Supplementary Materials for Programmable 3D metamaterials with tailored Poisson's ratios over large deformation

This file includes:

- Supplementary detailed description of computational algorithm
- Supplementary fabrication and measurement
- Captions for Movie 1 to 4
- Captions for Raw Data Folder

Other Supplementary Materials for this manuscript include the following:

- Movies 1 to 4
- Raw Data Folder

1 Our computational algorithm

In Section 1.2, we introduce a novel formulation to calculate the Poisson's ratio of metamaterial under large deformation through the definition of an optimization problem, then discuss the details of Newton's solver for computing the Poisson's ratio in Sec. 1.3, and finally present the inverse design framework for optimizing the metamaterial to possess the tailored Poisson's ratios in Sec. 1.4. Next, we discrete the inverse design framework to be computable in Sec. 1.5.

1.1 Geometric modeling

Our method for modeling microstructure is built upon the foundations laid by[1] and[2]. The algorithm takes as input a microstructure skeleton, defined by vertex positions and the accompanying connecting edges. The skeleton is subsequently thickened to a solid configuration by leveraging a series of geometric parameters, amalgamated as vector $\mathbf{p}$, including the vertices' positions, a designated radius and a smoothness parameter at each vertex. These parameters characterize a sphere for each vertex, and the edge geometry primitives $P_i\,(i=1,...,N_p)$ are defined as the convex hull of the endpoint spheres for each edge. The primitives form joints $Jm\,(m=1,...,N_J)$, each defined by a subset of the primitives, with smoothness determined by the vertex's smoothing parameter s_m. The level-set function $F(\mathbf{q})$ is computed at a point $\mathbf{q}\in\mathbb{R}^3$ based on a set of convex primitives with signed distance functions $F_i(\mathbf{q})$.

Blending algorithm For each joint, we define two signed distance functions: F_m^s representing the blended joint, and F_m^h representing the (non-smooth) exact union of the primitives:

$$\begin{aligned} F_m^s(\mathbf{q}) &= KS\left(F_i(\mathbf{q}), i\in J_m; s_m\eta_J\left(\frac{dist_{hull}(J_m)(\mathbf{q})}{R_m}\right)\right),\\ F_m^h(\mathbf{q}) &= \min_{i\in J_m} F_i(\mathbf{q}), \end{aligned} \tag{1}$$

where KS refers to the $Kreisselmeier-Steinhauser$ function, R_m is the minimal distance from the medial axis of any primitive in J_m to the convex hull boundary. and $\eta_J(t)=1-tanh(1.025max(1+t,0))^{10}$. The signed distance function to the combined joint pair is then:

$$\begin{aligned} F(\mathbf{q}) &= KS(F_a^s(\mathbf{q}),F_b^s(\mathbf{q});\tilde{s}_{ab}(\mathbf{q})),\\ \tilde{s}_{ab}(\mathbf{q})) &= s_{ab}\eta_F((F_a^h(\mathbf{q})-F_a^s(\mathbf{q}))(F_b^h(\mathbf{q})-F_b^s(\mathbf{q}))), \end{aligned} \tag{2}$$

where $\eta_F(t)=\tanh(1000t)$ and s_{ab} is the smoothing amounts defined on overlapping joint pair.

Therefore, we represent the solid region ω by $\mathbf{p}$. The objective is to derive a sequence $\{\mathbf{p}_i\}$ such that the Poisson's ratio of the thickened microstructure aligns with our predefined target values $\left\{\boldsymbol{\nu}_{\text{in}}(\varepsilon_z^i)\right\}$ and achieve complex stretching response of materials. Utilizing this approach ensures the fulfillment of the programmability requirement.

1.2 Poisson's ratio of hyperelastic material

Given a constitutive model of a material, suppose the strain energy or Helmholtz free energy is represented by $\Phi(\mathbf{F})$, we define its Poisson's ratio $\boldsymbol{\nu}_{\varepsilon_z}$ under the stretch along z direction as

$$\boldsymbol{\nu}_{\varepsilon_z} = \frac{1}{\varepsilon_z}\arg\min_{\boldsymbol{\varepsilon}}\Phi(\mathbf{F}_{\varepsilon_z}(\boldsymbol{\varepsilon}))d\omega, \tag{3}$$

where $\boldsymbol{\nu}_{\varepsilon_z}=(\nu_x,\nu_y)$ represents the lateral shrink ratio in the x-direction and the y-direction under the stretch, while $\boldsymbol{\varepsilon}=(\varepsilon_x,\varepsilon_y)$ indicates the corresponding strain components. ε_z denotes the stretch ratio and $\mathbf{F}_{\varepsilon_z}(\boldsymbol{\varepsilon})$ is defined as

$$\mathbf{F}_{\varepsilon_z}(\boldsymbol{\varepsilon}) = \begin{pmatrix} 1-\varepsilon_x & & \\ & 1-\varepsilon_y & \\ & & 1+\varepsilon_z \end{pmatrix}. \tag{4}$$

Intuitively, the definition in (3) identifies the lateral shrinkage when the material is subjected to a specified stretch. Based on the principle of minimum potential, this lateral shrinkage is the most probable in real-world scenarios as it minimizes the strain energy. Most of the time, we only consider the structures with cubic symmetry. Therefore, the problem (3) is simplified as

$$\nu_{\varepsilon_z} = \frac{1}{\varepsilon_z}\arg\min_{\varepsilon}\Phi(\mathbf{F}_{\varepsilon_z}(\varepsilon))d\omega, \tag{5}$$

where $\mathbf{F}_{\varepsilon_z}(\varepsilon)$ is a simplified notation of $\mathbf{F}_{\varepsilon_z}(\varepsilon,\varepsilon)$.

