Inverse Design of Mechanical Metamaterials with Target Nonlinear Response via a Neural Accelerated Evolution Strategy

Bolei Deng, Ahmad Zareei, Xiaoxiao Ding, James C. Weaver, Chris H. Rycroft, and Katia Bertoldi**

Materials with target nonlinear mechanical response can support the design of innovative soft robots, wearable devices, footwear, and energy-absorbing systems, yet it is challenging to realize them. Here, mechanical metamaterials based on hinged quadrilaterals are used as a platform to realize target nonlinear mechanical responses. It is first shown that by changing the shape of the quadrilaterals, the amount of internal rotations induced by the applied compression can be tuned, and a wide range of mechanical responses is achieved. Next, a neural network is introduced that provides a computationally inexpensive relationship between the parameters describing the geometry and the corresponding stress–strain response. Finally, it is shown that by combining the neural network with an evolution strategy, one can efficiently identify geometries resulting in a wide range of target nonlinear mechanical responses and design optimized energy-absorbing systems, soft robots, and morphing structures.

1. Introduction

From wearable devices and energy-absorbing systems to scaffolds and soft robots, many applications would benefit from the inverse design of materials with a target nonlinear mechanical response. For example, to enhance tissue regeneration, scaffolds should be designed to match the nonlinear response of the surrounding native tissue.[1] Further, wearable and flexible electronics must accommodate the large deformations of soft biological tissues and reduce the stresses induced on the skin by their presence.[2] Finally, it has been shown that reusable, rate-independent and self-recoverable energy-absorbing materials can be realized by engineering structures that display nonlinear responses characterized by sudden snapping-induced load drops.[3,4]

Mechanical metamaterials have recently emerged as an effective platform to engineer systems with mechanical behaviors that are governed by geometry rather than composition.[5–8] While initial efforts have focused on the design of metamaterials with negative properties in the linear regime,[9–12] more recently it has been shown that highly nonlinear responses (often accompanied by large internal rotations) can be triggered by introducing into the architectures slender elements that are prone to elastic instabilities.[5,13] These nonlinear behaviors not only display very rich physics but can also be exploited to enable advanced functionalities, such as shape morphing,[14,15] energy absorption[3,16–18] and programmability.[19–21] Although it is well known that such functionalities can be tuned by altering the underlying geometry, the identification of architectures that result in a target nonlinear response is a non-trivial task.

Robust and efficient algorithms have been established to guide the design of structures with target response in the linear regime. These include gradient based methods such as shape[22] and topology[23] optimization, as well as machine learning algorithms.[24–27] However, such approaches cannot be directly applied to the inverse design of nonlinear mechanical metamaterials. This is because the energy landscapes of the nonlinear systems typically display multiple minima separated by large energy barriers and, therefore, are very challenging to navigate. To efficiently explore such energy landscapes, metaheuristic algorithms such as evolution strategies,[28–30] genetic algorithms[31] and particle swarm optimization,[32] have been successfully used. Further, since these algorithms require solving many times the forward problem, recent efforts have focused on reducing their computational cost by coupling them

B. Deng, A. Zareei, X. Ding, J. C. Weaver, C. H. Rycroft, K. Bertoldi
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA
E-mail: boleideng@g.harvard.edu; bertoldi@seas.harvard.edu

B. Deng
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

B. Deng
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

C. H. Rycroft
Computational Research Division
Lawrence Berkeley Laboratory
Berkeley, CA 94720, USA

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/adma.202206238.

DOI: 10.1002/adma.202206238

with machine learning models trained to solve the forward problem.[33–38]

Here, we present a framework to design mechanical metamaterials with target nonlinear response. Our starting point is a metamaterial based on hinged rotating squares (see **Figure 1**a), which has recently attracted significant interest as it displays effective negative Poisson's ratio[39–41] and supports the propagation of solitary pulses.[42–44] We first show that changes in the shape of the quadrilateral units lead to a wide range of mechanical responses and identify the key ingredients governing such behaviors. Then, we use neural networks to accurately learn the relationship between the geometry of the metamaterials and their nonlinear mechanical response. Finally, we combine neural networks with an evolution strategy to efficiently identify geometries that exhibit target nonlinear stress–strain behaviors. The proposed neural accelerated evolution strategy holds potential for a range of applications that benefit from systems with a target nonlinear mechanical behavior, as demonstrated by the design of energy absorbing systems, soft robots and morphing structures.

2. Our Physical Platform

We start by testing under uniaxial compression two elastomeric metamaterials comprising a 10 × 8 array of hinged quadrilateral units flanked by two horizontal strips of solid material (Figure 1a). Both structures are realized by repeating a unit cell that consists of 2 × 2 units connected at their vertices by thin beam-like ligaments with width and length of 1 mm. The unit cell of one sample comprises four identical squares with center-to-center distance a = 10 mm, while that of the other consists of four different irregular quadrilaterals obtained by randomly perturbing the position of the squares vertices (while maintaining a unit cell size of 2a in both horizontal and vertical directions—Figure 1a and Sections S1 and S2, Supporting Information). In Figure 1b we show snapshots of the two structures at an applied compressive strain $\varepsilon = -0.1$, while in Figure 1c we report the stress–strain curves recorded during our tests. We find that, although in both samples the deformation localizes at the beam-like hinges (which are identical in the two samples), their mechanical responses are remarkably different. In the sample with square units the buckling (and the subsequent bending) of the beam-like ligaments triggered at a critical compressive strain $\varepsilon \approx -0.02$ makes all squares to alternatively rotate in clockwise and counter-clockwise direction.[39,45,46] The shrinking associated with such collective rotational motion accommodates most of the additional applied strain and therefore limits the amount of compression in the vertical ligaments, ultimately leading to a stress plateau. By contrast, geometric frustration prevents rotation of the units in the sample comprising irregular quadrilaterals. It follows that for the sample comprising irregular quadrilaterals the applied deformation is almost entirely accommodated by the axial compression of the ligaments aligned along the vertical direction, resulting in a stiff and near-linear response.

The results of Figure 1 show that the shape of the quadrilaterals has a profound effect on the nonlinear mechanical response of the resulting metamaterial (see also Movie S1, Supporting Information). To systematically explore such effect, we make use of numerical simulations. In particular, since

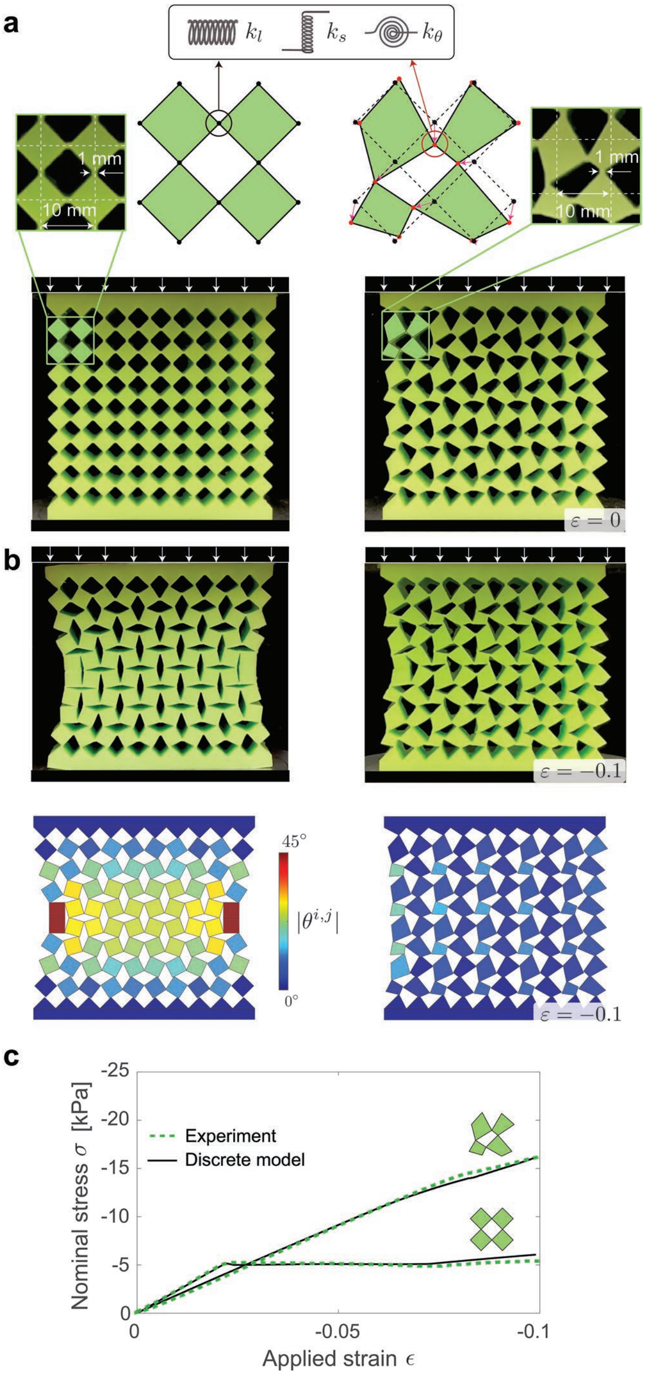

Figure 1. a) Top row: Unit cells of two structures comprising 2 × 2 quadrilateral units connected at their vertices by thin beam-like ligaments. The ligaments can be modeled using a combination of three linear springs with stiffness k_l, k_s, and k_θ. Bottom row: Pictures of the corresponding samples in the initial undeformed configuration. b) Top row: Pictures of the samples under $\varepsilon = -0.1$ compression strain. Bottom row: Numerical snapshots of the samples under $\varepsilon = -0.1$ compression strain. The color here indicates the local rotation of the quadrilaterals. c) Stress–strain response of the samples as measured in experiments (green dashed lines) and predicted by the discrete model (black lines).

our experiments indicate that under uniaxial compression the quadrilateral units remain rigid and the deformation localizes at the hinges, the nonlinear behavior of our structures can be described using a discrete model comprising rigid units connected at their vertices by a combination of three springs[45,46] (see Figure 1a): (i) a linear spring with stiffness $k_l = 0.47$ N mm^{-1} to capture the longitudinal response of the hinges; (ii) a linear spring with stiffness k_s = 0.16 N mm^{-1} to model their shearing; and (iii) a nonlinear hardening rotational spring that exert a torque $M = k_\theta(\Delta\theta + \gamma\Delta\theta^3)$ (where $\Delta\theta$ is the relative rotation between the connected quadrilaterals, k_θ = 0.088 N mm and $\gamma = -0.2$) to capture their bending (see Section S3, Supporting Information). By imposing force equilibrium at each unit, we derive a system of coupled nonlinear equations that we numerically solve to obtain the response of the structure. Note that in this study we consider metamaterials with out-of-plane thickness large enough to prevent out-of-plane deformation and, therefore, limit the analyses to in-plane deformation. To test the relevance of our discrete model, we first compare its predictions to the experimental results reported in Figure 1 (see also Movie S1, Supporting Information). We find a very good agreement between our experimental and numerical results both in term of stress–strain curves and deformation field, confirming the validity of our model.

Next, we create 7500 different unit cells by randomly choosing the position of the quadrilateral vertices while preserving periodicity (see **Figure 2**a and Section S4, Supporting Information) and use our discrete model to simulate the mechanical response of the resulting metamaterials under uniaxial compression. In Figure 2b we report the numerically predicted stress–strain curves for all generated architectures. We find that a variety of nonlinear mechanical responses emerge. These include near-linear behaviors with a wide range of tangent moduli (see curves (i) and (ii) in Figure 2b for the upper and lower limits) as well as highly nonlinear behaviors that exhibit strain-softening and sudden load drops (see curves (iii) and (iv) in Figure 2b for representative examples). Further, our numerical results indicate that such different responses are accompanied by distinct deformation modes (see also Movie S2, Supporting Information). Focusing on the four architectures whose response is highlighted in Figure 2b, we find that at $\varepsilon = -0.1$ the rotations of the units are relatively small for the two structures that display near-linear responses and much larger for the two exhibiting nonlinear behaviors (see Figure 2c (top) and Movie S2, Supporting Information). Importantly, the connection between internal rotations and nonlinearity is not limited to these four architectures. As shown in Figure 2d, for all 7500 metamaterials there is a correlation between the

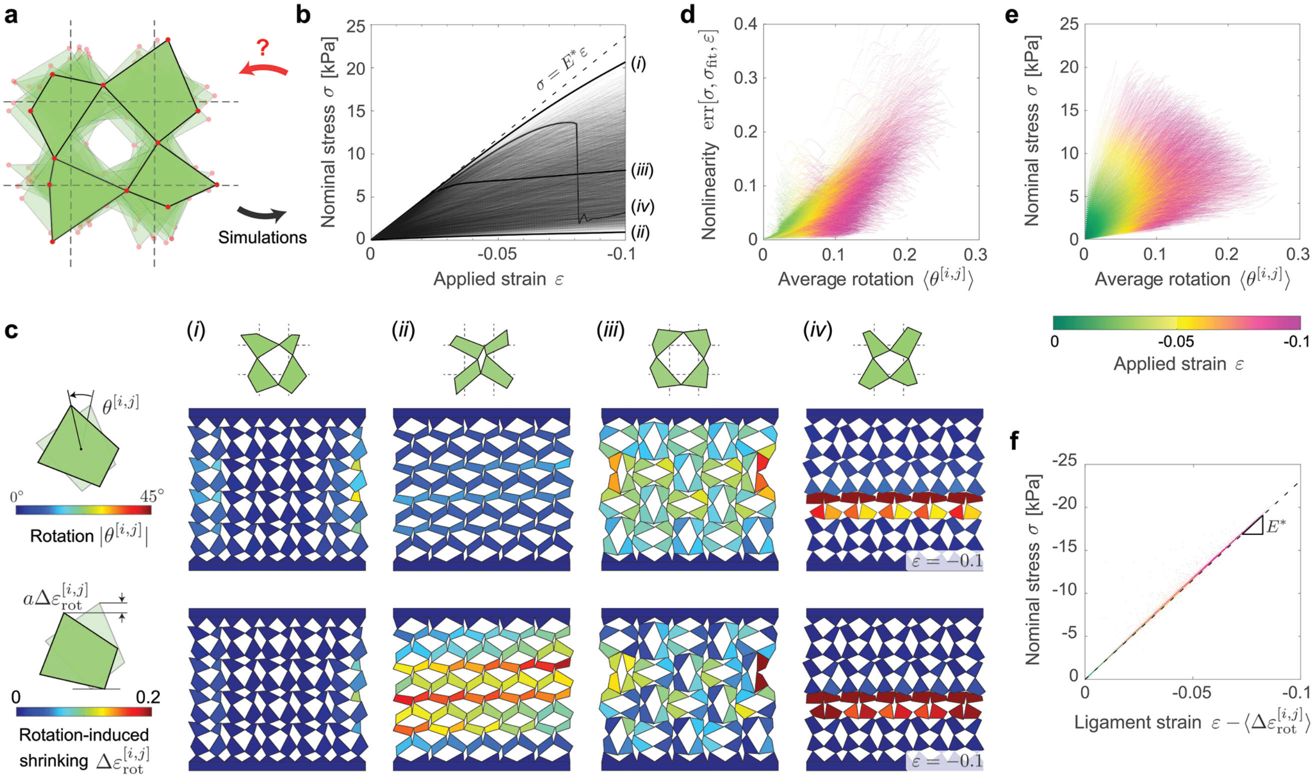

Figure 2. a) Unit cells generated by randomly perturbing the position of the quadrilaterals vertices, while preserving periodicity. b) Stress–strain curves predicted by the discrete model for 7500 different metamaterials realized by tessellating different unit cells. Four representative stress–strain curves are highlighted. These include (i)-(ii) near-linear behaviors with a wide range of stiffness; (iii) stress–strain curves with an initial linear regime followed by a plateau; and (iv) stress–strain curves displaying sudden drops in force. The upper bound $\sigma = E^*\varepsilon$ is shown as a dashed black line. c) Numerical snapshots of the structures highlighted in (b) at $\varepsilon = -0.1$. The color in the top row indicates the rotation of the quadrilaterals, while the color in the bottom row indicates the rotation-induced shrinking. d) Average nonlinearity ($\mathrm{err}[\sigma,\sigma_{\mathrm{fit}},\varepsilon]$) versus average rotation ($\langle\theta^{[i,j]}\rangle$) for all 7500 metamaterials at different level of applied strain. e) Nominal stress (σ) versus average rotation ($\langle\theta^{[i,j]}\rangle$) for all 7500 metamaterials at different levels of applied strain. f) Nominal stress (σ) versus strain accommodated by a column of vertical ligaments ($\varepsilon - \Delta\varepsilon_{\mathrm{rot}}$) for all 7500 metamaterials at different levels of applied strain.

average rotation of the units, $\langle\theta^{[i,j]}\rangle$ (where $\theta^{[i,j]}$ denotes the rotation of the [*i*,*j*]-th unit), and the amount of nonlinearity displayed by the corresponding stress–strain curve. Note that the latter is quantified using the normalized root mean squared error between the stress–strain curve and its best linear fit, err[σ,σ_{fit},ε], where $\sigma_{\text{fit}} = E\varepsilon$ represents the best linear fit to the curve and

$$\text{err}[f,g,\varepsilon] = \sqrt{\frac{\int_0^{\varepsilon}(f(\varepsilon') - g(\varepsilon'))^2 d\varepsilon'}{\int_0^{\varepsilon} f(\varepsilon')^2 d\varepsilon'}} \tag{1}$$

with the integrals being evaluated numerically. Near-linear responses (for which err[σ,σ_{fit},ε] → 0) are accompanied by small rotations of the units, whereas large internal rotations lead to highly nonlinear behaviors characterized by large values of err[σ,σ_{fit},ε].

