NSTO: Neural Synthesizing Topology Optimization for Modulated Structure Generation

Shengze Zhong, Parinya Punpongsanon, Daisuke Iwai, Kosuke Sato

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

Figure 1: *Structures with maximized stiffness and limited material consumption are optimized with low computation cost, all shown in color. The proposed NSTO also enables generating interpolated solutions with an instant network feedforward, like the seven drum shells.*

Abstract
Nature evolves structures like honeycombs at optimized performance with limited material. These efficient structures can be artificially created with the collaboration of structural topology optimization and additive manufacturing. However, the extensive computation cost of topology optimization causes low mesh resolution, long solving time, and rough boundaries that fail to match the requirements for meeting the growing personal fabrication demands and printing capability. Therefore, we propose the neural synthesizing topology optimization that leverages a self-supervised coordinate-based network to optimize structures with significantly shorter computation time, where the network encodes the structural material layout as an implicit function of coordinates. Continuous solution space is further generated from optimization tasks under varying boundary conditions or constraints for users' instant inference of novel solutions. We demonstrate the system's efficacy for a broad usage scenario through numerical experiments and 3D printing.

CCS Concepts
• Applied computing → Computer-aided design; • Computing methodologies → Shape analysis; Computer graphics;

1. Introduction

The evolution of nature has created various elaborate structures, such as lotus leaves and honeycombs, whose objective performance is maximized at an acceptable cost. Being enlightened, humans have created magnificent structures, from wagon wheels to Sagrada Familia. The study of optimal structural design finally evolves to topology optimization with growing algorithms and computation power.

Structural topology optimization is a mathematical method that maximizes the structure performance under constraints by reasonably distributing the material layout. Specifically, the optimization space is discretized into elements and iteratively filled or deprived of materials to approach design objects. While topology optimization has broad industrial appli-

Optimization space and optimized structure
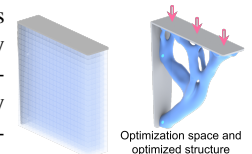
cations in mechanics and architecture [BS03], its exorbitant computational cost hinders its popularity for personal fabrication among ordinary users, especially those without high-performance hardware or cloud servers. To achieve fine structures with smooth boundaries, computation at high-performance workstations extend from hours to days [AALS17; LHZ*18]. This situation worsens when frequent design-parameter adjustments are required.

We proposed the neural synthesizing topology optimization (NSTO) to address these two problems for ordinary users, with the following physical-informed dual networks: *a:* a coordinate-based network for implicit-neural-represented structural topology optimization for computing fine and smooth structures with competitive performance and a much shorter time; *b:* a self-supervised auto-decoder network for further generating a series of structures under continuous boundary conditions for user selection. The dual networks are named *oscillator network* and *modulator network* respectively, since NSTO shares a similar workflow with an analog music synthesizer for generating, modulating, and filtering a structure (waveform). Such network design has been proven to have higher reconstruction quality than a single coordinate-based network for image and shape representation [MGB*21]. Figure 1 shows some of the structures optimized by the proposed system.

Oscillator network is a coordinate-based network that represents and optimizes a single structure at an arbitrary resolution. It inputs coordinates and outputs the material spatial layout so that concatenated network outputs can implicitly form a structure. The physical loss function is obtained by performing finite element analysis on the network outputted structure, and its gradient is leveraged for network updating. At the inference time, a super-resolution (i.e., voxel-wise interpolation) is performed to resample the structure representation to an arbitrary higher resolution.

Modulator network is an auto-decoder network that enhances the overall network expressiveness by layer-wisely modulating the oscillator network's weight. It is only activated to optimize multiple structures under varying boundary conditions. Within, multiple optimization subtasks are simultaneously carried out, and we expect the network to learn interval solutions among subtasks. For example, the user may input subtasks for 30% and 70% structure volume constraints and generate a novel solution of a 50% volume constraint. Each subtask is assigned a latent code, and these latent codes are randomly initialized, inputted, and updated with the network weight. During training, the latent codes are gradually clustered to separate positions in the latent space, labeling the corresponding boundary condition. Finally, the user can interpolate the latent codes to infer novel structures in a corresponding solution space.

We assert the rationality and necessity of NSTO from three aspects. Generally, structural topology optimization is a non-convex problem where its flat solution space leads to a series of near-optimal solutions [SP98]. Such property provides adequate space for new algorithm development, focusing either on performance [WDW16], manufacture [BTX20] or aesthetics [HLG19]. Technically, there is a disequilibrium between the slight structural performance enhancement and computation cost of high-resolution topology optimization, which indicates that multi-resolution structure representation is a potential solution [ZSCM17]. Practically,

users are also concerned about affordable computation cost and good structure appearance besides high-level performance.

We compare NSTO with both physical and data-driven methods and demonstrate its competitive optimization results and much shorter computation time for high-resolution structures. As a physics-informed system, it performs self-supervision with no dataset generation cost and no data-driven artifacts. Besides comparing NSTO with benchmark methods, a series of applications are presented, including the usage and numerical results. Finally, we 3D-printed the optimized structures to further verify the feasibility.

The main contributions are as follows:

- To improve topology optimization efficiency, the oscillator network optimizes and super-resolves structures in an implicit neural form, shortening computation from hours/minutes to seconds.
- To improve solution space generation performance, the modulator network enhances the overall expressiveness, enabling novel solution generation through feedforwarding latent codes.

2. Related works

In this section, we summarize the studies related to our method on the development and computation efficiency of topology optimization and the neural field techniques that the proposed topology optimization framework builds upon.

2.1. Topology optimization

Topology optimization aims to enhance structural performance by redistributing the spatial layout of the material. Its traditional applications include, but are not limited to stiffness enhancement [BPS00], natural frequency reduction [Ped00], and heat conduction [GBS06], and they have been broadening to personal fabrications of interactive toys, furniture, musical instruments, and so on.

Topology optimization has multiple mainstream algorithms. The explicit methods, such as the solid isotropic material with penalization (SIMP) [Ben89] and bidirectional evolutionary structural optimization (BESO) [HX07] discretize the optimization space into structured grids. These methods feature simple algorithms and fast convergence but result in manufacturing difficulties because of fuzzy boundaries. The implicit method based on a level set [YX04] parameterizes the structure as a contour to achieve smooth boundaries, but it needs to reset the level set equation to ensure continuous updating, which causes a computation burden. Moreover, topology optimization is unstable, as it includes checkerboard patterns, mesh dependencies, and local minima. On the contrary, however, these features widen the algorithm design space by enabling near-optimal options of structures with similar performance, such as using texture-driven topology optimization with personalized designs [HLG19], and bone-like infill optimization [WAWS18].

Large-scale topology optimization at high resolution has always been a challenging problem [MLR*21]. [AALS17] realized giga-voxel resolution plane wing topology optimization with the parallel operation of supercomputers in a few days. [WDW16] developed a high-performance GPU solver based on the algebraic multigrid method to improve the solving efficiency while ensuring convergence, thereby realizing million-voxel resolution optimization in
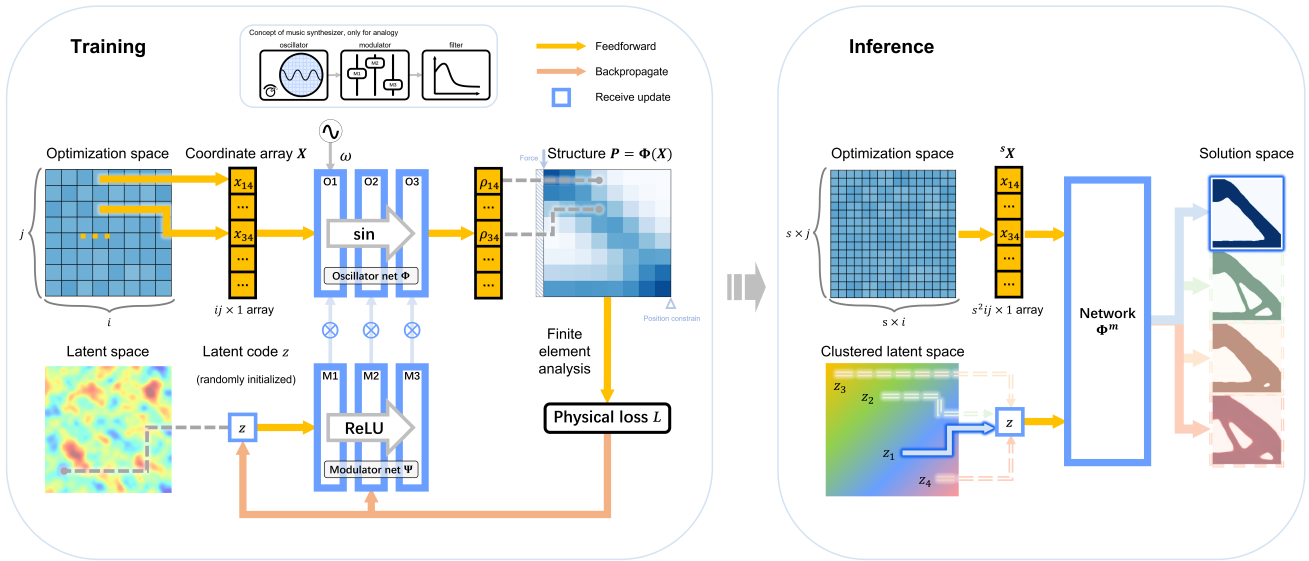
Figure 2: *NSTO system diagram. For single structure optimization, the oscillator network imports coordinate array X from optimization space and exports structure P, finite element analysis is performed to get the loss L. For solution space generation, the modulator network activates. Multiple subtasks are simultaneously optimized, and their corresponding latent codes z are randomly initialized, input, and optimized with network parameters. At inference, the user may super-resolve the solutions by taking a denser grid and generating novel solutions through feedforwarding interpolated latent code. The upper icons are for analogy.*

a few minutes with a personal computer. [LHZ*18] proposed a narrow-band topology optimization that focuses the computation on the generated thin structure area instead of equally solving the entire optimization space. They successfully optimized a billion-voxel bird beak structure using a workstation in 113.19 hours.