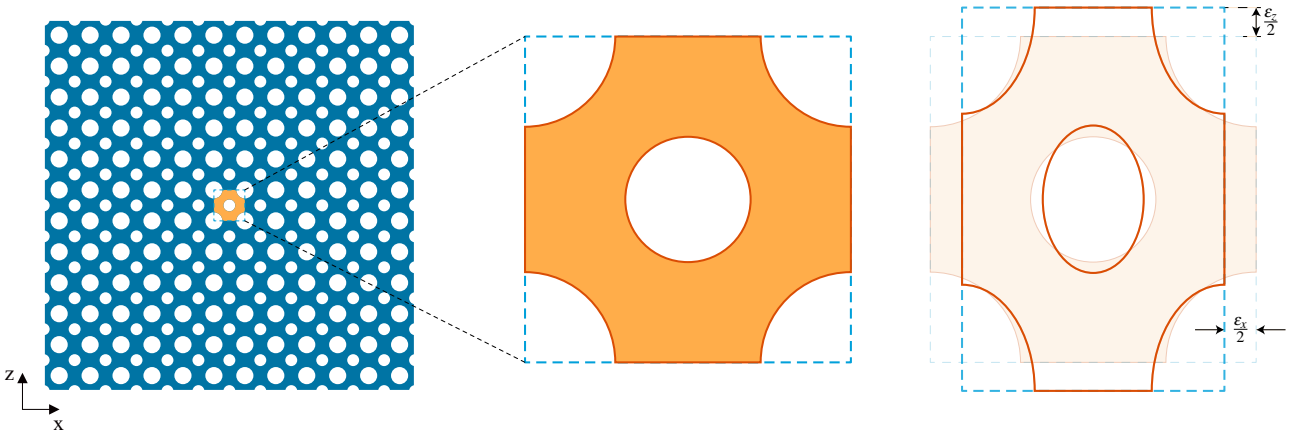

Figure 1. 2D illustration of unit cell and shape of microstructure. ε_x represents the strain in the x-direction of a unit cell when stretched along the z-direction with a strain of ε_z.

Homogenization Our definition is universally applicable to any hyperelastic constitutive model. We aim to leverage this definition to tackle challenges in microstructure design, with the specific objective of generating microstructures with predefined functionalities that feature programmable Poisson's ratios, even under large deformation conditions. As the constitutive model for the hyperelastic microstructure remains unknown, we turn to hyperelastic homogenization, as introduced by[3]. According to[3], the homogenized elastic energy of microstructure material, undergoing a macroscopic deformation gradient $\mathbf{F}$, involves solving the following nonlinear PDE:

$$\begin{cases} \nabla \cdot \dfrac{\partial \Phi}{\partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) = 0 \quad x \in \omega \\ \dfrac{\partial \Phi}{\partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) \cdot \mathbf{n} = 0 \quad x \in \partial\omega \backslash \partial Y \\ \qquad\qquad\qquad\qquad \mathbf{u}\#. \end{cases} \tag{6}$$

where Φ is the elastic energy of base material; the unit cell is denoted as $Y := [0,1]^3$; $\mathbf{u}\#$ means $\mathbf{u}$ is Y-periodic, i.e., $\mathbf{u}$ evaluates the same value on the opposite boundaries of Y; ω represents the region of solid in the unit cell; $\mathbf{n}$ is the normal vector field of the boundary surface of ω. We denote the solution of (6) as $\mathbf{u}^*$; then, the homogenized elastic energy is determined as

$$\Phi^H_\omega(\mathbf{F}) = \frac{1}{Y}\int_\omega \Phi(\nabla \mathbf{u}^* + \mathbf{F}) d\omega, \tag{7}$$

where $\Phi^H_\omega(\mathbf{F})$ denotes the homogenized elastic energy density of the macrostructure material with shape ω in the unit cell. We substitute (7) into (5) to evaluate the Poisson's ratio, but it entails solving (6) and the derivative of $\mathbf{u}^*$ with respect to variation of ω in each step during the optimization of (5).

Optimization problem for Poisson's ratio To facilitate the optimization, we reformulate (6) into a equivalent minimization problem of a functional:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\#} \int_\omega \Phi(\nabla \mathbf{u} + \mathbf{F}) d\omega. \tag{8}$$

Then, combined with (7), the homogenized elastic energy density is obtained by solving the following minimization problem:

$$\Phi^H_\omega(\mathbf{F}) = \min_{\mathbf{u}\#} \frac{1}{|Y|}\int_\omega \Phi(\nabla \mathbf{u} + \mathbf{F}) d\omega. \tag{9}$$

We jointly solve the minimization problems (5) and (9) as one problem:

$$\min_{\boldsymbol{\varepsilon}, \mathbf{u}\#} \frac{1}{|Y|}\int_\omega \Phi(\nabla \mathbf{u} + \mathbf{F}_{\varepsilon_z}(\boldsymbol{\varepsilon})) d\omega. \tag{10}$$

Consequently, the Poisson's ratio is

$$\boldsymbol{\nu}_{\varepsilon_z} = -\frac{1}{\varepsilon_z}\boldsymbol{\varepsilon}(\omega, \varepsilon_z), \tag{11}$$

where $\boldsymbol{\varepsilon}(\omega, \varepsilon_z)$ is the solution of (10).

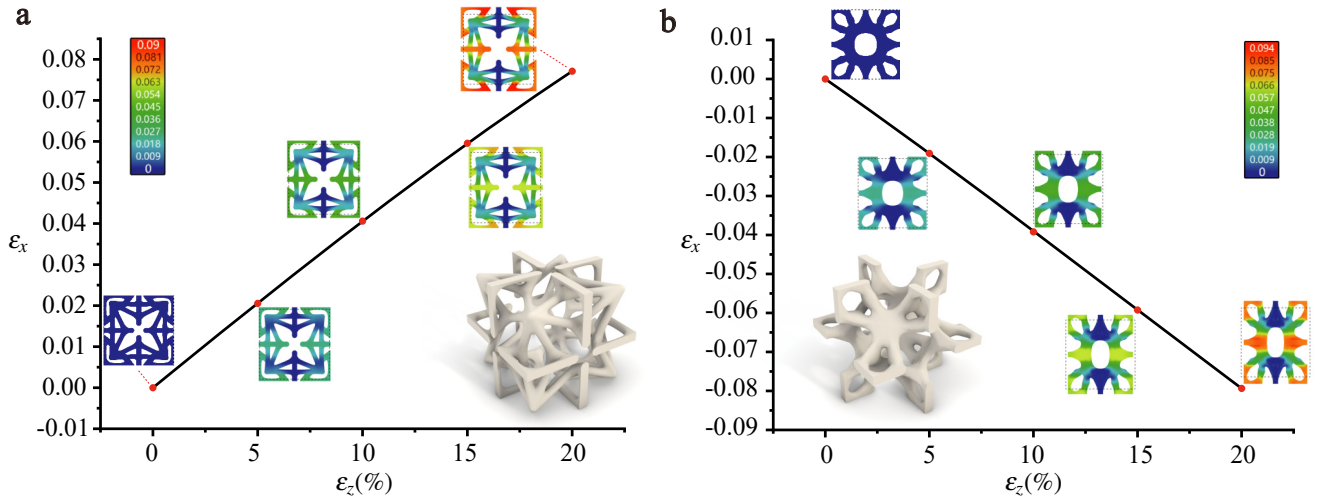

Figure 2. Deformation of the unit cell under stretch along z-direction when minimum of (10) is reached. a Unit cell with a Poisson's ratio of -0.4. **b** Unit cell with a Poisson's ratio of 0.4.

