NANO · MICRO small
www.small-journal.com

Text-to-Microstructure Generation Using Generative Deep Learning

Xiaoyang Zheng, Ikumu Watanabe,* Jamie Paik, Jingjing Li, Xiaofeng Guo, and Masanobu Naito*

Designing novel materials is greatly dependent on understanding the design principles, physical mechanisms, and modeling methods of material microstructures, requiring experienced designers with expertise and several rounds of trial and error. Although recent advances in deep generative networks have enabled the inverse design of material microstructures, most studies involve property-conditional generation and focus on a specific type of structure, resulting in limited generation diversity and poor human–computer interaction. In this study, a pioneering text-to-microstructure deep generative network (Txt2Microstruct-Net) is proposed that enables the generation of 3D material microstructures directly from text prompts without additional optimization procedures. The Txt2Microstruct-Net model is trained on a large microstructure-caption paired dataset that is extensible using the algorithms provided. Moreover, the model is sufficiently flexible to generate different geometric representations, such as voxels and point clouds. The model's performance is also demonstrated in the inverse design of material microstructures and metamaterials. It has promising potential for interactive microstructure design when associated with large language models and could be a user-friendly tool for material design and discovery.

1. Introduction

Designing novel materials with improved properties is highly dependent on understanding of material microstructures and how they affect material properties and performance. Such microstructural design methods have been used to fabricate high-entropy alloys with high strength and ductility,[1,2] functional polymers and composites with enhanced properties and intelligence,[3,4] and high-entropy ceramics with stability and resilience against extreme conditions.[5,6] More importantly, tailoring the microstructure arrangements enables the creation of metamaterials with unprecedented properties,[7–10] such as nanolattices with ultrahigh stiffness-to-density ratios,[11,12] deployable origami structures with multiple degrees of freedom,[13,14] and mechanical metamaterials with extraordinary elasticity tensors.[15–17] Materials with delicately designed microstructures can be chemically synthesized at the nanoscale and mesoscale,[18,19] physically created using microscale-based laser cutting and engraving methods,[13,20], and additively manufactured using a wide range of 3D-printable materials at multiple scales.[21–26]

Deriving inspiration from nature is the most fundamental approach for understanding the intricate 3D microstructures and functions of materials.[27–30] Generally, materials can be designed to mimic the architecture of biological systems,[31–34] mechanisms of animal locomotion,[35,36] and biological responses of such systems to external stimuli.[37–40] Mathematics,

X. Zheng, I. Watanabe
Center for Basic Research on Materials
National Institute for Materials Science
1-2-1 Sengen, Tsukuba 305-0047, Japan
E-mail: xyzheng1995@gmail.com; WATANABE.Ikumu@nims.go.jp

X. Zheng, J. Paik
Reconfigurable Robotics Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne 1015, Switzerland

J. Li
Graduate School of Comprehensive Human Sciences
University of Tsukuba
1-1-1 Tennodai, Tsukuba 305-8573, Japan

X. Guo
School of Materials Science and Engineering
Southwest University of Science and Technology
Mianyang 621010, China

M. Naito
Research Center for Macromolecules and Biomaterials
National Institute for Materials Science
1-2-1 Sengen, Tsukuba 305-0047, Japan

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/smll.202402685

DOI: 10.1002/smll.202402685

architecture, arts, and crafts have also inspired the design of material microstructures.[13,14,25,41–43] These microstructures can be manually modeled using computer-aided design (CAD) modeling tools or automatically generated using computational algorithms such as phase-field modeling,[43] mathematical modeling,[16,33] Voronoi tessellation,[44–46] and topology optimization.[47,48] These traditional design strategies follow a forward-design approach, in which the design space and outcome are dependent on experienced designers using several rounds of trial and error. Moreover, the effective properties of designed microstructures must be investigated using arduous and time-consuming computational simulations or experiments.

Recent advancements in deep learning (DL) have revolutionized material microstructure design processes[49–53]—for example, material microstructures with optimal properties can now be generated by unsupervised learning techniques using generative adversarial networks[54,55] and variational autoencoders (VAEs).[48,56] More importantly, DL frameworks based on supervised learning enable the inverse design of microstructures, where microstructures can be autonomously generated with the desired properties and functions.[45,46,57–69] The inverse design process can eliminate designers' intuition and experience, as well as the need for inefficient trial and error. The training data for inverse design are labeled geometric representations, where geometric representations can be either implicit—such as modeling parameters[59,60,68,70] and numerical representations of geometric elements[57,58,71]—or explicit—such as pixels (images),[46,55,62,63,69] voxels,[45,61,64] point clouds,[72] and meshes. In general, the labels are normalized effective properties of the corresponding geometric representations prepared by computational simulations (e.g., finite element method (FEM) simulations). However, such property-conditional microstructure generation can limit the diversity of microstructures generated and hinder the flexibility of human–computer interactions in the microstructure generation process. Moreover, most property-conditional generation studies—despite their substantial contributions to mapping structure–property relationships—have focused only on a specific type of microstructure.[45,46,57–69]

With recent advances in text-to-3D generative models, it is now possible to generate 3D objects using natural language descriptions. These text-to-3D generation models can be divided into two categories: the first category includes models trained directly on paired data (3D objects and their corresponding text captions),[73–77] and the other category includes models that are combined with pre-trained text-image models to optimize differentiable 3D representations.[78–87] The 3D object generation overhead varies from a number of graphics processing unit (GPU) seconds to multiple GPU hours, depending on the training methods, data formats of 3D representations, and rendering algorithms. Despite the successful implementations of text-conditional models used to generate 3D objects, there has been scant research focused on text-to-microstructure generation.[62,63,88] This can be attributed to the lack of large-scale paired microstructure datasets, lack of a strong 3D prior, and complexity of the DL framework compared with text-to-image generation. Consequently, these studies have suffered from difficulties in scaling diverse and complex text prompts, expensive optimization procedures, and inefficient and non-meaningful microstructure generation. For example, Hsu et al. proposed a DL frame work for the generation of 3D architected materials, which is not direct and requires additional procedure to translate a neural-network-generated image into a continuous 3D architecture.[63]

In this study, we proposed a text-to-microstructure deep generative network (Txt2Microstruct-Net) that could generate 3D material microstructures directly from text prompts without additional optimization procedures. We also created a large, paired microstructure dataset comprising 2,000 diverse 3D microstructures and their corresponding captions. The dataset could be easily expanded using the available modeling algorithms. The diversity of the training dataset and rationality of the DL framework enabled the generation of diverse and realistic microstructures using complex text prompts. Briefly, we created a large number of 3D microstructures covering metals, alloys, polymers, composites, ceramics, architected materials, and metamaterials. Each microstructure was captioned according to its type, geometric features, appearance, modeling method, and effective properties. The Txt2Microstruct-Net model was then trained via a multi-stage training method using the prepared dataset. We then showcased a range of Txt2Microstruct-Net-generated microstructures represented by voxels and cloud points. Finally, we demonstrated the inverse design of microstructures with target properties using the trained Txt2Microstruct-Net model and experimentally investigated the mechanical properties of a Txt2Microstruct-Net-generated mechanical metamaterial.

2. Results and Discussion

2.1. Deep Generative Model

To train the Txt2Microstruct-Net model, we prepared a labeled 3D microstructure dataset in the format $S = \{(\mathbf{V}_n, \mathbf{C}_n, \mathbf{I}_n)\}_{n=1}^{N}$. Each datapoint comprised a microstructure voxel grid $(\mathbf{V}_n)$ with 64 × 64 × 64 voxels, a caption $(\mathbf{C}_n)$ describing the microstructure, and three rendered images $(\mathbf{I}_n)$ of the microstructure in the x-, y-, and z- directions. To achieve a larger design space and wider diversity, we computationally created a range of material microstructures, including four categories—that is, metals and alloys, polymers and composites, ceramics, and architected materials and metamaterials—as shown in Figures S1–S4 (Supporting Information). Each category comprised five classes of material microstructures with different structural features. Moreover, each class comprised a hundred of geometries built using different modeling parameters to achieve diversity in terms of their volume fractions, geometric differences, elemental variability, and effective properties. The modeling methods for these microstructures are detailed in the Supporting Information. To achieve a higher diversity of captions, these 2,000 microstructures were randomly divided into ten subgroups and assigned to ten experts from the fields of mechanical engineering, materials science, computer science, and biology. Several hints—such as category, class, and modeling methods—were provided to help them caption these microstructures. Note that a bias might be introduced in these created captions even though ensuring the diversity. Figure S7 (Supporting Information) shows the vocabulary used in these captions. Each microstructure was rendered from the front, top, and left views (Figures S1–S4, Supporting Information). The rendering was