Differently, we find that there is no correlation between $\langle\theta^{[i,j]}\rangle$ and the the recorded nominal stress, σ (see Figure 2e). This is because small internal rotations can be amplified by the geometry of the quadrilaterals and result in large amount of shrinking along the vertical direction (see Figure 2c-ii). Such rotation-induced shrinking reduces the level of axial compression in the ligaments and therefore is expected to affect the measured nominal stress. To quantify this effect, we define the rotation-induced shrinking along the vertical direction for the [*i*,*j*]-th unit as

$$\Delta\varepsilon_{\text{rot}}^{[i,j]}(\varepsilon) = \frac{\left(y_{\text{top}}^{[i,j]}(\varepsilon) - y_{\text{b}}^{[i,j]}(\varepsilon)\right) - \left(y_{\text{top}}^{[i,j]}(0) - y_{\text{b}}^{[i,j]}(0)\right)}{a} \tag{2}$$

where $y_{\text{top}}^{[i,j]}(\varepsilon)$ and $y_{\text{b}}^{[i,j]}(\varepsilon)$ denote the *y*-coordinate of the top and bottom vertex of the [*i*,*j*]-th quadrilateral at ε. The numerical snapshots reported in Figure 2c (bottom) confirm that $\Delta\varepsilon_{\text{rot}}^{[i,j]}$ can be large even in structures that display small internal rotations (see Figure 2c-ii). As a matter of fact, there is a strong correlation between σ and the average rotation-induced shrinking, $\left\langle\Delta\varepsilon_{\text{rot}}^{[i,j]}\right\rangle$. More specifically, since the compressive strain accommodated by a column of vertical ligaments is given by $\varepsilon - \left\langle\Delta\varepsilon_{\text{rot}}^{[i,j]}\right\rangle$, we find that for all considered metamaterials

$$\sigma \approx E^*\left(\varepsilon - \left\langle\Delta\varepsilon_{\text{rot}}^{[i,j]}\right\rangle\right) \tag{3}$$

where E^* is the effective Young's modulus of the metamaterial in the absence of rotations (see Figure 2f). Note that $E^* = 10k_{\text{l}}/(9b)$, where *b* denotes the out-of-plane thickness of the metamaterial, since our structure comprises 10 columns of vertical ligaments with longitudinal stiffness k_{l}, each consisting of 9 ligaments arranged in series. It follows from Equation (3) that the stiffest response for the considered class of metamaterials can be achieved in the absence of rotations (i.e., for $\left\langle\Delta\varepsilon_{\text{rot}}^{[i,j]}\right\rangle = 0$) and is given by $\sigma = E^*\varepsilon$ (see dashed line in Figure 2b). As such, our results indicate that by tuning the amount of rotation-induced shrinking upon compression we can successfully manipulate the stress–strain curves of the considered class of metamaterials. Importantly, this tuning can be accomplished by varying the geometry of the quadrilateral units.

3. Neural Networks

While in Figure 2 we focus on the mechanical response of 7500 architectures, our numerically generated stress–strain curves enable prediction of the behavior of metamaterials realized out of arbitrary unit cells. This is because such data can serve as offline training of a machine-learning (ML) model that provides a computationally inexpensive relationship between the parameters describing the unit cell geometry and the corresponding stress–strain response. To this end, we represent the *p*-th unit cell in our database as a 16-dimensional vector $\mathbf{X}_p = [\mu_p^1, \nu_p^1, \ldots, \mu_p^8, \nu_p^8]^\top$, where μ_p^α and ν_p^α represent the perturbations in horizontal and vertical directions applied to its α-th independent vertex for the *p*-th unit cell (see Section S1, Supporting Information) and the corresponding stress–strain curve as a 100-dimensional vector $\boldsymbol{\sigma}_p = [\sigma_p(0), \ldots, \sigma_p(-0.1)]^\top$, which contains the stress values at 100 equally spaced strain points. Note that, although we simulated 7500 metamaterials, our dataset contains a total of *N* = 30 000 datapoints, since the response of all designs remains unaltered when they are reflected over the *x*-axis, the *y*-axis and in the origin (see **Figure 3**a). Out of the expanded dataset, we randomly choose $N_{\text{tr}} = 0.8N$ datapoints for training and the remaining $N_{\text{test}} = 0.2N$ for testing.

To facilitate training of the ML model, we reduce the dimensionality of the stress vectors by using principal component analysis (PCA).[47] In particular, we describe the stress–strain curve of the *p*-th design via its first 10 principal components, $\boldsymbol{\sigma}_p^{\text{PC}_{10}}$. Note that this leads to an average reconstruction error of $\frac{1}{N_{\text{tr}}}\sum_{p=1}^{N_{\text{tr}}}\text{err}[\boldsymbol{\sigma}_p, \boldsymbol{\sigma}_p^{\text{R}}, -0.1] = 0.3\ \%$, where $\boldsymbol{\sigma}_p$ denotes the true stress vector and $\boldsymbol{\sigma}_p^{\text{R}}$ the reconstructed one from the first 10 principal components (Section S5, Supporting Information).

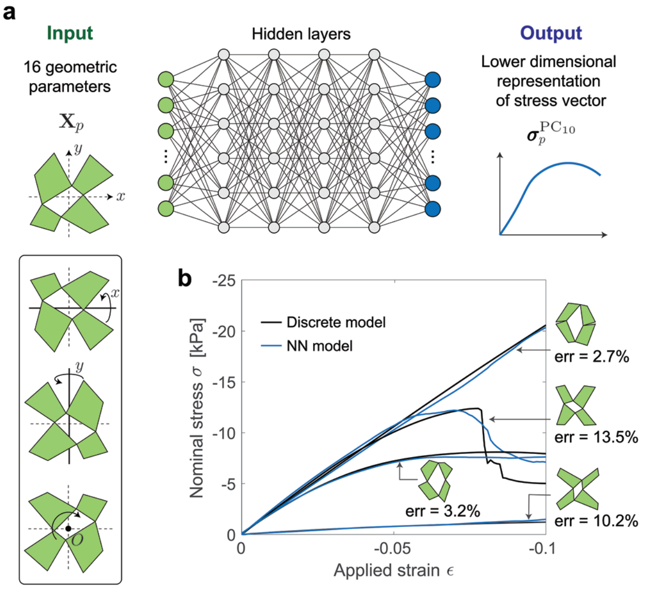

Figure 3. a) Our NN architecture has four hidden layers, each comprising 200 neurons. A 16-dimensional vector $\mathbf{X}_p$ is fed to the NN that is trained to predict the lower-dimensional representation of the corresponding stress vector $\boldsymbol{\sigma}_p^{\text{PC}_{10}}$. b) Comparison between the stress–strain curves predicted by our discrete (black solid lines) and NN (dashed blue lines) models for four geometries that are part of the test dataset.

To predict the lower dimensional representation of the stress vector, $\sigma_p^{\mathrm{PC}_{10}}$, for a given geometry, $\mathbf{X}_p$, we use a neural network (NN) architecture with four hidden layers, each comprising 200 neurons (see Figure 3a and Section S6, Supporting Information). We train the NN by selecting the neuron weights and biases that minimize

$$\mathcal{L}=\sum_{p=1}^{N_{\mathrm{tr}}}\sum_{s=1}^{10}\beta_s\left(\mathcal{F}_s(\mathbf{X}_p)-\sigma_{p,s}^{\mathrm{PC}_{10}}\right)^2 \tag{4}$$

where $\sigma_{p,s}^{\mathrm{PC}_{10}}$ is the s-th component of $\sigma_p^{\mathrm{PC}_{10}}$, $\mathcal{F}(\mathbf{X}_p)=[\mathcal{F}_1(\mathbf{X}_p),\ldots,\mathcal{F}_{10}(\mathbf{X}_p)]^\top$ denote the corresponding NN predictions and β_s is the variance explained in the s-th principal component (Section S5, Supporting Information). Finally, to evaluate the accuracy of our trained NN, we reconstruct the stress–strain response from $\mathcal{F}(\mathbf{X}_p)$ and calculate the average relative error on the test dataset as $\frac{1}{N_{\mathrm{test}}}\sum_{p=1}^{N_{\mathrm{test}}}\mathrm{err}[\sigma_p,\sigma_p^{\mathrm{NN}},-0.1]$, where σ_p^{NN} is the stress vector reconstructed from $\mathcal{F}(\mathbf{X}_p)$. We find that the trained NN accurately predicts the stress–strain curves associated to unseen designs, with an average relative error of 4.8%. As examples, in Figure 3b we focus on four designs that are part of the test dataset and display distinct mechanical behaviors. The comparison between the stress–strain curves predicted by our discrete and NN models for these four geometries indicates that our trained NN can capture all behaviors that are representative of the considered class of mechanical metamaterials, with discrepancies that are more accentuated for the stress–strain curves displaying sharp drops in force. This is because such sharp drops are induced by the snapping of a few unit cells (see numerical snapshot at $\varepsilon=-10$ in Figure 2c-iv)—a phenomenon that is known to be very sensitive to geometric perturbations[48] and, therefore, difficult to predict.

4. Inverse Design

Our trained NN enables us to efficiently determine the mechanical response of arbitrary metamaterials. However, since the connection between the geometry of the unit cells and the mechanical response of the corresponding structure is not trivial, identification of metamaterials with target behavior requires an efficient inverse design strategy. Toward this end, as shown in **Figure 4**a, we couple our trained NN with an evolution strategy (ES)—a type of stochastic global optimization algorithm inspired by the biological theory of evolution.[28] To test our approach, we focus on four distinct target behaviors (see Figure 4b)

$$\begin{aligned}
\sigma_{\mathrm{t}}^{(\mathrm{i})} &= E^*\varepsilon\\
\sigma_{\mathrm{t}}^{(\mathrm{ii})} &= 0.8\sigma_{\mathrm{soft}}\\
\sigma_{\mathrm{t}}^{(\mathrm{iii})} &= E^*(\varepsilon+9\varepsilon^2)\\
\sigma_{\mathrm{t}}^{(\mathrm{iv})} &= \begin{cases}0.8E^*\varepsilon, & \mathit{if}\ \varepsilon>-0.07\\ -0.025E^*, & \mathit{if}\ \varepsilon<-0.07\end{cases}
\end{aligned} \tag{5}$$

where σ_{soft} denotes the softest response within the dataset and "t indicates "target". Note that $\sigma_{\mathrm{t}}^{(\mathrm{i})}$ represents the stiffest possible response for the considered class of metamaterials (see Equation (3)), whereas $\sigma_{\mathrm{t}}^{(\mathrm{ii})}$ describes a near-linear stress–strain curve that is 20% softer than the softest one within the dataset. Further, $\sigma_{\mathrm{t}}^{(\mathrm{iii})}$ represents a smooth stress–strain curve

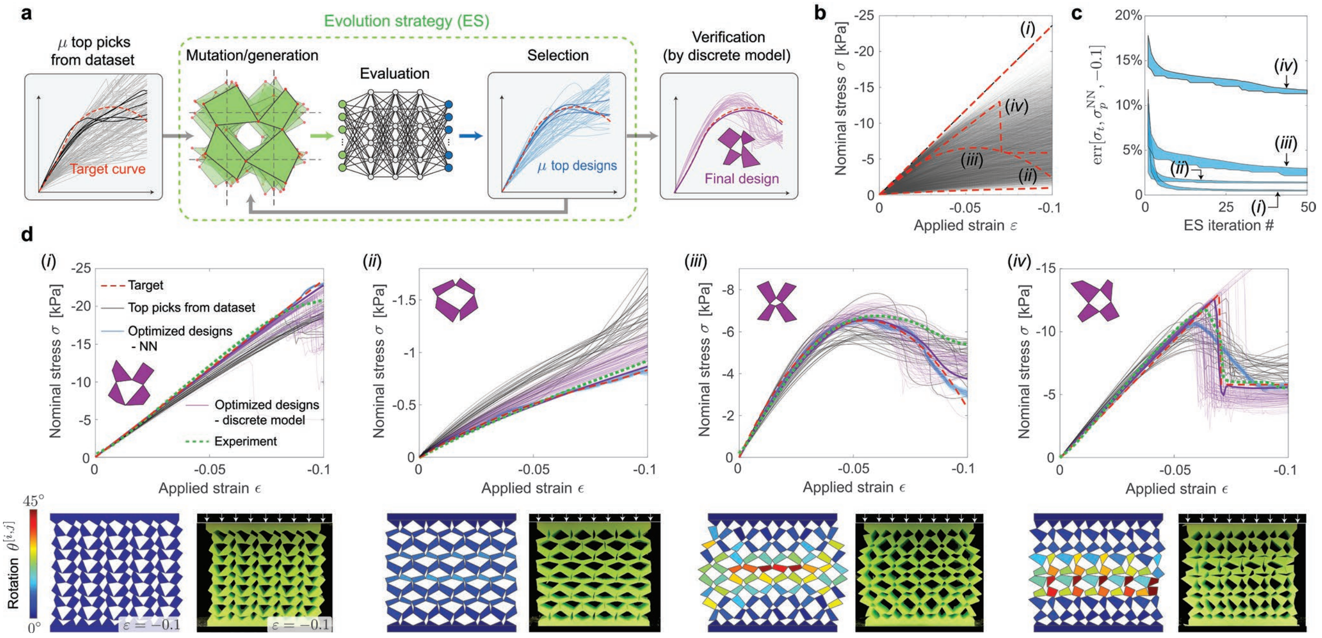

Figure 4. a) Schematic representing the proposed inverse design algorithm based on a neural accelerated evolution strategy. b) The four target stress–strain curves considered in this study. c) Evolution of $\max_p \mathrm{err}[\sigma_{\mathrm{t}},\sigma_p^{\mathrm{NN}},-0.1]$ during ES iterations for the four considered targets. d) Target response (red dashed line), top picks from dataset (black lines) and stress–strain curves predicted by our trained NN (blue lines) and our discrete model (purple lines) for the optimized designs. The designs that minimize $\mathrm{err}[\sigma_{\mathrm{t}},\sigma_p^{\mathrm{DM}},-0.1]$ are shown as inset. For each of these designs we also show numerical and experimental snapshots at $\varepsilon=-0.1$. The color in the numerical snapshots indicates the local rotation of the quadrilaterals.

that display a strain-induced stress drop more accentuated than those found in the dataset and $\sigma_{\mathrm{t}}^{(\mathrm{iv})}$ is characterized by a sharp drop in stress at an unseen location. To identify geometries leading to these target behaviors, we start by selecting μ = 100 stress–strain curves from our dataset that minimize $\mathrm{err}[\sigma_{\mathrm{t}}, \sigma_p, -0.1]$. As expected, no response in the dataset closely matches our four target behaviors (see black lines in Figure 4d)—with a minimum error between the top picks and the target ones equal to: (i) 11%, (ii) 25%, (iii) 13% and (iv) 12%. Next, in an attempt to reduce these errors, we generate λ = 49 new candidate geometries (the children) from each of the unit cells associated to these solutions (the parents) by randomly moving their eight independent vertices within boxes with edge of 0.1a centred around them. Further, to better explore the design space, we generate 5000 completely arbitrary unit cells that compete with the parents and children for consideration in the next iteration (see Figure S17, Supporting Information for a discussion on the benefits of adding these 5000 arbitrary unit cells). The parents, children, and arbitrary unit cells form a set of 10 000 geometries, whose stress–strain curves are determined using our trained NN. Out of this set, we then select the parents for the next iteration by choosing the μ = 100 designs that minimize $\mathrm{err}[\sigma_{\mathrm{t}}, \sigma_p^{\mathrm{NN}}, -0.1]$ and we continue this mutation/generation and selection process until there are μ = 100 designs for which $\max_p(\mathrm{err}[\sigma_{\mathrm{t}}, \sigma_p^{\mathrm{NN}}, -0.1]) < 1\%$, or 50 iterations are reached. It is important to note that the trained NN enables us to largely reduce the computation time required to identify the optimal designs, since it takes only 0.2 s for the NN to evaluate 10000 designs (whereas it takes 23.2 days for the the discrete model to evaluate 10000 designs).[49] In Figure 4c we report the evolution of $\mathrm{err}[\sigma_{\mathrm{t}}, \sigma_p^{\mathrm{NN}}, -0.1]$ for the best performing μ = 100 designs over the course of each iteration, whereas in Figure 4d we show as blue lines the stress–stress curves predicted by our trained NN for the optimized μ = 100 designs identified by the ES. We find that the error between the optimal curves and the target ones is largely reduced for the considered first three targets. Differently, the reduction in error is less significant for the fourth target response, as 10 principal components are not enough to capture a stress–strain curve with a sharp drop in load (see blue lines in Figure 4d-iv). Although such error could be reduced by considering a larger number of principal components to represent the stress–strain curve, it is important to point out that this would not necessarily translate to a better solution. This is because the trained NN introduces an error in the estimation of their mechanical responses, which becomes more accentuated for designs that are very different from those included in the training dataset. As a matter of fact, when we use the discrete model to simulate the final μ = 100 designs identified by ES, we find an average error $\frac{1}{\mu}\sum_p \mathrm{err}[\sigma_p, \sigma_p^{\mathrm{NN}}, -0.1]$ = 15% on these optimized designs (see purple lines in Figure 4d)—much higher than that obtained on the test set (4.8%). However, despite the error introduced by the NN, the μ = 100 optimized designs identified by ES display stress–strain curves that are on average much closer to the target curves than the close matches from the dataset (black curves). More specifically, we find that the minimum error between σ_{t} and the stress–strain curve predicted by the discrete model for the μ optimized geometries, σ_p^{DM}, is reduced to: (i) 1.9%, (ii) 4.7%, (iii) 6.6%, and (iv) 4.5% (the designs that best match the target curves are shown as thick purple curves in Figure 4d).

Next, we physically fabricate the optimized designs that closely match the target curves and report the experimentally recorded stress–strain curves (green dashed lines) as well as snapshots of the samples at $\varepsilon = -0.1$ in Figure 4d (see also Movie S3, Supporting Information). Despite the unavoidable imperfections introduced during fabrication and testing, the recorded stress–strain curves closely match the target ones. Focusing on the first target, we find that our optimization framework identifies a geometry that prevents internal rotations under compression and, as predicted by Equation (3), results in the stiffest possible response. As for the second target, the inversely designed metamaterial comprises highly tilted (almost flat) elongated quadrilaterals that maximize the rotation-induced shrinking and therefore lead to a very compliant response. Differently, the optimized geometry for the third target consists of elongated quadrilaterals that are almost aligned along the vertical direction. Upon compression these units gradually rotate (see Figure 4d-iii), so that the structure approaches the optimized geometry for the second target and, therefore, becomes much softer. Finally, the fourth optimized design comprises two rows of quadrilaterals that snap at the target strain, yielding a sudden load drop.

