paper, our objective is to maximize the semantic score, i.e., to maximize the similarity between the image I (Eq. 11) of the structure (with the rho-channel treated as the alpha-channel of an image) and the prompt P that describe the style of the structure, as shown in Eq. 8.

$$\underset{I}{\mathrm{argmax}} \; \mathrm{similarity}(I, P) \tag{8}$$

The image I and prompt P are, respectively, encoded as 512-dimensional latent codes lc_{img} and lc_{txt} by the image and text encoders of CLIP, as illustrated in Fig. 4a. The semantic similarity between them is established by utilizing cosine similarity, and negation is added to convert this semantic score into a loss function, as Eq. 9.

$$\mathcal{L}_{\mathrm{sem}} = -\cos(I, P) \tag{9}$$

Prior to acquiring the image latent code lc_{img}, we perform augmentation on the image I in order to gain controllability of the generated results and improve the convergence. Image augmentation has been previously validated in the research of text-guided image (Frans et al. 2021) and 3D shape (Michel et al. 2022) generation as a means to avoid generating content with a numerically high image-text similarity but hard for humans to identify.

During each optimization iteration, we augment image I with a batch size of B. Each augmentation consists of four components: *Random grayscale*, which transforms image I into grayscale with a specific probability, so as to encourage the system to focus more on the topology of the structure rather than merely altering textures; *Random resized crop*, which randomly crops a portion of the image and resizes it to its original dimensions to focus the system on the central parts of the structure; *Random affine*, which applies a random affine transform to the image to avoid generating adversarial solutions; and *Random background*, which generates a random Gaussian-blurred background to avoid the system cheating the semantic score $\mathcal{L}_{\mathrm{sem}}$ by generating textures with the same background color.

We recognize that readers may have doubts about the tendency of text-image models of stylizing the structure,
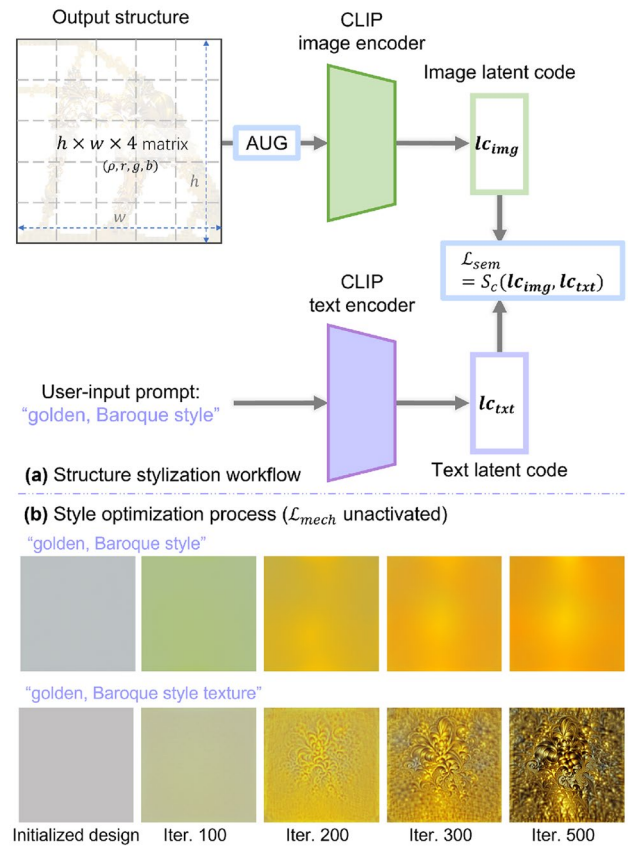
Fig. 4 Structure style optimization. **a** Structure stylization workflow. Augmented image batch and descriptive prompt are input to image and text encoders of CLIP correspondingly, and the difference between the output latent codes are minimized to enforce the structure style to be consistent with the prompt. **b** Structure optimization process. We visualize the process of structures (image in this case) that are solely optimized with the semantic score $\mathcal{L}_{\mathrm{sem}}$

i.e., that neural networks tend to optimize texture over topology to achieve higher semantic scores. We assert that this tendency can be regulated by enforcing grayscale image input to the neural network, through operations like an image alpha-channel penalty or a higher random grayscale probability. We will systematically demonstrate the controllability of the stylization in Sec. 4.

As for the prompt latent code lc_{txt}, it can be generated by feedforwarding the prompt into the text encoder. Additionally, through multiple experimental trials, we have found a correlation between the convergence speed of stylization and the choice of prompts. Generally, prompts that encompass a greater level of detail and incorporate additional semantic constraints lead to faster convergence. For instance, as depicted in Fig. 4b, the prompt "*golden, Baroque style*" only achieves a blurry golden image after 500 iterations. However, when the prompt is completed to "*golden, Baroque style texture*", more intricate details have been obtained.

In conclusion, we adopt a trained, fixed-parameter CLIP model (*ViT-B/32*) to infer the image and text latent codes, and minimize their difference to encourage the structure stylization that is semantically consistent with the text description.
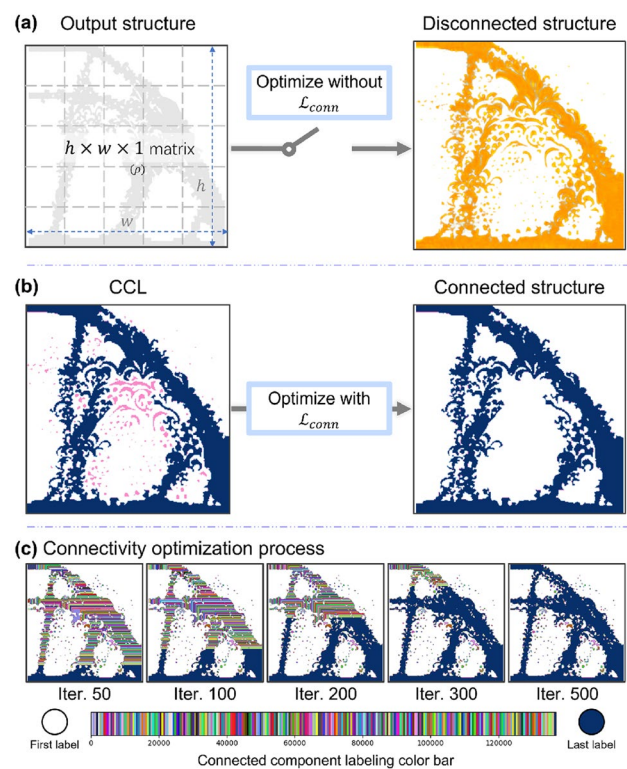
3.3 Connectivity optimization

In the preceding two sections, we performed topology optimization on an averaged-pooled structure and subjected it to text-guided stylization. These processes may result in the presence of disconnected parts, which are meaningless in actual fabrication. Thus, we aim to introduce constraints to ensure the generated structure is integral, meaning the structural density of the disconnected parts ρ_d should be zero. Note that the proposed connectivity constraint shares a similar motivation to the perimeter constraint, which indirectly suppresses the checkerboard pattern by minimizing the perimeter of internal boundaries of the material distribution (Borrvall 2001).

It is important to emphasize that while compliance optimization also promotes connected structures, a separate connectivity optimization step is necessary. This requirement arises due to the nature of density-based topology optimization, where a minimum density value (e.g., 1e-3) is assigned to each element in the structure to prevent numerical instabilities during convergence, rather than setting it to zero. Consequently, without connectivity optimization, the CLIP network may generate floating decorations in non-load-bearing regions of the structure in order to maximize appearance scores while incurring minimal penalties. Through empirical investigation, we have observed that this phenomenon occurs in the absence of connectivity constraints.