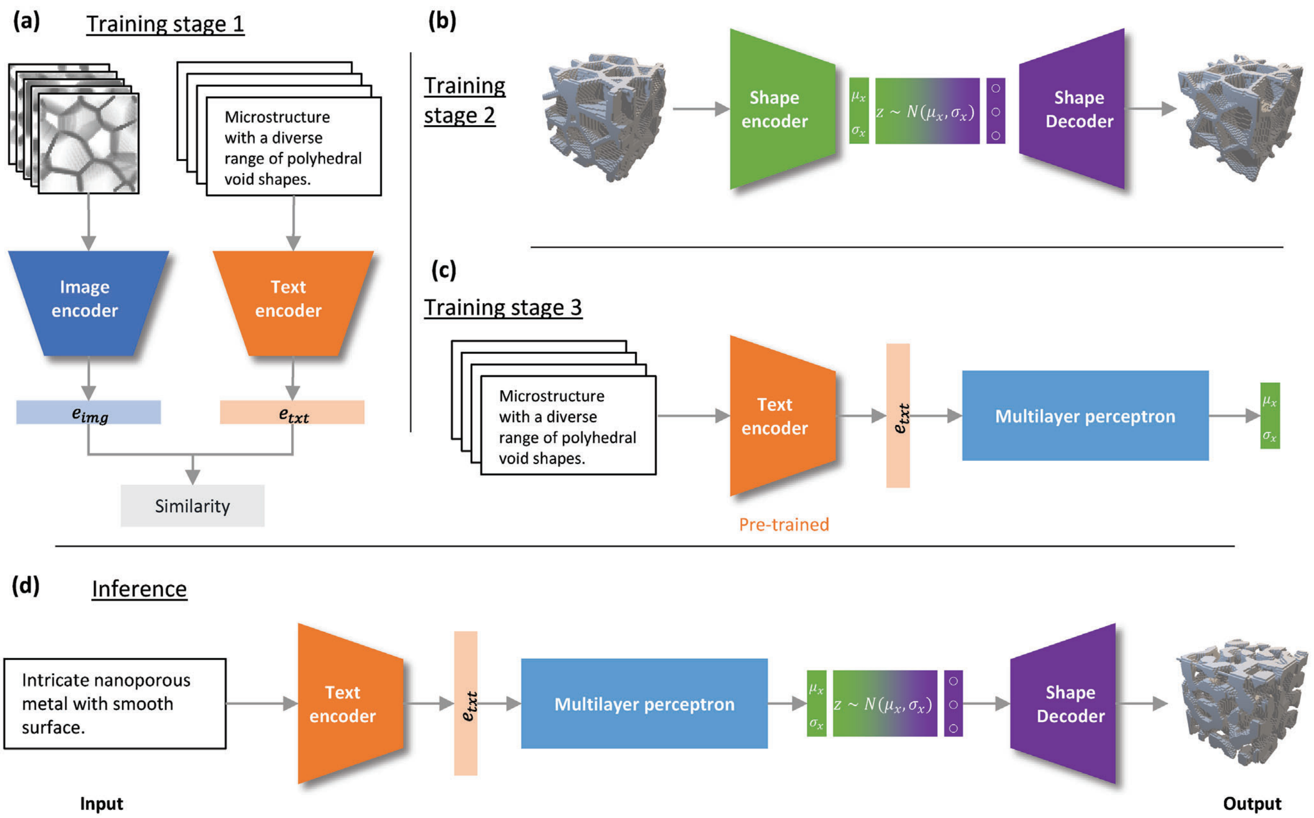
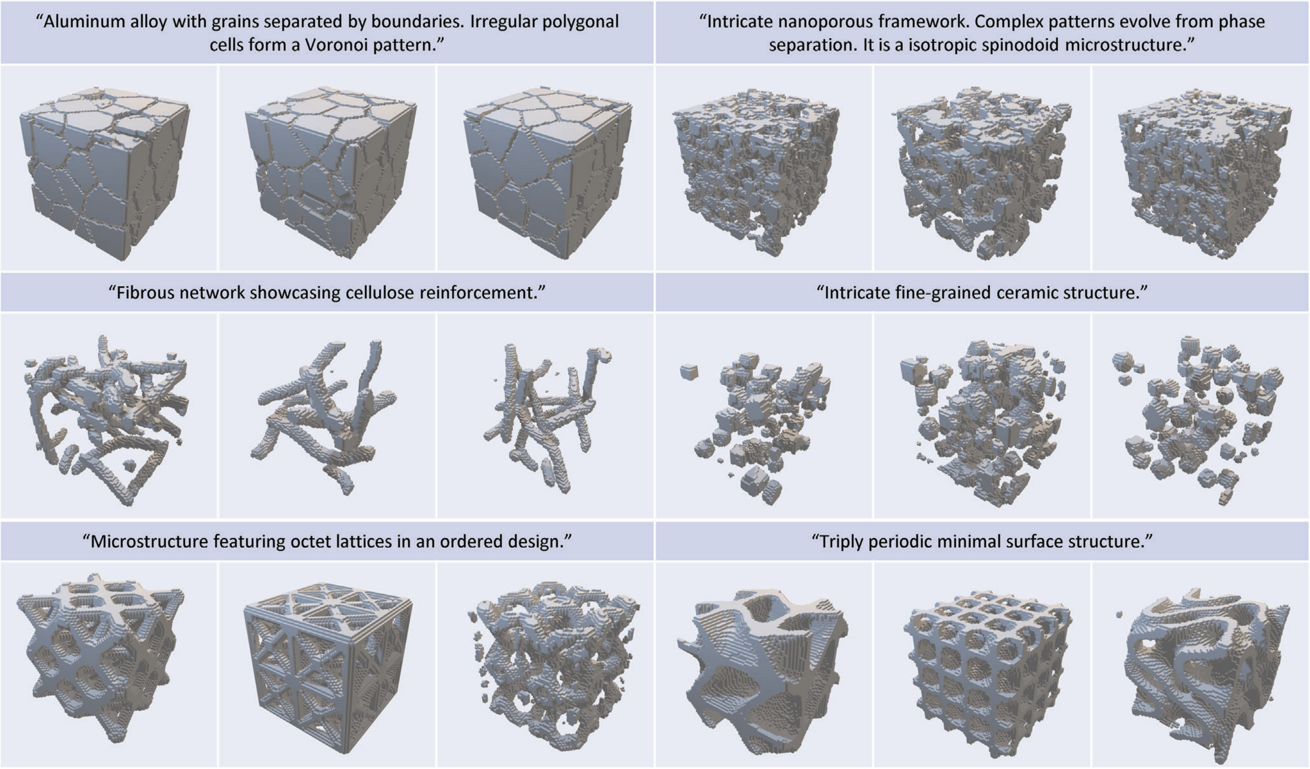


Figure 1. Txt2Microstruct-Net framework. a) Training stage 1: a dual encoder is trained to project the representation of images and their captions into the same embedding space. b) Training stage 2: a VAE is trained to embed the microstructures into a latent space in the form of Gaussian distribution. c) Training state 3: an MLP is trained to predict the shape embeddings of text embeddings generated by the trained text encoder. d) Inference phase: a text prompt is embedded by the trained text encoder, before being converted into the shape embedding; the shape embedding is decoded into 3D microstructure voxels by the trained shape decoder.

performed by overlaying the cross-sections of each view with an alpha channel for transparency. Consequently, the training dataset composed $N = 2,000$ datapoints and could be easily expanded using the modeling algorithms provided.

The Txt2Microstruct-Net model was trained using a multi-stage training approach, as shown in **Figure** 1a–c. The multi-stage training approach was used for resource efficiency and ensemble learning: neural networks in the first and second stages were trained independently, and then their training results were combined in the third stage to train the other neural network. This approach can help mitigate the risk of overfitting, save computational resources, and improve generalization. In the first stage, we trained a dual encoder neural network to build an interchangeable text-image latent space, inspired by the contrastive language–image pre-training (CLIP) approach.[89] Image and text encoders were trained simultaneously to project the representation of images ($\mathbf{I}_n$) and their captions ($\mathbf{C}_n$) into the same embedding space. This helped generate similar microstructures with similar but different natural language descriptions, because the caption embeddings were located near the embeddings of the images they described. We used a pre-trained Xception model as the base for the image encoder and a pre-trained BERT model as the base for the text encoder.[90,91] The pre-trained Xception and BERT models were fine-tuned in this first stage. In the second stage, we trained a VAE comprising a shape encoder and decoder. The shape encoder was trained to extract the shape embedding (e_n) with a Gaussian distribution ($z \approx \mathcal{N}(\mu_x, \sigma_x)$) for the microstructure collection ($\mathbf{V}_n$). The shape decoder learned to decompress the microstructure representation from the shape embedding. We used a voxel-based VAE as the base for the shape encoder and decoder.[92] In the third stage, we trained a multilayer perceptron (MLP) to generate a shape embedding (e_n) conditioned on caption embeddings from the text encoder. The MLP had four fully connected layers as its hidden layer, and output μ_x and σ_x using its output layer. Details of the training process are provided in the Supporting Information.

After the Txt2Microstruct-Net model had been well trained, it was used to generate a number of microstructures conditioned on the text prompt (Figure 1d). In the inference phase, a text prompt could be converted into a text embedding (e_{txt}) using the text encoder. The text embedding could then be used as the condition vector to generate the shape embeddings (e_n) using the MLP. As the text and image encoders were trained to bring the text and image embeddings into a joint latent space, the shape embeddings in the near latent space could be generated using similar text prompts. The shape embeddings could then be converted into 3D microstructures in the form of voxels using the shape decoder. The generated voxels could be further post-processed to


"Aluminum alloy with grains separated by boundaries. Irregular polygonal cells form a Voronoi pattern."

