

Shangting You¹

Department of NanoEngineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: shy139@eng.ucsd.edu

Jiaao Guan¹

Department Electrical and Computer Engineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: jig186@eng.ucsd.edu

Jeffrey Alido

Chemical Engineering Program,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: jalido@ucsd.edu

Henry H. Hwang

Department of NanoEngineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: hhh004@eng.ucsd.edu

Ronald Yu

Department of Computer Science and
Engineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: ronalyu@ucsd.edu

Leilani Kwe

Department of NanoEngineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: lkwe@ucsd.edu

Hao Su

Department of Computer Science and
Engineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mail: haosu@eng.ucsd.edu

Shaochen Chen²

Department of NanoEngineering,
University of California San Diego,
9500 Gilman Dr.,
La Jolla, CA 92093
e-mails: chen168@ucsd.edu;
chen168@eng.ucsd.edu

Mitigating Scattering Effects in Light-Based Three-Dimensional Printing Using Machine Learning

When using light-based three-dimensional (3D) printing methods to fabricate functional micro-devices, unwanted light scattering during the printing process is a significant challenge to achieve high-resolution fabrication. We report the use of a deep neural network (NN)-based machine learning (ML) technique to mitigate the scattering effect, where our NN was employed to study the highly sophisticated relationship between the input digital masks and their corresponding output 3D printed structures. Furthermore, the NN was used to model an inverse 3D printing process, where it took desired printed structures as inputs and subsequently generated grayscale digital masks that optimized the light exposure dose according to the desired structures' local features. Verification results showed that using NN-generated digital masks yielded significant improvements in printing fidelity when compared with using masks identical to the desired structures.

[DOI: 10.1115/1.4046986]

Keywords: additive manufacturing, computer-integrated manufacturing, nontraditional manufacturing processes, production systems optimization

1 Introduction

Light-based three-dimensional (3D) printing methods using digital light processing (DLP) technique [1,2], such as projection micro-stereolithography (PuSL) [3], dynamic optical projection

¹These authors contributed equally to this work.

²Corresponding author.

Manuscript received November 22, 2019; final manuscript received April 11, 2020; published online May 14, 2020. Assoc. Editor: Qiang Huang.

stereolithography [4], continuous liquid interface production [5], and microscale continuous optical bioprinting (μ COB) [6], have emerged as promising tools for fabricating functional devices such as tissue engineering scaffolds [6,7], implantable medical devices [8,9], microfluidic devices [10], microsensors [11], and micro-robots [12] for a variety of applications in medicine, manufacturing, and consumer products due to their rapid fabrication speed and micron-scale fabrication resolution. For these applications, high-fidelity fabrication for the 3D printed part as compared to its intended design is highly desired. However, functional devices are often made of a mixture of polymeric materials and functional elements such as cells [7,10], micro/nanoparticles [11,12], and micelles. These elements often scatter the incoming light during the photopolymerization process, and additionally, the polymer materials themselves may have turbidity, form crystallites, or even micropores during and/or after polymerization, which can also give rise to light scattering [13]. Such light scattering ultimately reduces fidelity in the final product due to suboptimal polymerization of both the intended and unintended design areas (Fig. 1). Figures 1(d) and 1(e) show a comparison between 3D printing of non-scattering material and scattering material under the same fabrication condition. In Fig. 1(d), we used 100% poly(ethylene glycol) diacrylate (PEGDA) ($M_n=575$) with 1% w/V Irgacure 819. In Fig. 1(e), we utilized 100% PEGDA ($M_n=575$), 1% w/V Irgacure 819, and 1% w/V glass microbeads (diameter = $4\text{ }\mu\text{m}$). The 3D printed “blood vessel” structure is hollow as shown in Fig. 1(d), while it is clogged in Fig. 1(e) due to light scattering. Therefore, it is challenging to achieve high-fidelity and fine resolution in printing turbid materials using light-based 3D printing methods. Generally, the fabrication resolution and fidelity achievable in turbid materials are worse than in optically clear materials; thus, it would take more effort to optimize proper exposure dosages during printing. Such process optimization adds significant cost and time for product development, making it especially onerous in the case of 3D bioprinting where expensive cells are involved in the printing process to make biological tissues.

In most implementations of light-based 3D printing, binary digital masks identical to the desired structures are used. We will term these “identical masks,” where in binary fashion, 1 (shaded white) represents light exposure and 0 (shaded black) represents no light exposure. However, directly applying these binary identical masks will not produce the exact copies of the desired structures due

to challenges such as the aforementioned light scattering effect, thus motivating us to modulate the input mask in ways that might sufficiently compensate. With advances in DLP techniques, patterns utilizing grayscale values (ranging from 0 to 255 instead of binary 0 and 1) can be employed in light-based 3D printing. We expect that by using grayscale masks which are not necessarily identical to the desired patterns could compensate or counterbalance the effect of scattering, thus resulting in 3D printed structures as designed. Unfortunately, the light scattering properties of the prepolymer materials are complicated (e.g., vary during printing and overtime), making it difficult to a priori model the scattering and photopolymerization behavior of these materials as well as calculate the grayscale digital masks for 3D printing.

Recent advances in machine learning (ML) technologies based on deep neural networks (NNs) have successfully demonstrated their capability in assisting industrial manufacturing [14]. For example, researchers have reported the use of an NN-based ML technique to help optimize the processing parameters of inkjet-based 3D printing [15], fused filament fabrication-based 3D printing [16,17], and laser powder bed fusion [18–21]. However, to the best of our knowledge, using ML to assist in DLP-based 3D printing has not yet been reported. Moreover, it is far more complicated for NN to optimize a 2D image than to optimize a few scalar parameters.

Here, we report the use of deep NN-based ML to study the 3D printing behavior of light scattering materials and to generate grayscale digital masks to mitigate the effect of scattering in light-based 3D printing. A NN is trained to model the inverse process of 3D printing, where the input is the desired structure, and the output is the grayscale digital mask. We used 300 mask-structure pairs, which were produced by an in-house DLP-based 3D printer, to train the NN. Masks generated by the NN have then used to 3D print the desired target structures. Finally, we compared the printed structures created from NN-suggested masks to the structures printed from traditional identical masks. The results show that higher fabrication resolution and better fidelity can be achieved by our ML-assisted approach.

