

Supplementary Materials for
**Computational discovery of microstructured composites with optimal
stiffness-toughness trade-offs**

Beichen Li *et al.*

Corresponding author: Wan Shou, wanshou@mit.edu, wshou@uark.edu;
Wojciech Matusik, wojciech@csail.mit.edu

Sci. Adv. **10**, eadk4284 (2024)
DOI: 10.1126/sciadv.adk4284

This PDF file includes:

Notes S1 to S5
Figs. S1 to S33
Tables S1 to S15
References

Note S1 Definition of Microstructures

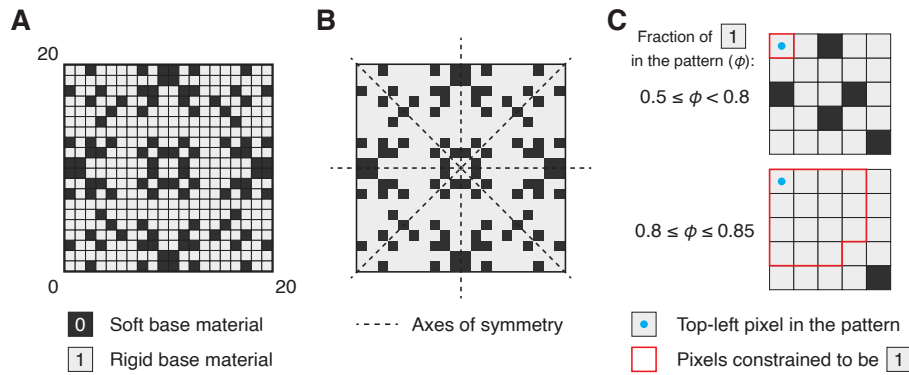


Fig. S1. Definition of microstructures. (A) An example microstructure. (B) Symmetry constraints on microstructure patterns. (C) Structural constraints on microstructure patterns depending on volume fractions of the rigid base material.

As the structural unit of microstructured composites in this work, a microstructure is digitally represented as a 20×20 binary matrix that uniquely defines the spatial arrangement of two base materials in a composite (Fig. S1A). The matrix is also called a *microstructure pattern*. The soft base material is indicated by a value of 0 in a microstructure pattern while the rigid material corresponds to 1. As a microstructure pattern is conceptually similar to an image, we refer to its elements as *pixels* to avoid confusion with those in the finite element method (FEM).

To construct the design space for computational exploration, we additionally pose two types of constraints on microstructure patterns. Symmetry constraints (Fig. S1B) require all microstructures to be invariant to reflections and 90-degree rotations so that they have identical mechanical properties in horizontal and vertical directions. Structural constraints (Fig. S1C) limit the volume fraction of the rigid material, denoted by ϕ , between 0.5 and 0.85. Also, pixels around the corner of a pattern are restricted to 1 depending on ϕ . If $0.5 \leq \phi < 0.8$, only the corner pixel is restricted; otherwise ($0.8 \leq \phi \leq 0.85$), the rule extends to every pixel whose center lies within a 4-pixel-wide radius from the corner vertex. The upper bound $\phi \leq 0.85$ ensures reliable and consistent toughness measurements in mechanical testing because the fracture behaviors of very stiff microstructures ($\phi > 0.85$) tend to be highly stochastic and result in low toughness (59). The lower bound $\phi \geq 0.5$ excludes designs with inferior Young's modulus and toughness due to a predominant influence from the soft material. All constraints aggregated, the design space of microstructures contains more than 10^{16} possibilities.

Our choice of pattern solution is primarily constrained by the 3D printer. A test specimen must contain a sufficient number of microstructure units to achieve proper homogenization and minimize the boundary effect (Fig. S27B). If the resolution is too

high, the fine features of a microstructure pattern will be completely blurred during 3D printing as the size of each individual pixel becomes comparable to a material droplet. However, if the resolution is too low, it will take a substantial number of microstructure units to realize such homogenization (otherwise, the large interfaces between base materials will become flaws and weaken the mechanical performance). This makes the resulting specimens virtually indistinguishable from single-phase materials. Thus, we used a 20×20 resolution with a pixel size of $260 \mu\text{m}$ to create a meaningful design space while preserving visible pixel boundaries in printed specimens.

Note S2 Interface Engineering

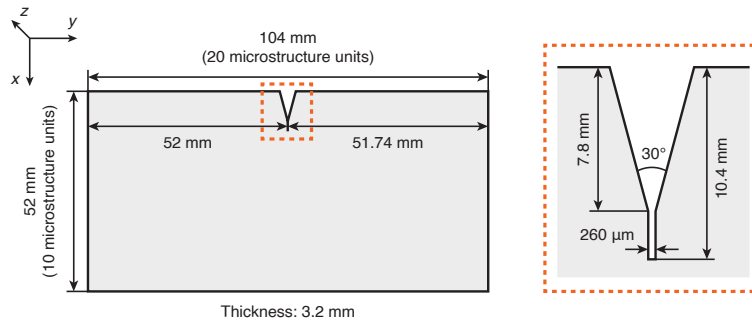


Fig. S2. Dimensions of a toughness specimen with the area around the notch zoomed in. Coordinate axes indicate the orientation of the specimen in 3D printing.

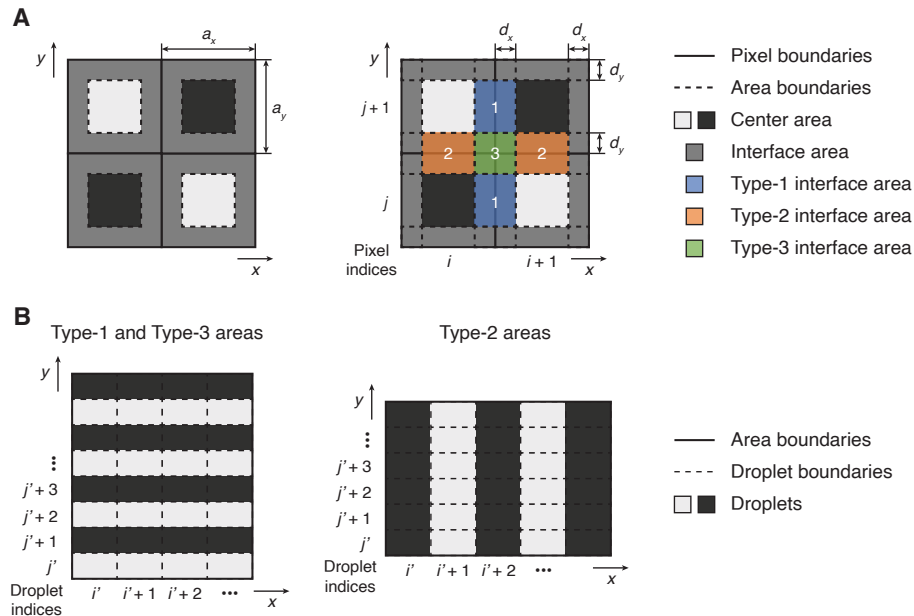


Fig. S3. Schematics of interface engineering. (A) Definition and classification of interface areas. (B) Droplet arrangements in interface areas. All coordinate axes conform with the definitions in Fig. S2.

Upon examination of printed specimens, we noticed that the interface between base materials is relatively strong but can still be the weakest spot during crack propagation (60). Thus, we engineered the interface to enhance its strength and improve the toughness of microstructures. This was done by applying manually designed droplet patterns in *interface areas*. As illustrated in Fig. S3, interface areas are located around shared edges or vertices of adjacent pixels, where one base material transitions into the other. There are three types of interface areas in total. Let (i, j) be the x and y indices of a pixel in an entire specimen model. Type-1 areas connect two neighboring pixels (i, j) and $(i + 1, j)$ along the x direction, i.e., the direction of crack propagation. Type-2 areas connect two neighboring pixels (i, j) and $(i, j + 1)$ along the y direction, i.e., the pulling direction in tensile testing.

Pixel dimensions		The center area		The interface area	
$x (a_x)$	$y (a_y)$	x	y	$x (d_x)$	$y (d_y)$
3	5	1	3	1	1
3	6	1	2	1	2
4	5	2	3	1	1
4	6	2	2	1	2

Table S1. Possible dimensions of a pixel and its subdivided areas after quantization into droplets for voxel printing. All units are droplets.

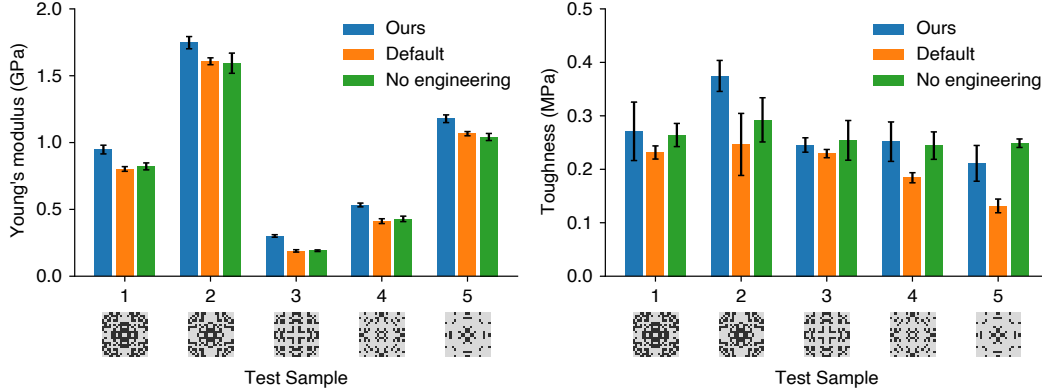


Fig. S4. Comparison of microstructure properties under different interface configurations. The tested configurations include: default to the printer (default), not engineered (no engineering), and engineered (ours). Standard deviations from multiple measurements are reflected in error bars.

Type-3 areas join all four surrounding pixels. Whenever an interface area covers two pixels of different material assignments, droplets in that area are specially arranged to approximate a 50/50 mixture of base materials. Similarly, let (i', j') be the x and y indices of a droplet in the specimen model. Droplets appear in alternating rows in Type-1 and Type-3 areas, where each droplet (i', j') follows the parity of $i + j'$. In contrast, droplets appear in alternating columns in Type-2 areas, each following the parity of $j + i'$. Such patterns remain constant in the z direction.

In the context of voxel printing, a pixel may contain 3×5 , 3×6 , 4×5 , or 4×6 droplets after quantization. Table S1 provides the quantized dimensions of a pixel's internal area contingent on the pixel size. Quantized dimensions of interface areas are then derived from the pixels they intersect with. Type-1 and type-3 areas may contain 2×2 or 2×3 droplets, while type-2 areas vary among 1×2 , 1×3 , 2×2 , and 2×3 droplets in size.

We compared our method of engineering the interface between base materials against two baseline approaches. By default, the 3D printer applies prescribed droplet configurations in interface areas that are invisible to the user. Conversely, the interface is completely user-specified in voxel printing mode. Therefore, it is possible to perform no engineering and make transitions at pixel boundaries as sharp as possible. To test these interface options, we randomly drew 5 samples from microstructures with $0.5 \leq \phi \leq 0.75$ and compared the mechanical properties of their printed specimens (Fig. S4).

Note S3 Computational Pipeline

3.1 Simulation

The simulator estimates the performance of microstructures without conducting physical measurements. Given an input microstructure pattern, it simulates the actual mechanical testing process on printed specimens using FEM. Young's modulus and toughness are then computed from recorded stress-strain data. The two virtual testers inside the simulator use the same material model but differ in implementation details and parameter settings.

The nonlinear stress-strain response of the base materials is approximated using a 2D Neo-Hookean model. We followed the strain energy density function in Sifakis et al. (61):

$$W = \frac{\mu}{2} (I_1 - 2 - \ln J) + \frac{\lambda}{2} (\ln J)^2, \quad (1)$$

where μ and λ are the Lamé parameters, $J = \det(F)$ is the determinant of the deformation gradient (F), and $I_1 = \text{tr}(C)$ is the first invariant of the right Cauchy-Green deformation tensor ($C = F^T F$). μ and λ are derived from Young's modulus (E) and Poisson's ratio (ν) as follows

$$\begin{aligned}\mu &= \frac{E'}{2(1+\nu')}, \\ \lambda &= \frac{E'\nu'}{(1+\nu')(1-2\nu')},\end{aligned}\tag{2}$$

where E' and ν' are converted from E and ν as follows (62)

$$\begin{aligned}\nu' &= \frac{\nu}{1+\nu}, \\ E' &= \frac{E(1+2\nu)}{(1+\nu)^2}.\end{aligned}\tag{3}$$

The Young's moduli and Poisson's ratios of the base materials were regarded as optimizable parameters in system identification (Section 3.2).

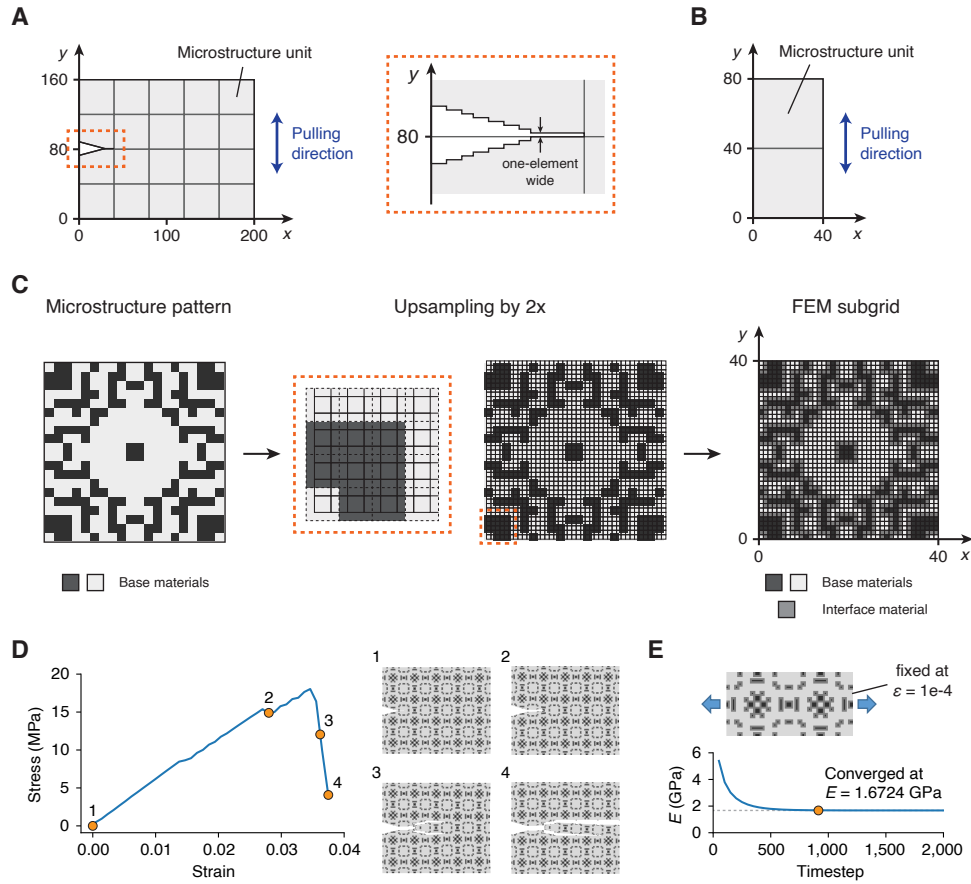


Fig. S5. Configuration of the FEM simulator. (A) Dimensions of the FEM grid in the toughness virtual tester. (B) Dimensions of the FEM grid in the Young's modulus virtual tester. (C) upsampling from a microstructure pattern to a 40x40 FEM subgrid. (D) An example simulation process of the toughness virtual tester. (E) An example simulation process of the Young's modulus virtual tester.

The virtual tester for toughness operates on a regular grid of 200x160 quadrilateral finite elements (Fig. S5A) where each node has four degrees of freedom. The grid contains 5x4 tiles of microstructure units where each unit covers a 40x40 sub-grid. Notably, a microstructure unit doubles the resolution of the pattern due to modeling the interface between base materials. The

interface was modeled as a third homogeneous material, referred to as the *interface material*. Finite elements of the interface material and the base materials are equally sized. This is evidenced by our microscopic observation of printed specimens, which indicates that the width of the interface is typically around $100\ \mu\text{m}$ and comparable to half of the pixel size. The computation of material assignment in a sub-grid is analogous to image upsampling. As shown in Fig. S5C, the FEM subgrid is superimposed on the original pattern where the center of the bottom-left element aligns with the center of the bottom-left pixel. An element inherits the corresponding pixel value if it is located entirely within a pixel; otherwise, the material assignment depends on the values of surrounding pixels in the same way as interface engineering. To mimic the pre-cracking of real specimens, a single-element-wide notch and a triangular cut were applied at the grid position of $(0, 80)$. The “removed” elements were assigned with a void material and do not generate nodal forces under deformation. Overall, this configuration correlates well with a printed specimen despite a simpler geometry to reduce computational cost. Dirichlet boundary conditions were enforced on the displacement in the y -direction during testing. Nodal forces in each element were computed from the Neo-Hookean model using four Gaussian quadrature points. Nodal velocities and displacements were then obtained from explicit time integration. Crack initiation and propagation were modeled on a per-element basis. To that end, strain energy densities were computed at the quadrature points together with nodal force contributions. An element was considered failed and reassigned with a void material once the maximum strain energy density at quadrature points exceeded a certain threshold specific to each material. Simulation terminated when the global strain went beyond 0.3 or the measured stress fell below 20% of the ultimate tensile strength, with the only exception being the soft material for which the maximum strain was unlimited. In the end, toughness was calculated by integrating the stress-strain curve.

While it is possible to compute nodal velocities and displacements using an implicit solver, we chose an explicit solver since explicit solvers are generally preferred for nonlinear FEM simulations that involve dynamic failure. In particular, the virtual tester must operate at a sufficiently small time step (around 10^{-7} s) to precisely capture the crack propagation in microstructures and maintain numerical stability. Under such a strict requirement, explicit solvers are more efficient due to simpler implementation and better compatibility with many-core hardware accelerators like graphical computing units (GPU). However, we noticed that the virtual tester ran too slow at a strain rate that matched actual mechanical testing. In this case, simulating a microstructure till an assumed failure strain of 0.02 required 1.62×10^8 time steps, taking almost one hour to finish on an NVIDIA Tesla V100 GPU under maximum throughput. The running time is impractical to our computational pipeline because the simulation budget will be too tight to allow for a proper exploration of the combinatorial design space. As a workaround for this challenge, we used a much larger strain rate (equal to 20) to speed up the simulation while introducing system-wide numerical damping to model material viscosity. Once the velocity field was updated in every time step, nodal velocities $v_{i,j}$ were smoothed by a 3×3 filter defined as

$$v'_{i,j} = (1 - \alpha)v_{i,j} + \frac{\alpha}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 v_{i+k,j+l}, \quad (4)$$

where $\alpha = \gamma\Delta t$ is the product of the time step Δt and a constant damping coefficient γ . Under the assumption that all nodes have identical mass, this step effectively interpolates between the momentum of each node and the mean momentum in a 3×3 neighborhood. A larger γ leads to stronger damping effects and potentially accounts for more material viscoelasticity. Our observations suggested that larger γ values work better with stiffer microstructures in suppressing oscillation. As there is no trivial way to determine the value of γ for all microstructures, we optimized γ in system identification instead.

The virtual tester for Young’s modulus has a much smaller grid of 80×40 quadrilateral elements, consisting of 2×1 microstructure units (Fig. S5B). The material assignment to elements was computed using the same image upsampling method and no notch or triangular cut was added. This configuration qualitatively matches the gauge area of an ASTM D638 Type 1 specimen. A constant displacement boundary condition was enforced in the y -direction where the global strain remained fixed at 10^{-4} . Nodal forces were computed from the aforementioned Neo-Hookean model. Explicit time integration was used to update velocities and displacements. Negative exponential damping was applied to the velocity field as follows

$$v'_{i,j} = v_{i,j}e^{-\gamma\Delta t}, \quad (5)$$

where $\Delta t \approx 10^{-7}$ s was used as the time step and the damping coefficient γ was optimized through system identification. The solver was allowed to reach a static equilibrium within 10^4 time steps, after which Young’s modulus was derived from the gauge stress. Although an implicit linear elasticity solver is usually used to virtually measure Young’s modulus, we directly reused the explicit solver in the toughness simulator for simplicity. Note that the Young’s modulus virtual tester is not the bottleneck of the whole simulation.