2.2. Neural field

Neural computing provides powerful tools for improving computation efficiency. One of the main-stream trends is training end-to-end data-driven networks for non-iterative topology optimization [WXP*21; WTTL20]. However, the generalization and interpretability of data-driven networks are still doubted, and their optimization quality is highly subject to the training data.

The recent development of the neural field [XTS*21] has ignited new opportunities in fields such as visual computing, signal processing, and physical simulation. A neural field, or coordinate-based neural network, is a neural implicit re-parameterization of signals. The network inputs a discrete sampling position and outputs the signal sampling value (e.g., image RGB value and audio waveform). It can date back to the compositional pattern producing networks (CPPN) [Sta07] in 2007; and after a series of fruitful research and development since 2019, it can now represent high-fidelity signals [TSM*20; SMB*20; MLL*21]. In the successive studies of [SMB*20; TSM*20], the Fourier domain neural field demonstrated improved high-frequency expression, which also enabled frequency adjustment of the network output [DPB21].

As for neural represented topology optimization, [HSG19] proactively proposed that the neural re-parameterization of topolo-

gies improved the optimization quality. [CS21; ZLZ*21] further verified various network models with classical tasks. [ZLCT21] proposed the converse density space objective, which applies the step-wise predicted structure to guide the network fitting.

Recent studies on implicit-neural-represented structural topology optimization were generally at an early stage and usually did not address practical issues such as optimization quality and computation cost. First, on the optimization quality, previous networks struggled to optimize high-frequency structure details [DT20; CS21], which are especially important for structures under distributed external force. Second, the optimization convergence was unsatisfactory, as hundreds of iterations were required for a classic benchmark task [ZLZ*21] (Messerschmitt-Bölkow-Blohm beam), while conventional explicit methods cost dozens of iterations [HX07; FS20]. Although several studies have demonstrated exciting progress, such as with the mesh-independent FEA solver (still tradeoff with the computation time) [ZLCT21], the overall effects are doubted due to the deficiency in the verification and comparison of the optimization-performance under complex tasks.

In conclusion, topology optimization provides a potent tool for high-quality structure design. However, its enormous computation hinders its popularity among ordinary users. To achieve the overall balance between the computation time and the structure quality, we contribute to a neural topology optimization framework for efficiently generating near-optimal structures with an appreciable reduction of time, and also enabling instant generation of novel solutions by performing multi-object optimization and latent space interpolation. Comprehensive comparison and various tasks were carried out to support the above statement.

3. Neural synthesizing topology optimization

NSTO consists of three parts:

- Sec. 3.1: A physical solver that adopts finite element analysis (FEA) to compute the structure performance.
- Sec. 3.2: An oscillator network that represents the structure in an implicit neural form for super-resolution.
- Sec. 3.3: A modulator network that enhances the network expressiveness for multi-structure generation.

The system overview based on these three sections is briefly shown in Equation 1 and Figure 2. First, the oscillator network Φ inputs an $n \times 3$ coordinates array X to generate an initial structure P. Second, the physical loss function L of the structure P is computed from the FEA solver. Third, the gradient of loss L is backpropagated as $BP(L)$ to update the network Φ. After the iterations of the above steps, the user may infer a super-resolved structure ${}^{s}P$ by inputting the coordinates array of an s-times denser grid ${}^{s}X$, similar to the image super-resolution methods [CLW21].

$$
\begin{aligned}
Training : &\begin{cases} P = \Phi(X) \\ L = \text{FEA}(P) \\ \Phi \Leftarrow \text{BP}(L) \end{cases} \\
Inference : &\quad {}^{s}P = \Phi({}^{s}X)
\end{aligned} \quad (1)
$$

The modulator network is activated to optimize multiple structures under varying boundary conditions or constraints (e.g., volume constraints) to interpolate solutions. The dual networks (i.e., oscillator and modulator) Φ^{m} input a latent code z with a coordinate array X and generate a corresponding structure as $P_i = \Phi^m(X, z_i)$, where i is the index of the subtasks. Procedures in Equation 1 are repeated to compute the loss and update the network and latent code.

3.1. Physical solver

This section establishes the physical loss function $L = \text{FEA}(P)$ and its gradient, and then introduce parameters required for FEA.

Structure P is represented by an $i \times j$ matrix $P_{ij} = \rho_{ij}^{\tau}$, according to the Solid Isotropic Material with Penalization (SIMP) method [Ben89]. Each of its element density $\rho \in [0,1]$ tells the material existence at that position. To comprehend this better, the reader may imagine a binary image P whose binary pixel values ρ describe the shape. The penalty factor τ is to enhance the convergence of the material density to binary values for clearly judging if the material should be allocated to that position [SP98], since an intermediate density such as 0.5 is practically meaningless in manufacturing.

Compliance C is selected as the optimization object in this study. Structure compliance minimization is a popular topic for designing lightweight and strong structures with minimal deformation under the same volume and load. As shown in Equation 2, C is the sum of all the finite elements' compliance weighted by P and formed by the structure deformation U and the element stiffness matrix K_e. A volume constraint is imposed, where $V(P) = \sum P$ is the structure volume, V_0 is the 100%-filled volume of the optimization space, and δ is the volume fraction.

$$
\begin{aligned}
\underset{P}{\text{argmin}}\, C(P) &= \sum P U^T K_e U \\
s.t. \quad V(P)/V_0 &\leq \delta
\end{aligned} \quad (2)
$$

Loss function L is constructed using the Augmented Lagrangian method [FG00], which transfers the volume-constrained compliance minimization problem into unconstrained form, as shown in Equation 3. λ is the Lagrangian multiplier that updates as $\lambda_{k+1} = \lambda_k + 2\sigma_k(\frac{V(P)}{V_0} - \delta)^2$, and σ_k is the penalty factor that exponentially grows with the iteration k, whose base number is empirically set as 1.1. A mean-squared-error volume constraint $(\frac{V(P)}{V_0} - \delta)^2$ is adopted to avoid convergence oscillations. Thus, the Lagrange multiplier and the penalty term are quadratic and fourth-power.

$$
\underset{P}{\text{argmin}}\, L(P) = C(P) + \lambda(\frac{V(P)}{V_0} - \delta)^2 + \sigma(\frac{V(P)}{V_0} - \delta)^4 \quad (3)
$$

Loss gradient $\frac{\partial L}{\partial \rho}$ with respect to each element's density is computed as shown in Equation 4, where the volume gradient $\frac{dV}{d\rho} = 1$. In Sec. 3.2, this loss gradient will be used to perform network backpropagation based on the chain rule.

$$
\frac{\partial L}{\partial \rho} = -\tau \rho^{\tau-1} U^T K_e U + 2\lambda(\frac{V(P)}{V_0} - \delta) + 4\sigma(\frac{V(P)}{V_0} - \delta)^3 \quad (4)
$$

With the loss function and its gradient established, we now introduce the necessary physical parameters for loss computation, following the standard FEA process.

First, we assemble the structure stiffness matrix K. The element stiffness matrix $K_e = \int_{v_e} B^T c B dv_e$ is computed from the strain matrix B, the material constitutive model c and the element volume v_e. In NSTO, we respectively apply the rectangular and hexahedron elements to 2D and 3D cases, each element having its specific B, c, and v_e. K is thus assembled from K_e by adding the K_e of each element to the corresponding position with the same degree-of-freedom index in the blank K matrix [Rao17].

