

# Inverse Design of Inflatable Soft Membranes Through Machine Learning

Antonio Elia Forte,\* Paul Z. Hanakata, Lishuai Jin, Emilia Zari, Ahmad Zareei, Matheus C. Fernandes, Laura Sumner, Jonathan Alvarez, and Katia Bertoldi\*

Across fields of science, researchers have increasingly focused on designing soft devices that can shape-morph to achieve functionality. However, identifying a rest shape that leads to a target 3D shape upon actuation is a non-trivial task that involves inverse design capabilities. In this study, a simple and efficient platform is presented to design pre-programmed 3D shapes starting from 2D planar composite membranes. By training neural networks with a small set of finite element simulations, the authors are able to obtain both the optimal design for a pixelated 2D elastomeric membrane and the inflation pressure required for it to morph into a target shape. The proposed method has potential to be employed at multiple scales and for different applications. As an example, it is shown how these inversely designed membranes can be used for mechanotherapy applications, by stimulating certain areas while avoiding prescribed locations.

elaborate shapes via folding,<sup>[7–9]</sup> morphable sheets have been realized by combining materials that can expand and contract by different amounts in response to external stimuli such as temperature, humidity, or pH;<sup>[4,10]</sup> inflating membranes reinforced with stiff components have shown promise for the realization of shape changing surfaces.<sup>[3,6,11–14]</sup> Focusing specifically on inflatable membranes, these are either made in a complex deflated shape and out of a single homogeneous material<sup>[3,11–15]</sup> or in a simple rest shape by optimizing the material locally to guide the deformation upon inflation.<sup>[6,16,17]</sup> However, irrespective of their fabrication method, programming 2D sheets to obtain a target 3D shape is a non-trivial task that typically requires the

## 1. Introduction

2D sheets that can morph from flat into 3D shapes have become a powerful and versatile platform to realize deployable systems,<sup>[1–3]</sup> frequency shifting antennae,<sup>[4]</sup> active building facades,<sup>[5]</sup> as well as camouflage devices.<sup>[6]</sup> Several avenues have been successful in achieving shape changing capabilities. Origami principles have enabled transformation of flat sheets into

use of optimization algorithms.<sup>[16,18–21]</sup> These include gradient-free algorithms<sup>[16,22]</sup> as well as gradient-based methods.<sup>[18,21]</sup>

Here, we consider inflatable membranes comprising soft and stiff domains and show how machine learning tools can be used to design configurations of such domain that result in target shapes upon inflation. While machine learning methods have been classically employed for image recognition<sup>[23]</sup> and language processing,<sup>[24]</sup> they have also recently emerged as powerful tools to solve mechanics problems.<sup>[25–38]</sup> Building on these recent successes, we demonstrate that these tools can be extended to study the nonlinear mechanics of inflatable systems. By using neural networks (NNs) trained on finite element (FE) simulations, we are able to solve the inverse design problem. This allows us to prescribe a target 3D shape in input and obtain a spatially defined 2D design for a soft membrane, comprising soft and stiff elastomeric pixels, as output. Such a designed membrane is then inflated to an optimal pressure—also instructed by the model—and morphs into the desired shape. The platform hereby introduced, despite being presented in a macroscale framework, is highly scalable and holds potential for many fields of science and engineering, enabling applications such as morphable surfaces for architecture, soft sensors, ergonomic garments, and medical devices. As an example, we show how these membranes can be used in mechanotherapy for wound healing.


A. E. Forte, L. Jin, E. Zari, A. Zareei, M. C. Fernandes, J. Alvarez, K. Bertoldi  
J.A. Paulson School of Engineering and Applied Sciences  
Harvard University  
Cambridge, MA 02138, USA  
E-mail: antonio.forte@kcl.ac.uk; bertoldi@seas.harvard.edu

A. E. Forte, E. Zari  
Department of Electronics  
Information and Bioengineering  
Politecnico di Milano  
Milan 20133, Italy

A. E. Forte  
Department of Engineering  
King's College London  
London WC2R 2LS, UK

P. Z. Hanakata  
Department of Physics  
Harvard University  
Cambridge, MA 02138, USA

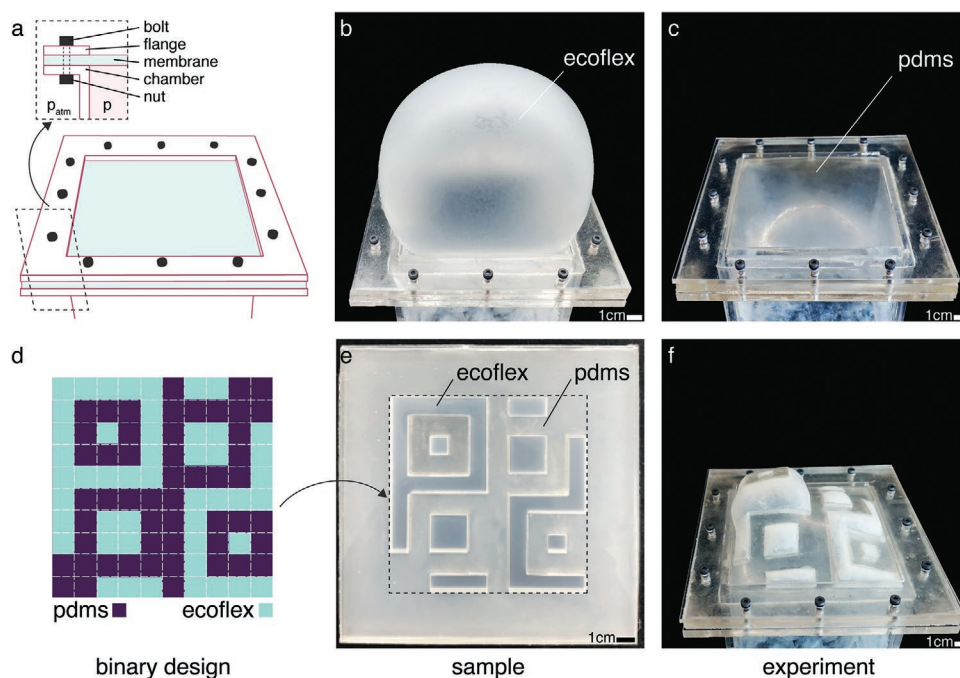
L. Sumner  
Independent researcher

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/adfm.202111610>.