In pursuit of maximal computation efficiency, our customized simulator was implemented using the Taichi programming language (71), an emerging high-performance programming language tailored to computer graphics applications including physics-based simulation. Taichi compiles our source code written in Python syntax into optimized machine programs that,

when executed, exploit the massive parallel computing power of GPUs. However, we found that simulating microstructures sequentially resulted in low GPU utilization due to inadequate workload. Our solution was adding a batching mechanism to enable concurrent simulation of multiple microstructures, where each microstructure can be evaluated using a unique set of optimizable parameters. This allows for maximum GPU usage regardless of its power, resulting in a much higher throughput.

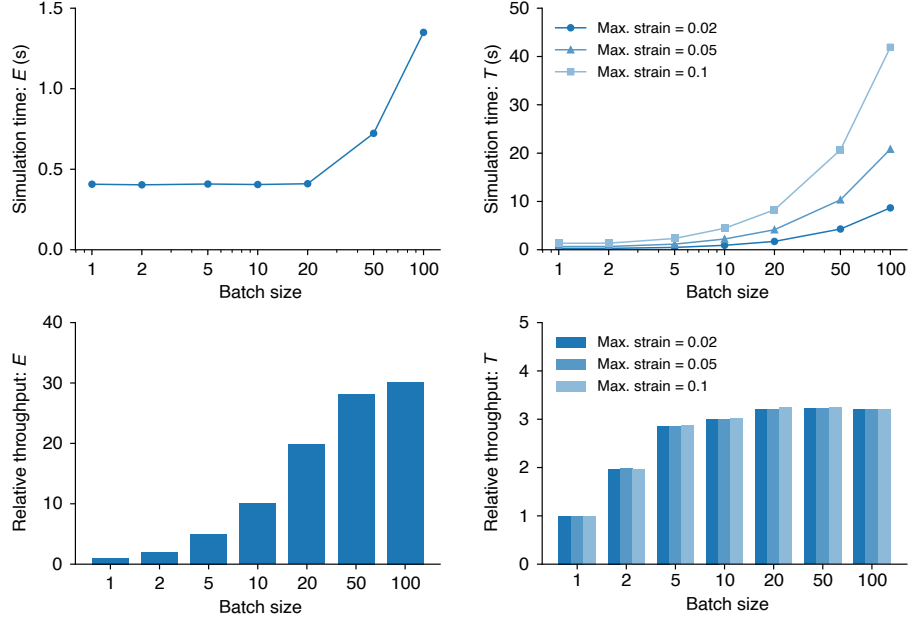


Fig. S6. Time consumption and throughput of the simulator under various input batch sizes. The column on the left contains results for Young’s modulus and the column on the right shows results for toughness. The toughness virtual tester was evaluated under three strain limits: 0.02, 0.05, and 0.1.

We evaluated simulation speed by timing the simulator under different batch sizes and strain limits. All performance data was collected on a machine with 8 CPU cores and an NVIDIA Tesla V100 GPU averaged across 5 consecutive runs. Results are shown in Fig. S6. For Young’s modulus, simulation only becomes slower when at least 20 microstructures are batched together. While increasing the batch size beyond this point leads to a much longer simulation time, the throughput continues to improve, implying that GPU utilization is still approaching its maximum. A batch size of 100 is required to reach a peak throughput of less than 0.015 s per sample. For toughness, the highest throughput occurs at a batch size of 20, where it takes less than 0.2 s to stretch a sample to $\varepsilon = 0.05$. Relative to the sequential setting, our batching mechanism yields a $30\times$ boost in throughput for Young’s modulus and more than $3\times$ for toughness regardless of the strain limit.

3.2 System Identificaton

System identification finds the optimal parameter setting of the simulator to match simulation results with physical measurements (Fig. S7). We first define some mathematical notations before providing a formal description of the problem. For any microstructure x , let $f(x, \theta) : X \times P \rightarrow \mathbb{R}$ denote the virtual tester for Young’s modulus with a parameter setting θ , where $X \subset \{0, 1\}^{20 \times 20}$ is the design space of microstructures and $P \subset \mathbb{R}^7$ is the feasible set of θ . The estimated Young’s modulus of x is directly obtained from $\hat{E} = f(x, \theta)$. Similarly, let $g(x, \eta) : X \times Q \rightarrow C[0, \infty)$ denote the virtual tester for toughness with a separate parameter setting η , where $Q \subset \mathbb{R}^{10}$ is the feasible set of η . The output g is not a single number but a stress-strain curve $\hat{\sigma}(\varepsilon)$ defined as a continuous function over non-negative strains. Thus, the toughness of x is estimated by computing the following integral

$$\hat{T} = I[\hat{\sigma}] = \int_0^\infty \hat{\sigma}(\varepsilon) d\varepsilon. \quad (6)$$

Both θ and η are tuples of optimizable simulation parameters, including

- Reference Young’s moduli of base materials (denoted as TB and VW) and the interface material (denotes as IF): E_{TB}, E_{VW}, E_{IF} ;
- Reference Poisson’s ratios: $\nu_{TB}, \nu_{VW}, \nu_{IF}$;

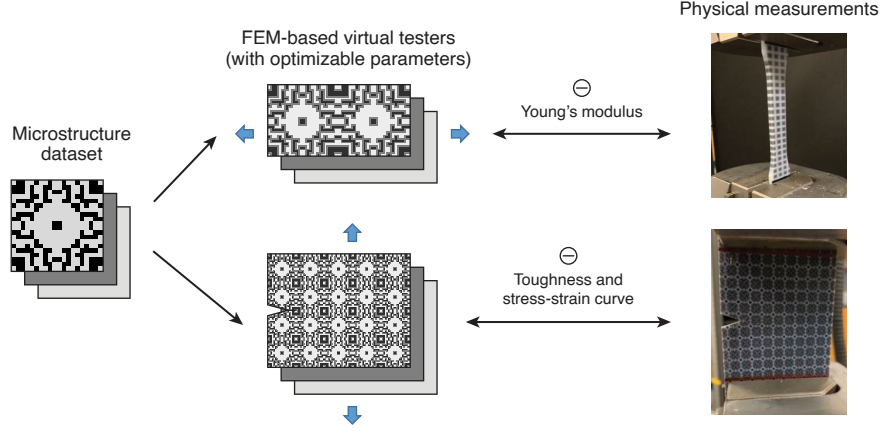


Fig. S7. Schematics of the system identification problem. Each virtual tester has a separate set of optimizable parameters. ' \ominus ' denotes error minimization via parameter optimization.

- Failure thresholds of strain energy density: W_{TB}, W_{VW}, W_{IF} ;
- A system-wide damping coefficient: γ .

Since fracture simulation is not involved in predicting Young's modulus, we have

$$\begin{aligned}\theta &= (E_{TB}, E_{VW}, E_{IF}, \nu_{TB}, \nu_{VW}, \nu_{IF}, \gamma) \\ \eta &= (E_{TB}, E_{VW}, E_{IF}, \nu_{TB}, \nu_{VW}, \nu_{IF}, W_{TB}, W_{VW}, W_{IF}, \gamma).\end{aligned}\quad (7)$$

Based on optimizable parameters, the prediction errors of virtual testers were minimized over a group of microstructures $X_c = \{x_i | i = 1, 2, \dots, N\}$, called the system identification dataset. The optimization process solved the following problems

$$\begin{aligned}\theta^* &= \operatorname{argmin}_{\theta} \sum_{i=1}^N w_i \frac{|\hat{E}_i - E_i|}{E_i} \\ \eta^* &= \operatorname{argmin}_{\eta} \sum_{i=1}^N w_i \frac{|\hat{T}_i - T_i| + \tilde{\lambda} I[|\hat{\sigma}_i - \sigma_i|]}{T_i},\end{aligned}\quad (8)$$

where $\hat{E}_i, \hat{T}_i, \hat{\sigma}_i$ are the predicted Young's modulus, toughness, and stress-strain curve of each microstructure x_i ; E_i, T_i, σ_i are the corresponding physical measurements. For Young's modulus, the objective is simply the average relative prediction error over the dataset. For toughness, the objective combines errors in both toughness values and stress-strain curves. Since stress-strain curves are functions, the error between two curves is calculated by integrating their absolute difference. Geometrically, this represents the symmetric difference of the areas under the curves. The two error terms are balanced by a multiplier $\tilde{\lambda} = 1$.

Additionally, as there are fewer optimizable parameters than microstructures in the dataset, we postulate that the simulation model might have an insufficient capacity to capture the variation of experiment data. Thus, each microstructure x_i is weighted by $w_i \in [0, 1]$ to indicate priority. The weight is defined as the product of two components $w_i = w_{1i} w_{2i}$ that reflect two intuitive principles. First, microstructures with better mechanical properties are prioritized for accuracy since Pareto-optimal microstructures are our focus. To that end, the physical measurements $y = (E, T)$ of microstructures in the dataset were normalized into $\tilde{y} = (\tilde{E}, \tilde{T})$ in a unit square, and a piecewise-linear Pareto front was computed in the normalized performance space. Then, the weight component w_{1i} of microstructure x_i was derived from the Euclidean distance between \tilde{y}_i and the Pareto front, denoted by d_i , via the following function (Fig. S9A)

$$w_1(d) = \begin{cases} 1 & 0 \leq d < 0.2 \\ 1.4 - 2d & 0.2 \leq d < 0.6 \\ 0.2 & d \geq 0.6 \end{cases} \quad (9)$$

Second, to prevent any bias from an uneven sample distribution in the performance space, we introduce w_{2i} as a penalty term for excessively densely distributed microstructures. The distribution is estimated using a density heatmap $\rho(\tilde{y})$ over the normalized

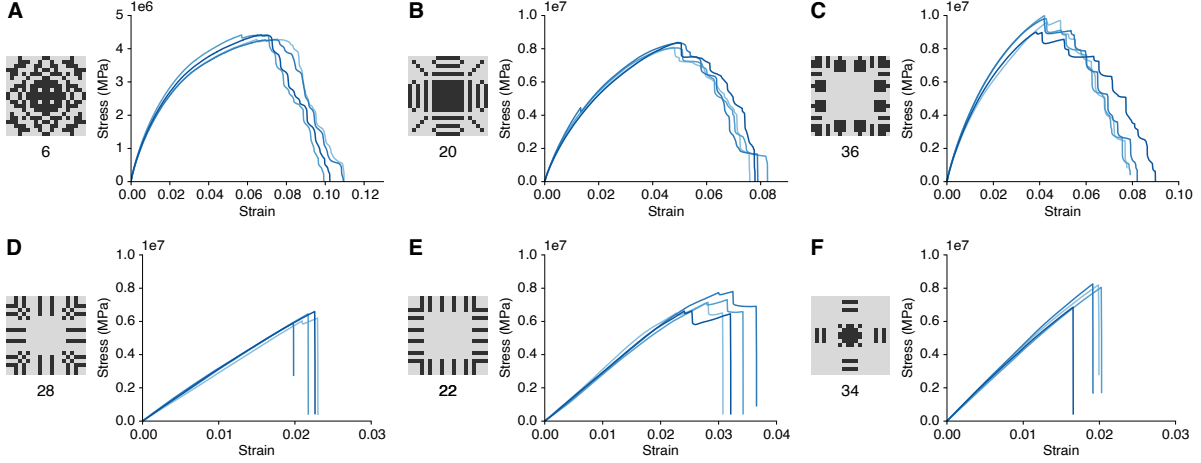


Fig. S8. Measured stress-strain curves from the toughness test of six microstructures discovered by our pipeline. The microstructures (A-F) are selected from random stratified sampling to cover a wide range of ϕ . We showcase the stress-strain curves of all tested specimens (distinguished by color) for each design.

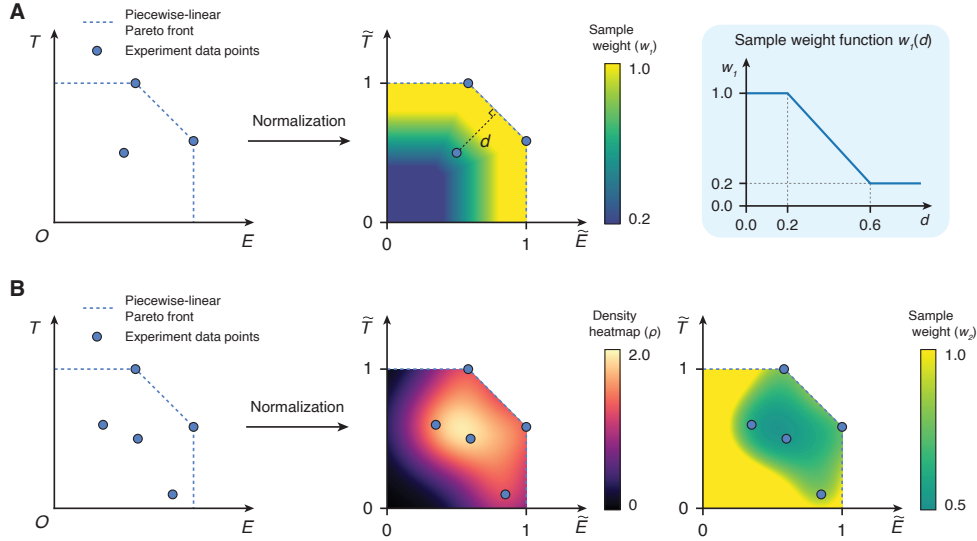


Fig. S9. Calculation of sample weights in system identification. (A) An example of deriving the first sample weight component, w_1 , in system identification using three data points. (B) An example of deriving the second sample weight component, w_2 , using five data points.

performance space (Fig. S9B). The density heatmap is derived from a Gaussian mixture model

$$\rho(\tilde{y}) = \sum_{i=1}^N \exp(-\gamma_d \|\tilde{y}_i - \tilde{y}\|_2^2) \quad (10)$$

where γ_d is a parameter that controls the spread of each Gaussian function. Quantitatively, $\rho(\tilde{y})$ estimates the number of microstructures in the vicinity of any position \tilde{y} , and varies smoothly between adjacent samples. Then, w_{2i} is defined as a weight discount for every sample x_i whose density value exceeds a certain threshold, ρ_0 .

$$w_2(\tilde{y}) = \min \left\{ \frac{\rho_0}{\rho(\tilde{y})}, 1 \right\} \quad (11)$$

In this study, we found $\gamma_d = 200$ and $\rho_0 = 2$ to be a desirable parameter setting. Noteworthy, the design principles of sample weights generalize to other problem domains. Proper modifications can be made when our approach is repurposed to other computational design problems.

We used batched Bayesian Optimization (BO) as the optimizer for both θ and η . The BO algorithm outputs the best solution after 200 iterations with 3 evaluations per iteration, hence performing 600 N simulations at most. The manually specified feasible ranges of simulation parameters are listed in Table S9 and S10. The algorithm was repeated with five different random seeds and the solution that had the smallest loss value was chosen. To harness the high throughput of the simulator, we implemented the algorithm using the GPyOpt library (63) and batched all 3 N simulations in a BO iteration into a single simulator run.

Sample	E (GPa)			Err. (%)	T (MPa)		Err. (%)
	Exp.	Sim.			Exp.	Sim.	
TB	1.0094e-3	1.0091e-3	0.031	5.4335e-2	5.4271e-2	0.118	
VW	2.6613	2.6613	0.0002	7.4217e-2	7.4484e-2	0.360	
IF	0.5642	0.5642	0.003	0.8274	0.8193	0.984	

Table S2. Simulation accuracy after system identification on each base material separately. The accuracy is measured by relative errors (Err.) of simulation predictions (Sim.) from physical measurements (Exp.)

Round	N	Rel. error: E (%)				Rel. error: T (%)			
		Reference		After		Reference		After	
		Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
1	18	31.09	93.15	6.53	22.95	135.59	767.44	22.85	62.23
2	26	25.19	93.15	6.29	20.40	108.26	767.44	23.92	100.53
3	34	21.11	93.15	6.32	22.75	93.09	767.44	24.68	84.30
4	42	18.95	93.15	6.50	25.40	82.38	767.44	27.10	88.87

Table S3. Simulation accuracy after system identification in each outer loop round since Round 1. Simulation errors of the updated model after system identification are compared against the reference model (in Round 0). N denotes the number of microstructures in the system identification dataset. Different from Fig. 3F, the error in each round is computed using microstructures in the dataset only.

In Round 0, we fitted the simulation to the base materials using system identification, obtaining a reference simulation model. It achieved almost perfect accuracy, where Young’s modulus predictions are within 0.1% off from measurements and toughness prediction errors are below 1% (Table S2). However, after the system identification in each round, the updated model had much smaller errors in predicting the performance of microstructures than the reference model (Table S3). From a theoretical perspective, a possible explanation is that the localized mechanical performance of base materials might differ from bulk properties due to the microstructured placement. Thus, modeling the base materials in microstructures using physical measurements on bulk specimens could result in larger simulation errors.

For a complete reference, Table S11 and S12 list the physical measurements, simulation predictions, and relative errors of all microstructures discovered by our computational approach in every round of the outer loop.

3.3 NMO - the Inner Loop

Microstructures with optimal trade-offs between simulated Young’s modulus and toughness were explored following the system identification step. Here, a multi-objective optimization (MOO) problem was solved in the microstructure design space. As pixel-wise material assignments are discrete variables, the MOO problem can be tackled using genetic algorithms like NSGA-II (26). However, a naive NSGA-II implementation often suffers from low sample efficiency attributed to uninformative random mutations and crossovers. Alternatively, topology optimization (TO) methods (16,58,64,65) formulate the MOO problem as continuous optimization where the base material properties of finite elements are parameterized by continuous material assignment variables. While the continuous formulation allows for gradient derivation with respect to material assignments, TO algorithms are prone to local optima and do not easily scale up to multiple objectives. Particularly, if a full Pareto front is desired rather than a single Pareto-optimal solution, TO has to harness many starting points and weighted combinations of objectives to properly explore the design space. Another potential option is multi-objective Bayesian optimization (MOBO) which performs a more guided exploration of the design space (29, 30, 66) than NSGA-II. Most MOBO algorithms leverage Gaussian-process-based (GP-based) surrogate models that make fast and approximative performance predictions, thus greatly reducing the number of simulations required. However, the surrogate models are heavily constrained by the Gaussian prior and

can struggle in approximating sophisticated discrete mappings. Furthermore, since the computational cost of fitting Gaussian processes is cubic to the number of data points, the dataset is often limited to a few hundred in practice, which is far from adequate to model the complex relationship between microstructure patterns and toughness.

Inspired by MOBO, we developed a neural network-accelerated multi-objective optimization algorithm (NMO) that is specially tailored to the design space of microstructure patterns, harnessing the impressive capabilities of deep learning. The NMO workflow is illustrated in Fig. 2a. In the beginning, a set of 10 microstructures were randomly sampled from the design space and simulated to obtain their performance, constituting an initial dataset. In every iteration, the dataset was used to train the predictor, which acts as a surrogate model for the simulator. Ten predicted Pareto-optimal samples were then computed by solving a surrogate multi-objective optimization problem where simulation was replaced by the surrogate model. As the surrogate model is much cheaper to evaluate, we applied NSGA-II to solve the surrogate problem, which ran for 200 generations at a population size of 100. Then, the predicted best microstructures were verified in simulation before being added to the dataset for the next iteration. The algorithm lasted for 500 iterations as dictated by a pre-specified budget of 5,000 simulations. Due to its iterative nature, NMO transforms into the inner loop of our pipeline, where simulation data and proposed designs alternately flow between the simulator and the predictor.

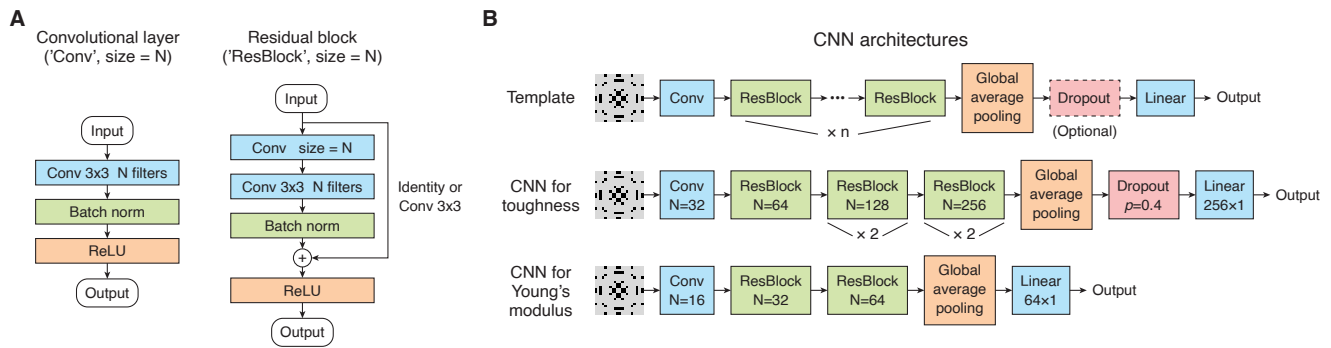


Fig. S10. CNN architectures used in the predictor. (A) Definitions of the architectural units of CNNs. (B) Prediction networks of Young's modulus and toughness constructed from an architecture template.