5. Conclusions

We have introduced a neural accelerated evolution strategy to identify mechanical metamaterials with target nonlinear response.

These optimized metamaterials provide opportunities for the design of a wide range of smart structures, including energy-absorbing systems and soft robots. To investigate the energy-absorbing performance upon impact of our inversely designed metamaterials, we conduct drop tests on the optimized structures reported in Figure 4. More specifically, we drop a circular acrylic disk (with mass m_{disk} = 100 g and diameter d_{disk} = 100 mm) onto their top surface from an height of 40 cm and measure the rebound height after the impact. As shown in **Figure 5**a and Figure S3, Supporting Information, we find that after the impact the disk bounces back to a height of 22.0, 15.3, 11.5 and 2.5 cm for the four considered structures, resulting in an absorption efficiency of 45%, 62%, 71% and 94%, respectively (see Section S2.C, Supporting Information for more details). The superior performance of the architecture optimized to match $\sigma^{(\mathrm{iv})}$ confirms that snapping provides a simple yet effective mechanism to realize reusable energy-absorbing materials[50] and further indicate that our numerical strategy can be harnessed to optimize their absorption efficiency.

Additionally, while in this study we have focused on target nonlinear stress–strain responses, our neural accelerated evolution strategy can be readily applied to other nonlinear properties, providing a platform to facilitate the design of soft robots and actuators. For example, it has been shown that a soft robot capable of moving through a channel when actuated with a single input can be realized by combining two flexible structures with Poisson's ratio of equal magnitude but opposite

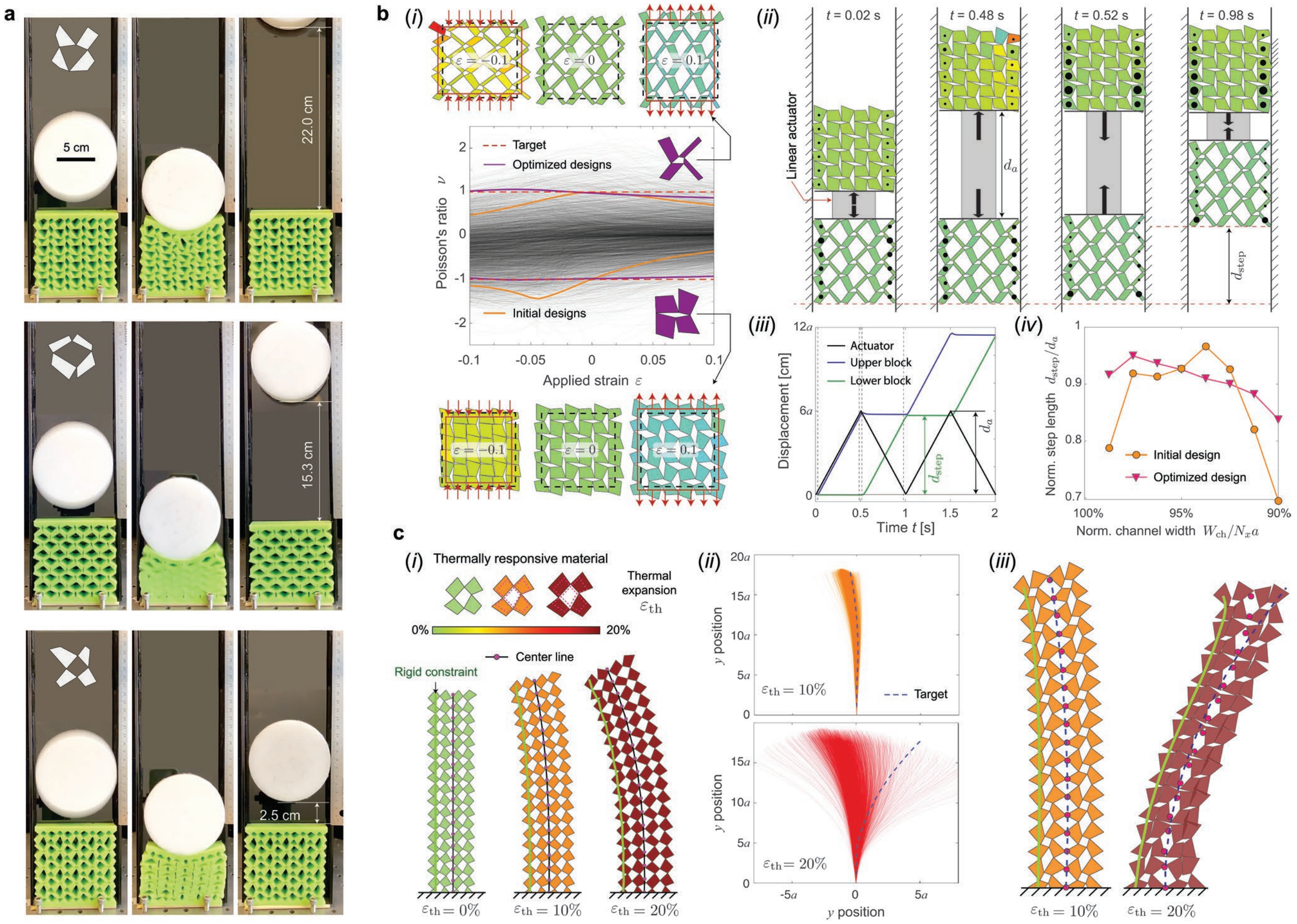

Figure 5. a) Energy absorption: Experimental images taken during the drop tests immediately before contact (left), at contact (center) and at the maximum height at which the disk bounces back after impact (right) for the metamaterial optimized to match $\sigma_t^{(i)}$ (top), $\sigma_t^{(ii)}$ (center) and $\sigma_t^{(iv)}$ (bottom). For these three structures we measure a contact time of 33, 75 and 96 ms and the height of the first bounce after the impact to be 22.0, 15.3, and 2.5 cm, resulting in an absorption efficiency $\eta = 45\%$, 62%, and 94%, respectively. b) Inverse design of soft robots that exploit auxetic behavior. (i) Evolution of the effective Poisson's ratio ν as a function of the applied strain ε predicted by the discrete model for 7500 different metamaterials realized by tessellating different unit cells. The neural accelerated evolution strategy is used to inverse design two architectures with constant Poisson's ratio $\nu(\varepsilon) = \pm 1$ for $\varepsilon \in [-0.1,0.1]$. (ii) Numerical snapshots of the soft robot realized by connecting the two inversely designed metamaterials though a linear actuator moving into a channel with width $W_{ch} = 0.99 N_x a$ at time $t = 0.02$ s, 0.48 s, 0.52 s, and 0.98 s. The color here indicates the local rotation of the quadrilaterals. (iii) Relative displacement imposed by the linear actuator (black line) and displacement of the center of mass of the lower (green line) and upper (blue line) blocks as a function on time. (iv) Evolution of the displacement recorded at the end of each cycle, d_{step} (normalized by the actuator stroke d_a) as a function of $W_{ch}/(N_x a)$ for the initial design (yellow markers) and the optimized design (magenta markers). c) Inverse design of structures capable of target deformations upon heating. (i) We consider a structure comprising 8×2 unit cells and assume that each quadrilateral unit expands uniformly under an increasing temperature. To trigger bending, we fix the center-to-center distance of the units on the left boundary. (ii) The distribution of the center lines of 7500 different unit cell deigns at $\varepsilon_{th} = 0.1$ (top) and 0.2 (bottom). The blue dashed lines represent the target curves at different thermal strains. (iii) The inverse designed metamaterial is able to closely match the target center lines at $\varepsilon_{th} = 0.1$ and 0.2.

sign.[51] However, since mechanical metamaterials typically exhibit varying Poisson's ratio in the nonlinear regime,[52,53] the efficiency of such robots largely varies with the width of the channel (Figure 5b(iv)). To overcome this limitation, we inverse design two architectures with constant Poisson's ratio $\nu = \pm 1$ over a large range of deformations (i.e., for $-0.1 < \varepsilon < 0.1$) via our neural accelerated evolution strategy. As shown in Figure 5b(iv), the identified optimal structures lead to a soft robot with inchworm locomotion that is less affected by the width of the channel (see Section S8.A, Supporting Information for more details). Finally, our neural accelerated evolution strategy can be exploited to identify structures capable of target deformation upon application of an external stimulus, opening avenues for the inverse design of metamaterial-based soft actuators. To demonstrate this point, we consider a structure comprising 8×2 unit cells and assume that each quadrilateral unit expands uniformly under an increasing temperature (Figure 5c(i)). When the center-to-center distance of the units on the left boundary is fixed, the structures built out of the 7500 unit cells considered in this study bend into a variety of configurations upon heating (Figure 5c(ii)). Remarkably, we find that also in this case the neural network is able to learn the relation between the unit cell design and the deformed configuration and can be combined with evolution strategy to efficiently inverse design architectures capable of supporting multiple target deformation modes (see Figure 5c(iii) for a structure that bends toward the left for a thermal expansion $\varepsilon_{\text{th}} = 10\%$ and toward the right for $\varepsilon_{\text{th}} = 20\%$—see Section S8.B, Supporting Information for more details).

To conclude, since the response of the inversely designed metamaterials is scale- and material-independent, we envisage that our strategy could be applied to the design of the next generation of flexible structures with target nonlinear behavior, ranging from large-scale energy-absorbing systems to small-scale robotic components.

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, OAC-2118201. K.B. also acknowledges support from the Simons Collaboration on Extreme Wave Phenomena Based on Symmetries.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

B.D. and A.Z. contributed equally to this work. B.D. and K.B. designed the research. B.D. designed and conducted experiments, coded the discrete model and designed the neural accelerated evolution strategy. A.Z. fabricated samples, conducted experiments and analyzed the performance of neural networks. X.D. contributed to the original research idea, helped with experiments, and conducted preliminary neural network training. J.W. printed the molds for samples fabrication. B.D., A.Z. and K.B. wrote the manuscript. K.B. and C.R. supervised the research.

Data Availability Statement

The data that support the findings of this study are openly available in github at https://github.com/boleideng94/metamaterials-inverse-design.

Keywords

mechanical metamaterials, neural networks, nonlinear response, optimization

Received: July 9, 2022
Published online:

[1] B. Chan, K. Leong, *Eur. Spine J.* **2008**, *17*, 467.
[2] Y. Ma, X. Feng, J. A. Rogers, Y. Huang, Y. Zhang, *Lab Chip* **2017**, *17*, 1689.
[3] S. Shan, S. H. Kang, J. R. Raney, P. Wang, L. Fang, F. Candido, J. A. Lewis, K. Bertoldi, *Adv. Mater.* **2015**, *27*, 4296.
[4] Y. Chen, L. Jin, *Adv. Funct. Mater.* **2021**, 2102113.
[5] K. Bertoldi, V. Vitelli, J. Christensen, M. Van Hecke, *Nat. Rev. Mater.* **2017**, *2*, 1.
[6] M. Kadic, G. W. Milton, M. van Hecke, M. Wegener, *Nat. Rev. Phys.* **2019**, *1*, 198.
[7] A. A. Zadpoor, *Mater. Horiz.* **2016**, *3*, 371.
[8] J. Christensen, M. Kadic, O. Kraft, M. Wegener, *MRS Commun.* **2015**, *5*, 453.
[9] Z. G. Nicolaou, A. E. Motter, *Nat. Mater.* **2012**, *11*, 608.
[10] R. Lakes, *Science* **1987**, *235*, 1038.
[11] R. Gatt, L. Mizzi, J. I. Azzopardi, K. M. Azzopardi, D. Attard, A. Casha, J. Briffa, J. N. Grima, *Sci. Rep.* **2015**, *5*, 8395.
[12] M. Kadic, T. Bückmann, N. Stenger, M. Thiel, M. Wegener, *Appl. Phys. Lett.* **2012**, *100*, 191901.
[13] P. M. Reis, *J. Appl. Mech.* **2015**, *82*, 111001.
[14] L. Jin, A. E. Forte, B. Deng, A. Rafsanjani, K. Bertoldi, *Adv. Mater.* **2020**, *32*, 2001863.
[15] 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. USA* **2019**, *116*, 20856.
[16] C. M. Portela, B. W. Edwards, D. Veysset, Y. Sun, K. A. Nelson, D. M. Kochmann, J. R. Greer, *Nat. Mater.* **2021**, *20*, 1491.
[17] B. Haghpanah, L. Salari-Sharif, P. Pourrajab, J. Hopkins, L. Valdevit, *Adv. Mater.* **2016**, *28*, 7915.
[18] D. Restrepo, N. D. Mankame, P. D. Zavattieri, *Ext. Mech. Lett.* **2015**, *4*, 52.
[19] B. Florijn, C. Coulais, M. van Hecke, *Phys. Rev. Lett.* **2014**, *113*, 175503.
[20] E. Medina, P. E. Farrell, K. Bertoldi, C. H. Rycroft, *Phys. Rev. B* **2020**, *101*, 064101.
[21] T. Chen, M. Pauly, P. M. Reis, *Nature* **2021**, *589*, 386.
[22] J. Sokolowski, J.-P. Zolésio, in *Introduction to Shape Optimization*, Springer, Berlin/Heidelberg, Germany **1992**, pp. 5–12.
[23] O. Sigmund, K. Maute, *Struct. Multidiscip. Optim.* **2013**, *48*, 1031.
[24] Y. Mao, Q. He, X. Zhao, *Sci. Adv.* **2020**, *6*, eaaz4169.
[25] S. Kumar, S. Tan, L. Zheng, D. M. Kochmann, *npj Comput. Mater.* **2020**, *6*, 73.
[26] G. X. Gu, C.-T. Chen, M. J. Buehler, *Ext. Mech. Lett.* **2018**, *18*, 19.
[27] Z. Zhang, G. X. Gu, *Theor. Appl. Mech. Lett.* **2021**, 100220.
[28] H.-G. Beyer, H.-P. Schwefel, *Nat. Comput.* **2002**, *1*, 3.
[29] N. Hansen, A. Ostermeier, *Evol. Comput.* **2001**, *9*, 159.
[30] N. Hansen, S. D. Müller, P. Koumoutsakos, *Evol. Comput.* **2003**, *11*, 1.
[31] L. D. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York **1991**.
[32] J. Kennedy, R. Eberhart, in *Proc. ICNN 95 - Int. Conf. on Neural Networks*, vol. 4, IEEE, Piscataway, NJ, USA **1995**, pp. 1942–1948.
[33] Z. Yang, C.-H. Yu, M. J. Buehler, *Sci. Adv.* **2021**, *7*, eabd7416.
[34] J. K. Wilt, C. Yang, G. X. Gu, *Adv. Eng. Mater.* **2020**, *22*, 1901266.
[35] G. X. Gu, C.-T. Chen, D. J. Richmond, M. J. Buehler, *Mater. Horiz.* **2018**, *5*, 939.
[36] S. Babaee, J. Shim, J. C. Weaver, E. R. Chen, N. Patel, K. Bertoldi, *Adv. Mater.* **2013**, *25*, 5044.
[37] L. Wu, L. Liu, Y. Wang, Z. Zhai, H. Zhuang, D. Krishnaraju, Q. Wang, H. Jiang, *Extr. Mech. Lett.* **2020**, *36*, 100657.
[38] S. Wu, C. M. Hamel, Q. Ze, F. Yang, H. J. Qi, R. Zhao, *Adv. Intell. Syst.* **2020**, *2*, 2000060.
[39] K. Bertoldi, P. Reis, S. Willshaw, T. Mullin, *Adv. Mater.* **2010**, *22*, 361.
[40] J. N. Grima, K. E. Evans, *J. Mater. Sci. Lett.* **2000**, *19*, 1563.
[41] Y. Cho, J.-H. Shin, A. Costa, T. A. Kim, V. Kunin, J. Li, S. Y. Lee, S. Yang, H. N. Han, I.-S. Choi, et al., *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 17390.

[42] B. Deng, J. Raney, V. Tournat, K. Bertoldi, *Phys. Rev. Lett.* **2017**, *118*, 204102.
[43] B. Deng, C. Mo, V. Tournat, K. Bertoldi, J. R. Raney, *Phys. Rev. Lett.* **2019**, *123*, 024101.
[44] H. Yasuda, L. Korpas, J. Raney, *Phys. Rev. Appl.* **2020**, *13*, 054067.
[45] C. Coulais, C. Kettenis, M. van Hecke, *Nat. Phys.* **2018**, *14*, 40.
[46] B. Deng, S. Yu, A. E. Forte, V. Tournat, K. Bertoldi, *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 31002.
[47] I. T. Jolliffe, J. Cadima, *Philos. Trans. R. Soc., A* **2016**, *374*, 20150202.
[48] A. Lee, F. López Jiménez, J. Marthelot, J. W. Hutchinson, P. M. Reis, *J. Appl. Mech.* **2016**, *83*, 11.
[49] CPU used is a 9th Generation “Coffee Lake” 2.4 GHz Intel Core i9 processor i9-9980HK.
[50] K. Fu, Z. Zhao, L. Jin, *Adv. Funct. Mater.* **2019**, *29*, 1901258.
[51] A. G. Mark, S. Palagi, T. Qiu, P. Fischer, in *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, USA **2016**, pp. 4951–4956.
[52] N. Pagliocca, K. Z. Uddin, I. A. Anni, C. Shen, G. Youssef, B. Koohbor, *Mater. Des.* **2022**, *215*, 110446.
[53] Y. Chen, T. Li, F. Scarpa, L. Wang, *Phys. Rev. Appl.* **2017**, *7*, 024012.