DOI: 10.1002/adfm.202111610

## 2. Our Platform

Our platform consists of a square sheet of elastomeric material with edges of 10 cm mounted on an acrylic chamber which



**Figure 1.** Bi-material membrane unlocks complex deformations. a) Inflation set up: the membrane (in cyan) is clamped between the acrylic chamber's edge underneath and the square flange aligned on top, using bolt and nuts. b) A flat membrane made out of Ecoflex morphs in a simple spherical shape upon inflation. c) If the same membrane is made out of a stiffer material (i.e. PDMS), the deformation is simply scaled down. d) We use a regular grid to divide the membrane's domain into subdomains named pixels, to which either material can be assigned, creating a binary design. e) The membrane can be fabricated as a continuous material. f) Upon inflation, the bi-material membrane assumes complex deformation, depending on the mutual position of stiff and soft pixels.

is pressurized (Figure 1a). As expected, when pressurized, the elastomeric membrane deforms out-of-plane achieving a dome-like shape with a height that depends both on the stiffness of the material and the thickness of the sheet. For example, under a pressure  $p = 1.7$  KPa, a membrane made out of Ecoflex (Ecoflex 00-30 with initial shear modulus  $\mu_{\text{Eco}} = 0.023$  MPa) with thickness  $h_{\text{Eco}} = 1$  mm and initial flexural rigidity of  $D_{\text{Eco}} = \frac{E_{\text{Eco}} h_{\text{Eco}}^3}{12(1-\nu_{\text{Eco}}^2)} = 7.67 \times 10^{-6}$  Pa · m<sup>3</sup> (where  $E_{\text{Eco}} = 2\mu_{\text{Eco}}(1 + \nu_{\text{Eco}})$  and  $\nu_{\text{Eco}} = 0.5$  are the Young's modulus and Poisson's ratio of Ecoflex) reaches a height of about 9 cm (Figure 1b). Differently, a membrane with thickness  $h_{\text{PDMS}} = 7$  mm made out of PDMS (SYLGARD 184 with initial shear modulus  $\mu_{\text{PDMS}} = 0.85$  MPa), for which the initial flexural rigidity is  $D_{\text{PDMS}} = 9.72 \times 10^{-2}$  Pa · m<sup>3</sup>, reaches a height about 1 cm when subjected to  $p = 10$  KPa (Figure 1c).

Whereas homogeneous membranes always lead to dome-like shapes, it has been shown that the range of achievable shapes for both elastomeric membranes and tubes can be enriched by incorporating stiffer components, such as fibers and sheets.<sup>[6,16,39–41]</sup> Therefore, to achieve more complex configurations, we realize our membranes out of a combination of stiff and flexible pixels. To demonstrate the concept, we partition the membrane with a 10×10 array of squares (all with edges of 1 cm) and assign to each pixel either a 7 mm thick layer of PDMS or a 1 mm thick layer of Ecoflex (Figure 1d). Such membranes are manufactured using a multistep molding procedure. First, we create a mold (with depth of 7 mm) with the negative shape of the stiff pixels of the binary design and fill it with PDMS. Then,

before the PDMS is completely cured (after 1.5 h), we remove 1 mm acrylic sheet from the mold to leave behind a 1 mm deep pocket corresponding to the soft pixels that we fill with Ecoflex (see Video S1, Supporting Information). Note that after curing a continuous membrane is obtained, due to the fact that the two elastomeric networks bind upon contact while curing (Section S1, Supporting Information Appendix). In Figure 1e, we show the fabricated membrane corresponding to the binary design reported in Figure 1d. As shown in Figure 1f for  $p = 5$  kPa, upon inflation, this membrane undergoes a complex transformation, bulging out of plane in a nonintuitive fashion. The non-linear behaviour of such a membrane is strongly governed by the location of the soft and stiff pixels on the square grid and their interactions. As such, understanding the relation between binary design and the resultant 3D inflated shape is non-trivial and requires an efficient inverse design strategy.

### 3. Inverse Design via Neural Networks

Inspired by recent works that have successfully employed machine learning methods to inverse-design complex physical systems, including mechanical<sup>[26,42]</sup> and optical<sup>[43]</sup> metamaterials as well as chemical compounds,<sup>[44,45]</sup> we employ fully connected NNs to identify binary design of pixelated membranes and associated pressure levels at which such membranes should be inflated to reach a target 3D shape.

To efficiently generate the large amounts of data for the NNs to be trained on, we conduct non-linear FE simulations within