Second, we obtain the structure deformation U (i.e., element-nodal-wise displacement) by solving the large-scale linear equation, Equation 5, where F is the external force. This is the most computationally expensive step in compliance-minimization topology optimization, as it usually takes up to 75% to 95% computation time in both the state-of-the-art and our methods. The GPU-based algebraic multigrid method (AMGX) [NAC*15] is applied for acceleration, as it is 1 to 2 orders of magnitude faster than the analytical solver in our experiment setup (Sec. 4.1). Specifically, the preconditioner is set as a multi-layer v-hierarchy with the Jacobi smoother. The convergence tolerance and the max iteration are set at $1e - 8$ and 100, respectively, for stable convergence in various tasks. Although the AMGX setup can be further optimized, we emphasize that NSTO focuses on the efficiency improvement brought by the implicit-neural-representation framework rather than by the linear equation solver, and there is no contradiction between the combined implementation of NSTO and advanced numerical solvers.

$$
KU = F \quad (5)
$$

In conclusion, the physical loss function L and its gradient for training the networks are established and computed with an accelerated algebraic multigrid solver.
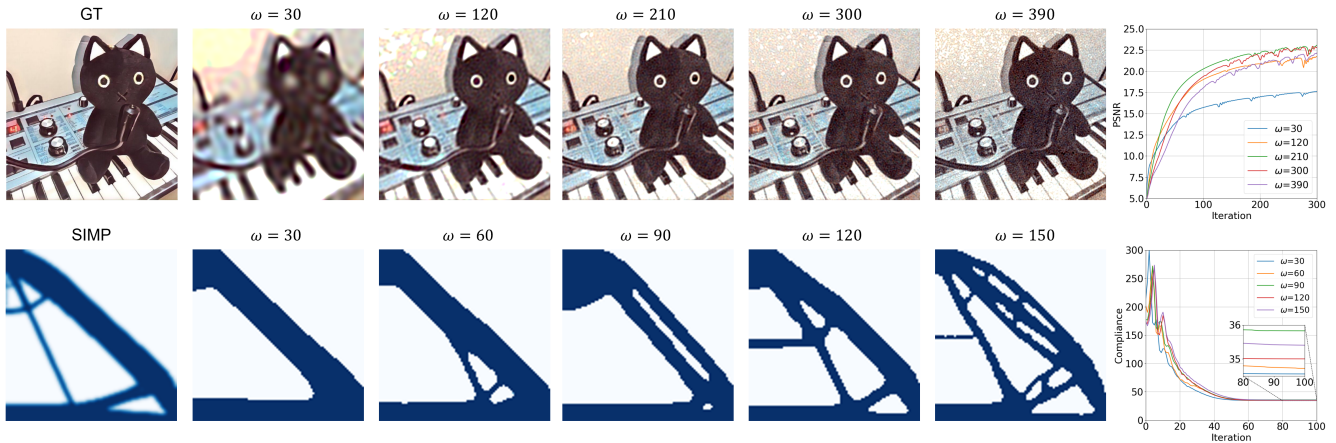
Figure 3: *The effect of the network frequency tuning mechanism on image fitting and topology optimization. High-frequency image feature increases with the ascending hyperparameter ω, as with the structures. The solution qualities are measured with PSNR and compliance.*

3.2. Oscillator network

This section focuses on the oscillator network that reparameterizes the structure material layout with coordinates, as $P = \Phi(X)$. The structure super-resolution and frequency tuning mechanisms are subsequently introduced.

The oscillator network applies a SIREN network [SMB*20], namely, a sinusoidal activated multilayer perceptron [GD98]. It has been proven capable of complex signal representation [MST*20; ZLY*21] and has significantly stronger high-frequency expressiveness than ReLU-activated networks. Equation 6 shows the network's layer-wise output $\phi_i(x)$, where w, b, and i are the neuron's weight, bias, and the layer index, respectively. The final layer of the network standardizes its output to the $[0,1]$ interval for the material density ρ, as $\text{Atan}(x) = \pi^{-1} \arctan \alpha x + 0.5$. α is a hyperparameter that adjusts the sensitive interval of the network outputs and is empirically set at $\alpha = 0.1$ for smoother convergence.

$$\phi_i(x) = \sin(w_i \phi_{i-1} x + b_i), \; w_1 = \omega \phi_0 \qquad (6)$$

Therefore, the oscillator network inputs coordinate x and outputs material density ρ, creating an in-between analog relationship, as shown in Equation 7. In practice, a full-batch training $P = \Phi(X)$ is applied for faster gradient descent. All the sampled coordinates x are assembled as a $(i \times j \times k, 3)$ array in 3D cases, where i, j, k are the axial resolutions of the optimization space. The network and its output structure are updated once per epoch.

$$\rho = \Phi(x) = \text{Atan}[w_i(\phi_{i-1} \circ \cdots \circ \phi_0)(x) + b_i] \qquad (7)$$

The loss function gradient of the net parameters is obtained through the chain rule as $\partial L / \partial w = (\partial L / \partial \rho) \times (\partial \rho / \partial w)$. $\partial L / \partial w$ is computed with Autograd [PGC*17]. $\partial L / \partial \rho$ is manually computed to avoid the conspicuous gradient computing costs due to the iterative solving of $KU = F$ using the algebraic multigrid method.

Among the feasible optimizers including Adam [KB15], L-BFGS-B [ZBLN97], and resilient backpropagation (Rprop) [RB93], we selected Rprop due to its faster convergence in most

tasks, despite its lower convergence stability compared to that of Adam. To balance the convergence speed and quality, we also set the penalty factor $\tau \in [1.5, 3]$, as shown in Equation 4.

At the inference stage, the user may discretize the optimization space into an s-times denser grid $^s X$ to obtain the s-times super-resolved structure $^s P$ as Figure 2. One may imagine discretizing an image with a smaller pixel size so that its boundary is smoothed.

$$^s P = \Phi(^s X) \qquad (8)$$

The Fourier-featured network further enables the structure frequency tuning mechanism, which positively relates the structure frequency to the network hyperparameter ω initialized in the first layer, as $w_1 = \omega \phi_0$. This mechanism enables the generation of high-frequency structural details that are typically obtained at high-resolution computation, thereby improving the robustness against local structural failures. Users can also reduce the structural frequency for higher manufacturability. In practice, the maximal ω is limited to avoid generating invalid structures during the super-resolution, since an excessive frequency will cause signal aliasing and worsen structural optimality. For an analogy, we demonstrate this mechanism with the image fitting task shown in Figure 3, where the network inputs pixel coordinates, outputs RGB values, and is trained to fit a ground truth image with the mean-squared error. We observed that the fitted image moved from underfitting to overfitting when ω was increased.

In conclusion, the oscillator network implicitly represents a structure material layout by relating the element density ρ to its coordinate x, thereby bringing the functions of structure super-resolution and frequency tuning. There are also several differences from explicit topology optimization methods. First, the initial element density is randomly distributed around $\rho = 0.5$, which probably converges to more local minima (i.e., optimized structures) than the uniformly initialized density in conventional methods. Second, the density filter in conventional methods [Bou01] is spared since the implicit neural representation has a similar effect, which is discussed from the perspective of the neural tangent kernel [DJ21].

3.3. Modulator network

This section focuses on the modulator network, which is used to enhance the performance of multi-structure optimization under varying boundary conditions so that novel structures can be generated in boundary-condition intervals.

Structure generation is the second research object that we hope to address besides optimization efficiency. In practical scenarios, users often adjust design parameters repetitively for desired structures, and each adjustment requires a complete optimization. For this reason, the network is expected to optimize multiple solutions under different boundary conditions labeled with the corresponding latent code z, so that new solutions can be instantly generated by interpolating the latent space (i.e., the latent codes interval).

To this end, structures with different boundary conditions are simultaneously optimized for generating interpolated solutions. Different boundary conditions or constraints are reflected in the loss function L, as shown in Equation 3. Here, we introduce two tasks. First, a varying volume constraint is realized by assigning different volume fractions δ in the loss function L. Second, a varying external force is realized by adjusting the force $\boldsymbol{F}$ in Equation 5 for steering the compliance C in the loss function L. For example, two structures with latent codes z_1, z_2 and volume fractions $\delta_1 = 0.3, \delta_2 = 0.7$ are assembled as a 'dataset' and alternately optimized. Users are expected to obtain a structure with a volume of $\frac{V}{V_0} = 0.5$ through feedfowarding an interpolated latent code $\frac{z_1+z_2}{2}$. In brief, the multi-structure generation task is to minimize a varying loss function that caters to different boundary conditions.

We attempt to use a single oscillator network as an autodecoder [PFS*19] for the task. Autodecoder is a multilayer perceptron that inputs concatenated coordinates and latent code (x, z), and outputs the density ρ, as $\rho = \Phi(x, z)$. During the training, the randomly initialized latent code is updated together with the

Dual networks (NSTO)   Single network (auto decoder)

network parameters so that the encoding procedure in a typical variational autoencoder [KW14] is spared. At the inference stage, users may feedforward interpolated latent codes for novel structure generation. The autodecoder is simple in design and efficient in training. However, we found that a single autodecoder struggles with the multi-structure generation task, mainly due to the lack of network expressiveness.

The modulator network is thus applied to promote the network performance. It is a ReLU-activated multilayer perceptron Ψ with a size identical to the oscillator. It applies the same strategy as that with an autodecoder, as the latent code is initialized with a normal distribution and updated with the network parameters like a 'layer0', thus sparing the encoder. The latent code is used to modulate the oscillator network to enhance network expressiveness [MGB*21], rather than generating structures itself. Equation 9 shows the output of the layer $\psi_i(z)$ and the network $\Psi(z)$, where z is the latent code and $\boldsymbol{w}'$ and $\boldsymbol{b}'$ are the layer's weight and bias.