"Intricate nanoporous framework. Complex patterns evolve from phase separation. It is a isotropic spinodoid microstructure."

"Fibrous network showcasing cellulose reinforcement."

"Intricate fine-grained ceramic structure."

"Microstructure featuring octet lattices in an ordered design."

"Triply periodic minimal surface structure."

Figure 2. Voxel-based microstructures generated by the Txt2Microstruct-Net model using different text prompts.

clean isolated noisy elements, before being converted into a mesh and geometrically optimized for simulation or fabrication.

2.2. Microstructure Generation Using Text Prompt

To qualitatively evaluate the generative capabilities of the trained Txt2Microstruct-Net model, we show several voxel-based microstructures generated by the Txt2Microstruct-Net model using different text prompts in **Figure 2**. It shows that multiple diverse microstructures could be generated using a single text prompt, which helped to generate new variations in the material design process. Moreover, it is evident that the microstructures based on different categories and classes could be generated using common semantic words, geometric attributes, and technical terms, proving that the Txt2Microstruct-Net model could capture the semantic notions of text prompts. Additional generated cases are shown in Figure S8 (Supporting Information) to demonstrate the reliability of the training results. It is evident that the generative quality of these microstructures could be qualitatively compared with the ground truth (Figures S1–S4, Supporting Information), and they exhibited good similarity in terms of appearance. However, It should be noted that a few isolated voxel clusters could be found in some generated microstructures, which could be attributed to the transposed convolution layers in the shape decoder. These isolated voxel clusters could be removed by filtering the threshold value of the isolated voxel numbers during post-processing. Moreover, it should be noted that the performance of the Txt2Microstruct-Net is not good at zero-shot generation. This can be attributed to the weakness of the language model (i.e., BERT model) in the material domain. In fact, BERT model can hardly capture the relationship between the textual features of qualitative descriptions and the geometrical features of 3D structures because BERT model was pre-trained using text from Wikipedia that lacks the specificity of the material domain.

To quantitatively evaluate the performance of the Txt2Microstruct-Net model, the intersection over union (IoU), class accuracy, and Fréchet inception distance (FID) were used to measure the reconstruction quality, generation accuracy, and generation quality, respectively. The IoU calculates the overlap between the ground-truth microstructure voxel ($\mathbf{V}_n$) and the reconstructed microstructure voxel ($\mathbf{V}_n^{re}$) in the reconstruction process of the VAE (Figure 1b), written as IoU = $|\mathbf{V}_n \cap \mathbf{V}_n^{re}|/|\mathbf{V}_n| \cup |\mathbf{V}_n^{re}|$. The IoU was calculated using the 2,000 voxelized microstructures using 64 × 64 × 64 voxels from the dataset. An IoU score of 0.9618 indicated a perfect match between $\mathbf{V}_n$ and $\mathbf{V}_n^{re}$, demonstrating that the microstructures could be perfectly reconstructed. To ensure that the Txt2Microstruct-Net model generated microstructures across classes, we trained a classifier to predict the classes of the given microstructures. The classifier was trained on 2,000 collected microstructures with an assigned label corresponding to one of the 20 classes. The classifier shared a similar architecture with the shape encoder but with 20 dimensions to its output layer. We then created 200 new text prompts (ten prompts for each class) that were conditioned to generate microstructures using the trained Txt2Microstruct-Net model. The class accuracy of the generated microstructures predicted by the classifier was used

Table 1. Comparison of the Txt2Microstruct-Net model with other DL models for voxel-based microstructure generation.

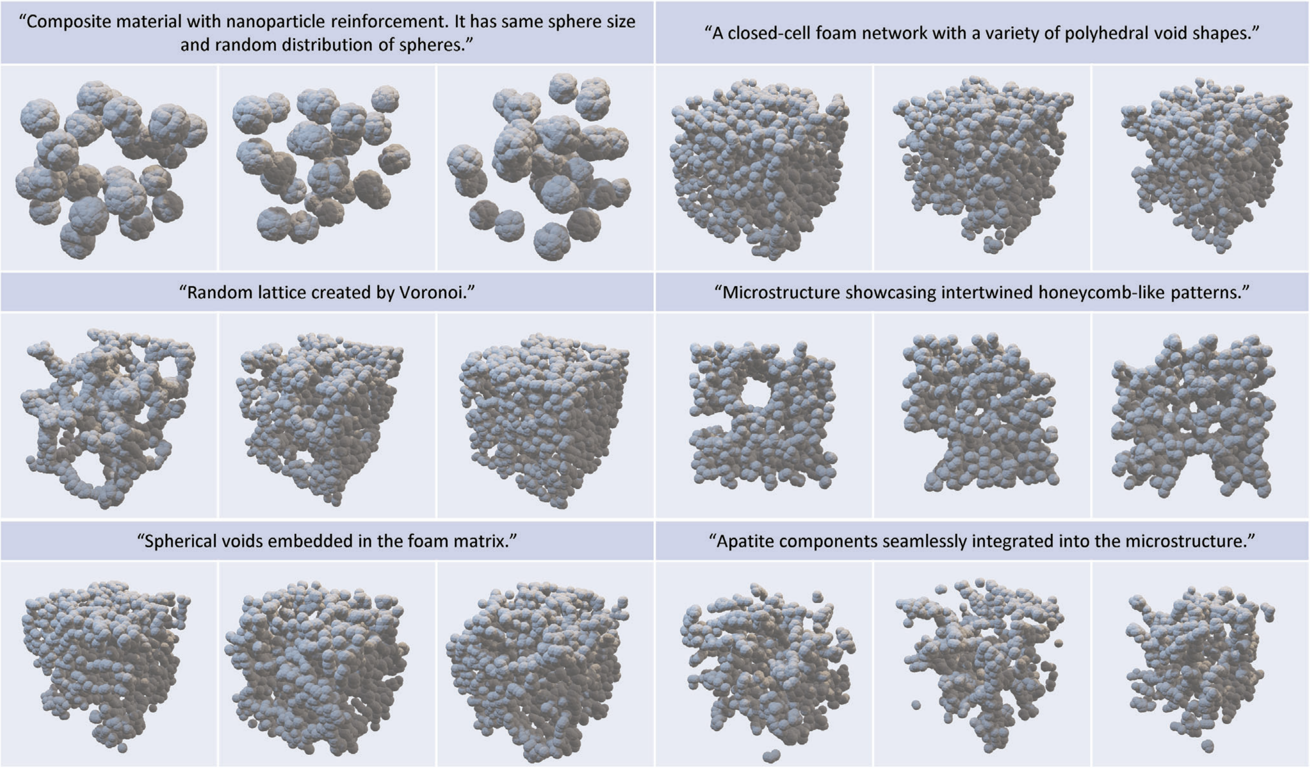
	Metrics				Other information					
DL framework	IoU	FID	Class accuracy	Mean square error	Input	Output	Microstructure type	Datapoints	Training time	GPU
SliceGAN[61]	–	–	–	–	Microstructure images	64 × 64 × 64 × 1 or 64 × 64 × 64 × 3 voxels	A variety of microstructures, such as polycrystalline grains, ceramic, carbon fiber rods, grain boundary, and battery separator	192	4 h	NVIDIA Titan Xp GPU
3D-CGAN[45]	–	53.91	–	0.01	Material property vectors	64 × 64 × 64 voxels	Voronoi-based open-cell foams	10000	32 h	NVIDIA RTX A6000 graphics card
Txt2Microstruct-Net	0.9618	72.08	0.8695	–	Text prompt	64 × 64 × 64 voxels	A diversity of microstructures, ranging from metals and ceramics to composites and architected materials	2000	49 min	NVIDIA RTX A6000 graphics card

The SliceGAN model takes images as inputs and outputs 3D microstructures;[61] the 3D-CGAN model takes material property vectors as inputs and outputs 3D microstructures;[45] and the Txt2Microstruct-Net model takes text prompts as inputs and outputs 3D microstructures. The 3D-CGAN model has a lower FID compared with the Txt2Microstruct-Net model, indicating a higher generation quality. This is because the 3D-CGAN model was trained on a single type of Voronoi-based open-cell foams. Note that the comparison lacks strictness because of the difference in input types and the scale of the training dataset.

as a metric to measure the generation accuracy. The resulting class accuracy of 0.8695 indicated that microstructures could be intentionally generated with the corresponding class. Finally, to evaluate the generation quality, we used the embedding taken from the penultimate layer of the pre-trained classifier to calculate the FID of input microstructures (2,000 real and generated microstructures), written as FID = $\|\mu_x - \mu_y\|^2 - Tr(C_x + C_y - 2(C_x C_y)^{1/2})$, where x and y denotes the feature vectors of the real and generated embeddings, respectively, μ_x and μ_y denote the magnitudes of the vectors, Tr denotes the trace of the matrix, and C_x and C_y denote the covariance matrix of the vectors. The FID score can be used to evaluate the quality of images generated by deep generative networks, where lower scores correspond to higher-quality images.[93] The resulting low FID score of 72 indicated a higher quality of microstructures generated by the Txt2Microstruct-Net model. We also compared these metrics with those of other studies in terms of voxel-based microstructure generation, as shown in **Table 1**.

