Supplementary Information

Morphable Architected Materials with Topologically Variable and Volumetric Reconfiguration

Kai Xiao[1], Yuhao Wang[1], Chao Song[2], Bihui Zou[1], Zihe Liang[1], Heeseung Han1, Qinyun Ding1, Yilin Du3, Shane Johnson[1], Arup Neogi[4], Jaehyung Ju[1]*

1. Global College, Shanghai Jiao Tong University, Shanghai 200240, Shanghai, China.
2. School of Engineering, Westlake University, Hangzhou 310030, Zhejiang, China
3. Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, California 90007, USA
4. Institute of Fundamental and Frontier Science, University of Electronic Science and Technology of China, Chengdu 611731, China

*Corresponding author: Jaehyung Ju (jaehyung.ju@sjtu.edu.cn)

Table of Contents

S1. Non-Collision Condition to Constrain the Overlapping Scalar

Non-Collision Constraints Between Two Triangular Faces

We introduce a method to ascertain the non-collision condition of two triangular faces in 3D space by employing the Separating Axis Theorem (SAT)45. The SAT is utilized to detect intersections between convex polygons by projecting them onto a set of potential separating axes. The absence of overlap along any of these axes confirms that the polygons do not intersect.

Given two triangular faces with vertices $A, B, C,$ and $P, Q, R,$ as shown in **Fig. S1**, we proceed by calculating the face normal vectors n_1 and n_2 for each triangle:

$$n_1 = (v_B - v_A) \times (v_C - v_A), \quad (S1)$$

$$n_2 = (v_P - v_R) \times (v_Q - v_R). \quad (S2)$$

The edge vectors for both triangles are given as $e_A = v_B - v_A$, $e_B = v_C - v_B$, $e_C = v_A - v_C$, $e_P = v_Q - v_P$, $e_Q = v_R - v_Q$, and $e_R = v_P - v_R$.

Now, we calculate the cross products of all possible edge pairs:

$$p_{axis} = \begin{bmatrix} e_A \times e_P \\ e_A \times e_Q \\ e_A \times e_R \\ e_B \times e_P \\ e_B \times e_Q \\ e_C \times e_R \\ e_A \times e_P \\ e_B \times e_Q \\ e_C \times e_R \\ n_1 \\ n_2 \end{bmatrix}. \quad (S3)$$

Given a set of potential separating axes and two triangles with vertices $A, B, C,$ and P, Q, R, we iterate through the k-th axis in p_{axis} (abbreviated as $p_{a.}$) and perform the following steps:

- The vertices of both triangles are projected onto the k-th axis. The projection of a vertex from triangles onto the k-th axis is given by the dot product $v \cdot p_{a._k}$
- Compute the 1D range (minimum and maximum) for each triangle on the k-th axis.

$$R_{\min 1_k} = \min\left(v_A \cdot p_{a._k}, v_B \cdot p_{a._k}, v_C \cdot p_{a._k}\right) \quad (S4)$$

$$R_{\max 1_k} = \max\left(v_A \cdot p_{a._k}, v_B \cdot p_{a._k}, v_C \cdot p_{a._k}\right) \quad (S5)$$

$$R_{\min 2_k} = \min\left(v_P \cdot p_{a._k}, v_Q \cdot p_{a._k}, v_R \cdot p_{a._k}\right) \quad (S6)$$

$$R_{\max 2_k} = \max\left(v_P \cdot p_{a._k}, v_Q \cdot p_{a._k}, v_R \cdot p_{a._k}\right) \quad (S7)$$

Next, we study the separation along the current k-th axis. If $R_{\max 1_k} < R_{\min 2_k}$ or $R_{\max 2_k} < R_{\min 1_k}$, a gap exists along the k-th axis, indicating that the triangles do not intersect. If no separation is found for any axis along p_{axis}, the triangles intersect.

Particularly, we define the overlap scalar along each k-th axis as:

$$O_k = \min\left(R_{\max 1_k} - R_{\min 2_k}, R_{\max 2_k} - R_{\min 1_k}\right). \quad (S8)$$

Notably, if the triangles do not intersect along the k-th axis, the O_k value is negative.