NMO has several major improvements over MOBO. Most importantly, the surrogate model is based on convolutional neural networks (CNN), which specializes in image-based reasoning, instead of Gaussian processes. We model toughness and Young's modulus using two separate residual networks (52). Both networks were constructed from the same architecture template demonstrated in Fig. S10. The template is a concatenation of one stem convolution layer, one or more residual blocks, a global average pooling layer, an optional dropout layer, and a fully connected layer. The network for toughness has a relatively larger capacity, with 5 residual blocks and a penultimate dropout layer for regularization. The network for Young's modulus is much smaller, with only 2 residual blocks. The networks were progressively trained on the dataset of microstructures discovered by the algorithm, where an Adam optimizer (67) was used at a learning rate of $1e-3$. The network for toughness underwent 300 training epochs whereas the network for Young's modulus went through 100 epochs. The maximum batch size was 2,048. To alleviate overfitting on small datasets, the networks were penalized by weight decay coefficients of $1e-2$ and $1e-6$, respectively. On a side note, the user may employ more complex network architectures, e.g., Vision Transformers (ViTs) (68), to accelerate the inner loop when applying our computational pipeline to alternative design problems. Our work uses CNNs as they typically perform better than ViTs in data-efficient scenarios and entail minimal expert knowledge of advanced neural network models for microstructure patterns.

The next improvement is a customized mutation operator in NSGA-II specially designed for microstructure patterns. Normally, NSGA-II strikes a balance between exploitation and exploration of the design space using a mutation operator and a crossover operator. In our case, we omitted the crossover step but augmented the mutation operator to enforce exploration. The customized mutation operator carried out one of the following options probabilistically.

- With $p_1 = 0.86$, flip a random pixel;
- With $p_2 = 0.11$, rewrite a rasterized line of pixels with a unified random value of 0 or 1, where the starting and ending positions are randomly chosen;
- With $p_3 = 0.03$, select a random pixel as the top-left corner, then rewrite a square region of pixels (the size of the square is randomly designated) with the complement of the top-left pixel.

After that, symmetry was restored by mirroring the changes across the entire pattern. Note that the mutated pattern might violate the structural constraints on the design space. Thus, potential violations were resolved through the procedure below

1. If the resulting ϕ falls below 0.5 after mutation, the pattern is flipped, and the four corner pixels are reassigned with the stiff material.
2. If other violations still exist, the operator continues applying new mutation attempts without undoing any changes until the violation is cleared.

Our implementation of the mutation operator takes into account that the toughness of a microstructure is directly affected by structural features in the pattern, such as pixels, beams, and chunks of the soft material. It balances between exploitation and exploration by modifying these structural features at multiple scales probabilistically.

The last improvement is a deduplication mechanism using hash tables. It prevents the algorithm from rediscovering previous samples in later iterations and allows the algorithm to escape local optima during exploration. Normally, single-pixel mutations create limited variation and easily result in duplicate patterns. With a hash table, however, the mutation operator automatically switches to larger perturbations once the hash table entries of neighboring patterns are filled. The exploration then continues in other regions of the design space. In our definition, the hash value of a microstructure pattern (viewed as a flattened array of binaries $x = (b_1, b_2, \dots, b_{400})$) was computed as

$$\text{hash}(x) = \left(\sum_{i=1}^{400} b_i k^{i-1} \right) \bmod m \quad (12)$$

where $k = 99,997$ and $m = 100,000,007$ are prime numbers. Each hash table had m entries. NMO had a global hash table, while each run of the surrogate problem solver used a separate table. After the modified NSGA-II algorithm finished, the predicted Pareto-optimal microstructures were ordered by non-dominated sorting and checked for hash collision inside the global hash table. The first 10 collision-free samples were validated by the simulator.

The exploration efficiency of NMO was evaluated using Pareto hypervolume growth within a budget of 5,000 simulations. Our modified NSGA-II algorithm was also evaluated. Compared with NMO, it still benefits from the custom mutation operator and the hash table albeit not having any surrogate model. For reference, we included a baseline NSGA-II implementation that performs single-pixel mutations and one-point crossovers at a rate of 0.1 and 0.8, respectively. Any crossover or mutation attempt that violated design constraints was reverted before trying the next one. Hence, the NSGA-II baseline is generally more conservative in exploration.

For TO, we chose bidirectional evolutionary structure optimization (BESO) (27,57) and solid isotropic material with penalization (SIMP) (58) as representatives since previous works have tackled tough composite design using both strategies (69, 70). Considering that they only optimize a single objective by definition, we extended them to finding the Pareto front by randomly sampling many weighted combinations of normalized Young's modulus and toughness as meta-objectives. To enforce exploration, we run BESO using multiple random initial solutions and set multiple random target volume fractions for SIMP. This way, both algorithms balance exploitation and exploration without leveraging any expert knowledge of Young's modulus or toughness, making a fair comparison with our method. Implementation-wise, we augmented the FEM simulator to support gradient calculation through auto-differentiation (71). We adopted the extended BESO method from (72) and solved SIMP using method of moving asymptotes (MMA). For MOBO algorithms, we evaluated DGEMO (29) and TSEMO (30) as baselines. DGEMO has state-of-the-art performance in several synthetic MOO benchmarks, while TSEMO is more commonly adopted.

Here, we additionally introduce the detailed optimization setting for TO-SIMP, where the MMA solver maximizes a weighted combination of normalized Young's modulus and toughness. An active constraint is imposed on the volume fraction of the rigid base material (ϕ). Below we provide the optimization statement in mathematical notations. Let $\rho \in [0, 1]^{20 \times 20}$ denote the density field of a microstructure unit. SIMP defines the element-wise Young's modulus E_e as

$$E_e(\rho_e) = \rho_e^p E_{\text{rigid}} + (1 - \rho_e^p) E_{\text{soft}} \quad (13)$$

where ρ_e refers to the density of the element e ; p denotes the SIMP penalization exponent; E_{rigid} and E_{soft} represent the Young's moduli of rigid and soft base materials. Let $\tilde{E}(\rho)$ be the normalized Young's modulus given by the Young's modulus simulator, $\tilde{T}(\rho)$ be the normalized toughness given by the toughness simulator. The following problem is solved using MMA:

$$\begin{aligned} \max_{\rho} \quad & w_1 \tilde{E}(\rho) + w_2 \tilde{T}(\rho) \\ \text{s.t.} \quad & \int_{\Omega} \rho \, d\Omega \leq \phi_{\text{max}} \\ & \rho \text{ obeys } p4m \text{ symmetry} \end{aligned} \quad (14)$$

Since the optimizer only converges to one microstructure design at a time, we repeat this process using randomly sampled w_1 , w_2 , and ϕ_{\max} to explore the design space in different directions. We use a constant SIMP penalization $p = 8$ in all cases without continuation.

Regarding the evaluation methodology, ten microstructures were simulated per iteration for all algorithms except TO which does not break down to generations. NMO and NSGA-II were run for 500 iterations. MOBO algorithms were capped by 200 iterations for an acceptable time consumption as fitting Gaussian processes became extremely slow with hundreds of samples. For TO algorithms, we paired random meta-objectives and initial designs (or target volume fractions) into optimization tasks and allowed the algorithm to execute as many optimization tasks as possible before exhausting the evaluation budget. All algorithms were repeated on the same five random seeds, which led to identical starting samples. The resulting curves of Pareto hypervolume growth were averaged.

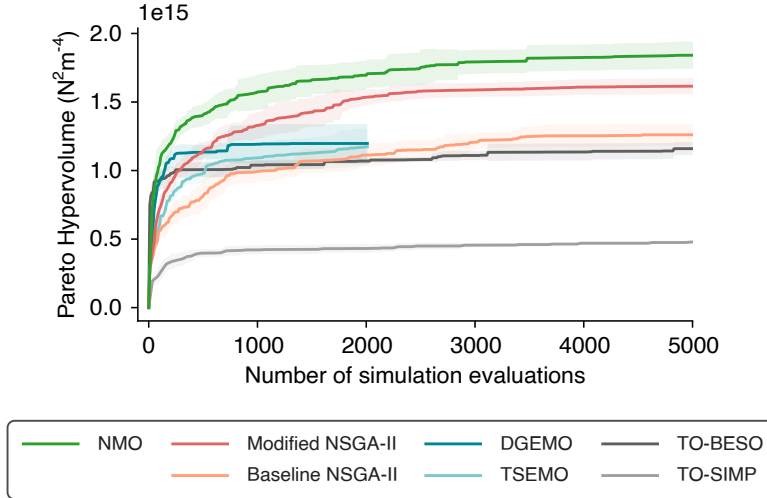


Fig. S11. Pareto hypervolume growth of the gamut using various optimization algorithms in Round 0. Candidates are NMO, NSGA-II, TO, and MOBO algorithms. Solid curves are averages. Colored regions indicate standard deviations.

Round	Pareto hypervolume		Improv. (%)	V_{target}	#Samples till V_{target}		Improv. (\times)
	NSGA-II	Ours			NSGA-II	Ours	
0	1.6154	1.8412	13.98	1.7202	20000	2200	9.10
1	1.6012	1.8835	17.63	1.8225	20000	4190	4.78
2	0.9956	1.0904	9.52	1.0776	19760	3240	6.10
3	1.0406	1.2016	15.47	1.0911	19950	1830	10.90
4	0.9457	1.0078	6.57	1.0078	17140	4990	3.44

Table S4. Comparison in exploration efficiency between NMO and our modified NSGA-II in each outer loop round. Exploration efficiency is measured by: (1) the discovered Pareto hypervolume within 5,000 simulations (unit: $10^{15} \text{N}^2 \text{m}^{-4}$); (2) the number of simulated samples before reaching a target Pareto hypervolume, denoted by V_{target} . V_{target} is equal to the Pareto hypervolume of NSGA-II at 20,000 simulations by default. If NMO does not reach that hypervolume within 5,000 simulations, we change V_{target} into NMO's hypervolume at 5,000 simulations instead.

Fig. S11 demonstrates the Pareto hypervolume growth of evaluated algorithms in Round 0. NMO secured victory with the fastest growth and the largest final hypervolume. The modified NSGA-II algorithm also pulled ahead of baseline NSGA-II considerably in hypervolume growth. In contrast, the TO variants failed to cover a decent Pareto hypervolume despite completing dozens of optimization tasks. They not only suffered from local optima but had to consume a majority of the budget on intermediate solutions along optimization trajectories, which crippled their sample efficiency. Both MOBO algorithms severely underperformed as well due to the aforementioned weaknesses of Gaussian processes. Furthermore, even when compared against our modified NSGA-II, NMO took up to an order of magnitude fewer simulations to reach the same Pareto hypervolume (Table S4). These results showcase NMO's evidently superior sample efficiency over other state-of-the-art structural optimization or generic MOO algorithms. In addition, we would like to note that NMO is compatible with any MOO

algorithm when solving the surrogate problem. It is recommended to choose an algorithm that adequately fits the design task for ideal exploration efficiency in practice.

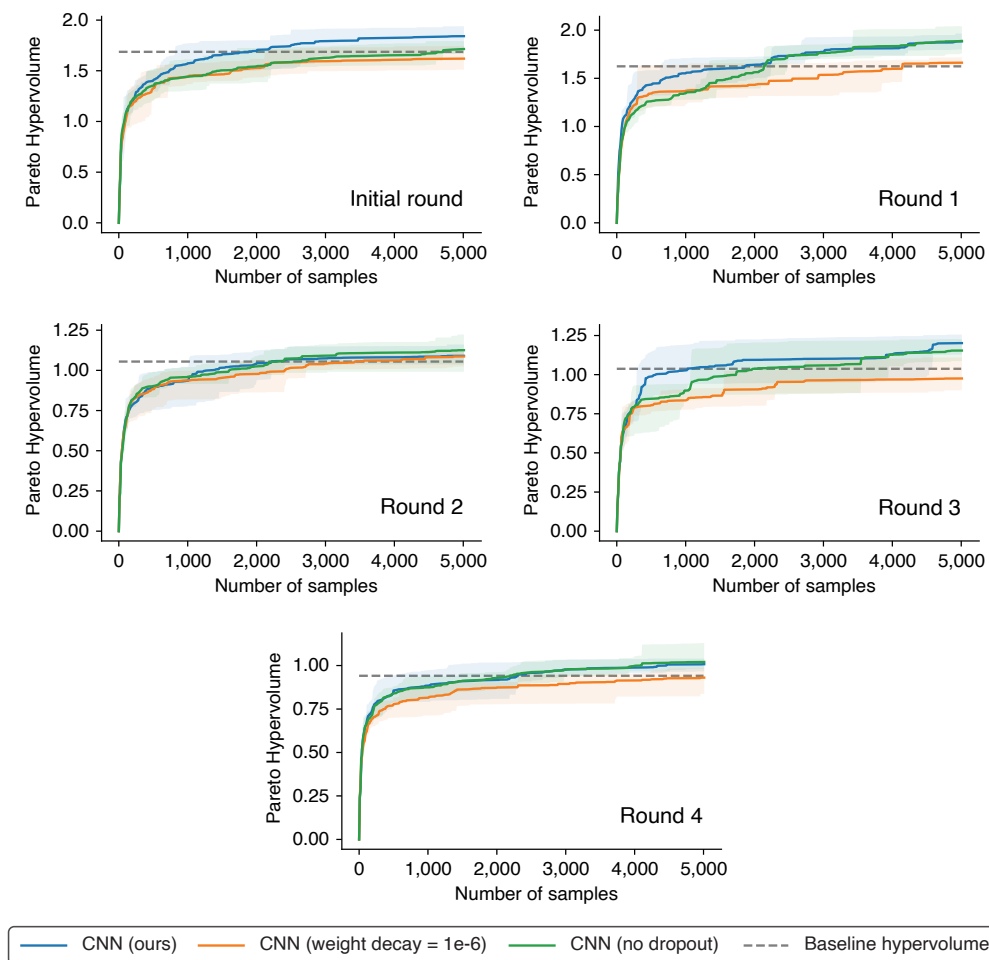


Fig. S12. Impact of network regularization on the Pareto hypervolume growth of NMO in all outer loop rounds. "Ours" means our hyperparameter setting including a large weight decay of $1e-2$ plus a penultimate dropout layer. The unit of the hypervolume is $10^{15} N^2 m^{-4}$.

By tuning the CNNs' hyperparameters, we noticed that network regularization plays an important role in accelerating hypervolume growth. In this experiment, our CNNs were compared against two less regularized variants, one using a smaller weight decay ($1e-6$) and the other excluding dropout before the fully-connected layer. As shown in Fig. S12, our networks outperformed less regularized versions and led to better hypervolume growth in most test cases. While the no-dropout variant had negligible wins in Rounds 2 and 4, the small-weight-decay variant kept trailing behind. Considering the relatively small amount of training data, a reasonable explanation is that regularization prevents the networks from overfitting and improves generalization to unseen samples. Thus, properly regularized networks can be better at discovering high-performing samples in simulation.

As an additional test, we evaluated the prediction accuracy of CNNs on training datasets in the last iteration of NMO (Fig. S13). The network for Young's modulus achieved very high accuracy with an average relative error of less than 2%. The network for toughness had larger errors in comparison, at 16.7% and 13.6%, respectively. Nonetheless, perfect accuracy is unnecessary in this case since the microstructures for simulation validation in the modified NSGA-II were selected by Pareto dominance rather than performance values. Therefore, it is sufficient to predict correct property rankings, which is indicated by the correlation coefficient (r) between predictions and the ground truth. Here, our network for toughness managed to get $r > 0.97$ and demonstrated much faster hypervolume growth than other evolutionary algorithms. As a side note, searching for the best network architecture or hyperparameter setting is beyond the scope of this work and thereby left for future research.

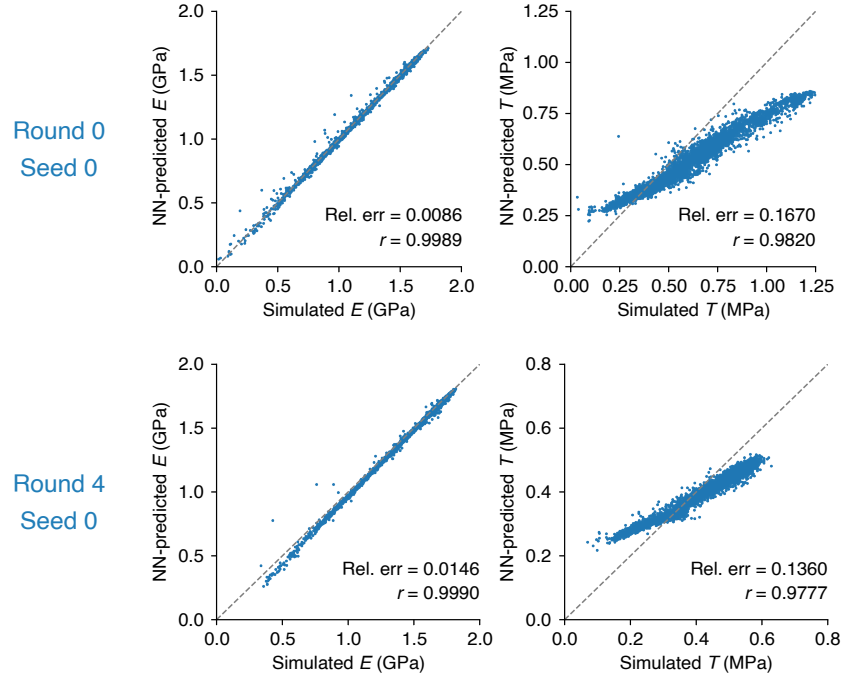


Fig. S13. CNN prediction accuracy in the last iteration of NMO. The examples shown are Round 0 and Round 4 with a random seed of 0. Simulation results are regarded as the ground truth. r means correlation coefficient.

3.4 Selection of Validation Samples

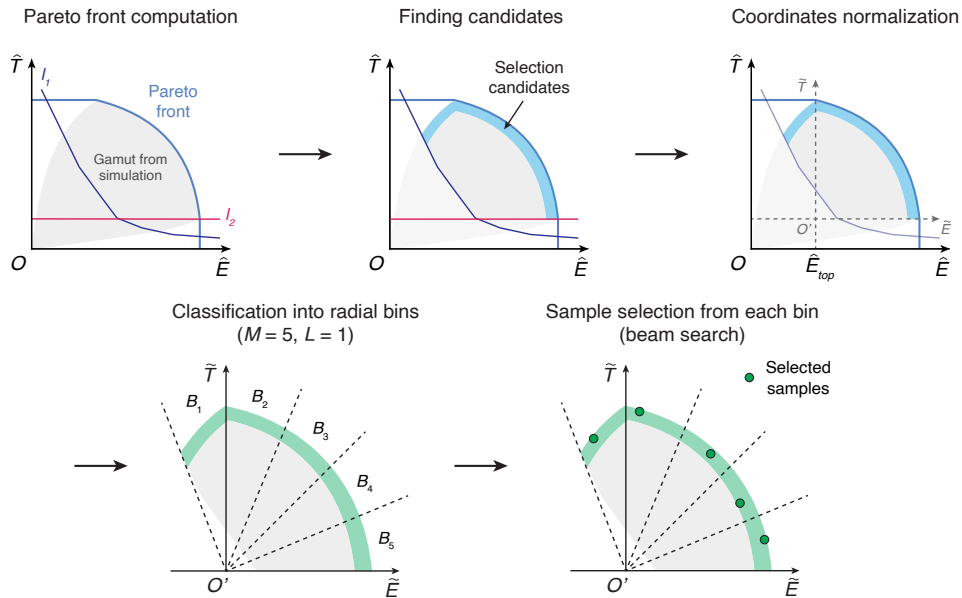


Fig. S14. The selection process of validation microstructures near the Pareto front. $M = 5$ and $L = 1$ are used in this illustration for simplicity.