ADVANCED MATERIALS

Supporting Information

for *Adv. Mater.,* DOI: 10.1002/adma.202206238

Inverse Design of Mechanical Metamaterials with Target Nonlinear Response via a Neural Accelerated Evolution Strategy

Bolei Deng, Ahmad Zareei, Xiaoxiao Ding, James C. Weaver, Chris H. Rycroft, and Katia Bertoldi**

Inverse design of mechanical metamaterials with target nonlinear response via a neural accelerated evolution strategy

Bolei Deng[†],[*]
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology, Cambridge, MA 02139 and
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139

Ahmad Zareei,[*] Xiaoxiao Zhang, James C. Weaver, and Katia Bertoldi[†]
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138

Chris H. Rycroft
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 and
Computational Research Division, Lawrence Berkeley Laboratory, Berkeley, CA 94720

(Dated: August 20, 2022)

S1. GEOMETRY

In this study we consider metamaterials comprising a 10 × 8 array of hinged quadrilateral units flanked by two horizontal strips of solid material. All structures are realized by repeating a unit cell that consists of 2×2 quadrilateral units connected at their vertices by thin beam-like ligaments with a width and length of 1 mm. The geometry of each unit cell is fully described by the positions of its 12 vertices (see red dots in Fig. S1). However, because of periodicity constraints, only the position of 8 vertices can be independently prescribed. To describe such positions for the p-th design, we introduce u_p^α and v_p^α ($\alpha = 1, 2, \cdots, 8$), which denote the displacement along the x and y direction of the α-th independent vertex of the p-th unit cell from the corresponding vertex of a square with diagonal length of $a = 10$ mm (highlighted with a black dot in Fig. S1). We then assemble u_p^α and v_p^α ($\alpha = 1, 2, \cdots, 8$) into a 16 dimensional vector $\mathbf{X}_p = [u_p^1, v_p^1, \cdots, u_p^8, v_p^8]^\top$ that fully describes the geometry of the p-th unit cell. In Table S2 we report the vectors $\mathbf{X}_p$ for all geometries considered in the main text.

Designs	u_p^1	v_p^1	u_p^2	v_p^2	u_p^3	v_p^3	u_p^4	v_p^4	u_p^5	v_p^5	u_p^6	v_p^6	u_p^7	v_p^7	u_p^8	v_p^8
Fig. 1 (squares)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fig. 1 (irregular)	0.167	0.157	0.058	-0.095	-0.139	-0.149	0.168	0.164	0.080	0.109	-0.155	0.180	-0.091	0.087	0.118	-0.028
Fig. 2c(i)	-0.088	-0.111	0.109	-0.057	-0.075	0.055	-0.128	0.339	0.103	-0.145	0.151	-0.347	0.136	0.159	0.177	0.019
Fig. 2c(ii)	-0.271	-0.177	0.160	-0.134	0.450	0.013	0.002	0.011	0.450	0.023	0.056	0.038	-0.254	0.162	0.178	-0.044
Fig. 2c(iii)	-0.387	0.046	-0.171	0.225	-0.450	0.096	-0.091	0.025	0.267	0.044	-0.101	-0.356	0.176	0.046	-0.016	0.450
Fig. 2c(iv)	-0.152	-0.006	-0.035	-0.300	-0.047	0.054	0.113	0.407	0.144	0.110	0.026	-0.115	0.169	0.042	0.037	-0.176
Fig. 4d(i)	-0.019	0.159	0.100	-0.024	-0.026	0.116	0.288	0.167	0.270	0.038	-0.086	-0.500	0.268	-0.016	0.237	-0.286
Fig. 4d(ii)	0.343	-0.152	0.033	0.026	-0.383	0.062	-0.109	-0.082	-0.297	-0.078	0.179	-0.092	0.356	0.143	-0.032	0.095
Fig. 4d(iii)	0.118	-0.023	0.057	-0.459	0.182	-0.178	0.033	0.218	-0.013	-0.102	-0.030	0.225	-0.183	-0.110	0.005	-0.384
Fig. 4d(iv)	-0.273	-0.166	0.005	-0.380	0.022	-0.252	-0.321	0.202	-0.056	-0.126	0.049	-0.152	0.102	-0.183	0.078	-0.111

TABLE S1. Vectors $\mathbf{X}_p$ for all geometries considered in the main text.

[*] B.D. and A.Z. contributed equally.

[†] boleiden@mit.edu; bertoldi@seas.harvard.edu

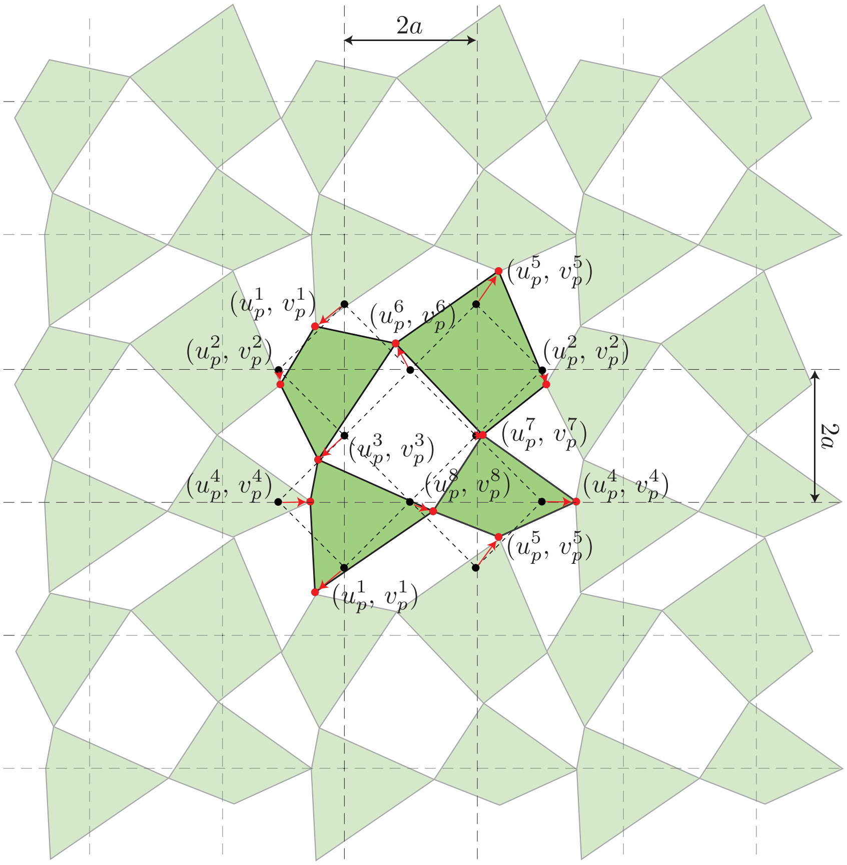

FIG. S1. Schematics of a unit-cell generated by randomly perturbing the position of the quadrilaterals vertices, while preserving periodicity.

S2. FABRICATION AND EXPERIMENTS

A. Fabrication

A molding approach is used to manufacture all our samples. First, we fabricated the negative mold using a 3D printer (Connex 500 available from Objet, Ltd.) with VeroClear (product number: RGD810, Objet) material. As shown in Fig. S2, the mold comprises: (i) a base plate with grooves; (ii) 4 side walls; and (iii) 25 pillars. The AutoCAD files used to generate the mold are openly available at `github.com/boleideng94/metamaterials-inverse-design`. To remove any caustics left behind by the 3D printing process, we clean all the components of the mold with Isopropyl alcohol. We then assemble the mold by inserting the pillars and the side walls in the grooves of the base plate. Next, we cast the structures using silicone rubber (Elite Double 32, Zhermack). Before replication, we spray a releasing agent (Easy Release 200 available from Smooth-On, Inc.) onto the molds for easy separation. We place the cast structures in a vacuum for degassing and allow them to be set at room temperature for curing for 30 min before de-molding.

B. Compression tests

For all our compression tests, we use an Instron 5969 universal testing machine with a 500 N load cell. The specimens were compressed using flat compression fixtures at a rate of 0.2 mm/s.

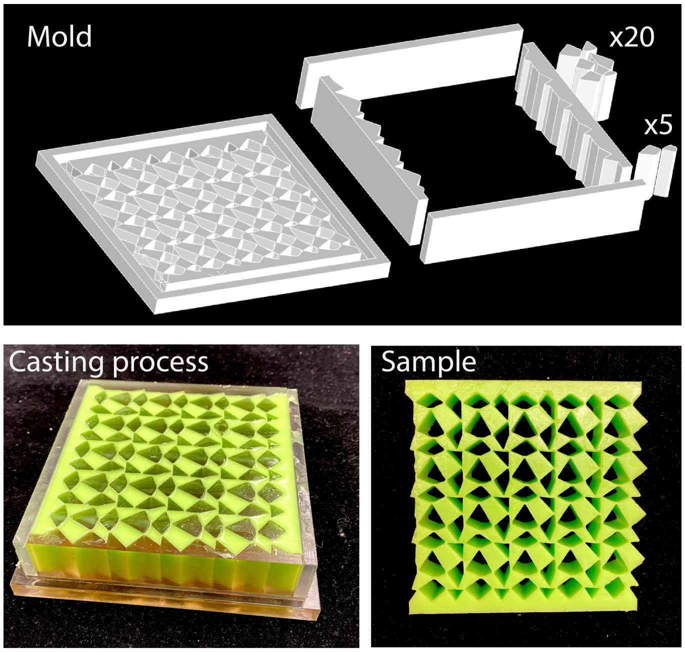

FIG. S2. Fabrication: 3D-printed molds are used to cast the metamaterials.

C. Drop tests

To investigate the energy-absorbing performance upon impact of our inversely designed metamaterials, we conduct drop tests on the optimized structures reported in Fig. 4 of the main text. More specifically, we drop a circular acrylic disk (with mass $m_{\text{disk}} = 100$ g and diameter $d_{\text{disk}} = 100$ mm) onto their top surface from an height $h_0 = 40$ cm (Fig. S3-a) and guide its trajectory via a channel built out of three acrylic plates. The tests are monitored with a high-speed camera (iPhone 11 SLO-MO) recording at 240 frames per second. From the recorded videos we then extract the contact time, t_{contact} as well as the height of the first bounce after the impact, h_1, from which we calculate the absorption efficiency as

$$\eta = \frac{m_{\mathrm{disk}} g h_0 - m_{\mathrm{disk}} g h_1}{m_{\mathrm{disk}} g h_0} = \frac{h_0 - h_1}{h_0}, \tag{S1}$$

where $g = 9.81$ m/s denotes the gravitational acceleration. As shown in Fig. 5 of the main text and Fig. S3-b, we find that $t_{contact} = 33$, 75, 54, and 96 ms and h_1= 22.0, 15.3, 11.5 and 2.5 cm for the four considered structures, resulting in an absorption efficiency η=45%, 62%, 71% and 94%, respectively.

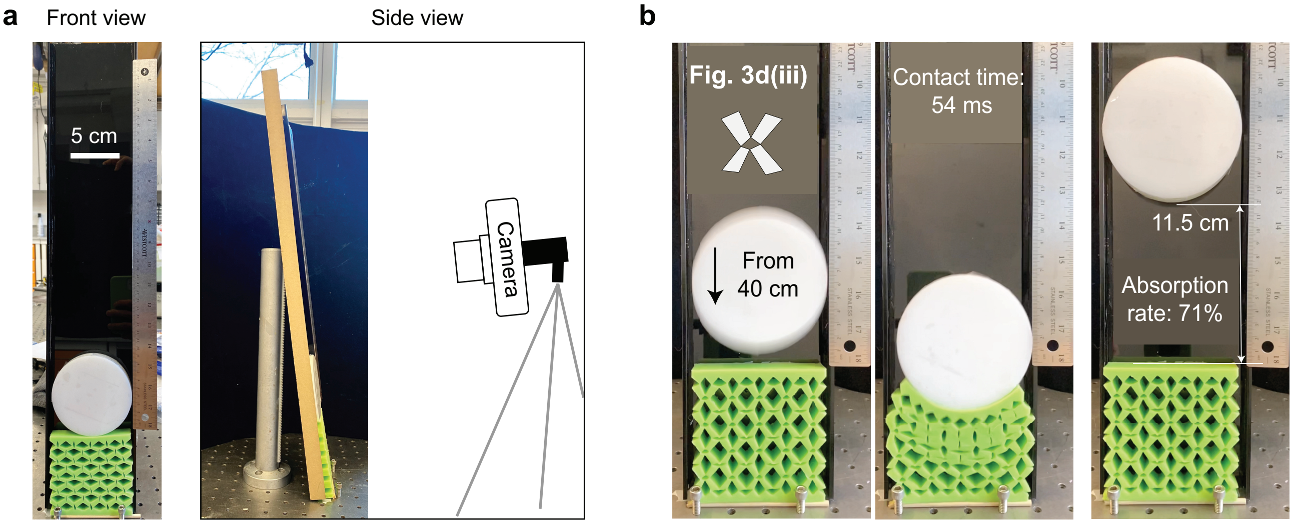

FIG. S3. Drop tests. (a) Experimental setup. (b) Experimental results for the design reported in Figure 3c(iii) of the main text.

S3. DISCRETE MODEL

Since our experiments indicate that under uniaxial compression the quadrilateral units remain rigid and the deformation localizes at the hinges, the nonlinear behavior of the considered metamaterials can be described using a discrete model comprising rigid units connected at their vertices by springs [1, 2] (see Fig. S4).

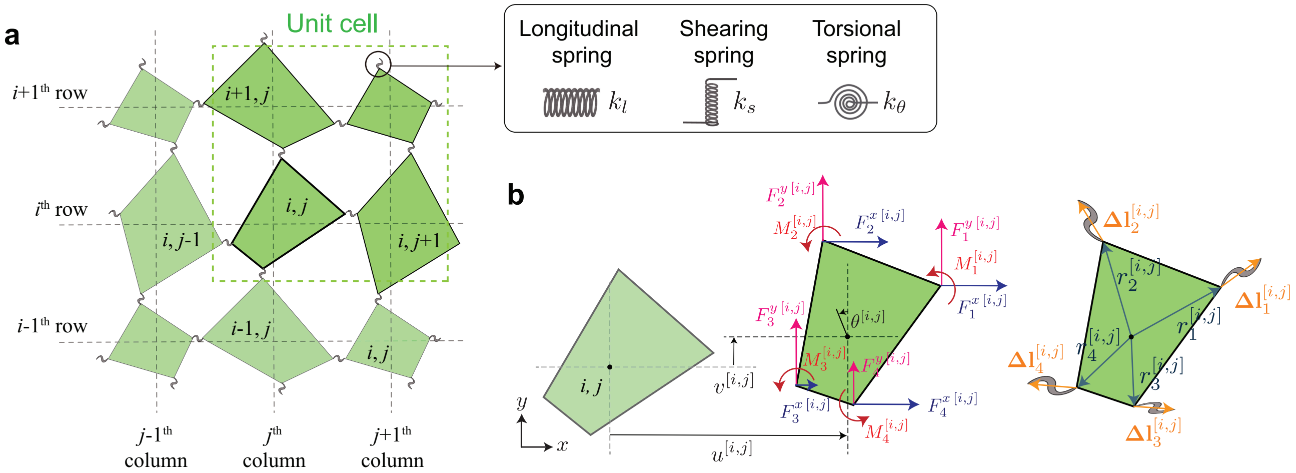

FIG. S4. (a) Discrete model based on rigid units connected at their vertices by springs. (b) Schematic of the $[i,\, j]$-th rigid square unit.

In our discrete model, we consider the quadrilaterals to be rigid and assign them three degrees of freedom: the displacement in the x-direction, u, the displacement in the y-direction, v, and the rotation around the z-axis, θ. As for the hinges, we model them using a combination of three springs (see Fig. S4a): their longitudinal response is captured by a linear spring with stiffness k_l; their shearing is captured by a linear spring with stiffness k_s and their bending is captured by a non-linear torsional spring, which obeys [2]

$$M = k_\theta(\theta + \gamma\theta^3), \tag{S2}$$

where M is the torque exerted by the spring, k_θ is the rotational stiffness and γ is a dimensionless material parameter. Note that, to facilitate the analysis, we assume that the longitudinal and shearing springs are always parallel either to the x or y axis (an assumption that is valid only for small global rotations of the system). Further, since all our samples comprise ligaments of identical shape, we assume that the spring stiffnesses do not vary as a function of the shape of the quadrilaterals.

Finally, since upon compression the edges of neighboring quadrilaterals may get in contact with each other and such contact affects the nonlinear response of the structure, we introduce an additional nonlinear rotational spring to capture contact with stiffness

$$k_{\text{cont}}(\beta) = \begin{cases} k^0_{\text{cont}}, \ \text{if } \beta < \beta_0 \\ 0, \ \text{if } \beta > \beta_0 \end{cases} \tag{S3}$$

where β denotes the angle between two edges of the neighboring quadrilaterals, which can be determined as a function of the geometric parameters (see Fig. S5). Note that this nonlinear spring is activated only when the angle β is smaller than a threshold value β_0 and induce and additional moment $M^{[i,j]}_{\text{cont},\,p=1}$ at the p-th vertex of the $[i,\, j]$-th unit. For example, if we focus on the contact between the $[i,\, j]$-th and $[i,\, j+1]$-th quadrilaterals, we find that

$$M^{[i,j]}_{\text{cont},\,p=1} = -M^{[i,j+1]}_{\text{cont},\,p=3} = -(\beta_1 - \beta_0)\, k_{\text{cont}}(\beta_1) + (\beta_2 - \beta_0)\, k_{\text{cont}}(\beta_2) \tag{S4}$$

where β_1 and β_2 denote the two angles between the edges of the two quadrilaterals (see Fig. S5).

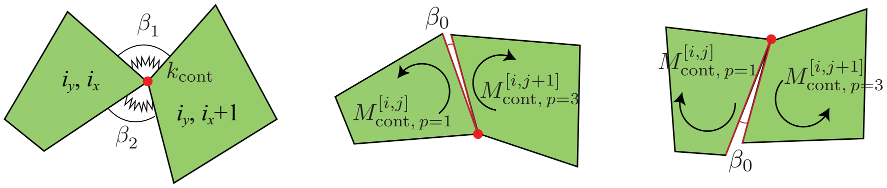
β_1
i_y, i_x
k_{cont}
i_y, i_x+1
β_2
β_0
$M^{[i,j]}_{\mathrm{cont},\,p=1}$
$M^{[i,j+1]}_{\mathrm{cont},\,p=3}$
$M^{[i,j]}_{\mathrm{cont},\,p=1}$
$M^{[i,j+1]}_{\mathrm{cont},\,p=3}$
β_0

FIG. S5. Contact: a nonlinear rotational spring is introduced to capture contact.