Therefore, the non-collision condition of the two triangles involves finding the s-th axis with the maximum value of O_s and requiring it to be negative.

$$s = argmax\,(O_k) \text{ for } k = 1,2,\ldots \quad (S9)$$

$$O_s < 0 \quad (S10)$$

This method systematically identifies potential non-collision scenarios between two triangular faces in a 3D space, ensuring accurate and efficient geometric modeling.

Selection of Triangular Pairs in an Assembly

To efficiently implement the non-collision condition for two triangles within an assembly, which typically comprises numerous triangular faces, we adopted a straightforward strategy based on the distance between centroids:

Let c_t be the position vector of the centroid of the t-th triangular face of the assembly. We then search the ξ number of nearest centroids of other triangular faces in the architected materials using MATLAB's 'knnsearch' tool. We denote the collection of these nearest faces as $\{t_1, t_2,\ldots, t_\xi\}$ for the current t-th triangle. The combination of choosing two triangles from this collection can be expressed as $\phi_t = C(\{t_1, t_2,\ldots, t_\xi\},2)$.

For example, if $\xi = 3$, we can write $\phi_t = \begin{bmatrix} t_1 & t_2 \\ t_1 & t_3 \\ t_2 & t_3 \end{bmatrix}$.

For an architected material consisting of a total T number of triangles, we iterate the above procedure to search the nearest centroids for every triangle and assemble all the triangles pairs as $\Phi = [\phi_1;\phi_2;\ldots;\phi_T]$. The triangular pairs selected to avoid collisions correspond to the unique rows of Φ. If Φ^u is a unique row of Φ, then the form of Φ^u is expressed as:

$$\Phi^u = \begin{bmatrix} \Phi^u{}_{1_1} & \Phi^u{}_{1_2} \\ \ldots & \\ \Phi^u{}_{I_1} & \Phi^u{}_{I_2} \end{bmatrix}. \#(S11)$$

The i-th row of Φ^u is $\left[\Phi^u{}_{i_1}\ \Phi^u{}_{i_2}\right]$, associating the $\Phi^u{}_{i_1}$-th and $\Phi^u{}_{i_2}$-th faces in the architected material to check collision.

This selection method streamlines the process of checking for collisions among multiple triangles in a complex assembly, focusing computational resources on the most probable interactions and significantly improving the efficiency of the design validation process.

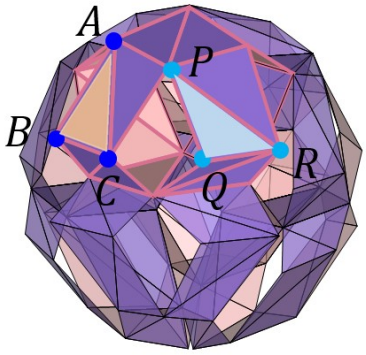

Fig. S1. Discretized Sphere with Non-uniformly Tessellated Unit Cells. This figure illustrates a sphere after volumetric mapping for discretization, showcasing an arrangement of non-uniformly tessellated unit cells. It highlights two specific triangles with vertices labeled A, B, C, and P, Q, R, demonstrating the implementation of non-collision conditions to ensure these triangles do not intersect.

S2. Topologically Variable Morphing with Miura-Origami Units

The inverse design principles for topologically variable morphing, as detailed in Equations (1)–(4) of the main text, have been successfully extended to other types of unit cells, including stacked Miura origami unit cells, configured in a $3 \times 3 \times 3$ system size. As demonstrated in **Fig. S2**, flat-foldable structures have been effectively constructed using the volumetric mapping of Miura origami unit cells. This process adheres to specified kinematic and kinetic constraints, ensuring that the morphing capabilities of the structures are both functional and predictable.