2 Experimental Setup

2.1 Modeling Digital Light Processing-Based Three-Dimensional Printing. Figure 1(a) shows the setup of a

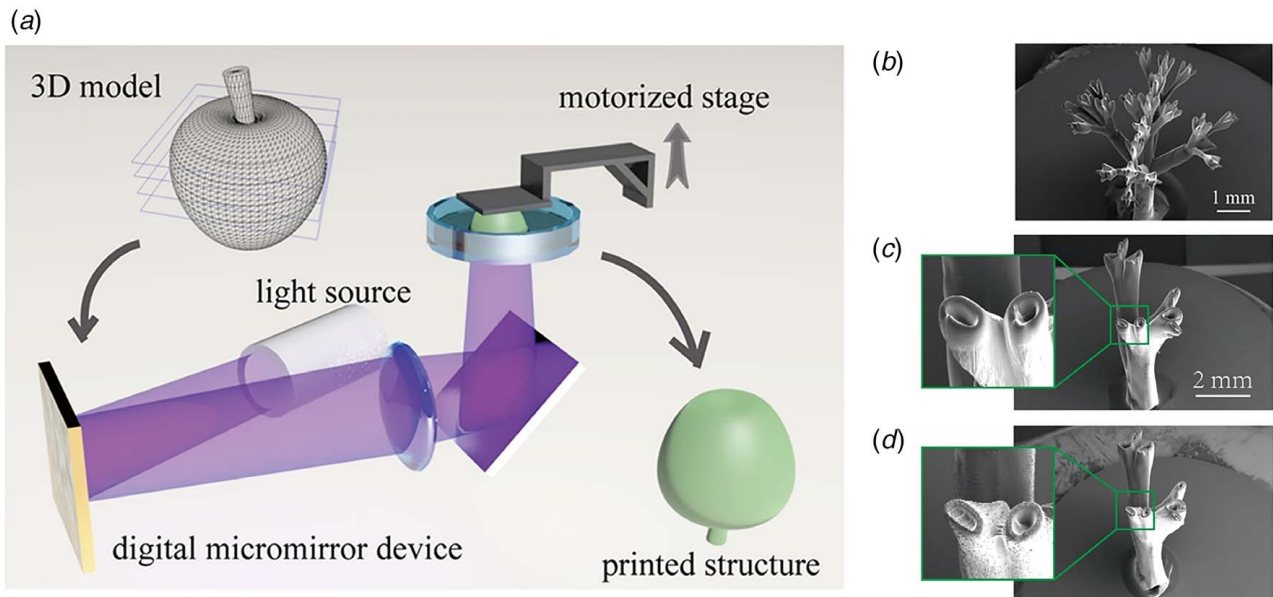


Fig. 1 Schematic of the 3D printer and the printed structures. (a) The DLP-based 3D printer setup. (b) SEM image of printed 3D “fractal tree” with a non-scattering material. (c) A “blood vessel” fabricated with a non-scattering material and the zoom-in detail of vessel opening. (d) A “blood vessel” fabricated with a scattering material and the zoom-in detail of vessel opening.

DLP-based 3D printer in this work. A light source of 405 nm wavelength illuminates the digital micromirror device (DMD). The DMD chip consists of 4 million of micromirrors, which can be individually controlled to flip toward two different directions, thus displaying a pattern. The reflected light is patterned by the DMD and is projected onto the liquid state prepolymer solution by lenses. A computer is used to control the DMD to display the cross sections at different position of the 3D model. Upon light exposure, the photosensitive material polymerizes, forming a thin layer of solid structure. The motorized stage brings the solidified part up by a layer thickness, which is typically tens to hundreds of microns, then unpolymerized material refills the gap. The DMD then displays the next cross section and photopolymerizes the next layer. By repeating this process, a 3D construct is thus printed. Figures 1(b) and 1(c) show the scanning electron microscopy (SEM) images of the printed structures using this 3D printer. The whole multi-layer 3D printing process can be discretized into multiple single-layer prints, which is the focus of this study.

There are several tunable variables for a single-layer print: (1) the digital mask determines the shape of the exposed area; (2) the exposure duration; and (3) the light intensity defines the exposure dose. We can combine these variables into a “generalized digital mask,” where the grayscale value of any given pixel on the digital mask represents the local exposure dose. We abstracted the 3D printer as a nonlinear time-invariant system. The input of the system is

the digital mask, which is a 512×512 pixel grayscale image, where the grayscale value represents the local exposure dose. The output of the system is the single-layer 3D printed structure, which is represented by a 512×512 binary image, where 1 represents a solidified area and 0 represents a void area.

2.2 Neural Network. The NN models the inverse process of 3D printing, which takes a 512×512 binary image (the desired printed structure) as an input and generates a 512×512 grayscale image (the digital mask) as the output (Fig. 2(a)). We used mask-structure pairs from the 3D printer to train the NN. The digital masks are grayscale images of an assortment of random shapes, e.g., checkerboards, discs, and rectangles (Fig. 3). The 3D printer then used these masks to print their corresponding structures. The exposure duration was fixed to 5 s. During printing, the actual light intensity of the maximum grayscale value (255) was measured to be 5.6 mW/cm^2 and 0 mW/cm^2 for the minimum grayscale value (0). The light source was a light-emitting diode centered at 405 nm wavelength. The prepolymer material was 50% (V/V) PEGDA ($M_n = 575$) aqueous solution, with 1% (w/V) lithium phenyl-2,4,6-trimethylbenzoylphosphine as photoinitiator, and 0.1% (w/V) glass microbeads (diameter = $4 \mu\text{m}$) as scattering particles. The microscope images of the printed structures are then processed into binary images with the use of custom code. We collected

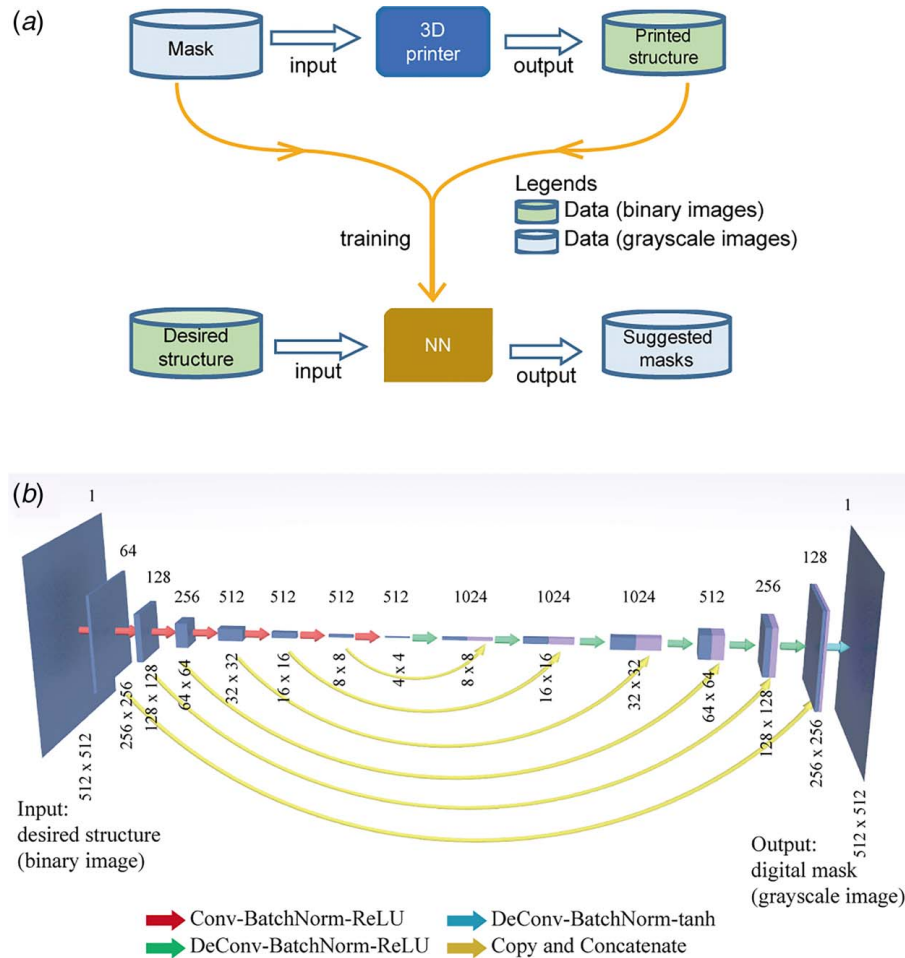


Fig. 2 Data flow and architecture of the neural network. (a) The 3D printer takes the digital masks as input, and output the printed structure. The neural network takes the desired structure as input, and output the suggested digital mask. The input and output of the 3D printer are used to train the neural network. **(b)** The neural network has a 14-layer convolutional neural network architecture with U-Net style skip connections. The rectangles represent the feature maps, with their feature resolution denoted at the bottom, and their feature channel number denoted on the top.

