

One String to Pull Them All: Fast Assembly of Curved Structures from Flat Auxetic Linkages

AKIB ZAMAN, MIT CSAIL, USA

JACQUELINE ASLARUS, MIT CSAIL, USA

JIAJI LI, MIT CSAIL, USA

STEFANIE MUELLER, MIT CSAIL, USA

MINA KONAKOVIĆ LUKOVIĆ, MIT CSAIL, USA

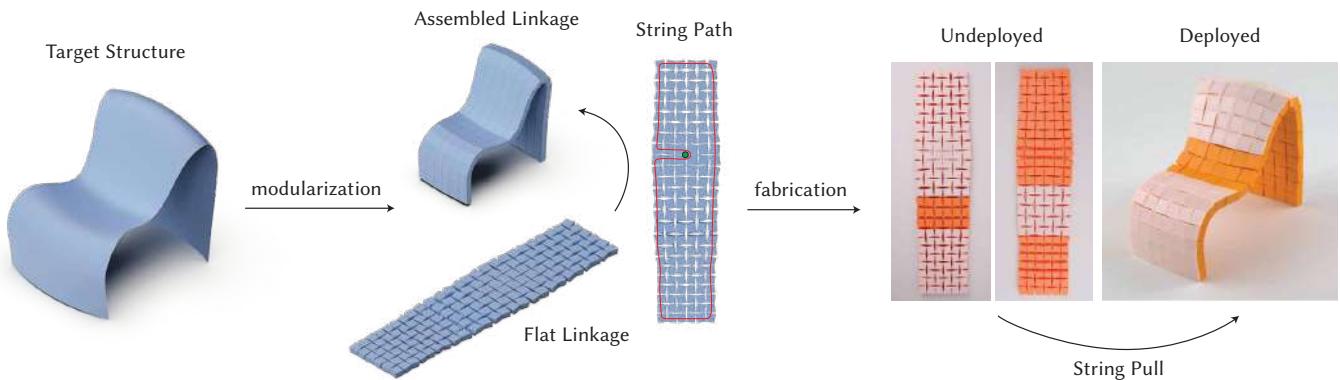


Fig. 1. Our method approximates a target structure with rigid quad tiles, forming a flat mechanical linkage. A string is then routed through an optimized subset of linkage elements. With a single pull of this string, the linkage assembles into a final chair configuration.

We present a computational approach for designing freeform structures that can be rapidly assembled from initially flat configurations by a single string pull. The target structures are decomposed into rigid spatially varied quad tiles that are optimized to approximate the user-provided surface, forming a flat mechanical linkage. Our algorithm then uses a two-step method to find a physically realizable string path that controls only a subset of tiles to smoothly actuate the structure from flat to assembled configuration. We initially compute the minimal subset of tiles that are required to be controlled with the string considering the geometry of the structure and interaction among the tiles. We then find a valid string path through these tiles that minimizes friction, which will assemble the flat linkage into the target 3D structure upon tightening a single string. The resulting designs can be easily manufactured with computational fabrication techniques such as 3D printing, CNC milling, molding, etc. in flat configuration that, in addition to manufacturing, facilitates storage and transportation. We validate our approach by developing a series of physical prototypes and showcasing various application case studies, ranging from medical devices, space shelters, to architectural designs.

Authors' Contact Information: Akib Zaman, MIT CSAIL, Cambridge, MA, USA, akib@mit.edu; Jacqueline Aslarus, MIT CSAIL, Cambridge, MA, USA, aslarus@mit.edu; Jiaji Li, MIT CSAIL, Cambridge, MA, USA, jiaji@mit.edu; Stefanie Mueller, MIT CSAIL, Cambridge, MA, USA, stefmue@mit.edu; Mina Konaković Luković, MIT CSAIL, Cambridge, MA, USA, minakl@mit.edu.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2025 Copyright held by the owner/author(s).
ACM 1557-7368/2025/12-ART
<https://doi.org/10.1145/3763357>

CCS Concepts: • Computing methodologies → Shape modeling; Mesh geometry models; • Applied computing → Computer-aided manufacturing.

Additional Key Words and Phrases: Computational fabrication, string-based actuation, metamaterials, inverse design, deployable structures

ACM Reference Format:

Akib Zaman, Jacqueline Aslarus, Jiaji Li, Stefanie Mueller, and Mina Konaković Luković. 2025. One String to Pull Them All: Fast Assembly of Curved Structures from Flat Auxetic Linkages. *ACM Trans. Graph.* 44, 6 (December 2025), 18 pages. <https://doi.org/10.1145/3763357>

1 Introduction

The construction of deployable structures from flat modular pieces simplifies both fabrication and assembly on site while reducing the transportation and storage requirements. By combining rigidity and flexibility, these structures find applications in fields ranging from architecture [Panetta et al. 2019] to electronics in robotics [Chen et al. 2024; Liu et al. 2024] and biomechanical devices [Wang et al. 2023b]. Despite this prominent progress, achieving freeform deployables that integrate both modularization and effective actuation remains an open challenge.