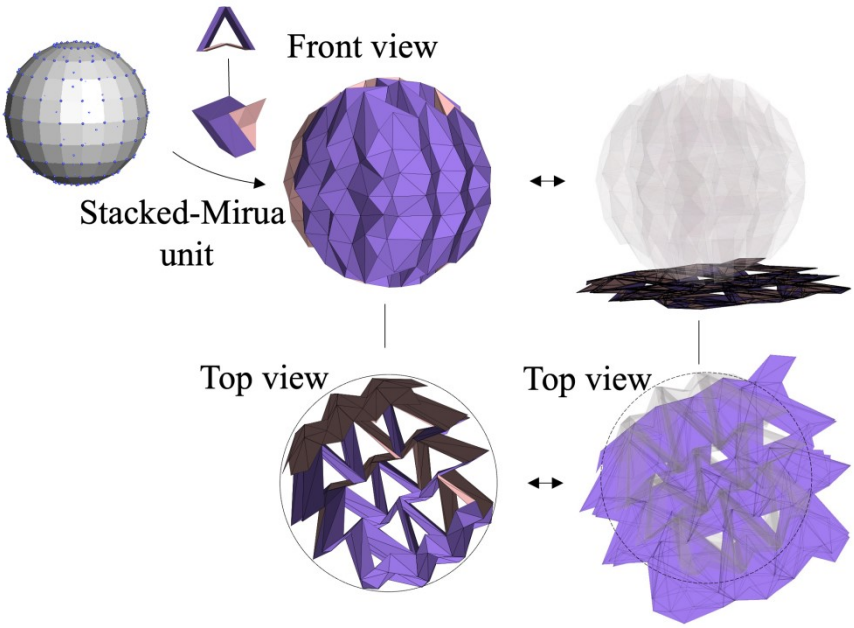

Fig. S2. Flat-Foldable Morphing of a Sphere with Stacked Miura Origami Unit Cells. This figure illustrates the transformation of a spherical structure into a flat-foldable form using an arrangement of Miura origami unit cells.

S3. Volumetric Mapping

This section details a discretization strategy for volumetric mapping aimed at various target morphing shapes, including cuboids, cylinders, and spheres.

Orthogonal Tessellation

Fig. S3 illustrates the procedure for mapping unit cells onto a convex shape. For convex target shapes, such as a sphere or cone, as illustrated in Fig. 2 of the main text, unit cells are volumetrically mapped using semi-discretized optimal transport techniques implemented by the open-source software Graphite44.

We developed a custom MATLAB script to generate an STL file for the discretized origami assembly (**Fig. S3a**). This assembly was then solidified by converting each unit into a convex hull, resulting in a convex hull assembly that covers only the external surface mesh (**Fig. S3b**).

Additionally, we prepared another STL file for the target shape (**Fig. S3d**) and aligned and rescaled these two STL files using Graphite44. The software adjusted the sharp edges and vertices of the STL file for the origami assembly (**Fig. S3c**) and transported it to match the target shape. The final transported mesh was then exported, as shown in Fig. S3e. The target shape was restored using the original vertex and face data from the origami assembly, thus completing the initial discretization of the convex target shape (**Fig. S3f**).

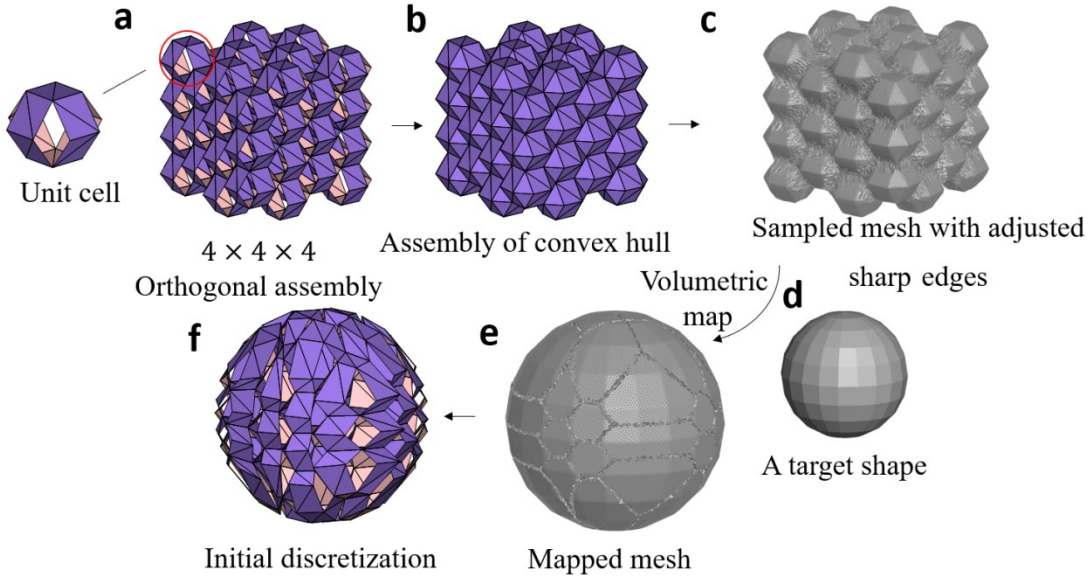

Fig. S3. Procedure of Mapping Unit Cells to a Convex Shape – Sphere. This figure demonstrates the step-by-step process used to map unit cells onto a spherical target shape, highlighting each stage of the volumetric mapping and mesh transformation.