The simulated gamut discovered by NMO typically contains a large pool of microstructures and a few samples are selected for physical validation due to a limited experiment budget. Overall, the selected samples should cover a wide performance

range and be distinguished from existing samples in appearance. To fulfill these requirements, we propose a two-stage selection approach visualized in Fig. S14.

The first stage extracted samples closest to the gamut boundary. This step can be done in various ways depending on the shape of the gamut. We employed a simple method where samples in the gamut were ranked in the order of Pareto dominance. By definition, Pareto-optimal samples of the entire gamut are in the 1st rank, and the i -th rank contains microstructures on the Pareto front of a partial gamut where those in higher ranks are removed. We regarded samples up to the 3rd rank as near the boundary. Meanwhile, the gamut was intersected by two benchmark lines l_1 and l_2 . l_1 is a polyline that connects the physical measurements of homogeneous composites. l_2 is a horizontal line going through the sample with the largest Young's modulus in simulation. Samples that were both near the boundary and above the benchmark lines were deemed as candidates for selection.

In the next stage, the candidate samples, $S \subset X$, were classified into two groups of radial bins B_1, B_2, \dots, B_L and $B_{L+1}, B_{L+2}, \dots, B_M$. M denotes the total number of samples to select, and L denotes the number of samples to select from the first group and that $0 \leq L < M$. To that end, we mapped the properties of samples in S to a normalized performance space in $[-1, 1] \times [0, 1]$. While both Young's modulus and toughness went through linear mappings, there are two circumstances for Young's modulus specifically. As shown in Fig. S14, the Young's modulus of the toughest sample \hat{E}_{top} is mapped to zero by default. However, the resulting image will not fit into $[-1, 1]$ if \hat{E}_{top} exceeds the midpoint of the Young's modulus range, hence the midpoint is mapped to zero in that case. Within the normalized performance space, the radial bins are outlined by $M + 1$ rays shooting from the origin, dubbed as r_1, r_2, \dots, r_{M+1} . r_{L+1} and r_{M+1} always align with the coordinate axes regardless of L . If $L > 0$, r_1 passes through the sample in S that has the largest orientation angle from the positive horizontal axis, φ_{max} . The i -th radial bin B_i refers to the area between r_i and r_{i+1} , and we let S_i denote the subset of S classified into B_i . B_1, B_2, \dots, B_L are equally sized and located in the second quadrant, while the other bins are equally sized and located in the first quadrant. The value of L is determined by φ_{max} using the rules below

$$L = \begin{cases} 0 & \varphi_{max} < \frac{\pi}{2} + \varphi_{th} \\ \left\lceil \left(1 - \frac{\pi}{2\varphi_{max}}\right) M \right\rceil & \varphi_{max} \geq \frac{\pi}{2} + \varphi_{th} \end{cases}, \quad (15)$$

where $\varphi_{th} = \pi/36$ (i.e., 5°). Practically, when φ_{max} is at least 95° , L is a minimum number such that the bins in the second quadrant are no larger than those in the first quadrant.

The following combinatorial optimization problem was subsequently solved to search for a group of microstructure patterns that maximally differentiate from physically measured ones. Let R denote the set of microstructures that have been discovered by the computational pipeline, and $\Omega = (x_1, x_2, \dots, x_M)$ (where $x_i \in S_i$ for $i = 1, 2, \dots, M$) denote a possible selection. The objective is to maximize the average L1 distance of each x_i to the rest of the samples in $R \cup \Omega$.

$$\Omega^* = \operatorname{argmax}_{\Omega} \frac{1}{M} \sum_{i=1}^M \min_{u \in R \cup \Omega \setminus \{x_i\}} \|x_i - u\|_1. \quad (16)$$

As the problem is likely intractable by enumeration when S contains over 100 samples, we used a beam search algorithm to find an approximate solution. The beam size was set to 100 to preserve the quality of the solution at an acceptable computational cost.

3.5 The Outer Loop

The outer loop was held between the mechanical tester and the simulator as follows. In Round 0, the mechanical tester conducted physical measurements on the base materials (the rigid, soft, plus the interface material). Our system then matched simulation and experiment using a system identification. Each material was involved in a separate system identification process that only optimized simulation parameters pertaining to that material, including Young's modulus, Poisson's ratio, energy density threshold, and damping coefficient. This yielded three optimized values of the damping coefficient where the median was chosen. Using the calibrated simulator, our system ran the inner loop to find microstructures with the optimal trade-offs between Young's modulus and toughness in simulation. As previously noted, we used the modified NSGA-II algorithm to compute the simulation Pareto front and left the NMO algorithm to a separate comparison with other optimization algorithms. The NSGA-II was run in five different random seeds under a simulation budget of 2×10^4 samples. The resulting gamuts were merged into one gamut with approximately 10^5 samples, after which 8 validation microstructures were selected near the Pareto front and transferred back to the mechanical tester for validation.

Starting from Round 1, our system maintained a dataset of physically measured microstructures, starting from 18 initial designs. Besides the proposed designs from Round 0, the dataset also had 10 random microstructures drawn from stratified sampling according to ϕ (Table S5). Microstructure patterns in each interval of ϕ were generated by randomly specifying a ϕ value and filling a zero matrix with a corresponding number of 1's. These random microstructures help regularize system



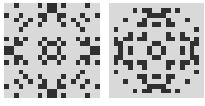
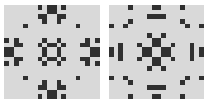
ϕ interval	Patterns
$0.5 \leq \phi < 0.6$	
$0.6 \leq \phi < 0.7$	
$0.7 \leq \phi < 0.8$	
$0.8 \leq \phi \leq 0.85$	

Table S5. Random microstructure designs in the system identification dataset categorized by ϕ .

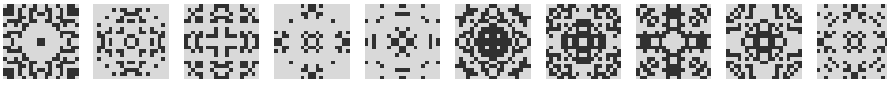
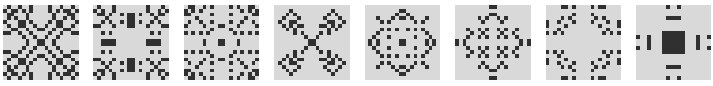

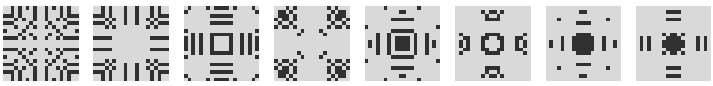
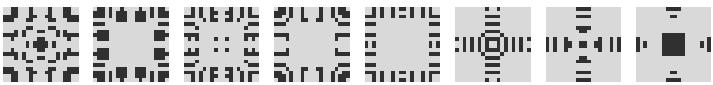

Sample No.	Source	Microstructure patterns
1 - 10	Random sampling	
11 - 18	Round 0	
19 - 26	Round 1	
27 - 34	Round 2	
35 - 42	Round 3	
43 - 50	Round 4	

Fig. S15. Microstructures discovered by the computational pipeline. The first row comprises 10 random patterns in the system identification dataset. The following rows are patterns proposed in outer loop rounds.

identification, especially in early rounds of the outer loop when the dataset is small. As shown in Fig. 3b, the mechanical tester provided the simulator with physical measurements of all microstructures in the dataset to improve its accuracy through system identification. Different from Round 0, system identification was conducted using experiment data from microstructures exclusively. After that, the simulator computed its gamut and proposed 8 Pareto-optimal designs back to the mechanical tester for validation. The proposed designs were then added to the microstructure dataset. Thus, the system identification dataset had $8n + 18$ microstructures at the end of Round n ($n \geq 0$). The outer loop stopped at $n = 4$.

Microstructured composites discovered during the outer loop (Fig. S15) were compared against homogeneous composites. Among the microstructures that showed better performance trade-offs than homogeneous counterparts, we picked five best samples near the Pareto front and with more than $1.5\times$ improvement in toughness from the benchmark line (Table S6). Sample 2 has the maximum improvement in both Young's modulus and toughness at $1.61\times$ and $2.76\times$, respectively. Samples 1 and 3 also show more than $2\times$ improvement in toughness. These results sufficiently prove that computationally designed microstructured composites present much better trade-offs between Young's modulus and toughness than homogeneous composites. We also

Sample	ϕ	Properties		Improvement (homogeneous)		Improv. (base)
		E (GPa)	T (MPa)	E (\times)	T (\times)	T (\times)
1	0.79	1.402	0.491	1.35	2.00	7.01
2	0.85	1.810	0.473	1.61	2.76	6.75
3	0.85	1.764	0.458	1.54	2.48	6.53
4	0.84	1.790	0.355	1.27	1.76	5.07
5	0.85	1.807	0.304	1.28	1.59	4.33

Table S6. Performance of five best microstructures discovered by our approach. Their improvements are calculated relative to homogeneous composites and base materials. For homogeneous composites, the improvement in one property is calculated assuming that the other property is identical. The largest improvement factors are marked in bold.

calculated the improvement of the best samples over base materials in toughness (VW+ has a higher toughness of 0.070 MPa than TB+), whereas Sample 1 has $7.01\times$ higher toughness than VW+.

Note S4 Analysis on Pareto-Optimal Microstructures

4.1 Discovery of Microstructure Families

Microstructure families were discovered from microstructures with near-optimal trade-offs between Young’s modulus and toughness after the nested-loop pipeline. Here, we grouped 11 near Pareto-optimal microstructures into 4 families by patterns and mechanical performance (Fig. 4b). They were referred to as seed microstructures and leveraged to generate pattern variations in each family. Then, we validated the discovered families by confirming that the generated microstructures had similar performance to seed microstructures. The procedure is detailed as follows.

Family	N	Rel. error: E (%)				Rel. error: T (%)			
		R4		FS		R4		FS	
		Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
1	3	10.77	19.98	6.25	12.56	40.17	48.25	4.55	8.28
2	4	2.99	5.45	1.22	3.45	41.61	56.81	11.32	21.29
3	2	7.69	9.64	2.13	4.22	47.42	62.99	2.69	5.35
4	2	3.96	5.76	2.16	4.29	33.03	43.64	0.45	0.73

Table S7. Prediction errors of the simulator after system identification in each microstructure family. The family-specific simulators (FS) are compared with the simulator in Round 4 (R4) using the performance of seed microstructures in each family. N denotes the number of samples. Better values are marked in bold. Physical measurements and corresponding predictions are provided in Table S14 and S15.

To start with, we ran system identification in each family to obtain a family-specific simulator. The system identification involved all near Pareto-optimal samples but assigned them with different weights. Samples in the family had a constant weight of 1, while others were weighted by 0.1 for regularization purposes. Such family-specific system identification leads to a better fitting quality within the family, as illustrated in Table S7. Since only seed microstructures and other Pareto-optimal microstructures are used, there is a closer match between simulation predictions and physical measurements than the simulator in Round 4. The contrast in prediction error is particularly obvious in toughness predictions where the family-specific (FS) simulator has an order of magnitude smaller relative errors in three families.

To generate variations in microstructure patterns, we performed dense evolutionary sampling around the seed microstructures. Similar to NSGA-II, the sampling algorithm began with the seed microstructures in the family and generated other microstructures iteratively using our modified mutation operator. However, there was no tournament selection of parents or competition among the population, and any mutation attempt was simply rejected if it led to a violation of design constraints. In every iteration, the algorithm kept generating mutated microstructures until 10 non-existing ones were found, after which they were evaluated using the family-specific simulator. The mutated microstructures were restricted within a maximum of 16-pixel difference in pattern (which is 4% of all pixels) from the closest seed microstructure. The hash table in Section 3.3 was adopted for deduplication. As the hash table entries were rapidly filled, it became increasingly difficult to generate new microstructures over time. Thus, the algorithm was limited by a maximum number of 3×10^5 proposed microstructures per

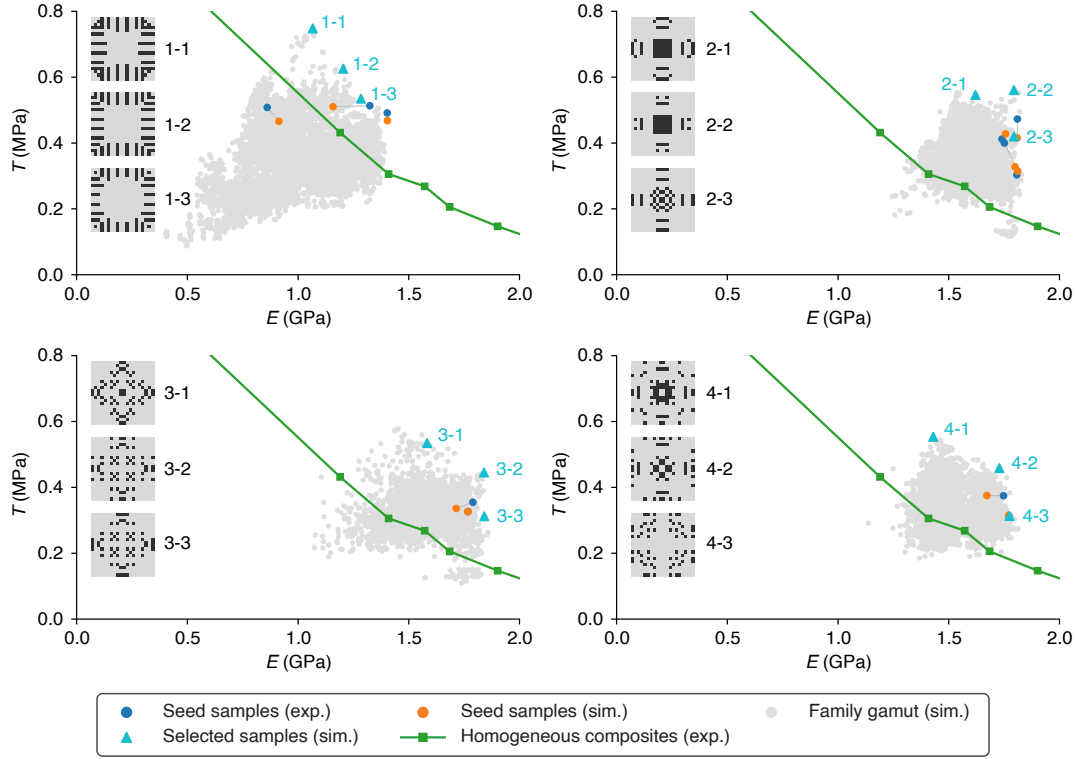


Fig. S16. Gamut of each microstructure family after dense sampling along with selected samples near the Pareto front for validation. Each family contains around 7,000 to 9,000 samples. 'Exp.' is short for physical measurements and 'sim.' for simulation predictions. The measurements and predictions of a seed microstructure are connected to indicate error.

iteration and terminated if it failed to complete an iteration within the limit. Our algorithm generated local gamuts for all four families as shown in Fig. S16.

Finally, we conducted physical measurements on three microstructures in the local gamut with near-Pareto-optimal predicted performance. The samples were selected using the algorithm in Section 3.4 (Fig. S16). Their experiment results revealed comparable performance to the seed microstructures (Table S13), hence verifying that the performance similarity among microstructures in the same family was preserved after dense sampling.

4.2 Dimensionality Reduction in Subfamilies

For more in-depth analysis, each microstructure family was divided into subfamilies according to similarity with seed microstructures, where a subfamily consists of a seed microstructure and others within a 16-pixel radius as specified in dense sampling. We ran Isomap to reduce 400D microstructure patterns into 2D embedding spaces (with a neighborhood size of 10) in each subfamily, shown in Fig. S17. Microstructure properties are color-coded in the figures. A higher intensity of the red channel indicates better toughness, whereas the blue channel represents Young's modulus. Our dense sampling algorithm was able to create a diversity of variants from the seed microstructures. New shape features can be seen in these variants, such as individual pixels, beams, arcs, and small chunks of soft material.

Regarding the distance metric for Isomap, we used the earth mover's distance (EMD) rather than the Euclidean distance. EMD, or the 1st Wasserstein distance (53), is a metric between probability distributions originally defined in optimal transportation theory (55). It intuitively measures the minimal work (defined as mass times distance) required to redistribute a pile of earth into another. Compared with the Euclidean distance, EMD better reflects the shape difference between microstructure patterns by taking into account both pixel values and locations. Since the original definition of EMD is not directly compatible with microstructure patterns, we converted the patterns into quarter-sized normalized histograms that represent discrete probability distributions. The conversion rules, illustrated in Fig. S18, treated pixels of the soft material as shape features and assigned non-zero values to the corresponding cells of the output histogram. Conversely, the other cells associated with the rigid material were set to zero. Therefore, the EMD between two microstructures x_a and x_b is calculated using the 1st Wasserstein distance

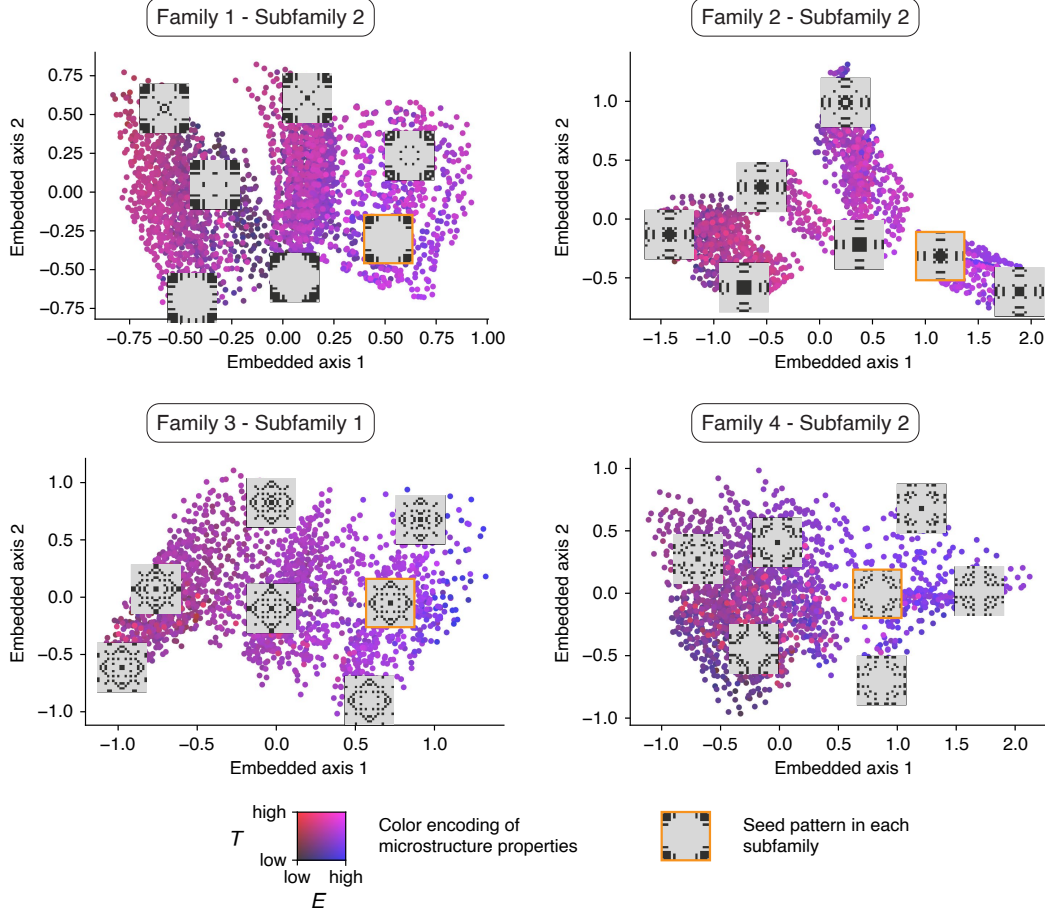


Fig. S17. Isomap embedding spaces of four subfamilies annotated with example patterns. Each subfamily contains around 2,000 to 3,000 microstructures and a few of them are visualized. Mechanical performance is encoded in colors (E : the blue channel; T : the red channel). Seed patterns are highlighted by orange boxes.