To investigate the flexibility of the Txt2Microstruct-Net model for other geometric representations, we replaced the voxel-based VAE with a point-cloud-based VAE for the point cloud generation. We retrained the Txt2Microstruct-Net model using the same procedure, but in training stage 2, we used another VAE network—that is, VG-VAE—for point-cloud-based microstructure reconstruction.[94] The training data for the VG-VAE were point cloud representations of 2,000 collected microstructures. Each point cloud was created by randomly sampling 2048 volumetric points from a mesh in the collected microstructures. The coordinates of these 2,048 volumetric points were normalized in a domain [− 0.5, 0.5] for all the point clouds for the VG-VAE training. **Figure 3** shows several point-cloud-based microstructures generated by the Txt2Microstruct-Net model using VG-VAE as the shape encoder and decoder. This shows that the Txt2Microstruct-Net model could generate microstructures in the point-cloud representations well, although isolated clusters appeared in some generations. However, compared to the voxel representations, the point-cloud representations could hardly preserve the delicate geometric features in complex microstructures, exhibiting variations in microstructures with high volume fractions (Figure 3). An alternative solution was to use mesh representations; however, the data sizes of these complex material microstructures varied individually, making it difficult to normalize them for DL compared to other 3D object datasets.[95]

A potential application of the Txt2Microstruct-Net model is the fabrication of generated microstructures conditioned using specific text prompts. **Figure 4** shows the process used to fabricate microstructures with different geometric representations using additive manufacturing. For the voxel representations, the isolated voxel clusters were cleared before smoothing the surface of the generated voxel representations using the non-uniform rational mesh smooth (NURMS) method (Figure 4a,b). The smoothed voxel representation was then converted into a mesh representation for either additive manufacturing or FEM simulations. Representations with separate elements could also be prepared in multiple phases using multimaterial 3D printers. For the point-cloud representations, a mesh was created by generating spheres using 2,048 volumetric points as center points. Depending on the volume fraction, geometric complexity, and geometric type, the


“Composite material with nanoparticle reinforcement. It has same sphere size and random distribution of spheres.”

“A closed-cell foam network with a variety of polyhedral void shapes.”

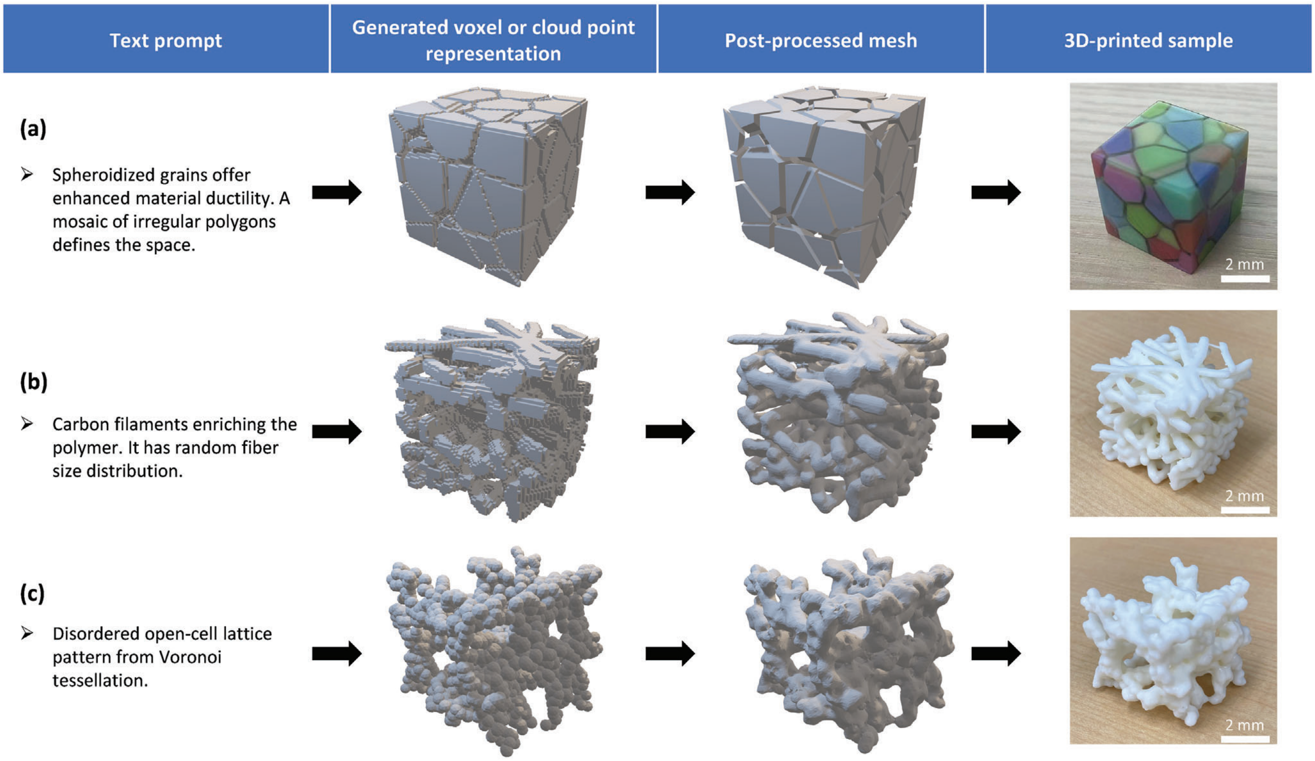
“Random lattice created by Voronoi.”

“Microstructure showcasing intertwined honeycomb-like patterns.”

“Spherical voids embedded in the foam matrix.”

“Apatite components seamlessly integrated into the microstructure.”

Figure 3. Point cloud-based microstructures generated by the Txt2Microstruct-Net model using different text prompts.



➢ Spheroidized grains offer enhanced material ductility. A mosaic of irregular polygons defines the space.

➢ Carbon filaments enriching the polymer. It has random fiber size distribution.

➢ Disordered open-cell lattice pattern from Voronoi tessellation.

Figure 4. Fabricating the Txt2Microstruct-Net-generated microstructures using additive manufacturing. a) Generating and fabricating a voxel-based microstructure with multimaterials. b) Generating and fabricating a voxel-based microstructure with a single material. c) Generating and fabricating a point-cloud-based microstructure.

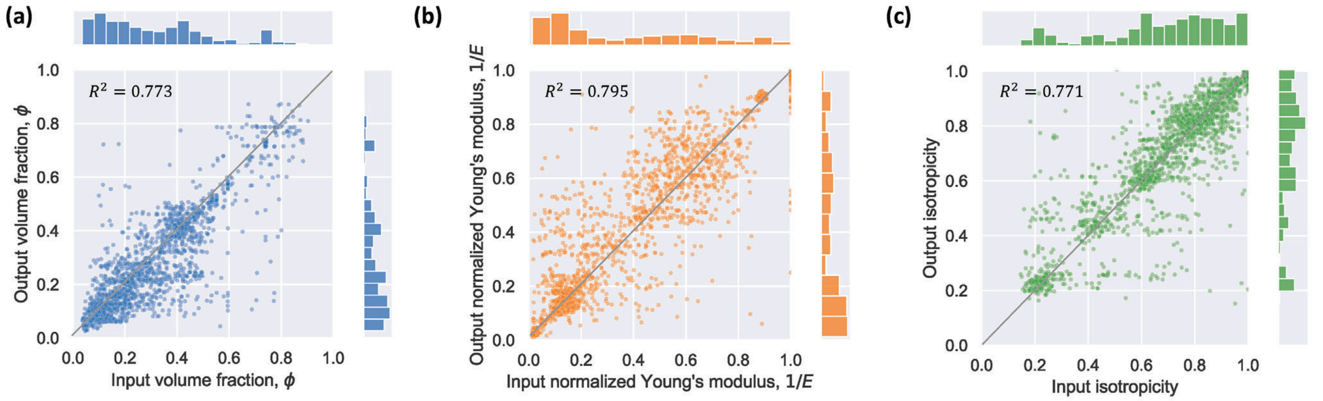


Figure 5. Inverse design of microstructures. Comparison between user-input and Txt2Microstruct-Net-output volume fraction (a), Young's modulus (b), and isotropicity (c).

radii of these spheres could be tuned to generate a suitable mesh, which could be further smoothed using the NURMS method for additive manufacturing (Figure 4c). Other post-processing methods—such as mesh editing using CAD tools and topology optimization—can also be used to modify the geometry of the generated microstructures.