If the target shape includes a concave form, such as an airplane, sculpture, or piece of furniture, as illustrated in Fig. 7 of the main text, an additional step is required to trim the units at the boundaries of the target shape prior to volumetric mapping. Fig. S4 demonstrates the procedure for mapping cells to a hollow shape. Initially, a periodically tessellated origami assembly is aligned to fill a hollow shape. To determine which units extend beyond the boundary, we calculate the normal (^{n_i}) of the nearest face, with the centroid (^{p_i}) on the target shape for each vertex (^{v_i}) of the unit using the `knnsearch` tool in MATLAB. The condition $(v_i - p_i) \cdot n_i > 0$ indicates that the i-th vertex ^{v_i} of the unit is outside the boundary. If all the nodes of a unit are found to be outside the boundary, the unit is removed before the assembly is mapped to the target shape. Further details about the discretization of a concave shape are provided in Fig. S4.

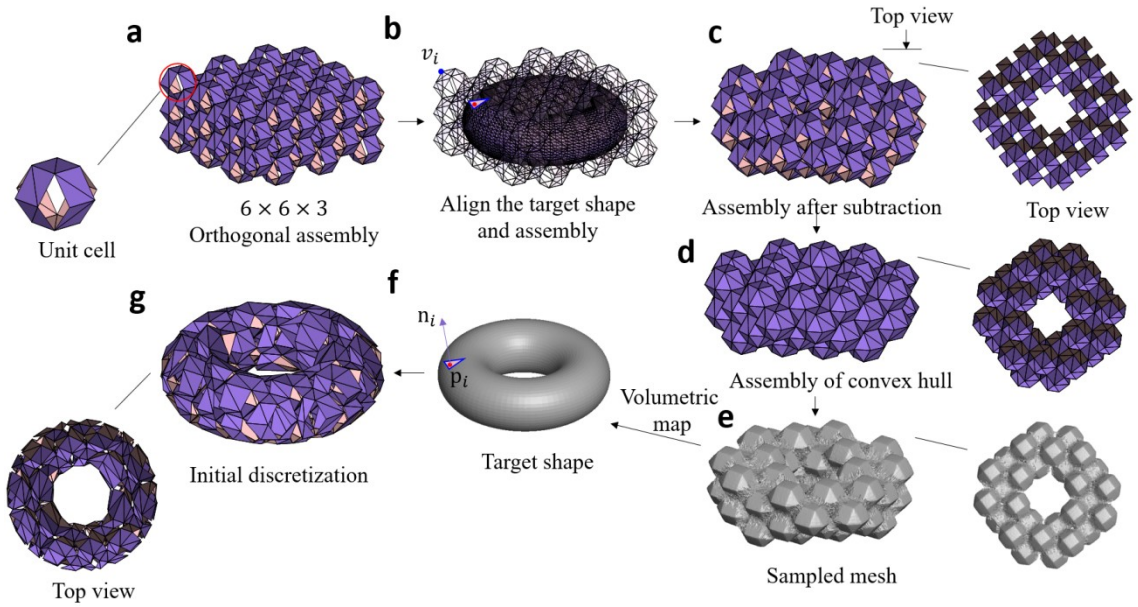

Fig. S4. Procedure of Mapping Unit Cells to a Hollow Shape – Torus. This figure details the step-by-step process used to map unit cells onto a torus, illustrating how units that extend beyond the boundary are identified and removed to achieve precise alignment with the target shape.

Cylindrical and Spherical Tessellation

To arrange modular origami units in a cylindrical configuration, we applied a simple coordinate transformation to the vertices $v = [v_x, v_y, v_z]$ of the orthogonal tessellation:

$$v_c = [R_c \cdot \cos\theta \quad R_c \cdot \sin\theta \quad v_y], \qquad (S12)$$

where $\theta = 2\pi \cdot \frac{v_x}{|L_x|}$, $|L_x|$ is the maximum length of the orthogonal assembly along the direction of lattice vector l_x, and $R_c = v_z$ (Fig. S5).