$$\begin{aligned} \psi_i(z) &= \text{ReLU}(\boldsymbol{w}'_i \psi_{i-1} z + \boldsymbol{b}'_i) \\ \Psi(z) &= \boldsymbol{w}'_i (\psi_{i-1} \circ \cdots \circ \psi_0)(z) + \boldsymbol{b}'_i \end{aligned} \qquad (9)$$
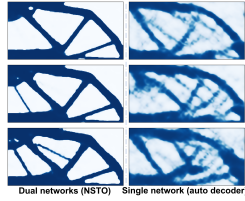
Algorithm 1 Neural synthesizing topology optimization

```
 1: Input: X, bc       # coordinate array and boundary condition
 2: Output: ˢP         # s-times super-resolved structure
 3:
 4: # Physical solver
 5: function FEA(X, bc)
 6:     F, δ, = bc
 7:     B, c, vₑ = ElemParas()      # element-dependent
 8:     Kₑ = ∫_{vₑ} BᵀcB dvₑ
 9:     K = assemble(Kₑ)
10:     U = AMG(K, F)
11:     C = ∑ PUᵀKₑU
12:     L = C + λ(V(P)/V₀ − δ)² + σ(V(P)/V₀ − δ)⁴
13:     return L, ∂L/∂ρ
14: end function
15:
16: # Single-structure optimization
17: procedure TRAINOSCNET(X, bc)
18:     OscNet.initialize()     # OscNet Φ
19:     optimizer.initialize()
20:     for epoch in range(e_max) do
21:         P = Φ(X)
22:         L = FEA(P, bc)
23:         Φ ⇐ BP(L)
24:         optimizer.step()
25:     end for
26:     return ˢP = Φ(ˢX)
27: end procedure
28:
29: # Multi-structure optimization, the main algorithm
30: procedure TRAINMODOSCNET(X, bc)
31:     OscNet.initialize()     # OscNet Φ
32:     ModNet.initialize()     # Modulated OscNet Φᵐ
33:     optimizer.initialize()
34:     for epoch in range(e_max) do
35:         for i in enumerate(subtasks) do
36:             Pᵢ = Φᵐ(X, zᵢ)
37:             L = FEA(Pᵢ, bcᵢ)
38:             Φᵐ ⇐ BP(L)
39:             optimizer.step()
40:         end for
41:     end for
42:     return ˢP = Φᵐ(ˢX, z)
43: end procedure
```

It enhances the expressiveness of the oscillator network through element-wise multiplication of each of its layers' outputs by the corresponding outputs of the oscillator, which are marked with the symbol $\otimes$ in Figure 2. For comparison, a single autodecoder that inputs concatenated latent code and coordinates $\boldsymbol{P} = \Phi(X, z)$ only causes a phase shift to the layer's output, as $\phi_i(x) = \sin(\boldsymbol{w}_{i,x}\phi_{i-1}x + \boldsymbol{w}_{i,z}\phi_{i-1}x + \boldsymbol{b}_i)$, so its expressiveness is weaker.

The dual networks Φ^m take the coordinate x and the latent code z respectively, and outputs the material density ρ of the z-labeled structure, as Equation 10. Through the feature maps of the two net-

works in training, we observed that both networks learn the optimal structure in cooperation. At inference, users can perform a simple network feedforward $P = \Phi^m(X,z)$ to infer a structure or perform an interpolation of latent codes for new designs.

$$\phi_i^m(x,z) = \psi_i(z)\sin(w_i\phi_{i-1}^m x + b_i)$$
$$\rho = \Phi^m(x,z) = \text{Atan}[\psi_i(z)(\phi_{i-1} \circ \cdots \circ \phi_0)(x) + b_i] \quad (10)$$

In conclusion, the modulator network is introduced to enhance the network's expressiveness. Therefore, the dual networks (oscillator and modulator) are capable of structure generation tasks under varying boundary conditions, enabling users to generate intermediate solutions by interpolating and feedforwarding the latent codes of optimized subtasks. The pseudo code of NSTO is shown in Algorithm 1, where function FEA summarizes the physical solver (Sec. 3.1), and the two procedures below summarize the training steps of the oscillator (Sec. 3.2) and the dual networks (Sec. 3.3).

4. Experiments

NSTO is experimentally introduced and verified from three aspects:

- Sec. 4.1: The experimental setup is briefly introduced.
- Sec. 4.2: The numerical performance of NSTO, including the convergence rate, optimization quality (compliance), and time, are demonstrated. The benchmark algorithms are compared.
- Sec. 4.3: Various applications are shown with instructions. Structures are 3D-printed for manufacturability verification.

4.1. Experiment setup

This section introduces the experimental environment, the network hyperparameters, and the training method.

NSTO experiments run on a desktop PC (Intel Core i9-10900KF, Nvidia RTX2080S 8GB, Ubuntu 16.04, Python 3.6), where GPU determines the network scale and FEA resolution.

The network hyperparameters were set as follows. The network width was commonly set at 512 for balanced VRAM consumption and performance, and the learning rate was fixed at 1e−4. The hyperparameter ω for structure frequency tuning was commonly set at 60 and could be adjusted according to the resolution and optimization object. We empirically set ω less than twice the maximum axial resolution to avoid the aliasing caused by an exorbitant frequency. The latent code dimension was set at 1D and 2D for the intuitive inference of the latent space and can be slid up for challenging tasks.

Full-batch training was performed by assembling all the discretized coordinates of the optimization space as an array and feeding it to the network as a whole so that the network outputs would be the complete element density values of a structure. The feedforwarding, FEA, backpropagation, and network updating are performed once per epoch. Generally, optimization converges after tens of epochs, depending on the loading condition.

4.2. Performance

This section introduces the numerical performance of NSTO from two aspects: *a:* its convergence and structural performance compared to those of the benchmark method; *b:* computation time for tasks with complex structures and boundary conditions.
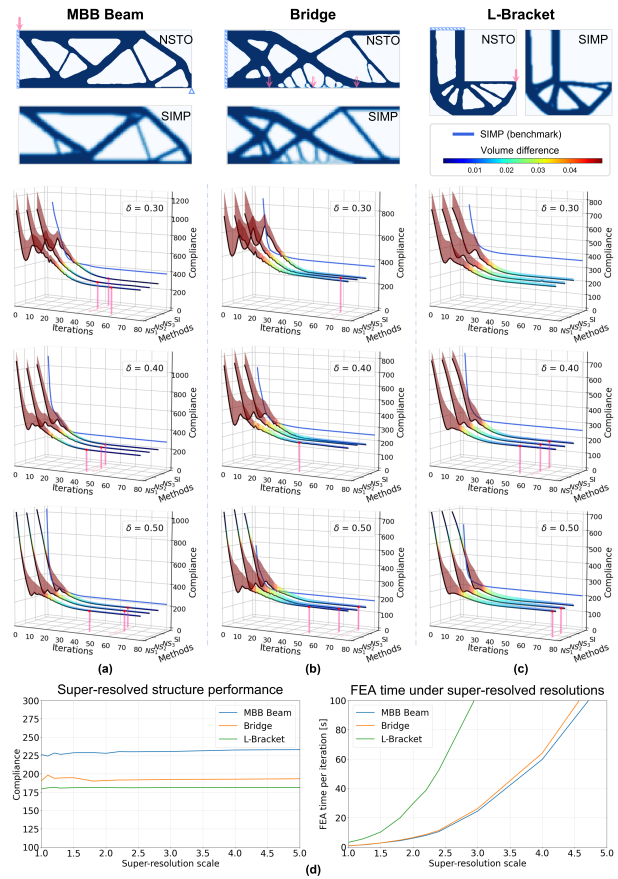

Figure 4: *Convergence and performance of NSTO, compared with SIMP in (a) MBB beam, (b) Bridge, and (c) L-Bracket optimization, where optimized structures are shown above the convergence plots. Red arrows indicate external force, and blue marks indicate fixed position constraints. (d) Performance of several super-resolved NSTO results and the ordinary FEA iteration time under the same resolutions.*

First, the convergence and the optimized structural performance (compliance) of NSTO and the benchmark SIMP are compared [FS20]. The Solid Isotropic Material with Penalization (SIMP) is a structural topology optimization algorithm that is widely applied in academies and industries for its simple implementation, fast convergence, and capability to optimize non-linear geometries and materials. It is also generally referenced as an algorithm benchmark. For fairness, we used the same algebraic-multigrid-method FEA solver [BOS22] for its stable solutions, computed in the same environment, and set up optimal configurations for SIMP by tuning the density filter radius. Note that the different initialization of NSTO and SIMP is only a derivative result of the algorithms' formulation of constrained optimization rather than being a decisive factor of structural optimality. Specifically, the initialization strategy of SIMP comes from the tight volume constraint imposed by its optimality criteria method, which is not necessary for NSTO.
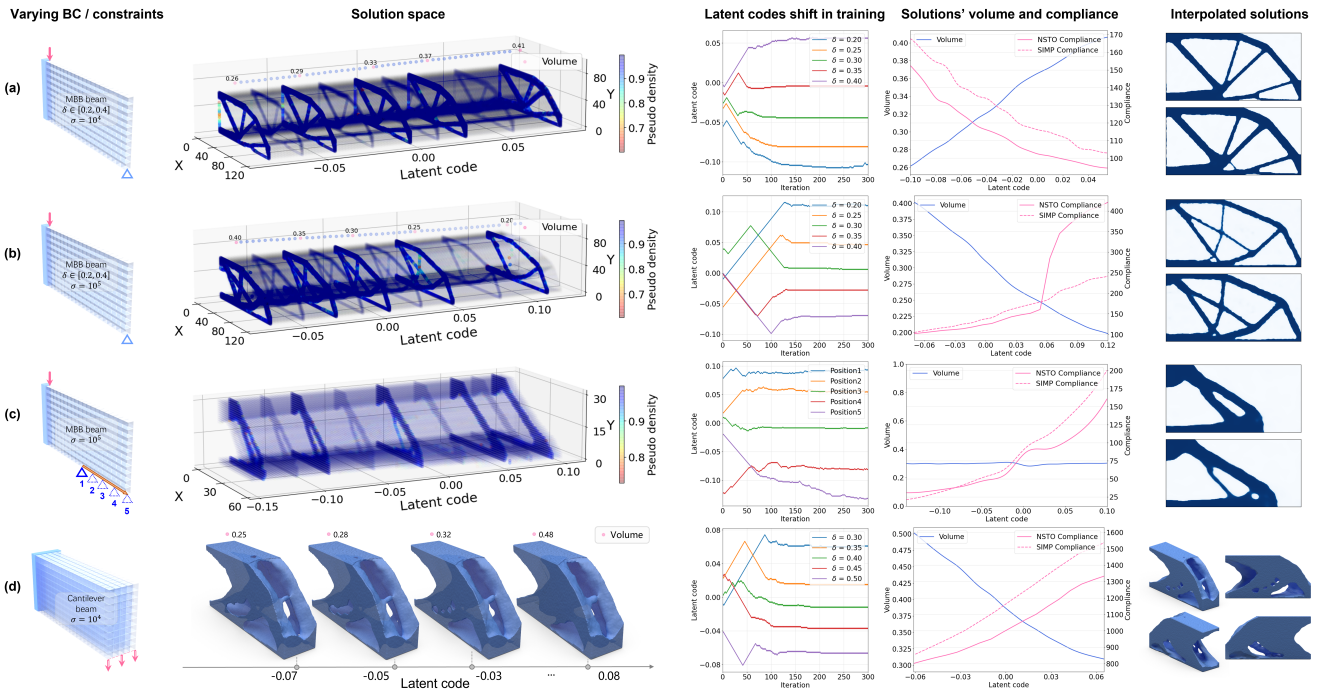
Figure 5: *Generated solution space of (a, b) varying volume-constrained MBB beams with different penalty factor σ; (c) varying supporting-position MBB beams; and (d) varying volume-constrained cantilever beams. Original subtasks are shown in solid lines, and interpolated solutions are shown in semi-transparent. The upper dots indicate the volume, and the color in the solution space indicates the pseudo density. The boundary conditions or constraints are shown in 3D for better visualization.*