Therefore, we employed connected component labeling (He et al. 2017) to identify disconnected regions within the structure and used them as a mask to construct the loss function $\mathcal{L}_{\text{conn}}$, so as to enforce the density value of disconnected parts ρ_d to be zero, as shown in Eq. 10.

$$\mathcal{L}_{\text{conn}} = |\rho_d| \tag{10}$$

The process is depicted in Fig. 5. Firstly, the structure density ρ was thresholded into binary values of 0 and 1, with a threshold of 0.1, and the portion with $\rho = 1$ was extracted as the mask M. Secondly, a labeling matrix Q of the same size as the structure ($h \times w \times 1$) was initialized with element values ranging from [1, hw]. The non-masked portion of the labeling matrix was then set to zero, i.e., $Q[\sim M] = 0$, and the matrix was iteratively subjected to max-pooling (kernel size = 3, stride = 1, padding = 1) to obtain the labeling of the connected regions within the structure. Finally, using the volume fraction δ as a threshold, we designated regions with a label corresponding to fewer than δhw elements as disconnected, and obtained the indices d of these elements,

Fig. 5 Structure connectivity optimization. **a** A structure could be disconnected without connectivity constraints. **b** Through applying connected component labeling, we successfully divide the disconnected parts marked in pink, and enforce them to be of zero density to optimize structure connectivity. **c** Connectivity optimization process, where colors indicate the component labels

thus making the density value of disconnected parts to be zero, ρ_d, in the structure.

Essentially, users need to specify one parameter for the connectivity constraint, which is the number of iterations for performing the connected component labeling (CCL). Let's assume we are performing CCL on a structure with a resolution of $h \times w$. In the worst case, where the entire structure resembles a checkerboard pattern, it would necessitate $h \times w$ iterations of CCL. In practice, due to the prevalence of extensive connected regions within optimized designs, we recommend users empirically reduce the iteration number to $0.5hw$ to expedite computation.

4 Experiments

Experiments overview. During the experiments, we focus on the mechanical performance of the structure (Sec. 4.1), the controllability and visual quality of text-guided generation (Sec. 4.2), and the connectivity of the structure (Sec. 4.3). Note the results presented in Sec. 4 are all based on 2D

structures or their sketch-ups for easier understanding, and we introduce the extension to 3D structures in Sec. 5.

Experiment environment. The proposed system runs on a laptop PC (CPU: Intel Core i9-13900HX, GPU: Nvidia RTX4080, OS: Windows11, Python3.9). In particular, we used the algebraic multigrid method (Wu et al. 2015; Bell et al. 2022) for an accelerated solving of the deformation matrix U of the structure during topology optimization. Most of the computations are run on the GPU, and VRAM is the determining factor of the computational scale.

Optimization parameters. The hyperparameters and settings for all the experiments involved in this paper are shown in Table 1. Within, the scale of the hash table and CNN are positively correlated with the convergence rate, the high-frequency details of the structure, and computational cost. Our system is capable of achieving convergence within 500 iterations for typical topological optimization tasks (Valdez et al. 2017) under various parameter settings. The CCL iterations are set to 2000 to accommodate tasks at various resolutions. Both iteration numbers are conservatively set.

Initialization. We randomly initialized the design parameters for allowing a certain degree of randomness in the optimized design, with the intention of enhancing the novel user experience when using our algorithm. The design variables consist of two components: feature values of multi-resolution grids, and weights of the decoding neural network (i.e., decode feature values to density and RGB values). The randomness can be disabled: First, *feature values of multi-resolution grids.* We set the initial feature values of multi-resolution grids as random values between -1^{-4} and -1^{-4}. After applying the Sigmoid activation function (Han and Moraga 1995), their values will be closely centered around 0.5, resembling traditional uniform initialization. All feature values of the multi-resolution grids can be set to 0 to achieve a conventional uniform design variable distribution; Second, *weights of the neural network.* Random weights of the neural network imply that under different random seeds, the network will decode the same feature value into different values. We can also fix the random seed to eliminate this randomness. The optimization solutions are influenced by the initial random design variables, as shown in Fig. 8.

Gradients computation. We employ the wording 'gradients' to elucidate the derivative of the loss term L with respect to the design variables. Within the realm of topological optimization, 'gradients' is also recognized as 'sensitivity.' To streamline the programming, we use the Autograd function of PyTorch (Paszke et al. 2019), which enables automatic gradient computation (i.e., no manual derivation of the derivatives is required). The derivatives of the three terms L_{mech}, L_{sem}, and L_{conn} are all computed in the same manner. The motivation for adopting automatic differentiation is its convenience. Specifically, when the system involves complex networks, manually deriving

Table 1 System hyperparameters

Parameter	Notation	Value
Layers of multi-resolution grids	L	16
Size of hash table	T	2^{19}
Feature dimensions per entry	F	2
Grid max/min resolution	N_{min}, N_{max}	8, 256
CNN layers		2
CNN channels		64
Optimization iterations		500
CCL iterations		2000
Loss function penalty factors	α, β, γ	$9e3, 1, 3e3$
Image batch size	b	16

the derivatives of the loss function with respect to network parameters can be cumbersome and error-prone.

Note that new training is required for each distinct optimization task, as the hash-encoded neural network is trained to implicitly represent a single structure optimized under specific boundary conditions and stylization. In other words, each task is independent and there is no shared prior between different tasks for accelerating the training. Due to the randomly initialized design, the final solutions may have a floating performance difference around $\pm 3\%$. Users may also fix the random seed to ensure the same solutions are obtained under identical input conditions.

4.1 Validation

This section presents the optimization performance of the system and verifies its core design.

Mechanical performance. To assess the mechanical performance, we conducted an evaluation of our system by comparing the optimization performance among the traditional SIMP method (here we adopt the 165-line Python code written by Niels Aage and Villads Egede Johansen) (Andreassen et al. 2011), our method with only L_{mech} activated, and our method with the full loss term L. We perform the comparison under three representative compliance minimization tasks: a Bridge, a Messerschmitt-Bölkow-Blohm (MBB) beam, and an L-bracket, as depicted in Fig. 6. Throughout the experiments, the inputs were specified as "*golden, Baroque style*". Finite-element analysis was conducted at a resolution of 64×64 (i.e., the original 256 resolution grids were subjected to pooling with a kernel size of 4), while the appearance stylization optimization was performed at a resolution of 256×256 over a total of 100 iterations.