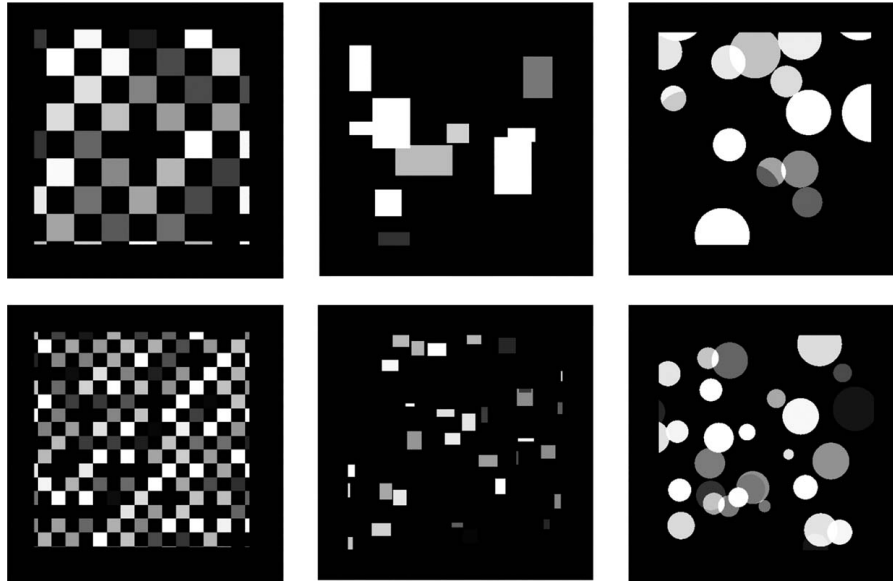


Fig. 3 Examples of the training digital masks

300 mask-structure pairs, which were then augmented to 900 pairs by random rotation about the image center. The detailed training method is available in the [Appendix](#).

The architecture of our NN is adapted from the generator design in image-to-image style transfer which is an encoder-decoder fully convolutional network with U-Net style skip connections [22–24], a schematic depiction of which is shown in Fig. 2(b). We also introduced partially cycle-consistent image-to-image translation loss to better meet our task characteristics which will be explained in the [Appendix](#). The NN consists of 14 building blocks, where each building block is a convolution or deconvolution layer followed by batch normalization [25] and the activation function. The convolution layers in the first seven building blocks are set with stride 2 to down-sample the features, while the deconvolution layers in the last seven building block up-sample the feature using stride 2 to recover

the output image to the original resolution. The activation function of the first 13 building block is a rectifier linear unit function [26]. The last building block uses a hyperbolic tangent function ($\tanh$) as the activation function in order to restrict the output value in the range of -1 and 1 , which is then linearly mapped to the grayscale value ranging from 0 to 255. There are also skip connections that copy the feature maps from the down-sampling process and append to the corresponding feature maps of the up-sampling process, allowing the network to learn more precise local information.

3 Results and Discussion

To evaluate our ML method, we designed several target structures that were never used in the course of training the NN. The

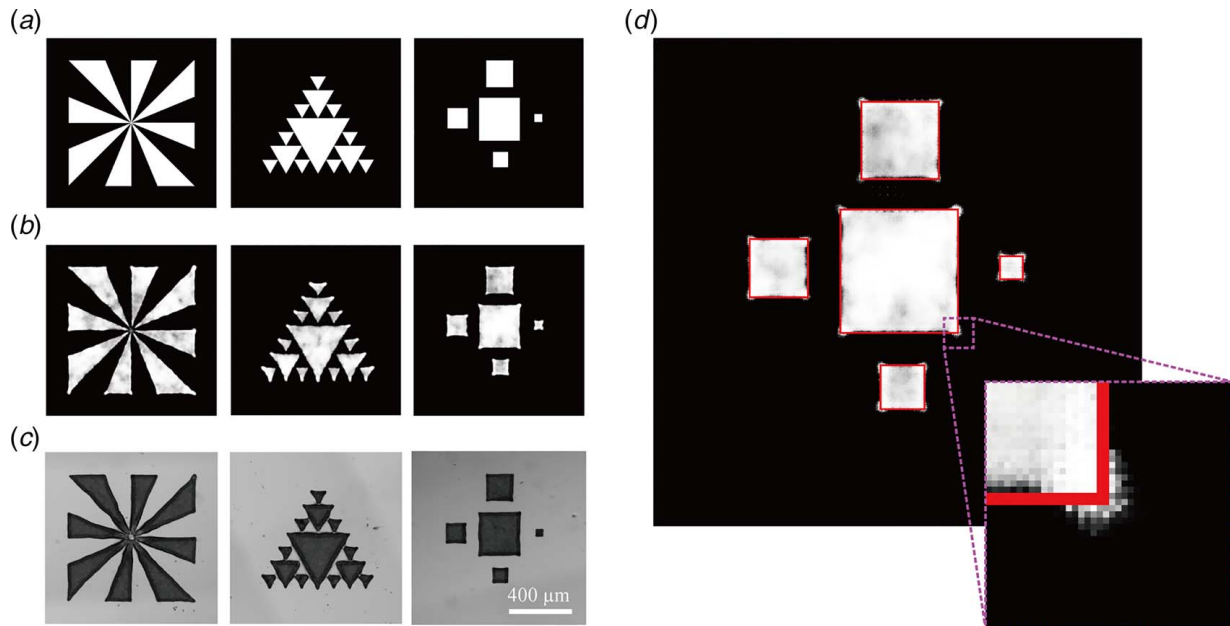


Fig. 4 Comparison of the targets, the NN-suggested masks, and the microscopic images of the printed structures. (a) The target structures. (b) The NN-suggested masks for the targets. (c) The actual printed structures using the masks from b. (d) The NN-suggested mask overlaid with its target (shown as red contour).