Deployables have recently gained significant traction in computational design and fabrication, yet existing actuation strategies struggle to achieve reversibility, scalability, and reliable support for shell-like, doubly curved geometries. Some of the existing approaches exploring reversible and scalable strategies face limitations while approximating geometries with variation of sign in the mean curvature (e.g., negative mean curvature [Konaković-Luković et al.

2018]), and they often require specialized equipment to actuate or risk potential jamming [Eguchi et al. 2022]. A method that unifies surface rationalization with optimized actuation enables structures to transition from flat to freeform configurations and allows fast, repeatable deployment as well as disassembly. This capability opens new opportunities across domains and scales, ranging from novel designs of compact, transportable biomedical devices, foldable robots, and consumer items (such as chairs or bike helmets) to large-scale portable space habitats and rapidly deployable emergency shelters or field hospitals during floods or earthquakes.

We introduce a new method for deploying a structure from the *flat* configuration to the *assembled* curved configuration with only a single pull of a string (see Figure 1). Our approach is based on the planar linkage of spatially varying quadrilateral tiles which are connected by rotational hinges at the vertices. The string then passes through only a small subset of tiles and lifts the structure simply by tightening it. This setup allows for reversible actuation: if the string is released, the structure returns to a flat configuration. The structure requires a simple unidirectional pulling mechanism, making it independent of the scale. Several key contributions enable our method:

- We approximate a user-provided structure by modularizing it into scale-varying quadrilateral frustums and derive a flat configuration by computing hinge placements that respect geometric curvature, forming a flat mechanical linkage.
- We present a general analysis showing that closing the boundary gaps alone suffices to guide all linkage tiles into the desired 3D configuration.
- We identify the minimum set of lift points for assembling the final structure and optimize the string routing to minimize channel friction. This effectively reduces the total resistive force during actuation of the structure from a flat to assembled configuration.
- Our method supports *reversible actuation* on a wide scale range, as demonstrated by proof-of-concept prototypes across various domains and scale. We show examples lifted by human hand, robotic arms, and architectural cranes.

Our method offers several key advantages beyond its ability to handle both flat fabrication and complex curvature. Since all tiles remain fully enclosed and connected through a single low-friction string path, the structure is simple to deploy, disassemble, and reuse. By eliminating the need for additional scaffolding or sealed layers, users also benefit from lower material costs and faster production times, an attribute crucial for emergency shelters, large-scale temporary architectural installations, and consumer products alike. We demonstrate potential applications throughout this work, ranging from compact medical wearables (e.g., a foldable backrest and foot splint, see Figure 2) to consumer items (e.g., foldable chair, see Figure 12), crane-lifted architectural structures (see Figure 17), and transportable space habitats (see Figure 16). Our method also has the potential to facilitate robotic assembly by replacing computationally-intensive and challenging assembly including individual block detection and motion planning of each block with the far simpler task of detecting and pulling a single string (See Figure 15 and Supplementary Video, 01:23).



Fig. 2. Application in personalized medical items such as splints (Left) and posture correctors (Right). Both are optimized for a given patient, then manufactured, stored and carried flat, and can be easily assembled by a single string pull.

The remainder of this paper is organized as follows. Section 2 discusses relevant papers and their connection to our work. Section 3 provides an outline of our method. Section 4 provides the design framework for the flat configuration and the placement of the hinges. Section 5 describes the string route finder algorithm. Section 6 discusses fabrication of the prototypes and application case studies. Section 7 highlights the limitations of our approach, and identifies potential future research directions.

2 Related Work

Deployable structures. In recent years, the design of fast deployable structures has received significant attention driven by applications from emergency shelters to space exploration. [Zheng et al. 2016] design deployable rigid scissor linkages that approximate 3D models while ensuring a collision-free expansion path. Subsequently, beam-based grid shells have been used to quickly deploy doubly-curved structures made from planar linkages [Becker et al. 2023; Panetta et al. 2019; Suzuki et al. 2023]. [Ren et al. 2022] propose Umbrella Meshes, an intricate design that can form any surface at the cost of a complex fabrication and assembly process. More recently, 3D printing for exploring rapid fabrication [Noma et al. 2020] and machine learning for exploring the design space of soft deployables [Ma et al. 2024] have been studied. While many of these systems expand to create an open deployed state, our linkage contracts to produce a solid final structure, eliminating the need for an additional material to enclose the final structure and provide a barrier between inside and outside.

Flat-Fabricable Structures. Decomposing a curved shape into flat fabricable pieces is of particular interest in deployables, as it simplifies manufacturing and transportation while enabling assembly into complex curved forms. [Malomo et al. 2018] embed flat panels with a flexible spiral microstructure that deforms into a target 3D shape when connected. Other works approximate surfaces with developable patches, yielding pieces suitable for flat fabrication [Bahraini et al. 2025; Zhao et al. 2023]. However, assembly in these methods requires handling disjoint pieces in a precise order, demanding substantial manual effort. By using linkage-based auxetics, we preserve global connectivity in flat modules, remove the need

for individual unit handling, and generate a string path for rapid assembly of flat-fabricated pieces into curved forms.