Firstly, we compared the optimized solutions from the SIMP method and our method with only L_{mech} activated, in other words, without semantic and connectivity constraints. We set the penalty factor of SIMP to be the same as of our
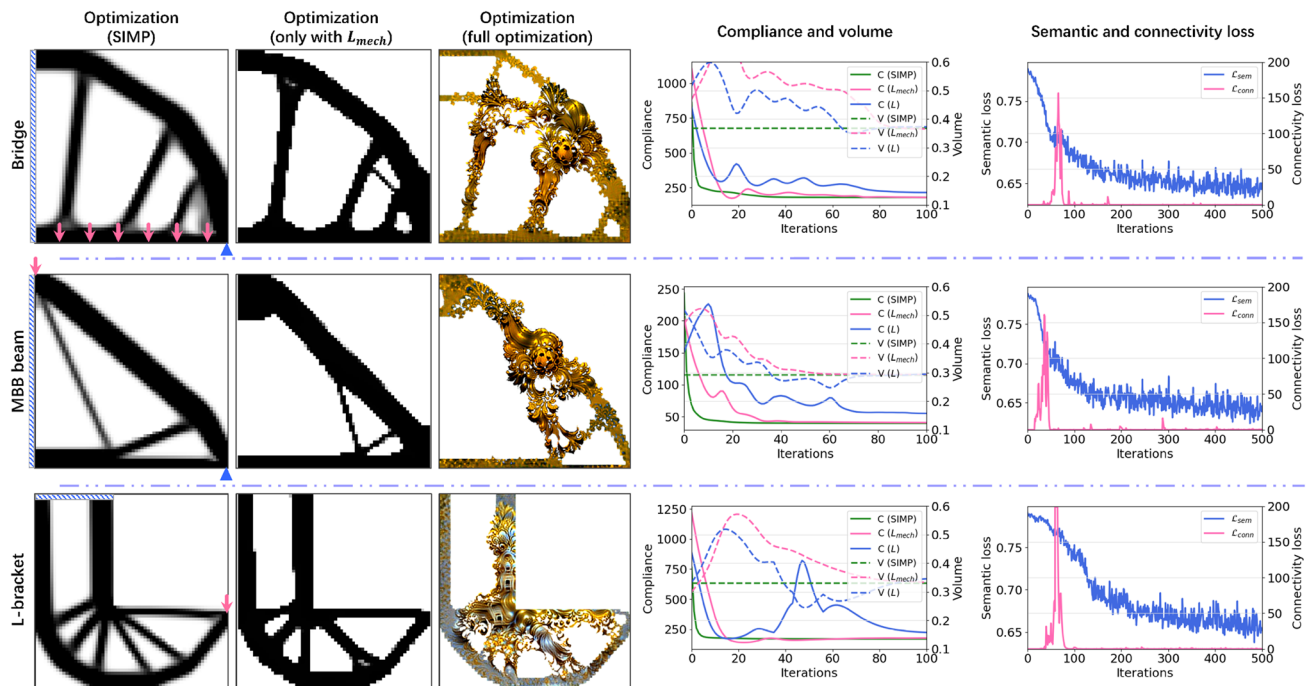
Fig. 6 A comparison of the optimization performance among the traditional SIMP method, our method with only L_{mech} activated, and our method with the full loss term L. The comparison is performed under three tasks. In the right figures, C denotes the compliance and V denotes the volume

method (p=2.0) and fine-tuned the sensitivity filter radius r_{min} to 1.5 for obtaining a good optimization outcome. It turns out that our method leads to structures with sharper edges after 100 iterations. Moreover, we numerically compare the performance. As for the Bridge, MBB-beam, and L-bracket optimization tasks, SIMP and our method (L_{mech}) yield structure with the compliance of (181.06, 181.08), (39.48, 40.80) and (169.80, 177.23), while their corresponding volumes are (0.368, 0.366), (0.293, 0.293) and (0.332, 0.333). The volume difference is due to the fact that we applied a penalty method to construct the loss, which inevitably leads to a different volume than the preset volume fraction *delta*. It turns out that our method shares a similar topology optimization performance to SIMP while featuring additional capabilities of structure resolution and pooling.

Second, we visually compared the stylized solutions to the previous ones and observed that they effectively preserve the primary load-bearing components while incorporating stylized elements around them. This observation is consistent with the mechanical performance results depicted in the "Compliance and volume" plots of Fig. 6. Specifically, our method yields structures with approximately 30.27% higher compliance (19.94%, 39.69%, and 31.17% higher compliance for the Bridge, MBB-beam, and L-bracket, respectively) while maintaining an enhanced aesthetic style compared to the strict compliance-minimized structures obtained through the traditional method.

Third, we observed that the proposed method tends to converge to a stable topology within 100 iterations. Meanwhile, the colored texture of the structure takes more iterations to enrich the details. The slower convergence speed of the texture is mainly due to the nature of semantic loss $\mathcal{L}_{sem}$, as shown in the "Semantic and connectivity loss" plots of Fig. 6. Here we leverage image augmentation (Fig. 10) to relieve this issue. Image augmentation is proven effective in various research of text-guided generation (Michel et al. 2022; Poole et al. 2022; Jain et al. 2022), despite its random augmentation (e.g., crop, affine transform, grayscale) will lead to a noisy semantic loss. Besides, we observed that the compliance minimization process also serves as an augmentation that accelerates the overall convergence, whose effects can be visualized from the comparison between the optimization process shown in Fig. 1 and Fig. 4. As for the influence of the connectivity constraint, L_{conn} has substantial values only during the transition interval when the structure evolves from the initial gray density field to a connected structure (approximately within the range of 0 to 100 iterations). Subsequently, once the optimization stabilizes, we found that L_{conn} always remains zero. The primary effect of the connectivity constraint lies in its capability to eliminate disconnected parts within the structure. This removal is a task challenging to achieve solely through the loss term associated with compliance minimization.
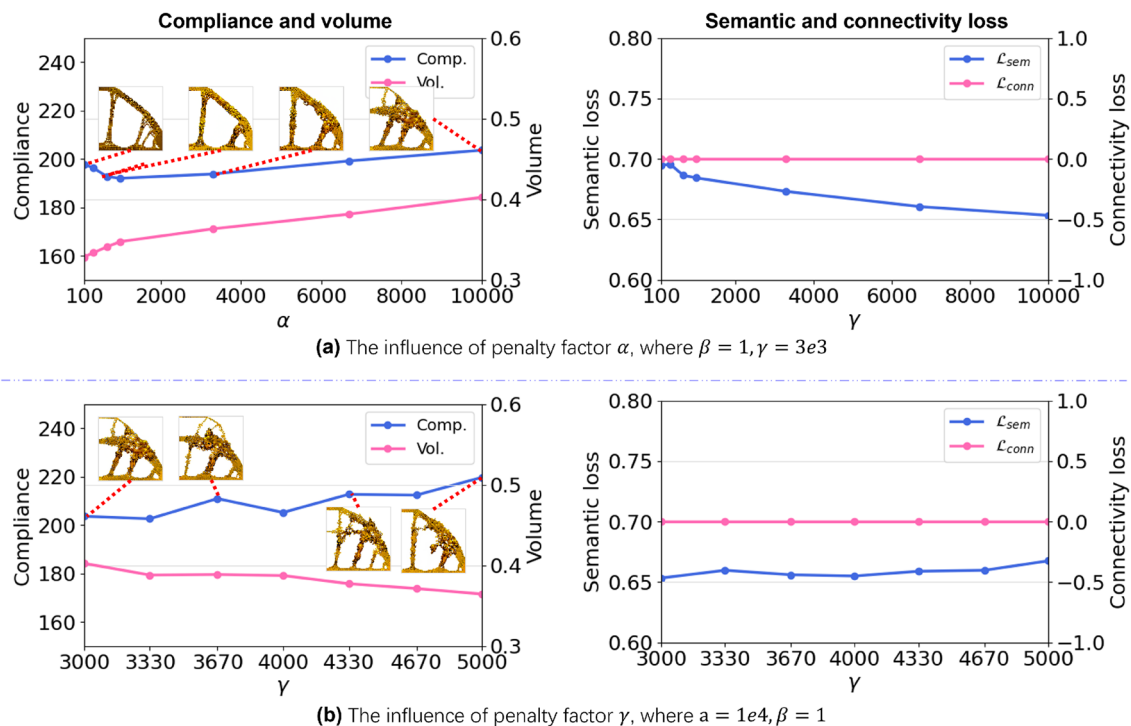
Influence of penalty factors. The penalty factors, α for the semantic loss $\mathcal{L}_{\text{sem}}$, β for the connectivity loss $\mathcal{L}_{\text{conn}}$, and γ for the volume, collectively impact the optimization results as constituents of the loss function $\mathcal{L}$ (Eq. 1). As depicted in Fig. 6, the connectivity loss, $\mathcal{L}_{\text{conn}}$, remains zero for the majority of the optimization process, indicating its limited influence on the final loss function. Therefore, we focus our analysis on the effects of α and γ, as illustrated in Fig. 7. In Fig. 7a, while keeping the volume penalty factor, γ, constant, we progressively increase α. It is observed that α exhibits a negative correlation with $\mathcal{L}_{\text{sem}}$ while displaying a positive correlation with compliance and volume. Evidently, this signifies that by increasing α, it is possible to trade-off the mechanical performance of the structure for a lower semantic loss, thus achieving a higher similarity with the user-defined prompt. Similarly, in Fig. 7b, by increasing the volume penalty factor, γ, the structure's volume can be reduced, albeit at the cost of an increase in semantic loss.