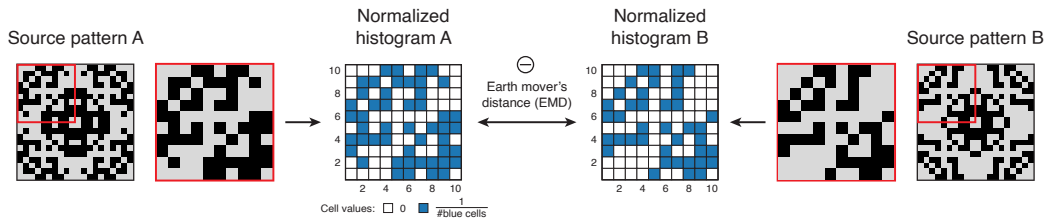


Fig. S18. Calculation of the earth mover's distance (EMD) between two microstructure patterns.

between their normalized histograms h_a and h_b ($h_a, h_b \in [0, 1]^{100}$).

$$\begin{aligned} \text{EMD}(x_a, x_b) &= W_1(h_a, h_b) \\ &= \min_{\pi \in \Pi(h_a, h_b)} \pi \otimes D, \end{aligned} \quad (17)$$

where $\Pi(h_a, h_b)$ is the collection of transportation plans, π , from h_a to h_b , given by

$$\Pi(h_a, h_b) = \{ \pi \in \mathbb{R}^{100 \times 100} \mid \pi \geq 0, \pi \mathbf{1} = h_a, \text{ and } \pi^T \mathbf{1} = h_b \}. \quad (18)$$

In the earth-moving analogy, π tabulates the amount of earth to transfer from one cell to another. D is the pairwise Euclidean distance matrix of histogram cells. $A \otimes B = \langle A, B \rangle_F = \text{tr}(A^T B)$ is the Frobenius inner product of two identically shaped matrices.

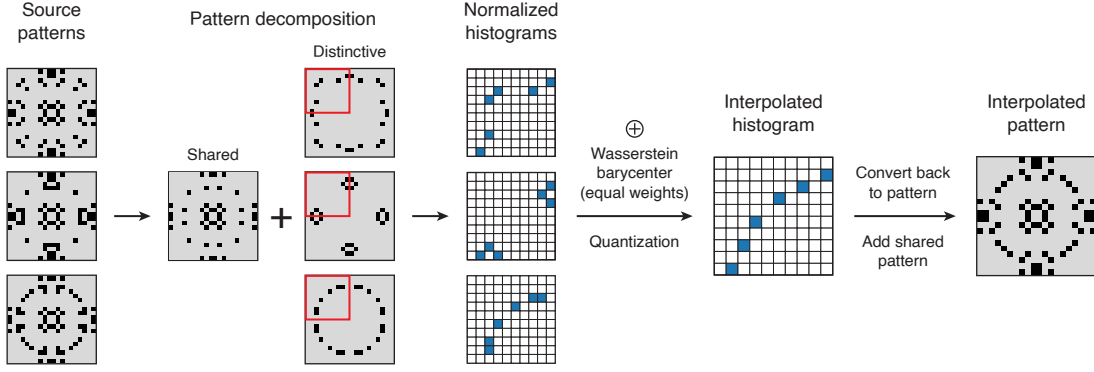


Fig. S19. Interpolation of neighboring microstructure patterns in an Isomap embedding space based on Wasserstein barycenters.

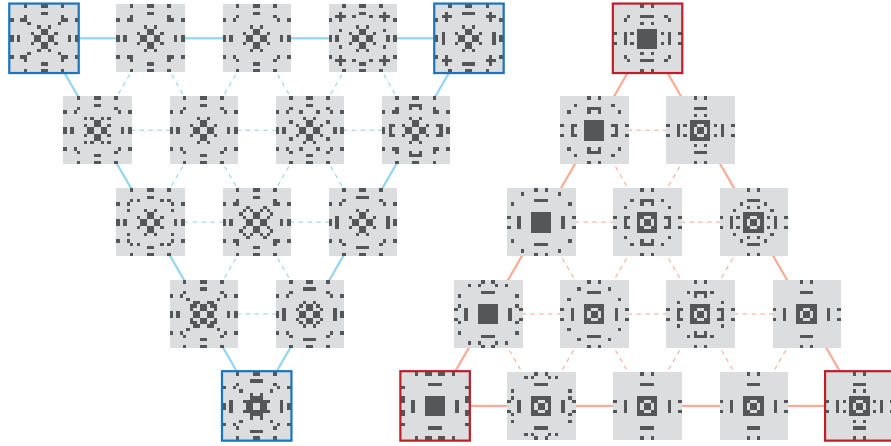


Fig. S20. Examples of interpolation between three microstructures in the same subfamily.

The Isomap embedding space of a subfamily can be further refined by samples not covered in dense sampling. To generate these samples, we built an interpolation model for K neighboring microstructure patterns x_1, x_2, \dots, x_K in the embedding space based on Wasserstein barycenters (54), which is widely used in shape interpolation (56). The interpolation algorithm is illustrated in Fig. S19. Each source pattern was decomposed into a common part containing shared shape features and an exclusive part containing distinctive features. The exclusive parts were converted into quarter-sized normalized histograms denoted by h_1, h_2, \dots, h_K . Our model interpolated among the histograms by computing their Wasserstein barycenter \bar{h} under the weight assignment w_1, w_2, \dots, w_K , defined as

$$\bar{h} = \operatorname{argmin}_h \sum_{i=1}^K w_i W_2(h_i, h). \quad (19)$$

$W_2(\cdot, \cdot)$ refers to the 2nd Wasserstein distance,

$$W_2(h_a, h_b) = \min_{\pi \in \Pi(h_a, h_b)} \pi \otimes D_2, \quad (20)$$

where $D_2 = D \circ D$ is the pairwise squared distance matrix (\circ refers to the Hadamard product). The minimization problem was tackled by solving the following equivalent linear programming problem (54) for the optimal transportation plans π_i^* from each

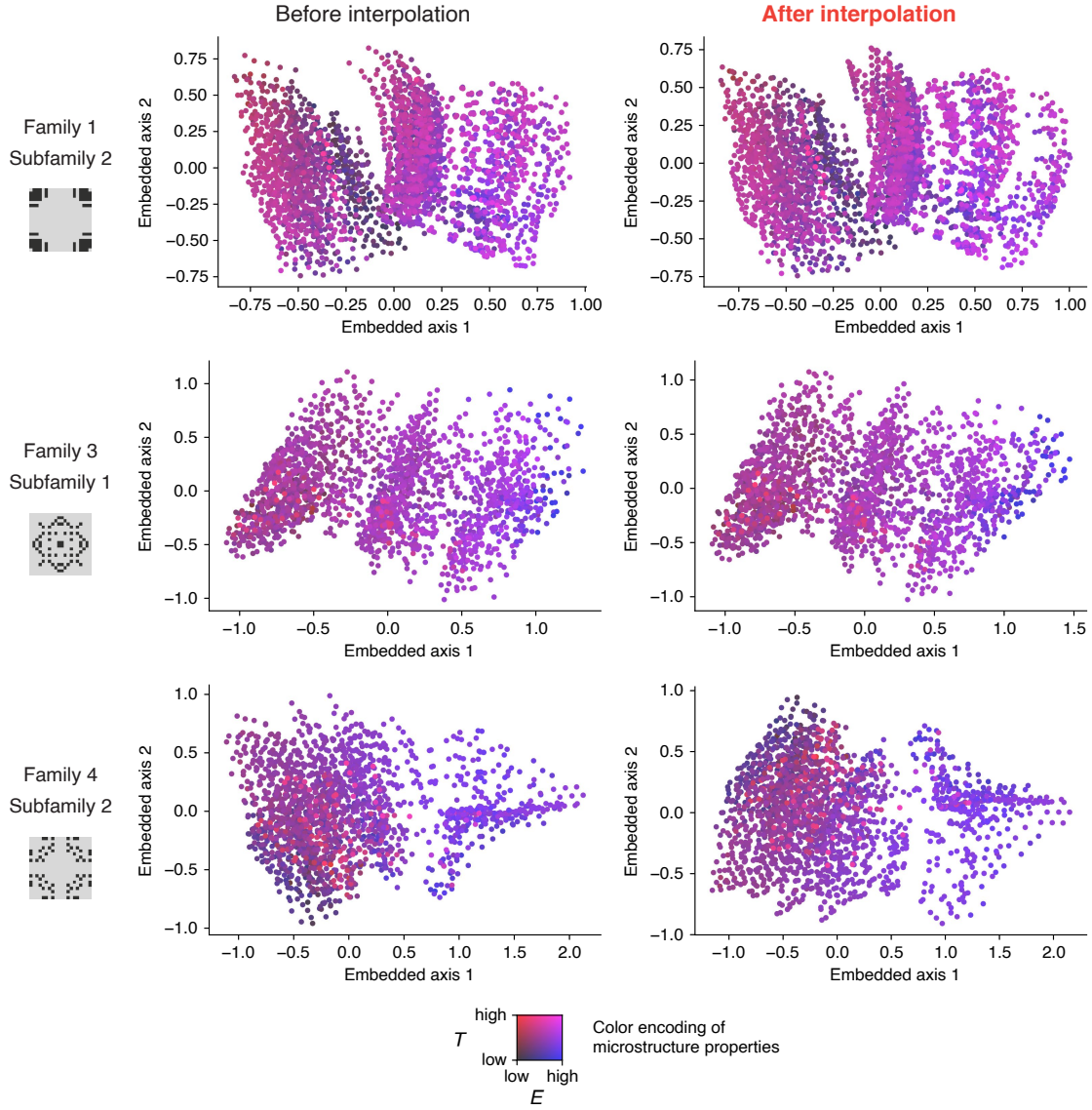


Fig. S21. Isomap embedding spaces of three subfamilies before and after refinement using interpolation. Microstructure properties are encoded in colors.

h_i to \bar{h} ($i = 1, 2, \dots, K$).

$$\begin{aligned}
 & \min_{\{\pi_i\}} w_i (\pi_i \otimes D_2) \\
 & s.t. \quad \pi_i \geq 0, \quad \forall i = 1, 2, \dots, K \\
 & \quad \pi_i \mathbf{1} = h_i, \quad \forall i = 1, 2, \dots, K \\
 & \quad \pi_i^T \mathbf{1} = \pi_i^T \mathbf{1}, \quad \forall i = 2, 3, \dots, K
 \end{aligned} \tag{21}$$

After that, \bar{h} was obtained by $\bar{h} = (\pi_i^*)^T \mathbf{1}$. To convert the interpolated histogram back to a microstructure pattern, \bar{h} was quantized into a binary matrix that indicates the material assignments of the upper-left quarter of the resulting pattern. Let c_i be the unified value of all non-zero cells in h_i and $c = (\sum_{i=1}^K c_i) / K$. A cell in \bar{h} translated into the soft material if no less than c , and the rigid material otherwise. Examples of microstructure pattern interpolation are shown in Fig. S20.

In practice, a Delaunay triangulation was computed in the 2D embedding space to determine neighboring microstructures, namely $K = 3$. The neighborhoods whose longest edge was between 4% and 40% of the diameter of the family (i.e., the distance between the two furthest points in the embedding space) were marked as candidates. New microstructures were then

Family	1			2				3		4		Total
Subfamily	1	2	3	1	2	3	4	1	2	1	2	
Before	3,251	2,959	3,313	1,862	1,861	1,917	1,859	1,860	1,862	1,997	1,915	24,656
After	3,328	3,164	3,371	1,939	1,904	2,054	1,903	1,988	1,964	2,112	2,044	25,771
Inc. (%)	2.37	6.97	1.75	4.14	2.31	7.15	2.37	6.88	5.48	5.76	6.74	4.52

Table S8. Increases in the size of subfamilies from interpolation.

generated from uniform grid interpolation in the candidate neighborhoods and added to the subfamily if they were within a 16-pixel radius from the seed microstructure and did not violate any structural constraint. Finally, the refined embedding space was recomputed using Isomap. Our interpolation algorithm generated about 5% more samples in each subfamily on average (Table S8). In Fig. S21, we make a side-by-side comparison between the Isomap embedding spaces before and after interpolation.

4.3 Toughening Mechanisms

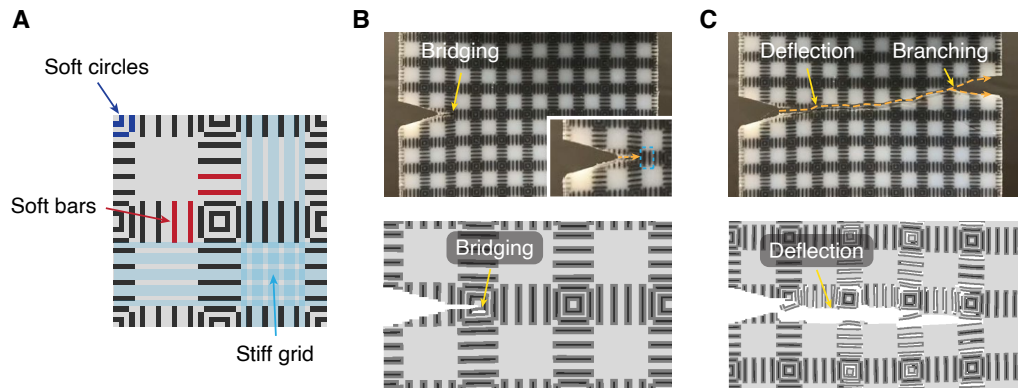


Fig. S22. Major shape features (A) and toughening mechanisms (B, C) of microstructures in Family 1.

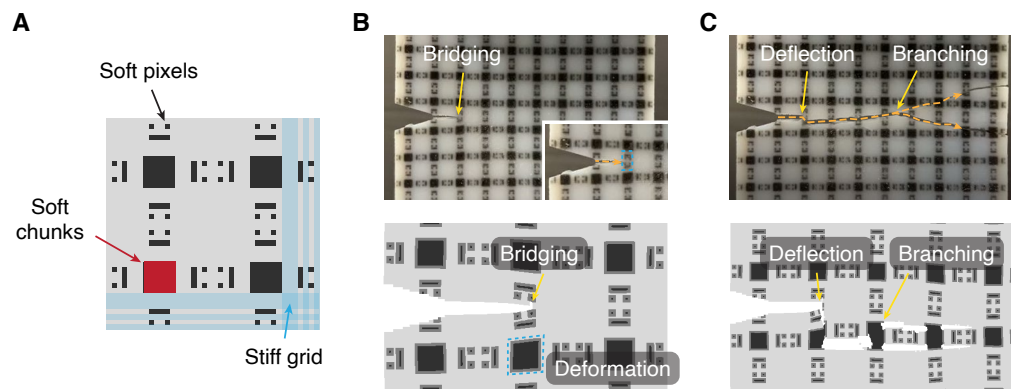


Fig. S23. Major shape features (A) and toughening mechanisms (B, C) of microstructures in Family 2.

Intrinsic toughening mechanisms of each microstructure family were analyzed from video recordings of validation samples during mechanical testing and simulation. We summarize the most representative mechanisms of each family and associate them with shape features in microstructure patterns as follows.

Family 1 maintains a relatively high fraction of soft pixels in the shape of bars or chunks as exemplified in Fig. S22A. This family is located on the left side of the performance gamut where Young's modulus is relatively small. To maintain proper

stiffness, the continuous areas of rigid material form horizontal and vertical grids (stiff grids in Fig. S22A). A distinctive feature appearing in this family is parallel bars of soft material. Their benefits are twofold. When perpendicular to the pulling direction, they introduce additional stress resistance through elastic deformation. Alternatively, when parallel to the pulling direction, or directly facing crack propagation, they effectively absorb energy and slow down the propagation. Bridging and deflection effects are the most prominent toughening mechanisms (Fig. S22B and S22C). A zig-zag crack path is observed during fracture which results in high toughness. The propagation energy accumulates under the trapping effect of soft material and dissipates via branched cracks (40).

Different from Family 1, Family 2 concentrates soft pixels in a cross-like band in the center of the unit (Fig. S23A). Microstructures withstand a larger strain via the elastic deformation of soft material chunks, as indicated by the red square in Fig. S23B. It is noteworthy that such deformation is accurately reflected in simulation. Stiff grids are also present as a means of improving Young's modulus. Bridging, deflection, and branching (Fig. S23B and S23C) are essential anti-cracking mechanisms in the family.

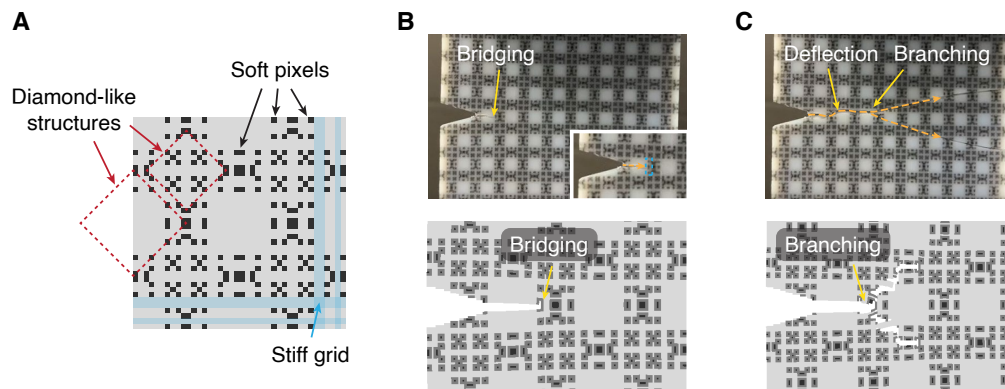


Fig. S24. Major shape features (A) and toughening mechanisms (B, C) of microstructures in Family 3.

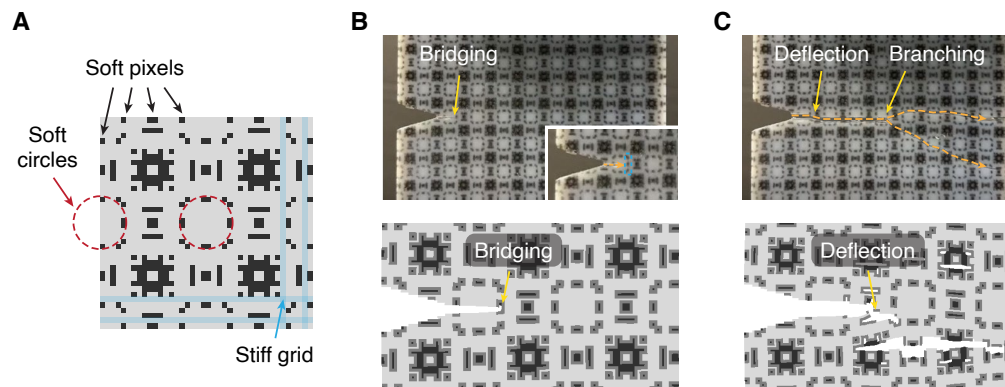


Fig. S25. Major shape features (A) and toughening mechanisms (B, C) of microstructures in Family 4.

While soft pixels are more concentrated in the two families above, they are mostly dispersed in Family 3. Discrete soft pixels form a quarter of a diamond-like structure at corners and another diamond structure in the center (Fig. S24A). Meanwhile, some soft pixels are located at the edges and ahead of the notch tip. These pixels are typically on the path of crack propagation and act as an energy dissipation zone that delays crack propagation. Since the interface is stronger than the soft material, the soft pixels work as defects that guide crack propagation. Such a deflection effect can increase the crack length dramatically and dissipate much more energy. Additional soft pixels introduced during dense sampling not only enrich the diversity of the family but also increase the overall stress resistance.

As for Family 4, the discrete soft pixels around corners roughly form a quarter of a circular outline (Fig. S25A). Similar to Family 3, some soft pixels are placed on the edges, which retards crack propagation by dampening the cracking velocity and dissipating energy through elastic deformation. Basic features like stiff grids are preserved to maintain Young's modulus. The

same anti-cracking mechanisms, such as bridging, deflection, and branching, are observed in this family (Fig. S25B and S25C). As Family 3 and 4 bear some resemblance in structure, we observed that they show comparable mechanical performance. Due to the dominance of rigid material at $0.8 \leq \phi \leq 0.85$, microstructures in both families tend to have higher Young's moduli but break at relatively smaller strains.

Note S5 Case Study: Stiffness-Porosity Trade-off

With carefully designed geometries, porous structures enable strong materials with light weights. In many engineering applications, it is desirable to design microstructures that are both stiff (large Young's modulus) and lightweight (high porosity). However, porosity and stiffness are often contradictory since materials with higher porosity are generally weaker due to less material supporting the external force. Due to its crucial impact on engineering, the stiffness-porosity trade-off has been extensively studied in the field of solid mechanics and topology optimization. In this section, we directly apply our optimization pipeline to identify the Pareto front between stiffness and porosity for porous microstructure designs. We demonstrate that our pipeline not only remains effective in Pareto front discovery but also has adequate scalability to efficiently explore even larger combinatorial design spaces.