Table 1: *NSTO and SIMP benchmark solutions in 80 iterations. C is the compliance, V and δ are the optimized and constrained volume fraction.*

	MBB Beam		Bridge		L-Bracket	
Method	C↓	V↓ (δ)	C↓	V↓ (δ)	C↓	V↓ (δ)
NSTO (ω=60)	292.88	0.31 (0.30)	287.30	0.31 (0.30)	**223.05**	0.32 (0.30)
NSTO (ω=120)	**287.18**	0.31 (0.30)	279.65	0.31 (0.30)	223.39	0.31 (0.30)
NSTO (ω=180)	297.33	0.30 (0.30)	**267.09**	0.31 (0.30)	225.82	0.31 (0.30)
SIMP	373.89	0.30 (0.30)	343.37	0.30 (0.30)	352.10	0.30 (0.30)
NSTO (ω=60)	224.44	0.41 (0.40)	207.95	0.41 (0.40)	180.08	0.42 (0.40)
NSTO (ω=120)	226.36	0.41 (0.40)	208.18	0.41 (0.40)	**179.46**	0.42 (0.40)
NSTO (ω=180)	**221.80**	0.41 (0.40)	**190.48**	0.41 (0.40)	181.35	0.42 (0.40)
SIMP	271.00	0.40 (0.40)	241.05	0.40 (0.40)	249.39	0.40 (0.40)
NSTO (ω=60)	188.46	0.51 (0.50)	164.23	0.51 (0.50)	155.69	0.51 (0.50)
NSTO (ω=120)	186.69	0.51 (0.50)	150.46	0.51 (0.50)	155.58	0.51 (0.50)
NSTO (ω=180)	**184.79**	0.51 (0.50)	**149.47**	0.51 (0.50)	**155.06**	0.52 (0.50)
SIMP	213.80	0.50 (0.50)	174.21	0.50 (0.50)	198.46	0.50 (0.50)

Convergence of NSTO was tested in three benchmark structural topology optimization scenarios [VBO*17]: the optimization of the Messerschmitt-Bölkow-Blohm (MBB) beam, bridge, and L-Bracket. Their boundary conditions covered the ranges of the concentrated and distributed external force and the irregular optimization space, as shown in the top part of Figure 4. Specifically, the MBB beam was fixed at two bottom ends and received a top-central concentrated force. The bridge structure was also fixed at two ends but subjected to uniformly distributed force on the lower

surface. The L-Bracket was optimized in an L-sized optimization space, where it was fixed at the top and received a bending force on the right. Conventionally, the first two tasks leveraged symmetric constraints for computation efficiency, so half of these structures were optimized. During the experiments, the MBB beam and bridge were optimized at a 120×40 resolution, and the L-Bracket was optimized at a 100×100 resolution. The detail boundary condition setting followed the benchmark [VBO*17].

The convergence results under different volume fractions δ and frequency hyperparameters ω are enumerated in the middle of Figure 4. The black lines and the rainbow-colored strips indicate the compliance and the volume loss at the current iteration, respectively. The blue lines indicate the compliance of the benchmark SIMP method. The red dots and the pink lines indicate the convergence iteration of NSTO, whose criteria are set as a 0.3% compliance difference from the previous iteration and a 1% volume difference from the fraction. The results show that NSTO under different ω values converged towards optimal solutions at a stable pace.

Performance of the structures were comprehensively compared. As shown in Figure 4 and Table 1, NSTO had lower compliance than SIMP in all the benchmark cases. The clear structure boundaries of the NSTO solutions also led to minor efforts in mesh post-processing. As for the volume difference with the volume fraction, NSTO had a 1% error mainly due to the loose volume constraints from the augmented Lagrangian method. Besides the $1\times$ resolu-

tion structure comparison, we super-resolved the solutions to observe their shape and performance change. The structures were super-resolved 10 times, but no artifacts appear, as shown in the top of Figure 4. Then, we performed FEA on the super-resolved structure and observed that the compliance remained stable in the tested range ($1-5\times$ axial super-resolution), which verified the feasibility of implicit-neural-represented structure optimization. Compared with the negligible time for structure super-resolution (i.e., a simple network feedforward), performing FEA at the same resolution was much more time-consuming, as shown in the bottom right part of Figure 4. The above results verify the first objective of NSTO, that is, to improve the computation efficiency of structural topology optimization for ordinary users. We re-emphasize that NSTO contributes to the overall structure optimization framework rather than to the numerical FEA solver on which it is built.

Second, the dual networks were examined in three solution space generation tasks: the varying volume-constrained MBB beam task MBB_v, the varying supporting-position MBB beam task MBB_s, and the varying volume-constrained 3D cantilever beam task CB_v, as shown in Figure 5. The latent codes of all the tasks were set at 1D to directly view the solution space $P = \Phi^m(X, z)$ that corresponded to the latent codes $z: \mathbb{R}^1$.

The MBB_v task adopted varying volume constraints to generate MBB beams with intermediate volume as Figure 5 (a, b). To this end, the network simultaneously trained five optimization subtasks with equally distributed volume constraints as $\delta \in [0.20, 0.25, 0.30, 0.35, 0.40]$. Note that the loss function L was subject to varying constraints δ, which are necessary for other 1D manipulations such as varying external force. During the training, the latent codes of each subtask gradually mapped to the 1D latent-space positions in the order of volume constraint δ and presented an approximately linear relationship with δ. We uniformly sampled along the z axis at the inference stage and computed the volume and compliance of the corresponding output structure. Most interpolation solutions perform better than SIMP, consistent with the observations in Figure 4. We also note that when an excessively large penalty factor σ (Equation 3) was imposed on the volume constraint, there was a trade-off of optimality, which was manifested as ascending compliance in Figure 5 (b) and should be avoided.

The MBB_s task adopted five equally separated fixed-position constraints to generate MBB beams supported at intermediate positions, as shown in Figure 5 (c). The constraint moved from the middle bottom to the right end of the optimization space, labeled as position $[1, 2, 3, 4, 5]$. The movement of the constraint was achieved by zeroing the corresponding values in the structure deformation array U to be 0 at the new fixed positions. Unlike the volume constraint, the fixed-position constraint indirectly influenced the loss through the compliance C. At the inference stage, smoothly evolving structures were generated, showing NSTO's generation capability with more complex constraints.

The CB_v task of generating varying-volume cantilever beams was performed to verify the dual networks' capability in more challenging 3D tasks, as shown in Figure 5 (d). Within the volume constraint interval of the five subtasks $\delta \in [0.30, 0.35, 0.40, 0.45, 0.50]$, it showed lower compliance than SIMP, which verifies the validity of the solution space generation in 1D latent code scenarios.