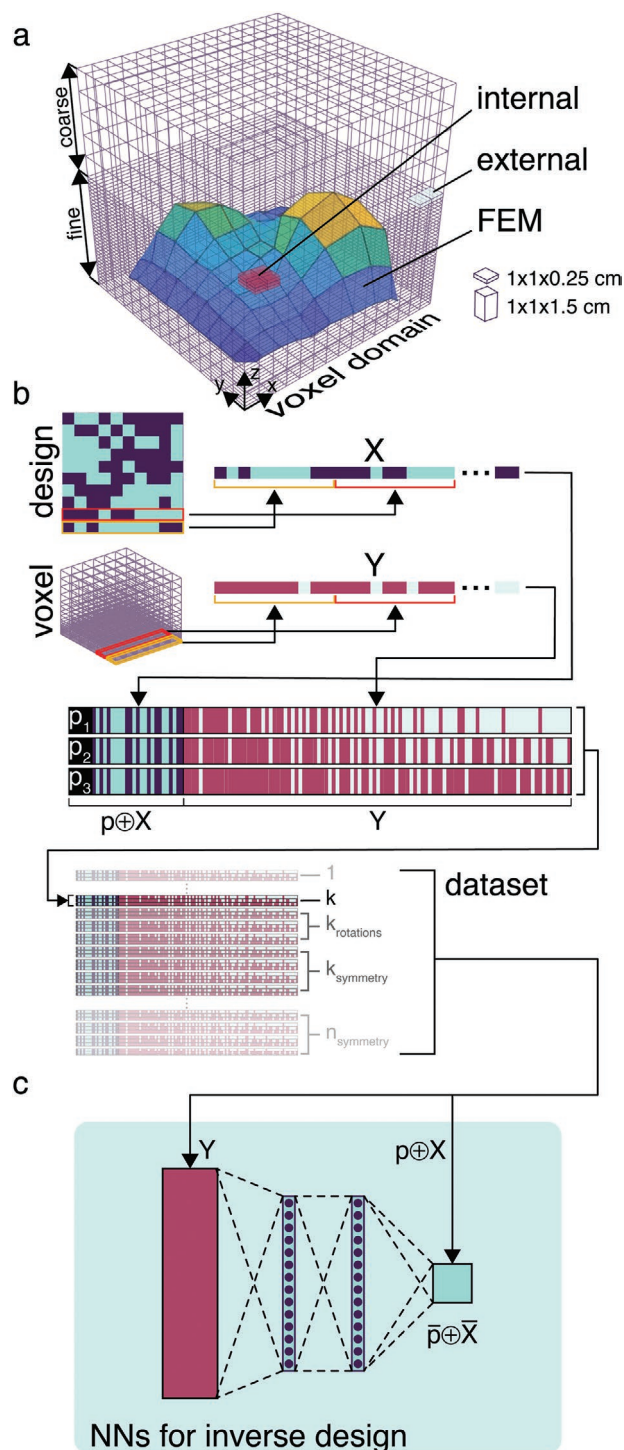
ABAQUS 2019/Standard. In all our simulations, we discretize the PDMS and Ecoflex pixels with four-node general-purpose shell elements (S4R element type) and four-node membrane elements (M3D4 element type), respectively. Further, we i) model the response of both elastomers using an incompressible Gent material, ii) fix all nodes located on the four edges of the models, iii) apply a pressure  $p$  (with  $p \in [0, 3.5]$  kPa) directly on the bottom surface, and iv) solve for the deformation using the dynamic implicit solver, while monitoring the kinetic energy to ensure quasi-static conditions. We then export the deformed configurations of the membrane at  $p = 1.5, 2.5,$  and  $3.5$  kPa and use the method of voxelization<sup>[46]</sup> to represent them. Specifically, we start with a rectangular cube with dimensions  $15 \times 15 \times 15$  cm, which contains all inflated membranes for the three pressure levels considered and split the domain into smaller cubes known as voxels (Figure 2a). We then assign a value of 1 if the voxel's centroid falls below the inflated membrane (Figure 2a, internal pixels), and 0 otherwise (Figure 2a, external pixels). Since the height of the majority of the inflated designs is lower than 8 cm, we use voxels with dimensions  $1 \times 1 \times 0.25$  cm to cover the height up to  $z = 8$  cm and  $1 \times 1 \times 1.5$  cm for  $z > 8$  cm (Figure 2a). Further, since the corners of the considered rectangular cube are never reached by the membranes upon inflation, to reduce the size of our domain, we remove the corresponding voxels from the analysis, resulting in 9220 voxels.

Next, we flatten our  $10 \times 10$  binary designs onto a 100-dimensional vector  $\mathbf{X}$  containing 0s and 1s in correspondence to the soft and stiff pixels, respectively, and the numerically obtained inflated shape at the three considered levels of pressure onto three 9220-dimensional vectors  $\mathbf{Y}$ s containing 0s and 1s in correspondence to the internal and external voxels, respectively (Figure 2b). We then train the NNs to perform a mapping  $\mathbf{Y} \rightarrow \bar{p} \oplus \bar{\mathbf{X}}$ , where the overbar is used to represent the quantities predicted by the NNs and  $\oplus$  denotes concatenation (Figure 2c). In this study, we use fully connected NNs with two hidden layers having identical number of neurons,  $N_{\text{neu}}$ , and iteratively update the neuron weights and biases to make the output conform to the true  $\mathbf{X}$  and  $p$  by minimizing<sup>[26]</sup>

$$\mathcal{L}_{\text{NN}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} |\mathbf{X}_i - \bar{\mathbf{X}}_i|^2 + \lambda |p_i - \bar{p}_i|^2 \quad (1)$$