5.1 Problem setting

Design space. We consider single-material, periodic porous structures where each unit is characterized by a regular grid of 182×104 elements (Fig. S26A). An element can be either void or filled with solid material. This constitutes a much larger design space than the 20×20 patterns we considered for the stiffness-toughness trade-off. However, a valid microstructure design must contain a single-connected solid phase to exhibit non-zero stiffness. Therefore, we adopt the topology generation algorithm introduced in (73) to create valid random designs. Meanwhile, we focus on a porosity range from 0.1 to 0.75 to exclude impractical designs with excessive or insufficient material. To further ensure that the generated designs are isotropic, we require all microstructure patterns to have $p6m$ symmetry.

Fabrication and measurement. Test samples are fabricated on acrylic plates with a thickness of 1.6 mm (McMaster-Carr 8560K172). Microstructure patterns are carved onto the acrylic plate via laser cutting (Universal Laser Systems VLS6.75). More specifically, the laser is programmed to cut along the interface between material and void to produce the designed porous structure (Fig. S26B). The dimensions of each unit cell are 15 mm in width and 26 mm in height, resulting in a pixel size of 0.14 mm. Since the resolution of the laser cutter is below 0.05 mm, this fabrication method is sufficiently accurate. Each sample consists of 3×5 unit cells, with a total size of 75 mm in width and 78 in height. In addition, two rectangular gripping areas with a height of 50 mm each are concatenated to the microstructure.

We measure the effective Young's modulus of a microstructure design on a tensile machine (Instron with 500N load cell, see Fig. S27A). The test sample is fixed onto the Instron using two pneumatic grippers (the pressure is 100 pounds per square inch). During the tensile test, the Instron stretches each sample to 0.1 mm at a rate of 1 mm/minute. The effective Young's modulus of the sample is then calculated based on the slope of the measured force-displacement curve and sample size. To prove that the boundary effect is negligible for a sample with 3×5 unit cells, we measured the effective Young's modulus using samples of various sizes: 2×4 , 2.5×4 , and 3×5 units. According to Fig. S27B, Young's modulus converges at our designated sample size, indicating that the 3×5 size already suffices to minimize the boundary effect. Moreover, we measured several Young's moduli of the same microstructure along multiple directions to verify its isotropy. This is realized by fabricating samples with tilted microstructures. As shown in Fig. S27C, the microstructure design exhibits almost identical Young's moduli in different directions. In all experiments, we measure each test sample 4 times and report the average Young's modulus and standard deviation. These measurements are guaranteed to be consistent since the laser cutter has a sufficiently high resolution and the acrylic plates are relatively thin (3 mm). Nonetheless, we made an extra effort to fabricate three samples for a fixed microstructure design, and the measured Young's moduli differ by less than 1%. Therefore, to save time and energy, we only fabricate one sample per microstructure design in this case study.

Simulation. We developed an FEM simulator to calculate the homogenized Young's modulus of a porous microstructure specifically for this study. The FEM grid has identical dimensions to a microstructure unit (i.e., 180×104 elements). Since stiffness calculation is a linear problem, we use linear elasticity to model the base material. The model is characterized by two exposed parameters: Young's modulus E_{base} and Poisson's ratio ν_{base} of the base material. Each finite element is parameterized by material density using a continuous variable between 0 and 1. A density equal to 1 represents the base material whereas a zero density corresponds to void. Based on the homogenization theory, we computed the homogenized stiffness tensor of the microstructure using periodic boundary conditions, from which we derived homogenized Young's moduli in multiple directions. The minimal Young's modulus was chosen as the objective for structural optimization to ensure material isotropy. Moreover, we implemented gradient calculation and SIMP in the simulator to compare our pipeline against existing topology optimization algorithms that utilize sensitivity information.

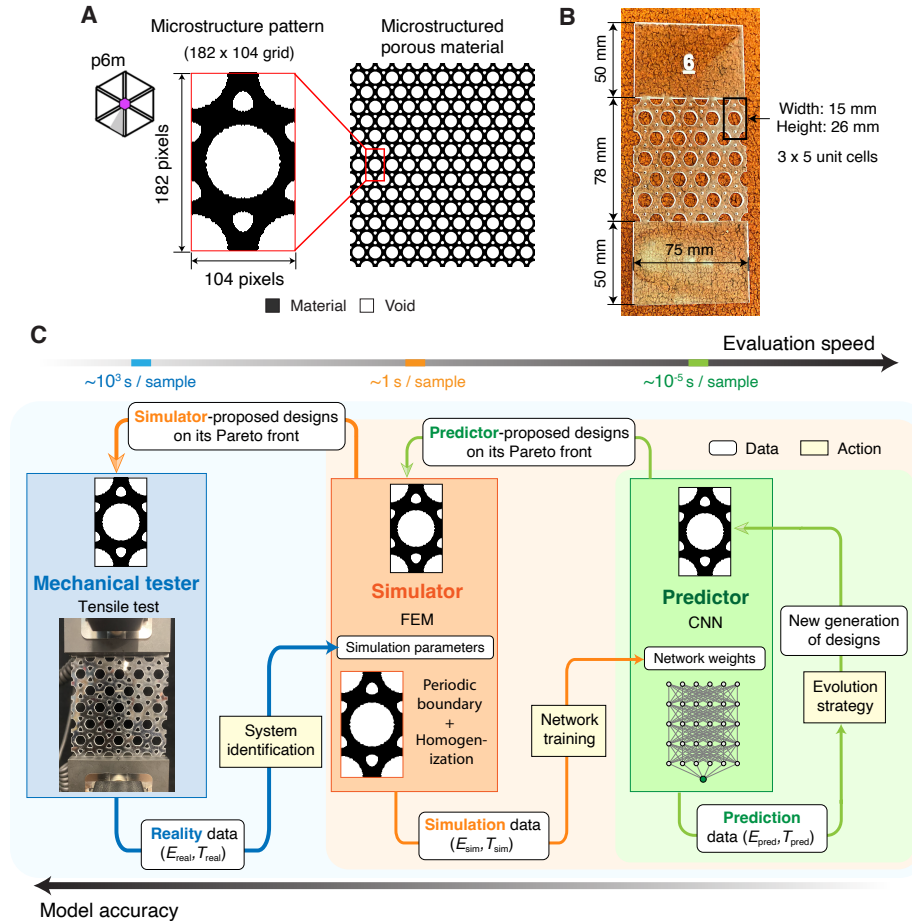


Fig. S26. Application of our approach to discover porous microstructures with optimal stiffness-porosity trade-offs. (A) The porous microstructures consist of periodic unit cells defined on a 182×104 regular grid, where each element represents either material or void. We enforce p6m symmetry on microstructure designs for isotropy. (B) Sample photograph of a manufactured porous microstructure. The sample is fabricated by laser-cutting a clear acrylic plate. (C) Workflow of our pipeline to discover the stiffness-porosity Pareto front.

5.2 Results

In the initial round, the exposed parameters of the simulator, E_{base} and ν_{base} , were determined by the Young's modulus and Poisson's ratio of an acrylic plate. Since the vendor only provides a flexural modulus of 2.76 GPa and leaves out the value of its Poisson's ratio, we made an initial guess of $E_{base} = 2.76$ GPa and $\nu_{base} = 0.33$.

We ran the inner loop of our pipeline (NMO) to find the Pareto front of the simulator, from which we selected 10 designs for experimental verification (see the orange dashed line and markers in Fig. S28). As expected, there is a noticeable gap between the experimental Pareto front (blue) and the simulation Pareto front (orange), since the exact properties of the base material are unknown. At the beginning of Round 1, our pipeline conducted system identification on discovered designs by adjusting the values of E_{base} and ν_{base} . Next, we ran the inner loop (NMO) again and identified 10 additional designs on the simulation Pareto front. After physically measuring their Young's moduli, we noticed that the measurements were almost identical to simulation predictions (see Round 1 in Fig. S28). In contrast to the stiffness-toughness problem, our pipeline is able to close the sim-to-real gap after merely one round. The reasons for the fast progress include: 1) stiffness is a linear property; 2) the fabrication process of test samples is precise; 3) experimental behaviors of the samples are very consistent; 4) the simulator only has two exposed parameters.

Hashin-Shtrikman upper bound. Once our pipeline finished, we verified that the discovered designs on the experimental

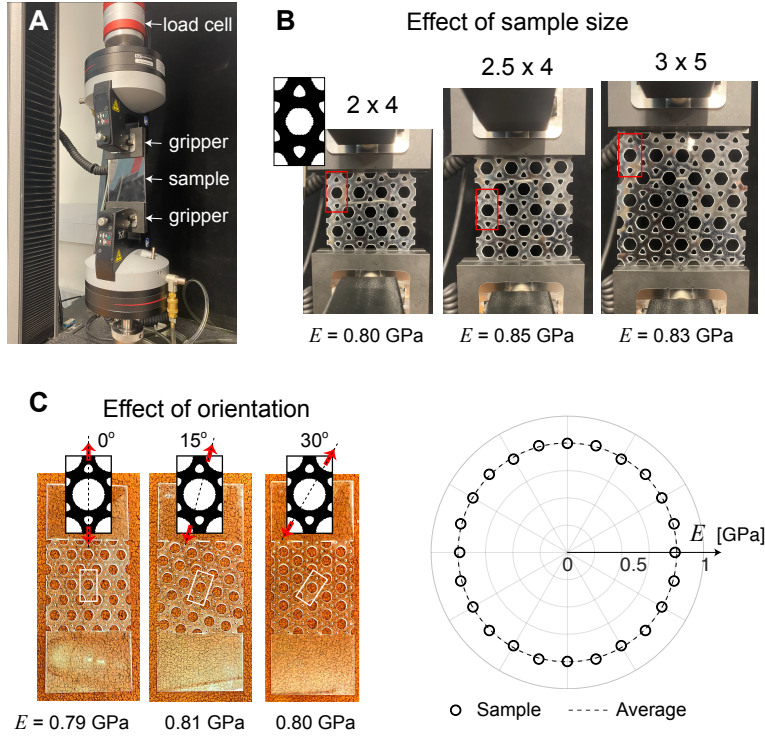


Fig. S27. Experimental measurement of Young's modulus. (A) The sample is gripped by the Instron on both ends outside the microstructure area. The Instron then stretches the sample by 0.1 mm at a rate of 1 mm/minute. The effective Young's modulus is derived from the slope of the measured force-displacement curve. (B) Measured effective Young's moduli for samples comprising different numbers of units. The Young's modulus barely varies as we increase the sample size, indicating that the boundary effect of a 3×5 sample is sufficiently small. (C) Measured effective Young's moduli along different directions, obtained from samples with rotated microstructures. Since the microstructure pattern obeys $p6m$ symmetry, we only need to rotate the microstructures by up to 30° . The results prove that the elastic response of the measured design is isotropic.

Pareto front are theoretically optimal solutions. In 1963, Hashin and Shtrikman derived the theoretical upper bound of the Young's modulus of an isotropic material under different porosities as (41,42,73),

$$E_{HS} = E_{\text{base}}(1 - \phi)/(1 + 2\phi), \quad (22)$$

where E_{base} is the Young's modulus of the base material, ϕ is porosity, and E_{base} gives the Hashin-Shtrikman (HS) upper bound. Therefore, for any porous material with porosity ϕ , its isotropic Young's modulus $E \leq E_{HS}(\phi)$. Equation (22) defines the theoretical upper bound for the Pareto front of Young's modulus and porosity, providing a benchmark for structural optimization algorithms. In Fig. S28, we compare experimental and simulation results of Pareto-optimal designs discovered by our pipeline with the HS upper bound (black line). We find that our results closely align with the theoretical upper bound. This proves that our pipeline is able to find theoretically optimal solutions to the stiffness-porosity problem.

5.3 Sample efficiency

In this section, we compare NMO with modified NSGA-II and TO in sample efficiency when discovering the simulation Pareto front. Here, the modified NSGA-II algorithm uses a carefully designed mutation operator. The mutation operator probabilistically adds/removes a group of randomly selected boundary pixels, adds/removes ellipses with random shapes and orientations, or adds a line with random thickness and orientation. Each random number controlling the magnitude of a mutation is drawn from an exponential distribution governed by its rate parameter λ . The TO algorithm implements SIMP and runs continuous gradient-based optimization using the method of moving asymptotes (MMA). Echoing the valid porosity range of the design space, we set the volume fraction constraint ($V_{\text{max}} = 1 - \phi$) between 0.25 and 0.9 and maximize the homogenized Young's modulus to discover porous microstructures near the HS upper bound. Considering that TO only yields one design at a time, we conduct a series of optimization tasks under evenly sampled V_{max} and arrange their order using a greedy strategy to expedite hypervolume growth. More specifically, we choose each subsequent task to maximize the expected

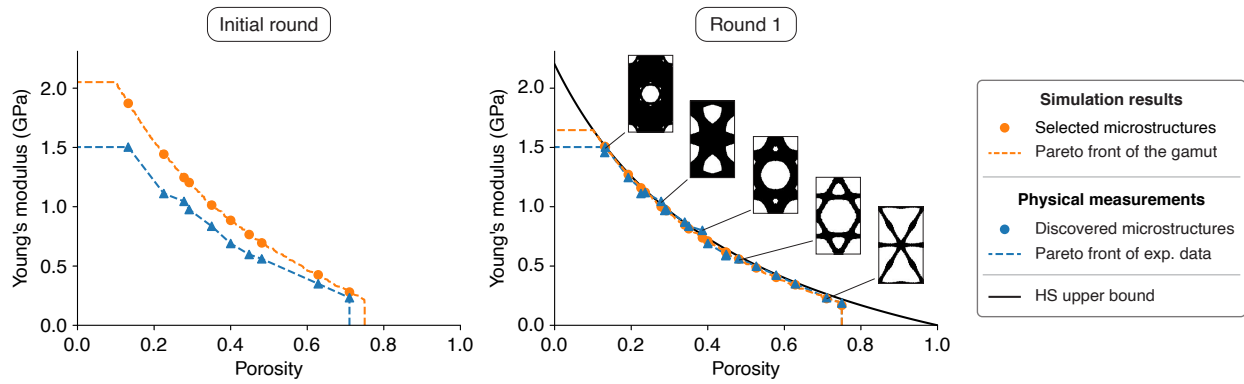


Fig. S28. Evolution of experimental and simulation Pareto fronts during the outer loop. The sim-to-real gap closes after one round. The discovered microstructures are validated against the HS upper bound (black line). We provide snapshots of several representative designs on the HS upper bound.

Pareto hypervolume improvement from discovered designs assuming that the HS upper bound will be reached. The resulting design is discarded if it violates the volume fraction constraint after quantization.

Fig. S29 illustrates how Pareto hypervolume grows with the number of simulation evaluations for NMO, NSGA-II, and TO-SIMP (with an MMA solver). Considering that TO only yields one optimal design at a time, we conduct a series of optimizations using different weighted combinations of properties and active constraints. Their order is arranged in a greedy strategy that maximizes theoretical hypervolume improvement according to the HS upper bound. Again, NMO shows the highest sample efficiency especially in the early stage of the optimization when fewer than 1,000 samples are evaluated (Fig. S29A). It takes NMO only 400 simulations to reach 90% hypervolume of the theoretical HS upper bound, 900 simulations for NSGA-II, and almost 4,000 simulations for TO (Fig. S29C). This is because the neural network inside NMO quickly explores the entire design space and proposes valuable designs to the simulator after the optimization starts. Since stiffness measurement is a linear problem, CNNs can learn to predict Young's modulus reasonably well with a small amount of training data. This allows NMO to rapidly expand its Pareto hypervolume. In comparison, TO has to follow a sequence of optimization trajectories to reach the entire HS upper bound, consuming a large number of intermediate evaluations despite the usage of MMA.

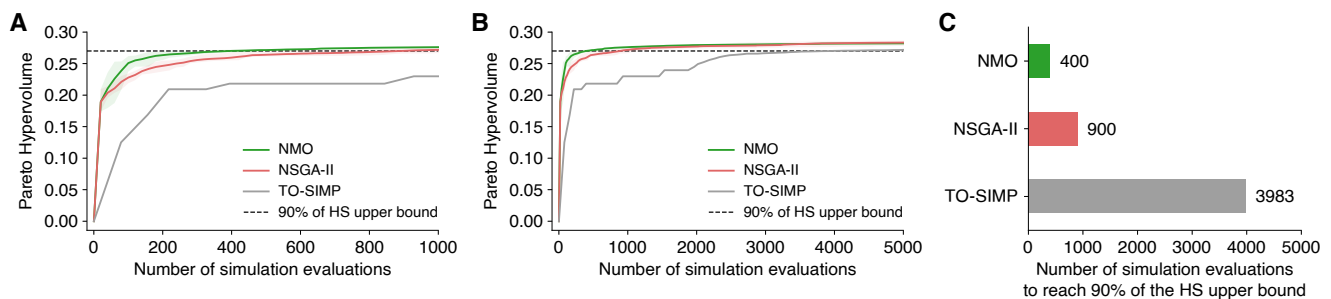


Fig. S29. Comparison between NMO and other multi-objective optimization algorithms in Pareto hypervolume growth. The comparison is made within a budget of (A) 1,000 simulation evaluations and (B) 5,000 simulation evaluations. (C) Number of evaluations required for NMO and other baseline algorithms to reach 90% of the theoretical maximal hypervolume.

We note that NMO is surpassed by NSGA-II as the Pareto hypervolume approaches the HS upper bound (Fig. S29B). This is mainly due to the shrinking room for Pareto hypervolume growth at a later stage of the optimization. Although the neural network initially accelerates the exploration of the design space, giving accurate predictions for remaining designs near the HS upper bound eventually comes in higher priority. Therefore, the hypervolume growth of NMO slows down as the network tries to improve predictions for those top-performing designs. The stiffness-toughness trade-off introduced in the manuscript, however, has an extremely non-smooth landscape and an unknown theoretical upper bound. In that case, it is always preferable to explore the design space using NMO to expedite hypervolume growth within a tight simulation budget.

A Tables

Young's modulus			Toughness		
Name	Min.	Max.	Name	Min.	Max.
E_{TB}	8.08e5	1.21e6	E_{TB}	1e5	2e6
E_{VW}	2.13e9	3.19e9	E_{VW}	1e7	1e9
E_{IF}	4.51e8	6.77e8	E_{IF}	1e7	1e9
ν_{TB}	0.4	0.5	ν_{TB}	0.4	0.5
ν_{VW}	0.2	0.4	ν_{VW}	0.2	0.4
ν_{IF}	0.3	0.4	ν_{IF}	0.3	0.4
-	-	-	W_{TB}	5e4	1e6
-	-	-	W_{VW}	5e4	1e7
-	-	-	W_{IF}	1e6	1e8
γ	2e4	2e6	γ	2e4	4e6

Table S9. Value ranges of simulation parameters used in system identification when fitting to physical measurements of base materials.

Young's modulus			Toughness		
Name	Min.	Max.	Name	Min.	Max.
E_{TB}	8e5	1.2e6	E_{TB}	5e5	5e8
E_{VW}	2.08e9	3.12e9	E_{VW}	5e7	1e9
E_{IF}	4.48e8	6.72e8	E_{IF}	1e7	5e8
ν_{TB}	0.4	0.5	ν_{TB}	0.4	0.5
ν_{VW}	0.2	0.4	ν_{VW}	0.2	0.4
ν_{IF}	0.3	0.4	ν_{IF}	0.3	0.4
-	-	-	W_{TB}	5e4	1e6
-	-	-	W_{VW}	5e5	1e7
-	-	-	W_{IF}	1e6	1e8
γ	2e4	2e6	γ	2e4	4e6

Table S10. Value ranges of simulation parameters used in system identification when fitting to physical measurements of microstructures.