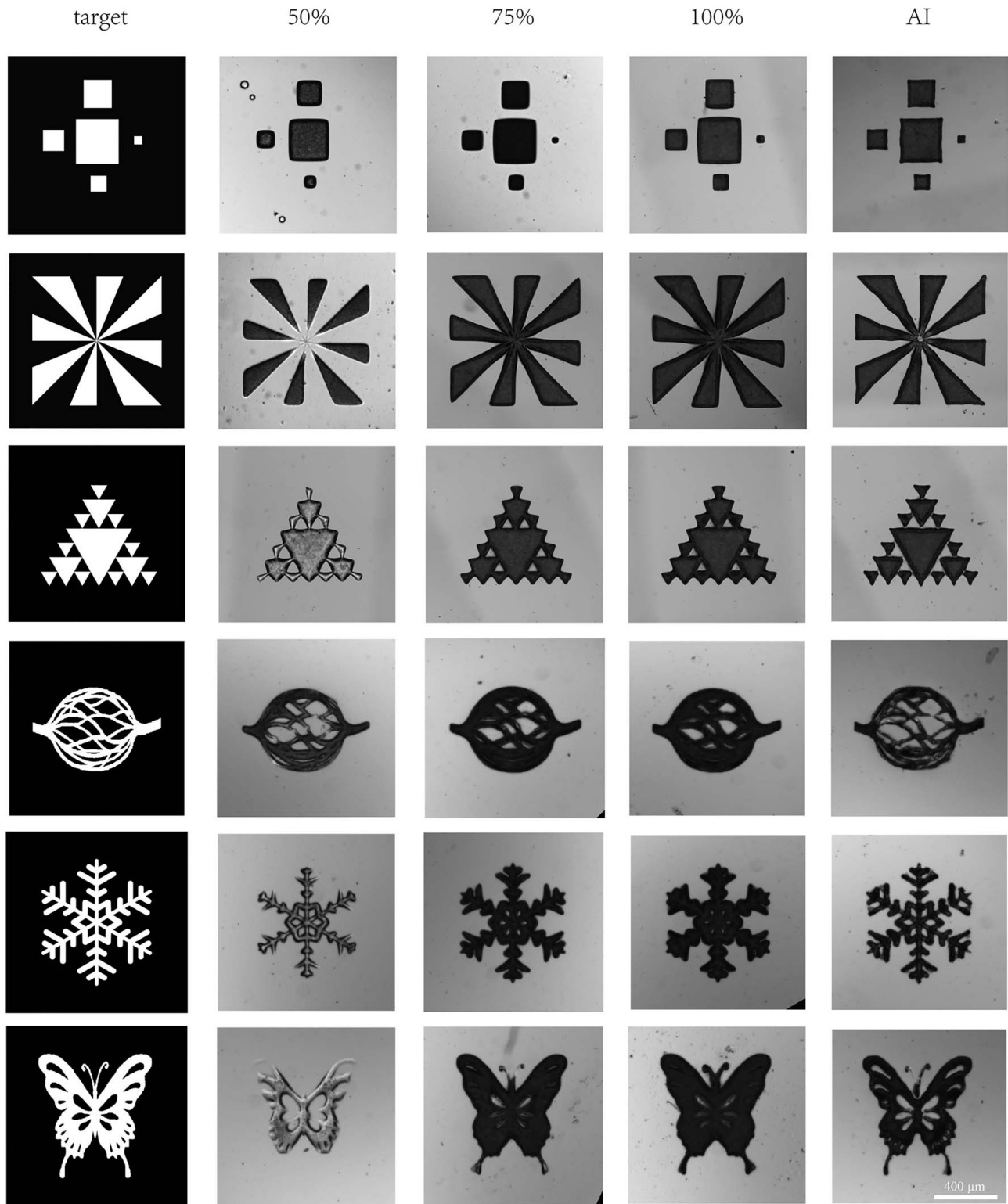


Fig. 5 Printed structures using NN-suggested mask and identical mask. The first column is the target structures (designs). The second to fourth columns are the printing result using the identical masks at 50%, 75%, and 100% exposure dose. The fifth column uses the NN-suggested masks.

NN-suggested digital masks for these targets, after which we then used our 3D printer to print the structures using these masks and compare them to the targets.

As shown in Fig. 4, the NN-suggested masks (Fig. 4(b)) are not necessarily identical to the desired structure (Fig. 4(a)). Compared to the traditional “identical masks,” the NN-suggested masks show significant use of grayscale variation and local feature deformation.

3D printing using the NN-suggested masks resulted in structures (Fig. 4(c)) that more closely match our desired structures. We find that the NN-suggested mask is able to compensate for the scattering effect by “stretching out” at the corners and “squeezing in” at the edges, as shown in Fig. 4(d). The NN’s behavior matches our human intuition of how we would counter the scattering effect.

