



HAL
open science

Machine learning and sequential subdomain optimization for ultrafast inverse design of 4D-printed active composite structures

Xiaohao Sun, Luxia Yu, Liang Yue, Kun Zhou, Frédéric Demoly, Ruike Renee Zhao, H. Jerry Qi

► **To cite this version:**

Xiaohao Sun, Luxia Yu, Liang Yue, Kun Zhou, Frédéric Demoly, et al.. Machine learning and sequential subdomain optimization for ultrafast inverse design of 4D-printed active composite structures. Journal of the Mechanics and Physics of Solids, In press, pp.105561. 10.1016/j.jmps.2024.105561 . hal-04450140

HAL Id: hal-04450140

<https://hal.science/hal-04450140>

Submitted on 9 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

**Machine learning and sequential subdomain optimization for
ultrafast inverse design of 4D-printed active composite structures**

Xiaohao Sun^{1#}, Luxia Yu^{1#}, Liang Yue^{1#}, Kun Zhou², Frédéric Demoly^{3,4}, Ruike Renee
Zhao⁵, H. Jerry Qi^{1*}

¹The George W. Woodruff School of Mechanical Engineering, Georgia Institute of
Technology, Atlanta, GA 30332, USA

²Singapore Centre for 3D Printing, School of Mechanical and Aerospace Engineering,
Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

³ICB UMR 6303 CNRS, Belfort-Montbéliard University of Technology, UTBM, 90010
Belfort, France

⁴Institut universitaire de France (IUF)

⁵Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA

These authors contributed equally to this work.

*Corresponding author: qih@me.gatech.edu

Abstract

Shape transformations of active composites (ACs) depend on the spatial distribution and active response of constituent materials. Voxel-level complex material distributions offer a vast possibility for attainable shape changes of 4D-printed ACs, while also posing a significant challenge in efficiently designing material distributions to achieve target shape changes. Here, we present an integrated machine learning (ML) and sequential subdomain optimization (SSO) approach for ultrafast inverse designs of 4D-printed AC structures. By leveraging the inherent sequential dependency, a recurrent neural network ML model and SSO are seamlessly integrated. For multiple target shapes of various complexities, ML-SSO demonstrates superior performance in optimization accuracy and speed, delivering results within second(s). When integrated with computer vision, ML-SSO also enables an ultrafast, streamlined design-fabrication paradigm based on hand-drawn targets. Furthermore, ML-SSO empowered with a splicing strategy is capable to design diverse lengthwise voxel configurations, thus showing exceptional adaptability to intricate target shapes with different lengths without compromising the high speed and accuracy. As a comparison, for the benchmark three-period shape, the finite element method and evolutionary algorithm (EA) method was estimated to need 227 days for the inverse design; the ML-EA achieved design in 57 min; the new ML-SSO with splicing strategy requires only 1.97 s. By further leveraging approximate symmetries, the highly efficient ML-SSO is employed to design active shape changes of 4D-printed lattice structures. The new ML-SSO approach thus provides a highly efficient tool for the design of various 4D-printed, shape-morphing AC structures.

Keywords: active composites; morphing structures; 4D printing; inverse design; machine learning.

1. Introduction

Shape transformation of active composites (ACs) depend on the spatial distribution and active response of constituent materials. Multimaterial 3D/4D printing[1-5] allows for easy implementation of voxel-level distribution of active materials/properties, thus permitting a vast space of active shape changes that can be achieved. To fully exploit this space, inverse design (i.e., finding the optimal material distribution to achieve a target shape) is highly desired, which is yet challenging due to the tremendous design space[6, 7]. To tackle these challenges, gradient-based and gradient-free methods have been developed. For instance, the gradient-based topology optimization (TO) has been used for designing active shape changes of ACs[8-10] or optimizing structure compliances[11]. The TO, however, typically requires the complicated derivation of gradients and may encounter difficulties when the ACs involve geometric or material nonlinearity. Alternatively, the gradient-free evolutionary algorithm (EA) has also been adopted for designing active shape changes of ACs[12-14]. In these EA implementations, finite element (FE) methods are used for numerous shape predictions (referred to as FE-EA), leading to high computational cost. Although reduced-order models have been developed to replace FE for faster EA, they suffer from either relatively low accuracy (against FE) [15] or limited design freedom (1D material distribution) [16]. Accurate and efficient inverse design strategies remain to be developed.

Machine learning (ML)[17] opens up new avenues for developing fast, computationally affordable, and high-fidelity models for both forward prediction and inverse design of a variety of material responses. Existing works mainly focused on optimizing or predicting mechanical properties of materials, such as strength and toughness of composites[18, 19], stress and strain fields of composites[20], Poisson's ratio of auxetic metamaterials[21], responses of soft pneumatic robots[22], among others [23, 24]. However, there is limited

work on exploiting ML for designing active shape-change responses of ACs[7]. For voxelized AC beams, Zhang et al.[25] explored performances of multiple ML models for the forward prediction problem. Our recent work[26] demonstrated that the recurrent neural network (RNN) is particularly suited for the beam problem as it inherently preserves a sequential dependency, similar to that in beam deformation. The RNN-based ML model thus demonstrated exceptional accuracy in forward shape predictions based on material distributions. The ML was then integrated with EA (ML-EA), enabling fast inverse designs for complicated target 2,5D shapes, typically in 10-60 minutes. Although ML-EA has significantly outperformed conventional methods such as FE-EA, it also has two major limitations. First, there remains a strong demand for faster or even real-time inverse designs for AC beams. One key limiting factor for the design speed is that EA seeks solutions in a vast design space comprising all voxels, which is relatively inefficient. Second, an ML model, once trained, is specific to certain beam length and voxel configuration, making it inapplicable to the design of target shapes that have different lengths (hence different voxel numbers) or finer features (such as large curvature) beyond the design space of current voxels.

This work addresses these issues by leveraging the sequential dependency of beam deformations. We first present a design approach that seamlessly combines an RNN ML model and a sequential subdomain optimization (SSO) algorithm to realize ultrafast inverse designs for target shapes of various complexities, producing results in second(s), which is hundreds or thousands times faster than conventional ML-EA method. In addition, the sequential characteristic allows for accurate predictions for beams with fewer lengthwise voxels than that of the training data. It also permits the splicing of multiple predicted shapes, extending accuracy to beams with more lengthwise voxels. Moreover, this splicing strategy, when integrated with ML-SSO, also permits ultrafast inverse designs for a spectrum of

target intricate shapes with different lengths, using lengthwise voxel configurations different from that for the ML training. Finally, the highly efficient ML-SSO is employed for the inverse design of active shape changes of 4D-printed lattice structures. Our approach thus facilitates the development of 4D printing towards intelligent and streamlined design and fabrication of various shape-morphing AC structures.