Table 2: *Resolution and computation times. $*$ marks the solution space generation tasks.*

Object	Resolution	V/V_0	C/C_0	Iter.	FEA(s)	Total(s)
Shelf	$62 \times 15 \times 62$	0.26	1.85	71	160.37	160.55
Desk	$39 \times 27 \times 30$	0.14	22.71	166	199.92	200.33
Torsion	$20 \times 20 \times 40$	0.36	0.23	241	144.54	145.20
Daruma	$18 \times 15 \times 17$	0.07	1.00	85	15.32	15.51
Atlas	$32 \times 32 \times 60$	0.34	0.44	54	135.71	135.86
Guitar	240×177	0.76	1.04	311	56.41	57.12
$*MBB_v$	120×60	[0.20, 0.40]	/	300×5	139.66	146.27
$*MBB_s$	60×30	0.30	/	300×5	61.45	68.09
$*CB_v$	$40 \times 20 \times 10$	[0.30, 0.50]	/	300×5	444.97	495.57
$*Drums$	$180 \times 1 \times 60$	0.30	/	71×8	358.67	364.98

Computation time of NSTO and SIMP in the Daruma task			
Algorithm	NSTO	[1]SIMP	[2]SIMP	[3]SIMP
Resolution	$18 \times 15 \times 17$	$18 \times 15 \times 17$	$36 \times 30 \times 35$	$55 \times 46 \times 53$
Total(s)	15.51	16.03	120.37	475.56

Finally, the overall computation time of NSTO for various challenging tasks is introduced in Table 2. Among these tasks, the fastest Daruma doll infill optimization took only 15s, the complex Atlas sculpture 135.86s, and the solution space generation of drum shells 364.98s, which generated 7 drum shells. For reference, the state-of-the-art research [WDW16] developed a high-performance GPU solver that optimized *kitten* in 263.33s, *bunny* in 400.36s and *Neptune* sculpture in 709.16s. The Daruma task exemplifies the computation time of NSTO and SIMP. Within 85 iterations, the two methods used up similar times at the same FEA resolution. Nevertheless, the SIMP computation time increased cubically at higher resolutions (i.e., at the $1\times, 2\times, 3\times$ super-resolution) and could have exceeded the VRAM capacity.

In NSTO, assembling the global stiffness matrix K and solving the structure deformation uses up most of the computation time, whereas the other parts have negligible costs. For example, the network's average feedforward and backpropagation time were 0.0645s and 0.0766s for a 240×80 resolution MBB beam optimization. The stiffness matrix assembly time was 0.486s, and the linear elasticity solving time was 0.160s (with AMGX [NAC*15]), taking up around 82.04% of the computation time. Similar time uses of around 80% to 90% were observed at various resolutions.

In conclusion, the convergence, optimization performance, and generation capability are presented and analyzed. Compared with the benchmark SIMP, NSTO demonstrated high optimization quality. The modulator and oscillator networks also demonstrated structure generation capability under direct and indirect varying boundary conditions or constraints.

4.3. Applications

This section demonstrates a wide range of personal fabrication applications with NSTO, including *a:* 2D optimization tasks of printing infills and electric guitar body; *b:* 3D optimization tasks of the bookshelf, table, torsion, Daruma doll,, and Atlas sculpture; *c:* multi-structure optimization of the jazz drum shells.

3D Printing infill aims to save materials to ensure structural strength. It is also one of the most practical applications of struc-
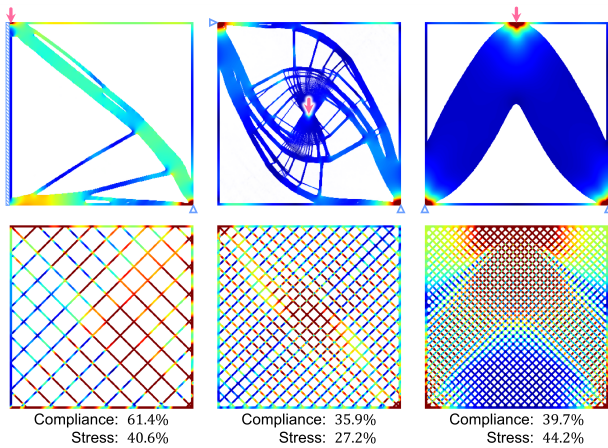
Figure 6: *Comparison of the topology-optimized and grid infills under different infill percentages and boundary conditions. The red and blue colors indicate high and low stress, respectively.*
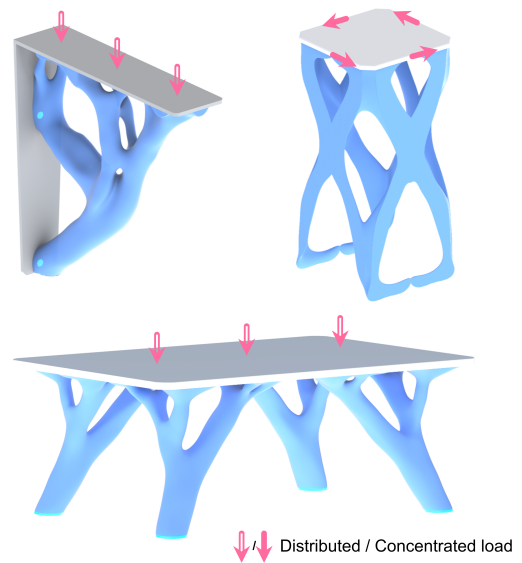

Figure 8: *Structure generation results of shelf, desk and torsion structures. Red arrow show the external load's positions and it's type, light blue areas indicate the fixed positions.*

Figure 7: *Optimized electric guitar body under planar-distributed loads, featuring spiderweb-like features.*

tural topology optimization in personal fabrication. Here, we compare three pairs of topology optimized infill structures with regular grid infills under different external forces, as shown in Figure 6. The optimized structures have the same volume fraction (i.e., material consumption) as the grid infill: 22.3%, 37.7%, and 57.9%. Compared with the uniform grid infill, the optimized infill structures showed lower compliance of 61.4%, 35.9% and 39.7%, and lower mean stress of 40.6%, 27.2% and 44.2% respectively, therefore allowing the design of more robust loaded structures.

Guitar, flute [UPSW16], metallophone [BLT*15], and various 3D-printed musical instruments were created by artists and hobbyists because of their unique appearance and acoustic properties. Here, we optimized an electric guitar body under a planar-distributed load, as shown in Figure 7. The 2D optimization results were stretched up and Boolean-intersected with the original body to achieve unique designs. The solution ensured structural stiffness and demonstrated a spider web-like personalized feature, which could be stronger and more attractive than repetitive features.

We reviewed NSTO's performance on non-shape-constraint and complex-shape-constraint tasks with two applications: the book-

shelves, table, and torsion tasks, and the Daruma dolls and Atlas sculpture tasks.

Bookshelf, desk, and torsion structures were successively optimized, as shown in Figure 8. They were under the representative boundary conditions of the distributed force and torque, and no shape constraints were applied. After the optimization, all the structures were meshed with the marching cube algorithm [LC87].

The shelf was fixed at two points and subjected to a uniformly distributed force. Compared with the concentrated-load cases, the optimization objects under the distributed load tended to grow more high-frequency branches, which indicates the convergence and high-frequency expressiveness of the optimization algorithm. The solution revealed that the thick structures of the shelf were extended to the fixed end, and the thin structures were rooted to the loaded surface, verifying that NSTO can generate sufficient high-frequency structural details in 3D cases.

The desk was subjected to a distributed normal force on the surface and was supported at four corners. For simplicity, mechanical symmetry constraints were applied by fixing the horizontal degrees of freedom of the two symmetry planes (the vertical axis points towards the height direction) to optimize a quarter of the desk. The solution presented a natural dendritic pattern that diffused from the constraint position to the distributed load.

The torsion structure was subjected to four circumferential external forces as torques while fixed at the bottom. One-quarter of the structure was symmetrically mapped to the other three parts through shape regulation to form a more aesthetic solution. Thanks to the representation freedom that NSTO provided, the mapping between the structural features can also develop in various ways other than symmetric shape regulation, such as multi-scale struc-
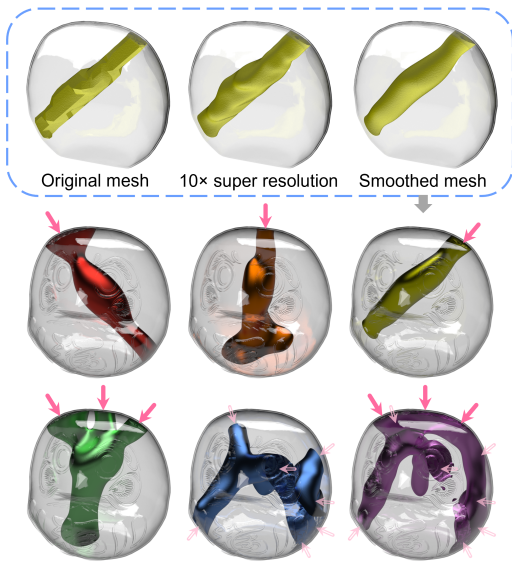
Figure 9: *The optimization process of Daruma dolls and the final solutions under various load conditions.*
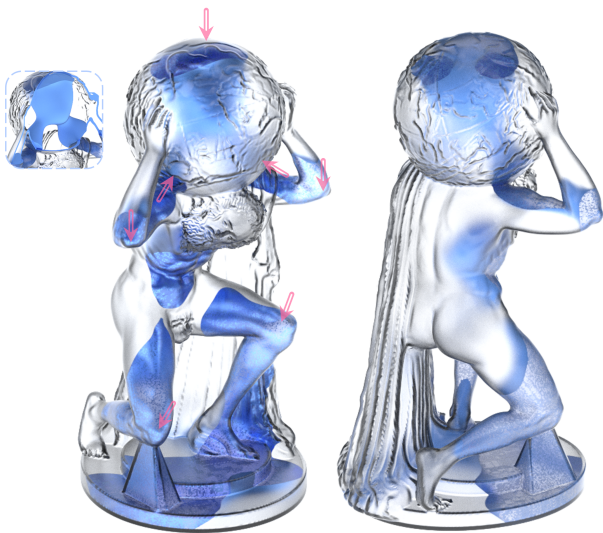

Figure 10: *Optimized Atlas sculpture. A section drawing is attached for viewing inside the sculpture.*

tural optimization [ZSCM17] or the personalized topology texture [HLG19]. In brief, the above solutions intuitively demonstrated NSTO's capability in non-shape-constraint tasks.