1.3 Newton's method for Poisson's ratio

We use the Newton's method to solve (10). We denote

$$\begin{aligned} \bar{\Phi}(\mathbf{u},\boldsymbol{\varepsilon}) &= \frac{1}{|Y|}\int_{\omega}\Phi(\nabla\mathbf{u}+\mathbf{F}_{\varepsilon_z}(\boldsymbol{\varepsilon}))d\omega, \\ \mathbf{R} = \frac{\partial\bar{\Phi}}{\partial\mathbf{u}},\ \mathbf{r} &= \frac{\partial\bar{\Phi}}{\partial\boldsymbol{\varepsilon}},\ \mathbf{K} = \frac{\partial^2\bar{\Phi}}{\partial\mathbf{u}\partial\mathbf{u}}. \end{aligned} \tag{12}$$

The algorithm is listed as follows:

1. Set $\mathbf{u}^0=\mathbf{0}, \boldsymbol{\varepsilon}^0=\mathbf{0}$, and $k=0$.
2. Compute residual $\mathbf{z}^k = \begin{pmatrix}\mathbf{R}^k\\ \mathbf{r}^k\end{pmatrix}$, $z_k = \|\mathbf{z}^k\|$. Terminate the solver and output $\mathbf{u}$ and ε if $z_k/z_0 < 10^{-5}$.
3. Solve for the descent direction $\begin{pmatrix}\mathbf{d}_{\mathbf{u}}^k\\ \mathbf{d}_{\boldsymbol{\varepsilon}}^k\end{pmatrix}$:

$$\begin{pmatrix}\mathbf{K}^k & \frac{\partial\mathbf{R}}{\partial\boldsymbol{\varepsilon}}^k\\ \frac{\partial\mathbf{r}}{\partial\mathbf{u}}^k & \frac{\partial\mathbf{r}}{\partial\boldsymbol{\varepsilon}}^k\end{pmatrix}\begin{pmatrix}\mathbf{d}_{\mathbf{u}}^k\\ \mathbf{d}_{\boldsymbol{\varepsilon}}^k\end{pmatrix} = -\begin{pmatrix}\mathbf{R}^k\\ \mathbf{r}^k\end{pmatrix}. \tag{13}$$

4. Perform a standard Armijo backtracking algorithm to determine the step size α.
5. Update the approximation of solution

$$\begin{pmatrix}\mathbf{u}^{k+1}\\ \boldsymbol{\varepsilon}^{k+1}\end{pmatrix} = \begin{pmatrix}\mathbf{u}^k\\ \boldsymbol{\varepsilon}^k\end{pmatrix} + \alpha\begin{pmatrix}\mathbf{d}_{\mathbf{u}}^k\\ \mathbf{d}_{\boldsymbol{\varepsilon}}^k\end{pmatrix},$$

and go back to step 2 with $k := k+1$.

The subscript k means that the function evaluates at $(\mathbf{u}^k, \boldsymbol{\varepsilon}^k)$.

1.4 Inverse design of hyperelastic microstructure

Our goal of generating microstructure with a given Poisson's ratio is formulated as problem:

$$\min_{\omega} J(\omega) = \frac{1}{2}\int_0^{\varepsilon_z^{\max}} |\boldsymbol{\nu}_{\varepsilon_z} - \boldsymbol{\nu}_{\text{in}}(\varepsilon_z)|^2 d\varepsilon_z, \tag{14}$$

where $\boldsymbol{\nu}_{\varepsilon_z} = \frac{1}{\varepsilon_z}\boldsymbol{\varepsilon}(\omega,\varepsilon_z)$ and ε_z is the stretch ratio, $\boldsymbol{\nu}_{\text{in}}(\varepsilon_z)$ is the tailored Poisson's ratio that is a function as ε_z, and $[0,\varepsilon_z^{\max}]$ is the considered range of the stretch ratio.

There are requirements on the volume ratio of the microstructure, which is easily integrated into our algorithm. The volume constrain is described as

$$V(\omega) = \frac{1}{|Y|}\int_{\omega} 1 d\omega \in [v_{\min}, v_{\max}], \tag{15}$$

where the lower and upper bounds on the volume ratio are given by $V_{\min}$ and $V_{\max}$.

Sensitivity analysis The sensitivities of the objective function (14) and the volume constraint (15) are determined as

$$\frac{\partial J}{\partial \omega} = \int_0^{\varepsilon_z^{max}} \left(\boldsymbol{\nu}_{\varepsilon_z} - \boldsymbol{\nu}_{\text{in}}(\varepsilon_z)\right) \cdot \frac{\partial \boldsymbol{\nu}_{\varepsilon_z}}{\partial \omega}, \tag{16}$$

$$\frac{\partial V}{\partial \omega} = \frac{1}{|Y|}\int_{\omega} \mathbf{v}\cdot\mathbf{n} dA, \tag{17}$$

where the derivative with respect to ω means the rate of change when the boundary of ω is moving along a specific velocity field $\mathbf{v}$, that is

$$\frac{\partial f(\omega)}{\partial \omega} := \lim_{t\to 0} \frac{f(\omega + t\mathbf{v}) - f(\omega)}{t}$$

for any function $f(\omega)$.

For the evaluation of $\frac{\partial \boldsymbol{\nu}_{\varepsilon_z}}{\partial \omega}$, we notice that

$$\begin{pmatrix} \frac{\partial \bar{\Phi}}{\partial \mathbf{u}}(\omega, \boldsymbol{\varepsilon}, \mathbf{u}) \\ \frac{\partial \bar{\Phi}}{\partial \boldsymbol{\varepsilon}}(\omega, \boldsymbol{\varepsilon}, \mathbf{u}) \end{pmatrix} = \mathbf{0}, \tag{18}$$

holds for any ω when $\boldsymbol{\varepsilon}$ and $\mathbf{u}$ are the solution of (10), where $\bar{\Phi}(\omega, \boldsymbol{\varepsilon}, \mathbf{u}) := \frac{1}{|Y|}\int_{\omega} \Phi(\nabla \mathbf{u} + \mathbf{F}_{\varepsilon_z}(\boldsymbol{\varepsilon})) d\omega$. Then, we take derivatives on (18) and get the following equations:

$$\begin{pmatrix} \frac{\partial^2 \bar{\Phi}}{\partial \mathbf{u}^2} & \frac{\partial^2 \bar{\Phi}}{\partial \boldsymbol{\varepsilon} \partial \mathbf{u}} \\ \frac{\partial^2 \bar{\Phi}}{\partial \mathbf{u} \partial \boldsymbol{\varepsilon}} & \frac{\partial^2 \bar{\Phi}}{\partial \boldsymbol{\varepsilon}^2} \end{pmatrix} \begin{pmatrix} \frac{\partial \mathbf{u}}{\partial \omega} \\ \frac{\partial \boldsymbol{\varepsilon}}{\partial \omega} \end{pmatrix} = - \begin{pmatrix} \frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \mathbf{u}} \\ \frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \boldsymbol{\varepsilon}} \end{pmatrix}. \tag{19}$$