2.3. Inverse Design of Microstructures

Another promising application of the Txt2Microstruct-Net model is the inverse design of material microstructures. Unlike qualitative generation using text prompts, inverse design enables the quantitative generation of microstructures to satisfy the input target properties. To this end, we calculated additional labels for these microstructures—that is, their effective properties—including their volume fraction, effective Young's modulus, and isotropicity. The effective Young's modulus was calculated by isotropically approximating the stiffness tensor, and the isotropicity was calculated as: isotropicity = min $(E_{\theta,\phi})/\max(E_{\theta,\phi})$, where $E_{\theta,\phi}$ denotes the direction-dependent Young's modulus obtained by solving the stiffness tensor.[96] The stiffness tensors were calculated using a numerical homogenization method.[45,97] These additional labels were appended to the microstructure captions during training. Prior to the inverse design, we trained a solver to predict the effective properties of the inversely designed microstructures (Figure S6, Supporting Information). We used the above-mentioned method to train the Txt2Microstruct-Net model using property-appended captions. **Figure** 5a–c compare the input labels (i.e., the target volume fraction, Young's modulus, and isotropicity) and output labels (i.e., the volume fraction, Young's modulus, and isotropicity of the generated microstructures) of 2,000 randomly inversely designed microstructures. Here, each coordinate of the scatter plot corresponded to an input and output label. The difference between the input and output labels could then be evaluated by linearly fitting these scatters with a bisection line, and a position closer to the bisection line represents better inverse-design performance. Although the input and output labels were similar in terms of property distribution, they were discrete to the bisection line, indicating poor inverse-design performance compared to other DL-based inverse-design frameworks.[45,46,57,59,65,66,68–72] This could be attributed to the differences in the training methods—for example, the Txt2Microstruct-Net model was trained on word embeddings, whereas the others were generally trained on the embeddings of numeric property vectors. Compared with vector embeddings, word embeddings obtained by large language models can have difficulty capturing precise target properties, resulting in poor inverse-design performance. However, word embeddings enable the processing of more information (e.g., microstructure types, geometric features, and semantic words). Consequently, the Txt2Microstruct-Net model enables the inverse design of diverse microstructures, whereas the other DL-based inverse-design frameworks can only generate a single type of microstructures generally.

To demonstrate the design space for mechanical metamaterials, we generated pairs of auxetic metamaterials using a text prompt, as shown in **Figure** 6a. An auxetic metamaterial is a mechanical metamaterial with a negative Poisson's ratio. Typically, it shrinks dimensionally upon uniaxial compressive loading, which is highly dependent on its microstructure. Notably, there were some isolated voxels in these Txt2Microstruct-Net-generated microstructures, which could result in geometric imperfections. An generated auxetic metamaterial was post-processed and compared with a reference auxetic metamaterial from the dataset, as shown in Figure 6b. Like the reference auxetic metamaterial with a perfect cubic symmetry, the generated metamaterial exhibited good cubic symmetry, although a negligible imperfection was evident in the direction-dependent Young's modulus curves (Figure 6c). The two auxetic metamaterials with 3 × 3 × 3 unit cells were 3D printed using a rubber-like resin, and they exhibited good agreement in terms of appearance (Figure 6d).

To investigate the impact of geometric imperfections on the mechanical properties of the generated and reference auxetic metamaterials, we conducted a systematic analysis using FEM simulations and uniaxial compression tests. **Figure** 7a shows a sequence of progressively deformed shapes of the generated auxetic metamaterials under four different levels of compressive engineering strain obtained from the FEM results. Compared to the reference auxetic metamaterial that contracted in all directions, the generated metamaterial exhibited no apparent shrinkage along the y-axis, indicating a partially non-negative Poisson's

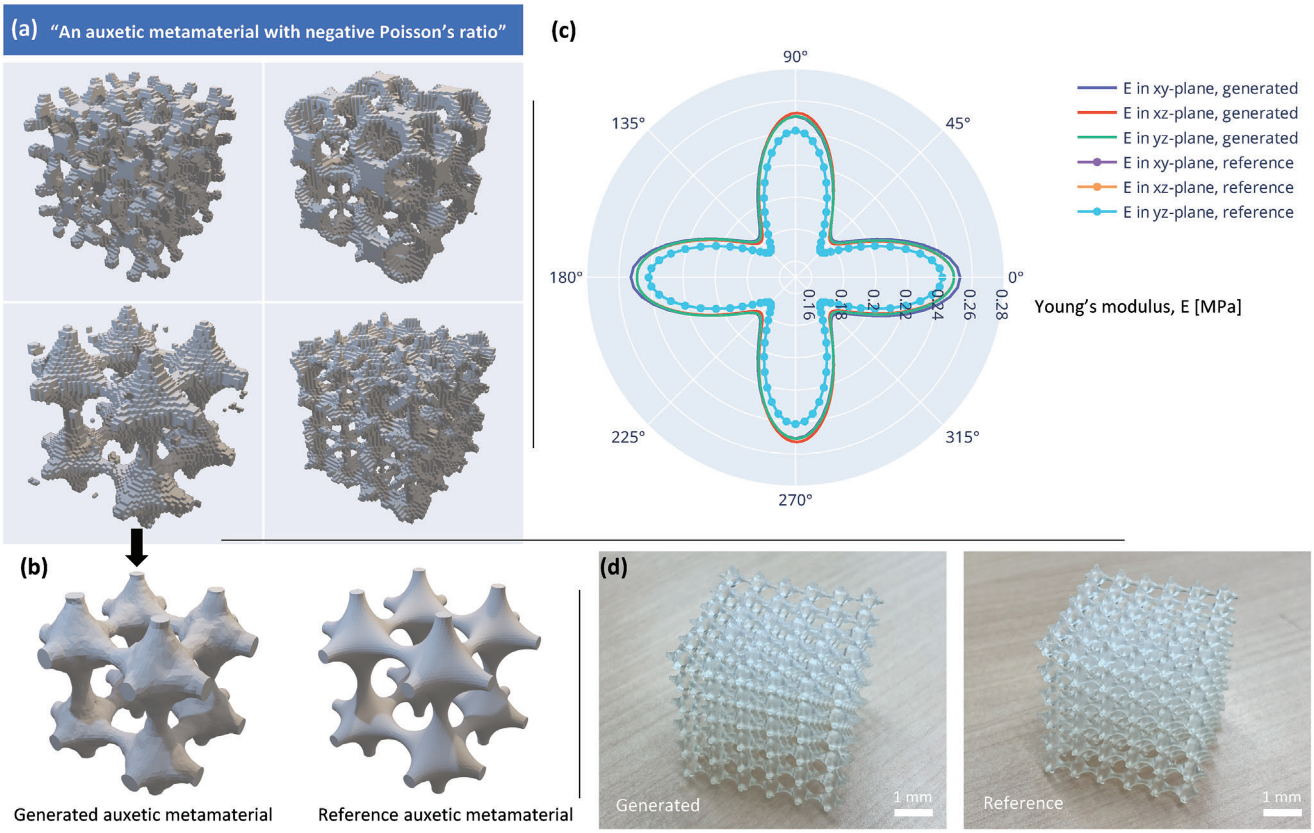


Figure 6. Inverse design of auxetic metamaterials. a) Four Txt2Microstruct-Net-generated auxetic metamaterials generated using the same text prompt. b) Renderings of a generated and reference auxetic metamaterial. b) Direction-dependent Young's moduli of the two auxetic metamaterials. d) 3D-printed samples of the two auxetic metamaterials.

ratio. Similar results were evident in the experiments, where the 3D-printed generated sample exhibited non-auxetic behavior along the y-axis. Figure 7b shows a detailed comparison between the Poisson's ratio–strain curves obtained from the simulations and experiments. It shows that the Poisson's ratio decreases with compressive strain, except for the ν_{zy} of the generated auxetic metamaterial. The partial non-auxeticity can be attributed to the geometric imperfection of the generated auxetic metamaterial, whose auxetic behavior results from buckling instability.[16] Geometric imperfections affect not only the auxetic behavior but also the stress–strain curves, as shown in 7c, where fluctuations appear on the stress–strain curve of the 3D-printed generated sample. The results indicate that voxel-based generation conditioned on text prompts exhibits a weakness for the metamaterial design, where precise control over the microstructure is required. By contrast, DL frameworks that output implicit geometric representations perform well in generating delicately designed metamaterials.[57,59,66,70–72]

3. Conclusion

We developed a generative DL-based framework, Txt2Microstruct-Net, for text-to-microstructure generation. The Txt2Microstruct-Net model was trained using 2,000 microstructures with captions describing their categories, classes, modeling methods, geometric features, and appearances. This model can rapidly generate multiple diverse voxel-based microstructures without an optimization process using text prompts as inputs. Further, we also showcased the flexibility of point-cloud-based microstructure generation and the inverse design of microstructures using property-appended labels. However, the Txt2Microstruct-Net model exhibited a weakness in metamaterial design, where delicate microstructure design was required. It shows that the text description of microstructures can hardly capture the geometric and physical information of a microstructures. A universal method to describe a microstructure is expected for addressing this issue. The geometric imperfections resulting from voxel representation and word embeddings hindered precise control over the microstructures during the generation process. This is expected to be improved using implicit geometric representations that can precisely control the type, arrangement, and interface of microstructural elements using additional modeling algorithms.