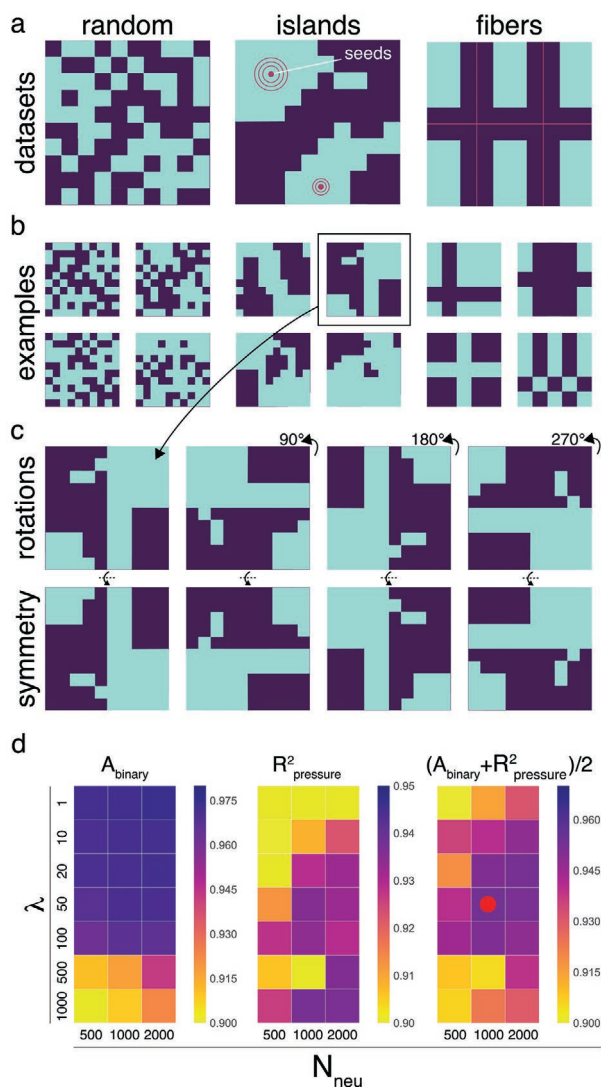
where  $N_{\text{train}}$  is the number of training datapoints and  $\lambda$  is an adjustable hyperparameter that controls the relative weight between the mean squared distance in  $\mathbf{X}$  and  $p$ . Note that, in order to obtain binary values for  $\bar{\mathbf{X}}$ , we employ a sigmoid function as an output filter on the output layer. Additional details about the NNs' hyperparameters (e.g., learning rate) are reported in Section S5, Supporting Information Appendix.

In order to obtain accurate predictions in the inverse design problem, the NNs need i) to have enough training data to capture the whole design space and ii) an optimized architecture (i.e., optimal  $N_{\text{neu}}$  and  $\lambda$ ). The most common strategy to generate large training datasets is based on randomly generated data (in our case, this translates to generating pixelated membrane designs where the locations for the soft and stiff pixels are randomly assigned).<sup>[25,28]</sup> However, this approach would require an extremely large number of simulations, since it would be



**Figure 2.** Data voxelization, processing, and structuring. a) The inflated shape from each FEM simulation is mapped into a voxelated domain comprising a fine and a coarse region. The voxels whose centroid falls below the membrane are identified as internal and external otherwise. b) The membrane's binary design and the voxel domain are flattened into two 1D binary arrays named  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively, and concatenated with the corresponding pressure level ( $p_1, p_2$  or  $p_3$ ), producing three arrays per simulation. The binary design can be rotated and mirrored augmenting the number of datapoints eightfold. c) The  $\mathbf{Y}$  and  $p \oplus \mathbf{X}$  matrices are fed to the NNs in the input and output layers, respectively.





**Figure 3.** Dataset classes and hyperparameter search. a) To train the NNs, we use three classes of designs: random, islands, and fibers. b) Examples from the three classes are reported to appreciate their topological characteristics. c) Each design can be rotated and mirrored to generate eight different datapoints. d) Effect of the number of neurons in the NNs layers,  $N_{\text{neu}}$ , and the hyperparameter  $\lambda$  on  $A_{\text{binary}}$ ,  $R^2_{\text{pressure}}$ , and  $(A_{\text{binary}} + R^2_{\text{pressure}})/2$ . The red marker indicates the combination of these two parameters that gives the best average accuracy between the two metrics  $((A_{\text{binary}} + R^2_{\text{pressure}})/2)$ .

rare to sample data points containing large clusters of soft/stiff pixels or lines of soft/stiff pixels that are known to have a profound effect on the mechanical behavior of the membranes.<sup>[6]</sup> Therefore, guided by previous studies showing that the performances of NNs improve when the model is trained with diverse (non-redundant) datasets,<sup>[47–52]</sup> we adopt three different strategies to generate 2D pixelated binary designs whose inflated shapes predicted via FE are used to train the NNs. Specifically, we simulate via FE the behavior upon inflation of i) 2500 membranes in which we randomly assign a value of 0 or 1 to each pixel (Figure 3a, left); ii) 2500 membranes in which a few (1–15) pixels (“seeds”) located at random positions are allowed to grow

in all directions and convert the neighbor pixels from stiff to soft until a critical ratio of soft pixels to all pixels is reached,<sup>[38]</sup> effectively creating “islands” of soft pixels (Figure 3a, center); iii) 2500 membranes realized employing logical operators to combine row and column vectors of stiff pixels running from and to opposite edges, to create features resembling fibers (Figure 3a, right). We refer to these three datasets as i) random, ii) islands, and iii) fibers and report design examples for each dataset in Figure 3b. Note that, although for each dataset we simulate 2500 membranes, we are able to create a total of 60 000 datapoints per dataset since i) each of these designs can be rotated and mirrored seven times (see Figure 3c) and ii) for each of them, we export the inflated configuration at three different levels of pressure. Additional details about the algorithms used to generate the pixelated membranes are reported in Section S4, Supporting Information Appendix.