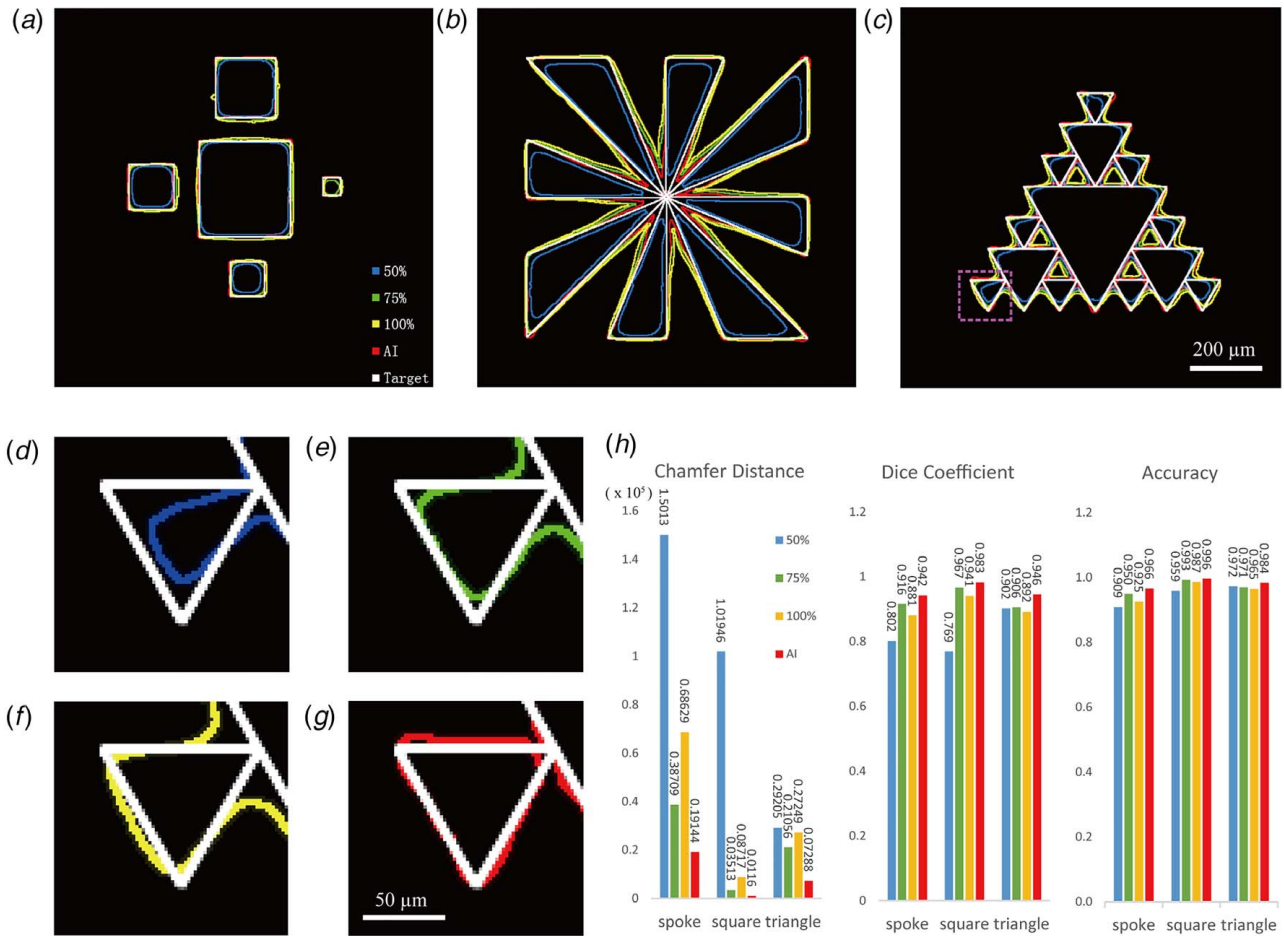


Fig. 6 Printed structures using NN-suggested masks and identical masks. (a)–(c) Overlaid contours of the structures printed under different exposure doses with identical masks indicated as dose percentages in the legend, the structure printed with NN suggested masks indicated as AI in the legend, and the target ground truth structure indicated as Target. (d)–(g) Zoom-in views in the dashed frame in panel (c), where the contours of the structures printed under four different conditions are isolated. (h) The Chamfer distances, Dice coefficient, and Accuracy score between the printed structures and the desired targets.

More importantly, using the NN-suggested mask, we can reach a printing fidelity that cannot be achieved by using traditional “identical masks.” Traditionally, people tend to optimize the printing process by only tuning the exposure dose but keep using an “identical mask.” To showcase the fidelity achievable with our method, we 3D printed and then compared the same target shape using both the NN-suggested masks as well as the traditional “identical masks.” For the traditional “identical masks,” we used exposure doses of 50% (2.8 mW/cm²), 75% (4.2 mW/cm²), and 100% (5.6 mW/cm²) to simulate how an operator might optimize the 3D printing by tuning the global exposure dose.

Figure 5³ shows the printed results of several targets using traditional “identical masks” at different exposure doses or using the NN-suggested masks. These targets include simple geometries as well as complex shapes. For all these targets, the NN-suggested masks yield the best printing fidelity and are able to fabricate the finest features such as sharp corners.

Figure 6 shows the overlaid contours of part of the printed structures in Fig. 5. The contours represent the structures printed using the “identical masks” at 50%, 75%, and 100% exposure doses, the structure printed using NN suggested masks, and the target structure as shown by the legend. Compared with the targets, the 50% dose structures are shrunken considerably compared with the

target (Fig. 6(d)), indicating under-exposure condition; while the 100% dose structures are expanded well beyond the target structure (Fig. 6(f)), indicating over-exposure. The 75% dose structures expand at the edges but also shrink at the corners, thus indicating over-exposure at some locations and under-exposure at others (Fig. 6(e)). This suggests that by using an “identical mask,” we cannot achieve proper exposure across the entire printing area.

The NN-suggested masks digitally individualize the local exposure dose depending on the local feature; hence, an optimized map of exposure doses can be achieved across the entire printing area. We can see from Figs. 6(a)–6(g) that the red (NN-suggested) contours best match the targets. Significantly, fine features such as sharp corners seem particularly well-preserved. Thus, the NN-suggested masks outperform the “identical masks.”

In order to quantitatively evaluate the printing fidelity of the printed structures shown in Figs. 5(a)–5(c), we compare those printed structures with their targets using Chamfer distance defined by Eq. (1) [27], Dice coefficient (equivalently F1 score) defined by Eq. (2) [28], and the accuracy score defined by accuracy = number of correct pixels/total number of pixels.

$$\text{Chamfer}(P, Q) = \sum_{p \in P} \min_{q \in Q} \|p - q\|_2 + \sum_{q \in Q} \min_{p \in P} \|p - q\|_2 \quad (1)$$

In Eq. (1), P and Q are the two binary images (or, two sets of non-zero pixels) we compare, p and q are the positions of individual

³The artistic butterfly silhouette used in Fig. 5 is designed by Vexels.com.

non-zero pixels from the corresponding sets.

$$Dice(P, Q) = \frac{2 \sum_i^N p_i q_i}{\sum_i^N p_i^2 + \sum_i^N q_i^2} \quad (2)$$

In Eq. (2), P and Q are the two binary images we compare, p_i and q_i are the binary pixel values of index i . N is the total number of pixels on the image.

If a pair of images have a greater similarity, they have a smaller Chamfer distance and a larger Dice coefficient and accuracy score. The calculated Chamfer distances, Dice coefficient, and accuracy between the 12 printed structures and their corresponding targets are shown in Fig. 6(h). All three indices suggest that using the NN-suggested mask, we can achieve better printing fidelity, which matches our qualitative evaluation.

The evaluation results have shown that our ML approach can help to address the scattering issue in light-based 3D printing, achieving a better fabrication fidelity and resolution than the traditional non-ML method. It should be noted that the NN we presented does not take the material properties as input variables. Therefore, NN should be trained individually for different materials. However, we believe that our method and the NN architecture can be applied to different materials to address the scattering problem in light-based 3D printing.

4 Conclusions

We have successfully demonstrated the use of ML to assist light-based 3D printing. When using turbid materials to fabricate functional devices, the printing fidelity deteriorates due to light scattering. The NNs allow us to study the relationship between the input digital mask and the actual output printed structure. Using the image-to-image fully convolutional NN that takes the desired structure as input and the grayscale digital mask as output, we succeeded in training the NN with a notably small amount of data (300 original mask-structure pairs). After training, the NN provides a digital mask with a digitally optimized light dose map. Compared to traditional “identical masks,” the NN-suggested masks mitigate the scattering effect and enable better fabrication resolution and fidelity. Such intelligent advice empowered by ML could minimize the trial-and-error inherent in optimizing printing parameters, thus significantly reducing the costs across the board for a time, labor, resources, customized parts, and time-to-delivery.

One limitation of our approach is that we do not consider the 3D printer’s specifications and the materials’ properties as input parameters of the neural network. Therefore, one trained neural network can only be applied to the specific material and specific 3D printer which it is trained with. In order to apply it to another material or a different 3D printer, the neural network need to be re-trained by a training dataset generated on the specific 3D printer with the specific material.

We expect that this method can be applied to light-based 3D bio-printing, where complex 3D scaffolds embedding live cells with micron-scale features to mimic the native biological tissue are highly desired. This will further create a new paradigm for 3D bio-printing of functional organs and tissues due to expensive cell sources, patient-specific design, and required microscale printing resolution.

Acknowledgment

This work was supported in part by the National Institutes of Health (R01EB021857, R21 AR074763, R33HD090662) and National Science Foundation (CMMI-1907434). Part of the work is performed at the San Diego Nanotechnology Infrastructure (SDNI) of UCSD, a member of the National Nanotechnology Coordinated Infrastructure (NNCI), which is supported by NSF (Grant ECCS-1542148).

Appendix

Neural Network Implementation and Objective Function.

Our objective is to find proper masks according to any target structure. Intuitively, we can achieve this objective by training the NN alone with the mask-structure data pairs we have made. The trained NN will learn to mimic the mapping from our dataset. However, we have limited data to train the NN, which prevents our NN from generalizing well on the entire structure space.