It is noteworthy that the selection of α, β, and γ depends on various factors, including the boundary conditions of the topology optimization problem or user-input prompts, introducing a degree of uncertainty. To alleviate the difficulties associated with user penalty factor selection, we propose a simple method. Upon examining the magnitudes of the different terms in the loss function, compliance is determined
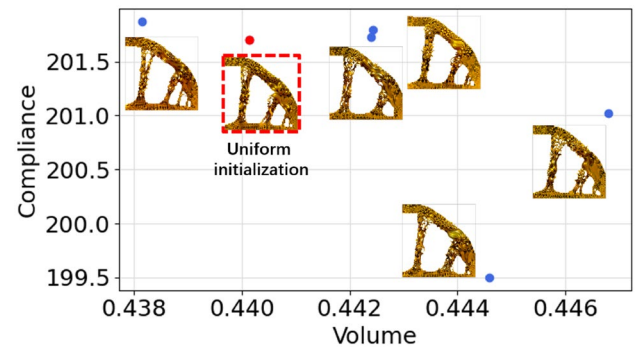
Fig. 8 Randomness of our approach. The optimized structure is plotted beside its mechanical performance (the blue dots). The structure optimized with uniform design parameters initialization is marked with the red dot

by the boundary conditions of the topology optimization, semantic similarity takes values between 0 and 1, while connectivity plays a minor role in the optimization process. Therefore, we neglect the impact of $\mathcal{L}_{\text{conn}}$ on the loss function $\mathcal{L}$ by setting $\beta = 1$. Additionally, we only activate $\mathcal{L}_{\text{mech}}$ to evaluate the convergence of the structure's compliance, and empirically set α and γ to be one order of magnitude larger than the compliance. Consequently, the weighted

(a) The influence of penalty factor α, where $\beta = 1, \gamma = 3e3$

(b) The influence of penalty factor γ, where a $= 1e4, \beta = 1$

Fig. 7 The impact of penalty factors on optimization results is examined by observing the optimized Bridge structure under different combinations of penalty factors. Specifically, the structure optimizations are conducted at the locations indicated by the dots in the plots, and part of the results are demonstrated. The results are pointed to their corresponding dots by red dashed lines. **a** The influence of the penalty factor α while keeping β and γ fixed. **b** The influence of the penalty factor γ while keeping α and β fixed

terms in the loss function are of similar magnitudes. After initializing the weights using the aforementioned method, users can adjust the weights within one to two orders of magnitude to achieve personalized design requirements. Furthermore, we can generate a series of optimized solutions under different penalty factors and employ user-in-the-loop Bayesian optimization to select appropriate penalty factors.

Convergence. We compare the optimization convergence of our multi-resolution hash-encoded network and the Fourier-featured network (Chandrasekhar and Suresh 2021) as implicit-neural-representation-based topology optimization techniques, as illustrated in Fig. 9. Both methods encode the input coordinates to generate the respective feature values (e.g., RGB color of an image, or density of a structure). In the comparison experiment, the hashed grids utilized a two-layer CNN, while the Fourier-featured network employed a three-layer sinusoidal activated multi-layer perceptron of 512 layer widths, with the first sinusoidal activation layer set at a frequency of 90 to ensure the capture of adequate

high-frequency structural details. Upon completion of 500 iterations, it was found that our hashed-grids-based method obtained a more abundant representation of high-frequency structural details and better mechanical and aesthetic performance, thereby validating the enhancement in convergence speed achieved through the reduction of neural network parameters.

Computational cost. We examine the training time and memory consumption of the optimization of a bridge with a resolution of 256×256, as shown in Fig. 11a. Under 100 optimization iterations, the average iteration time was 0.587s (i.e., a total time of 58.66s), with the computation time for the loss functions of topology, appearance semantic score, and connectivity being 0.203s, 0.060s, and 0.139s, respectively. In other words, topology optimization consumes the most computation (training) time as 35%, while style optimization and connectivity optimization costs 10% and 24%. The rest training time is spent on the feedforward and backpropagation (i.e., automatic differentiation) of the neural network. The peak memory consumption under the 256×256 resolution during training was 1.43GB, which is within the computational capability of mainstream commercial GPUs, shown in Fig. 11b. Within, CLIP (clip-Vit-B-32) consumed a fixed amount of VRAM as 1.07GB. A trained network under the settings listed in Table 1 costs 67.1MB for storage.

Ablation study of image augmentation and loss terms. The augmentation of structural images plays a significant role in both convergence speed and quality. Here we visualize their effects under 100 iterations of optimization. In Fig. 10a–e, image augmentations were successively removed to observe their impact. The result indicated that: the presence of a random background is crucial to avoiding the generation of adversarial content. Without it, the network tends to generate textures of the same color as the background in an effort to cheat for a higher semantic loss. The random affine transforms and random resized crop (10% of the image) focus the network's attention on local regions of the structure, allowing for fine-grained updates to be made to both topology and texture. The random grayscale, which
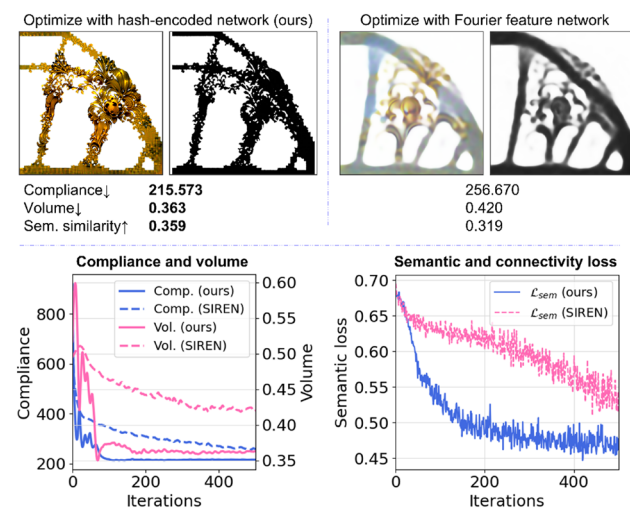