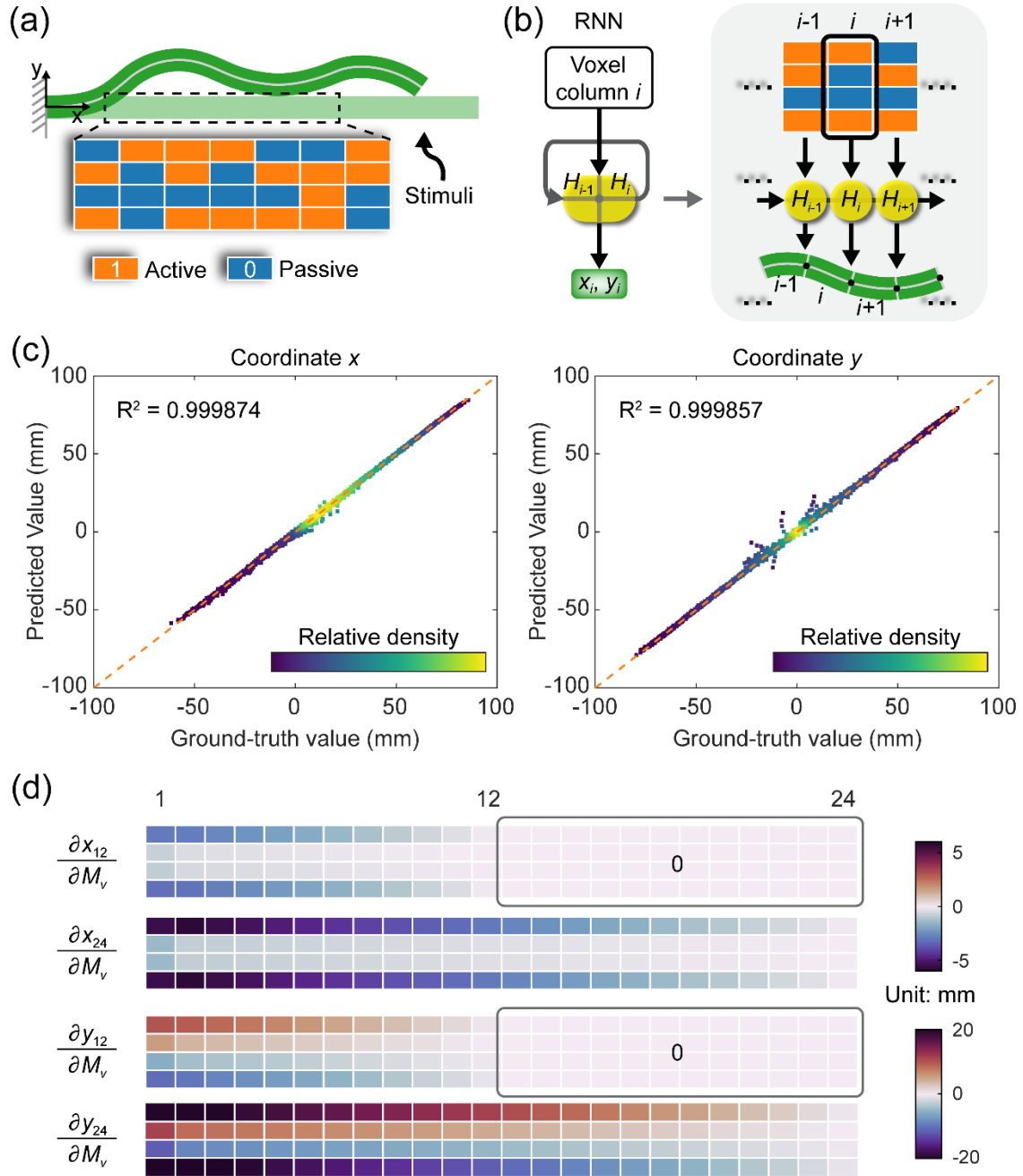


Figure 1. Schematic illustration of the active composite (AC) beam and the sequential dependency. (a) Actuation of an AC beam with voxel-level material distribution. The two materials, encoded as “1” (active) and “0” (passive), have a volume expansion mismatch. (b) Architecture of an RNN, which shows a similar structure with the AC beam. (c) Density scatter plots of the ground-truth versus ML-predicted values of the coordinate x (left) and y (right). The color indicates the relative point density. (d) Gradients (or sensitivities) of actuated coordinates (x, y) of the 12th and 24th voxel columns with respect to all voxels (i.e., $\partial x_i / \partial M_v$ and $\partial y_i / \partial M_v$ for $i=12$ and 24 , where $v=1, 2, \dots, 4 \times 24$ represents the voxel

number, M_v represents the material encoding of voxel v , based on an all-passive ($M_v=0$) state. The color of an arbitrary voxel represents the gradient of x_i or y_i (unit: mm) with respect to the encoding of this voxel (unitless).

2. Results and discussions

2.1 Physical problem and ML model

The physical problem is illustrated in **Figure 1a**. Here we consider an AC beam whose left end is fixed. It is made of two materials: an active material '1' expands under the thermal actuation while a passive material '0' does not. The beam is composed of N_x (length- or x -direction) \times N_y (thickness- or y -direction) voxels, each assigned with an active or passive material. The active shape change, which is induced by the expansion mismatch under the thermal actuation, depends on the material distribution (**Figure 1a**). With a dataset generated by FE simulations, an RNN-based ML model is trained to predict actuated shapes for given material distributions (**Figure 1b**). RNN can use the “past” information to predict the “current” response and thus is highly capable for learning sequential data. It is noted the deflection of the AC beam has a similar sequential characteristic, i.e., the displacement of any points only depends on the points to their left (or before them); the displacement of the points to their right (or after them) would not affect their displacement. This feature inspires us to use RNN. For each step, RNN receives a single column of voxel encodings and produces the coordinates of the column (sampled on mid-axis of the beam). By repeating this process, RNN sequentially processes all voxel columns to predict the shape of entire beam (**Figure 1b**). Details on the FE model, dataset, and the architecture and training of the ML model are provided in Sections 4.1 and 4.2.

As shown in our previous study[26], the RNN ML model demonstrates high accuracy for cases with different numbers of voxels. In this work, we adopt the ML model trained on $24 (N_x) \times 4 (N_y)$ voxels. Notably, the coordinates (x, y) are sampled from the mid-axis,

rightmost mesh point of each voxel column to represent a shape, which is different from the middle point as used in our previous work (**Figure 1b**). This choice of sampling points has two merits. First, it achieves higher prediction accuracy with $R^2 = 0.99987$ for x and 0.99986 for y , as shown in regression (or density scatter) plots of the ground-truth versus ML-predicted coordinates of the testing dataset (**Figure 1c**). The increased accuracy is attributed to the better sequential characteristic of the column rightmost points. Second, the new sampling points enable the ultrafast SSO optimizations as will be introduced in Section 2.2.

As mentioned above, since the left end of the actuated beam is fixed, the coordinate of a voxel column is dependent on the voxels in the current column and the left of the current column (nearer the fixed end). Similarly, the architecture of RNN determines that the “current” output is dependent on the “current” and “past” inputs. To illustrate this feature, we display the gradients (or sensitivities) of RNN-predicted coordinates (x, y) of the 12th and 24th voxel columns with respect to all voxels (i.e., $\partial x_i / \partial M_v$ and $\partial y_i / \partial M_v$ for $i=12$ and 24 , where $v=1, 2, \dots, 4 \times 24$ represents the voxel number, M_v represents the material encoding of voxel v), based on a state where all voxels are initially passive materials ($M_v=0$) (**Figure 1d**). The gradients are calculated through automatic differentiation as described in Section 4.3. We see that (x_{12}, y_{12}) depend on the first 12 voxel columns only, while (x_{24}, y_{24}) depend on all the voxels. The results evidently showcase that the RNN inherently preserves the sequential data dependency of the beam shape.

2.2 ML-SSO approach for ultrafast inverse design

Although the ML model achieves high accuracy in the forward prediction, applying it to the inverse map (from shape to design) is still challenging due to the “one-to-many” characteristic (i.e., one actuated shape could be achieved by distinct designs)[7]. This is

particularly the case considering the large design space and the high-dimensional data for shape representation. To address this challenge, a straightforward approach is to combine ML with optimization algorithms such as EA[26]. However, as a stochastic search algorithm, EA seeks solutions in a vast design space comprising all voxels, which is inefficient.

