Supporting Information

Polymerization Threshold Validation

In order to validate the dark field microscopical measurements, presented in Section 5.2 of the main paper, we optically evaluate the fabricated lines on separate samples before sample development with an *in situ* microscope. The reported data from *in situ* and dark field microscopy matches well and gives a consistent trend for the threshold (see Figure 12).
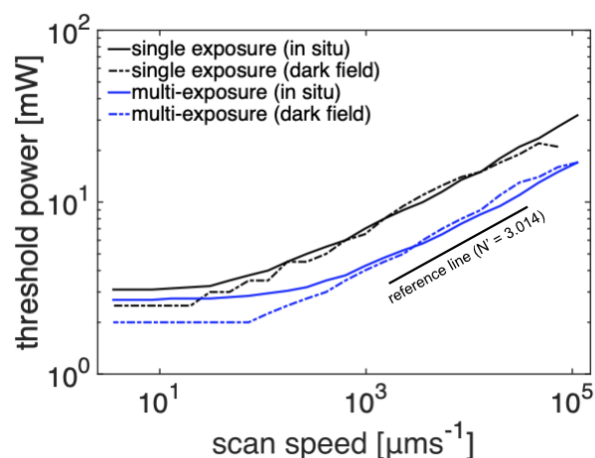

Figure 12: Validation of the dark field microscopical measurements by *in situ* microscopy. The solid lines depict the *in situ* measurements and the dash-dotted lines show the measurements from the dark field microscopy.

Laser intensity Distribution

The intensity near the focal point of the laser $I(\mathbf{x})$ is commonly approximated with a Gaussian distribution. In cylindrical coordinates, the local intensity can be expressed as a function of the radius r and the axial position z, relative to the focal point

43

$$I(r, z) = I_0 \frac{w_0^2}{w^2(z)} e^{-2r^2/w^2(z)}. \tag{27}$$

Here, $I_0 = I(r = 0, z = 0)$ is the laser peak intensity. The beam radius $w(z)$ at position z is

$$w(z) = w_0 \sqrt{1 + \left(\frac{z\lambda}{w_0^2 \pi} \right)^2}, \tag{28}$$

where λ is the wavelength of the laser in the liquid photoresist. This expression is often used to determine the transmitted power of a laser beam as a function of its peak intensity. Integrating the intensity distribution, using the shell method, yields the power P transmitted within a circle of radius r at position z

$$P(r, z) = 2\pi \int_0^r r^* I(r^*, z) \mathrm{d}r^* = \frac{\pi I_0 w_0^2}{2} \left[1 - e^{-2r^2/w^2(z)} \right]. \tag{29}$$

The total power that is transmitted by the beam can be determined by setting $r \to \infty$

$$P_0 = \lim_{r \to \infty} P(r, z = 0) = \frac{1}{2} \pi I_0 w_0^2. \tag{30}$$

We use this expression to determine the peak intensity I_0 during irradiation with a specified laser power P_0. On the contrary, the peak intensity of the beam at an arbitrary position z can be determined as the limit of the transmitted power within a circle of radius $r \to 0$, divided by the corresponding area

$$I(r \to 0, z) = \lim_{r \to 0} \frac{P_0 \left[1 - e^{-2r^2/w^2(z)} \right]}{\pi r^2} = \frac{2P_0}{\pi w^2(z)}. \tag{31}$$

Complementary Experimental Data

The experimental validation of the numerical simulations, given in Section 5.3, shows that shrinkage-induced deformations and structural failures are dependent on the laser trajectories. In order to verify consistency in the structural behavior, we fabricated the described structures with different hatching distances. We fabricate 5 samples for each configuration. Figure 13 shows SEM images of suspended plates that have been fabricated with the same laser trajectories as discussed in Section 5.3 (Top rows: Meander pattern, bottom rows: Snake pattern), for two exemplary hatching distances.