Hence we have

$$\left[\frac{\partial^2 \Phi}{\partial \boldsymbol{\varepsilon}^2} - \frac{\partial^2 \Phi}{\partial \boldsymbol{\varepsilon} \partial \mathbf{u}} \left(\frac{\partial^2 \Phi}{\partial \mathbf{u}^2}\right)^{-1} \frac{\partial^2 \Phi}{\partial \mathbf{u} \partial \boldsymbol{\varepsilon}}\right] \frac{\partial \boldsymbol{\varepsilon}}{\partial \omega} = -\frac{\partial^2 \Phi}{\partial \omega \partial \boldsymbol{\varepsilon}} + \frac{\partial^2 \Phi}{\partial \boldsymbol{\varepsilon} \partial \mathbf{u}} \left(\frac{\partial^2 \Phi}{\partial \mathbf{u}^2}\right)^{-1} \frac{\partial^2 \Phi}{\partial \omega \partial \mathbf{u}}. \tag{20}$$

After solving this equation for $\frac{\partial \boldsymbol{\varepsilon}}{\partial \omega}$, we get the sensitivity $\frac{\partial \boldsymbol{\nu}_{\varepsilon_z}}{\partial \omega} = \frac{1}{\varepsilon_z}\frac{\partial \boldsymbol{\varepsilon}}{\partial \omega}$. The second-order derivatives are computed as

$$\begin{aligned}
&\frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \boldsymbol{\varepsilon}} = [\frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \varepsilon_x} \quad \frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \varepsilon_y}]^T, \quad \frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \varepsilon_x}[1, \mathbf{v}] = \int_{\partial\omega} \frac{\partial \Phi}{\partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) : -\mathbf{F}_x (\mathbf{v}\cdot\mathbf{n}) dA, \\
&\frac{\partial^2 \bar{\Phi}}{\partial \boldsymbol{\varepsilon}^2} = \begin{pmatrix} \frac{\partial^2 \bar{\Phi}}{\partial {\varepsilon_x}^2} & \frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_x \partial \varepsilon_y} \\ \frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_y \partial \varepsilon_x} & \frac{\partial^2 \bar{\Phi}}{\partial {\varepsilon_y}^2} \end{pmatrix}, \\
&\frac{\partial^2 \bar{\Phi}}{\partial {\varepsilon_x}^2}[1,1] = \int_{\omega} \mathbf{F}_x : \frac{\partial^2 \Phi}{\partial \mathbf{F} \partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) : \mathbf{F}_x d\omega, \quad \frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_x \partial \varepsilon_y}[1,1] = \int_{\omega} \mathbf{F}_x : \frac{\partial^2 \Phi}{\partial \mathbf{F} \partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) : \mathbf{F}_y d\omega, \\
&\frac{\partial^2 \bar{\Phi}}{\partial \boldsymbol{\varepsilon} \partial \mathbf{u}} = [\frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_x \partial \mathbf{u}} \quad \frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_y \partial \mathbf{u}}]^T, \quad \frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_x \partial \mathbf{u}}[\delta \mathbf{u}, 1] = \int_{\omega} \nabla \delta \mathbf{u} : \frac{\partial^2 \Phi}{\partial \mathbf{F} \partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) : -\mathbf{F}_x d\omega, \\
&\frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \mathbf{u}}[\delta \mathbf{u}, \mathbf{v}] = \int_{\partial\omega} \frac{\partial \Phi}{\partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) : \nabla \delta \mathbf{u} (\mathbf{v}\cdot\mathbf{n}) dA, \\
&\frac{\partial^2 \bar{\Phi}}{\partial \mathbf{u}^2}[\mathrm{d}\mathbf{u}, \delta \mathbf{u}] = \int_{\omega} \nabla \delta \mathbf{u} : \frac{\partial^2 \Phi}{\partial \mathbf{F} \partial \mathbf{F}}(\nabla \mathbf{u} + \mathbf{F}) : \nabla \mathrm{d}\mathbf{u} d\omega,
\end{aligned} \tag{21}$$

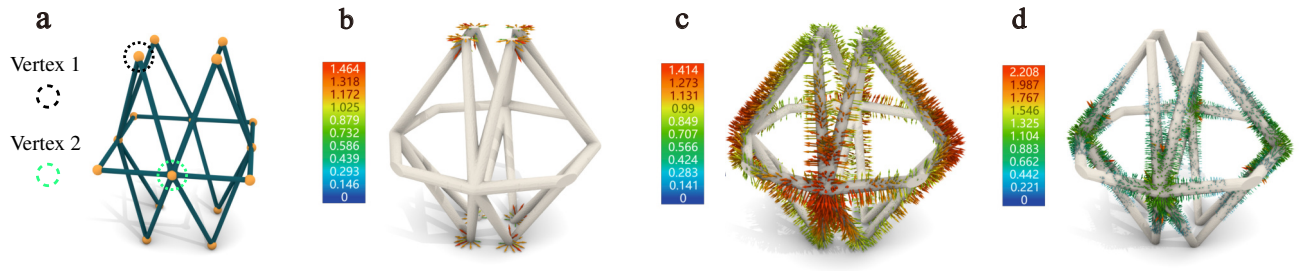

Figure 3. Visualization of shape velocity field when perturbing the geometry parameter. a Vertices to be perturbed in the skeleton. Perturbing the geometric parameters of a specific vertex means perturbing the geometric parameters of a set of vertices generated by the elemental skeleton since skeleton is generated by symmetry from primitive unit skeleton, as shown in Figure 1**b** of the main text. **b** Perturbing the radius of vertex 1. **c** Perturbing the position of vertex 2. **d** Perturbing the smoothness of vertex 2.