Round	No.	ϕ	Exp.	Sim. (R0)	Rel. err.	Sim. (R1)	Rel. err.	Sim. (R2)	Rel. err.	Sim. (R3)	Rel. err.	Sim. (R4)	Rel. err.
0	1	0.61	0.640	0.402	0.372	0.653	0.020	0.633	0.011	0.632	0.013	0.637	0.004
	2	0.76	1.400	0.980	0.300	1.281	0.085	1.216	0.131	1.185	0.153	1.185	0.154
	3	0.67	0.948	0.663	0.301	0.931	0.018	0.889	0.062	0.878	0.074	0.881	0.070
	4	0.83	1.541	1.333	0.135	1.646	0.068	1.559	0.012	1.510	0.020	1.505	0.023
	5	0.84	1.747	1.509	0.136	1.826	0.045	1.718	0.017	1.654	0.053	1.647	0.058
	6	0.51	0.347	0.118	0.659	0.346	0.002	0.340	0.020	0.349	0.005	0.357	0.029
	7	0.51	0.301	0.133	0.560	0.370	0.229	0.363	0.204	0.370	0.228	0.378	0.254
	8	0.54	0.330	0.023	0.932	0.347	0.053	0.342	0.037	0.356	0.080	0.368	0.115
	9	0.60	0.533	0.261	0.511	0.526	0.013	0.511	0.041	0.515	0.034	0.523	0.020
	10	0.75	1.178	0.986	0.163	1.231	0.045	1.180	0.002	1.155	0.020	1.153	0.021
	11	0.66	0.954	0.603	0.368	0.801	0.160	0.784	0.178	0.778	0.185	0.780	0.182
	12	0.74	1.308	0.980	0.251	1.232	0.059	1.171	0.105	1.142	0.127	1.140	0.128
	13	0.76	1.375	1.075	0.218	1.330	0.033	1.264	0.080	1.231	0.104	1.228	0.107
	14	0.82	1.659	1.255	0.243	1.539	0.072	1.470	0.114	1.424	0.141	1.420	0.144
	15	0.84	1.790	1.479	0.174	1.775	0.008	1.681	0.061	1.625	0.092	1.617	0.096
	16	0.85	1.767	1.523	0.138	1.827	0.034	1.729	0.021	1.674	0.053	1.666	0.057
	17	0.85	1.771	1.604	0.094	1.915	0.081	1.807	0.020	1.742	0.016	1.733	0.022
	18	0.85	1.807	1.729	0.043	2.080	0.151	1.934	0.070	1.844	0.020	1.833	0.014
1	19	0.64	0.819	0.605	0.261	0.854	0.042	0.810	0.012	0.793	0.032	0.796	0.028
	20	0.58	0.860	0.736	0.144	0.919	0.068	0.876	0.018	0.854	0.007	0.853	0.008
	21	0.67	0.980	0.874	0.108	1.115	0.139	1.054	0.076	1.021	0.042	1.019	0.041
	22	0.77	1.324	1.162	0.123	1.474	0.113	1.378	0.041	1.328	0.003	1.325	0.001
	23	0.79	1.572	1.356	0.138	1.678	0.067	1.566	0.004	1.502	0.045	1.496	0.049
	24	0.79	1.402	1.479	0.054	1.792	0.278	1.661	0.185	1.583	0.129	1.574	0.123
	25	0.85	1.764	1.613	0.086	1.963	0.113	1.838	0.042	1.760	0.002	1.751	0.007
	26	0.85	1.810	1.738	0.040	2.089	0.154	1.945	0.074	1.856	0.025	1.845	0.019
2	27	0.68	0.987	0.774	0.215			0.961	0.026	0.944	0.044	0.944	0.043
	28	0.74	1.227	1.056	0.140			1.273	0.038	1.232	0.004	1.230	0.002
	29	0.75	1.277	1.166	0.087			1.361	0.066	1.313	0.028	1.309	0.025
	30	0.79	1.308	1.281	0.021			1.480	0.132	1.432	0.095	1.427	0.091
	31	0.80	1.450	1.355	0.065			1.577	0.088	1.516	0.046	1.510	0.042
	32	0.85	1.630	1.541	0.055			1.782	0.093	1.709	0.048	1.702	0.044
	33	0.85	1.690	1.723	0.019			1.923	0.138	1.833	0.085	1.822	0.078
	34	0.85	1.679	1.724	0.027			1.930	0.150	1.841	0.097	1.831	0.091
3	35	0.68	0.847	0.724	0.145					0.890	0.051	0.891	0.052
	36	0.66	0.861	0.904	0.050					1.035	0.203	1.033	0.200
	37	0.74	1.317	1.033	0.216					1.206	0.084	1.204	0.086
	38	0.76	1.302	1.207	0.073					1.362	0.047	1.358	0.043
	39	0.79	1.528	1.305	0.146					1.459	0.045	1.454	0.048
	40	0.82	1.629	1.443	0.114					1.594	0.021	1.587	0.025
	41	0.85	1.741	1.673	0.039					1.805	0.037	1.796	0.032
	42	0.85	1.751	1.749	0.001					1.858	0.061	1.847	0.054
4	43	0.70	1.093	0.968	0.115							1.112	0.017
	44	0.74	1.164	1.015	0.128							1.167	0.003
	45	0.77	1.297	1.151	0.112							1.299	0.002
	46	0.79	1.372	1.248	0.090							1.396	0.017
	47	0.84	1.683	1.505	0.105							1.639	0.026
	48	0.85	1.683	1.538	0.086							1.672	0.006
	49	0.85	1.649	1.561	0.053							1.701	0.032
	50	0.85	1.593	1.680	0.055							1.801	0.130

Table S11. Experimental Young’s moduli, simulation predictions, and relative errors of all microstructures discovered by our approach. R_i = Round i . Text in blue indicates predictions for the next round. The unit of Young’s modulus is GPa.

Round	No.	ϕ	Exp.	Sim. (R0)	Rel. err.	Sim. (R1)	Rel. err.	Sim. (R2)	Rel. err.	Sim. (R3)	Rel. err.	Sim. (R4)	Rel. err.
0	1	0.61	0.323	0.528	0.633	0.361	0.116	0.254	0.214	0.239	0.261	0.309	0.045
	2	0.76	0.201	0.329	0.637	0.227	0.131	0.236	0.173	0.143	0.290	0.221	0.102
	3	0.67	0.271	0.398	0.470	0.395	0.458	0.327	0.207	0.286	0.056	0.273	0.009
	4	0.83	0.276	0.307	0.114	0.274	0.007	0.246	0.109	0.190	0.310	0.203	0.264
	5	0.84	0.375	0.277	0.261	0.310	0.171	0.282	0.247	0.214	0.429	0.211	0.436
	6	0.51	0.309	0.487	0.574	0.255	0.177	0.275	0.110	0.190	0.387	0.275	0.110
	7	0.51	0.245	0.231	0.057	0.270	0.099	0.132	0.461	0.141	0.425	0.221	0.098
	8	0.54	0.117	0.022	0.809	0.102	0.126	0.078	0.329	0.102	0.124	0.143	0.226
	9	0.60	0.252	0.222	0.119	0.344	0.365	0.194	0.227	0.204	0.188	0.244	0.029
	10	0.75	0.211	0.482	1.283	0.343	0.622	0.263	0.248	0.227	0.073	0.343	0.624
	11	0.66	0.371	1.304	2.515	0.410	0.105	0.359	0.031	0.310	0.165	0.386	0.039
	12	0.74	0.261	1.112	3.268	0.407	0.564	0.413	0.586	0.218	0.162	0.394	0.514
	13	0.76	0.391	1.091	1.791	0.398	0.018	0.294	0.247	0.303	0.225	0.420	0.074
	14	0.82	0.109	0.945	7.674	0.159	0.457	0.120	0.104	0.110	0.012	0.099	0.095
	15	0.84	0.355	0.865	1.436	0.236	0.335	0.278	0.218	0.179	0.495	0.131	0.630
	16	0.85	0.326	0.786	1.407	0.342	0.048	0.315	0.035	0.276	0.154	0.222	0.319
	17	0.85	0.313	0.603	0.927	0.291	0.069	0.222	0.292	0.223	0.289	0.243	0.224
	18	0.85	0.304	0.424	0.395	0.378	0.244	0.324	0.067	0.305	0.005	0.271	0.108
1	19	0.64	0.424	0.295	0.304	0.700	0.650	0.295	0.304	0.540	0.275	0.448	0.056
	20	0.58	0.426	0.553	0.300	0.892	1.095	0.355	0.166	0.345	0.189	0.458	0.075
	21	0.67	0.227	0.669	1.950	0.970	3.277	0.455	1.005	0.418	0.843	0.428	0.889
	22	0.77	0.513	0.554	0.079	0.982	0.913	0.449	0.125	0.526	0.024	0.346	0.326
	23	0.79	0.426	0.452	0.062	0.865	1.033	0.452	0.063	0.494	0.161	0.343	0.193
	24	0.79	0.491	0.315	0.359	0.680	0.383	0.371	0.245	0.447	0.089	0.254	0.482
	25	0.85	0.458	0.280	0.389	0.584	0.276	0.431	0.058	0.374	0.182	0.266	0.418
	26	0.85	0.473	0.314	0.336	0.391	0.172	0.309	0.346	0.315	0.333	0.206	0.564
2	27	0.68	0.315	0.612	0.943			0.598	0.897	0.338	0.074	0.467	0.483
	28	0.74	0.331	0.395	0.195			0.611	0.847	0.340	0.029	0.304	0.081
	29	0.75	0.493	0.423	0.141			0.589	0.196	0.530	0.075	0.390	0.209
	30	0.79	0.381	0.364	0.045			0.541	0.417	0.269	0.295	0.394	0.034
	31	0.80	0.106	0.304	1.865			0.505	3.761	0.175	0.648	0.172	0.620
	32	0.85	0.369	0.349	0.053			0.466	0.263	0.330	0.105	0.289	0.216
	33	0.85	0.303	0.266	0.122			0.369	0.218	0.151	0.501	0.260	0.143
	34	0.85	0.360	0.310	0.139			0.327	0.092	0.172	0.521	0.218	0.394
3	35	0.68	0.339	0.504	0.485					0.641	0.888	0.291	0.142
	36	0.66	0.508	0.363	0.286					0.684	0.347	0.306	0.397
	37	0.74	0.276	0.389	0.412					0.716	1.598	0.360	0.307
	38	0.76	0.278	0.386	0.388					0.667	1.401	0.308	0.108
	39	0.79	0.351	0.434	0.235					0.633	0.802	0.312	0.111
	40	0.82	0.329	0.479	0.453					0.544	0.650	0.265	0.196
	41	0.85	0.412	0.294	0.287					0.424	0.030	0.237	0.424
	42	0.85	0.400	0.239	0.403					0.261	0.346	0.173	0.568
4	43	0.70	0.357	0.713	0.997							0.590	0.653
	44	0.74	0.247	0.640	1.593							0.564	1.287
	45	0.77	0.326	0.544	0.669							0.546	0.672
	46	0.79	0.425	0.516	0.213							0.510	0.200
	47	0.84	0.263	0.361	0.374							0.450	0.712
	48	0.85	0.348	0.398	0.144							0.398	0.144
	49	0.85	0.348	0.419	0.203							0.369	0.060
	50	0.85	0.394	0.302	0.233							0.256	0.350

Table S12. Experimental toughness, simulation predictions, and relative errors of all microstructures discovered by our approach. R_i = Round i . Text in blue indicates predictions for the next round. The unit of toughness is MPa.

Family	1		2		3		4	
	No.	E (GPa)	No.	E (GPa)	No.	E (GPa)	No.	E (GPa)
Seed microstructures	22	1.324	18	1.807	15	1.790	5	1.747
	24	1.402	26	1.810	16	1.767	17	1.771
	36	0.861	41	1.741				
			42	1.751				
Validation microstructures	1-1	1.253	2-1	1.541	3-1	1.470	4-1	1.447
	1-2	1.329	2-2	1.779	3-2	1.703	4-2	1.679
	1-3	1.485	2-3	1.703	3-3	1.689	4-3	1.621

Family	1		2		3		4	
	No.	T (MPa)	No.	T (MPa)	No.	T (MPa)	No.	T (MPa)
Seed microstructures	22	0.513	18	0.304	15	0.355	5	0.375
	24	0.491	26	0.473	16	0.326	17	0.313
	36	0.508	41	0.412				
			42	0.400				
Validation microstructures	1-1	0.369	2-1	0.343	3-1	0.330	4-1	0.381
	1-2	0.447	2-2	0.388	3-2	0.269	4-2	0.347
	1-3	0.420	2-3	0.458	3-3	0.350	4-3	0.433

Table S13. Comparison between the physical measurements of seed microstructures and validation microstructures in each family. Microstructures in the same family show comparable Young's moduli and toughnesses.

Seed microstructures

Family	No	ϕ	Exp.	Sim. (R4)	Rel. err.	Sim. (FS)	Rel. err.
1	22	0.77	1.324	1.325	0.001	1.158	0.126
	24	0.79	1.402	1.574	0.123	1.403	0.001
	36	0.66	0.861	1.033	0.200	0.913	0.061
2	18	0.85	1.807	1.833	0.014	1.799	0.004
	26	0.85	1.810	1.845	0.019	1.809	0.001
	41	0.85	1.741	1.796	0.032	1.756	0.009
	42	0.85	1.751	1.847	0.054	1.812	0.034
3	15	0.84	1.790	1.617	0.096	1.714	0.042
	16	0.85	1.767	1.666	0.057	1.767	0.000
4	5	0.84	1.747	1.647	0.058	1.672	0.043
	17	0.85	1.771	1.733	0.022	1.771	0.000

Validation microstructures

Family	No	ϕ	Exp.	Sim. (R4)	Rel. err.	Sim. (FS)	Rel. err.
1	F1-1	0.73	1.253	1.222	0.024	1.066	0.149
	F1-2	0.75	1.329	1.359	0.022	1.204	0.094
	F1-3	0.79	1.485	1.451	0.022	1.284	0.135
2	F2-1	0.81	1.541	1.654	0.073	1.621	0.052
	F2-2	0.85	1.779	1.825	0.026	1.794	0.009
	F2-3	0.85	1.703	1.827	0.073	1.795	0.054
3	F3-1	0.83	1.470	1.510	0.027	1.582	0.076
	F3-2	0.85	1.703	1.725	0.013	1.840	0.081
	F3-3	0.85	1.689	1.724	0.021	1.841	0.090
4	F4-1	0.80	1.447	1.426	0.015	1.430	0.012
	F4-2	0.84	1.679	1.699	0.012	1.728	0.030
	F4-3	0.85	1.621	1.736	0.071	1.774	0.094

Table S14. Experimental Young's modulus, simulation predictions, and relative errors of seed microstructures and validation microstructures in each family. 'R4' refers to the virtual testers in Round 4. 'FS' means family-specific virtual testers. The unit of Young's modulus is GPa.

Seed microstructures

Family	No	ϕ	Exp.	Sim. (R4)	Rel. err.	Sim. (FS)	Rel. err.
1	22	0.77	0.513	0.346	0.326	0.510	0.006
	24	0.79	0.491	0.254	0.482	0.468	0.048
	36	0.66	0.508	0.306	0.397	0.466	0.083
2	18	0.85	0.304	0.271	0.108	0.328	0.080
	26	0.85	0.473	0.206	0.564	0.416	0.121
	41	0.85	0.412	0.237	0.424	0.428	0.039
	42	0.85	0.400	0.173	0.568	0.315	0.213
3	15	0.84	0.355	0.131	0.630	0.336	0.054
	16	0.85	0.326	0.222	0.319	0.326	0.000
4	5	0.84	0.375	0.211	0.436	0.375	0.002
	17	0.85	0.313	0.243	0.224	0.315	0.007

Validation microstructures

Family	No	ϕ	Exp.	Sim. (R4)	Rel. err.	Sim. (FS)	Rel. err.
1	F1-1	0.73	0.369	0.375	0.017	0.748	1.028
	F1-2	0.75	0.447	0.386	0.136	0.626	0.399
	F1-3	0.79	0.420	0.305	0.275	0.535	0.272
2	F2-1	0.81	0.343	0.273	0.204	0.546	0.593
	F2-2	0.85	0.388	0.115	0.703	0.561	0.446
	F2-3	0.85	0.458	0.209	0.544	0.420	0.083
3	F3-1	0.83	0.330	0.281	0.147	0.535	0.623
	F3-2	0.85	0.269	0.197	0.271	0.445	0.653
	F3-3	0.85	0.350	0.169	0.516	0.312	0.106
4	F4-1	0.80	0.381	0.258	0.321	0.554	0.455
	F4-2	0.84	0.347	0.252	0.274	0.459	0.322
	F4-3	0.85	0.433	0.241	0.444	0.313	0.277

Table S15. Experimental toughness, simulation predictions, and relative errors of seed microstructures and validation microstructures in each family. 'R4' refers to the virtual testers in Round 4. 'FS' means family-specific virtual testers. The unit of toughness is MPa.

B Figures

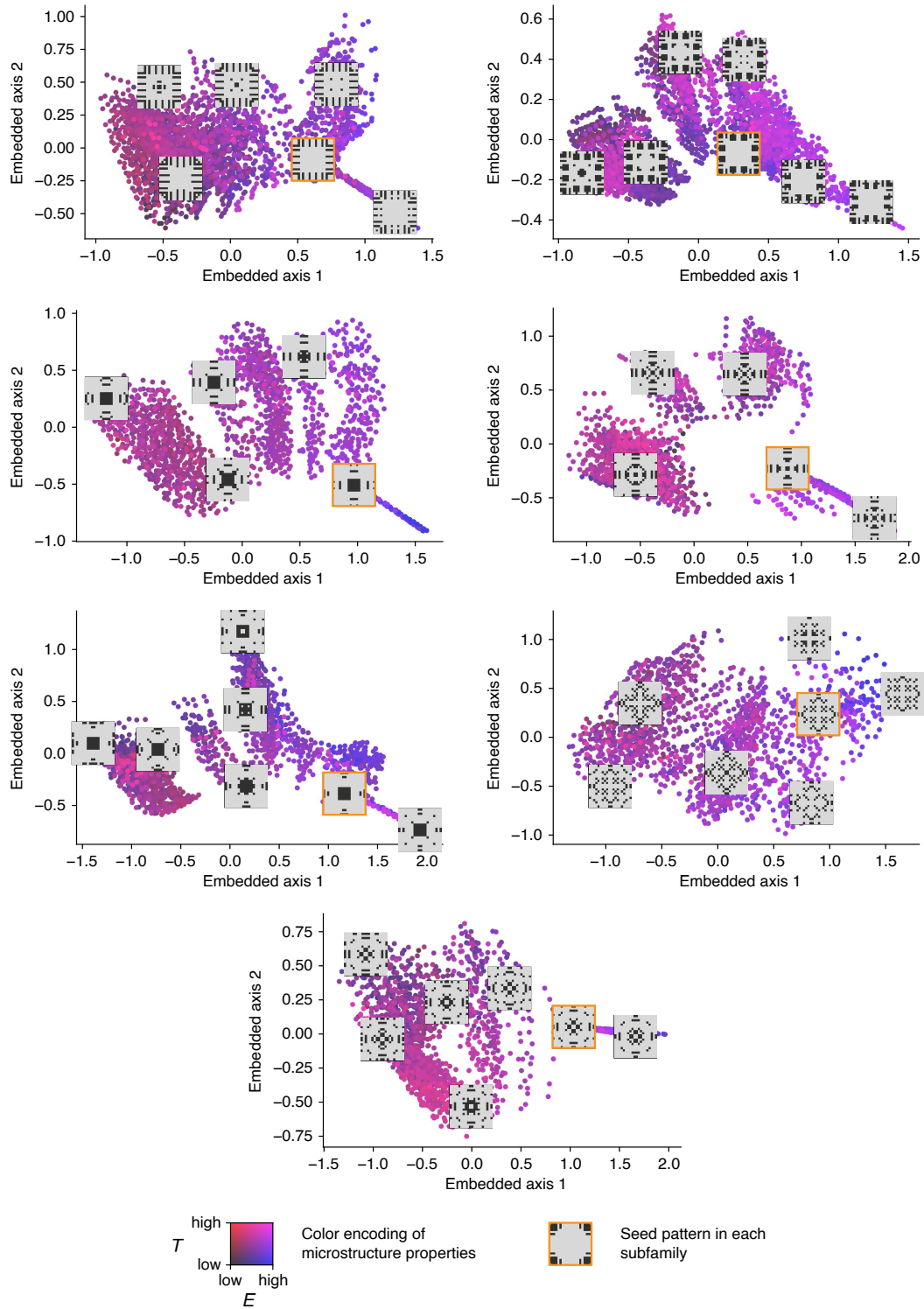


Fig. S30. Isomap embedding spaces of subfamilies not displayed in Fig. S17. Each embedding space contains around 2,000 to 3,000 samples and a few of them are visualized. Microstructure properties are encoded in colors (E : the blue channel; T : the red channel). Seed patterns are marked in orange boxes.