Fig. 9 A comparison of the optimized results after 500 iterations to the methods using the Fourier-featured network SIREN
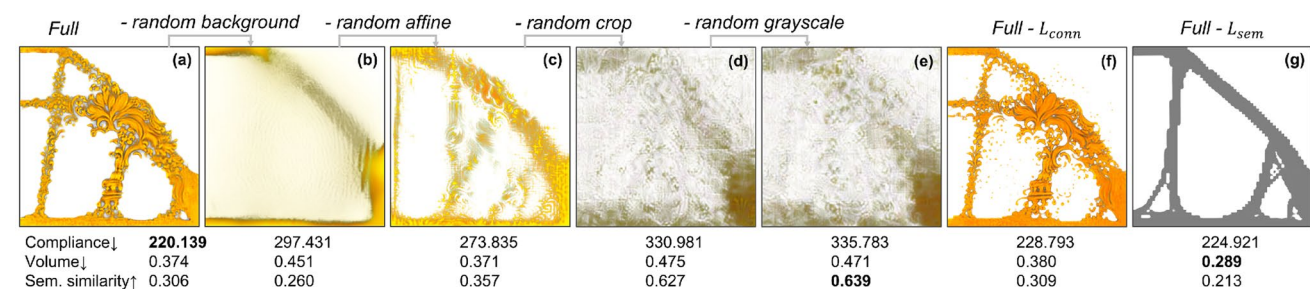
Fig. 10 Ablation study of the designs in the proposed method. **a–e** study the image augmentation, wherein random background, random affine transform, random resized crop, and random grayscale, are suc-

cessively removed. **f, g** depict the ablation study of connectivity and stylization optimization, where '*full*' represents the utilization of all proposed image augmentation and loss functions
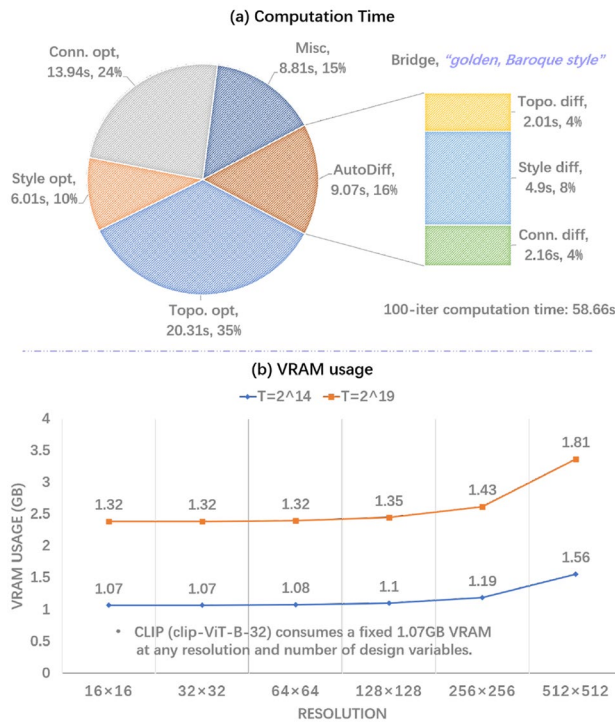
Fig. 11 Computational cost. **a** The 100-iteration computation time of a bridge structure under the stylization prompt "golden, Baroque style." **b** The VRAM usage under different resolutions and number of design variables

randomly converts 10% of the image to grayscale, encourages the network to focus more on the structure's topology than its texture. In Fig. 10e, the removal of the connectivity loss terms reveals a notable inhibitory effect on disconnected

parts within the structure. Lastly, the stylization loss term $\mathcal{L}_{sem}$ was removed to serve as a reference in the absence of stylization.

Comparison with texture-guided stylization. Finally, we perform a visual comparison between texture-guided (Martínez et al. 2015; Hu et al. 2019; Navez et al. 2022) and our prompt-guided stylization, as shown in Fig. 12. The results indicate that owing to the highly abstract nature of semantic representations, we can stylize the structure from a holistic perspective, which would be much more difficult for texture-guided methods.

4.2 Stylization gallery

In this section, we present the controllability of stylization and a stylization gallery.

Stylization controllability. In addition to modifying the description prompt, the structure style can also be controlled by adjusting the hyperparameters of the system. In Fig. 13a, the control of the high-frequency details (i.e., length-scale) of the structure's topology and texture is achieved by adjusting the maximum resolution N_{max} of the multi-resolution hashed grids, as shown in Fig. 15. Specifically, N_{max} is negatively correlated with the length-scale of the structure. The reason is that when N_{max} takes a small value, the features of the structure (i.e., density and color) will be interpolated from a sparser grid, which is similar to applying a low-pass filter on the geometry and color of the structure over the design space.

Given that the topology and texture features of the structure are obtained from the bilinear interpolation of hashed grids vertices values, the maximum resolution N_{max}
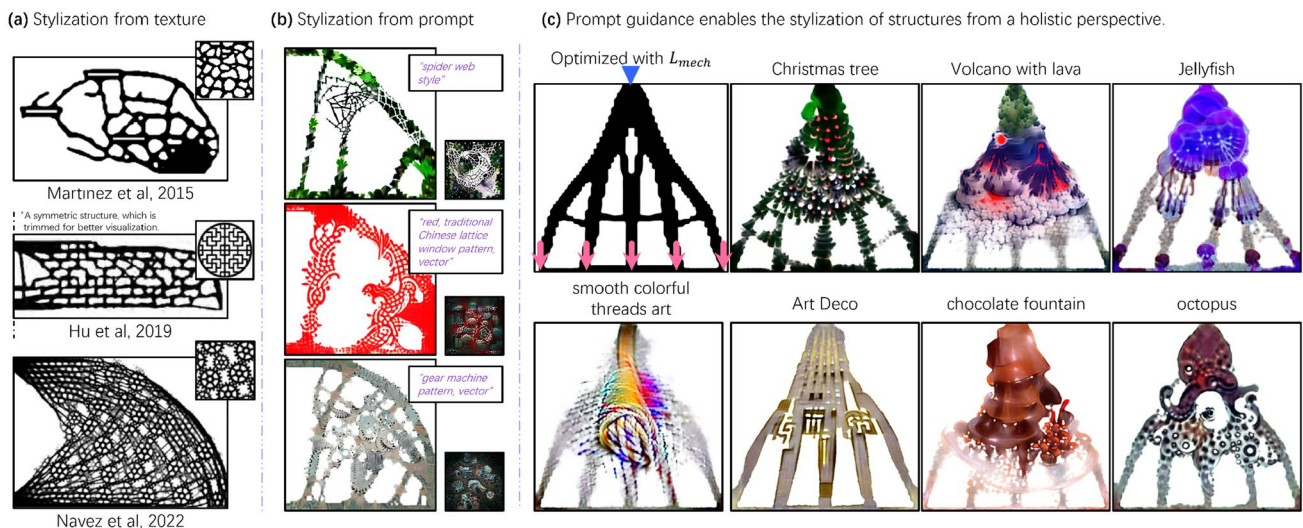

Fig. 12 Visual comparison between texture-guided and prompt-guided stylization. **a** Texture-stylized results from the state of the art research. **b** Prompt-stylized results. The tiny figures in the lower right

are optimized only with the semantic constraint L_{sem} by our method. **c** Prompt guidance enables unique stylization