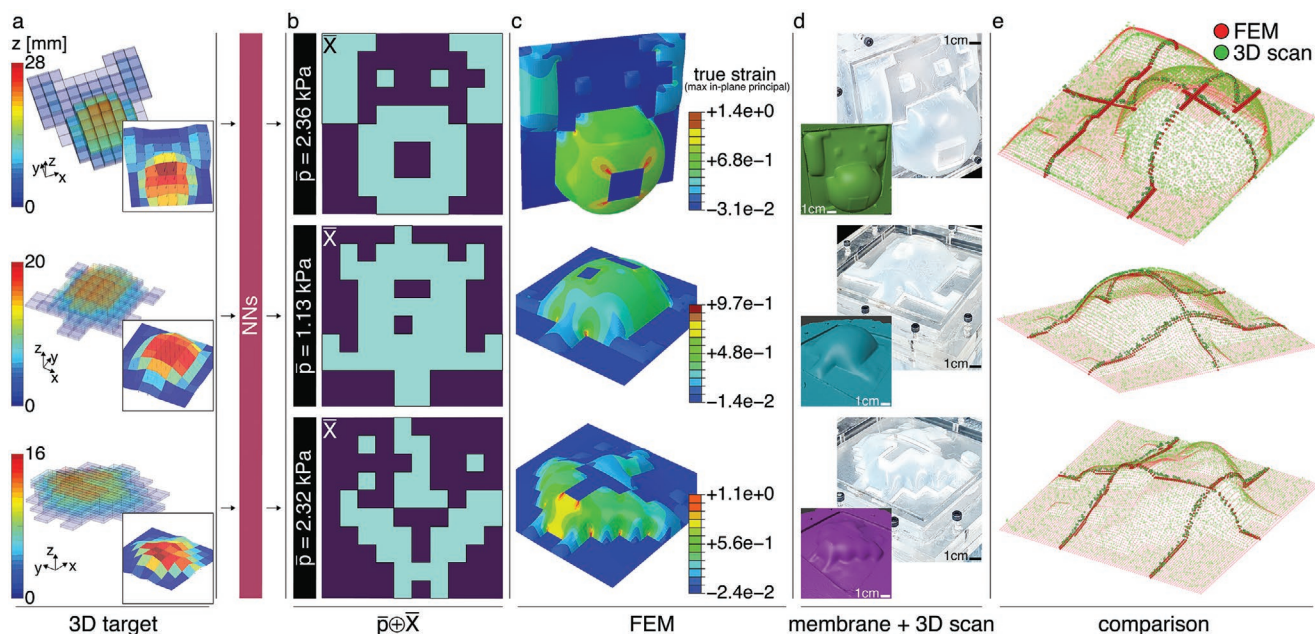
To optimize our NNs, we vary the number of neurons  $N_{\text{neu}}$  and the hyperparameter  $\lambda$ . We first merge the generated 180 000 datapoints into a single dataset. We then use 80% of such dataset to run multiple training sessions, and for each session, we vary the number of neurons  $N_{\text{neu}}$  and the hyperparameter  $\lambda$ . To attest the performance of each trained model, we use 10% of the datapoints as validation set and the remaining as test set, and introduce two metrics: an accuracy on the predicted binary design,  $A_{\text{binary}}$ , and an accuracy on the predicted pressure level,  $R^2_{\text{pressure}}$ . Specifically,  $A_{\text{binary}}$  is evaluated by counting the number of correctly identified pixels and therefore defined as

$$A_{\text{binary}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \frac{N_{\text{correct}}^i}{N_{\text{pixels}}} \quad (2)$$

where  $N_{\text{correct}}^i$  is the total number of correctly predicted pixels for the  $i$ -th binary design,  $N_{\text{pixels}}$  is the total number of pixels in the membrane (i.e., 100), and  $N_{\text{test}}$  is the number of datapoints used for testing (i.e., 18 000). Differently, since the pressure is a continuous variable, we define the accuracy on the predicted pressure level  $\bar{p}$  as

$$R^2_{\text{pressure}} = 1 - \frac{\sum_{i=1}^{N_{\text{test}}} |p_i - \bar{p}|^2}{\sum_{i=1}^{N_{\text{test}}} |p_i - \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} p_i|^2} \quad (3)$$

By systematically investigating the effect of  $\lambda$  and  $N_{\text{neu}}$  on the two metrics, we find that larger values for  $N_{\text{neu}}$  generally lead to a better test accuracy as the model capacity is increased (Figure 3d). Note that, since increasing the number of neurons often leads to overfitting,<sup>[53–55]</sup> we use early stopping rules<sup>[53]</sup> to determine how many iterations can be run before the NNs begin to over-fit. An increase in  $\lambda$  results in larger  $R^2_{\text{pressure}}$ , but lower  $A_{\text{binary}}$  (Figure 3d). Hence, we use the average accuracy between the two metrics,  $(A_{\text{binary}} + R^2_{\text{pressure}})/2$ , to determine the optimized NN architecture, which we find to be characterized by  $N_{\text{neu}} = 1000$  and  $\lambda = 50$ —values that are fixed for the next analyses. Additionally, it is worth noticing that the NNs trained on a combination of all datapoints outperform the same model trained on a single class of data, confirming that a diverse dataset leads to better performances (more information on NNs’ performances with different training are reported in Section 5, Supporting Information Appendix).



**Figure 4.** Inverse design of target 3D shapes. a) Target 3D shapes that resemble a dog face (top), a turtle (center), and a manta ray (bottom) are fed in the NNs. b) The NNs provide optimal inflation pressure,  $\bar{p}$ , and binary designs,  $\bar{\chi}$ , as outputs. c) The binary designs are inflated at the corresponding pressure through FE. The colors indicate maximum in-plane principal true strains. d) The designs are fabricated, inflated at the corresponding pressure, and 3D-scanned. e) The clouds of points from the FE simulations (red dots) and the 3D-scans (green dots) are overlapped and compared. Solid markers are highlighted along cutting planes to better show the overlapping.

#### 4. Inverse Design of Soft Membranes

Having verified the accuracy of the NNs on a test set of unseen designs generated through three different algorithms, we then employ them to inverse-design target 3D shapes. Specifically, we feed a 3D shape as input to our trained NNs and obtain as output a 2D binary design for the soft membrane, along with the pressure necessary to reach the target shape upon inflation. To demonstrate the process, we select shapes that resemble a dog face, a turtle, and a manta ray (Figure 4a). Each target shape is flattened onto a 9220-dimensional vector,  $\mathbf{Y}^{target}$ , which is fed into the NNs. As output, for each design we obtain a pixelated membrane design  $\bar{\chi}$  and inflation pressure  $\bar{p}$  (Figure 4b).