Deployable auxetics. [Konaković et al. 2016] propose approximating doubly-curved geometries with linkage-based auxetics, which use kirigami cut patterns to tessellate a planar surface into unit cells that rotate about each other at hinge points. By spatially-grading the unit cells, auxetic linkages encode a unique shape under maximal extension [Konaković-Luković et al. 2018]. This mechanism eliminates complexities in assembly that arise from handling many disconnected tiles, and it lends itself to both additive and subtractive manufacturing methods. These previous works serve as a motivation for our method, which also uses spatially-graded auxetics to encode a target 3D geometry. Further studies have explored kirigami cut patterns that exhibit bistability, which additionally ensure that the 3D state remains stable [Chen et al. 2021; Jiang et al. 2022; Rafsanjani and Pasini 2016].

Quad meshes for design. Quad meshes are popular in architectural geometry since they offer easier fabrication enabling flat cutting of glass or metal panels without complex molds. However, maintaining planarity in physical quad-mesh structures has always been challenging. Earlier works from the graphics community proposed several strategies to preserve planarity, such as conjugate curve networks [Liu et al. 2006], parallel meshes to reduce torsion [Pottmann et al. 2007], and node-angle control for urban structures and transformable shades [Jiang et al. 2020b]. Recent work [Jiang et al. 2024] proposed a quad-mesh mechanism by connecting rigid quads with edge hinges for transformable configurations, but these can disrupt flat-to-lifted transitions by altering geodesics. In contrast, vertex-based hinges can preserve infinitesimal motion between configurations.

Quad kirigami-based auxetics. Approximating a target structure using quad kirigami-based auxetics has recently gained attention. [Choi et al. 2019] introduce thin-sheet quad kirigami-based auxetics for approximating 2D and 3D shapes from a flat state, later extending their approach for compact, rigid-deployable structures that can reconfigure between planar or tubular forms [Choi et al. 2021]. [Jiang et al. 2022] model rectangular kirigami auxetics under constant mean stretch, and further integrate kirigami cuts and fold lines to create curved box-based shapes that fully unfold [Jiang et al. 2020a]. [Dudte et al. 2023] employ a linear algebra-based method for designing quad-auxetic morphing structures, without detailed actuation methods. [Wang et al. 2023a] demonstrate a reconfigurable quad kirigami structure actuated via embedded magnetic particles but did not target 3D shape approximation, restricting its broader applicability. In this work, we utilize quad-based kirigami to create a quadrilateral grid-like pattern for the kirigami cuts, enabling a more direct string path than triangular Kagome lattices. Unlike previous works, we focus on extruded quadrilateral tiles. Finally, using the quad kirigami pattern as the basis of structural programming, we determine the optimal string path to minimize friction.

Actuation mechanisms for 2D to 3D shape morphing. A key aspect of designing shape-morphing materials is selecting an effective actuation mechanism. Heat has been widely explored [McBride et al. 2018; Narumi et al. 2023; Nojoomi et al. 2021] but often requires

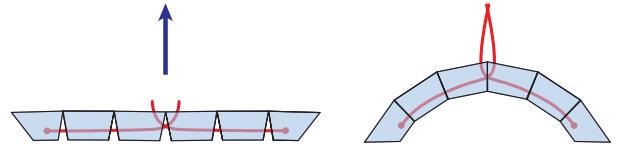


Fig. 3. Illustration of string actuation and tile assembly (side view).

specialized manufacturing techniques and materials that limit its scale. Various works [Gu et al. 2019, 2020; Jourdan et al. 2023; Koh et al. 2023; Wang et al. 2018; Yuan et al. 2024] use thermoshrinkage to create low-cost 3D structures, but they rely on 3D-printing techniques intrinsic to that process, limiting their applications. Actuation via pre-stretched elastic materials can achieve impressive auto-assembly, but the planar state is inherently unstable, complicating flat transport and storage [Guseinov et al. 2017; Jourdan et al. 2022; Ma et al. 2024]. This challenge is addressed in [Guseinov et al. 2020], where elastic sheets are held in place by structures that soften with heat. Many works [He et al. 2024; Panetta et al. 2021; Ren et al. 2024; Siéfert et al. 2019] use inflation, pneumatic actuation, or gravitational loading, which are often well suited for large-scale structures. However, for linkage-based auxetics, inflation and gravitational loading cannot approximate mixed-curvature surfaces [Konaković-Luković et al. 2018], and pneumatic membranes can obstruct large gaps, preventing assembly [Eguchi et al. 2022]. Unlike aforementioned methods, string-pulling offers a scalable, reversible actuation mechanism that is flexible enough to adapt to complex geometries, which inspired our method.

String-based actuation. Actuating 3D structures via string-pulling has been explored in multiple other works. [Kilian et al. 2017] find a network of strings and surface point connections that fold creased sheets into a target curved origami shape when pulled. [Niu et al. 2023] unfold the target shape into a connected set of faces and weave a string through vertices that are joined in the 3D state. [Yang et al. 2024] assemble 3D structures from a flat grid of tiles with an array of fibers passing through each column and row of tiles. Since the tiles are disconnected, actuation via string pulling requires that separate fibers must be threaded through each tile in two directions to fully constrain their motion. With our design, the global connectivity ensures that only a subset of the tiles needs to be controlled to actuate the final 3D shape. Finding an optimal string routing is generally challenging and previous works emphasize reducing friction in the string path as a key objective, either by minimizing total length [Demaine et al. 2023], or total turn cost [Demaine et al. 2024].