This study demonstrated a pioneering and alternative way to design and generate material microstructures using text prompts, which was more user-friendly for non-specialists compared with other DL frameworks whose inputs were property vectors. The proposed framework can be optimized and extended to the inverse design of more microstructures by extending the training dataset (e.g., including more additional details of microstructures in text descriptions), replacing geometric representations, optimizing neural networks (e.g., using generative

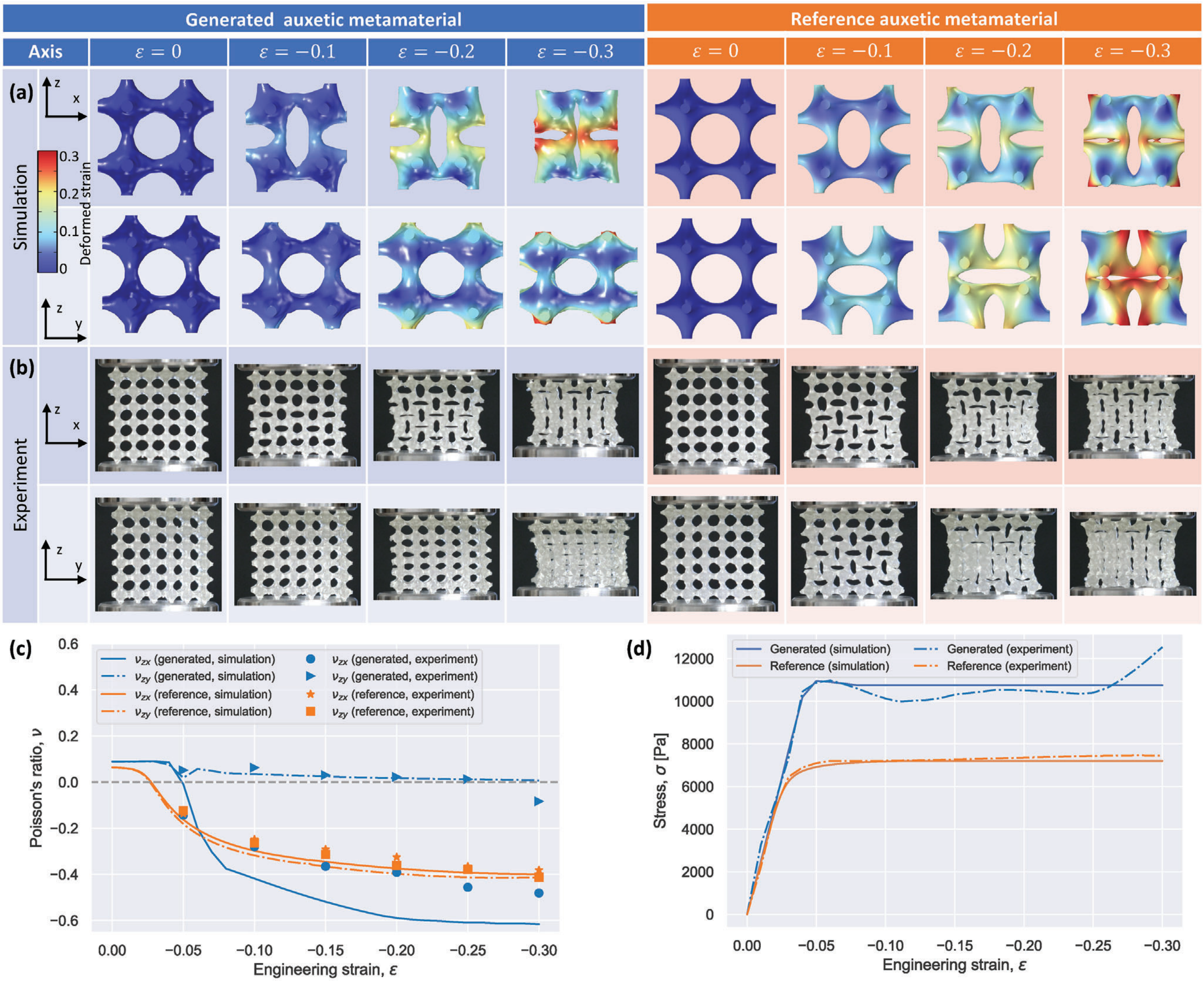


Figure 7. Mechanical property comparison between a generated and reference auxetic metamaterial. Progressively deformed shapes at different compressive strains from FEM simulations (a) and uniaxial compression tests (b). Poisson's ratio–strain curves (c) and stress–strain curves (d) from FEM simulations and uniaxial compression tests.

diffusion models), and using fine-tuning process (e.g., using supervised learning and reinforcement learning from human feedback). Taking advantage of text mining and information extraction using large language models, the Txt2Microstruct-Net model is a promising tool for material discovery and informatics.[98,99]

4. Experimental Section

Dataset Preparation: The methods used to create these microstructures are detailed in the Supporting Information. After generating mesh files (.stl format) of these microstructures, they were converted into 3D voxel arrays ($\mathbf{V}_\mathbf{n}$) comprising 64 × 64 × 64 voxels using the Python library VoxelFuse.[100] The 3D voxel arrays of all microstructures were stored as a numpy array (npy format) using a Boolean data type referring to different phases. The captions of these microstructures ($\mathbf{C}_\mathbf{n}$) were stored in a text file (.csv format). Images of each microstructure ($\mathbf{I}_\mathbf{n}$) were rendered from the front, top, and left views using the voxelized geometry. An increasing value of the alpha channel was used to overlay the sliced layers, and the images were stored as transparent grayscale images (.png format) of 299 × 299 pixels. Transparency ensured the preservation of interior microstructural features. The point clouds of these microstructures ($\mathbf{P}_\mathbf{n}$) were created by randomly sampling 2,048 volumetric points from their mesh files using the Python library Trimesh.[101] Consequently, a $\{(\mathbf{V}_\mathbf{n}, \mathbf{C}_\mathbf{n}, \mathbf{I}_\mathbf{n})\}_{n=1}^{N}$ dataset was obtained for voxel-based training, and a $\{(\mathbf{P}_\mathbf{n}, \mathbf{C}_\mathbf{n}, \mathbf{I}_\mathbf{n})\}_{n=1}^{N}$ dataset was obtained for point cloud-based training.

Implementation of Deep Generative Model: The training process for the Txt2Microstruct-Net model was conducted using TensorFlow (version 2.12.0) on a single NVIDIA RTX A6000 graphics card (48 GB GPU Memory) running on a Linux system. The Linux environment was developed using Python 3.10 and CUDA 11.8. The neural networks, training process, and training results are detailed in the Supporting Information, and the codes are available at https://github.com/xyzheng-ut/Txt2Microstruct-Net.

Additive Manufacturing: All models used for additive manufacturing were post-processed by removing isolated noisy voxels and smoothing their surfaces using the NURMS method. Models for different purposes were fabricated using various 3D printers and materials. The 3D-printed colored sample (Figure 4a) was fabricated using a multimaterial 3D printer (PolyJet J850 Prime 3D Printer, Stratasys, USA) with vivid translucent colored materials (VeroVivid Color Family, Stratasys, USA). The 3D-printed white samples (Figure 4b,c) were fabricated using a stereolithography (SLA) 3D printer (Form 3, Formlabs, USA) with a photopolymer resin (White resin, Formlabs, USA). The 3D-printed translucent samples (Figure 6d) were fabricated using an SLA 3D printer with a rubber-like photopolymer resin (Elastic resin, Formlabs, USA). Specifically, 3D-printed auxetic metamaterials were modified using $3 \times 3 \times 3$ unit cells of $40 \times 40 \times 40$ mm^3 dimensions, which represented the geometric features of periodic porous materials for mechanical testing.[33] Moreover, a layer thickness of 0.05 mm and operating temperature 33°C were used in the printing process. The samples were washed using isopropanol after 3D printing, followed by curing at 60°C for 15 min using Form Cure (Formlabs, USA).

Mechanical Testing: The mechanical properties of the 3D-printed auxetic metamaterials were examined via uniaxial compression tests using a motorized test stand (AGXplus-10kN, Shimadzu, Japan). Static compression tests were conducted at a vertically constant speed of 10 mmmin^{-1}, based on the ASTM D695–15 standard. The stop condition was set to a compressive strain of 0.3 to avoid full contact of the structural elements of these auxetic metamaterials. The deformation processes were recorded using two high-speed cameras (front and side views). The stress–strain curves were obtained using the recorded load and displacement data. The Poisson's ratios were evaluated by extracting the displacements of the nodes of the deformed geometries from the recorded videos via postprocessing in MATLAB (Version R2021b, MathWorks, USA).