To measure the accuracy of the designs identified by the NNs, we start by employing FE to simulate the inflation of the binary design  $\bar{\chi}$  until the pressure reaches  $\bar{p}$ . As shown in Figure 4c, the numerically obtained inflated shapes qualitatively match the 3D target ones. To better quantify the similarity between the two sets of shapes, we flatten each numerically obtained shape into a vector  $\mathbf{Y}^{FE}$  and compare all its internal voxels to the corresponding one in  $\mathbf{Y}^{target}$ . We then use the ratio between the number of voxels correctly predicted,  $N_{correct}$ , and the number of internal voxels in the target shape,  $N_{target}$  as the accuracy metric

$$A_{membrane} = \frac{N_{correct}}{N_{target}} \quad (4)$$

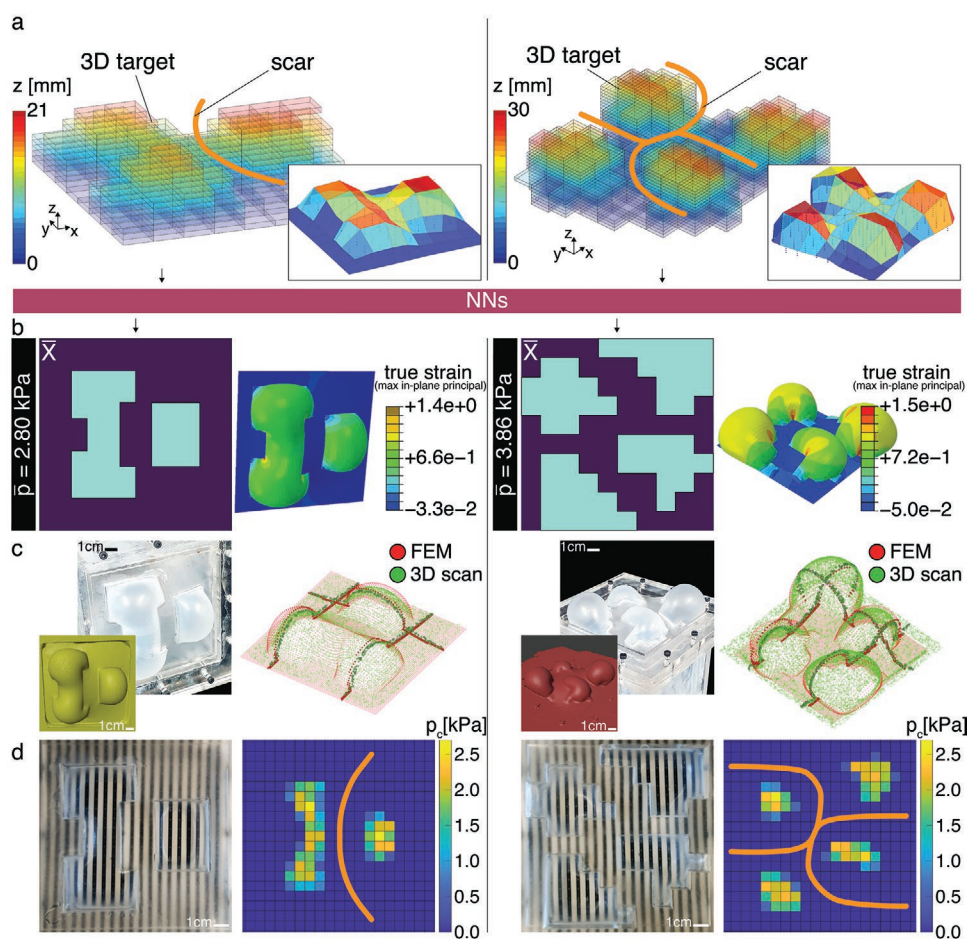
Using Equation (4), we obtain  $A_{membrane} = 0.975, 0.945, 0.996$  for the dog face, turtle, and manta ray, respectively. Such high accuracy values are indicative of effective NNs that provide

pixelated designs and pressures leading to 3D shapes extremely close to those targeted.

Next, we physically fabricate the designs identified by the NNs. In Figure 4d, we report snapshots of the physical membranes inflated to the optimal pressure provided by the NNs. As is noticeable, despite the unavoidable imperfections introduced during fabrication and testing, the 3D shapes obtained upon inflation are clearly recognizable. To compare such shapes to those obtained via FE simulations, we use a hand-held 3D scanner (Artec Space Spider, Artec Studio 14.1.1.75) and record the experimentally obtained surface profiles at  $\bar{p}$  (insets in Figure 4d). As shown in Figure 4e, we find excellent agreement between the numerically predicted and experimentally obtained inflated shapes, confirming the validity of our approach: (more information on the experiments are reported in Section S2 and S3, Supporting Information Appendix).

Having demonstrated that our NNs, trained with a combination of three different datasets, can be used to identify soft membranes capable of mimicking target 3D shapes upon inflation, we then explore how these can be harnessed for applications. Specifically, since it is known that the application of compressive loading around a wound site can reduce healing time and formation of hypertrophic scars,<sup>[56,57]</sup> we design soft membranes that apply pressure in targeted areas while avoiding contact with sensitive locations.

To demonstrate our approach, we focus on the two scar profiles highlighted in orange in Figure 5a and aim at realizing membranes that upon inflation have their maximum elevation (along the  $z$  direction) in the areas surrounding the scars and minimum elevation in the areas where the scars lie. We expect such membranes to apply compressive loading to the region



**Figure 5.** Inverse design of target 3D shapes for mechanotherapy: a) Target 3D-shapes that can stimulate the tissue around pre-defined scar profiles during inflation are fed in the NNs. b) The NNs provide optimal inflation pressure,  $\bar{p}$ , and binary designs,  $\bar{X}$ , as outputs, which are inflated at the corresponding pressure through FE. Numerical snapshots of the inflated membranes are shown, with the color indicating the maximum in-plane principal true strain. c) The designs are fabricated, inflated at the corresponding pressure, and 3D-scanned. The clouds of points from the FE simulations (red dots) and the 3D-scans (green dots) are overlapped and compared. Solid markers are highlighted along cutting planes to better show the overlapping. d) The membranes are fixed upside down at a 15 mm distance from a pressure mat and imaged from the top through a transparent pressure chamber. The measured contact pressures are reported, along with their locations and the pre-defined scar profile.