To address this challenge, we introduce a slave NN which learns the mapping of the 3D printer (the mapping from the mask to the structure). The slave NN has a similar architecture as the master NN as shown in Fig. 2(b), and the difference is that the last layer of the slave NN has two output channels and the tanh is replaced with a softmax function. We train the master NN, which models the inverse process of 3D printing (i.e., from the structure to the mask), based on not only the data generated by the actual 3D printer but also the extra information provided by the slave NN, thus we can allow the master NN to fit a broader and smoother distribution of data.

We formulate the training process of the NN as solving a minimization problem with the following objective function. For simplicity in expression, we define a function $S: X \rightarrow Y$, which represents the slave NN that learns the mapping from any grayscale mask x to the binary structure y . In the optimal case, the slave NN behaves exactly as the 3D printer, which is denoted as S^* . Similarly, we denote $M: Y \rightarrow Z$ as the transformation of master NN that learns a mapping from the desired binary structure y to a suggested grayscale mask z .

We first look at the slave NN. Since we want the slave NN to mimic the behavior of a 3D printer, i.e., we want the output of S to be as close as the output of S^* for the same input x , the ideal objective function for the slave NN can be expressed as Eq. (A1)

$$L_{Slave}^* = E_x[loss(S(x), S^*(x))] \quad (A1)$$

where $loss$ can be any loss function.

However, we only have limited access to the 3D printer S^* due to the cost and time, and we decide therefore only to use the data pre-generated by S^* for training, which are the 300 mask-structure pairs. We denote the training dataset as (X, Y) , such that any (x, y) chosen from (X, Y) satisfies $S^*(x) = y$. According to this data set, we can come up with a new objective function as Eq. (A2)

$$L_{Slave} = E_{x,y}[loss(S(x), y)] \quad (A2)$$

This objective (Eq. (A2)) is ultimately the same as the ideal objective (Eq. (A1)) if we have unlimited data. However, due to the limited amount of data we have, we expect the trained S to have a relatively high variance. This variance can be a form of useful noise to help generalize the master NN, which will be trained using this slave NN.

Next, we are looking at the master NN. We define the ideal objective equation for optimizing the master NN as Eq. (A3)

$$L_{Master}^* = E_y[loss(S^*(M(y)), y)] \quad (A3)$$

We can see that M is trained to minimize the error between the actual output structure and the desired structure. Again, we use a data set (X, Y) instead of S^* during training. According to this data set, we can come up with a new objective function as Eq. (A4)

$$L_{Master,1} = E_{x,y}[loss(M(y), x)] \quad (A4)$$

In the optimal case $M(y) = x$. Since we know $S^*(x) = y$, we will have $S^*(M(y)) = y$ for the y ’s in set Y . This means the new objective leads to the same optimized M as the ideal objective (Eq. (A3)). However, we will have a severe generalization issue with this new objective. Imagine a small disturbance is applied to the input y , a well generalized $M(y)$ will then output an image $\tilde{x}$ close to x . However, notice that S^* can potentially be very non-linear, which means that $S^*(\tilde{x})$ can be very different from $S^*(x)$ even though $\tilde{x}$ is very close to x . We may then interpret this effect as having

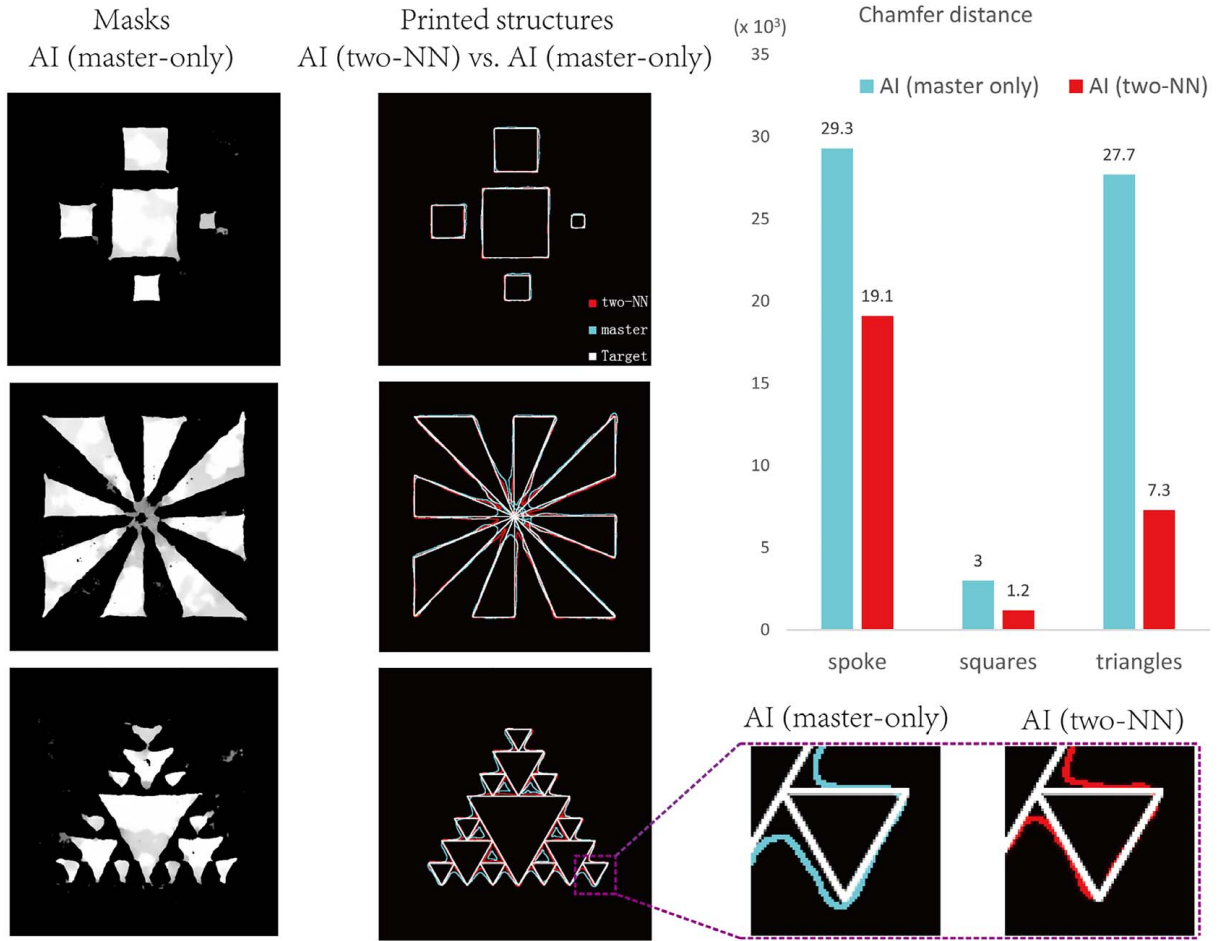


Fig. 7 Comparison between the two-NN design and the master-only design. The first column is the masks generated by the master-only design. The masks generated by the two-NN design can be found in Fig. 4. The second column is the contours of the printed structures using the two-NN design and the master-only design comparing with the target. The bar chart is the Chamfer distance between the prints using two-NN design and the target, and the Chamfer distance between the prints using master-only design and the target.

relatively high variance if $S^*(\tilde{x})$ is not smooth enough. Due to this issue, we need to find a way to better generalize M .