where $\mathbf{u}$ is the solution of (10); $\mathbf{F}_x$ is the derivative $\frac{\partial \mathbf{F}_{\varepsilon_x}}{\partial \varepsilon_x}$; the notation in the bracket means the direction to evaluate the derivative. For example,

$$\begin{aligned} \frac{\partial \bar{\Phi}}{\partial \mathbf{u}}[\mathrm{d}\mathbf{u}] &= \lim_{t\to 0} \frac{\bar{\Phi}(\mathbf{u}+t\mathrm{d}\mathbf{u}) - \bar{\Phi}(\mathbf{u})}{t}, \\ \frac{\partial^2 \bar{\Phi}}{\partial \mathbf{u}^2}[\mathrm{d}\mathbf{u}, \delta\mathbf{u}] &= \lim_{t\to 0} \frac{\frac{\partial \bar{\Phi}}{\partial \mathbf{F}}[\mathrm{d}\mathbf{u}](\mathbf{u}+t\delta\mathbf{u}) - \frac{\partial \bar{\Phi}}{\partial \mathbf{F}}[\mathrm{d}\mathbf{u}](\mathbf{u})}{t}. \end{aligned} \tag{22}$$

The calculation for $\mathbf{F}_y$, $\frac{\partial^2 \bar{\Phi}}{\partial \omega \partial \varepsilon_y}$, $\frac{\partial^2 \bar{\Phi}}{\partial {\varepsilon_y}^2}$ and $\frac{\partial^2 \bar{\Phi}}{\partial \varepsilon_y \partial \mathbf{u}}$ can be carried out by analogy.

1.5 Discretization

FEM setting We discretize (10) by quadratic tetrahedral FEM. We use straight-edged elements (subparametric FEM) for representing the solid region ω to simplify meshing and the sensitivity formulas. We only generate mesh for only $1/8$ cell due to cubic symmetry, and mirror it to get the entire domain. This guarantees the periodicity of the mesh. Note that there are duplicated FEM nodes on the opposite boundaries due to periodicity, we consider each group of them as a single node and only assign one Degree of freedom (Dof) and shape function to it.

In the FEM setting, the displacement and its gradient are represented by

$$\mathbf{u} = \sum_a \mathbf{u}^a N^a, \tag{23}$$

$$\nabla \mathbf{u} = \sum_a \mathbf{u}^a \otimes \nabla N^a, \tag{24}$$

where a enumerates all the FEM nodes, and N_a is the shape function assigned to it. $\mathbf{u}^a$ is the nodal displacement. Then the

linear functionals in (13) are discretized into matrix format:

$$
\begin{aligned}
&\mathbf{R} = (\mathbf{R}^a)_{a\in\mathscr{V}}, \quad \mathbf{R}^a = \sum_{e\in\mathscr{N}_a}\int_{\omega_e}\frac{\partial\Phi}{\partial\mathbf{F}}\nabla N^a d\omega,\\
&\mathbf{r} = [r_x \quad r_y]^T, \quad r_x = -\int_{\omega}\frac{\partial\Phi}{\partial\mathbf{F}}:\mathbf{F}_x d\omega,\\
&\mathbf{K} = (\mathbf{K}^{ab})_{a,b\in\mathscr{V}}, \quad \mathbf{K}^{ab}_{ij} = \int_{\omega}\sum_{m,n}\frac{\partial^2\Phi}{\partial F_{jm}\partial F_{in}}N^a_{,n}N^b_{,m}d\omega,\\
&\frac{\partial\mathbf{R}}{\partial\boldsymbol{\varepsilon}} = \left(\frac{\partial\mathbf{R}}{\partial\varepsilon_x}, \quad \frac{\partial\mathbf{R}}{\partial\varepsilon_y}\right)^T, \quad \frac{\partial\mathbf{R}}{\partial\varepsilon_x} = \left(\frac{\partial\mathbf{R}^a}{\partial\varepsilon_x}\right)_{a\in\mathscr{V}}, \quad \frac{\partial\mathbf{R}^a}{\partial\varepsilon_x} = \sum_{e\in\mathscr{N}_a}\int_{\omega_e} -\mathbf{F}_x:\frac{\partial^2\Phi}{\partial\mathbf{F}\partial\mathbf{F}}\cdot\nabla N^a d\omega,\\
&\frac{\partial\mathbf{r}}{\partial\mathbf{u}} = (\frac{\partial\mathbf{R}}{\partial\boldsymbol{\varepsilon}})^T,\\
&\frac{\partial\mathbf{r}}{\partial\boldsymbol{\varepsilon}} = \begin{pmatrix} \frac{\partial r_x}{\partial\varepsilon_x} & \frac{\partial r_y}{\partial\varepsilon_x} \\ \frac{\partial r_x}{\partial\varepsilon_y} & \frac{\partial r_y}{\partial\varepsilon_y}\end{pmatrix}, \quad \frac{\partial r_x}{\partial\varepsilon_x} = \int_{\omega}\mathbf{F}_x:\frac{\partial^2\Phi}{\partial\mathbf{F}\partial\mathbf{F}}:\mathbf{F}_x d\omega, \quad \frac{\partial r_x}{\partial\varepsilon_y} = \frac{\partial r_y}{\partial\varepsilon_x} = \int_{\omega}\mathbf{F}_x:\frac{\partial^2\Phi}{\partial\mathbf{F}\partial\mathbf{F}}:\mathbf{F}_y d\omega,
\end{aligned}
\tag{25}
$$

where we have omit the superscript k of iteration and all the derivatives of Φ evaluate at $\nabla\mathbf{u}+\mathbf{F}$. $\mathscr{V}$ is the set of FEM nodes ; $\mathscr{N}_a$ is the set of incident elements of a. Similarly, the derivatives in (21) evaluate to