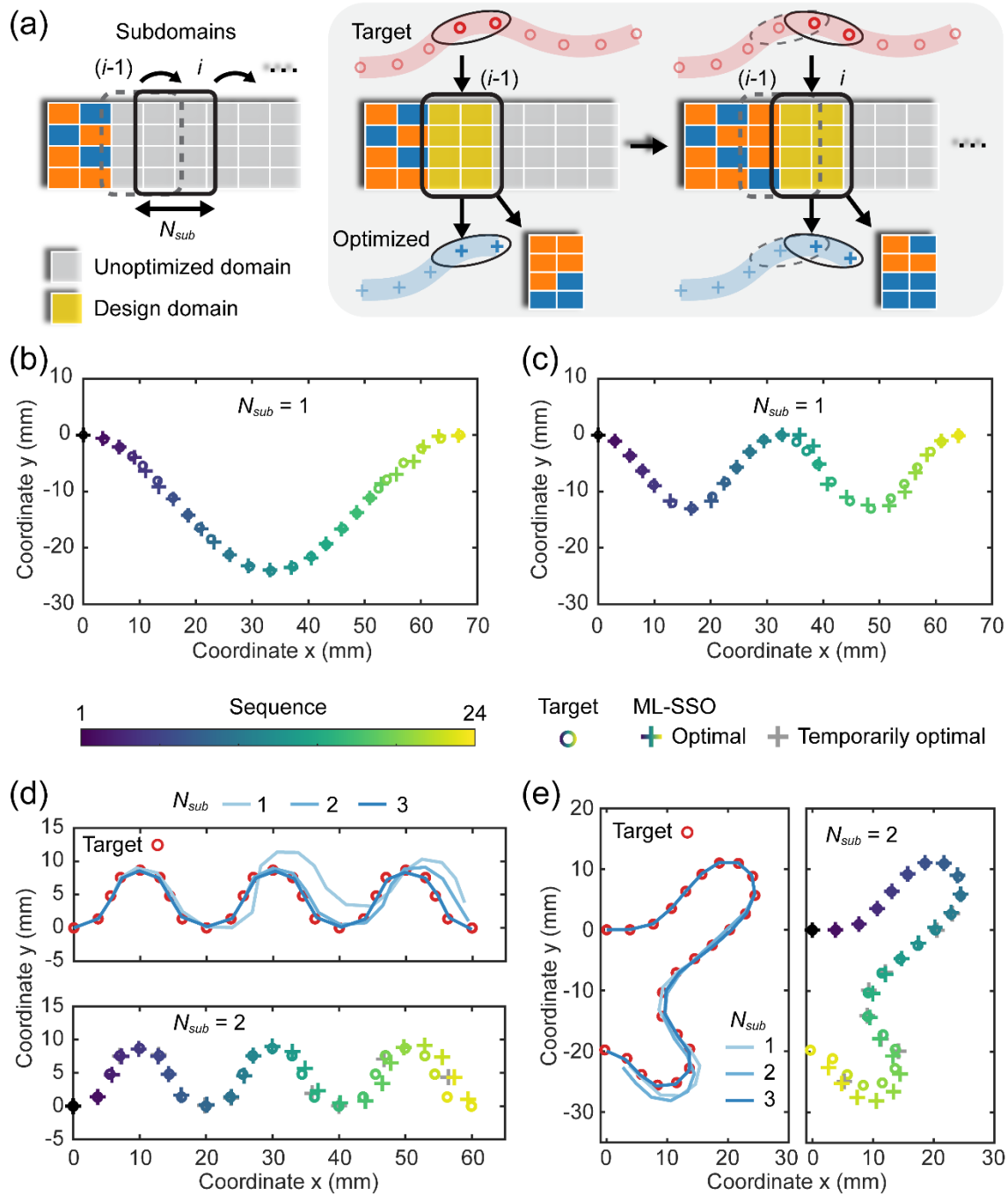


Figure 2. Demonstration of ML-SSO on inverse design problems. (a) Schematic of ML-SSO. (b) One-period target and ML-SSO optimized shape with $N_{sub}=1$. (c) Two-period target and ML-SSO optimized shape with $N_{sub}=1$. (d) Three-period target, ML-SSO optimized shape with different N_{sub} values, and illustration of optimization sequence with $N_{sub}=2$. (e) Half-butterfly target, ML-SSO optimized shape with different N_{sub} values, and illustration of optimization sequence with $N_{sub}=2$. All ML-SSO designs are based on $N_x=24$.

In this work, we present an ML-empowered SSO approach which can further leverage the inherent sequential characteristic. Instead of directly optimizing the entire domain, this approach sequentially optimizes subdomains, composed of N_{sub} columns, from left to right (**Figure 2a**). In each optimization step, SSO receives the coordinates of N_{sub} target points (highlighted red “o”) and optimizes the material distribution in the corresponding subdomain (highlighted yellow domain) such that the RNN-predicted coordinates of actuated points (highlighted blue “+”) match with the targets. Once a subdomain, e.g., ($i-1$), is optimized, the target points and design domain move one step forward for the next subdomain, i , as seen by the one-step sliding of the highlighting window in **Figure 2a**. Due to sequential dependency, the optimization of subdomain i would not affect the optimized preceding subdomains (those to the left of subdomain i) except the overlapping domain of ($i-1$) and i (see the rightmost panel of **Figure 2a**). Here, we choose N_{sub} to range from 1 to 3. When $N_{sub}=1$, there is no overlap between two consecutive subdomains. As will be detailed later, a larger N_{sub} can improve the optimization accuracy. For the optimization of individual subdomains, we employ a brute-force approach, calculating the subdomain shapes of all possible designs and retaining the optimal one. This ensures the local optimality of found solution while not requiring many shape evaluations as the chosen N_{sub} is small. For larger N_{sub} , the subdomain EA or other stochastic search algorithms may be used to reduce computational cost. The loss function L_{sub} for each subdomain is defined as

$$L_{sub} = \sqrt{\frac{1}{N_{sub}} \sum_{j=1}^{N_{sub}} \left[\left(x_{sub,j} - \hat{x}_{sub,j} \right)^2 + \left(y_{sub,j} - \hat{y}_{sub,j} \right)^2 \right]} \quad (1)$$

where $(x_{sub,j}, y_{sub,j})$ are the ML-predicted sampled point coordinates for the j -th column of the considered subdomain, $(\hat{x}_{sub,j}, \hat{y}_{sub,j})$ are the corresponding target coordinates. Compared to the ML-EA on entire domain, ML-SSO reduces computational cost in two aspects. First, the design space of a single subdomain is much smaller than the whole domain, thus drastically reducing the number of potential design evaluations required. For

instance, for $N_{sub} = 2$, the brute-force method assesses 256 ($=2^{Ny \times N_{sub}}=2^8$) potential subdomain designs per step, totaling 5888 ($=256 \times 23$, where $23 = (N_x - N_{sub} + 1)$ is the number of subdomains to optimize) evaluations, which is much smaller than the number (typically 75,000-375,000[26]) of global ML-EA. Second, these are the numbers of potential designs that need to be evaluated. In ML-SSO, the evaluation of each candidate design (comprising N_{sub} columns only) is also more efficient. This is because during each optimization step, RNN permits only processing the current subdomain and predicting its coordinates ($x_{sub,j}$, $y_{sub,j}$), $j \in [1, N_{sub}]$, based on the stored hidden state, thereby bypassing the need to evaluate the preceding or subsequent subdomains. This further significantly reduces computational costs compared to ML-EA which demands a full-domain shape evaluation for each candidate design.

2.3 Performance of ML-SSO on numerically generated target shapes

We now consider multiple numerically generated target shapes of different complexities to evaluate the performance of ML-SSO with different N_{sub} values. In this sub-section, we focus on the case with 24×4 voxels. The first two target shapes, the one-period and two-period sinusoidal shapes, are optimized using $N_{sub} = 1$. To provide a more intuitive understanding of ML-SSO, in **Figure 2b** and **2c**, we show the optimized and target shapes with symbols of varying colors to visualize the optimization sequence. Each optimized point (“+”) is the one closest to its the corresponding target point (“o”) among all possible designs for the specific subdomain. For both two targets, excellent agreements between the optimized shapes and the targets are achieved. Although some discrepancies can be observed, e.g., in the middle of the two-period shape where the curvature is relatively large, the subsequent optimized points can still accurately capture the target shape (**Figure 2c**), demonstrating the high capability of our approach.

Next, two more complicated targets are considered: the three-period shape and the half-butterfly shape. As shown in **Figure 2d** and **2e**, the sensitivity of optimized shapes to the N_{sub} is studied. For both targets, $N_{sub}=1$ leads to notable discrepancies between the optimized and target shapes. With the increased N_{sub} , the optimization accuracy is significantly improved. This is because ML-SSO operates on a greedy strategy, selecting the best shape at each step. Such an approach might lead to a short-sighted problem for regions with sharp turns, missing the target of the subsequent subdomain in the favor of the best match for the current one. Increasing N_{sub} effectively broadens the optimization foresight and can, to some extent, mitigate this problem, although it concurrently escalates the computational cost. To gain more insights, we show in **Figure 2d** and **2e** the optimization sequence for the cases with $N_{sub}=2$, where each voxel column (except the first and last ones) is involved in two consecutive subdomains and thus optimized twice (see, for example, the overlapping domain of $(i-1)$ and i in the rightmost panel of **Figure 2a**). The temporarily optimal point denotes the one achieved in the first optimization of its involved subdomain and is depicted in gray “+” symbols. The discrepancy between the temporarily optimal point and the optimal point often appears in regions with sharp turns, manifesting the short-sighted problem.

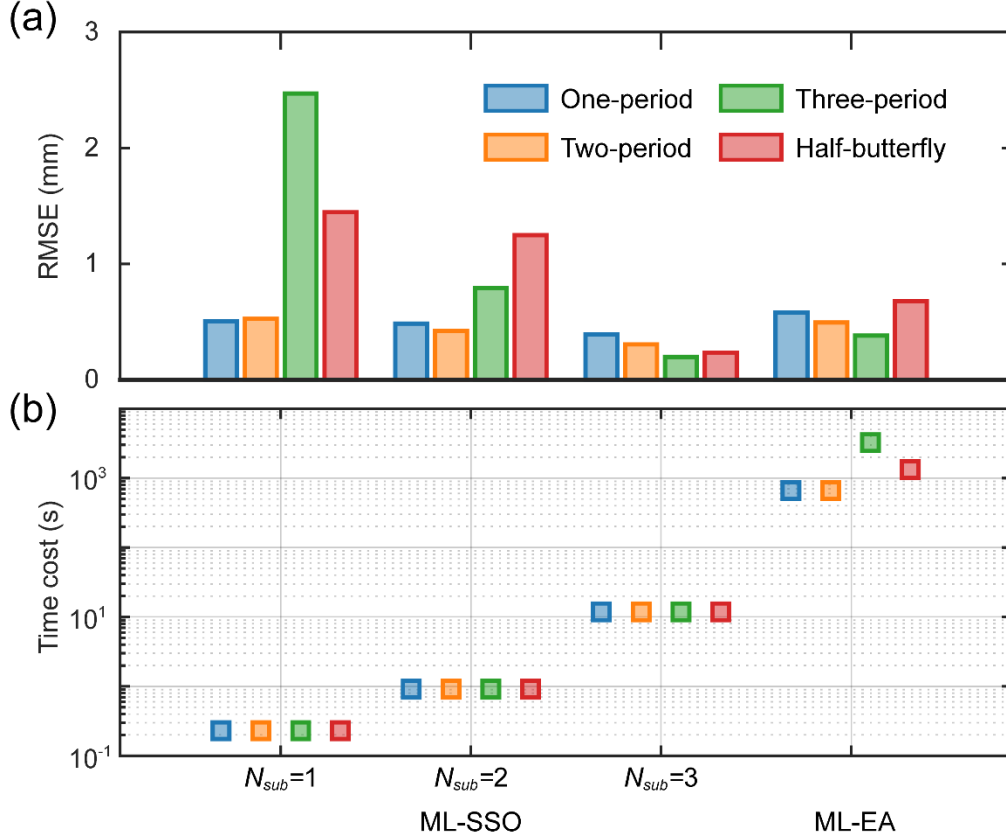


Figure 3. (a) RMSEs of optimized shapes by ML-SSO (with different N_{sub} values) and ML-EA for the four target shapes considered in **Figure 2**. The RMSE is calculated using Eq.(2). (b) Time cost of the corresponding optimizations, which are run using the Intel Core i9-10900 CPU and NVIDIA Quadro P620 GPU.