Under the assumptions listed above, the equations of motion for the $[i,\, j]$-th quadrilateral are given by

$$
\begin{aligned}
m^{[i,\,j]}\ddot{u} &= \sum_{p=1}^{4} F_p^{x\ [i,\,j]},\\
m^{[i,\,j]}\ddot{v} &= \sum_{p=1}^{4} F_p^{y\ [i,\,j]},\\
J^{[i,\,j]}\ddot{\theta} &= \sum_{p=1}^{4} M_p^{\ [i,\,j]},
\end{aligned}
\tag{S5}
$$

where $m^{[i,\,j]}$ and $J^{[i,\,j]}$ are the mass and moment of inertia of the $[i,\, j]$-th rigid unit. Note that in this study, since we only consider quasi-static deformations, we assume $m^{[i,\,j]} = 1$ g and moment of inertia $J^{[i,\,j]} = \frac{1}{6} m^{[i,\,j]} (\frac{a}{\sqrt{2}})^2 = 8.33$ g·mm^2 for all units. Moreover, $F_p^{x\ [i,\,j]}$ and $F_p^{y\ [i,\,j]}$ are the forces in the x-direction and the y-direction generated at the p-th vertex of the $[i,\, j]$-th unit by the springs and $M_p^{[i,\,j]}$ represents the corresponding moment (see Fig. S4b). Specifically, these forces and moments are given by

$$
\begin{aligned}
F_p^{x\ [i,\,j]} &= \mathbf{k}_p \cdot \mathbf{\Delta l}_p^{[i,\,j]} \cdot \hat{\mathbf{e}}_x,\\
F_p^{y\ [i,\,j]} &= \mathbf{k}_p \cdot \mathbf{\Delta l}_p^{[i,\,j]} \cdot \hat{\mathbf{e}}_y,\\
M_p^{\ [i,\,j]} &= -k_\theta \Delta\theta_p^{[i,\,j]} - \mathbf{r}_p^{[i,\,j]}(\theta^{[i,\,j]}) \times \left(\mathbf{k}_p \cdot \mathbf{\Delta l}_p^{\ [i,\,j]}\right) \cdot \hat{\mathbf{e}}_z + M^{[i,j]}_{\mathrm{cont},\,p},
\end{aligned}
\tag{S6}
$$

with

$$
\mathbf{k}_p = \begin{bmatrix} k_l & 0 \\ 0 & k_s \end{bmatrix}, \quad \text{for } p = 1, 3,
\tag{S7}
$$

and

$$
\mathbf{k}_p = \begin{bmatrix} k_s & 0 \\ 0 & k_l \end{bmatrix}, \quad \text{for } p = 2, 4.
\tag{S8}
$$

Furthermore, $\Delta\theta_p^{[i,\,j]}$ is the change in angle experienced by the rotational spring connected to the p-th vertex of the $[i,\, j]$-th rigid unit

$$
\begin{aligned}
\Delta\theta_1^{[i,\,j]} &= \theta^{[i,\,j]} + \theta^{[i,\,j+1]},\\
\Delta\theta_2^{[i,\,j]} &= \theta^{[i,\,j]} + \theta^{[i+1,\,j]},\\
\Delta\theta_3^{[i,\,j]} &= \theta^{[i,\,j]} + \theta^{[i,\,j-1]},\\
\Delta\theta_4^{[i,\,j]} &= \theta^{[i,\,j]} + \theta^{[i-1,\,j]}.
\end{aligned}
\tag{S9}
$$

Moreover, $\mathbf{r}_p^{[i,\,j]}$ denotes the vector that connects the center of the $[i,\, j]$-th rigid unit to its p-th vertex at a specific rotation angle $\theta^{[i,\,j]}$ (see Fig. S4b)

$$
\mathbf{r}_p^{[i,\,j]}(\theta^{[i,\,j]}) = \mathbf{R}(\theta^{[i,\,j]})\, \mathbf{r}_p^{[i,\,j]}(0)
\tag{S10}
$$

where

$$
\mathbf{R}(\theta^{[i,\,j]}) = \begin{bmatrix} \cos\theta^{[i,\,j]} & \sin\theta^{[i,\,j]} \\ -\sin\theta^{[i,\,j]} & \cos\theta^{[i,\,j]} \end{bmatrix}
\tag{S11}
$$

and $\mathbf{r}_p^{[i,\,j]}(0)$ is the initial value of $\mathbf{r}_p^{[i,\,j]}(\theta^{[i,\,j]})$ at $\theta^{[i,\,j]}=0$ and is determined by the specific shape of the $[i,\,j]$-th quadrilateral. Note that for the considered unit cell $\mathbf{r}_p^{[i,\,j]}(0)$ can be expressed as

$$\begin{aligned}
&\mathbf{r}_1^{[i,\,j]}(0)=[\frac{a}{2}+u_p^8,\,v_p^8], &&\mathbf{r}_2^{[i,\,j]}(0)=[u_p^3,\,\frac{a}{2}+v_p^3],\\
&\mathbf{r}_3^{[i,\,j]}(0)=[-\frac{a}{2}+u_p^4,\,v_p^4], &&\mathbf{r}_4^{[i,\,j]}(0)=[u_p^1,\,-\frac{a}{2}+v_p^1]\\
&\mathbf{r}_1^{[i,\,j+1]}(0)=[\frac{a}{2}+u_p^4,\,v_p^4], &&\mathbf{r}_2^{[i,\,j+1]}(0)=[u_p^7,\,\frac{a}{2}+v_p^7],\\
&\mathbf{r}_3^{[i,\,j+1]}(0)=[-\frac{a}{2}+u_p^8,\,v_p^8], &&\mathbf{r}_4^{[i,\,j+1]}(0)=[u_p^5,\,-\frac{a}{2}+v_p^5]\\
&\mathbf{r}_1^{[i+1,\,j]}(0)=[\frac{a}{2}+u_p^6,\,v_p^6], &&\mathbf{r}_2^{[i+1,\,j]}(0)=[u_p^1,\,\frac{a}{2}+v_p^1],\\
&\mathbf{r}_3^{[i+1,\,j]}(0)=[-\frac{a}{2}+u_p^2,\,v_p^2], &&\mathbf{r}_4^{[i+1,\,j]}(0)=[u_p^3,\,-\frac{a}{2}+v_p^3]\\
&\mathbf{r}_1^{[i+1,\,j+1]}(0)=[\frac{a}{2}+u_p^2,\,v_p^2], &&\mathbf{r}_2^{[i+1,\,j+1]}(0)=[u_p^5,\,\frac{a}{2}+v_p^5],\\
&\mathbf{r}_3^{[i+1,\,j+1]}(0)=[-\frac{a}{2}+u_p^6,\,v_p^6], &&\mathbf{r}_4^{[i+1,\,j+1]}(0)=[u_p^7,\,-\frac{a}{2}+v_p^7]
\end{aligned} \tag{S12}$$

where u_p^α and v_p^α defines as in Fig. S1.

Finally, $\mathbf{\Delta l}_p^{[i,\,j]}$ is a vector whose entries provide the change in length along the x- and y- directions of the linear springs connected to the p-th vertex,

$$\begin{aligned}
\mathbf{\Delta l}_1^{[i,\,j]}&=\left(u^{[i,\,j+1]}-u^{[i,\,j]}\right)\hat{\mathbf{e}}_x+\left(v^{[i,\,j+1]}-v^{[i,\,j]}\right)\hat{\mathbf{e}}_y+\mathbf{\Delta r}_3^{[i,\,j+1]}-\mathbf{\Delta r}_1^{[i,\,j]},\\
\mathbf{\Delta l}_2^{[i,\,j]}&=\left(u^{[i+1,\,j]}-u^{[i,\,j]}\right)\hat{\mathbf{e}}_x+\left(v^{[i+1,\,j]}-v^{[i,\,j]}\right)\hat{\mathbf{e}}_y+\mathbf{\Delta r}_4^{[i+1,\,j]}-\mathbf{\Delta r}_2^{[i,\,j]},\\
\mathbf{\Delta l}_3^{[i,\,j]}&=\left(u^{[i,\,j-1]}-u^{[i,\,j]}\right)\hat{\mathbf{e}}_x+\left(v^{[i,\,j-1]}-v^{[i,\,j]}\right)\hat{\mathbf{e}}_y+\mathbf{\Delta r}_1^{[i,\,j-1]}-\mathbf{\Delta r}_3^{[i,\,j]},\\
\mathbf{\Delta l}_4^{[i,\,j]}&=\left(u^{[i-1,\,j]}-u^{[i,\,j]}\right)\hat{\mathbf{e}}_x+\left(v^{[i-1,\,j]}-v^{[i,\,j]}\right)\hat{\mathbf{e}}_y+\mathbf{\Delta r}_2^{[i-1,\,j]}-\mathbf{\Delta r}_4^{[i,\,j]},
\end{aligned} \tag{S13}$$

with

$$\mathbf{\Delta r}_p^{[i,\,j]}=\mathbf{r}_p^{[i,\,j]}(\theta^{[i,\,j]})-\mathbf{r}_p^{[i,\,j]}(0). \tag{S14}$$

For a structure comprising N_y rows and N_x columns of quadrilaterals, substitution of Eqs. (S6)-(S14) into Eqs. (S6) yields a set of $3N_xN_y$ ordinary differential equations that we numerically solve using the 4th order Runge-Kutta method (via the Matlab function `ode45`). Although our samples comprise a 10×8 array of hinged quadrilateral units flanked by two horizontal strips of solid material, in all our simulations we consider an array of 10×10 quadrilaterals and fix the degrees of freedom of those forming the top and bottom row. Specifically, at the bottom row we impose

$$u^{[1,j]}=0,\ v^{[1,j]}=0,\ \text{and}\ \theta^{[1,j]}=0,\ \forall j \tag{S15}$$

whereas at the top row we prescribe

$$u^{[10,j]}=0,\ v^{[10,j]}=c_{\text{top}}\,t,\ \text{and}\ \theta^{[10,j]}=0,\ \forall j \tag{S16}$$

where $c_{\text{top}}=1$ mm/s (note that at this slow loading rate, no significant dynamic effects are observed). Finally, to suppress high frequency vibrations, we introduce a viscous damping term with a linear damping parameter $\eta=0.05$ (note that for the considered slow loading rate the value of such damping coefficient does not affect the measured stress-strain curve).

A. Estimation of k_l, k_s, k_θ, γ, k^0_{cont} and β_0

To connect the discrete model to our physical samples, we need to estimate k_l, k_s, k_θ, γ, k^0_{cont} and β_0. To this end, we focus on a uniaxial compression test conducted on a sample comprising an array of squares with diagonal $a=10$ mm, connected at the vertices by thin beam-like ligaments with a width and length of 1 mm. Remarkably, each of the parameters entering in our model control a specific feature of the stress-strain obtained for such sample (see Fig. S6a):

- k_l controls the slope of the initial linear response;
- k_θ controls the strain at which buckling occurs;
- γ controls the slope of the stress-strain curve after buckling;
- β_0 controls the strain at which contact-induced stiffening kicks in;
- k^0_{cont} controls the slope of the stress-strain curve in the post-contact regime.

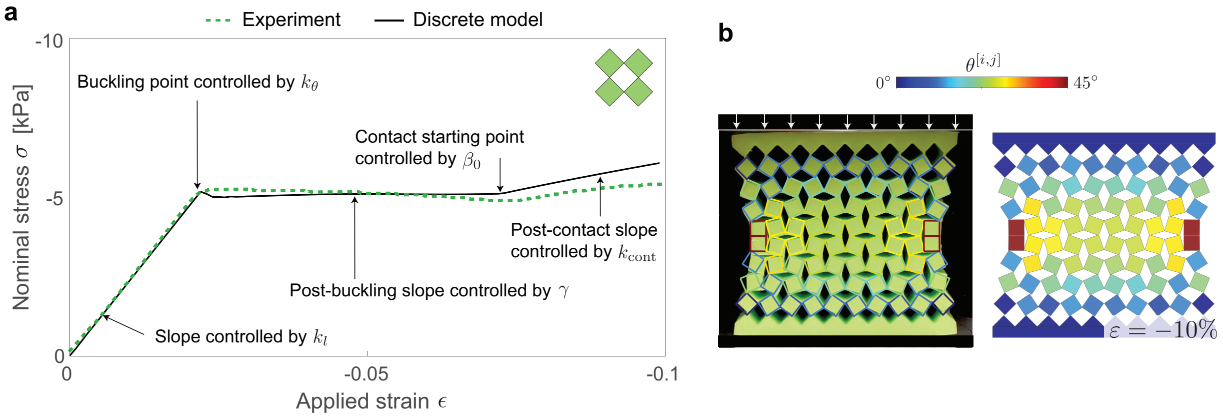

FIG. S6. (a) Each of the parameters entering in our model control a specific feature of the stress-strain obtained for such sample. Stress-strain curve for a sample comprising identical squares with center-to- center distance $a = 10$ mm as measured in experiments (green dashed line) and predicted by the model (black line). (b) Comparison between experimental and numerical snapshots at $\varepsilon = -0.1$ for a sample sample comprising identical squares with center-to-center distance $a = 10$ mm. To facilitate the comparison, the outlines of the squares from simulation are superimposed on the experimental pictures.

We find that the experimentally obtained stress-strain curve for a sample comprising identical squares with center-to- center distance $a = 10$ mm is nicely captured by the discrete model when choosing $k_l = 0.47$ N/mm, $k_\theta = 0.88$N·m, $\gamma = -0.2$, $\beta_0 = 0.1$ rad (5.7°) and $k^0_{\text{cont}} = 0.024$ N/mm. However, it is important to note that such stress-strain curve does not allow us to estimate k_s (as the shearing in the ligament for the considered structure upon compression is negligible). To overcome this limitation, we assume that

$$\frac{k_s}{k_l} = \frac{G}{E} = \frac{1}{2(1+\nu)} = \frac{1}{3}, \tag{S17}$$

where E is the initial Young's modulus of the elastomeric material, $G = E/(2+2\nu)$ is its initial shear modulus and $\nu = 0.5$ is its Poisson's ratio. It follows from Eq. (S17) that $k_s = k_l/3 = 0.16$ N/m.

Finally, we note that, when using the values of the parameters estimated above, the discrete model not only captured the stress-strain curve of the metamaterials based on squares (Fig. S6a) but also its deformation field (Fig. S6b). Further, as shown in the main text, the model can be used to predict the nonlinear response of structures realized out of arbitrary quadrilaterals.

S4. DATASET GENERATION

The machine learning model needs to have enough training data in order to accurately predict the nonlinear response of metamaterials realized out of arbitrary unit cells. To generate such data, we first produce metamaterial geometries using a two-step random sampling method and then evaluate their mechanical response using our discrete model.

In this section, we first describe two different sampling approaches (direct random sampling and two-step random sampling) and then show that the two-step sampling method leads to a more diverse dataset.

A. Direct random sampling

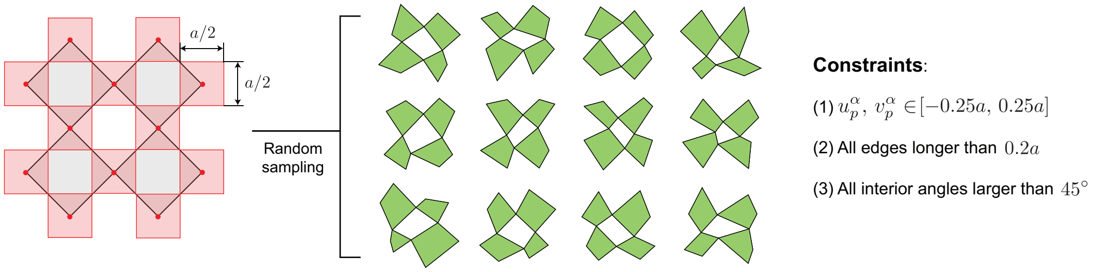
$a/2$
$a/2$
Random sampling
Constraints:
(1) u_p^α, $v_p^\alpha \in [-0.25a,\, 0.25a]$
(2) All edges longer than $0.2a$
(3) All interior angles larger than $45°$

FIG. S7. Direct random sampling. Unit cell geometries are generated by randomly moving their eight independent vertices within boxes with edge of $0.5a$ centred around them.

To generate metamaterial designs using the direct random sampling approach, we start with a unit cell comprising four hinged squares with a diagonal length of a and randomly move its eight independent vertices within square boxes with edges of $0.5a$ centered around them (see red boxes in Fig. S7). Note that, since all the vertices are bounded in boxes that do not overlap, interpenetrations of the quadrilaterals are prevented. Further, we impose that in each generated design (1) all interior angles are larger than 45 degrees and (2) all edges are longer than $0.2a$. These two constraints are introduced to ensure that the response of the metamaterials can be accurately captured by our discrete model and are determined by conducting Finite Element (FE) simulations. In our FE analysis (conducted using the commercial package Abaqus/Standard), (*i*) we consider two identical half rhombic units connected by a thin beam-like ligament with width and length of 1 mm (see Fig. S8-a); (*ii*) we assume plane stress conditions; (*iii*) we mesh the models using hybrid quadratic triangular elements (Abaqus element type: CPS8); (*iv*) we use nearly incompressible Neo-Hookean model material with initial shear modulus $\mu = 0.4$ MPa to capture the material response. To determine the spring stiffness k_l and k_s we apply an horizontal displacement δ_l and a vertical displacement δ_s to the two boundaries of our model, respectively (see insets in Figs. S8-b and S8-c). The stiffness is then calculated from the measured reaction forces F_l and F_s (given by the sum of all reaction forces at the nodes located on one of the two boundaries) as

$$k_l = \frac{F_l}{2\delta_l}, \quad k_s = \frac{F_s}{2\delta_s}. \tag{S18}$$

As for the stiffness k_θ, the two rhombic units are loaded by rotating the the two boundaries of our model by an angle $\Delta\theta$. The stiffness k_θ is then calculated from the resulting moment M_t as

$$k_\theta = \frac{M_t}{2\Delta\theta}. \tag{S19}$$

In Figs. S8b-d we report the numerically predicted evolution of k_l, k_s and k_θ as a function of the internal angle of the rhombic units. We find that the spring stiffnesses are relatively constant for large internal angles, but decrease noticeably for internal angles smaller than 45°. As such, by introducing the constraint *all internal angle* $> 45°$, we can simplify the model and keep k_l, k_s and k_θ constant as we vary the shape of the units.