Finite Element Method Simulation: The mechanical properties of the generated and reference auxetic metamaterials were visually examined using FEM simulations. A nonlinear FEM simulation (COMSOL Multiphysics version 6.1, COMSOL, Sweden) was performed to examine the large-deformation behavior of these auxetic metamaterials. An incompressible neo-Hookean material model was assigned to the model using an experimentally measured Young's modulus of 0.6615 MPa.[16] As the auxetic metamaterials can suffer from buckling instability, a linearized buckling analysis was first performed to compute the shape of the first-order buckling mode. A post-buckling analysis was performed on the the buckled geometry using a parametric sweep of the z-axis displacement and a stop condition when adjacent boundaries were in contact. The Poisson's ratios and stress–strain curves were calculated via the post-buckling analysis. The models were meshed using approximately 3×10^5 second-order tetrahedral solid elements. Periodic boundary conditions were implemented using the representative volume element method.[102,103]

Numerical Homogenization: The stiffness tensors of the microstructures used for inverse design were calculated using a numerical homogenization method. The homogenized 6×6 stiffness tensor was obtained by calculating the element displacements and global displacement field in the cube domain using iterations for six load cases (i.e., three compressions along the x, y, and z axes, and three shearing loads), as detailed in previous studies.[45,97] Homogenization was implemented using MATLAB with a $64 \times 64 \times 64$ voxel array consisting of 0s and 1s, where 0 and 1 represent different phases. For simplicity, two linear elastic models with different parameters for the material microstructures with two solid phases were used. The Young's modulus and Poisson's ratio for the two phases were set to 100 GPa and 0.3, and 1 GPa and 0.3, respectively. For material microstructures with a solid and void phases, a linear elastic model was used for the solid phase with a Young's modulus and Poisson's ratio of 100 GPa and 0.3, respectively. The volume fraction referred to the proportion of the solid (stiffer) phase and was calculated as $\sum \mathbf{V}_\mathbf{n}/64^3$.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This research was supported by a Grant-in-Aid for JSPS Fellows PD (Grant Number 22KJ0407) and Young Researchers' Exchange Programme—Special 2023 Call Japan of Japanese-Swiss Science and Technology Programme (Project no. JP_EG_special_032023_11). The authors acknowledged the support from the university club, Research Support, University of Tsukuba (RSUT) (https://sites.google.com/view/rs-tsukuba).

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Keywords

architected material, artificial intelligence, deep generative model, deep learning, inverse design, metamaterial, microstructure

Received: April 8, 2024
Published online: May 21, 2024

[1] S. Chen, Z. H. Aitken, S. Pattamatta, Z. Wu, Z. G. Yu, D. J. Srolovitz, P. K. Liaw, Y.-W. Zhang, *Nat. Commun.* **2021**, *12*, 4953.
[2] E. P. George, D. Raabe, R. O. Ritchie, *Nat. Rev. Mater.* **2019**, *4*, 515.
[3] Y. Xia, Y. He, F. Zhang, Y. Liu, J. Leng, *Adv. Mater.* **2021**, *33*, 2000713.
[4] F. D. Jochum, P. Theato, *Chem. Soc. Rev.* **2013**, *42*, 7468.
[5] S. Divilov, H. Eckert, D. Hicks, C. Oses, C. Toher, R. Friedrich, M. Esters, M. J. Mehl, A. C. Zettel, Y. Lederer, E. Zurek, J.-P. Maria, D. W. Brenner, X. Campilongo, S. Filipovic, W. G. Fahrenholtz, C. J. Ryan, C. M. DeSalle, R. J. Crealese, D. E. Wolfe, A. Calzolari, S. Curtarolo, *Nature* **2024**, *625*, 66.
[6] C. Oses, C. Toher, S. Curtarolo, *Nat. Rev. Mater.* **2020**, *5*, 295.
[7] P. Jiao, J. Mueller, J. R. Raney, X. Zheng, A. H. Alavi, *Nat. Commun.* **2023**, *14*, 6004.
[8] C. M. Soukoulis, M. Wegener, *Nat. Photonics* **2011**, *5*, 523.
[9] Y. Li, W. Li, T. Han, X. Zheng, J. Li, B. Li, S. Fan, C.-W. Qiu, *Nat. Rev. Mater.* **2021**, *6*, 488.
[10] K. Fan, W. J. Padilla, *Mater. Today* **2015**, *18*, 39.
[11] X. Zheng, H. Lee, T. H. Weisgraber, M. Shusteff, J. DeOtte, E. B. Duoss, J. D. Kuntz, M. M. Biener, Q. Ge, J. A. Jackson, S. O. Kucheyev, N. X. FANG, C. M. Spadaccini, *Science* **2014**, *344*, 1373.
[12] J. Bauer, L. R. Meza, T. A. Schaedler, R. Schwaiger, X. Zheng, L. Valdevit, *Adv. Mater.* **2017**, *29*, 1701850.
[13] J. L. Silverberg, A. A. Evans, L. McLeod, R. C. Hayward, T. Hull, C. D. Santangelo, I. Cohen, *Science* **2014**, *345*, 647.
[14] J. T. Overvelde, T. A. De Jong, Y. Shevchenko, S. A. Becerra, G. M. Whitesides, J. C. Weaver, C. Hoberman, K. Bertoldi, *Nat. Commun.* **2016**, *7*, 10929.
[15] J. Berger, H. Wadley, R. McMeeking, *Nature* **2017**, *543*, 533.
[16] X. Zheng, X. Guo, I. Watanabe, *Mater. Des.* **2021**, *198*, 109313.
[17] T. Frenzel, M. Kadic, M. Wegener, *Science* **2017**, *358*, 1072.
[18] H. Lee, S. Kwon, J. Min, S.-M. Jin, J. H. Hwang, E. Lee, W. B. Lee, M. J. Park, *Science* **2024**, *383*, 70.
[19] R. Xu, Z. Zeng, Y. Lei, *Nat. Commun.* **2022**, *13*, 2435.
[20] M. Malinauskas, A. Žukauskas, S. Hasegawa, Y. Hayasaki, V. Mizeikis, R. Buividas, S. Juodkazis, *Light: Sci. Appl.* **2016**, *5*, e16133.
[21] D. Gu, X. Shi, R. Poprawe, D. L. Bourell, R. Setchi, J. Zhu, *Science* **2021**, *372*, eabg1487.