We further quantitatively compare the performance of ML-SSO with different N_{sub} values on these four targets in terms of accuracy (**Figure 3a**) and time cost (**Figure 3b**). The ML-EA optimization results are already available[26] and also compared here. The accuracy is quantified by the root-mean-squared errors (RMSEs)

$$\text{RMSE} = \sqrt{\frac{1}{2N_x} \sum_{i=1}^{N_x} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]} \quad (2)$$

between the optimized coordinates (x_i, y_i) and target coordinates (\hat{x}_i, \hat{y}_i) for all N_x sampling points. As shown in **Figure 3**, for the relatively simple one-period and two-period shapes, ML-SSO with $N_{sub}=1$ which only takes 0.23 s achieves similar RMSEs with that of ML-

EA which takes 660 s. For the most complex three-period shape, ML-SSO with $N_{sub}=3$ takes 11.8s and achieves much better RMSEs than that of ML-EA, which takes 3240 s. In short, ML-SSO outperforms or matches ML-EA in optimization accuracy while being significantly (at least two orders of magnitude) faster. This demonstrates very high efficacy of our approach in solving inverse material design problems for complicated target shapes, which is attributed to the seamless integration of RNN-ML and SSO that fully leverages sequential characteristics for highly efficient optimal design searches.

2.4 Streamlined 4D printing design and fabrication based on hand-drawn lines

Next, we apply the ultrafast ML-SSO approach to hand-drawn target shapes to enable a streamlined design-fabrication paradigm for 4D printing. In general, these target shapes are more challenging to optimize, because the discrete target points are uniformly sampled from the identified shape, which could have inappropriate spacing and thus produce difficulties for the optimization. Following our previous work[26], computer vision (CV) algorithms are employed for the automatic identification of target shapes from hand-drawn lines. The ML-SSO is then utilized for the inverse design, and the optimized material distributions can be readily 4D-printed using the grayscale digital light processing (g-DLP)[2, 4] technique. The material system is different from that of our previous work[26]. More details on the printing method, material and actuation mechanism are provided in Section 4.4.

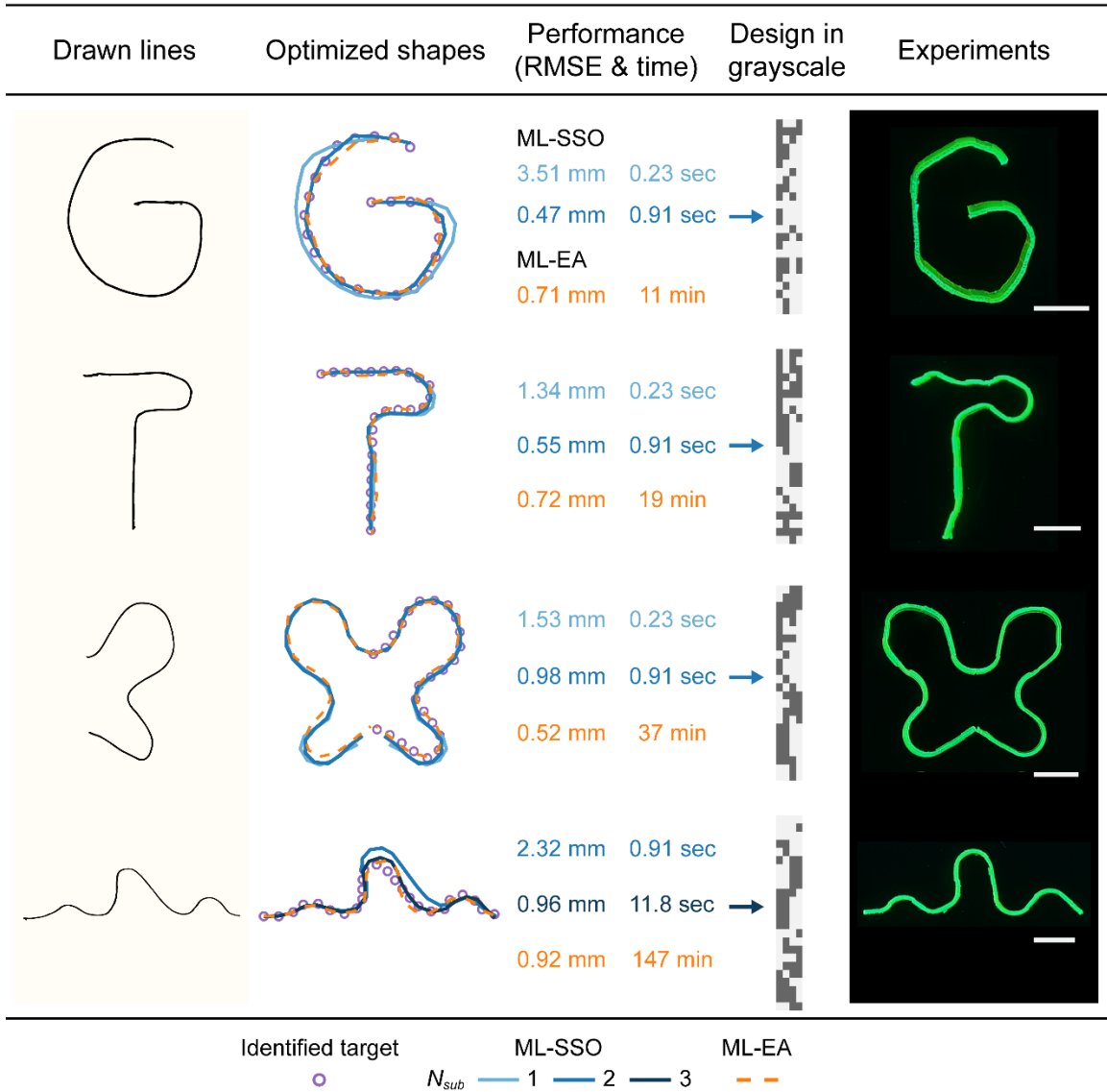


Figure 4. Ultrafast ML-SSO enabled streamlined design and fabrication for 4D printing based on hand-drawn lines. From top to bottom: “G”, “T”, half-butterfly, mountain. From left to right: hand-drawn lines, optimized shapes by ML-SSO (with different N_{sub} values) and ML-EA, the corresponding performance in terms of RMSE and time cost, the optimal ML-SSO designs selected (with greater N_{sub} , and RMSE < 1 mm) out of those in column 3 (arrows), and 4D-printed, actuated shapes. All scale bars: 10 mm.

Figure 4 shows the ML-SSO design and fabrication results for four different target shapes. The ML-EA optimized designs are available[26] and also presented here for the evaluation of ML-SSO performance. Shown in five columns (from left to right) are hand-drawn lines,

optimized shapes by ML-SSO (with different N_{sub} values, solid lines) and ML-EA (dashed lines) compared against the targets (symbols), the corresponding performance in terms of RMSE and time cost, the optimal ML-SSO designs selected (with greater N_{sub} , and RMSE < 1 mm) out of those in column 3 (arrows), and experimentally 4D-printed, actuated shapes. In column 5, all printed strips are initially flat and only the actuated shapes are shown. For all the targets, the optimized shapes by both ML-SSO and ML-EA agree well with the target shapes, whilst ML-SSO is much faster than ML-EA. The optimal ML-SSO designs are finally converted into grayscale slices for 4D printing through g-DLP, and the printed strips transform into target shapes upon actuation. These results demonstrate an ultrafast, streamlined design-fabrication paradigm for 4D printing.

2.5 Active beams with varying number of lengthwise voxels: forward predictions

The preceding sections demonstrate ultrafast inverse designs enabled by ML-SSO for both numerically generated and hand-drawn target shapes. However, these designs have been constrained to fixed beam length (80 mm) and voxel configuration (24×4 voxels) on which the ML model is trained, implying a relatively limited design space. For example, a configuration of $N_x=24$ lengthwise voxels might struggle to capture complicated target shapes, such as the six-period shape, which would be very challenging for $N_x=24$ as it would require four voxels in the length direction to capture one period. Therefore, more lengthwise voxels ($N_x > 24$) are needed to tackle this design problem.