Finally, in Fig. S8-e we report the stress distribution near the thin ligament for two rhombic units with an internal angle of 90° loaded by applying a vertical displacement to the two boundaries of our model (as shown in Fig. S8-c). As expected, we find that stress concentrates near the hinge and quickly decrease in the units. More specifically, we find that at a distance $0.2a$ from the hinge, the stress is approximately 15% of the maximum stress recorded in the ligaments. As such, to minimize the coupling between the ligaments we choose all edges to be longer than $0.2a$.

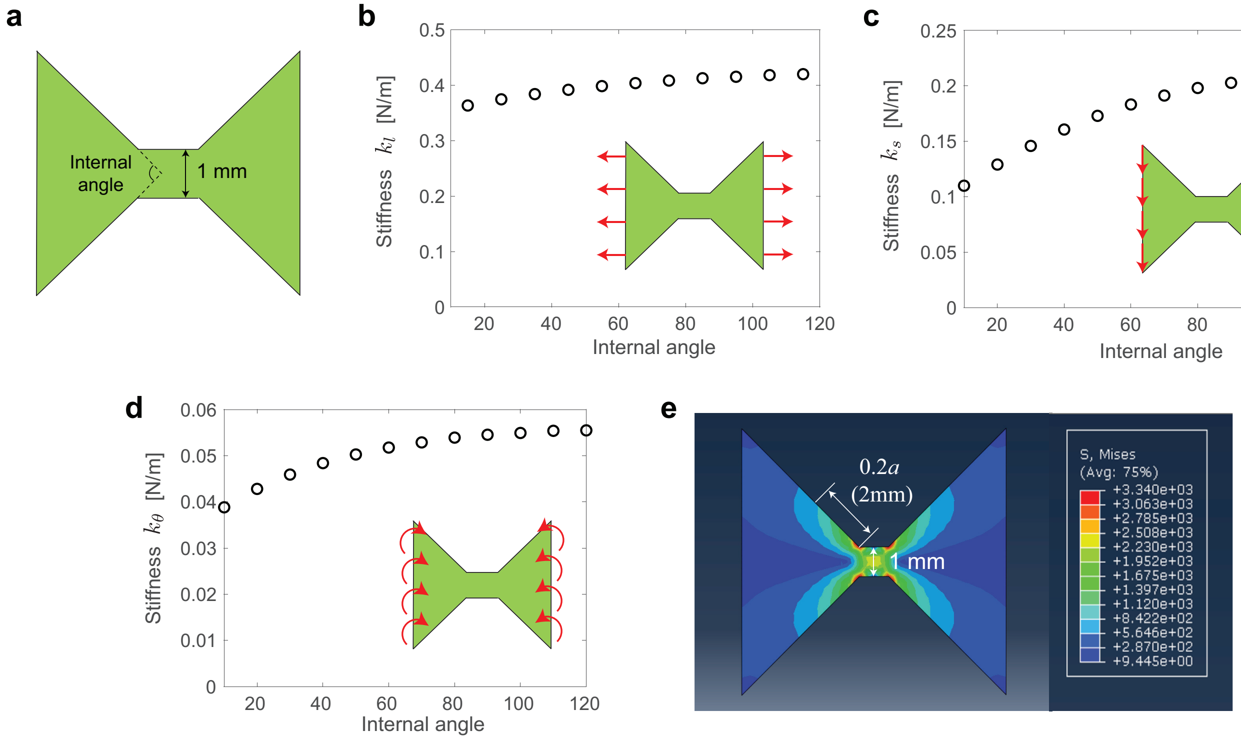

FIG. S8. (a) Schematics of the model used to determine k_l, k_s and k_θ as a function of the units' internal angle. (b)-(d) Numerically predicted evolution of (b) k_l, (c) k_s and (d) k_θ as a function of the internal angle of the rhombic units. Schematics highlighting the boundary conditions used in our FE simulations to determine the spring stiffness are shown as insets. (e) Von Mises stress distribution near the thin ligament for two rhombic units with an internal angle of 90° loaded by applying a vertical displacement to the two boundaries of our model (as shown in panel (c)).

B. two-step random sampling

Using the direct sampling method, a large number of geometries must be generated to produce highly symmetric designs, which typically lead to extreme stress-strain curves. To identify more efficiently such highly symmetric geometries, we implement a two-step sampling method. This algorithm consists of two steps (see Fig. S9)

- *Step 1:* we start with a unit cell comprising four hinged squares with diagonal length of a and randomly perturb its vertices, while constraining it be symmetric about the x and y axes. Such constraints can be enforced by imposing

$$\begin{aligned} &u_p^1 = -u_p^5,\ \ u_p^3 = -u_p^7,\ \ v_p^2 = -v_p^4,\ \ v_p^6 = -v_p^8, \\ &u_p^2 = u_p^4 = u_p^6 = u_p^8 = v_p^1 = v_p^3 = v_p^5 = v_p^7 = 0. \end{aligned} \tag{S20}$$

  Eq. (S20) reduces to four the number of vertex coordinates that can be independently prescribed. These four coordinates are obtained by randomly assigning to u_p^5, u_p^7, v_p^4 and v_p^8 values in the range $-0.25a$, $0.25a$ (so that the vertices are always within boxes with edge of $0.5a$). Note that in this study we generate 250 highly symmetric and arbitrary geometries (see blue designs in Fig. S9 for representative examples).

- *Step 2:* Each of the 250 highly symmetric geometries generated in *Step 1* is perturbed by randomly moving its eight independent vertices within square boxes with edges of $0.5a$ centered around them (see red boxes highlighted for one design in Fig. S9). Note that each highly symmetric design is perturbed 50 times (leading to a total of 7500 geometries) and that, to generate designs for which the deformation localize at the hinges, we impose that each quadrilateral in the unit cell has all edges longer than $0.2a$ and all interior angles larger than 45°.

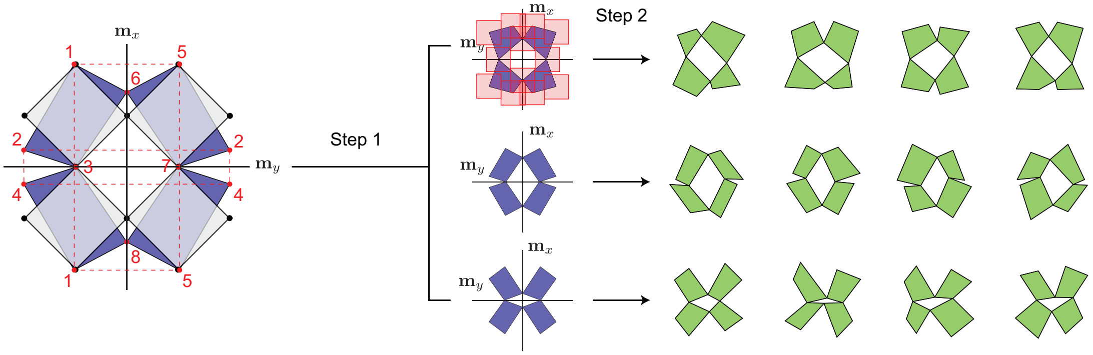

FIG. S9. Two-step random sampling. *Step 1*: We randomly generate unit cells that are symmetric about the x and y axes. *Step 2*: We perturb the highly symmetric designs from Step 1 by randomly moving their eight independent vertices.

C. Comparison between two sampling methods

In Fig. S10 we report the histogram of the maximum stress upon compression numerically predicted for 7500 metamaterials generated using the direct random sampling method (blue) and two-step random sampling method (red). We find that there is more variation in the maximum stress values for the geometries generated using the two-step random sampling method. More specifically, we calculate a standard deviation of 2.40 for these geometries, while for those generated using the direct sampling method we obtain a standard deviation of 1.81. As such, in this study, we use the two-step sampling method to generate the geometries used to populate the dataset, since it leads to more diverse data.

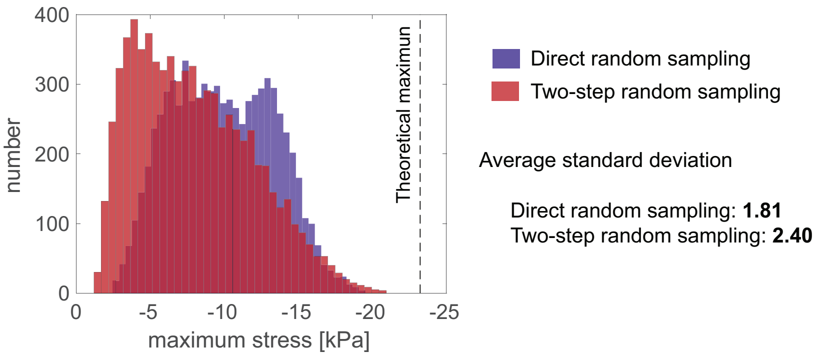

FIG. S10. Histogram of the maximum stress predicted upon compression for 7500 metamaterials generated using the direct random sampling method (blue) and two-step random sampling method (red).

S5. PRINCIPAL COMPONENT ANALYSIS

To facilitate training of the machine learning model, we reduce the dimensionality of the nonlinear measurements by using principal component analysis (PCA) [3]. Stress-strain curve of the p-th design is represented by a 100-dimensional vector

$$\boldsymbol{\sigma}_p = \left[\sigma_p(0), \cdots, \sigma_p(-0.1)\right]^\top, \tag{S21}$$

which contains the stress values at 100 equally spaced strain points. We first assemble the stress vectors of the training samples into a stress matrix

$$\boldsymbol{\sigma} = \left[\boldsymbol{\sigma}_1 \; \boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_{N_{\mathrm{tr}}}\right]^\top, \tag{S22}$$

and mean-center it to obtain

$$\hat{\sigma}_{p,s} = \sigma_{p,s} - \bar{\sigma}_s, \tag{S23}$$

where $\sigma_{p,s}$ indicates the s-th component of the p-th stress vector and

$$\bar{\sigma}_s = \frac{1}{N_{\mathrm{tr}}} \sum_{p=1}^{N_{\mathrm{tr}}} \sigma_{p,s}, \tag{S24}$$

N_{tr} denoting the number of datapoints used for training. Then, we perform singular value decomposition on the resulting covariance matrix and rewrite it as

$$\frac{\hat{\boldsymbol{\sigma}}^\top \hat{\boldsymbol{\sigma}}}{N_{\mathrm{tr}} - 1} = \mathbf{V} \boldsymbol{\Sigma}^{\mathbf{2}} \mathbf{V}^\top, \tag{S25}$$

where $\mathbf{V}$ is a 100×100 matrix with orthonormal columns (i.e. $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$, with $\mathbf{I}$ denoting the identity matrix) that represent the principal directions. Further, $\boldsymbol{\Sigma}^2 = \mathrm{diag}(\beta_1, \cdots, \beta_{100})$ contains the eigenvalues β_s that determine the variance explained in each principal component. Finally, to reduce the dimensionality of the data from 100 to $n < 100$, we assemble into $\mathbf{V}_n$ the columns of $\mathbf{V}$ associated to the n largest explained variances and calculate the first n principal components of $\hat{\boldsymbol{\sigma}}_p$ as

$$\boldsymbol{\sigma}_p^{\mathrm{PC}_n} = \hat{\boldsymbol{\sigma}}_p \mathbf{V}_n. \tag{S26}$$

Note that for this study we choose $n = 10$, since this leads to an average error

$$\frac{1}{N_{\mathrm{tr}}} \sum_{p=1}^{N_{\mathrm{tr}}} \mathrm{err}[\boldsymbol{\sigma}_p, \boldsymbol{\sigma}_p^R, -0.1] = 0.3\%, \tag{S27}$$

where $\boldsymbol{\sigma}_p$ and $\boldsymbol{\sigma}_p^R = \boldsymbol{\sigma}_p^{\mathrm{PC}_{10}} \mathbf{V}_{10}^\top + \bar{\boldsymbol{\sigma}}$ denote the true and reconstructed stress vectors, respectively. As a result, our training data set consists of N_{tr} pairs $\{(\mathbf{X}_p, \boldsymbol{\sigma}_p^{\mathrm{PC}_{10}})\}_{p=1,\cdots,N_{\mathrm{tr}}}$, where $\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}} = [\sigma_{p,1}^{\mathrm{PC}_{10}}, \cdots, \sigma_{p,10}^{\mathrm{PC}_{10}}]^\top$ contains the 10 first principal components describing the stress vector associated to the p-th unit cell.

S6. NEURAL NETWORK

To predict the lower-dimensional representation of the stress vector, $\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}}$, for a given geometry, $\mathbf{X}_p$, we use a neural network (NN) architecture with four hidden layers, each comprising 200 neurons. For implementation and training of the NN we use the PyTorch package [4]. Out of the expanded datasets comprising $N = 30000$ datapoints, we randomly choose $N_{\mathrm{tr}} = 0.8N$ data points for training and the remaining $N_{\mathrm{test}} = 0.2N$ for testing. We train the NN by selecting the weights and biases of the neurons that minimize

$$\mathcal{L} = \sum_{p=1}^{N_{\mathrm{tr}}} \sum_{s=1}^{10} \beta_s \left(\mathcal{F}_s(\mathbf{X}_p) - \sigma_{p,s}^{\mathrm{PC}_{10}} \right)^2, \tag{S28}$$

where $\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}}$ are the true values of the first 10 principal components use to represent the stress-strain curve associated to the p-th design, $\boldsymbol{\mathcal{F}}(\mathbf{X}_p) = [\mathcal{F}_1(\mathbf{X}_p), \cdots \mathcal{F}_{10}(\mathbf{X}_p)]^\top$ denote the corresponding NN predictions and β_s represents the variance explained in each principal component. The training is performed using the following parameters

- **Hidden layer dimensions**: 200, 200, 200, 200
- **Activation functions**: ReLU for the first three layers and no activation for the last layer
- **Feature scaling**: PCA
- **Optimization algorithm**: Adam
- **Learning rate**: 10^{-4}
- **Batch size**: 64
- **Number of epochs**: 100
- **Dropout**: None

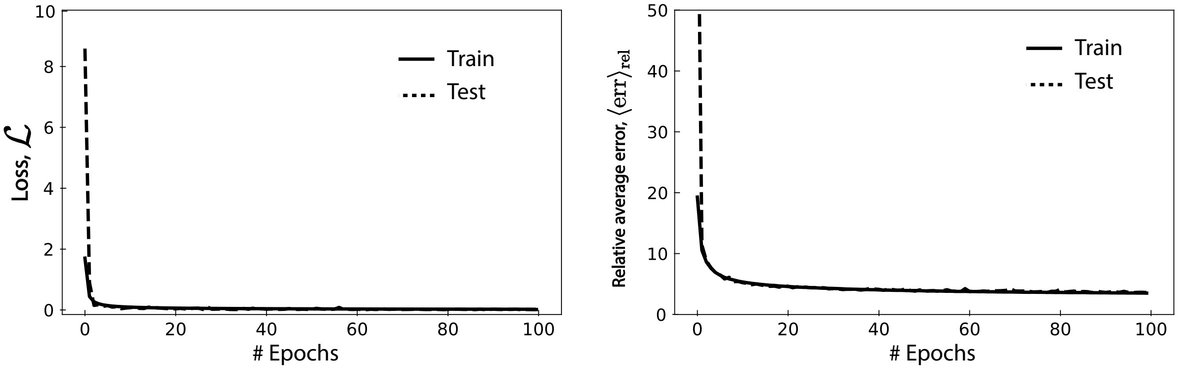

FIG. S11. (a) Evolution of $\mathcal{L}$ and (b) $\langle \mathrm{err} \rangle_{\mathrm{rel}}$ as a function of the number of epochs averaged over the whole training and testing dataset. The neural network has 200x200x200 hidden layers with ReLU activation function, optimized with a 10^{-4} learning rate with Adam optimizer and a 64 batch-size.

- **Weight Decay**: 10^{-3}

To evaluate the accuracy of our trained NN, we then calculate the average relative error on the test dataset as

$$\langle \mathrm{err} \rangle_{\mathrm{rel}} = \frac{1}{N_{\mathrm{test}}} \sum_{p=1}^{N_{\mathrm{test}}} \mathrm{err}[\boldsymbol{\sigma}_p, \boldsymbol{\sigma}_p^{\mathrm{NN}}, -0.1], \tag{S29}$$

where the err function is defined in Eq. (1) of the main text and $\boldsymbol{\sigma}_p^{\mathrm{NN}} = \boldsymbol{\mathcal{F}}(\mathbf{X}_p)\mathbf{V}_{10}^{\top} + \bar{\boldsymbol{\sigma}}$. In Fig. S11a we show the evolution of $\mathcal{L}$ and $\langle \mathrm{err} \rangle_{\mathrm{rel}}$ as a function of the number of epochs. Using the training parameters reported above we find that after 100 epochs $\langle \mathrm{err} \rangle_{\mathrm{rel}} = 3.59\%$.

Finally, in Table S2 we report the average relative error on the test data-set, $\langle \mathrm{err} \rangle_{\mathrm{rel}}$, obtained for different combinations of training parameters. We find that the average relative error obtained with the hyper-parameters used in this study cannot be largely lowered by varying the parameters. More specifically, the results indicate that (*i*) an increase in the number of hidden layers results in a small reduction of $\langle \mathrm{err} \rangle_{\mathrm{rel}}$; (*ii*) the learning rate and the batch size have small effect on $\langle \mathrm{err} \rangle_{\mathrm{rel}}$; (*iii*) a Sigmoid activation function worsen the accuracy of the network; (*iv*) the Adam optimizer performs better than SGD. Further, to ensure that there is no overfitting, we introduce

$$\mathcal{O}_l = \frac{\mathcal{L}_{\mathrm{train}}}{\mathcal{L}_{\mathrm{test}}}, \quad \text{and} \quad \mathcal{O}_e = \frac{\langle \mathrm{err} \rangle_{\mathrm{train}}}{\langle \mathrm{err} \rangle_{\mathrm{test}}}, \tag{S30}$$

where $\mathcal{L}_{\mathrm{train}}$ and $\mathcal{L}_{\mathrm{test}}$ are the loss of the proposed neural network calculated for the training and testing datasets (as defined in Eq. (S28)), $\langle \mathrm{err} \rangle_{\mathrm{train}}$ and $\langle \mathrm{err} \rangle_{\mathrm{test}}$ are the average relative error calculated for the training and testing datasets (as defined in Eq. (S29)). For a given neural network $\mathcal{O}_l$ and $\mathcal{O}_e$ can vary between zero and one. If a neural network explains the training data extremely well, but it is not able to generalize to the testing dataset (i.e., if there is overfitting), then $\mathcal{O}_l, \mathcal{O}_e \to 0$. Differently, if the network performs similarly on both training dataset and testing dataset, $\mathcal{O}_e, \mathcal{O}_l \to 1$. In Table S2 we report $\mathcal{O}_e$ and $\mathcal{O}_l$ for all considered variants of the neural network. As expected, we find that as we increase the number of layers or the neurons in a layer both $\mathcal{O}_e$ and $\mathcal{O}_l$ decrease - a signature of overfitting (e.g., for a network with 4 layers with 100, 200, or 300 neurons at each layer, respectively we have $\mathcal{O}_l = 0.78, 0.5, 0.34$ and $\mathcal{O}_e = 0.99, 0.87, 0.79$). For the neural network used in this study we have $\mathcal{O}_l = 0.5$ and $\mathcal{O}_e = 0.87$. The fact that $\mathcal{O}_e$ is close to unity clearly shows that our model generalize well to the testing dataset.