3 Method Overview

The goal of our method is to approximate a target surface with modular tiles that can be fabricated flat and pulled with a string into a 3D configuration. A straightforward approach is to decompose the surface into disconnected components, but the string must then traverse long, irregular paths across parts, causing high and unpredictable friction. Additionally, the components may collide in

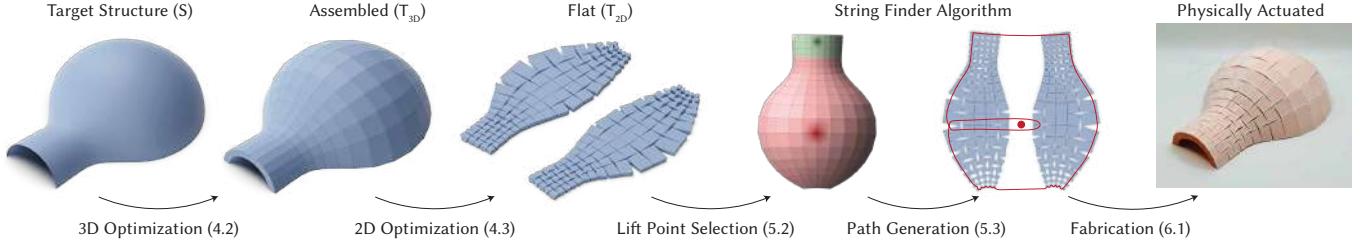


Fig. 4. Design to assembly pipeline. Starting from a freeform target surface, we (1) optimize a quad tiling that encodes the geometry, (2) flatten the tiles with non-overlapping layout and compute the hinges, (3) select a minimal set of lift points, (4) compute a low-friction closed string path (Red line with red dot as selected lift point), and (5) fabricate the planar linkage and pull a single string to assemble it into the target 3D shape.

unpredictable ways, so their movements must be carefully choreographed to avoid deadlocks that prevent assembly. Without regular connections among the components, this is a difficult process to model and control. To overcome these challenges, we leverage auxetic linkages [Jiang et al. 2022; Konaković et al. 2016; Mir et al. 2014], which flatten the freeform geometry while preserving global connectivity. In square quad auxetic linkages, gaps form rhombi that can close in two symmetric ways. Encoding geometry with spatially varying quadrilaterals breaks this symmetry, and each gap now closes in a single direction. By design, closure of all gaps yields the desired assembly of target structure and the design mechanism requires placement of the string through only a subset of tiles.

In Section 4, we first rationalize the surface into flat modular tiles and compute the appropriate connectivity. In Section 5, we then compute a physically realizable string path to drive the transition from flat to assembled through three major components. First, we prove that placing the boundary tiles in their target configuration uniquely determines the geometry of the entire structure and rigidifies the linkage, so we route the string along the boundary first. Second, we select the minimum set of lift points (locations where the string enters and exits the structure, and is pulled, as depicted in Figure 3) to reduce the energy required to assemble the structure. This reduces the number of tiles on the string path and simplifies routing. Finally, we connect the lift points to the boundary, generating the final string route that minimizes friction from directional turns inside the string channel. The overall pipeline (see Figure 4) involving surface rationalization on a regular quad grid, combined with these aforementioned routing components, produces a low-friction path that ensures a smooth deployment of the structure.

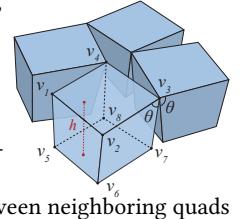
4 Surface Rationalization

First, we design the assembled 3D configuration of a linkage that approximates the target structure by optimizing the geometry of each quadrilateral tile and their dihedral angles (Section 4.2). Referencing this design, we create a corresponding flat configuration (Section 4.3) and compute the hinges at locations that respect local geometry, maintain regular connectivity, and enable complete contact between side faces when deployed (Section 4.4).

4.1 Initialization

Unit Tile. A unit tile in our system is a quadrilateral frustum of a

given height that is defined by eight vertices, forming six quadrilateral faces in total. The top and bottom faces of a tile are always parallel to each other. We illustrate a unit cell (see inset on the right) containing four tiles, where v_i are the locations of the vertices, h is the thickness of the tile, and θ is the dihedral angle in the shared plane between neighboring quads in the assembled configuration. Tile connectivity is inspired by the regular quad auxetic linkage (see Supplement, Figure 18). Each tile can be at most connected to four other tiles, sharing a unique joint with its neighbors. This creates a topological cut forming an empty space among the tiles, which we have defined as a gap. Generally, a gap is a quadrilateral-shaped void between a cell of four tiles. However, for irregular cells along the boundary, a gap can be formed between groups of two or three tiles.



Configurations. Throughout our method, we optimize two different configurations of tiles, *assembled* (denoted with subscript 3D) and *flat* (denoted with subscript 2D). As these describe two different states of the same linkage, there is a one-to-one correspondence between the vertices of the configurations.