To further enhance design capabilities, we aim for the ML-based methods to address both forward and inverse problems for beam structures with varying lengths or N_x values. Here, we fix the voxel number $N_y=4$ in thickness and the voxel size. Thus, the sequential (column-by-column) prediction ability of the RNN-ML allows it to be directly applied to structures with different N_x values and hence different lengths. For the case $N_x > 24$, we

propose an ML-based splicing method, with no need for retraining the model, for forward predictions with different N_x values.

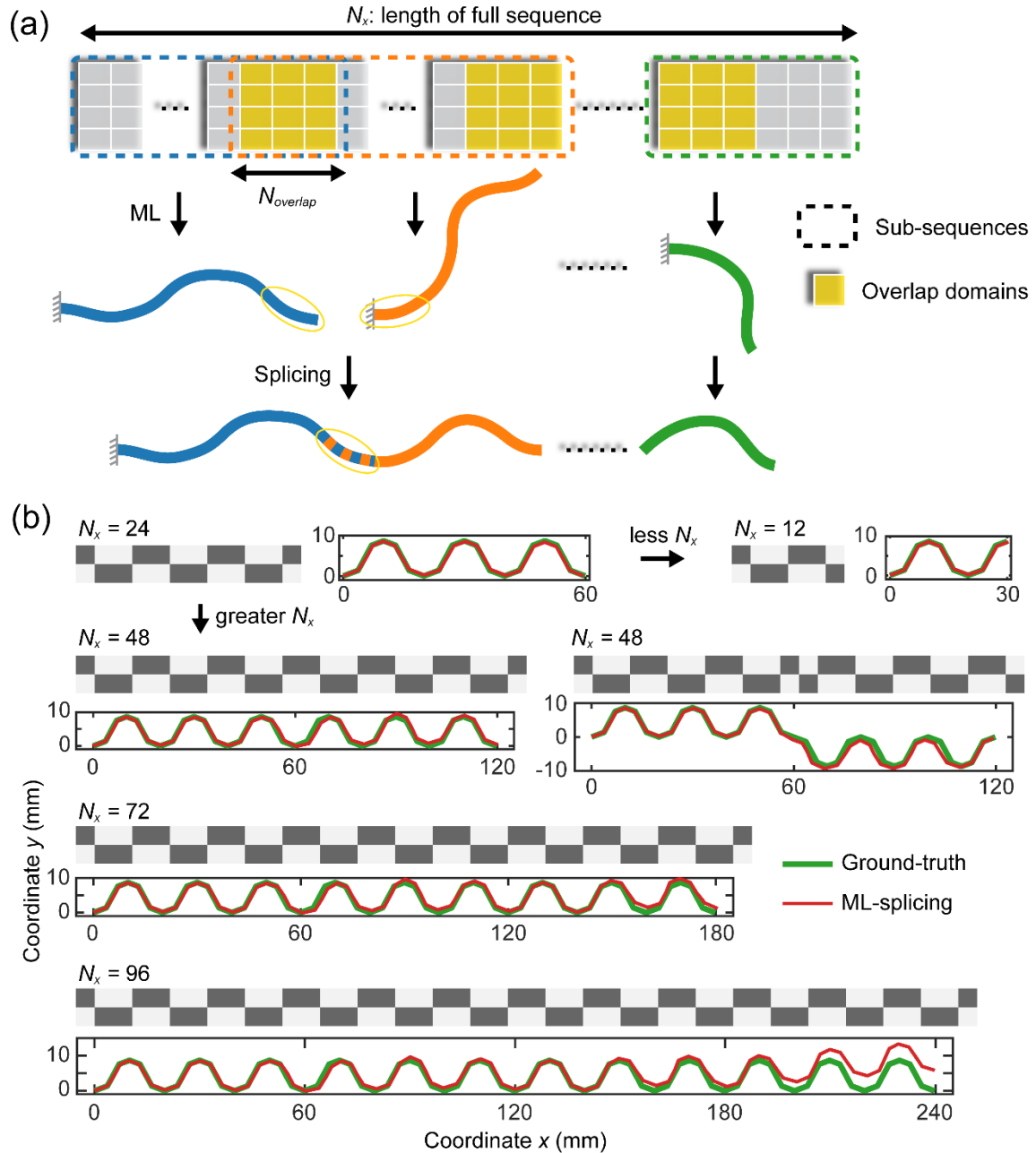


Figure 5. Demonstration of ML-splicing on forward prediction problems of beam structures with different N_x values. (a) Schematic of ML-splicing. (b) Prediction results for benchmark input sequences of different N_x values.

As shown in **Figure 5a**, an input sequence (referring to a material distribution) of any length N_x can be divided into multiple subsequences with overlaps, each with a length of 24, except for the last subsequence which can be shorter than 24. We use ML to predict the shape of all subsequences, then concatenate them in order to obtain the full-sequence shape. The predicted shapes of all subsequences are initially based on a fixed boundary condition on the left end, which we refer to as being in their local spaces, while the shape of the first subsequence is in the real space. We sequentially transform each subsequence shape into the real space by matching the coordinates in the overlapping regions between two consecutive subsequences. Here, we use the length of overlapping region to be $N_{overlap}=6$. Note that the correct local concatenation angle cannot be determined if there is no overlapping region.

Next, we perform forward predictions on multiple benchmark shapes of different lengths, including $N_x=12, 24, 48, 72, 96$ (**Figure 5b**). The input sequences with $N_x \leq 24$ are directly predicted using the RNN-ML without accuracy degradation. The input sequences with $N_x > 24$ are obtained by concatenating multiple sequences of the three-period shape and predicted using the ML-splicing strategy. Directly applying ML to these sequences would yield less accurate results. By using the ML-splicing strategy, we can achieve very high prediction accuracy. Note that the ML prediction errors will inevitably accumulate with increasing N_x and splicing times. Nonetheless, the ML-splicing predictions remain accurate for N_x up to 72, three times $N_x=24$; a noticeable discrepancy only arises at $N_x=96$. Our strategy thus demonstrates excellent robustness significantly (three times) beyond the sequence length ($N_x=24$) of the ML training set.