$$
\begin{aligned}
&\frac{\partial^2\bar{\Phi}}{\partial p\partial\boldsymbol{\varepsilon}} = (\frac{\partial^2\bar{\Phi}}{\partial p\partial\varepsilon_x}, \frac{\partial^2\bar{\Phi}}{\partial p\partial\varepsilon_y})^T, \quad \frac{\partial^2\bar{\Phi}}{\partial p\partial\varepsilon_x} = \int_{\partial\omega}\frac{\partial\Phi}{\partial\mathbf{F}}:-\mathbf{F}_x\sum_b(\mathbf{v}^b\cdot\mathbf{n}^b)N^b dA,\\
&\frac{\partial^2\bar{\Phi}}{\partial\boldsymbol{\varepsilon}^2} = \begin{pmatrix} \frac{\partial^2\bar{\Phi}}{\partial{\varepsilon_x}^2} & \frac{\partial^2\bar{\Phi}}{\partial\varepsilon_x\partial\varepsilon_y} \\ \frac{\partial^2\bar{\Phi}}{\partial\varepsilon_y\partial\varepsilon_x} & \frac{\partial^2\bar{\Phi}}{\partial{\varepsilon_y}^2}\end{pmatrix}, \quad \frac{\partial^2\bar{\Phi}}{\partial\varepsilon_x^2} = \sum_e\int_{\omega_e}\mathbf{F}_x:\frac{\partial^2\Phi}{\partial\mathbf{F}\partial\mathbf{F}}:\mathbf{F}_x d\omega, \quad \frac{\partial^2\bar{\Phi}}{\partial\varepsilon_x\partial\varepsilon_y} = \sum_e\int_{\omega_e}\mathbf{F}_x:\frac{\partial^2\Phi}{\partial\mathbf{F}\partial\mathbf{F}}:\mathbf{F}_y d\omega,\\
&\frac{\partial^2\bar{\Phi}}{\partial\boldsymbol{\varepsilon}\partial\mathbf{u}} = [\frac{\partial^2\bar{\Phi}}{\partial\varepsilon_x\partial\mathbf{u}} \quad \frac{\partial^2\bar{\Phi}}{\partial\varepsilon_y\partial\mathbf{u}}]^T, \quad [\frac{\partial^2\bar{\Phi}}{\partial\varepsilon_x\partial\mathbf{u}}]_a = \sum_{e\in\mathscr{N}_a}\int_{\omega_e}(\mathbf{I}\otimes\nabla N^a):\frac{\partial^2\Phi}{\partial\mathbf{F}\partial\mathbf{F}}:-\mathbf{F}_x d\omega,\\
&[\frac{\partial^2\bar{\Phi}}{\partial p\partial\mathbf{u}}]_a = \int_{\partial\omega}\frac{\partial\Phi}{\partial F}:(\mathbf{I}\otimes\nabla N^a)\sum_b(\mathbf{v}^b\cdot\mathbf{n}^b)N^b dA,\\
&\frac{\partial^2\bar{\Phi}}{\partial\mathbf{u}^2} = \mathbf{K},
\end{aligned}
\tag{26}
$$

where p is an entry of geometry parameter $\mathbf{p}$, and $\mathbf{v}^b$means the velocity of node b when p increase in a unit speed. The normal vector on node b is denoted as $\mathbf{v}^b$. Here, we interpolate the nodal normal velocity with shape function. All the integrals in (25) and (21) can split into sum of integrals on surface triangles or tetrahedrons, which is computed through numerical quadrature.

For the intergral in (14) and its sensitivity (16), we sample a sequence of stretch ratio and discretize them as

$$
J(\omega) = \frac{1}{2}\sum_{i=0}^{N}|\boldsymbol{\nu}_{\varepsilon_z^i} - \boldsymbol{\nu}_{\text{in}}(\varepsilon_z^i)|^2, \tag{27}
$$

$$
\frac{\partial J}{\partial\omega} = \sum_{i=0}^{N}\left(\boldsymbol{\nu}_{\varepsilon_z^i} - \boldsymbol{\nu}_{\text{in}}(\varepsilon_z^i)\right)\cdot\frac{\partial\boldsymbol{\nu}_{\varepsilon_z^i}}{\partial\omega}, \tag{28}
$$

respectively, where $\varepsilon_z^i, i = 1,\cdots,N$ is the sequence of sampled stretch ratios from range $[0,\varepsilon_z^{max}]$.

After the above discretization, we obtain the following optimization problem:

$$
\begin{aligned}
&\min_{\mathbf{p}} J(\mathbf{p}) = \frac{1}{2}\sum_{i=0}^{N}|\boldsymbol{\nu}_{\varepsilon_z^i} - \boldsymbol{\nu}_{\text{in}}(\varepsilon_z^i)|^2\\
&\text{s.t. } V(\mathbf{p}) = \sum_e|\omega_e|/|Y| \in [v_{min}, v_{max}].
\end{aligned}
\tag{29}
$$

The sensitivities of both the objective function and the constraint can be assessed based on the preceding discussion. To solve equation (29), we employ the Method of Moving Asymptotes (MMA).

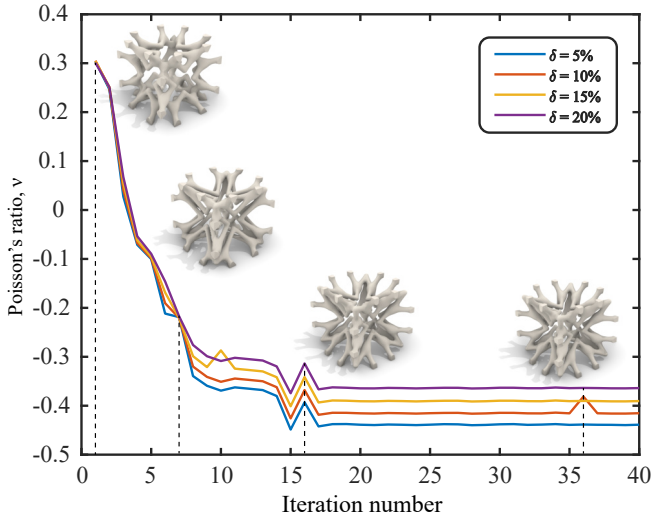

Figure 4. Evolution of Poisson's ratio during the optimization of the unit cell S_1^I.

Implementation We implement our algorithm in C++ language based on the open source framework provided by[2]. The sensitivity with respect to each geometry parameter is computed by substituting the corresponding normal velocity field into (26) before solving (20). We consider the stretch ratio ranging from 0 to 0.2, and sample 4 stretch ratios $(0.05, 0.10, 0.15, 0.20)$ in it to formulate (29).

2 More evaluations and results

2.1 Fabrication

The samples with different Poisson's ratios consisted of $3 \times 3 \times 6$ unit cells with lattice constant 22mm, as shown in Figure. 5, respectively. The lattice constant is set to ensure that the finest features within the unit cell can be properly processed during the manufacturing process. For the convenience of experimentation, clamping structures were added to both ends of the model. Because of the complexity of the shapes of the designed structures, the samples were fabricated with a light-curing 3D printer (S-TPU 500) (Supplementary Movie 3). The superelastic material used for printing is LUVOSINT X92A-2, which has a Poisson's ratio of 0.45, consistent with the design parameters. After printing, there are several post-processing steps required Figure. 7:

1. The first step is to remove any burrs or rough edges by sanding or polishing.
2. The second step is to blow off any surface powder or loose material.
3. The third step involves cleaning the printed object by immersing it in an alcohol solution.
4. After cleaning, the object should be removed from the alcohol solution and thoroughly dried by blowing off any remaining alcohol.
5. The fifth step is to soak in a specially formulated chemical solution.
6. Finally, the object is placed in a curing chamber to solidify and complete the post-processing. the post-processing.