around the wound when inflated against the skin and, therefore, to promote healing. To obtain pixelated membrane designs resulting in the target shapes shown in Figure 5a upon inflation, we flatten their voxelated shapes onto 9220-dimensional vectors  $Y^{\text{target}}$  and feed them into the trained NNs. As for the membranes shown in Figure 3, we then use FE to simulate the behavior of the designs identified by our NNs and find very good agreement between the target and numerically simulated inflated shapes (Figure 5b) testified by  $A_{\text{membrane}}=0.949$  and 0.932 for the  $c$  and  $\chi$ -scars, respectively. Further, we build the physical membranes, and also in this case found that they nicely match the target shapes (Figure 5c). Finally, to evaluate the pressure locally applied by the inflated membranes around the two considered scars, we position the deflated membranes at 15 mm from a pressure mat (Tekscan - Model 5250). When inflated at  $\bar{p}$ , the membranes come in contact with the pressure mat which records the locally applied contact pressure,  $p_c$ . In Figure 5d, we show the top view of the inflated membrane pushing against the mat (visible through the transparent

pressure chamber) as well as the recorded pressure distribution overlaid with the scar profiles. As clearly visible, the inflated 3D shapes optimized through our NNs are able to apply pressure around the prescribed areas without touching the scars.

## 5. Conclusion

In summary, to realize membranes that can morph into pre-programmed shapes upon inflation, we have employed NNs that are trained to identify a pixelated membrane design and inflation pressure leading to the desired 3D shape. The data required to train the NNs were obtained by simulating the membrane inflation through FE, and to guarantee the creation of a diverse dataset, three different algorithms have been used to produce pixelated designs. We have then employed our trained NNs to inverse-design a few user generated 3D shapes and showed how such platform could be used to create patient-specific devices for mechanotherapy routines where it



is important to stimulate the tissue around prescribed areas (scars) to enhance healing and reduce recovery time.

Despite having presented results at the centimeter scale, our methodology is scale independent and can benefit a range of applications where having an inverse-design strategy could facilitate and improve the design process itself. Examples might include ergonomic designing, patient-specific medical devices, architectural components, and shape-morphing acoustic devices. Additionally, we have shown that our NNs trained on only 7500 forward FEM simulations can successfully solve an inverse problem with  $2^{100}$  possible designs. This reinforces previous findings which identified machine learning methods as a valuable complementary tool to established mechanical approaches.<sup>[25–27,29–31,36]</sup> The performance of our model can be further improved by applying convolutional neural networks (CNNs) as the filters and pooling layers are efficient in capturing spatial correlation and locality in a sparse data.<sup>[25]</sup> In particular, 3D CNNs, which are widely used for point cloud labeling in computer vision,<sup>[58,59]</sup> would be ideal to handle the voxels of arbitrary 3D shapes. Moreover, recent deep learning methods such as conditional generative adversarial neural networks, with image-to-image translation capabilities,<sup>[60–62]</sup> could also be employed to solve inverse-design problems similar to the one hereby described.

## Supporting Information

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

## Acknowledgements

This research was supported by the NSF grants DMR-2011754, DMREF-1922321, and OAC-2118201. P.Z.H. acknowledges support through the NSF grant DMR-1608501. P.Z.H. thanks Ekin D. Cubuk for helpful discussions.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

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

## Keywords

inverse design, machine learning, membranes, shape morphing, soft matter

Received: November 14, 2021

Revised: December 9, 2021

Published online:

- [1] S. J. Callens, A. A. Zadpoor, *Mater. Today* **2018**, *21*, 241.
- [2] G. P. Choi, L. H. Dudte, L. Mahadevan, *Nat. Mater.* **2019**, *18*, 999.
- [3] E. Siéfert, E. Reyssat, J. Bico, B. Roman, *Nat. Mater.* **2019**, *18*, 24.
- [4] J. W. Boley, W. M. van Rees, C. Lissandrello, M. N. Horenstein, R. L. Truby, A. Kotikian, J. A. Lewis, L. Mahadevan, *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 20856.
- [5] L. Tomholt, O. Geletina, J. Alvarenga, A. V. Shneidman, J. C. Weaver, M. C. Fernandes, S. A. Mota, M. Bechthold, J. Aizenberg, *Energy Build.* **2020**, *226*, 110377.
- [6] J. Pikul, S. Li, H. Bai, R. Hanlon, I. Cohen, R. Shepherd, *Science* **2017**, *358*, 210.
- [7] L. H. Dudte, E. Vouga, T. Tachi, L. Mahadevan, *Nat. Mater.* **2016**, *15*, 583.
- [8] L. H. Dudte, G. P. Choi, L. Mahadevan, *Proc. Natl. Acad. Sci. USA* **2021**, *118*, 21.
- [9] S. Felton, M. Tolley, E. Demaine, D. Rus, R. Wood, *Science* **2014**, *345*, 644.
- [10] J. Kim, J. A. Hanna, M. Byun, C. D. Santangelo, R. C. Hayward, *Science* **2012**, *335*, 1201.
- [11] E. Siéfert, E. Reyssat, J. Bico, B. Roman, *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 16692.
- [12] E. Siéfert, J. Bico, E. Reyssat, B. Roman, *J. Mech. Phys. Solids* **2020**, *143*, 104068.
- [13] T. Gao, E. Siéfert, A. DeSimone, B. Roman, *Adv. Mater.* **2020**, *32*, 2004515.
- [14] E. Siéfert, E. Reyssat, J. Bico, B. Roman, *Soft Matter* **2020**, *16*, 7898.
- [15] M. Skouras, B. Thomaszewski, B. Bickel, M. Gross, in *Computer Graphics Forum*, Vol. 31, Wiley Online Library **2012**, pp. 835–844.
- [16] L. Jin, A. E. Forte, B. Deng, A. Rafsanjani, K. Bertoldi, *Adv. Mater.* **2020**, *32*, 2001863.
- [17] F. Connolly, C. J. Walsh, K. Bertoldi, *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 51.
- [18] L. H. Dudte, E. Vouga, T. Tachi, L. Mahadevan, *Nat. Mater.* **2016**, *15*, 583.
- [19] M. Konaković, K. Crane, B. Deng, S. Bouaziz, D. Piker, M. Pauly, *ACM Transactions on Graphics (TOG)* **2016**, *35*, 89.
- [20] A. S. Gladman, E. A. Matsumoto, R. G. Nuzzo, L. Mahadevan, J. A. Lewis, *Nat. Mater.* **2016**, *15*, 413.
- [21] J. Panetta, F. Ivoranu, T. Chen, E. Siéfert, B. Roman, M. Pauly, *ACM Transactions on Graphics (TOG)* **2021**, *40*, 39.
- [22] R. Bouzidi, Y. Lecieux, *Acta Astronaut.* **2012**, *74*, 69.
- [23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, *Neural Computation* **1989**, *1*, 541.
- [24] Y. Goldberg, *Journal of Artificial Intelligence Research* **2016**, *57*, 345.
- [25] P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, H. S. Park, *Phys. Rev. Lett.* **2018**, *121*, 255304.
- [26] P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, H. S. Park, *Physical Review Research* **2020**, *2*, 042006.
- [27] M. A. Bessa, P. Glowacki, M. Houlder, *Adv. Mater.* **2019**, *31*, 1904845.
- [28] G. X. Gu, C.-T. Chen, D. J. Richmond, M. J. Buehler, *Mater. Horiz.* **2018**, *5*, 939.
- [29] C. Yang, Y. Kim, S. Ryu, G. X. Gu, *Mater. Des.* **2020**, *189*, 108509.
- [30] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M. Bessa, *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 26414.
- [31] Z. Yang, C.-H. Yu, M. J. Buehler, *Sci. Adv.* **2021**, *7*, eabd7416.
- [32] J. K. Wilt, C. Yang, G. X. Gu, *Adv. Eng. Mater.* **2020**, *22*, 1901266.
- [33] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A deep learning framework for solution and discovery in solid mechanics, **2020**.
- [34] D. W. Abueidda, M. Almasri, R. Ammourah, U. Ravaioli, I. M. Jasiuk, N. A. Sobh, *Compos. Struct.* **2019**, *227*, 111264.
- [35] C.-T. Chen, G. X. Gu, *Advanced Theory and Simulations* **2019**, *2*, 1900056.

- [36] C. Ma, Z. Zhang, B. Luce, S. Pusateri, B. Xie, M. H. Rafiei, N. Hu, *npj Computational Materials* **2020**, *6*, 40.
- [37] M. Bessa, S. Pellegrino, *Int. J. Solids Struct.* **2018**, *139*, 174.
- [38] Y. Mao, Q. He, X. Zhao, *Sci. Adv.* **2020**, *6*, eaaz4169.
- [39] F. Connolly, P. Polygerinos, C. J. Walsh, K. Bertoldi, *Soft Rob.* **2015**, *2*, 26.
- [40] F. Connolly, C. J. Walsh, K. Bertoldi, *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 51.
- [41] P. Polygerinos, Z. Wang, J. T. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, C. J. Walsh, *IEEE Transactions on Robotics* **2015**, *31*, 778.
- [42] S. Kumar, S. Tan, L. Zheng, D. M. Kochmann, *npj Computational Materials* **2020**, *6*, 73.
- [43] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, W. Cai, *Nano Lett.* **2018**, *18*, 6570.
- [44] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik, *ACS central science* **2018**, *4*, 268.
- [45] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik, A. Zhavoronkov, *J. Chem. Inf. Model.* **2018**, *58*, 1194.
- [46] A. Kaufman, D. Cohen, R. Yagel, *Computer* **1993**, *26*, 51.
- [47] P. Y. Simard, D. Steinkraus, J. C. Platt, et al., in *Icdar*, vol. 3, Citeseer **2003**.
- [48] A. Torralba, A. A. Efros, in *CVPR 2011*, IEEE, Piscataway, NJ **2011**, pp. 1521–1528.
- [49] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q. V. Le, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2019**, pp. 113–123.
- [50] K. Vodrahalli, K. Li, J. Malik, *arXiv preprint arXiv:1811.12569* **2018**.
- [51] V. Birodkar, H. Mobahi, S. Bengio, *arXiv preprint arXiv:1901.11409* **2019**.
- [52] S. Paul, J. H. Bappy, A. K. Roy-Chowdhury, in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Piscataway, NJ **2016**, pp. 494–498.
- [53] I. V. Tetko, D. J. Livingstone, A. I. Luik, *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 826.
- [54] C. M. Bishop, *Machine Learning* **2006**, *128*, 9.
- [55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *Journal of Machine Learning Research* **2014**, *15*, 1929.
- [56] L. A. Barnes, C. D. Marshall, T. Leavitt, M. S. Hu, A. L. Moore, J. G. Gonzalez, M. T. Longaker, G. C. Gurtner, *Advances in Wound Care* **2018**, *7*, 47.
- [57] G. C. Gurtner, R. H. Dauskardt, V. W. Wong, K. A. Bhatt, K. Wu, I. N. Vial, K. Padois, J. M. Korman, M. T. Longaker, *Ann. Surg.* **2011**, *254*, 217.
- [58] D. Maturana, S. Scherer, in *2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ **2015**, pp. 3471–3478.
- [59] J. Huang, S. You, in *2016 23rd Int. Conf. on Pattern Recognition (ICPR)*, IEEE, Piscataway, NJ **2016**, pp. 2670–2675.
- [60] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, J. Choo, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2018**, pp. 8789–8797.
- [61] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**, pp. 1125–1134.
- [62] M. Mirza, S. Osindero, *arXiv preprint arXiv:1411.1784* **2014**.