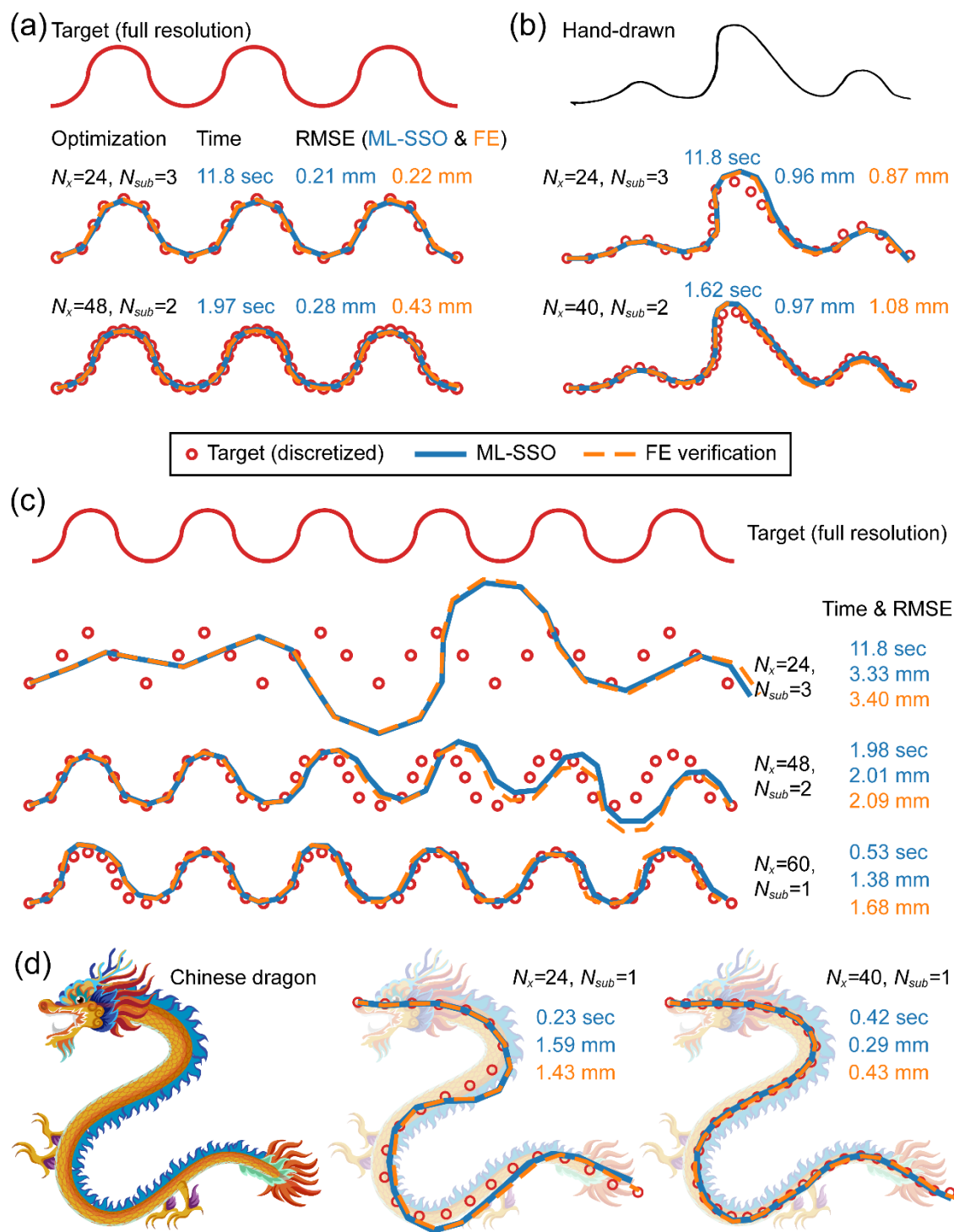


Figure 6. Demonstration of the splicing-empowered ML-SSO on inverse design problems using different N_x (numbers lengthwise voxels) or numbers of sampling points. Multiple target shapes are considered, including (a) three-period shape, (b) hand-drawn mountain shape, (c) six-period shape, and (d) “Chinese dragon” profile. Image by brgfx on Freepik (<https://www.freepik.com/free-vector/chinese-dragon-flying-clouds-isolated-white->

background_18054261.htm).

2.6 Active beams with varying number of lengthwise voxels: inverse designs

Next, we combine ML-SSO with the splicing strategy for inverse design problems. The splicing strategy works in a slightly different way from the forward prediction. In the forward prediction, each subsequence shape is transformed into the real space to infer the correct shape. In the inverse design, instead of transforming the shapes of potential designs into the real space for comparisons against the target, we transform the target shape (which is initially in the real space) into the local space for the optimization. This enables the seamless use of ML-SSO for each subsequence, bypassing unnecessary shape transformations and improving the optimization speed. We use multiple complicated target shapes (i.e., involving relatively sharp turns when represented using $N_x=24$ sampling points) to demonstrate the capability of our splicing-empowered ML-SSO. **Figure 6** shows the optimization results for these target shapes obtained with different N_x values. The optimized shapes together with the corresponding RMSEs and time cost are presented. FE simulations for the optimized designs are also performed, and the FE predicted shapes and their RMSEs are provided. The first two targets are the most complex target shapes considered so far: the three-period shape (**Figure 6a**) and hand-drawn mountain (**Figure 6b**). With the original $N_x=24$, $N_{sub}=2$ is not sufficient to accurately capture the two targets (**Figure 2d** and **4**); using $N_{sub}=3$ significantly increases the accuracy (with RMSEs of 0.21 mm and 0.96 mm) but also takes longer optimization time (11.8 s) (**Figure 6a** and **6b**). By using greater N_x values (48 and 40 respectively), which offer more sampling points, these complex targets with sharp turns can be rapidly captured with $N_{sub}=2$, achieving similar accuracies (with RMSEs of 0.28 mm and 0.97 mm) in much shorter times (1.97 s and 1.62 s) (**Figure 6a** and **6b**). To put these results in perspective, for the three-period shape, FE-EA approach was estimated to need 227 days for the inverse design, the ML-EA approach obtained the design in 54 min, our new approach requires a merely 1.97 s.

To further evaluate the capability of our approach, we consider a more complex six-period target shape obtained by chaining two three-period shapes (**Figure 6c**). The original $N_x=24$ fails to capture this target (RMSE=3.33 mm) even with $N_{sub}=3$ (which takes 11.8 s), and therefore greater N_x values are used. We then use $N_x=48$ with $N_{sub}=2$ and achieve an improved accuracy (RMSE=2.01 mm) in a much shorter time (1.98 s), but an evident discrepancy can still be observed. By further increasing $N_x=60$, even a smaller $N_{sub}=1$ successfully yields a high-precision optimized shape (RMSE=1.38 mm) in only 0.53 s (**Figure 6c**). As the last example, we take the profile of a “Chinese dragon” art as the target shape and again achieve excellent optimization results (RMSE=0.29 mm) very rapidly (0.42 s with $N_x=40$ and $N_{sub}=1$) (**Figure 6d**).

These results show that although we train the ML model on a certain voxel configuration, the splicing strategy allows for the ultrafast and accurate forward predictions and inverse designs for structures with variable N_x values and lengths by leveraging the inherent sequential characteristics.

2.7 Applications to rapid design of 4D-printed lattice structures

The inverse designs demonstrated so far have been constrained to the AC beam structures. However, the voxel-level, ultrafast inverse shape-change designs for more complex structures such as lattice structures are highly desired, which could have broader applications but is also more challenging due to increased number of involving voxels in an entire lattice. In this sub-section, we further show that by leveraging appropriate symmetries, the ML-SSO can be applied in designing the active shape transformation of 4D-printed lattice structures. We focus on a periodic square lattice and optimize the material distribution on its edges so that they achieve target shapes under actuation. The

design strategy can be also applied to other lattice structures, such as triangular and hexagonal lattices.

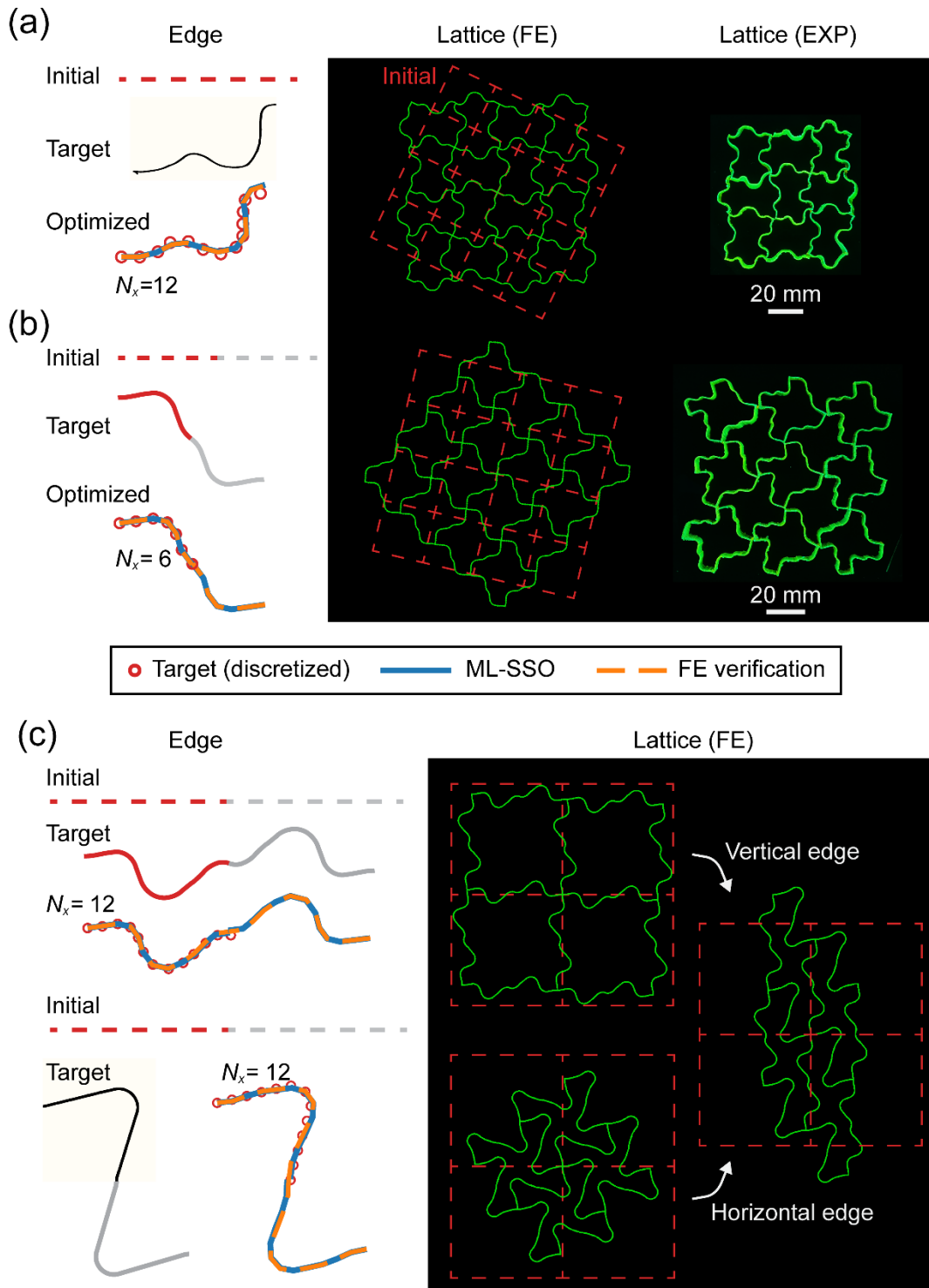


Figure 7. Applications of ML-SSO to rapid design for active lattice structures. (a) Arbitrary

edge design with half of the hand-drawn mountain as a target. (b) Anti-symmetric edge design with a simple wavy target. (c) Anti-symmetric edge design with two more complex target shapes. In (a-c), the dashed lines represent the initial shapes of edges or lattices before actuation.