While the structures in the top rows retain the square geometry to a large degree, significantly stronger deformations and structural failure can be seen in the structures in the bottom rows. For a hatching distance of $0.3\,\mu m$, the structures consistently have strong deformations at the top edge. The structures fabricated with a hatching distance of $0.5\,\mu m$, consistently break at the bottom-right corner and in the upper third along the left edge.
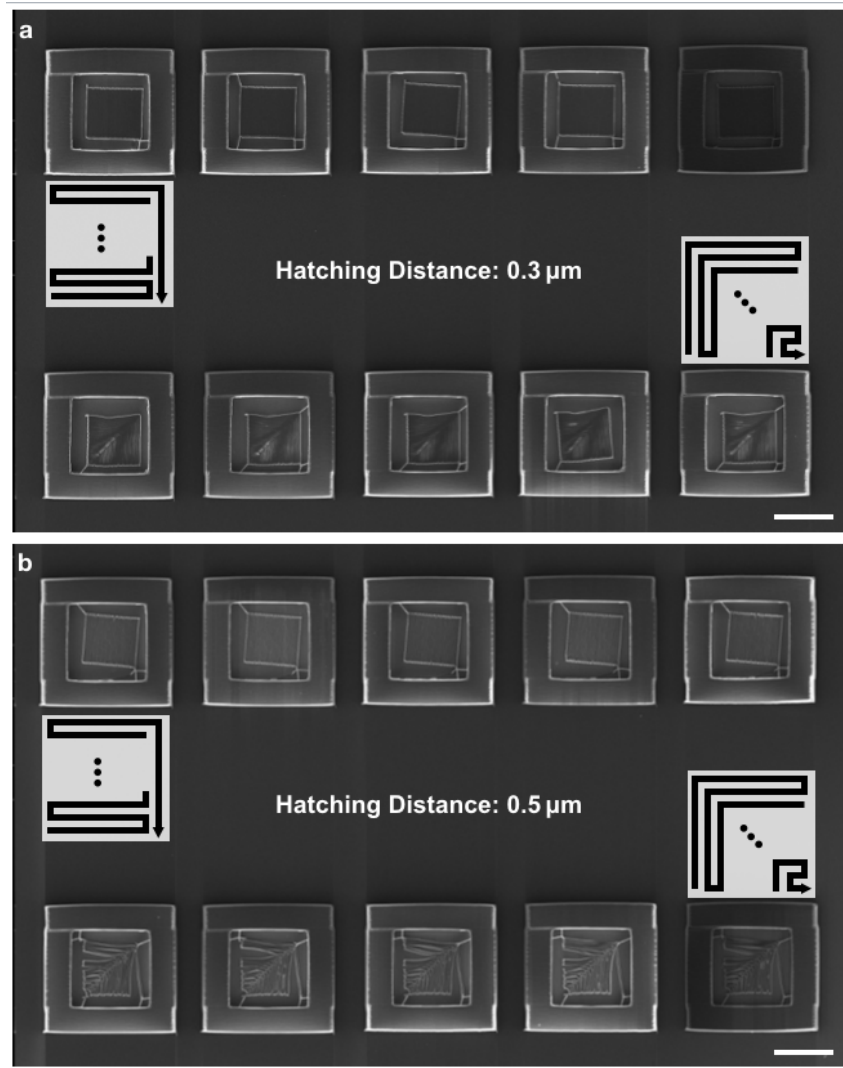
Figure 13: Influence of the hatching distance on the structural deformations. SEM images of sheets fabricated with varying hatching distances and two different laser trajectories. Each sample has been fabricated five times to verify consistency of the observed deformations.

Computational Workflow

The discussed ANN and the underlying semi-empirical model are implemented in multiple programming languages. A schematic workflow of the codes and the produced data is given in Figure 14. The stochastic trajectory design is implemented with the software Matlab. The identified trajectories are parameterized and embedded into Fortran subrou-

tine files, which are created individually for each trajectory. Besides, the parameterized representations of the trajectories are stored in a separate file, which is later complemented with the simulation results to create a training set for the ANN.

A custom written input file contains the simulation data and is run with an Abaqus solver in conjunction with each of the created subroutine files. The simulation results are saved in separate output databases which can be accessed with the provided Python interface. The extracted data is then added to the training dataset. The ANN is implemented, optimized and evaluated with the software Matlab.
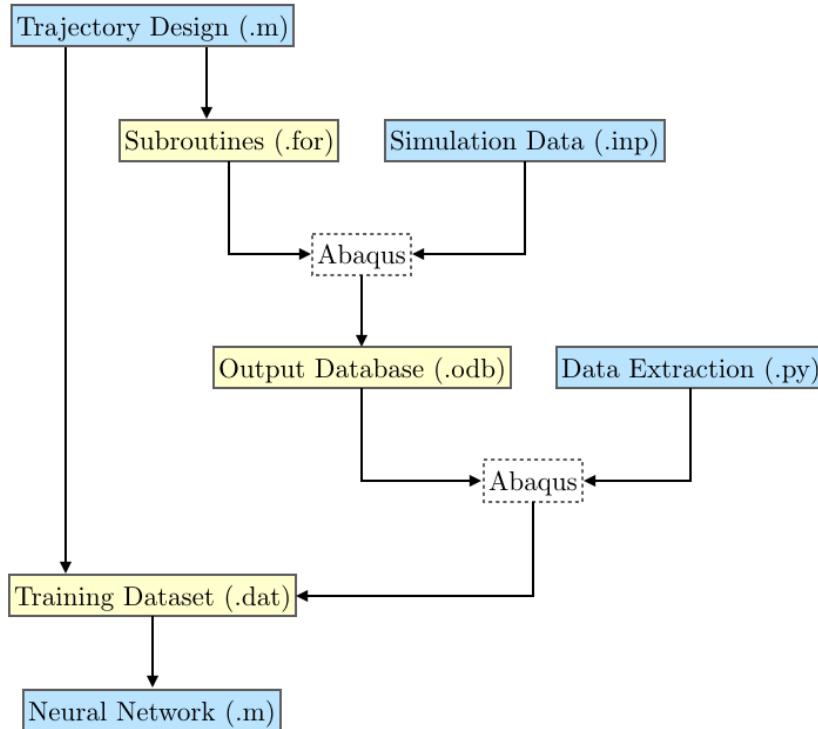

Figure 14: Data flow for the presented semi-empirical numerical model. The formats of the code blocks are given in brackets. The blue shaded components indicate programmed codes, whereas the yellow shaded components signify data and codes that are created by the program.