Similarly, we use

$$v_s = [R_s \cdot \cos\alpha \cdot \cos\beta \quad R_s \cdot \cos\alpha \cdot \sin\beta \quad R_s \cdot \sin\alpha], \qquad (S13)$$

where $\alpha = 2\pi \cdot \frac{v_y}{|L_y|}, \beta = \pi \cdot \left(\frac{v_x}{|L_x|} - \frac{1}{2}\right)$, and $R_S = v_z$.

By leveraging the two deployed states of unit cell ' b' as introduced in the main text, we implemented transformations from orthogonal tessellations to cylindrical and spherical shapes, as illustrated in Fig. S5a and Fig. S5b. These transformations followed the same techniques used in the orthogonal tessellation section to discretize the target shapes further.

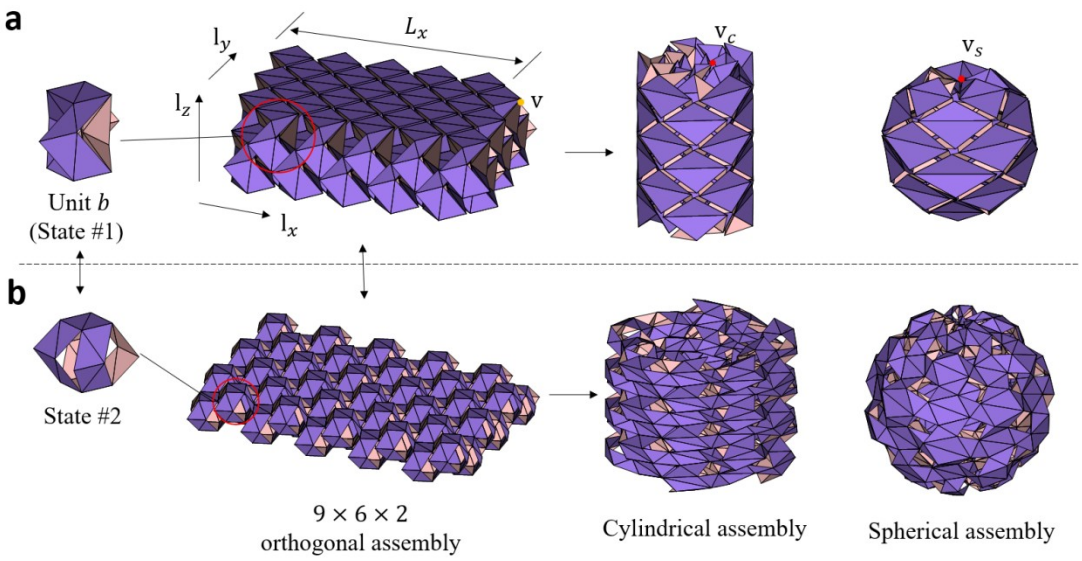

Fig. S5. Cylindrical and Spherical Tessellation: (a) Depicts unit cell 'b' in state #1 and its corresponding orthogonal, cylindrical, and spherical assemblies. (b) Shows unit cell 'b' in state #2 and its transformations into orthogonal, cylindrical, and spherical assemblies.

Fig. S6 illustrates Poisson's ratios of architected materials with various target shapes—cylinders and spheres—and differing unit cell designs—flat-foldable and curvilinear. The designs using flat-foldable units for topologically variable morphing exhibit a wide range of Poisson's ratios $(-100 < \nu < 100)$, due to the intrinsic axial-lateral mechanical coupling[65,66] that comes with flat foldability, as demonstrated in Fig. S6a and Fig. S6b. In contrast, designs using curvilinear units display narrower ranges of Poisson's ratios $(-5 < \nu < 5)$, resulting from the loss of flat foldability while maintaining volumetric expansion along azimuthal and polar directions. Compared to metamaterials discussed in existing literature, the architected materials in this study occupy a substantial design space of Poisson's ratios with volumetric change, as depicted in the charts in Fig. S6b.