S7. INVERSE DESIGN USING NEURAL NETWORK

While in this study we use a neural accelerated evolution strategy to identify geometries resulting in target nonlinear mechanical responses, we also considered using a neural network to solve the inverse problem. More specifically, we tested two different approaches: a direct inverse design approach via neural network and an indirect inverse design approach via the forward neural network. In the following, we describe both approaches and show that they do perform as well as our neural accelerated evolution strategy.

Hidden layer dimensions	Activation function	Optimization Alg.	Learning Rate	Batch Size	$\langle \mathrm{err} \rangle_{\mathrm{rel}}$	$\mathcal{O}_l$	$\mathcal{O}_e$
200,200,200,200	**ReLU**	**Adam**	**1.00E-04**	**64**	**2.97**	**0.50**	**0.87**
100,100,100,100	ReLU	Adam	1.00E-04	64	3.59	0.78	0.99
300,300,300,300	ReLU	Adam	1.00E-04	64	2.54	0.34	0.79
300,300,300	ReLU	Adam	1.00E-04	64	2.8	0.46	0.86
200,200,200	ReLU	Adam	1.00E-04	64	3.25	0.64	0.94
100,100,100	ReLU	Adam	1.00E-04	64	3.8	0.84	1.00
200,200,200,200,200	ReLU	Adam	1.00E-04	64	2.53	0.44	0.86
200,200,200,200	Sigmoid	Adam	1.00E-04	64	20.5	1.05	1.00
200,200,200,200	ReLU	SGD	1.00E-05	64	4.9	0.92	1.04
200,200,200,200	ReLU	Adam	1.00E-03	64	3.41	0.52	0.91
200,200,200,200	ReLU	Adam	1.00E-05	64	3.77	0.84	0.99
200,200,200,200	ReLU	Adam	1.00E-04	32	3.06	0.47	0.90
200,200,200,200	ReLU	Adam	1.00E-04	128	2.84	0.54	0.89

TABLE S2. Effect of different neural network hyper-parameters on the average relative error on the test dataset.

A. Direct inverse design approach via neural network

In this approach, in addition to the neural network used to provide a relationship between the unit cell geometry and the stress-strain response, we introduce another neural network that is trained to identify a metamaterial design leading to the desired stress-strain curve. The inverse neural network is similar to that used for the forward model (four hidden layers of each 200 neurons, with ReLU activation function, Adam optimizer with a learning rate of 10^{-4}, and weight decay of 10^{-3}) and is trained by selecting the weights and biases of the neurons that minimize

$$\mathcal{L} = \sum_{p=1}^{N_{\mathrm{tr}}} \sum_{s=1}^{16} \left(\mathcal{G}_s(\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}}) - X_{p,s} \right)^2 , \tag{S31}$$

where $\boldsymbol{\mathcal{G}}(\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}}) = [\mathcal{G}_1(\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}}), \cdots, \mathcal{G}_{16}(\boldsymbol{\sigma}_p^{\mathrm{PC}_{10}})]^\top$ is the neural network prediction for the 16 geometric parameters that fully describe a unit cell. Note that also in this case, out of the expanded dataset comprising $N = 30000$ datapoints, we randomly choose $N_{\mathrm{tr}} = 0.8N$ datapoints for training of the inverse NN.

To test this approach, we focus on the four target stress-strain curves considered in Fig. 4 of the main text. We use the trained inverse neural network to identify optimal geometries and then predict the stress-strain response of these designs via our discrete model. As shown in Fig. S12, we find that the error between the stress-strain curves obtained from the optimal designs and the target ones is much larger than that between the closest matches from the dataset and the target ones. The poor performance of the inverse neural network can be attributed to the fact that multiple geometries exist that lead to the same stress-strain curve (i.e. the loss function defined in Eq. (S31) is ill-posed) [5].

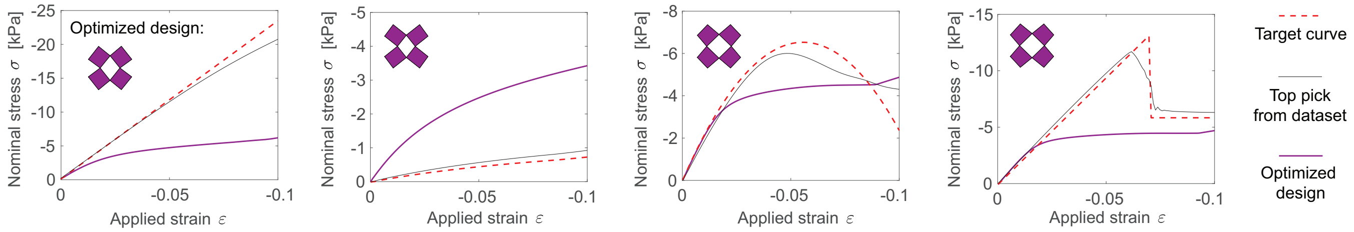

FIG. S12. Direct inverse design approach via neural network. Target response (red dashed line), best match from dataset (black lines) and optimal designs identified by the inverse NN (purple lines).

B. Indirect inverse design approach via the forward neural network

To overcome the limitation of our first approach, inspired by [5], we introduce a second neural network $\mathcal{G}$ (with comprises four hidden layers of each 200 neurons, ReLU activation function, Adam optimizer with a learning rate of 10^{-4}, and weight decay of 10^{-3}) and train it to become the inverse of our forward neural network $\mathcal{F}$. Such training is conducted by selecting the weights and biases of the neurons that minimize

$$\mathcal{L} = \sum_{p=1}^{N_{\rm tr}} \sum_{s=1}^{16} \left(\mathcal{G}_s(\boldsymbol{\mathcal{F}}(\mathbf{X}_p) - \mathbf{X}_{p,s}\right)^2 . \tag{S32}$$

To test this approach, we again focus on the four target stress-strain curves considered in Fig. 4 of the main text. We use the trained neural network $\mathcal{G}$ to identify optimal geometries and then predict their stress-strain response via our discrete model. Note that we tested neural networks of different size (keeping the size of forward, $\mathcal{F}$, and inverse, $\mathcal{G}$, neural networks identical). As shown in Fig. S13, we find that all considered neural networks (i.e. 100×100×100, 200×200×200 and 300×300×300) cannot identify designs better than the best pick from dataset. In table S3 we summarize the error between the four target curve considered in Fig. 4 of the main text and the response of the optimized designs identified using the inverse design methods employed in this study. Although the indirect inverse design performs much better than the direct one, it still fails to identify designs that are better than the best picks from dataset. Differently, the proposed neural accelerated evolution strategy is capable of find design that more closely match all four target responses, proving the performance and robustness of our method.

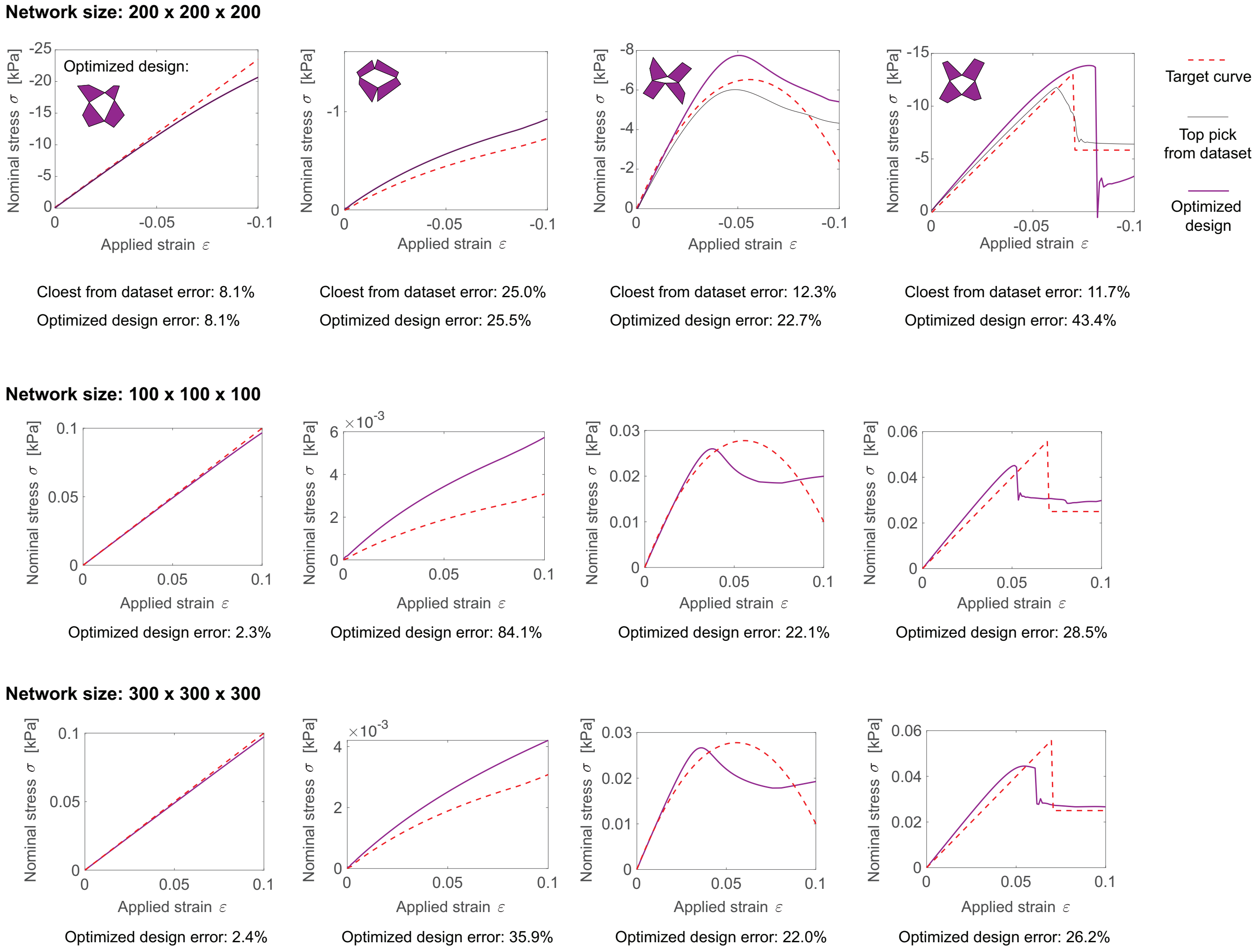

FIG. S13. Indirect inverse design approach via the forward neural network $\mathcal{F}$ and a second neural network $\mathcal{G}_s$. Target responses (red dashed line), best matches from dataset (black lines) and stress-strain curves for the optimized design identifies by the neural network $\mathcal{G}_s$ (purple lines). This approach is implemented using neural networks $\mathcal{F}$ and $\mathcal{G}_s$ of different sizes: 200×200×200 (top row), 100 × 100 × 100 (middle row) and 300 × 300 × 300 (bottom row). Note that $\mathcal{F}$ and $\mathcal{G}_s$ have always identical size.

	Fig. 4d(i)	**Fig. 4d(ii)**	**Fig. 4d(iii)**	**Fig. 4d(iv)**
Top pick from dataset (baseline)	8.1%	25.0%	12.3%	11.7%
Direct inverse design (200 × 200 × 200)	419%	72.9%	27.0%	45.8%
Indirect inverse design (100 × 100 × 100)	2.3%	84.1%	22.1%	28.5%
Indirect inverse design (200 × 200 × 200)	8.1%	25.5%	22.7%	43.4%
Indirect inverse design (300 × 300 × 300)	2.4%	35.9%	22.0%	26.2%
Neural accelerated evolution strategy (200 × 200 × 200)	**1.9%**	**4.7%**	**6.6%**	**4.5%**

TABLE S3. Minimum error between σ_t and the stress-strain curve predicted by the discrete model for the optimized geometries, σ^{DM}, for the four target curve considered in Fig. 4 of the main text using the direct inverse design methods, the indirect inverse design method (with three different networks) and the proposed neural accelerated evolution strategy. We also report the error for the top picks from the dataset.

S8. INVERSE DESIGN OF SOFT ROBOTS AND ACTUATORS

While in this study we have focused on target nonlinear stress-strain responses, our strategy can be readily applied to other non-linear properties, providing a platform to facilitate the design of soft robots and soft actuators.

A. Soft robots that exploit auxetic mechanical metamaterials

It has been shown that two flexible structures with Poisson's ratio of equal magnitude but opposite sign can lead a soft robot capable of moving through a channel when actuated with a single input rather than three independent ones [6]. More specifically, this implementation of the robot consists of a linear actuator linking two passive elastic blocks. The two blocks have width larger than that of the channel so that they grip. Further, they ideally have identical Young's moduli, and Poisson's ratios of equal magnitude but opposite sign. This means that when the actuator expands and pushes against the two blocks, the normal force exerted by the two blocks on the channel have opposite signs. The auxetic block contracts laterally and therefore applies less normal force to the channel, whereas the opposite is true for the other block with positive Poisson's ratio (see Fig. S14). Thus, during the expansion phase, the auxetic block slips forward along the channel. During the contraction phase the effect is reversed: tension causes the auxetic's normal force to increase, and the normal material's to decrease. So in this phase the normal material slips.

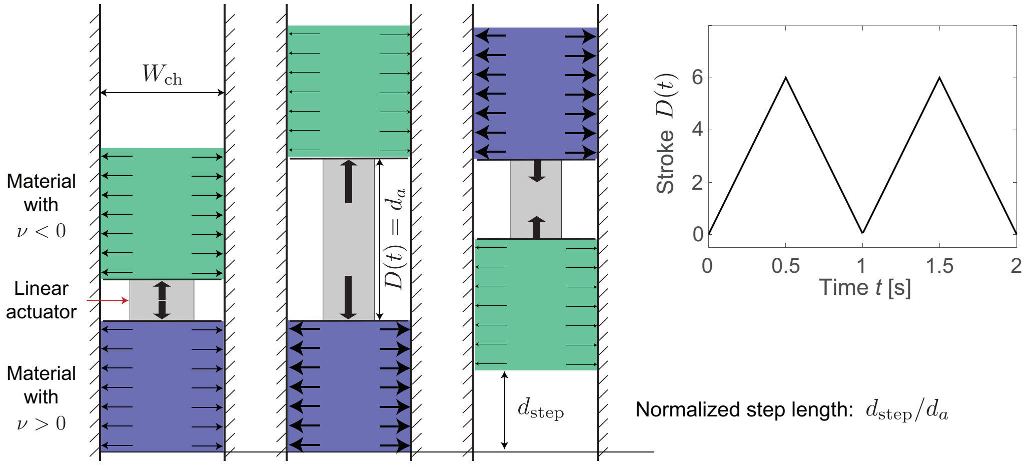

FIG. S14. A schematic of the soft robot design using auxetic mechanical metamaterials.

Here, we investigate the performance of such soft robots built using mechanical metamaterials based on hinged quadrilaterals. We first show how we characterize the response of the considered metamaterials, then present the numerical simulations that we conduct to characterize the performance of the robots and finally discuss the results.

a. Data generation We consider metamaterials comprising 3×3 identical unit cells and characterize their response by solving Eq. (S5), while imposing

$$v^{[1,j]} = 0, \text{ and } v^{[6,j]} = c_{\text{top}}\, t, \ \forall j \tag{S33}$$

and leaving the lateral boundaries traction-free. Note that in all our simulations we set $c_{\text{top}} = 1$ mm/s - a loading rate slow enough to prevent dynamic effects. An effective structural Poisson's ratio is then calculated as

$$\nu(\varepsilon) = -\frac{\varepsilon_h}{\varepsilon} = -\frac{1}{\varepsilon a N_x N_y} \sum_{i=1}^{N_y} \left(u^{[i,N_x]} - u^{[i,1]} \right) \tag{S34}$$

where ε is the applied strain in vertical direction and ε_h denote the lateral strain. In Fig. S15a we show the numerically predicted evolution of ν as a function of the applied strain for the 7500 metamaterials generated from the unit cells considered in this study. We find that a variety of nonlinear responses emerge. These include (*i*) large positive Poisson's ratio (see curve (*i*) in Fig. S15-a); (*ii*) negative Poisson's ratio (see curve (*ii*) in Fig. S15-a); (*iii*) almost vanishing Poisson's ratio (see curve (*iii*) in Fig. S15-a); (*iv*) Poisson's ratio that largely varies as a function of the applied strains (see curve (*iv*) in Fig. S15-a).

b. Numerical simulations of the robots In the numerical simulations of the robot we consider two pieces of the 3×3 unit cells metamaterials described above and connect them by imposing

$$\frac{1}{6}\sum_{j=1}^{6}\left(v_{top}^{[1,j]} - v_{bottom}^{[6,j]}\right) = D(t), \tag{S35}$$

where $D(t)$ describes the relative displacement imposed by the linear actuator, $v_{top}^{[1,j]}$ denotes the vertical displacement of the j-th unit in the bottom row for the top block and $v_{bottom}^{[6,j]}$ is the vertical displacement of the j-th unit in the top row for the bottom block. Note that in this study we choose

$$D(t) = \begin{cases} \dfrac{2d_a t}{T} & \text{if } t - (n_{\text{cycle}} - 1)T \le T/2 \\ 2d_a - \dfrac{2d_a t}{T} & \text{if } t - (n_{\text{cycle}} - 1) > T/2 \end{cases} \tag{S36}$$

where n_{cycle} is an integer that denote the number of the current actuation cycle, $T = 1$ s denotes the period of actuation and d_a represents the stroke of the actuator (see Fig. S14). Since the lateral boundaries of the two blocks are in contact with the walls of the rigid channel, we connect all units of the left most and right most columns to springs with stiffness $k_{\text{tube}} = 10k_l$. These springs introduce normal contact forces

$$N_{contact}^{[i,j]} = k_{\text{tube}}\, u^{[i,j]}, \qquad \text{for } j = 1 \text{ and } 6 \tag{S37}$$

Note that as long as $k_{\text{tube}} \gg k_l$, the specific value of k_{tube} does not affect the normal force.