Parametrization. To initialize the assembled and flat configurations, we take inspiration from [Konaković et al. 2016] and use conformal mapping to find the correspondence between the 3D configuration of an auxetic linkage and its 2D configuration. We begin with a target surface S and apply the Boundary First Flattening method [Sawhney and Crane 2017] to flatten it into a planar domain $\Omega \subset \mathbb{R}^2$, creating a conformal map $c : S \rightarrow \Omega$. Where possible, the boundary of a conformal map is set to create a rectangular flattened domain Ω (as shown in Fig. 5 for example). We overlay a regular quad grid connected in a mesh on top of Ω and crop away quads that are not fully contained within the boundary of Ω . This cropping determines the number of unit tiles in the linkage and yields the mesh M_{2D} defining the top faces of the tiles in the flat configuration. If the conformal scale factor (CSF) exceeds 2, indicating that a quad auxetic pattern cannot locally expand enough to approximate the target shape, we introduce splits into M_{2D} that enable additional deformation (see Section 4.5). Next, we use the inverse conformal map c^{-1} to lift M_{2D} into the 3D space, producing the mesh M_{3D} that defines the top faces of the tiles in the assembled configuration. When the target model allows, we align the grid with principal

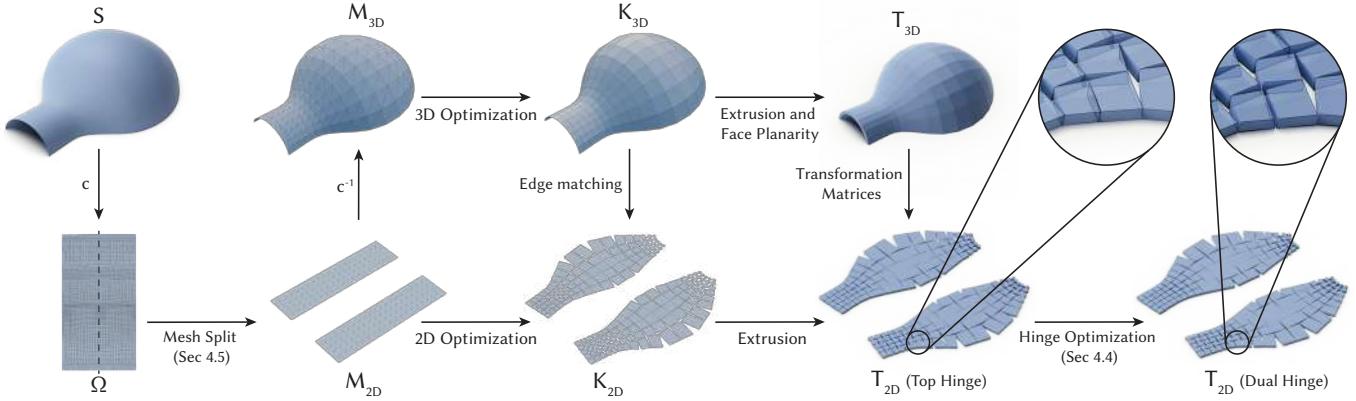


Fig. 5. Surface rationalization pipeline to generate optimized tiles in the assembled T_{3D} and flat T_{2D} configurations.

curvature directions to achieve approximately conical meshes [Liu et al. 2006].

4.2 Assembled Configuration

The conformal lifting of M_{2D} creates M_{3D} , an initial quad approximation of the target surface that corresponds to the top face of the tiles in the auxetic linkage. However, this mapping does not preserve the planarity of the quad faces due to the discrete nature of the problem, which is imperative to create an auxetic linkage with a flat configuration. Simply optimizing the quadrilateral faces for planarity may cause dramatic deviations from the target shape, so we consider keeping the quads as close as possible to the target surface during optimization. Additionally, if the quadrilateral faces deform significantly from squares and create large disparities in size, the auxetic linkage encounters increased resistance when opening or closing [Choi et al. 2021], hindering our primary objective of enabling an easy transition from flat to assembled configuration.

These objectives can be easily formulated using the projection-based approach presented in [Bouaziz et al. 2012, 2014] to obtain the optimized lifted mesh K_{3D} , the optimized quad approximation of the target surface, which corresponds to the final top faces of the tiles. From K_{3D} , we then extrude the faces and, if required, fix planarity to obtain the final mesh T_{3D} , consisting of quadrilateral frustum tiles (see Figure 5). Let E_{Surface} , E_{Square} , and E_{Planar} be the energy functions for the target surface approximation, the squareness of the quadrilaterals, and the planarity of the quadrilaterals, respectively. We minimize the combined energy function $E_{\text{Assembled}}$ over the vertex positions v of the mesh M_{3D} , defined as:

$$E_{\text{Assembled}}(v) = \omega_1 E_{\text{Planar}}(v) + \omega_2 E_{\text{Square}}(v) + \omega_3 E_{\text{Surface}}(v) \quad (1)$$

where ω_1 , ω_2 , and ω_3 are non-negative weighting coefficients that balance the contributions of the respective energy components to the overall objective of minimizing the energy function. Using the projection operator as described in [Bouaziz et al. 2012, 2014], these terms can be expressed as summarized in the following paragraphs.