When optimizing structures with shape constraints, the boundary structures are first voxelized into structured grids. Then, the material density values outside the structure are set at 0. A thin shell is preserved for 3D printing.

Daruma dolls were optimized under several groups of external

Figure 11: *Solution space of drum shells, where diameter height ratio is the varying boundary condition. The top image shows the mapping between annular and planar structures, and the bottom image shows the solutions with latent codes.*

forces, including the concentrated force at the top and the discrete force over the whole body, as shown in Figure 9. NSTO generated an intuitive structure along the direction of the concentrated force, and discrete-force-basis structures are much more complex. The upper part of Figure 9 presents the optimized structure with a 10-times axial super-resolution (i.e., a 1000-times voxel super-resolution). Then, surface Laplacian smoothing was selectively applied. It should be noted that the Daruma optimization took only around 15s.

Atlas sculpture was selected for the complex-shape-constraint optimization, as shown in Figure 10. Multi-directional external forces were added to its raised sphere and joints. We observed that supporting structures were grown from the contact position to strengthen the local stiffness, proving that NSTO is effective for complex-shape-constraint situations.

Finally, we leveraged the dual networks for the drum shell optimization under a varying volume constraint, as shown in Figure 11. *Drum shells* are annular thin-walled structures responsible for carrying the load. While resisting radial tension and axial pressure, they need to be lightweight to retain greater resonance during vibration for producing longer sustain, which is favored by drummers. To this end, the drum shells were optimized under volume constraints, radial tension, and axial pressure, where the diameter height ratio was set as the dynamic boundary condition to find the optimal design scheme for the drums of different sizes. The solutions will be used as the lightweight outer layer of the drum shell to improve the structural stiffness without affecting the inner air vibration.

Figure 12: *3D-printed topology-optimized structures using the fused deposition modeling 3D printer (Ultimaker 3).*

The diameter height ratio of the drum shell ranged from 5.5/14 for the snare drum to 16/16 for the floor tom. Therefore, we used 1D latent code and optimized the solutions through 8 subtasks with diameter height ratios uniformly distributed between [0.3, 0.7]. To make more efficient use of the optimization space, the shell was mapped into a $180 \times 1 \times 60$ flat plate. The bottom constraints were evenly moved upward to modify the diameter height ratio. At the inference stage, users may slide along the 1D latent space at the inference stage and freely select the desired solution according to each drum's diameter height ratio.

The topology-optimized structures were printed with the fused deposition modeling printer to verify the printability, as Figure 12.

4.4. Limitations and future works

The selection and refinement of the structure discretization influenced the super-resolved results of the optimized structure. Specifically, when optimizing structures at a single resolution, the network may overfit current solutions after extensive training, thus causing resolution-dependant artifacts such as zigzag structure boundaries due to the rectangular finite elements. A potential solution is to compute the structural performance with random numbers, such as through Monte Carlo integration, although the solving efficiency provided by the grid elements will be lost [LHZ*18].

Our future work is to explore the strategy of solution spaces optimized under multiple boundary conditions or constraints. Currently, we implemented a 2D solution space under varying volume constraints and supporting positions. The solution space demonstrates a radiative pattern of constraints, calling for more efficient trajectory
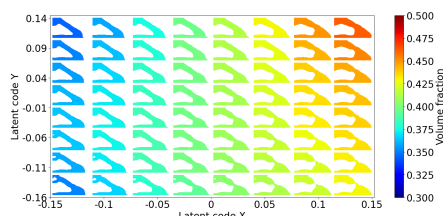
planning strategies and methods for imposing varying boundary conditions or constraints in high-dimensional situations.

The relationship between network frequency features and the structural performance at various resolutions can be the topic of another future work. In NSTO, the network frequency hyperparameter ω has a positive relationship with the structure details, resulting in compliance change. Viewing and optimizing the structural performance from the Fourier space is a potential angle.

5. Conclusions

Previously, most structural topology optimization frameworks were for pure numerical research or industrial applications, leaving a gap between the requirements of ordinary users and algorithm developments. Therefore, we proposed the NSTO framework, which provides optimal structures conducive to manufacturing at a lower computational cost and enables novel structure generation.

NSTO significantly improves the optimization efficiency from the resolution aspect to ensure optimal performance. It also expands the solution space purely under the physical constraints of multiple subtasks through self-supervision. The optimization results under complex boundary conditions are mapped to a latent space, so users may infer novel solutions without repeating the topology optimization process. With the injection of creativity, users are expected to optimize sets of functional structures for decoration or interaction. We supported the above claims through a comprehensive study of the algorithm performance, comparison with the benchmark method, and examination of a wide range of tasks with complex geometric constraints and boundary conditions.

Acknowledgements

References

[AALS17] AAGE, NIELS, ANDREASSEN, ERIK, LAZAROV, BOYAN S., and SIGMUND, OLE. "Giga-voxel computational morphogenesis for structural design". *Nature* 550.7674 (2017), 84–86. ISSN: 14764687. DOI: 10.1038/nature23911 2.

[Ben89] BENDSØE, M. P. "Optimal shape design as a material distribution problem". *Struct Optim* 1.4 (1989), 193–202. ISSN: 09344373. DOI: 10.1007/BF01650949 2, 4.

[BLT*15] BHARAJ, GAURAV, LEVIN, DAVID IW, TOMPKIN, JAMES, et al. "Computational design of metallophone contact sounds". *ACM Transactions on Graphics (TOG)* 34.6 (2015), 1–13 10.

[BOS22] BELL, NATHAN, OLSON, LUKE N, and SCHRODER, JACOB. "Pyamg: algebraic multigrid solvers in python". *Journal of Open Source Software* 7.72 (2022), 4142 7.

[Bou01] BOURDIN, BLAISE. "Filters in topology optimization". *International journal for numerical methods in engineering* 50.9 (2001), 2143–2158 5.

[BPS00] BUHL, T., PEDERSEN, C. B.W., and SIGMUND, O. "Stiffness design of geometrically nonlinear structures using topology optimization". *Struct Multidiscip Optim* 19.2 (2000), 93–104. ISSN: 1615147X. DOI: 10.1007/s001580050089 2.

[BS03] BENDSOE, MARTIN PHILIP and SIGMUND, OLE. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2003 2.

[BTX20] BI, MINGHAO, TRAN, PHUONG, and XIE, YI MIN. "Topology optimization of 3D continuum structures under geometric self-supporting constraint". *Addit Manuf* 36 (2020), 101422. ISSN: 22148604. DOI: 10.1016/j.addma.2020.101422 2.

[CLW21] CHEN, YINBO, LIU, SIFEI, and WANG, XIAOLONG. "Learning Continuous Image Representation with Local Implicit Image Function". *Proc IEEE/CVF Conf Comput Vis Pattern Recognit*. 2021, 8624–8634. ISBN: 9781665445092. DOI: 10.1109/cvpr46437.2021.00852. arXiv: 2012.09161 4.

[CS21] CHANDRASEKHAR, AADITYA and SURESH, KRISHNAN. "Multi-Material Topology Optimization Using Neural Networks". *CAD Comput Aided Des* 136.3 (2021), 1135–1149. ISSN: 00104485. DOI: 10.1016/j.cad.2021.103017 3.

[DJ21] DUPUIS, BENJAMIN and JACOT, ARTHUR. "DNN-Based Topology Optimisation: Spatial Invariance and Neural Tangent Kernel". *arXiv Prepr arXiv210605710* (2021). arXiv: 2106.05710. URL: http://arxiv.org/abs/2106.05710 5.

[DPB21] DOOSTI, NIKAN, PANETTA, JULIAN, and BABAEI, VAHID. "Topology Optimization via Frequency Tuning of Neural Design Representations". *Symp Comput Fabr*. 2021, 1–9. ISBN: 9781450390903. DOI: 10.1145/3485114.3485124 3.

[DT20] DENG, HAO and TO, ALBERT C. "Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design". *Comput Mech* 66.2 (2020), 449–469. ISSN: 14320924. DOI: 10.1007/s00466-020-01859-5 3.

[FG00] FORTIN, MICHEL and GLOWINSKI, ROLAND. *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*. Elsevier, 2000 4.

[FS20] FERRARI, FEDERICO and SIGMUND, OLE. "A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D". *Structural and Multidisciplinary Optimization* 62.4 (2020), 2211–2228 3, 7.

[GBS06] GERSBORG-HANSEN, A., BENDSØE, M. P., and SIGMUND, O. "Topology optimization of heat conduction problems using the finite volume method". *Struct Multidiscip Optim* 31.4 (2006), 251–259. ISSN: 1615147X. DOI: 10.1007/s00158-005-0584-3 2.

[GD98] GARDNER, MATT W and DORLING, SR. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences". *Atmospheric environment* 32.14-15 (1998), 2627–2636 5.

[HLG19] HU, JINGQIAO, LI, MING, and GAO, SHUMING. "Texture-guided generative structural designs under local control". *CAD Comput Aided Des* 108 (2019), 1–11. ISSN: 00104485. DOI: 10.1016/j.cad.2018.10.002 2, 11.

[HSG19] HOYER, STEPHAN, SOHL-DICKSTEIN, JASCHA, and GREYDANUS, SAM. "Neural reparameterization improves structural optimization". *arXiv Prepr arXiv190904240* (2019). arXiv: 1909.04240. URL: http://arxiv.org/abs/1909.04240 3.