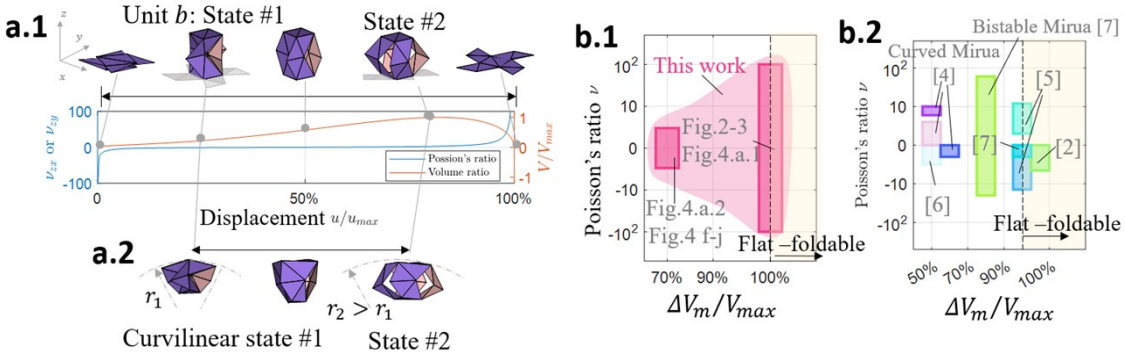

Fig. S6. Poisson's ratio of Architected Materials. (a.1) A periodic unit $'b'$ with changing Poisson's ratio and volume V. V_{max} represents the maximum effective volume that the architected materials can occupy during folding; (a.2) A curvilinear unit 'b' used for spherical or cylindrical tessellation; (b.1) The range of tunable Poisson's ratios of the architected materials in this study; (b.2)Compares these with existing 3D origami metamaterials from references [67-71]. ΔV_m is the maximum volume change of the architected materials during folding. $\Delta V_m/V_{max} = 100\%$ indicates flat-foldability.

S4. Design Map of Shape-lockable Architected Materials

The shape-locking design principles detailed in Equation (5) of the main text have been applied to various morphing designs, encompassing different target shapes and system sizes. **Fig. S7** illustrates the flat-foldable morphing of a sphere, hyperboloid, and cone within a $3 \times 3 \times 3$ system size, along with phase maps showing the ratio of E_{Max}/E_{Min} for different unit geometries.

In general, the shape-locking capability is maintained as the number of units increases, although the morphing behavior exhibits slight variations in geometric parameters. This is illustrated in **Fig. S7a**, which presents actuation energy maps for a sphere using a 3 × 3 × 3 assembly. These results are compared to those of the 2 × 2 × 2 system discussed in Fig. 3g of the main text.

The diversity of target shapes influences the topology of the design map. Although the range of E_{Max}/E_{Min} values and the principle of identifying lockable structures are consistent with those described in Fig. 3g of the main text, the lockable structures, characterized by a higher E_{Max}/E_{Min} are readily identifiable in the region where $\omega < 0$ and $|\delta|$ shows elevated values.