Planarity Constraint. The energy function E_{Planar} quantifies the deviation from co-planarity among a set of vertices within four

vertices of a quad mesh. Let $Q(M_{3D})$ be the set of all quads in M_{3D} . For each quad having four vertices at coordinates $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_\ell$, we form a 3×4 matrix, $V_q = [\mathbf{v}_i \ \mathbf{v}_j \ \mathbf{v}_k \ \mathbf{v}_\ell]$ and define the planarity energy as:

$$E_{\text{Planar}} = \sum_{q \in Q(M_{3D})} \|V_q - P_p(V_q)\|_F^2 \quad (2)$$

where $P_p(V_q)$ represents the projection of V_q onto its best-fit plane determined by the projection operator P_p .

Surface Constraint. To keep a faithful approximation of the target surface, we introduce the following surface closeness energy term:

$$E_{\text{Surface}} = \sum_{i \in \mathcal{V}(M_{3D})} \|\vec{v}_i - P_S(\vec{v}_i)\|_2^2$$

where \mathcal{V} is the set of all the vertices in M_{3D} , \vec{v}_i denotes a position vector of a vertex in \mathcal{V} , and P_S is the closest point on the target surface S .

Square Constraint. To reduce size disparities among the quads, we set each edge's target length as the average of its two adjacent quads' mean side-lengths and use the similarity constraint to encourage square-like shapes. Let $\mathcal{E}(M_{3D})$ be the set of all edges in the mesh M_{3D} , which includes the four side edges of each quadrilateral. An edge is defined by a pair of vertex indices (i, j) , where \vec{v}_i and \vec{v}_j are the position vectors of the vertices with indices i and j , respectively. For each edge e_{ij} shared by the quads Q_a and Q_b , let \bar{L}_a and \bar{L}_b denote the mean length of the side edge of Q_a and Q_b , respectively. Then, the target length is defined as $L_{\text{Target}}(e_{ij}) = \frac{\bar{L}_a + \bar{L}_b}{2}$. By averaging each edge's target length between its two adjacent quads' mean side-lengths, larger quads shrink and the smaller ones expand, reducing size disparities. Thus, we define

$$E_{\text{Length}} = \sum_{(i,j) \in \mathcal{E}(M_{3D})} \|(\vec{v}_j - \vec{v}_i) - P_E(\vec{v}_i, \vec{v}_j)\|_2^2, \quad (3)$$

where $P_E(\vec{v}_i, \vec{v}_j)$ is a projection operator that generates a target edge vector with the target length L_{target} in the direction of the original edge and is defined as:

$$P_E(\vec{v}_i, \vec{v}_j) = L_{\text{Target}}(e_{ij}) \times \frac{\vec{v}_j - \vec{v}_i}{\|\vec{v}_j - \vec{v}_i\|} \quad (4)$$

Next, to drive each quad toward an exact square shape, we use:

$$E_{\text{Shape}} = \sum_{q \in Q(M_{3D})} \|v_q - P_Q(v_q)\|_2^2,$$

where q is the index set of vertices of quad Q , and P_Q is the projection to the closest square [Umeyama 1991]. Thus, the total square energy, E_{Square} is the sum of E_{Length} and E_{Shape} .

Extrusion ($K_{3D} \rightarrow T_{3D}$). Now we provide thickness to the tiles according to the user's choice in the optimized 3D mesh K_{3D} . When possible, during the parametrization step we align the quad grid with principal curvature directions to obtain near-conical meshes, enabling planar extrusion of faces by offsetting all vertices of K_{3D} mesh along its normals. However, we note that this is not always possible, for example, in Cloud (Figure 10), and Lilium (Figure 11). We then optimize the vertices using Equation 2 to enforce face planarity on top, bottom, and contact faces. This optimization yields the final deployed configuration of linkage tiles T_{3D} . Note that the placement of hinges between these tiles is done as a post-processing step described in Section 4.4.

4.3 Flat Configuration

To obtain the flat layout of the quad auxetic structure, we begin with the clipped mesh M_{2D} , which was used to conformally map to 3D. An isomorphic quad auxetic linkage is encoded in both the flat (M_{2D}) and the optimized lifted (K_{3D}) configurations. We optimize the vertices of M_{2D} to match the edge lengths of K_{3D} , yielding the planar linkage K_{2D} . Subsequently, we extrude K_{2D} matching the thickness of T_{3D} to generate the flat configuration of the tiles, T_{2D} (see Figure 5). During edge matching, it is essential to ensure that all quadrilaterals lie in a non-conflicting configuration without overlaps. We additionally provide the user with the option to control the bounds for the opening angles of the gaps, as different fabrication methods and resources may require more empty space among tiles for effective fabrication. We formulate the 2D optimization to minimize the energy function E_{Flat} , defined as:

$$E_{\text{Flat}}(v) = \omega_1 E_{\text{Edge}}(v) + \omega_2 E_{\text{Collision}}(v) + \omega_3 E_{\text{Fab}}(v) \quad (5)$$

The energy function E_{Edge} quantifies the length mismatch between the corresponding edges in the initial flat (M_{2D}) and assembled non-extruded K_{3D} configurations. We define E_{Edge} using Equation 3 where projection operator P_E is identical to Equation 4 and the target length L_{target} is the corresponding edge length of K_{3D} .