Fig. 13 Stylization controllabil-
ity. **a** The structural topology
and texture characteristics are
modulated by adjusting the
maximum resolution, N_{max}, of
the hashed grids. **b** The topol-
ogy-only stylization is amplified
by promoting grayscale image
inputs and increasing the penali-
zation, p, of the structure image
alpha-channel
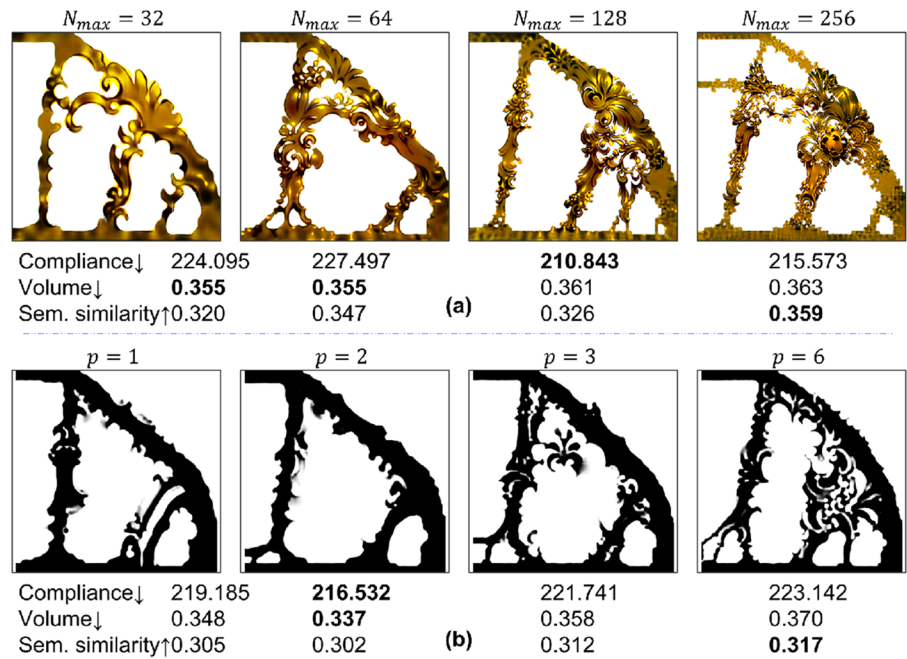
of the grids directly determines the level of detail in the
solution, and reducing N_{max} can be viewed as adding a
low-pass filter to the solution. The same concept is applied
in the length-scale control of density-based topology opti-
mization: filters are applied to avoid the checkerboard pat-
tern. From a frequency domain perspective, this is equiva-
lent to applying a low-pass filter to the density field of
the structure, i.e., removing high-frequency components
(i.e., checkerboard) at each optimization iteration. Con-
sequently, this regulation will assist the user in balancing
the trade-off between structural details and manufacturing
difficulty or cater to a personal aesthetic sense.

In Fig. 13b, the focus is on the system's ability to opti-
mize the structural topology. In cases where color 3D man-
ufacturing may be unavailable, we expect the system to
still be able to express its stylization through the topology.
To encourage the network to focus on the structural topol-
ogy, stronger grayscale inputs are encouraged. Enhanc-
ing topological stylization involves two steps: first, the
structure images are all converted to grayscale; second,
the image's transparency is penalized by a factor p, as
shown in Eq. 11.

$$I = Y\rho^p + Z(1 - \rho^p), \tag{11}$$

where $I \in \mathbb{R}^{h \times w \times 3}$ is the structural image, $Y \in \mathbb{R}^{h \times w \times 3}$ is the
RGB channel of the image, $\rho \in \mathbb{R}^{h \times w \times 1}$ is the density chan-
nel of the structure (i.e., the alpha-channel of the image),
$Z \in \mathbb{R}^{h \times w \times 3}$ is the random background, and p is the penalty
factor. As p increases, the network's output becomes increas-
ingly binary, thus the optimization will also become more

focused on the structural topology, which is validated by the
increasing semantic score. Note that we applied the same
volume constraint in the optimizations presented in Figs. 9,
10 and 13 for enabling a consistent comparison.

Stylization gallery. We have validated the efficacy of
stylization in three prototypical topology optimization
cases, as shown in Fig. 14. The results demonstrate that
the system not only generates textures that align with tex-
tual description, but also stylizes the structure in one piece
rather than repetitively mimicking local texture patterns
(e.g., Baroque style decorative patterns, spiderweb pat-
terns, and branch patterns). Please take note that our pro-
posed methodology facilitates the optimization of numer-
ous abstract aesthetics (e.g., "*wood appliques*" and "*floral
ornament*") that pose a challenge to be solely represented
by a single exemplar, as demonstrated in Sec. 2.2. Specifi-
cally, the application of periodic replication of exemplars
on a structural level proves to be more suitable for styliza-
tions encompassing periodic geometric features, such as
the "*Eiffel Tower*" or a "*spider's web*". These approaches,
however, encounters difficulties when dealing with intri-
cate or abstract stylized objectives. Consequently, the
above observation provides a justification for considering
the holistic optimization of textual descriptors.

Moreover, users may tweak the input text to fine-tune
the appearance, as shown in Fig. 16. While simultaneously
achieving performance and global stylization in structural
design remains a challenging task, we believe that the pro-
posed system provides users with an accessible and effi-
cient tool to rapidly iterate ideas.
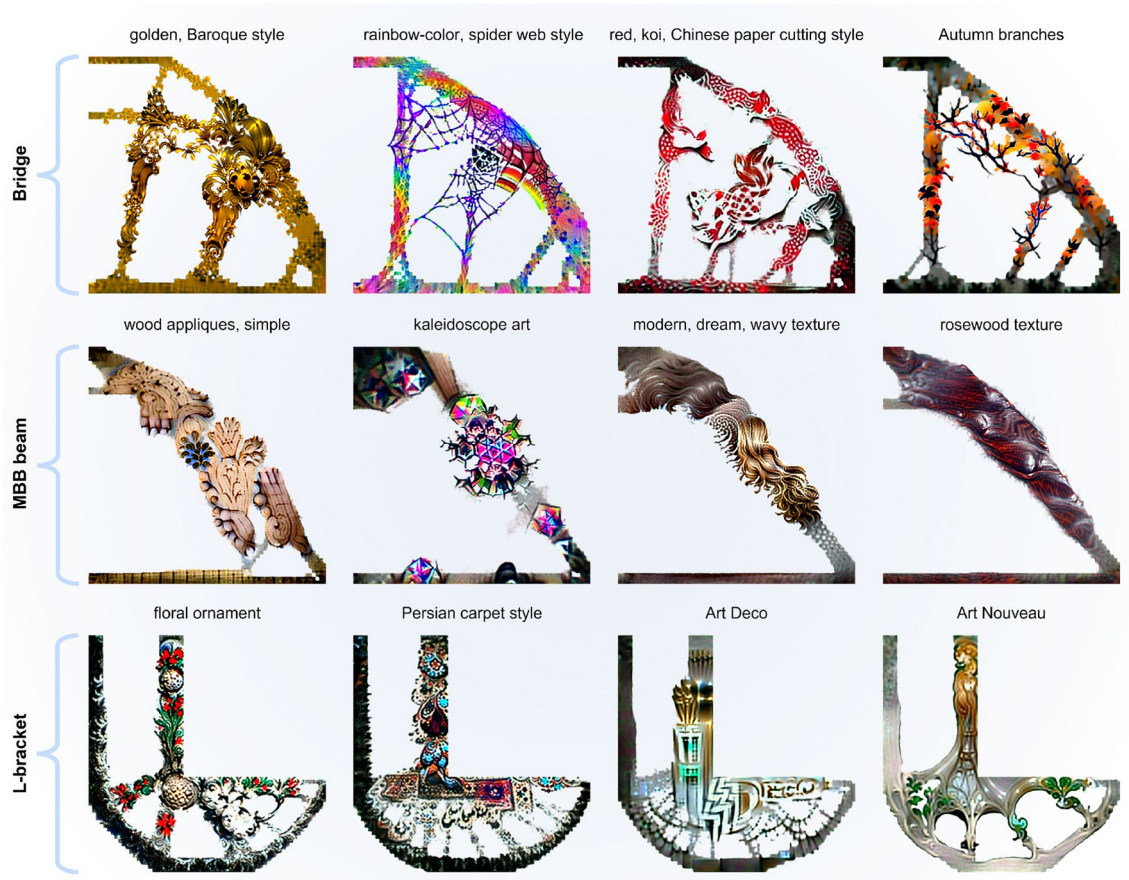
Fig. 14 Stylization gallery of topology-optimized structures, including (top) bridges, (middle) MBB-beams, and (bottom) L-brackets. The prompts that were used for stylization are shown above each structure
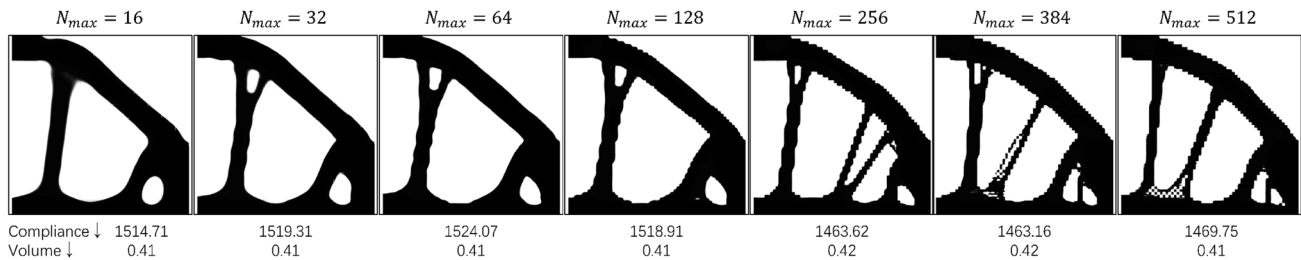