Let's take a look at another way to address the ideal master NN objective (Eq. (A3)) which is to replace S^* by its approximation, S . The objective becomes Eq. (A5)

$$L_{Master,2} = E_y[\text{loss}(S(M(y)), y)] \quad (\text{A5})$$

This objective has the benefit of generalizing well thanks to the information previously learned from S . The disadvantage is the cumulated error in the term $S(M(y))$. Notice that both S and M in this objective are neural networks and will potentially have some error even after convergence. If we use this $L_{Master,2}$ as objective, the calculation $S(M(y))$ will suffer from the error of both S and M , making the trained M biased according to the error from S .

Since the two feasible master NN objective functions either have a small bias or small variance, we decide to combine them. The resulting objective combining $L_{Master,1}$ and $L_{Master,2}$ (Eqs. (A3) and (A4)), becomes Eq. (A6)

$$L_{Master} = E_y[\text{loss}(S(M(y)), y)] + \lambda * E_{x,y}[\text{loss}(M(y), x)] \quad (\text{A6})$$

where λ is a tunable weight term to control the tradeoff between bias and variance. Notice that the first term is also a cycle-consistent loss, which is empirically proved to add performance to the

network when the data set is having this cycle-consistency property [29–31].

We decide to train S and M simultaneously instead of full training S before starting to train M . Our final objective function for both master NN and slave NN, therefore, is the combination of Eq. (A2) and Eq. (A6).

$$L = E_y[\text{loss}(S(M(y)), y)] + \lambda_1 * E_{x,y}[\text{loss}(M(y), x)] + \lambda_2 * E_{x,y}[\text{loss}(S(x), y)] \quad (\text{A7})$$

The advantage of training the master and slave NN simultaneously using Eq. (A7) is demonstrated in Fig. 7, where only training the master NN with Eq. (A4) is used for comparison. We noticed that the masks on the leftmost row of Fig. 7 generated by the master-only design contains some “ghost images,” which may be due to the poor generalization capability of the master-only design. The contours of the printed structures are shown in the second column, where the contours are the target, the two-NN design, and the master-only design indicated by the legend. We can see that the two-NN contours perform better than the master-only contours. The calculated Chamfer distances (Eq. (1)) also prove that the prints using the master-only design have a greater Chamfer distance than the prints using our two-NN design.

Following this approach, we can interpret S as a gradually reducing noise term, and this noise will make the output of S to bounce

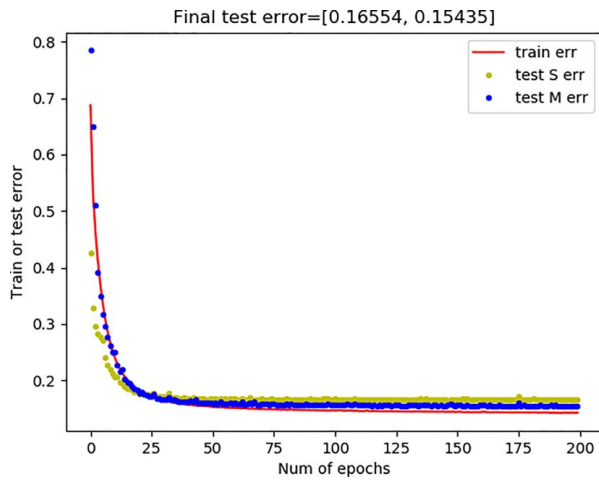


Fig. 8 Plot of errors versus episodes. The red line represents the loss calculated for master during training. The yellow and blue dots are errors of slave and master NN calculated using the test set. Final test error shows the value for slave and master NN errors after the last (200) epoch.

around the ground truth after convergence. We find this noise useful to generalize M to fit the distribution of varying output of S and avoid overfitting M to a deterministic output of S . We decide not to apply other commonly used noises, like adding Gaussian noise to the input or applying dropout to the network weights during training [32], while updating M . In terms of S , we add Gaussian noise to the input with zero mean, initial standard deviation 0.1, and 0.97 decay rate at each iteration.

In terms of hyperparameter choices, we use the L1 norm as the loss function for comparing grayscale images, and we use the cross-entropy loss for the loss between binary images. The tradeoff weight λ_1 is set to 0.1 and λ_2 is 1.

Data Augmentation. Consider the small amount of training data we had access to, we implemented data augmentation techniques to better train the networks. Typical image processing tasks use a hierarchical algorithm [33] or use data augmentation to improve the learning of many invariance properties such as shift-invariance, rotational invariance, and deformation invariance [23,24,34]. We augment our data by applying rotation and flipping with respect to the image center. In our implementation, the 300 pairs of data were augmented to 900 pairs following this method.

Hyperparameters and Training Details. We trained both networks under PyTorch [35] framework with mini-batch stochastic gradient descent using Adam solver [36]. The batch size is set to 10 due to the memory limit (8 GB). The momentum parameters are set to (0.9, 0.999), and the learning rate is 0.00001. All network model weights and biases are initialized using random samples from normal distribution with zero mean and 0.02 standard deviation. The whole training process took six and a half hours on a personal computer with a GTX 1070Ti discrete graphics processing unit (GPU). Once trained, the NN takes only a few seconds to calculate an output digital mask on a personal computer, and the speed is about ten times faster when utilizing the GPU. The training curve Fig. 8 shows that the network is converging smoothly. We used 33 pairs of test data isolated from the train data to calculate the test errors, and these errors represent that our networks can indeed generalize well on unseen data. Although the error value does not give intuitive results, since the slave NN and master NN errors are calculated differently, we can still find out that the slave NN is converging faster than the master NN. This might prove that the mapping

learned by slave NN is a relatively easier transformation compared to what master NN learns.