Collision Avoidance. $E_{\text{Collision}}$ quantifies the overlap between quads in M_{2D} and follows the non-penetration constraint introduced by [Konaković et al. 2016] to avoid collision of quad meshes in the flat configuration. We again use the projection-based approach from [Bouaziz et al. 2012] to solve this optimization problem.

Fabrication Clearance. We allow users to set a minimum gap angle θ_{\min} to match fabrication constraints, while the maximum angle is fixed at 90° . If this angle lies outside the range $[\theta_{\min}, 90^\circ]$, we

penalize its deviation by minimizing E_{Fab} . We provide the detailed formulation of E_{Fab} in Supplement, Appendix A.

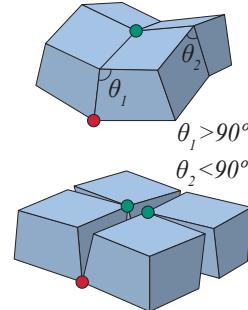
Extrusion ($K_{2D} \rightarrow T_{2D}$). After achieving the flat, non-extruded configuration (K_{2D}), we compute the final flat configuration by finding the bottom-surface vertices and generating the side faces. We first compute the rigid transformation matrices, comprising of rotation R and translation t , that transform each top-face vertex to its corresponding bottom-face vertex in T_{3D} , as derived in Section 4.2. Applying these transformations to the vertices of K_{2D} , we obtain the coordinates of the bottom surface in the flat configuration. We then use these obtained vertices on the top and bottom surface to generate the side faces, producing the corresponding tiles in the flat configuration (T_{2D}).

4.4 Hinge Placement Strategy

For an auxetic linkage to be functional, hinges need to be placed between every two connected linkage elements. The hinges can vary depending on the size of the linkage and fabrication technique. However, the difference between our approach and previous works involving auxetic linkages [Jiang et al. 2022; Konaković-Luković et al. 2018] is our elements are frustums and the hinges can be placed either on top or bottom surfaces, where the neighboring frustums touch in the flat configuration (see Supplement, Figure 19). During extrusion ($K_{3D} \rightarrow T_{3D}$), tiles that ensure complete contact between side faces in the assembled configuration may also create a dihedral angle $\theta > 90^\circ$. Placing all hinges on the top surface to connect the tiles would then result in overlapping regions in the flat configuration (see Figure 5, callouts), which is physically invalid.

To solve this issue, we propose two design choices that can be selected based on user requirements and resource availability. In the first approach, hinges are always placed on the top surface of the tiles on T_{2D} , and any overlapping region is removed from one of the tiles sharing the hinge (see Supplement, Figure 19, top row). This decreases the contact area between neighboring tile faces, which reduces compressive load capacity and diminishes the structure's interior aesthetics. Nevertheless, it offers manufacturing simplicity. When the overlapping region is large, removing it fails to provide sufficient compression, which is crucial for

keeping the structure assembled. To address this issue, our second approach places a hinge strategically either on top or bottom surfaces, so that no overlap occurs (see Inset on the left). We determine whether the hinge should be placed on the top or bottom by evaluating normal curvature in the direction of the opening dihedral angle at the joint location. If this curvature is negative, then the dihedral angles exceed 90° and the hinge should be placed at the bottom of the shared edge. Otherwise, the hinges should be placed at the top (see Inset). Although this approach can be more time-consuming to manufacture, it provides a solution that creates complete contact among the side faces of the tiles in the assembled configuration.



Introducing these dual-hinges triggers a cascading overlap among adjacent tiles, so we must apply a coordinated, global repositioning that preserves each tile’s rigidity, prevents collision, and aligns appropriate vertices for placing hinges. To find a non-overlapping geometric positioning of the tiles with this second strategy, we formulate an additional optimization step on the previously computed flat configuration T_{2D} . Using mesh T_{2D} , derived from Section 4.3, we minimize energy function E_{Hinge} on T_{2D} :

$$E_{\text{Hinge}}(v) = \omega_1 E_{\text{Rigid}}(v) + \omega_2 E_{\text{Collision}}(v) + \omega_3 E_{\text{Conn}}(v) \quad (6)$$

To maintain the rigidity of the tile, we impose 18 *Edge* (see Equation 3) constraints per tile: four top edges, four bottom edges, four side edges, and all six face diagonals, following Maxwell framework (see Supplement, Appendix B.1). Extending the 2D collision detection between triangular faces implemented by [Konaković et al. 2016], we implement *Collision* constraint for rigid 3D tiles, ensuring that the tiles never collide with each other during actuation.