Further, the friction between the metamaterials and the walls of the channel is captured by applying the following force to the units on leftmost and rightmost columns

$$f_{friction}^{[i,j]} = -\mu N_{contact}^{[i,j]} \text{sign}\left(\frac{d\,v^{[i,j]}}{d\,t}\right), \qquad \text{for } j = 1 \text{ and } 6 \tag{S38}$$

where μ is the friction coefficient between the metamaterials and the wall of the channel (in this study we use $\mu = 0.5$).

It follows that the governing equation (Eq. (S5)) become

$$\begin{aligned} m^{[i,\,j]}\ddot{u} &= N_{contact}^{[i,j]} + \sum_{p=1}^{4} F_p^{x\ [i,\,j]}, \\ m^{[i,\,j]}\ddot{v} &= f_{friction}^{[i,j]} + \sum_{p=1}^{4} F_p^{y\ [i,\,j]}, \\ J^{[i,\,j]}\ddot{\theta} &= \sum_{p=1}^{4} M_p^{\ [i,\,j]}. \end{aligned} \tag{S39}$$

Eq. (S39) are solved numerically together with Eq. (S35) to simulate the robot.

c. Results To begin with, we select from the database shown in Fig. S15-a two metamaterials characterized by ν =1 and -1 in the undeformed configuration (see yellow curves in Fig. S15-b). We then simulate the response of the robot realized by connecting these two metamaterials when placed in channels of different width W_{ch}. As shown in Fig. S15c, we find that the efficiency of the robot largely varies with the width of the channel (see yellow markers). Such variation is due to the fact that the width of the channel changes the level of lateral strain experienced by the metamaterials and this ultimately alters the effective Poisson's ratio of the metamaterials.

To overcome this limitation and realize a robot can efficiency move through channels of different width, we use our neural accelerated evolution strategy to identify two architectures with constant Poisson's ratio $\nu = \pm 1$ for $-0.1 < \varepsilon < 0.1$. As for the case of the stress-strain curve, we first train our neural network to to learn the relation between the unit cell design and the evolution of ν as a function of the applied deformation. To this end, we represent ν for the p-th unit cell in our database as 199-dimensional vector

$$\left[\nu(\varepsilon = -0.1), \cdots, \nu(\varepsilon = 0) \cdots, \nu(\varepsilon = 0.1)\right]^{\top}, \tag{S40}$$

which contains the Poisson's ratio at 199 equally spaced strain points ranging from $\varepsilon = -0.1$ to $\varepsilon = 0.1$. Then, we use the same neural network architecture introduced to learn the stress-strain curves and couple it with ES to identify

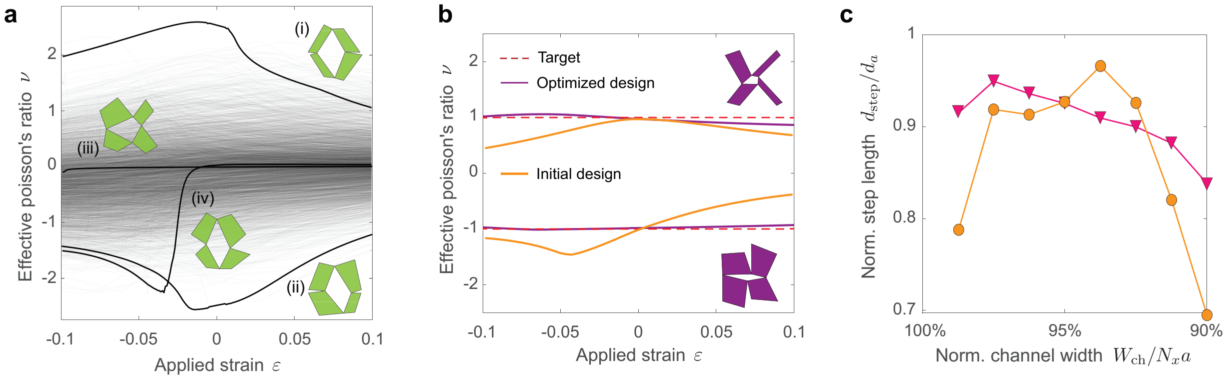

FIG. S15. (a) Effective Poisson's ratio vs strain curves predicted by the discrete model for 7500 different metamaterials realized by tessellating different unit cells. Four representative $\nu - \varepsilon$ curves are highlighted. These include (*i*) large positive Poisson's ratio; (*ii*) negative Poisson's ratio ; (*iii*) almost vanishing Poisson's ratio; (*iv*) Poisson's ratio that largely varies as a function of the applied strains. (b) Yellow lines: $\nu - \varepsilon$ curves for the two metameterials selected from our database to realize the robot. These two metamaterials are characterized by $\nu = 1$ and -1 at $\varepsilon = 0$. Red dashed lines: Target responses. Purple lines: $\nu - \varepsilon$ curves predicted by our discrete model for the optimized designs. (c) Evolution of the displacement recorded at the end of each cycle, d_{step} (normalized by the actuator stroke d_a) as a function of $W_{ch}/(N_x a)$ for the initial design (yellow markers) and the optimized design (magenta markers).

designs with the target effective Poisson's ratio evolution. In Fig. S15b we show the two unit cells identified by our numerical algorithm and their corresponding nonlinear Poisson's ratios. We find that both of them closely match the target and have Poisson's ratio very close to ±1 over the range of considered deformation. Finally, we simulate the robot realized connecting these two optimized metamaterials. As shown in Fig. S15c (magenta markers), our results indicate that the efficiency of this robot is much less affected by the width of the channel.

B. Thermally responsive soft actuators

Finally, we exploit our neural accelerated evolution strategy to identify structures capable of target deformation upon application of an external stimulus. To demonstrate this point, we consider a metamaterial comprising 8×2 unit cells (i.e. $N_x = 6$ columns and $N_y = 18$ rows of quadrilateral units) and assume that each quadrilateral unit expands uniformly under an increasing temperature (see Fig. S16a).

a. Numerical simulations To simulate such structures, we assume that the $\mathbf{r}_p^{[i,j]}(0)$ defined in Eq. (S12) vary as

$$\mathbf{r}_{p,th}^{[i,j]}(0) = (1 + \varepsilon_{\text{th}})\,\mathbf{r}_p^{[i,j]}(0), \tag{S41}$$

where $\varepsilon_{\text{th}} \in [0, 0.2]$ define the thermal expansion of the rigid unit. Additionally, in order to trigger bending deformations, we fix the center-to-center distance of the units in the leftmost column by introducing springs with stiffness $k_{\text{wire}} = 10k_l$ between the neighboring units on the left boundary. Such springs result in additional forces,

$$\begin{aligned} F_{\text{wire},x}^{[i,1]} &= k_{\text{wire}}\left(u^{[i+1,1]} - 2u^{[i,1]} + u^{[i-1,1]}\right), \\ F_{\text{wire},y}^{[i,1]} &= k_{\text{wire}}\left(v^{[i+1,1]} - 2v^{[i,1]} + v^{[i-1,1]}\right), \end{aligned} \tag{S42}$$

which are added to the governing equations (Eq. (S5)). To simulate the response of the structures upon heating, we then solve the governing equations for increasing values of ε_{th} (we monotonically increase ε_{th} from 0 to 0.2).

b. Data generation We simulate the response of the 7500 structures realized by tessellating the 7500 unit cells considered in this study. From each simulation we then extract the average displacement in x-direction of the i-th row of units for increasing values of ε_{th}

$$< u_i >^{\varepsilon_{\text{th}}} = \frac{1}{N_x}\sum_{j=1}^{N_x} u^{[i,j]}|_{\varepsilon_{\text{th}}} \tag{S43}$$

where $u^{[i,j]}|_{\varepsilon_{\text{th}}}$ indicates the numerically predicted displacements in x-direction of the $[i, j]$-th unit at ε_{th}.

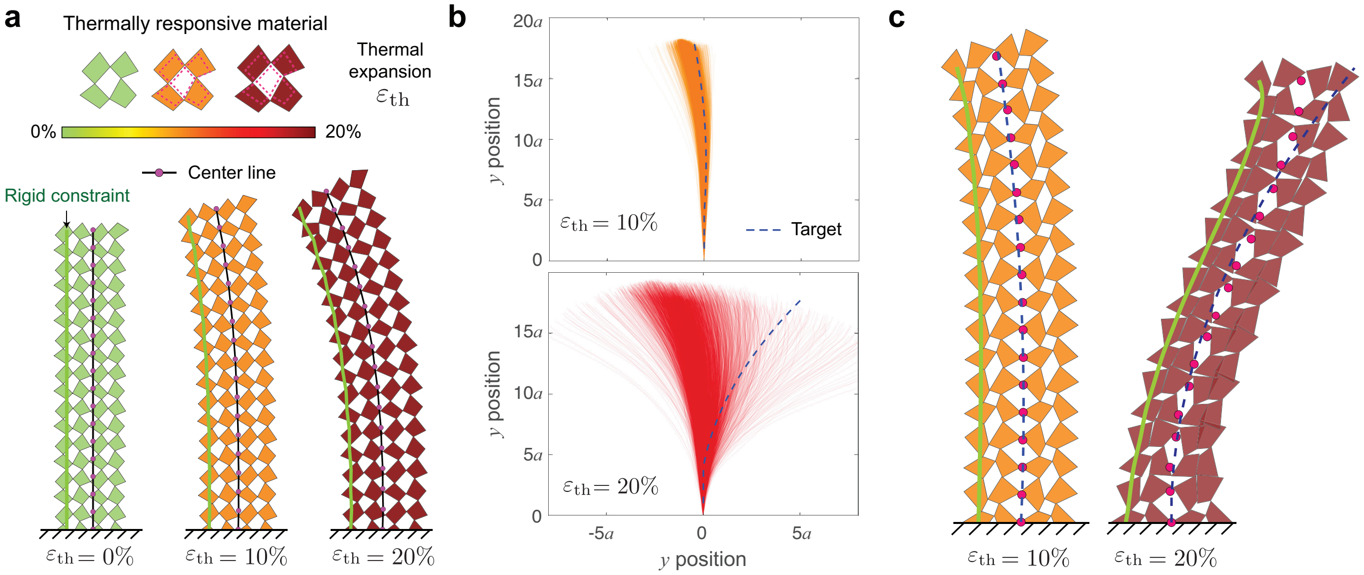

FIG. S16. Inverse design of structures capable of target deformations upon heating. (a) We consider a structure comprising 8×2 unit cells and assume that each quadrilateral unit expands uniformly under an increasing temperature. To trigger bending, we fix the center-to-center distance of the units on the left boundary. (b) The distribution of $< u_i >^{\varepsilon_{\rm th}}>$ $(i = 1, 2, ..., 18)$ for 7500 different unit cell designs at $\varepsilon_{\rm th}$= 0.1 (top) and 0.2 (bottom). The blue dashed lines represent the target curves at the two different thermal strains. (c) Our numerical strategy identifies architectures capable of supporting the target deformation modes described by Eq. (S45).

As for the case of the stress-strain curve, we then train our neural network to to learn the relation between the unit cell design and the deformation of the structure (described by $< u_i >^{\varepsilon_{\rm th}}$ - since the displacement in y-direction is much smaller and, therefore, can be neglected) as a function of $\varepsilon_{\rm th}$. To this end, we represent the deformation for the p-th unit cell in our database as N_y-dimensional vector

$$\left[< u_1 >^{\varepsilon_{\rm th}}, < u_2 >^{\varepsilon_{\rm th}}, \cdots, ..., < u_{N_y} >^{\varepsilon_{\rm th}}\right]^\top \tag{S44}$$

c. Inverse design Here, we train the NN and couple with an evolution strategy to identify an architecture whith target deforamtion described by

$$\begin{aligned} < u_i >_{target}^{\varepsilon_{\rm th}=0.1} &= -0.0062 i^2 + 0.0494 i - 0.0432, \\ < u_i >_{target}^{\varepsilon_{\rm th}=0.2} &= 0.0247 i^2 - 0.0474 i + 0.0247, \end{aligned} \tag{S45}$$

where $i = 1, .., 18$.

As shown in Fig. S16c, we find that also in this case our strategy can identify architectures capable of supporting the target deformation modes described by Eq. (S45).

S9. ADDITIONAL RESULTS

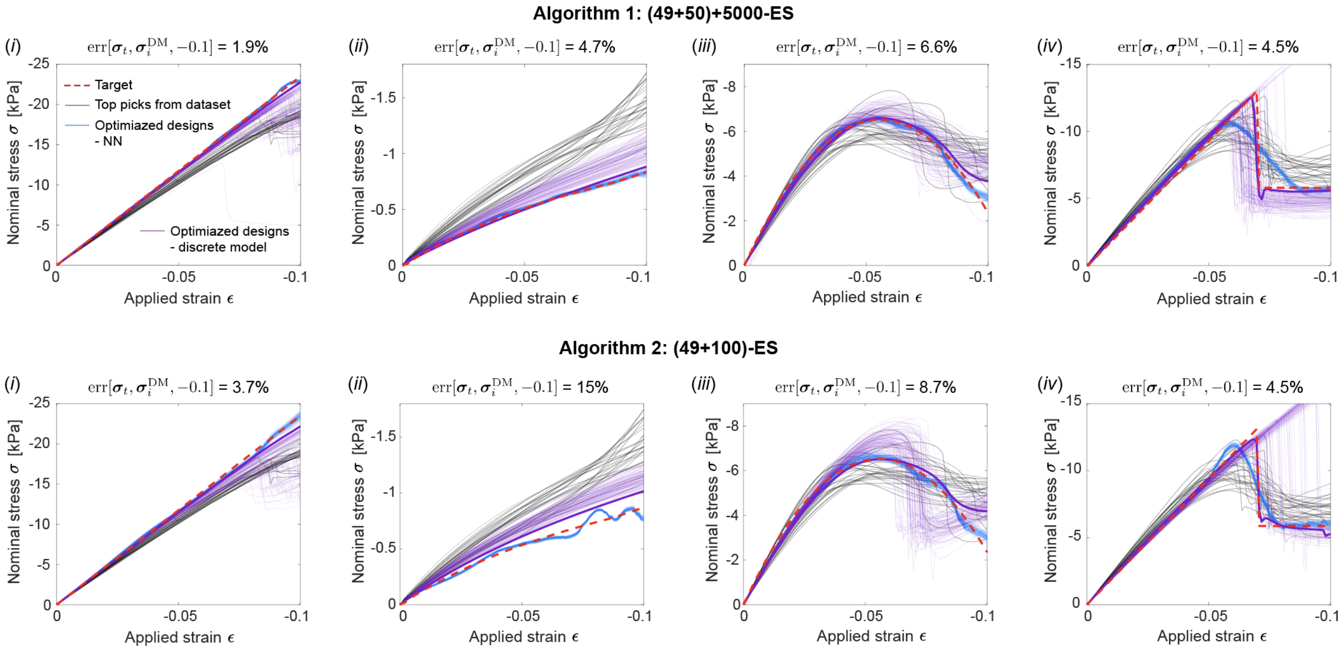

FIG. S17. Effect of the 5000 randomly generated samples on the performance of evolution strategy (ES). For the ES used in our study we select $\lambda = 49$ children, $\mu = 100$ parents and $\psi = 5000$ randomly generated samples within each iteration (Algorithm 1: (49+100)+5000-ES). To test the effect of the additional $\psi = 5000$ random designs on the performance of the algorithm, we compare its results with those obtained considering $\lambda = 99$, $\mu = 100$ and $\psi = 0$ (Algorithm 2: (49+100)-ES). Note that both algorithms require evaluation of 10000 design at each iteration. We find that Algorithm 1 yields better results than Algorithm 2 for all four considered targets cases with the minimum error equal to (*i*) 1.9%, (*ii*) 4.7%, (*iii*) 6.6%, and (*iv*) 4.5% for Algorithm 1 and (*i*) 3.7%, (*ii*) 5.7%, (*iii*) 8.7%, and (*iv*) 4.5% for Algorithm 2. Further, when we run both algorithms 50 times on case (iii), we found that Algorithm 1 gives an average error of 5.4%, whereas Algorithm 2 leads to an average error of 5.9%. In each plot we show the target response (red dashed line), top picks from dataset (black lines) and stress-strain curves predicted by our trained NN (blue lines) and our discrete model (purple lines) for the optimized designs. The best optimal designs identified by the discrete model are highlighted by thick purple lines.

REFERENCES

[1] C. Coulais, C. Kettenis, and M. van Hecke, A characteristic length scale causes anomalous size effects and boundary programmability in mechanical metamaterials, Nature Physics **14**, 40 (2018).

[2] B. Deng, S. Yu, A. E. Forte, V. Tournat, and K. Bertoldi, Characterization, stability, and application of domain walls in flexible mechanical metamaterials, Proceedings of the National Academy of Sciences **117**, 31002 (2020).

[3] I. T. Jolliffe and J. Cadima, Principal component analysis: a review and recent developments, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **374**, 20150202 (2016).

[4] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019) pp. 8024–8035.

[5] S. Kumar, S. Tan, L. Zheng, and D. M. Kochmann, Inverse-designed spinodoid metamaterials, npj Computational Materials **6**, 1 (2020).

[6] A. G. Mark, S. Palagi, T. Qiu, and P. Fischer, Auxetic metamaterial simplifies soft robot design, in *2016 IEEE international conference on robotics and automation (ICRA)* (Ieee, 2016) pp. 4951–4956.