References

- [1] Hwang, H. H., Zhu, W., Victorine, G., Lawrence, N., and Chen, S., 2018, "3D-Printing of Functional Biomedical Microdevices via Light-and Extrusion-Based Approaches," *Small Methods*, **2**(2), p. 1700277.
- [2] Zhu, W., Ma, X., Gou, M., Mei, D., Zhang, K., and Chen, S., 2016, "3D Printing of Functional Biomaterials for Tissue Engineering," *Curr. Opin. Biotechnol.*, **40**, pp. 103–112.
- [3] Sun, C., Fang, N., Wu, D. M., and Zhang, X., 2005, "Projection Micro-Stereolithography Using Digital Micro-Mirror Dynamic Mask," *Sens. Actuators A: Phys.*, **121**(1), pp. 113–120.
- [4] Zhang, A. P., Qu, X., Soman, P., Hribar, K. C., Lee, J. W., Chen, S., and He, S., 2012, "Rapid Fabrication of Complex 3D Extracellular Microenvironments by Dynamic Optical Projection Stereolithography," *Adv. Mater.*, **24**(31), pp. 4266–4270.
- [5] Tumbleston, J. R., Shirvanyants, D., Ermoshkin, N., Januszewicz, R., Johnson, A. R., Kelly, D., Chen, K., Pinschmidt, R., Rolland, J. P., Ermoshkin, A., Samulski, E. T., and DeSimone, J. M., 2015, "Continuous Liquid Interface Production of 3D Objects," *Science*, **347**(6228), pp. 1349–1352.
- [6] Zhu, W., Qu, X., Zhu, J., Ma, X., Patel, S., Liu, J., Wang, P., Lai, C. S. E., Gou, M., Xu, Y., Zhang, K., and Chen, S., 2017, "Direct 3D Bioprinting of Prevascularized Tissue Constructs With Complex Microarchitecture," *Biomaterials*, **124**, pp. 106–115.
- [7] Ma, X., Qu, X., Zhu, W., Li, Y.-S., Yuan, S., Zhang, H., Liu, J., Wang, P., Lai, C. S. E., and Zanella, F., 2016, "Deterministically Patterned Biomimetic Human iPSC-Derived Hepatic Model via Rapid 3D Bioprinting," *Proc. Natl. Acad. Sci.*, **113**(8), pp. 2206–2211.
- [8] Koffler, J., Zhu, W., Qu, X., Platoshyn, O., Dulin, J. N., Brock, J., Graham, L., Lu, P., Sakamoto, J., Marsala, M., Chen, S., and Tuszynski, M. H., 2019, "Biomimetic 3D-Printed Scaffolds for Spinal Cord Injury Repair," *Nat. Med.*, **25**(2), pp. 263–269.
- [9] Zhu, W., Tringale, K. R., Woller, S. A., You, S., Johnson, S., Shen, H., Schimelman, J., Whitney, M., Steinauer, J., Xu, W., Yaksh, T. L., Nguyen, Q. T., and Chen, S., 2018, "Rapid Continuous 3D Printing of Customizable Peripheral Nerve Guidance Conduits," *Mater. Today*, **21**(9), pp. 951–959.
- [10] Liu, J., Hwang, H. H., Wang, P., Whang, G., and Chen, S., 2016, "Direct 3D-Printing of Cell-Laden Constructs in Microfluidic Architectures," *Lab Chip*, **16**(8), pp. 1430–1438.
- [11] Kim, K., Zhu, W., Qu, X., Aaronson, C., McCall, W. R., Chen, S., and Sirbulu, D. J., 2014, "3D Optical Printing of Piezoelectric Nanoparticle-Polymer Composite Materials," *ACS Nano*, **8**(10), pp. 9799–9806.
- [12] Zhu, W., Li, J., Leong, Y. J., Rozen, I., Qu, X., Dong, R., Wu, Z., Gao, W., Chung, P. H., Wang, J., and Chen, S., 2015, "3D-Printed Artificial Microfish," *Adv. Mater.*, **27**(30), pp. 4411–4417.
- [13] You, S., Wang, P., Schimelman, J., Hwang, H. H., and Chen, S., 2019, "High-Fidelity 3D Printing Using Flashing Photopolymerization," *Addit. Manuf.*, **30**, p. 100834.
- [14] Wuest, T., Weimer, D., Irgens, C., and Thoben, K.-D., 2016, "Machine Learning in Manufacturing: Advantages, Challenges, and Applications," *Prod. Manuf. Res.*, **4**(1), pp. 23–45.
- [15] Wang, T., Kwok, T.-H., Zhou, C., and Vader, S., 2018, "In-Situ Droplet Inspection and Closed-Loop Control System Using Machine Learning for Liquid Metal Jet Printing," *J. Manuf. Syst.*, **47**, pp. 83–92.
- [16] Gardner, J. M., Hunt, K. A., Ebel, A. B., Rose, E. S., Zylich, S. C., Jensen, B. D., Wise, K. E., Siochi, E. J., and Sauti, G., 2019, "Machines as Craftsmen: Localized Parameter Setting Optimization for Fused Filament Fabrication 3D Printing," *Adv. Mater. Technol.*, **4**(3), p. 1800653.
- [17] Khanzadeh, M., Rao, P., Jafari-Marandi, R., Smith, B. K., Tschoop, M. A., and Bian, L., 2017, "Quantifying Geometric Accuracy With Unsupervised Machine Learning: Using Self-Organizing Map on Fused Filament Fabrication Additive Manufacturing Parts," *J. Manuf. Sci. Eng.*, **140**(3), p. 031011.
- [18] Gaikwad, A., Imani, F., Yang, H., Reutzel, E., and Rao, P., 2019, "In Situ Monitoring of Thin-Wall Build Quality in Laser Powder Bed Fusion Using Deep Learning," *Smart Sustainable Manuf. Syst.*, **3**(1), p. 20190027.
- [19] Scime, L., and Beuth, J., 2018, "Anomaly Detection and Classification in a Laser Powder bed Additive Manufacturing Process Using a Trained Computer Vision Algorithm," *Addit. Manuf.*, **19**, pp. 114–126.
- [20] Scime, L., and Beuth, J., 2018, "A Multi-Scale Convolutional Neural Network for Autonomous Anomaly Detection and Classification in a Laser Powder bed Fusion Additive Manufacturing Process," *Addit. Manuf.*, **24**, pp. 273–286.
- [21] Williams, J., Dryburgh, P., Clare, A., Rao, P., and Samal, A., 2018, "Defect Detection and Monitoring in Metal Additive Manufactured Parts Through Deep Learning of Spatially Resolved Acoustic Spectroscopy Signals," *Smart Sustainable Manuf. Syst.*, **2**(1), p. 20180035.
- [22] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., 2017, "Image-to-Image Translation with Conditional Adversarial Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, pp. 5967–5976.

- [23] Ronneberger, O., Fischer, P., and Brox, T., 2015, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, N. Navab, J. Hornegger, W.M. Wells, and A.F. Frangi, ed., Springer International Publishing, Cham, pp. 234–241.
- [24] Long, J., Shelhamer, E., and Darrell, T., 2015, "Fully Convolutional Networks for Semantic Segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, June 7–12, pp. 3431–3440.
- [25] Ioffe, S., and Szegedy, C., 2015, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, Lille, France, July 6–11.
- [26] Nair, V., and Hinton, G. E., 2010, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th international conference on machine learning (ICML-10)*, Haifa, Israel, June 21–24, pp. 807–814.
- [27] Fan, H., Su, H., and Guibas, L. J., 2017, "A Point Set Generation Network for 3D Object Reconstruction From a Single Image," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26, pp. 605–613.
- [28] Milletari, F., Navab, N., and Ahmadi, S.-A., 2016, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, Stanford, CA, pp. 565–571.
- [29] Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., and Efros, A. A., 2016, "Learning Dense Correspondence via 3D-Guided Cycle Consistency," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, June 26–July 1, pp. 117–126.
- [30] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A., 2017, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 22–29.
- [31] Godard, C., Mac Aodha, O., and Brostow, G. J., 2017, "Unsupervised Monocular Depth Estimation With Left-Right Consistency," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26, pp. 270–279.
- [32] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 2014, "Dropout: A Simple Way to Prevent Neural Networks From Overfitting," *J. Mach. Learn. Res.*, **15**, pp. 1929–1958.
- [33] Huang, J., Kwok, T.-H., and Zhou, C., 2019, "Parametric Design for Human Body Modeling by Wireframe-Assisted Deep Learning," *Comput.-Aided Des.*, **108**, pp. 19–29.
- [34] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012, "ImageNet Classification With Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, Curran Associates, Inc., pp. 1097–1105.
- [35] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A., 2017, Automatic Differentiation in PyTorch.
- [36] Kingma, D. P., and Ba, J., 2015, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, May 7–9.