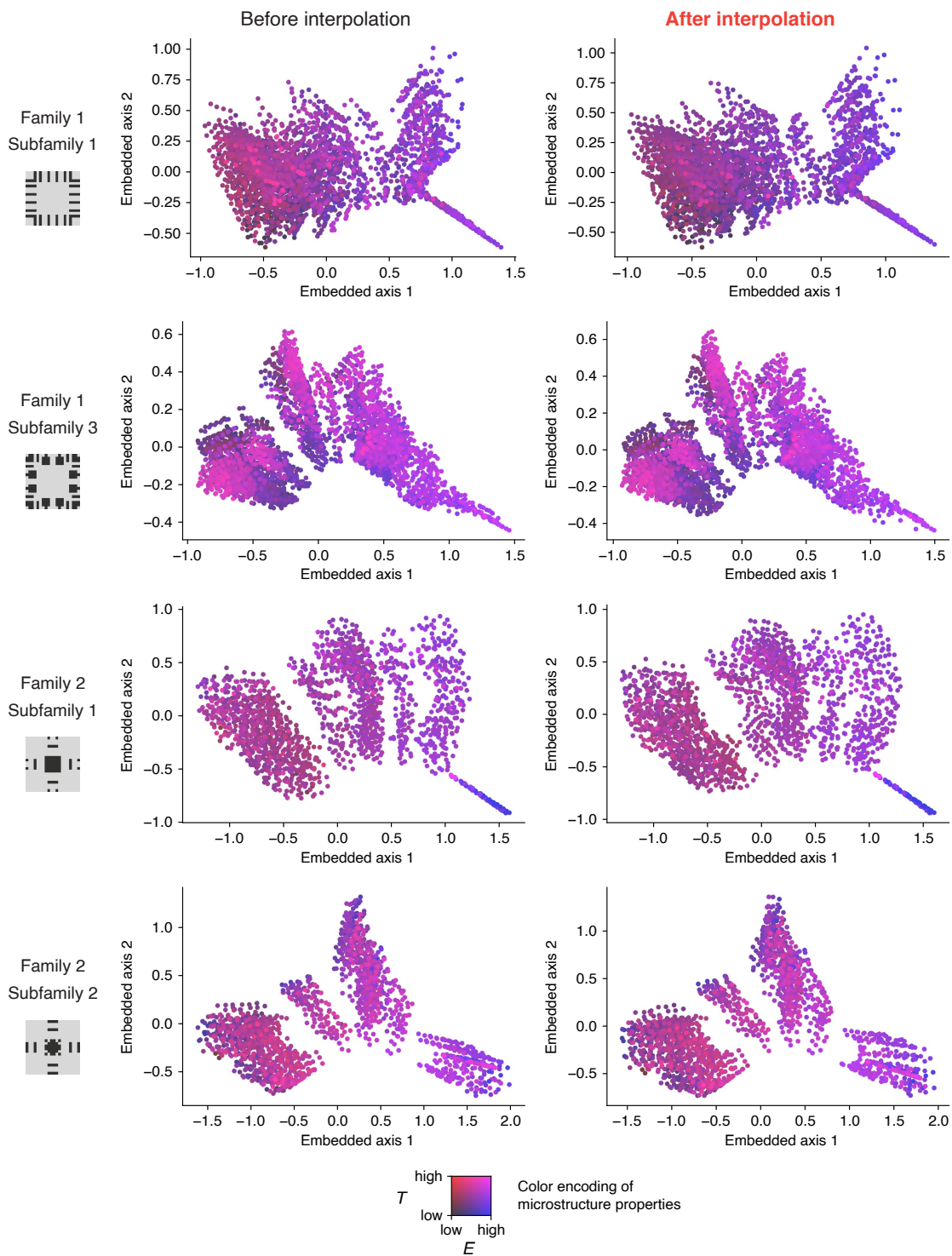


Fig. S31. Isomap embedding spaces of more subfamilies before and after refinement using interpolation (Part 1). Microstructure properties are encoded in colors.

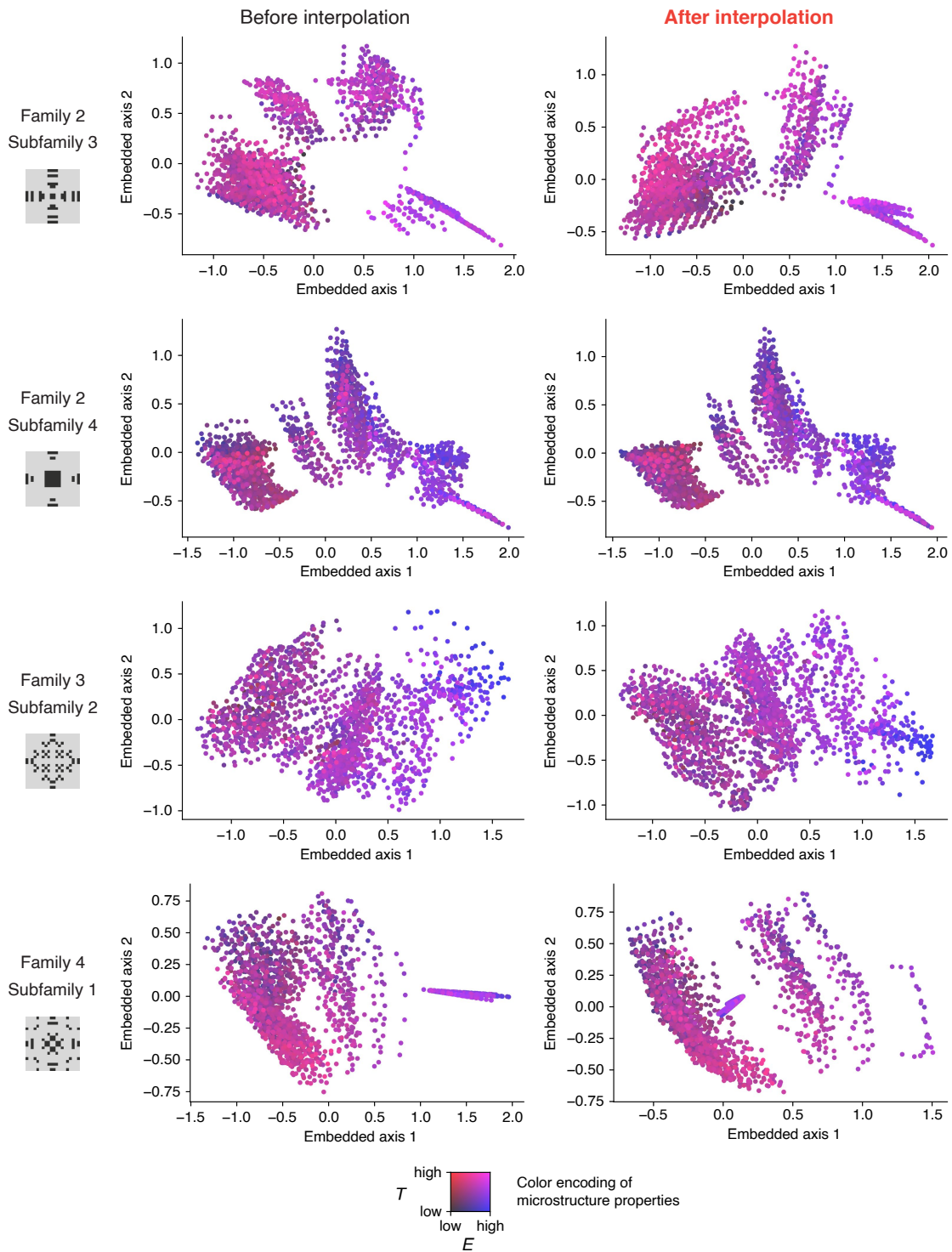


Fig. S32. Isomap embedding spaces of more subfamilies before and after refinement using interpolation (Part 2). Microstructure properties are encoded in colors.

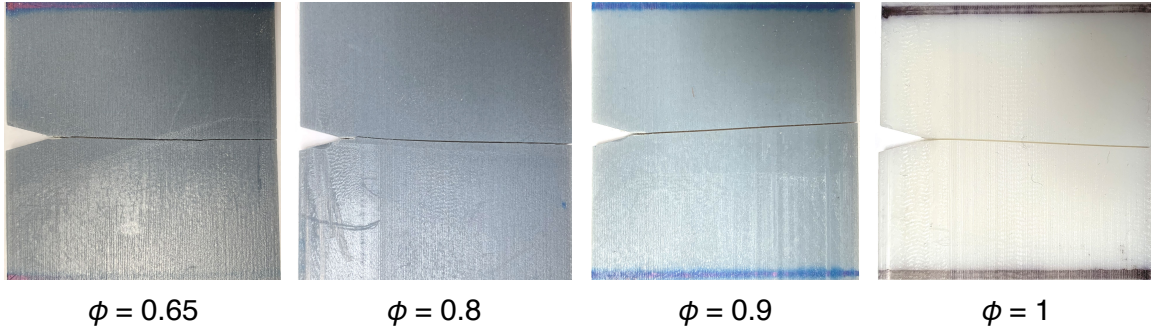


Fig. S33. Clean cleavage is observed after crack propagation in homogeneous composites with various fractions of rigid material.

REFERENCES AND NOTES

1. R. O. Ritchie, The conflicts between strength and toughness. *Nat. Mater.* **10**, 817–822 (2011).
2. G. Mayer, Rigid biological systems as models for synthetic composites. *Science* **310**, 1144–1147 (2005).
3. C. Sanchez, H. Arribart, M. M. G. Guille, Biomimetism and bioinspiration as tools for the design of innovative materials and systems. *Nat. Mater.* **4**, 277–288 (2005).
4. L. S. Dimas, G. H. Bratzel, I. Eylon, M. J. Buehler, Tough composites inspired by mineralized natural materials: Computation, 3d printing, and testing. *Adv. Funct. Mater.* **23**, 4629–4638 (2013).
5. M. A. Meyers, J. McKittrick, P.-Y. Chen, Structural biological materials: Critical mechanics-materials connections. *Science* **339**, 773–779 (2013).
6. Z. Yin, F. Hannard, F. Barthelat, Impact-resistant nacre-like transparent materials. *Science* **364**, 1260–1263 (2019).
7. N. A. Dudukovic, E. J. Fong, H. B. Gameda, J. R. DeOtte, M. R. Ceron, B. D. Moran, J. T. Davis, S. E. Baker, E. B. Duoss, Cellular fluidics. *Nature* **595**, 58–65 (2021).
8. T. A. Schaedler, A. J. Jacobsen, A. Torrents, A. E. Sorensen, J. Lian, J. R. Greer, L. Valdevit, W. B. Carter, Ultralight metallic microlattices. *Science* **334**, 962–965 (2011).
9. X. Zheng, H. Lee, T. H. Weisgraber, M. Shusteff, J. DeOtte, E. B. Duoss, J. D. Kuntz, M. M. Biener, Q. Ge, J. A. Jackson, S. O. Kucheyev, N. X. Fang, C. M. Spadaccini, Ultralight, ultrastiff mechanical metamaterials. *Science* **344**, 1373–1377 (2014).
10. M.-S. Pham, C. Liu, I. Todd, J. Lertthanasarn, Damage-tolerant architected materials inspired by crystal microstructure. *Nature* **565**, 305–311 (2019).

11. M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, R. J. Wood, An integrated design and fabrication strategy for entirely soft, autonomous robots. *Nature* **536**, 451–455 (2016).
12. N. Aage, E. Andreassen, B. S. Lazarov, O. Sigmund, Giga-voxel computational morphogenesis for structural design. *Nature* **550**, 84–86 (2017).
13. D. Chen, M. Skouras, B. Zhu, W. Matusik, Computational discovery of extremal microstructure families. *Sci. Adv.* **4**, eaao7005 (2018).
14. K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
15. G. X. Gu, C.-T. Chen, D. J. Richmond, M. J. Buehler, Bioinspired hierarchical composite design using machine learning: Simulation, additive manufacturing, and experiment. *Mater. Horiz.* **5**, 939–945 (2018).
16. O. Sigmund, K. Maute, Topology optimization approaches. *Struct. Multidiscipl. Optim.* **48**, 1031–1055 (2013).
17. J. Wu, O. Sigmund, J. P. Groen, Topology optimization of multi-scale structures: A review. *Struct. Multidiscipl. Optim.* **63**, 1455–1480 (2021).
18. Y. Yang, Z. Song, G. Lu, Q. Zhang, B. Zhang, B. Ni, C. Wang, X. Li, L. Gu, X. Xie, H. Gao, J. Lou, Intrinsic toughening and stable crack propagation in hexagonal boron nitride. *Nature* **594**, 57–61 (2021).
19. A. J. D. Shaikeea, H. Cui, M. O’Masta, X. R. Zheng, V. S. Deshpande, The toughness of mechanical metamaterials. *Nat. Mater.* **21**, 297–304 (2022).
20. J. U. Surjadi, Y. Lu, Design criteria for tough metamaterials. *Nat. Mater.* **21**, 272–274 (2022).

21. G. R. Zavala, A. J. Nebro, F. Luna, C. A. Coello Coello, A survey of multi-objective metaheuristics applied to structural optimization. *Struct. Multidiscipl. Optim.* **49**, 537–558 (2014).
22. M. Hossain, C.-J. Hsueh, B. Bourdin, K. Bhattacharya, Effective toughness of heterogeneous media. *J. Mech. Phys. Solids* **71**, 15–32 (2014).
23. A. Kumar, O. Lopez-Pamies, The phase-field approach to self-healable fracture of elastomers: A model accounting for fracture nucleation at large, with application to a class of conspicuous experiments. *Theor. Appl. Fract. Mech.* **107**, 102550 (2020).
24. B. Bourdin, G. A. Francfort, J.-J. Marigo, The variational approach to fracture. *J. Elast.* **91**, 5–148 (2008).
25. Y. Jia, O. Lopez-Pamies, X. S. Zhang, Controlling the fracture response of structures via topology optimization: From delaying fracture nucleation to maximizing toughness. *J. Mech. Phys. Solids* **173**, 105227 (2023).
26. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002).
27. X. Huang, Y. Xie, Topology optimization of nonlinear structures under displacement loading. *Eng. Struct.* **30**, 2057–2068 (2008).
28. O. Sigmund, A 99 line topology optimization code written in matlab. *Struct. Multidiscipl. Optim.* **21**, 120–127 (2001).
29. M. K. Lukovic, Y. Tian, W. Matusik, Diversity-guided multi-objective bayesian optimization with batch evaluations. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)* (2020).
30. E. Bradford, A. M. Schweidtmann, A. Lapkin, Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm. *J. Glob. Optim.* **71**, 407–438 (2018).

31. M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
32. S. Shalev-Shwartz, *Online learning and online convex optimization* (Foundations and trends in Machine Learning, 2011), vol. 4, pp. 107–194 .
33. Q. Xie, M.-T. Luong, E. Hovy, Q. V. Le, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2020), pp. 10687–10698.
34. U. G. Wegst, H. Bai, E. Saiz, A. P. Tomsia, R. O. Ritchie, Bioinspired structural materials. *Nat. Mater.* **14**, 23–36 (2015).
35. M. Balasubramanian, E. L. Schwartz, The isomap algorithm and topological stability. *Science* **295**, 7 (2002).
36. G. Bao, Z. Suo, Remarks on crack-bridging concepts. *Appl. Mech. Rev.* **45**, 355–366 (1992).
37. G. X. Gu, M. Takaffoli, M. J. Buehler, Hierarchically enhanced impact resistance of bioinspired composites. *Adv. Mater.* **29**, 1700060 (2017).
38. S. Suresh, Micromechanisms of fatigue crack growth retardation following overloads. *Eng. Fract. Mech.* **18**, 577–593 (1983).
39. K. M. Flores, R. H. Dauskardt, Enhanced toughness due to stable crack tip damage zones in bulk metallic glass. *Scr. Mater.* **41**, 937–943 (1999).
40. M. Ramulu, A. Kobayashi, *Mechanics of crack curving and branching—A dynamic fracture analysis* (Springer, 1985), pp. 61–75.
41. Z. Hashin, *Theory of Mechanical Behavior of Heterogeneous Media* (Towne School of Civil and Mechanical Engineering, University of Pennsylvania, 1963), pp. 1–9.
42. Z. Hashin, S. Shtrikman, A variational approach to the theory of the elastic behaviour of multiphase materials. *J. Mech. Phys. Solids* **11**, 127–140 (1963).

43. B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams, A. G. Doyle, Bayesian reaction optimization as a tool for chemical synthesis. *Nature* **590**, 89–96 (2021).
44. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
45. M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read, D. Baker, Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
46. D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2101784118 (2021).
47. M. Raissi, A. Yazdani, G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030 (2020).
48. P. Bauer, A. Thorpe, G. Brunet, The quiet revolution of numerical weather prediction. *Nature* **525**, 47–55 (2015).
49. T. Schneider, J. Teixeira, C. S. Bretherton, F. Brient, K. G. Pressel, C. Schär, A. P. Siebesma, Climate goals and computing the future of clouds. *Nat. Clim. Chang.* **7**, 3–5 (2017).

50. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **4**, eaau5872 (2019).
51. Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, F. Durand, Taichi: A language for high-performance computation on spatially sparse data structures. *ACM Trans. Graph.* **38**, 1–16 (2019).
52. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (IEEE, 2016), pp. 770–778.
53. C. Villani, *Topics in optimal transportation* (American Mathematical Society, 2003), vol. 58.
54. M. Agueh, G. Carlier, Barycenters in the wasserstein space. *SIAM J. Math. Anal.* **43**, 904–924 (2011).
55. L. V. Kantorovich, “On the translocation of masses,” (Doklady Akademii Nauk SSSR, 1942), vol. 37, pp. 199–201.
56. J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, L. Guibas, Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.* **34**, 1–11 (2015).
57. O. M. Querin, G. P. Steven, Y. M. Xie, Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Eng. Comput.* **15**, 1031–1048 (1998).
58. M. P. Bendsøe, Optimal shape design as a material distribution problem. *Struct. Optim.* **1**, 193–202 (1989).
59. G. X. Gu, F. Libonati, S. D. Wettermark, M. J. Buehler, Printing nature: Unraveling the role of nacre’s mineral bridges. *J. Mech. Behav. Biomed. Mater.* **76**, 135–144 (2017).
60. M. J. Mirzaali, A. H. de la Nava, D. Gunashekar, M. Nouri-Goushki, R. P. E. Veeger, Q. Grossman, L. Angeloni, M. K. Ghatkesar, L. E. Fratila-Apachitei, D. Ruffoni, E. L.

- Doubrovski, A. A. Zadpoor, Mechanics of bioinspired functionally graded soft-hard composites made by multi-material 3d printing. *Compos. Struct.* **237**, 111867 (2020).
61. E. Sifakis, J. Barbic, Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction (ACM, 2012), pp. 1–50.
62. P. Kelly, Solid mechanics part I: An introduction to solid mechanics. A creative commons attributions, Mountain View, CA 94042 (2013), pp. 241–324. (2013), pp. 241–324.
63. J. González, Z. Dai, P. Hennig, N. Lawrence, *Artificial intelligence and statistics* (PMLR, 2016), pp. 648–657.
64. M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Eng.* **71**, 197–224 (1988).
65. M. P. Bendsoe, O. Sigmund, *Topology optimization: Theory, methods, and applications* (Springer Science & Business Media, 2003).
66. J. Knowles, ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **10**, 50–66 (2006).
67. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG] (2014).
68. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 [cs.CV] (2020).
69. L. Xia, D. Da, J. Yvonnet, Topology optimization for maximizing the fracture resistance of quasi-brittle composites. *Comput. Methods Appl. Mech. Eng.* **332**, 234–254 (2018).

70. J. B. Russ, H. Waisman, Topology optimization for brittle fracture resistance. *Comput. Methods Appl. Mech. Eng.* **347**, 238–263 (2019).
71. Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, F. Durand, DiffTaichi: Differentiable programming for physical simulation. arXiv:1910.00935 [cs. LG] (2019).
72. L. Xia, F. Fritzen, P. Breitkopf, Evolutionary topology optimization of elastoplastic structures. *Struct. Multidiscipl. Optim.* **55**, 569–581 (2017).
73. Y. Mao, Q. He, X. Zhao, Designing complex architected materials with generative adversarial networks. *Sci. Adv.* **6**, eaaz4169 (2020).