Fig. 15 Length-scale control of the structure through progressively tuning N_{max}

4.3 Full-color 3D-printing

We 3D-printed sample objects generated from our method to validate the connectivity of the structures. As demonstrated in Fig. 17, we printed a set of topologically optimized bookshelves, which are subjected to distributed loads on the top surface and fixed on the sides. After completing the optimization, we fed refined grids into the network to obtain higher-resolution structures with smoother surface contours (Chandrasekhar and Suresh 2021). The 2D structures were then sketched and converted into a mesh in PLY format using the marching cube method (Lorensen and Cline 1987), with structural textures represented by vertex color. The experimental results attest to the significance of incorporating a connectivity loss term in the optimization process and exhibit the structural capacity of weighty external loads.
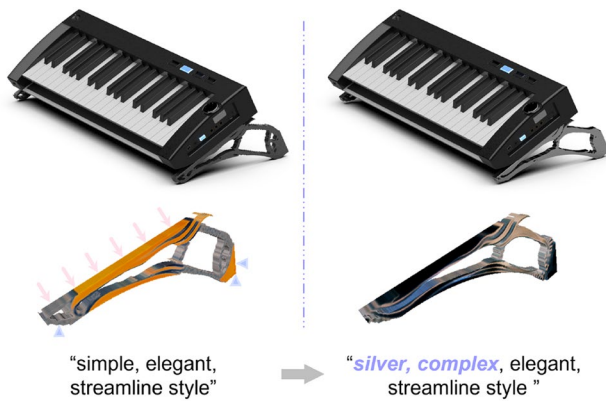
Fig. 16 Editing text-based stylization of a synthesizer stand. The textual description "streamline style" maintains the geometric feature, and the edited text "silver, complex" leads to an appearance update
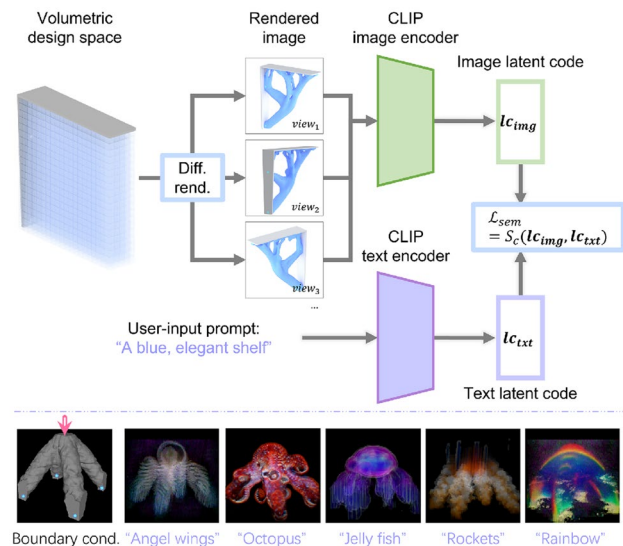

Fig. 17 3D-printing results of stylized topological optimized structures. The prompt was "*golden, Baroque style*". The stylized structures were 3D-printed using the *XYZ da Vinci Color mini*

Fig. 18 The optimization workflow for 3D stylized topology optimization. The main difference with 2D cases is that images are rendered from the 3D volumetric representation rather than directly obtained. Then, multi-view images are feedforward to the CLIP network for assessment of their semantic similarity to the user-input prompt. We visualize several optimized 3D structures below, labeled by their stylization prompts, respectively

5 Limitations and future works

The system can be extended to stylized topology optimization in 3D, as shown in Fig. 18. The primary difference between 3D and 2D problems lies in the fact that the structure images are obtained from the differentiable rendering $I \in \mathbb{R}^{h \times w \times 3}$ of a 3D volumetric representation, which is a 3D grid with four channels of colors and density: $S \in \mathbb{R}^{x \times y \times z \times 4}$. This is achieved by leveraging the neural radiance field (Mildenhall et al. 2021), which samples the structural features along the camera view directions within the optimization space, and accumulates these features into the pixels corresponding to each camera view, so as to render an image. The computational cost of obtaining high-quality optimized structures significantly increases with the dimensionality. Currently, our pipeline that performs topology and appearance optimization synchronously in each iteration often requires tens of minutes to produce a well-stylized 3D result.

Another major drawback of the proposed method lies in the application of penalty optimization. In this case, it is much more difficult to reach a desired volume fraction (or other pre-defined constraints) due to the nature of penalty optimization. The same situations are happening to stress and displacement-constrained optimization. This limitation is also noticeable when performing pure topology optimization tasks: both the optimization quality and the convergence speed are not as good as traditional explicit methods. Introducing conventional optimizers like MMA into the current method would be a promising solution.

In future work, we will enhance the parameterization of the system to gain more controllability over stylization (e.g., parameterize the structure with skeletons for manual shape adjustment). Additionally, we believe that extending the system for multi-material 3D-printing is also a promising avenue.

6 Conclusions

The simultaneous pursuit of functional and aesthetical design in commercial or personalized products has long been a challenging task, requiring designers to possess a sound understanding of physics and an impeccable sense of aesthetics. We present a text-guided stylized topological optimization method, achieved through the introduction of a large-scale text-image neural network. Upon input of mechanical design requirements and a textual description of

desired structure style, our system is capable of generating full-color, 3D-printable solutions with stylistic tunability.

We consider the proposed system both as a fabrication-ready design tool for DIY enthusiasts, and a backbone and source of inspiration for advanced structure stylization design using powerful and controllable generative methods (Zhang and Agrawala 2023). At present, there are numerous intriguing issues that remain to be investigated, such as the stylization of multi-material structures and part-aware shape parameterization (Hertz et al. 2022). We believe these developing techniques will finally aid in making topology optimization a more user-friendly automated tool, improving design efficiency and inspiring design creativity.