The aforementioned fifth and sixth steps can make the surface of the printed samples smoother, thereby making them closer to the design models.

2.2 Measurement

For Poisson's ratios, the samples in Figure. 5 was tested via longitudinal uniaxial tensionTension tests were performed using a SHIMADZU AGS-X machine as illustrated in Supplementary Movie 4. To achieve significant deformation, we applied a longitudinal displacement of 30% at a speed of 0.5mm/sec. We placed markers on the samples and recorded the entire tension process using a camera (Canon D7) with 1920×1080 pixels. The coordinate information of the markers was obtained using Adobe Effect and analyzed with MATLAB. To mitigate the influence of fixed boundaries on the measurement values, we computed an approximate value for the Poisson's ratio of the unit cell based on the deformation of the central square region in

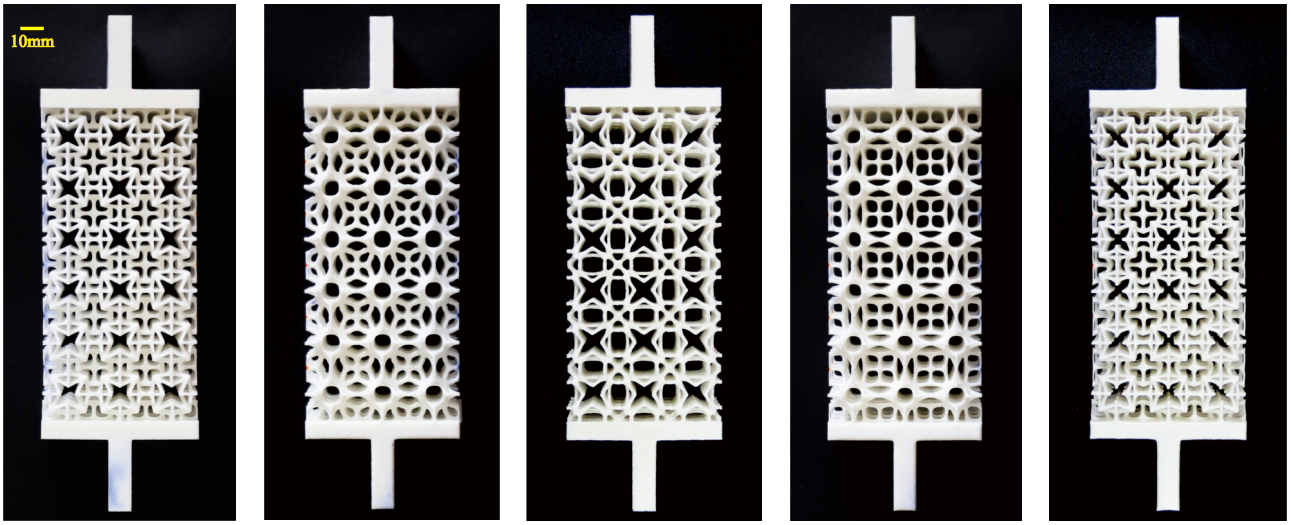

Figure 5. The illustration of 3D-printed models

the sample. The feasibility of this method has been confirmed in other articles[4].The Poisson's ratios were approximated using the formula:

$$\varepsilon_x = \frac{((\tilde{x}_1 - \tilde{x}_0) + (\tilde{x}_3 - \tilde{x}_2))}{((x_1 - x_0) + (x_3 - x_2))} - 1, \quad \varepsilon_z = \frac{((\tilde{z}_0 - \tilde{z}_2) + (\tilde{z}_1 - \tilde{z}_3))}{((z_0 - z_2) + (z_1 - z_3))} - 1, \quad \nu_x = -\frac{\varepsilon_x}{\varepsilon_z}, \tag{30}$$

where parameters such as x_0 are shown in the Figure. 6. The calculation method for ν_y is exactly the same, and will not be described again.

3 Captions

3.1 Movies

Movie 1 This video illustrates the pipeline of the algorithm proposed in our paper.

Movie 2 This video illustrates the uniaxial stretching simulation process of unit cells with different Poisson's ratio behaviors and parameter interpolation of sequence I .

Movie 3 This video illustrates the manufacturing and post-processing steps.

Movie 4 This video showcases the experimental process and the results for sequence III.

3.2 Raw data

The STL files of the optimized unit cells are categorized based on their functionalities and stored in the folder named 'Raw Data'.

References

1. Panetta, J. *et al.* Elastic textures for additive fabrication. *ACM Trans. Graph.* **34** (2015).

2. Panetta, J., Rahimian, A. & Zorin, D. Worst-case stress relief for microstructures. *ACM Trans. Graph.* **36** (2017).

3. Nakshatrala, P., Tortorelli, D. & Nakshatrala, K. Nonlinear structural design using multiscale topology optimization. part i: Static formulation. *Comput. Methods Appl. Mech. Eng.* **261-262**, 167–176 (2013).

4. Zong, H., Zhang, H., Wang, Y., Wang, M. Y. & Fuh, J. Y. On two-step design of microstructure with desired poisson's ratio for am. *Mater. & Des.* **159**, 90–102 (2018).