[22] J. Fan, L. Zhang, S. Wei, Z. Zhang, S.-K. Choi, B. Song, Y. Shi, *Mater. Today* **2021**, *50*, 303.
[23] R. L. Truby, J. A. Lewis, *Nature* **2016**, *540*, 371.
[24] X. Kuang, D. J. Roach, J. Wu, C. M. Hamel, Z. Ding, T. Wang, M. L. Dunn, H. J. Qi, *Adv. Funct. Mater.* **2019**, *29*, 1805290.
[25] X. Zheng, W. Smith, J. Jackson, B. Moran, H. Cui, D. Chen, J. Ye, N. Fang, N. Rodriguez, T. Weisgraber, C. M. Spadaccini, *Nat. Mater.* **2016**, *15*, 1100.
[26] X. Zheng, I. Watanabe, S. Wang, T.-T. Chen, M. Naito, *Mater. Des.* **2024**, *237*, 112548.
[27] C. Zhang, D. A. Mcadams, J. C. Grunlan, *Adv. Mater.* **2016**, *28*, 6292.
[28] Z. Jia, Z. Deng, L. Li, *Adv. Mater.* **2022**, *34*, 2106259.
[29] H. Dai, W. Dai, Z. Hu, W. Zhang, G. Zhang, R. Guo, *Adv. Sci.* **2023**, *10*, 2207192.
[30] N. K. Katiyar, G. Goel, S. Hawi, S. Goel, *NPG Asia Mater.* **2021**, *13*, 56.
[31] M. C. Fernandes, J. Aizenberg, J. C. Weaver, K. Bertoldi, *Nat. Mater.* **2021**, *20*, 237.
[32] L. Jin, M. Yeager, Y.-J. Lee, D. J. O'Brien, S. Yang, *Sci. Adv.* **2022**, *8*, eabq3248.
[33] X. Zheng, Z. Fu, K. Du, C. Wang, Y. Yi, *J. Mater. Sci.* **2018**, *53*, 10194.
[34] D. Yan, J. Chang, H. Zhang, J. Liu, H. Song, Z. Xue, F. Zhang, Y. Zhang, *Nat. Commun.* **2020**, *11*, 1180.
[35] J.-S. Koh, E. Yang, G.-P. Jung, S.-P. Jung, J. H. Son, S.-I. Lee, P. G. Jablonski, R. J. Wood, H.-Y. Kim, K.-J. Cho, *Science* **2015**, *349*, 517.
[36] Z. Zhakypov, K. Mori, K. Hosoda, J. Paik, *Nature* **2019**, *571*, 381.
[37] K. Liu, R. Sun, C. Daraio, *Science* **2022**, *377*, 975.
[38] H. Cui, Q. Zhao, L. Zhang, X. Du, *Adv. Intell. Syst.* **2020**, *2*, 2000138.
[39] S. Li, H. Bai, R. F. Shepherd, H. Zhao, *Angew. Chem., Int. Ed.* **2019**, *58*, 11182.
[40] R. O. Ritchie, X. R. Zheng, *Nat. Mater.* **2022**, *21*, 968.
[41] X. Fang, J. Wen, L. Cheng, D. Yu, H. Zhang, P. Gumbsch, *Nat. Mater.* **2022**, *21*, 869.
[42] Q. Ge, A. H. Sakhaei, H. Lee, C. K. Dunn, N. X. Fang, M. L. Dunn, *Sci. Rep.* **2016**, *6*, 31110.
[43] A. Vidyasagar, S. Krödel, D. M. Kochmann, *Proc. R. Soc. A: Math., Phys. Eng. Sci.* **2018**, *474*, 20180535.
[44] S. Zhao, T. M. Evans, X. Zhou, *Powder Technol.* **2018**, *323*, 323.
[45] X. Zheng, T.-T. Chen, X. Jiang, M. Naito, I. Watanabe, *Sci. Technol. Adv. Mater.* **2023**, *24*, 2157682.
[46] X. Zheng, T.-T. Chen, X. Guo, S. Samitsu, I. Watanabe, *Mater. Des.* **2021**, *211*, 110178.
[47] J. Zhou, I. Watanabe, T. Yamada, *Compos. Struct.* **2022**, *302*, 116200.
[48] H. T. Kollmann, D. W. Abueidda, S. Koric, E. Guleryuz, N. A. Sobh, *Mater. Des.* **2020**, *196*, 109098.
[49] X. Zheng, X. Zhang, T.-T. Chen, I. Watanabe, *Adv. Mater.* **2023**, *35*, 2302530.
[50] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. Billinge, E. Holm, S. P. Ong, C. Wolverton, *npj Comput. Mater.* **2022**, *8*, 59.
[51] S. G. Louie, Y.-H. Chan, F. H. da Jornada, Z. Li, D. Y. Qiu, *Nat. Mater.* **2021**, *20*, 728.
[52] T. D. Sparks, S. K. Kauwe, M. E. Parry, A. M. Tehrani, J. Brgoch, *Annu. Rev. Mater. Res.* **2020**, *50*, 27.
[53] D. Lee, W. Chen, L. Wang, Y.-C. Chan, W. Chen, *Adv. Mater.* **2023**, *36*, 2305254.
[54] Y. Mao, Q. He, X. Zhao, *Sci. Adv.* **2020**, *6*, eaaz4169.
[55] S. C.-y. Shen, M. J. Buehler, *Commun. Eng.* **2022**, *1*, 37.
[56] V. Oommen, K. Shukla, S. Goswami, R. Dingreville, G. E. Karniadakis, *npj Comput. Mater.* **2022**, *8*, 190.
[57] B. Deng, A. Zareei, X. Ding, J. C. Weaver, C. H. Rycroft, K. Bertoldi, *Adv. Mater.* **2022**, *34*, 2206238.
[58] J.-H. Bastek, S. Kumar, B. Telgen, R. N. Glaesener, D. M. Kochmann, *Proc. Natl. Acad. Sci.* **2022**, *119*, 2111505119.
[59] S. Kumar, S. Tan, L. Zheng, D. M. Kochmann, *npj Comput. Mater.* **2020**, *6*, 73.
[60] N. A. Alderete, N. Pathak, H. D. Espinosa, *npj Comput. Mater.* **2022**, *8*, 191.
[61] S. Kench, S. J. Cooper, *Proc. Natl. Acad. Sci.* **2021**, *3*, 299.
[62] Z. Yang, M. J. Buehler, *Front. Mater.* **2021**, *8*, 740754.
[63] Y.-C. Hsu, Z. Yang, M. J. Buehler, *APL Mater.* **2022**, *10*, 4.
[64] Z. Yang, M. J. Buehler, *Adv. Mater.* **2023**, *35*, 2301449.
[65] X. Shang, Z. Liu, J. Zhang, T. Lyu, Y. Zou, *Mater. Today* **2023**, *70*, 71.
[66] C. S. Ha, D. Yao, Z. Xu, C. Liu, H. Liu, D. Elkins, M. Kile, V. Deshpande, Z. Kong, M. Bauchy, X. (Rayne) Zheng, *Nat. Commun.* **2023**, *14*, 5765.
[67] B. Peng, Y. Wei, Y. Qin, J. Dai, Y. Li, A. Liu, Y. Tian, L. Han, Y. Zheng, P. Wen, *Nat. Commun.* **2023**, *14*, 6630.
[68] J.-H. Bastek, D. M. Kochmann, *Proc. Natl. Acad. Sci.* **2023**, *5*, 1466.
[69] X. Y. Lee, J. R. Waite, C.-H. Yang, B. S. S. Pokuri, A. Joshi, A. Balu, C. Hegde, B. Ganapathysubramanian, S. Sarkar, *Nat. Comp.. Sci.* **2021**, *1*, 229.
[70] Y. Wang, Q. Zeng, J. Wang, Y. Li, D. Fang, *Comp. Methods in Applied Mechanics and Engineering* **2022**, *401*, 115571.
[71] M. Maurizi, C. Gao, F. Berto, *npj Comput. Mater.* **2022**, *8*, 247.
[72] T. Jin, X. Cheng, S. Xu, Y. Lai, Y. Zhang, *J. Mech. Phys. Solids* **2023**, *179*, 105398.
[73] M. Li, Y. Duan, J. Zhou, J. Lu, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2023**, pp. 12642–12651.
[74] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, K. Kreis, *Adv. Neural Inf. Proc. Syst.* **2022**, *35*, 10021.
[75] R. Fu, X. Zhan, Y. Chen, D. Ritchie, S. Sridhar, *Adv. Neural Inf. Proc. Syst.* **2022**, *35*, 8882.
[76] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, S. Savarese, in *Computer Vision–ACCV 2018: 14th Asian Conf. on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, Springer, Berlin, Heidelberg **2019**, pp. 100–116.
[77] Z. Liu, Y. Wang, X. Qi, C.-W. Fu, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2022**, pp. 17896–17906.
[78] A. Sanghi, H. Chu, J. G. Lambourne, Y. Wang, C.-Y. Cheng, M. Fumero, K. R. Malekshan, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2022**, pp. 18603–18613.
[79] N. Mohammad Khalid, T. Xie, E. Belilovsky, T. Popa, in *SIGGRAPH Asia 2022 conference papers*, Association for Computing Machinery, New York **2022**, pp. 1–8.
[80] J. Xu, X. Wang, W. Cheng, Y.-P. Cao, Y. Shan, X. Qie, S. Gao, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition.*, IEEE, Piscataway, NJ **2023**, pp. 20908–20918.
[81] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, S. Fidler, *Adv. Neural Inf. Process, Syst.* **2022**, *35*, 31841.
[82] Z. Liu, P. Dai, R. Li, X. Qi, C.-W. Fu, ISS: image as stepping stone for text-guided 3D shape generation, *arXiv preprint arXiv:2209.04145* **2022**.
[83] Z. Zhao, W. Liu, X. Chen, X. Zeng, R. Wang, P. Cheng, B. Fu, T. Chen, G. Yu, S. Gao, Michelangelo: conditional 3D shape generation based on shape-image-text aligned latent representation, *arXiv preprint arXiv:2306.17115* **2023**.
[84] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, M. Chen, Point-E: a system for generating 3D point clouds from complex prompts, *arXiv preprint arXiv:2212.08751* **2022**.
[85] H. Jun, A. Nichol, Shap-E: generating conditional 3D implicit functions, *arXiv preprint arXiv:2305.02463* **2023**.

[86] J. Wei, H. Wang, J. Feng, G. Lin, K.-H. Yap, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2023**, pp. 16805–16815.

[87] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, B. Poole, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2022**, pp. 867–876.

[88] M. J. Buehler, *Oxf. open mater. sci.* **2022**, *2*, itac010.

[89] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, Gretchen Krueger, Ilya Sutskever, in *Int. Conf. on Machine Learning*, PMLR, New York **2021**, pp. 8748–8763.

[90] F. Chollet, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**, pp. 1251–1258.

[91] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics*, Human Language Technologies, Minneapolis, Minnesota **2019**, *1*, pp. 4171–4186.

[92] A. Brock, T. Lim, J. M. Ritchie, N. Weston, Generative and discriminative voxel modeling with convolutional neural networks, *arXiv preprint arXiv:1608.04236* **2016**.

[93] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6629.

[94] T. Anvekar, R. A. Tabib, D. Hegde, U. Mudengudi, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2022**, pp. 2978–2985.

[95] C. Nash, Y. Ganin, S. A. Eslami, P. Battaglia, presented at *Int. Conf. on Machine Learning*, PMLR, New York **2020**, pp. 7220–7229.

[96] X. Zheng, X. Guo, Y. Yang, Z. Fu, K. Du, C. Wang, Y. Yi, *J. Phys. Chem. C* **2018**, *122*, 16803.

[97] G. Dong, Y. Tang, Y. F. Zhao, *J. Eng. Mater. Techn.* **2019**, *141*, 011005.

[98] T. Gupta, M. Zaki, N. A. Krishnan, Mausam, *npj Comput. Mater.* **2022**, *8*, 102.

[99] F. Yin, X. Chen, C. Zhang, B. Jiang, Z. Zhao, J. Fan, G. Yu, T. Li, T. Chen, ShapeGPT: 3D shape generation with a unified multi-modal language model, *arXiv preprint arXiv:2311.17618* **2023**.

[100] C. Brauer, D. M. Aukes, in *Proc. of the 5th Annual ACM Symp. on Computational Fabrication*, Association for Computing Machinery, New York **2020**, pp. 1–8.

[101] Dawson-Haggerty, et al., Trimesh, version 3.2.0, https://trimesh.org/ (accessed: December 2019).

[102] I. Watanabe, D. Setoyama, N. Nagasako, N. Iwata, K. Nakanishi, *Int. J. Numer. Methods Eng.* **2012**, *89*, 829.

[103] I. Watanabe, A. Yamanaka, *Int. J. Mech. Sci.* **2019**, *150*, 314.