Declarations

References

Aage N, Andreassen E, Lazarov BS, Sigmund O (2017) Giga-voxel computational morphogenesis for structural design. Nature 550(7674):84–86

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines of code. Struct Multidisc Optim 43:1–16

Bell N, Olson LN, Schroder J (2022) PyAMG: algebraic multigrid solvers in python. J Open Source Softw 7(72):4142. https://doi.org/10.21105/joss.04142

Borrvall T (2001) Topology optimization of elastic continua using restriction. Arch Comput Methods Eng 8:351–385

Bruggi M, Duysinx P (2012) Topology optimization for minimum weight with compliance and stress constraints. Struct Multidisc Optim 46:369–384

Chandrasekhar A, Suresh K (2021) Tounn: topology optimization using neural networks. Struct Multidisc Optim 63:1135–1149

Chen M, Lau M (2022) Learning 3d shape aesthetics globally and locally. In: Computer graphics forum

Chen X, Tao Y, Wang G, Kang R, Grossman T, Coros S, Hudson SE (2018) Forte: user-driven generative design. In: Proceedings of the 2018 CHI conference on human factors in computing systems, pp 1–12

Creusen ME, Schoormans JP (2005) The different roles of product appearance in consumer choice. J Prod Innov Manag 22(1):63–81

Dbouk T (2017) A review about the engineering design of optimal heat transfer systems using topology optimization. Appl Therm Eng 112:841–854

Decatur D, Lang I, Hanocka R (2022) 3d highlighter: localizing regions on 3d shapes via text descriptions. arXiv preprint arXiv:2212.11263

Frans K, Soros LB, Witkowski O (2021) Clipdraw: exploring text-to-drawing synthesis through language-image encoders. arXiv preprint arXiv:2106.14843

Hamming R (1952) Mathematical methods in large-scale computing units. Math Rev 13(1):495

Han J, Moraga C (1995) The influence of the sigmoid function parameters on the speed of backpropagation learning. In: International workshop on artificial neural networks, Springer, pp 195–201

He L, Ren X, Gao Q, Zhao X, Yao B, Chao Y (2017) The connected-component labeling problem: a review of state-of-the-art algorithms. Pattern Recogn 70:25–43

Hertz A, Perel O, Giryes R, Sorkine-Hornung O, Cohen-Or D (2022) Spaghetti: editing implicit shapes through part aware generation. ACM Trans Graphics (TOG) 41(4):1–20

Hu J, Li M, Gao S (2019) Texture-guided generative structural designs under local control. Comput Aided Des 108:1–11

Huang X, Xie YM (2009) Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. Comput Mech 43:393–401

Jain A, Mildenhall B, Barron JT, Abbeel P, Poole B (2022) Zero-shot text-guided object generation with dream fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 867–876

Kazi RH, Grossman T, Cheong H, Hashemi A, Fitzmaurice GW (2017) Dreamsketch: early stage 3d design explorations with sketching and generative design. In: UIST, pp 401–414

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980

Li D, Levin DI, Matusik W, Zheng C (2016) Acoustic voxels: computational optimization of modular acoustic filters. ACM Trans Graphics (TOG) 35(4):1–12

Loos S, Svd W, Graaf N, Hekkert P, Wu J (2022) Towards intentional aesthetics within topology optimization by applying the principle of unity-in-variety. Struct Multidisc Optim 65(7):185

Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3d surface construction algorithm. Comput Graphics 21(4):163–169

Ma J, Li Z, Zhao ZL, Xie YM (2021) Creating novel furniture through topology optimization and advanced manufacturing. Rapid Prototyping J 27(9):1749–1758

Martínez J, Dumas J, Lefebvre S, Wei LY (2015) Structure and appearance optimization for controllable shape design. ACM Trans Graphics (TOG) 34(6):1–11

Michel O, Bar-On R, Liu R, Benaim S, Hanocka R (2022) Text2mesh: text-driven neural stylization for meshes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13492–13502

Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R (2021) Nerf: representing scenes as neural radiance fields for view synthesis. Commun ACM 65(1):99–106

Müller T, Evans A, Schied C, Keller A (2022) Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans Graphics (ToG) 41(4):1–15

Navez T, Schmidt MP, Sigmund O, Pedersen CB (2022) Topology optimization guided by a geometrical pattern library. Struct Multidisc Optim 65(4):108

Nobel-Jørgensen M, Malmgren-Hansen D, Bærentzen JA, Sigmund O, Aage N (2016) Improving topology optimization intuition through games. Struct Multidisc Optim 54:775–781

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J and Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 32

Poole B, Jain A, Barron JT, Mildenhall B (2022) Dreamfusion: text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988

Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, et al (2021) Learning transferable visual models from natural language supervision. In: International conference on machine learning, PMLR, pp 8748–8763

Rao SS (2017) The finite element method in engineering. Butterworth-Heinemann, New Delhi

Rojas-Labanda S, Stolpe M (2015) Benchmarking optimization solvers for structural topology optimization. Struct Multidisc Optim 52(3):527–547

Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695

Rozvany GI (2009) A critical review of established methods of structural topology optimization. Struct Multidisc Optim 37:217–237

Sigmund O (2001) A 99 line topology optimization code written in matlab. Struct Multidisc Optim 21:120–127

Sigmund O, Maute K (2013) Topology optimization approaches: a comparative review. Struct Multidisc Optim 48(6):1031–1055

Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Struct Optim 16:68–75

Sitzmann V, Martel J, Bergman A, Lindell D, Wetzstein G (2020) Implicit neural representations with periodic activation functions. Adv Neural Inf Process Syst 33:7462–7473

Tancik M, Srinivasan P, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ramamoorthi R, Barron J, Ng R (2020) Fourier features let networks learn high frequency functions in low dimensional domains. Adv Neural Inf Process Syst 33:7537–7547

Tsai T, Cheng C (2013) Structural design for desired eigenfrequencies and mode shapes using topology optimization. Struct Multidisc Optim 47:673–686

Valdez SI, Botello S, Ochoa MA, Marroquín JL, Cardoso V (2017) Topology optimization benchmarks in 2d: results for minimum compliance and minimum volume in planar stress problems. Arch Comput Methods Eng 24:803–839

Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1–2):227–246

Wein F, Dunning PD, Norato JA (2020) A review on feature-mapping methods for structural optimization. Struct Multidisc Optim 62:1597–1638

Woldseth RV, Aage N, Bærentzen JA, Sigmund O (2022) On the use of artificial neural networks in topology optimisation. Struct Multidisc Optim 65(10):294

Wu J, Dick C, Westermann R (2015) A system for high-resolution topology optimization. IEEE Trans Visual Comput Graphics 22(3):1195–1208

Xie Y, Takikawa T, Saito S, Litany O, Yan S, Khan N, Tombari F, Tompkin J, Sitzmann V, Sridhar S (2022) Neural fields in visual computing and beyond. In: Computer Graphics Forum, Wiley Online Library, pp 641–676

Yang R, Chahande A (1995) Automotive applications of topology optimization. Struct Optim 9:245–249

Yu Y, Jang IG, Kwak BM (2013) Topology optimization for a frequency response and its application to a violin bridge. Struct Multidisc Optim 48:627–636

Zehnder J, Li Y, Coros S, Thomaszewski B (2021) Ntopo: mesh-free topology optimization using implicit neural representations. Adv Neural Inf Process Syst 34:10,368-10,381

Zhang L, Agrawala M (2023) Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543

Zhang Y, Kwok TH (2019) Customization and topology optimization of compression casts/braces on two-manifold surfaces. Comput Aided Des 111:113–122

Zhang W, Yuan J, Zhang J, Guo X (2016) A new topology optimization approach based on moving morphable components (mmc) and the ersatz material model. Struct Multidisc Optim 53:1243–1260

Zhong S, Punpongsanon P, Iwai D, Sato K (2022) NSTO: neural synthesizing topology optimization for modulated structure generation. In: Computer graphics forum. https://doi.org/10.1111/cgf.14700

Zhou C, Loy CC, Dai B (2022) Extract free dense labels from clip. In: Computer vision–ECCV 2022: 17th European conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII, Springer, pp 696–712

Zhu JH, Zhang WH, Xia L (2016) Topology optimization in aircraft and aerospace structures design. Arch Comput Methods Eng 23:595–622