[HX07] HUANG, X. and XIE, Y. M. "Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method". *Finite Elem Anal Des* 43.14 (2007), 1039–1049. ISSN: 0168874X. DOI: 10.1016/j.finel.2007.06.006 2, 3.

[KB15] KINGMA, DIEDERIK P. and BA, JIMMY LEI. "Adam: A method for stochastic optimization". *3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc* (2015). arXiv: 1412.6980 5.

[KW14] KINGMA, DIEDERIK P. and WELLING, MAX. "Auto-encoding variational bayes". *2nd Int Conf Learn Represent ICLR 2014 - Conf Track Proc* (2014). arXiv: 1312.6114 6.

[LC87] LORENSEN, WILLIAM E. and CLINE, HARVEY E. "Marching cubes: A high resolution 3D surface construction algorithm". *Proc 14th Annu Conf Comput Graph Interact Tech SIGGRAPH 1987* 21.4 (1987), 163–169. DOI: 10.1145/37401.37422 10.

[LHZ*18] LIU, HAIXIANG, HU, YUANMING, ZHU, BO, et al. "Narrow-band topology optimization on a sparsely populated grid". *SIGGRAPH Asia 2018 Tech Pap SIGGRAPH Asia 2018* 37.6 (2018), 1–14. ISSN: 15577368. DOI: 10.1145/3272127.3275012 2, 3, 12.

[MGB*21] MEHTA, ISHIT, GHARBI, MICHAËL, BARNES, CONNELLY, et al. "Modulated Periodic Activations for Generalizable Local Functional Representations". *arXiv Prepr arXiv210403960* (2021). arXiv: 2104.03960. URL: http://arxiv.org/abs/2104.03960 2, 6.

[MLL*21] MARTEL, JULIEN N.P., LINDELL, DAVID B., LIN, CONNOR Z., et al. "Acorn: Adaptive coordinate networks for neural scene representation". *ACM Trans Graph* 40.4 (2021). ISSN: 15577368. DOI: 10.1145/3450626.3459785. arXiv: 2105.02788 3.

[MLR*21] MUKHERJEE, SOUGATA, LU, DONGCHENG, RAGHAVAN, BALAJI, et al. "Accelerating Large-scale Topology Optimization: State-of-the-Art and Challenges". *Arch Comput Methods Eng* (2021), 1–23 2.

[MST*20] MILDENHALL, BEN, SRINIVASAN, PRATUL P., TANCIK, MATTHEW, et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*. Vol. 12346 LNCS. Springer. 2020, 405–421. ISBN: 9783030584511. DOI: 10.1007/978-3-030-58452-8_24. arXiv: 2003.08934 5.

[NAC*15] NAUMOV, M., ARSAEV, M., CASTONGUAY, P., et al. "AMGX: A library for GPU accelerated algebraic multigrid and preconditioned iterative methods". *SIAM J Sci Comput* 37.5 (2015), S602–S626. ISSN: 10957200. DOI: 10.1137/140980260 4, 9.

[Ped00] PEDERSEN, N. L. "Maximization of eigenvalues using topology optimization". *Struct Multidiscip Optim* 20.1 (2000), 2–11. ISSN: 1615147X. DOI: 10.1007/s001580050130 2.

[PFS*19] PARK, JEONG JOON, FLORENCE, PETER, STRAUB, JULIAN, et al. "Deepsdf: Learning continuous signed distance functions for shape representation". *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*. Vol. 2019-June. 2019, 165–174. ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00025. arXiv: 1901.05103 6.

[PGC*17] PASZKE, A, GROSS, S, CHINTALA, S, et al. "Automatic differentiation in pytorch". (2017). URL: https://openreview.net/forum?id=BJJsrmfCZ 5.

[Rao17] RAO, SINGIRESU S. *The finite element method in engineering*. Butterworth-heinemann, 2017, 1–763. ISBN: 9780128117682. DOI: 10.1115/1.3167179 4.

[RB93] RIEDMILLER, MARTIN and BRAUN, HEINRICH. "Direct adaptive method for faster backpropagation learning: The RPROP algorithm". *1993 IEEE Int Conf Neural Networks*. IEEE. 1993, 586–591. ISBN: 0780312007. DOI: 10.1109/icnn.1993.298623 5.

[SMB*20] SITZMANN, VINCENT, MARTEL, JULIEN N.P., BERGMAN, ALEXANDER W., et al. "Implicit neural representations with periodic activation functions". *Adv Neural Inf Process Syst* 2020-Decem (2020). ISSN: 10495258. arXiv: 2006.09661 3, 5.

[SP98] SIGMUND, O. and PETERSSON, J. "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima". *Struct Optim* 16.1 (1998), 68–75. ISSN: 09344373. DOI: 10.1007/BF01214002 2, 4.

[Sta07] STANLEY, KENNETH O. "Compositional pattern producing networks: A novel abstraction of development". *Genet Program Evolvable Mach* 8.2 (2007), 131–162. ISSN: 13892576. DOI: 10.1007/s10710-007-9028-8 3.

[TSM*20] TANCIK, MATTHEW, SRINIVASAN, PRATUL P., MILDENHALL, BEN, et al. "Fourier features let networks learn high frequency functions in low dimensional domains". *Adv Neural Inf Process Syst* 2020-Decem (2020). ISSN: 10495258. arXiv: 2006.10739 3.

[UPSW16] UMETANI, NOBUYUKI, PANOTOPOULOU, ATHINA, SCHMIDT, RYAN, and WHITING, EMILY. "Printone: interactive resonance simulation for free-form print-wind instrument design". *ACM Transactions on Graphics (TOG)* 35.6 (2016), 1–14 10.

[VBO*17] VALDEZ, S IVVAN, BOTELLO, SALVADOR, OCHOA, MIGUEL A, et al. "Topology optimization benchmarks in 2D: Results for minimum compliance and minimum volume in planar stress problems". *Archives of Computational Methods in Engineering* 24.4 (2017), 803–839 8.

[WAWS18] WU, JUN, AAGE, NIELS, WESTERMANN, RUDIGER, and SIGMUND, OLE. "Infill Optimization for Additive Manufacturing-Approaching Bone-Like Porous Structures". *IEEE Trans Vis Comput Graph* 24.2 (2018), 1127–1140. ISSN: 10772626. DOI: 10.1109/TVCG.2017.2655523. arXiv: 1608.04366 2.

[WDW16] WU, JUN, DICK, CHRISTIAN, and WESTERMANN, RUDIGER. "A System for High-Resolution Topology Optimization". *IEEE Trans Vis Comput Graph* 22.3 (2016), 1195–1208. ISSN: 10772626. DOI: 10.1109/TVCG.2015.2502588 2, 9.

[WTTL20] WANG, C., TAN, X. P., TOR, S. B., and LIM, C. S. "Machine learning in additive manufacturing: State-of-the-art and perspectives". *Addit Manuf* 36 (2020), 101538. ISSN: 22148604. DOI: 10.1016/j.addma.2020.101538 3.

[WXP*21] WANG, DALEI, XIANG, CHENG, PAN, YUE, et al. "A deep convolutional neural network for topology optimization with perceptible generalization ability". *Eng Optim* (2021). ISSN: 10290273. DOI: 10.1080/0305215X.2021.1902998 3.

[XTS*21] XIE, YIHENG, TAKIKAWA, TOWAKI, SAITO, SHUNSUKE, et al. "Neural Fields in Visual Computing and Beyond". *arXiv Prepr arXiv211111426* (2021). arXiv: 2111.11426. URL: http://arxiv.org/abs/2111.11426 3.

[YX04] YULIN, MEI and XIAOMING, WANG. "A level set method for structural topology optimization and its applications". *Adv Eng Softw* 35.7 (2004), 415–441. ISSN: 09659978. DOI: 10.1016/j.advengsoft.2004.06.004 2.

[ZBLN97] ZHU, CIYOU, BYRD, RICHARD H, LU, PEIHUANG, and NOCEDAL, JORGE. "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization". *ACM Transactions on mathematical software (TOMS)* 23.4 (1997), 550–560 5.

[ZLCT21] ZEHNDER, JONAS, LI, YUE, COROS, STELIAN, and THOMASZEWSKI, BERNHARD. "NTopo: Mesh-free Topology Optimization using Implicit Neural Representations". *arXiv Prepr arXiv210210782* (2021). arXiv: 2102.10782. URL: http://arxiv.org/abs/2102.10782 3.

[ZLY*21] ZHANG, JIAKAI, LIU, XINHANG, YE, XINYI, et al. "Editable free-viewpoint video using a layered neural representation". *ACM Trans Graph* 40.4 (2021), 1–18. ISSN: 15577368. DOI: 10.1145/3450626.3459756. arXiv: 2104.14786 5.

[ZLZ*21] ZHANG, ZEYU, LI, YU, ZHOU, WEIEN, et al. "TONR: An exploration for a novel way combining neural network with topology optimization". *Comput Methods Appl Mech Eng* 386 (2021), 114083. ISSN: 00457825. DOI: 10.1016/j.cma.2021.114083 3.

[ZSCM17] ZHU, BO, SKOURAS, MÉLINA, CHEN, DESAI, and MATUSIK, WOJCIECH. "Two-scale topology optimization with microstructures". *ACM Trans Graph* 36.5 (2017), 1. ISSN: 15577368. DOI: 10.1145/3095815. arXiv: 1706.03189 2, 11.