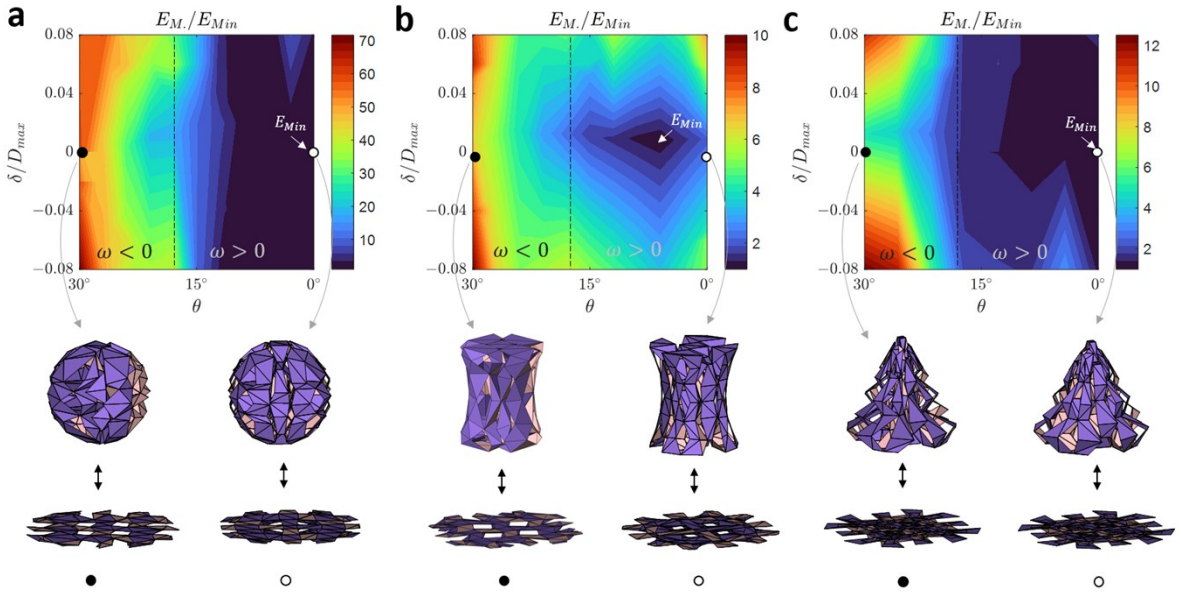

Fig. S7. Actuation Energy Maps for Flat Foldable and Shape Lockable Morphing of 3D Curvilinear Architected Materials. This figure shows that actuation energy maps for architected materials with a 3×3×3 system size, where E_{Max}denotes the maximum stored energy during folding, and E_{Min} represents the minimum stored energy necessary for morphing within the design map. Design map detailing the actuation energy levels associated with flat-foldable morphing; (a) Sphere assembly showing energy distribution during morphing; (b)Hyperboloid assembly with highlighted energy aspects; (c) Cone assembly demonstrating energy requirements for morphing.

S5. Settings for FE Simulation and Homogenization to Obtain Effective Modulus

We employed the MERLIN2 FE software to discretize origami-inspired architected materials into bar structures, which were then used to perform FE simulations under conditions of large deformation. The material details were defined using the N4B5 model, which was applied to discretize the faces of the architected materials and assign stiffness parameters to both the bar elements and rotational springs. Displacement control was utilized for the simulations. Detailed parameters and boundary conditions used to flatten the architected materials with multiple-point loading are available in the "Bistability" folder within the open-source repository.

Using the MERLIN2 FE software, we also evaluated the static responses of various configurations—both compact and expanded—of the architected materials under small-deformation conditions with multiple-point loading. The results, shown in **Fig. S8a-c**, provide values for the effective bulk and shear moduli for architected materials designed with a spherical boundary.

$$B_S = -V_s \cdot \frac{dP_s}{dV_s} = -\frac{1}{12\pi R} \cdot \frac{d^2E_s}{ds^2} \qquad (S14)$$

where B_S denotes the effective bulk modulus under pressure P_s, implemented as a multi-point load on the vertices of architected materials along the spherical boundary with radius R, and dV_s represents the infinitesimal volume change at the current configuration, E_s indicates the stored elastic energy, and s represents the displacement along the radial direction.

For architected materials defined by a cubic boundary, as illustrated in **Fig. S8d**, the bulk modulus B_c is calculated using the following equation:

$$B_c = -V_c \cdot \frac{dP_c}{dV_c} = -\frac{1}{9L} \cdot \frac{d^2E_c}{ds^2} \qquad (S15)$$

Here, the terms $\frac{d^2E_s}{ds^2}$ and $\frac{d^2E_c}{ds^2}$ are calculated numerically using the MERLIN2 FE software.

Additionally, the effective shear modulus G for certain configurations of the architected materials is determined by dividing the shear stress τ, applied via multiple-point loads, by the resulting shear strain γ. These configurations are displayed in **Fig. S8e-h**. This method, implemented in MERLIN2, allows for precise calculation of G under varying load conditions and configurations:

$$G = \frac{\tau}{\gamma} \qquad (S16)$$

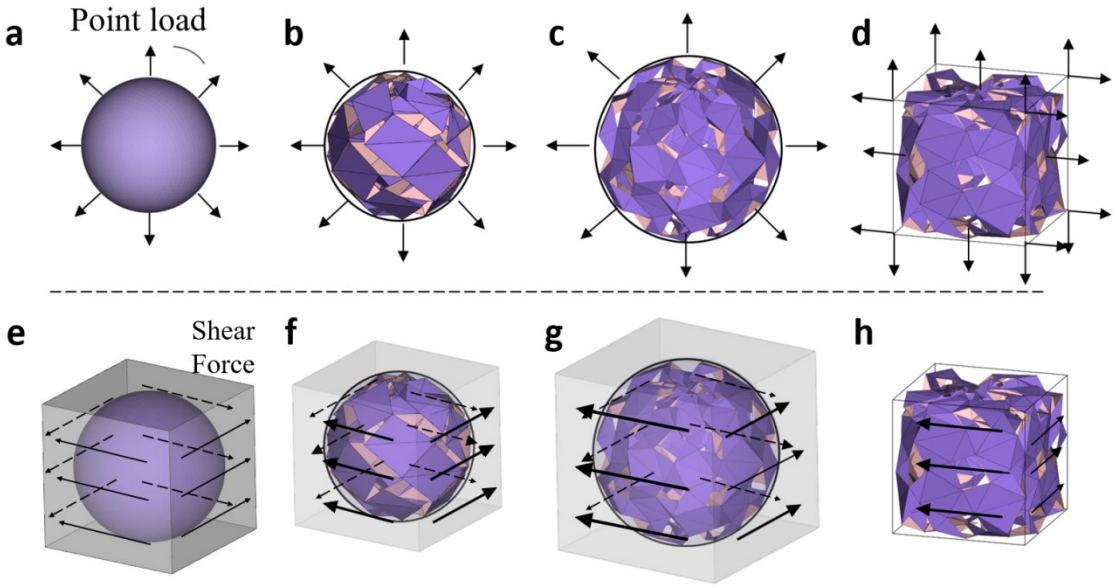

Fig. S8. Schematics of Multiple-Points Loading Conditions for Obtaining the Linear Response of Architected Materials. Illustrations depict the loading scenarios that evaluate the linear mechanical responses of both compact and expanded configurations of spherical and cubic architected materials.

S6. Boundary Conditions and Materials Properties in MERLIN 2 FE Simulations

Boundary Conditions:

1. We employed a displacement-controlled loading scheme, as specified in our MERLIN2 code ('LoadType','Displacement'). This approach offers enhanced stability, particularly for highly nonlinear morphing scenarios.
2. Boundary nodes were carefully selected to ensure accurate transformation between morphing states. For instance, in the spherical morphing case, we applied displacement boundary conditions to specific control nodes (e.g., nodes 39, 38, 13, 12, etc.).
3. Displacement vectors were computed as the difference between the flat and expanded configurations, enabling precise control over the morphing process.
4. Zero-displacement support conditions were applied where necessary to suppress rigid body motion while preserving the desired morphing response.

Material Properties:

1. The bar elements followed an Ogden hyperelastic constitutive model with a modulus of 2×10^{3} units ('BarCM', @(Ex)Ogden(Ex, 2e3)).
2. The cross-sectional area of the bars was set to 2×10^{-1} ('Abar', 2e-1).
3. Rotational springs were incorporated with distinct stiffness values for bending (Kb = 0.3) and folding (Kf = 0.006).
4. We used an enhanced linear model for both bending and folding springs, with carefully tuned parameters to achieve appropriate bistable responses.

These boundary conditions and material models were validated against our experimental results, producing force-displacement curves that reasonably matched the bistable behavior observed in physical prototypes. Furthermore, we have made our FE implementation publicly available on https://github.com/KaiXiao55/3DMorphingArchitectedMaterials.

S7. Codes and data

We have made the source data and code available in a repository to facilitate the reproduction of the figures presented in the main text, accessible via Ref. 2. The contents are organized as follows:

- Folder #1 - Topologically Variable Morphing Plot: Contains the code and source data necessary to generate Fig. 1, which plots the design space for morphing structured materials.

- Folder #2 - Inverse Design Results: Includes data such as nodes and faces for spherically, hyperboloidally, and conically morphing architected materials, as shown in Figs. 2f–h.

- Folder #3 - Bistability: Provides numerical evidence of shape-lockable architected materials, which can be analyzed through the force-displacement curves generated by the "Merlin2" software.

- Folder #4 - Ashby Chart: Contains the source code required to reproduce Fig. 6k.

- Folder #5: Offers the code necessary to convert zero-thickness architectural materials into 3D-printable versions, applicable in flat and expanded states.

References

1 Ye, H., Liu, Q., Cheng, J., Li, H., Jian, B., Wang, R., Sun, Z., Lu, Y. & Ge, Q. Multimaterial 3D printed self-locking thick-panel origami metamaterials. *Nature Communications* **14**, 1607 (2023).

2 *https://github.com/KaiXiao55/3DMorphingArchitectedMaterials*