First, we consider an arbitrary target edge shape. The material distribution is designed to satisfy 4-fold rotational symmetry about each joint to prevent mismatches in structural deformation. Due to the size limit of the printing stage, we use half of the hand-drawn mountain as a target ($N_x=12$). We adapt its optimized design for the lattice structure, validate the design through FE simulations, and then 4D print the lattice structure (**Figure 7a**). The FE results show that all edges of the lattice transform to the desired shape. In experiments, we only print individual square lattices due to the size limit. These printed lattices, initially square, transform into the target shape upon actuation. When tessellated, the structure matches with the FE result. The tessellation is used solely for improved visualization and does not affect the actuated shape.

Note that for an arbitrary target shape considered above, four (2×2) lattice cells form a periodic unit. Next, we consider anti-symmetric target edge shapes, which are common in lattice structures, such that a single lattice cell would be periodic. In this case, a translational symmetry in the material distribution is satisfied, ensuring the structural compatibility. Due to the anti-symmetry of the edge, the design is carried out on half of the domain. For a simple wavy target ($N_x=6$ on its half), we perform the inverse design and 4D printing. The initially square lattices, when actuated and tessellated, agree excellently with the FE predictions (**Figures 7b**). Furthermore, we consider two more complex target shapes ($N_x=12$ on the half) and use them for multiple lattice structures with identical target shapes for all edges or distinct targets for horizontal and vertical edges. For all lattices, excellent agreement is achieved among the target, the optimized shape, and the actuated edge shapes of the lattice obtained from FE simulations (**Figure 7c**). These results demonstrate the high

efficacy and flexibility of our design strategy for lattice structures.

It is worth noting that a translational symmetric material distribution can be used to ensure the structural compatibility as long as the target edge shape has equal slopes at two ends, without necessarily being antisymmetric. Additionally, by employing appropriate target shapes for horizontal and vertical edges, our approach can be readily used for designing lattices with target macroscopic deformations, such as horizontal and vertical normal strains, shear strains, and/or rigid rotations. Moreover, our design strategy is also readily applicable to other lattice structures, such as triangular and hexagonal lattices. These results and discussions highlight the high flexibility of our approach in designing the active shape change of lattice structures.

3. Conclusions

We present an approach for ultrafast inverse design of 4D-printed AC beams by combining ML and SSO. An RNN ML model is rapid and accurate in forward shape predictions based on the material distributions. We then integrate ML with SSO for the inverse design of material distributions based on the target shapes. For multiple target shapes of different complexities, ML-SSO outperforms or is comparable to ML-EA in optimization accuracy while being two to three orders of magnitude faster, delivering results in mere second(s). The CV-integrated ML-SSO then demonstrates an ultrafast, streamlined design-fabrication paradigm based on hand-drawn targets. Furthermore, although the ML model is trained on a fixed voxel configuration ($N_x=24$, $N_y=4$), we present a splicing strategy that achieves accurate shape predictions for beams with varying numbers (N_x) of lengthwise voxels with no need for retraining the ML model. This strategy, when integrated with ML-SSO, enhances adaptability for ultrafast inverse designs, accommodating target shapes of diverse complexities or lengths. For highly complex target shapes, the splicing-empowered ML-

SSO, simply with $N_x > 24$, proves to be more robust and rapid than the regular ML-SSO (which requires $N_x = 24$), achieving better or similar optimization results while reducing time cost from 11.8 s to 1.6-2 s. For example, as a comparison, for the benchmark three period shape, the FE-EA method was estimated to need 227 days for the inverse design; the ML-EA achieved design in 57 min; the new ML-SSO with splicing strategy requires only 1.97 s. Finally, the highly efficient ML-SSO is employed for the inverse design of active shape changes of 4D-printed lattice structures. Our approach thus offers an intelligent design-fabrication paradigm for 4D printing of various shape-morphing AC structures.

4. Models and Methods

4.1 Finite element model for data generation

We perform FE simulations using the commercial software ABAQUS (version 2018, Simulia, Providence, RI). The FE model follows our previous work[26] and is briefly summarized here. An active beam with the left end fixed and under the plane strain conditions is considered. It has a dimension of 80 mm long \times 1 mm thick and is partitioned into $N_x \times N_y$ voxels for material assignment. Both active and passive materials are modeled as incompressible neo-Hookean solids with the same Young's modulus but different coefficients of thermal expansion (0.001 for active and 0 for passive materials). The shape change is induced by applying a 100 °C temperature increase to the entire beam, which results in a linear strain mismatch of 0.1 between the two materials. Mesh convergence study is performed and $960 \times 12 = 11520$ hybrid plane strain (CPE4H) elements are adopted. The FE model is automatically generated and run through a Python script.

To generate a dataset for ML, we focus on the voxel configuration of 24 (N_x) \times 4 (N_y), which has a large design space of 2^{96} ($\approx 7.92 \times 10^{28}$) possible material distributions. 8600

random material distributions are created, and FE simulations are used to obtain their ground-truth actuated shapes. The generated dataset is then split into training, validation, and testing datasets with fractions of 0.7, 0.15, and 0.15. Three datasets (training, validation, and testing) show similar statistical distributions[26].

To use RNN, a similar data structure for the beam deformation problem is utilized. A column (y-direction) of voxels is the input of a single step and the entire input of a material distribution is restructured into a sequence of N_x columns of voxels. Similarly, a sequence of outputs is formed by coordinates (x, y) sampled from all columns. In general, sampling points can be arbitrarily chosen from available mesh points. Here, to reduce computational cost and better leverage the sequential dependency, we choose the sampling points to be the mid-axis, rightmost mesh point of each column.

4.2 Construction and training of the RNN model

Our network architecture consists of a sequence input layer, an LSTM[27] layer, a fully connected layer, and a regression layer. The LSTM is a special type of RNN that addresses the issues of vanishing or exploding gradients presented in long sequences. The implementation, training, and testing are conducted using Matlab (2020a, MathWorks, Natick, MA). Before the training, all the input and output data are normalized through $x' = (x - \text{mean}(x)) / \text{std}(x)$, where x and x' are the raw and normalized feature values, respectively, and mean is the mean value and std is the standard deviation. The randomly generated raw inputs (numerous '1' and '0') show a mean value of 0.5 and a standard deviation of 0.5. As a result, the input state '0' and '1' become '-1' and '1' after normalization. Such normalization is found to improve the network performance. The hidden size (number of neurons of each neural layer) of LSTM is set as 50, and the LSTM layer has $200 \times (50 + N_y + 1)$ learnable parameters. LSTM can process a sequence input of any lengths, and different time steps of

LSTM share the same learnable parameters.

Two ML models are trained to predict coordinate x and y separately to better identify their respective errors. Let F_x and F_y denote the built models for x and y , respectively, we have

$$\mathbf{x} = F_x(\boldsymbol{\theta}_x; \mathbf{M}), \mathbf{y} = F_y(\boldsymbol{\theta}_y; \mathbf{M}), \quad (3)$$

where \mathbf{x} and \mathbf{y} are predicted coordinate vectors whose components, x_i and y_i , $i \in [1, N_x]$, are coordinates of sampling points, respectively. \mathbf{M} is the digital encoding of a material distribution (an $N_y \times N_x$ array composed of “1” and “0”). $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ denote the learnable parameter sets for F_x and F_y , respectively. The loss function is defined as the half-mean-squared-error between the predicted coordinates (x_i or y_i) and true coordinates (x_i^{true} or y_i^{true}) for a specific material distribution, i.e.,

$$Loss(x) = \frac{1}{2N_x} \sum_{i=1}^{N_x} (x_i - x_i^{true})^2, \quad Loss(y) = \frac{1}{2N_x} \sum_{i=1}^{N_x} (y_i - y_i^{true})^2, \quad (4)$$

for coordinate x and y , respectively. The training of ML models is to find the optimal $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ that minimize the loss values on the training set, i.e., $\min_{\boldsymbol{\theta}_x} Loss(x)$ and $\min_{\boldsymbol{\theta}_y} Loss(y)$.

The initial learning rate is set to 0.005, which decreases by multiplying a factor of $1/\sqrt{2}$ every 50 epochs. The training stops after the validation loss converges. The mini-batch size during training is set to 64 or 10% of the training set size, whichever is smaller. The adaptive moment estimation (Adam)[28] optimizer is used to train the network.