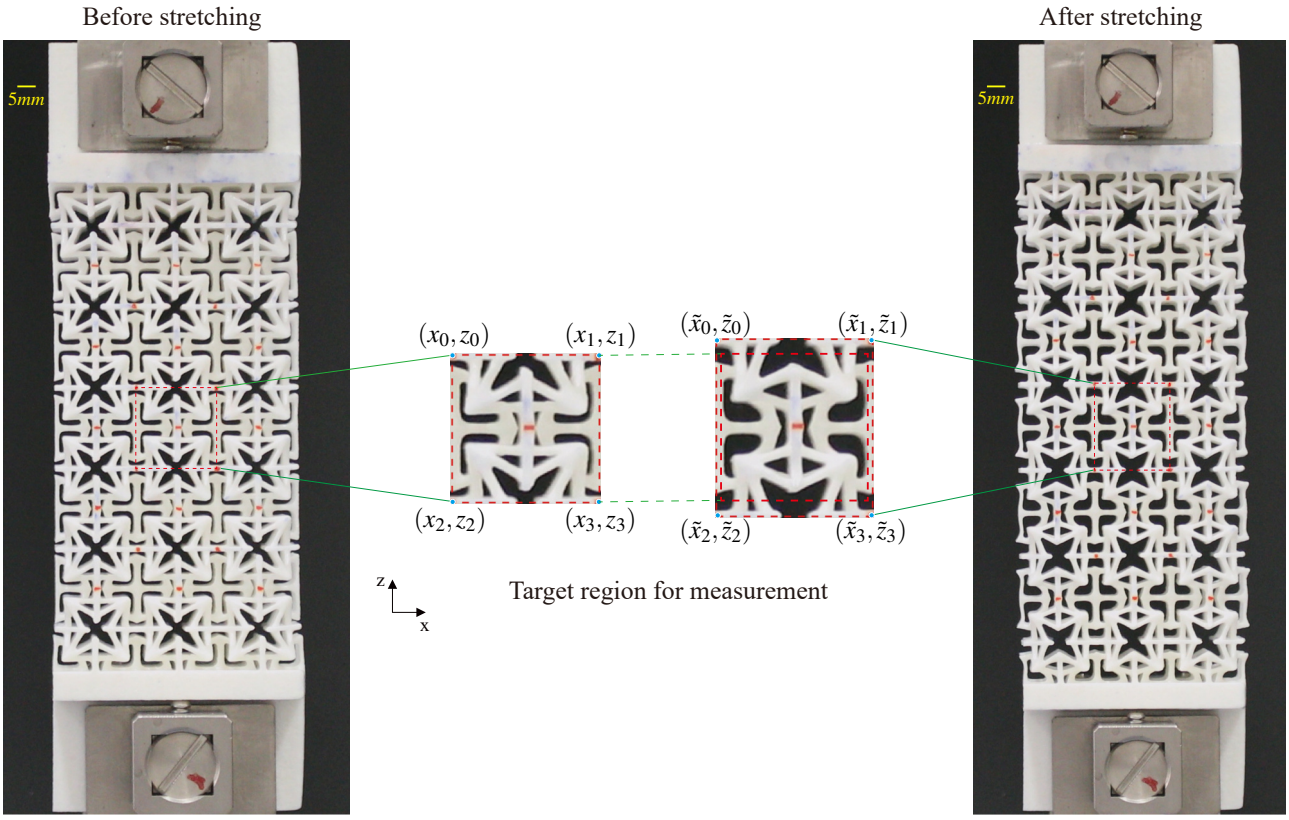

Figure 6. Calculation of Poisson's ratio in experiments

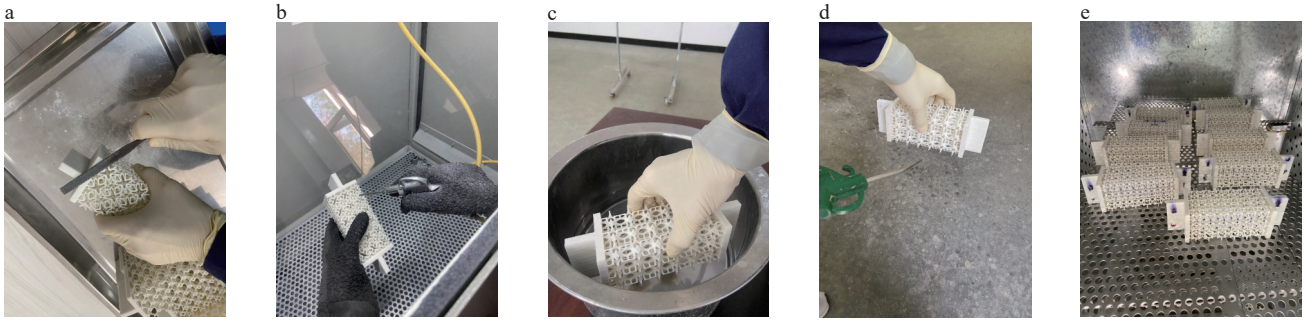

Figure 7. Post-processing after printing

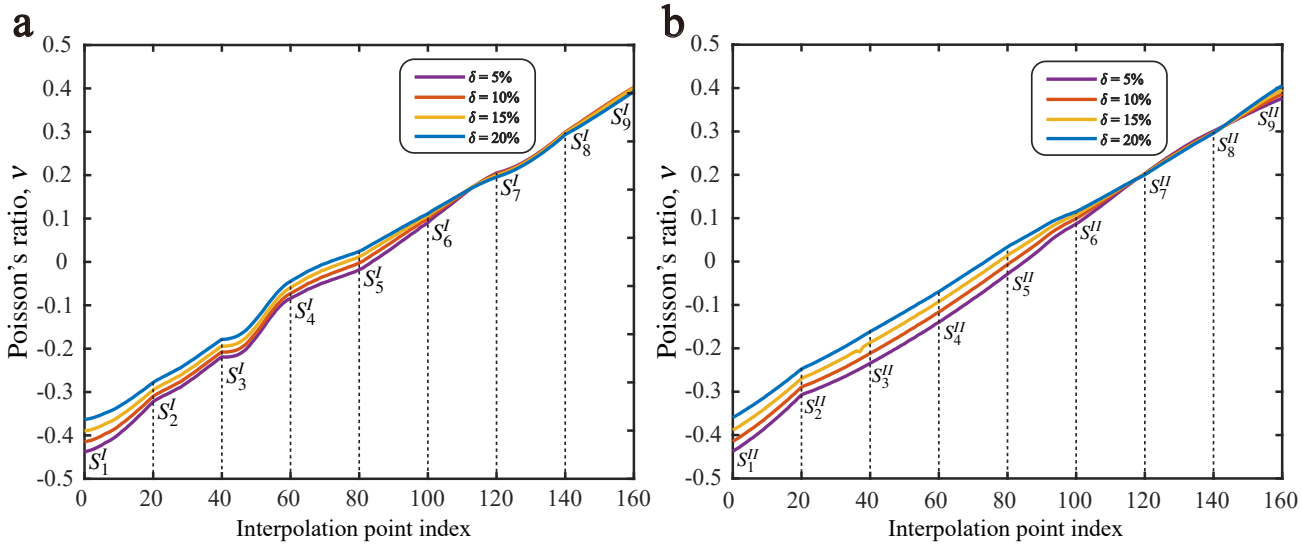

Figure 8. **a** The Possion's ratios of the interpolation cells of sequence I (in **b** for sequence II)under tensile strains of 5%, 10%, 15%, and 20%.