Moreover, to fully exploit the parallel computing capability of ML and improve the optimization speed, the shape evaluation process in ML-SSO is implemented in a vectorized form. In addition, we examine the time cost for ML prediction of massive (5000) data with different batch sizes, and the optimal batch size is found to be 512. Note that in ML-SSO, the number of all potential designs for each subdomain is $2^{(N_y \times N_{sub})}$. We thus choose the batch size to be 512 for designs with $N_{sub}=3$ and to be $2^{(N_y \times N_{sub})}$ for $N_{sub}=1$

and 2. These strategies exploit the ML capability to deliver ultrafast and massive predictions.

4.3 Method of gradient calculation for Figure 1d

The gradients of actuated coordinates \mathbf{x} or \mathbf{y} with respect to the voxel encoding can be written as

$$\partial\mathbf{x}/\partial\mathbf{M} \text{ or } \partial\mathbf{y}/\partial\mathbf{M} \quad (5)$$

Since \mathbf{x} and \mathbf{y} are functions of \mathbf{M} as described by (3), which involves differentiable operations only, we use automatic differentiation to evaluate the gradients in Eq.(5) through the *dlgradient* function in Matlab. As defined in Section 4.2, \mathbf{M} is an array composed of “1” and “0”. Here the component of \mathbf{M} is denoted by M_v , which represents the material encoding of voxel v , where $v \in [1, N_y \times N_x]$ represents the voxel number. The gradient values, $\partial x_i / \partial M_v$ and $\partial y_i / \partial M_v$, for $i=12$ and 24 , based on an all-passive state ($M_v=0$ for any $v \in [1, N_y \times N_x]$), are then displayed in **Figure 1d**.

4.4 Materials and 4D printing

The photocurable resin is prepared by mixing isobornyl acrylate (IOBA, Sigma-Aldrich) and aliphatic urethane diacrylate (AUD, Ebecryl 8402, Allnex, GA, USA) in a weight ratio of 1:1. Then, 1 wt% photoinitiator (Irgacure 819, Sigma-Aldrich), 0.08 wt% photo absorber (Sudan I, Sigma-Aldrich), and 0.04 wt% fluorescent dye (Solvent green 5, Orichem International Ltd., Hangzhou, Zhejiang, China) are added. The resin is thoroughly mixed before printing.

The grayscale digital light processing (g-DLP) printing technique[4] is used to print the designed structure, where the degree of curing (DoC) can be locally controlled by the

assigned light intensity. Our ML-SSO designs are transformed into grayscale printing slices, where the active (“1”) and passive (“0”) phases correspond to grayscale percentages of 0% (hence higher light intensity) and 60% (hence lower light intensity), respectively, which later spatially assign the high-DoC and low-DoC phases in the printed structure, respectively. The printed structure is then placed in an 80°C oven for 8 hours to facilitate monomer volatilization. The low-DoC phase contains more residual monomers that can volatilize and thus shows more volume shrinkage than the high-DoC phase at elevated temperatures. The shrinkage strain mismatch of the two phases induces the shape transformation.

4.5 Modification of optimal designs

Note that the material properties in experiments are different from those used in FE simulations. Experimental characterizations show that the printed two material phases show a modulus ratio of 0.06, while the ML-EA design assumes the identical modulus for two constituent phases. Additionally, the practical expansion mismatch is identified to be 0.05, which is also different from that used in ML-EA (i.e., 0.1). Such issues can be resolved by retraining the ML model based on the FE data with practical material properties (expansion mismatch and modulus difference) and re-running the ML-EA. Here, instead of retraining a new model, we adopt a design conversion strategy[26] to approximately compensate effects of property difference of the two phases on the shape change, i.e., the optimal designs are converted using the analytical curvatures of multi-layer composite beams. The effectiveness of this strategy has been validated by FE simulations and experiments, as detailed in our previous work[26].

Acknowledgments

H.J.Q. acknowledges the support of an AFOSR grant (FA9550-20-1-0306; Dr. B.-L. “Les”

Lee, Program Manager) and a gift fund from HP, Inc.

References

1. Ge, Q., H.J. Qi, and M.L. Dunn, *Active materials by four-dimension printing*. Applied Physics Letters, 2013. **103**(13): p. 131901.
2. Kuang, X., et al., *Grayscale digital light processing 3D printing for highly functionally graded materials*. Science Advances, 2019. **5**(5): p. eaav5790.
3. Cheng, J., et al., *Centrifugal multimaterial 3D printing of multifunctional heterogeneous objects*. Nature Communications, 2022. **13**(1): p. 7931.
4. Yue, L., et al., *Single-vat single-cure grayscale digital light processing 3D printing of materials with large property difference and high stretchability*. Nature Communications, 2023. **14**(1): p. 1251.
5. Yue, L., et al., *Cold-programmed shape-morphing structures based on grayscale digital light processing 4D printing*. Nature Communications, 2023. **14**(1): p. 5519.
6. Demoly, F., et al., *The status, barriers, challenges, and future in design for 4D printing*. Materials & Design, 2021. **212**: p. 110193.
7. Sun, X., et al., *Perspective: Machine learning in design for 3D/4D printing*. Journal of Applied Mechanics, 2024. **91**(3): p. 030801.
8. Maute, K., et al., *Level Set Topology Optimization of Printed Active Composites*. Journal of Mechanical Design, 2015. **137**(11): p. 111402.
9. Geiss, M.J., et al., *Combined Level-Set-XFEM-Density Topology Optimization of Four-Dimensional Printed Structures Undergoing Large Deformation*. Journal of Mechanical Design, 2019. **141**(5).
10. Tanaka, M., et al., *Turing pattern-based design and fabrication of inflatable shape-morphing structures*. Science Advances, 2023. **9**(6): p. eade4381.
11. Zolfagharian, A., et al., *Topology-Optimized 4D Printing of a Soft Actuator*. Acta Mechanica Solida Sinica, 2020. **33**(3): p. 418-430.
12. Hamel, C.M., et al., *Machine-learning based design of active composite structures for 4D printing*. Smart Materials and Structures, 2019. **28**(6): p. 065005.
13. Wu, S., et al., *Evolutionary Algorithm-Guided Voxel-Encoding Printing of Functional Hard-Magnetic Soft Active Materials*. Advanced Intelligent Systems, 2020. **2**(8): p. 2000060.
14. Athinarayanarao, D., et al., *Computational design for 4D printing of topology optimized multi-material active composites*. npj Computational Materials, 2023. **9**(1): p. 1.
15. Sossou, G., et al., *Design for 4D printing: Modeling and computation of smart materials distributions*. Materials & Design, 2019. **181**: p. 108074.
16. Wang, L., et al., *Evolutionary design of magnetic soft continuum robots*.

- Proceedings of the National Academy of Sciences, 2021. **118**(21): p. e2021922118.
17. Guo, K., et al., *Artificial intelligence and machine learning in design of mechanical materials*. Materials Horizons, 2021. **8**(4): p. 1153-1172.
 18. Gu, G.X., et al., *Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment*. Materials Horizons, 2018. **5**(5): p. 939-945.
 19. Chen, C.-T. and G.X. Gu, *Effect of Constituent Materials on Composite Performance: Exploring Design Strategies via Machine Learning*. Advanced Theory and Simulations, 2019. **2**(6): p. 1900056.
 20. Yang, Z., C.-H. Yu, and M.J. Buehler, *Deep learning model to predict complex stress and strain fields in hierarchical composites*. Science Advances, 2021. **7**(15): p. eabd7416.
 21. Wilt, J.K., C. Yang, and G.X. Gu, *Accelerating Auxetic Metamaterial Design with Deep Learning*. Advanced Engineering Materials, 2020. **22**(5): p. 1901266.
 22. Zolfagharian, A., et al., *4D printing soft robots guided by machine learning and finite element models*. Sensors and Actuators A: Physical, 2021. **328**: p. 112774.
 23. Roach, D.J., et al., *Utilizing computer vision and artificial intelligence algorithms to predict and design the mechanical compression response of direct ink write 3D printed foam replacement structures*. Additive Manufacturing, 2021. **41**: p. 101950.
 24. Rawat, S. and M. Shen, *A novel topology design approach using an integrated deep learning network architecture*. arXiv preprint arXiv:1808.02334, 2018.
 25. Zhang, Z. and G.X. Gu, *Finite-Element-Based Deep-Learning Model for Deformation Behavior of Digital Materials*. Advanced Theory and Simulations, 2020. **3**(7): p. 2000031.
 26. Sun, X., et al., *Machine Learning-Evolutionary Algorithm Enabled Design for 4D-Printed Active Composite Structures*. Advanced Functional Materials, 2022. **n/a**(n/a): p. 2109805.
 27. Hochreiter, S. and J. Schmidhuber, *Long Short-Term Memory*. Neural Computation, 1997. **9**(8): p. 1735-1780.
 28. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.