To ensure that computed hinge pairs coincide and form shared pivot points during layout optimization, we impose a penalty when these points move apart. We minimize the connection energy E_{Conn} defined as:

$$E_{\text{Conn}} = \sum_{u \in H(T_{2D})} \|\mathbf{x}_u - P_{\text{Conn}}(\mathbf{x}_u)\|_2^2,$$

where $H(T_{2D})$ is the set of hinge vertex indices in the flat configuration T_{2D} , $\mathbf{x}_u \in \mathbb{R}^3$ is the current position of vertex u , and the projection operator $P_{\text{Conn}}(\mathbf{x}_u)$ returns the position of the hinge vertex paired with vertex u . Minimizing E_{Conn} simultaneously aligns every hinge-vertex pair and adjusts all tile positions to produce a non-overlapping flat layout with correctly located hinge points. After optimizing the hinge locations in the flat configuration T_{2D} , we merge each hinge vertex across its adjacent tiles, turning the disconnected set of quads into a single, connected linkage.

4.5 Mesh Splitting

The bounded conformal scale factor λ is the maximum local stretching allowed when mapping a curved surface onto a plane, ensuring that no linkage exceeds its physical deformation limits [Konaković et al. 2016]. The scale factor λ is constrained between 1 and a constant $\sigma > 1$, which depends on the linkage topology (triangular, quadrilateral, hexagonal, etc.). These bounds imply that not all surfaces can be physically realized in the 3D configuration of a flat auxetic linkage and λ must remain strictly below its theoretical upper limit. For triangular auxetic designs, the maximum area expansion is $\sigma = 4$, and for quadrilateral auxetic linkages we find $\sigma = 2$ (see Supplement, Figure 18).

Previous works utilize cone singularities and cuts within the auxetic topology to tackle geometric structures with $\sigma > 2$ [Jiang et al. 2022; Konaković et al. 2016; Konaković-Luković et al. 2018]. While these methods can reduce the scale factor of a structure to within fabricable limits, it interrupts the uniform quad grid. We observed this uniformity is an essential element and placing a complete split in the structure lends itself well to the grid configuration of quad auxetic linkages (see Figure 10 for examples). This approach naturally aligns with the minimal-link principle proposed by [Chen et al.

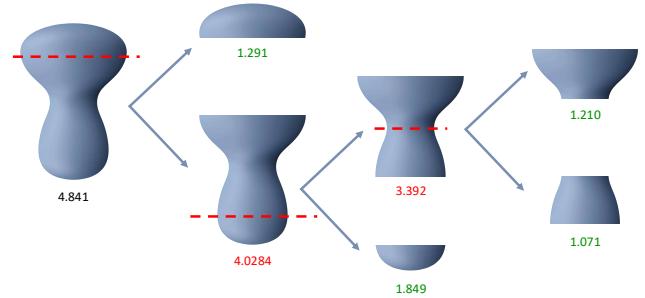


Fig. 6. Illustration of split placement based on locations of high curvature to reduce the conformal scale factor σ . We continue splitting the mesh until each part satisfies $\sigma \leq 2$.

2020], which shows that assembling such blocks need boundary connections between them. This motivates our approach since a string threaded inside these segments can easily provide necessary minimum links providing connectivity and rigidity during transition from flat to assembled configuration (see details in Section 5).

We introduce splits only in meshes where the conformal scale factor is greater than 2. For example, in Figure 10, 3 out of 5 examples have a split and we have used this technique on most of the meshes. After generating the quad mesh M_{3D} of the target structure and the corresponding M_{2D} , we introduce splits hierarchically within M_{2D} that bisect the mesh along the grid direction and through the point of highest Gaussian curvature to effectively reduce the scale factor. A complete split maintains regular connections within the linkage and eliminates combinatorial singularities or irregular vertices in both lifted and flat configurations. The split reduces the scale factor more effectively and remains parallel to the grid of auxetic mesh, facilitating the insertion of strings and keeping the same actuation mechanism. See Figure 6 for an example of a scale factor reduction, where the split direction follows the quad grid. Note that aligning the quad grid with all high curvature points of target mesh is not always possible. Thus, we leave feature alignment to the user’s discretion based on application-specific requirements.

5 String Placement and Actuation

After designing the mechanical linkage, we compute a physically realizable string path passing through a carefully selected subset of tiles to enable smooth actuation. The resulting path guides the string through specific lifting points: a location on the surface where the string enters and exits the structure, and it is being lifted and tightened from that location. Pulling the string at these lifting points lifts the corresponding regions and progressively constrains the movements of the tiles along the defined path, assembling the structure (see Figure 3).

A key observation we make is that once the boundary tiles are positioned into their target configuration, the geometry of the entire structure is uniquely determined. Thus, we first weave the string through the boundary tiles, fully closing and rigidifying the assembly. We provide further details on this claim in the following section (see Section 5.1). Next, we identify the minimum set of additional

lifting points required to achieve smooth actuation while minimizing frictional resistance. To do this, we introduce an algorithm that selects the minimum set of lift points based on the energetic interaction among candidate points (see Section 5.2). Among all valid routes connecting the lift points to the boundary, we compute the optimal string path that minimizes channel friction and guarantees connectivity across split components (see Section 5.3). We then discuss simulation of the actuation mechanism along with implementation details (see Section 5.4).

5.1 Constraining Boundary Tiles

The boundary of the quad auxetic linkage plays a critical role in constraining the motion of the entire structure. When a continuous string is routed through all boundary tiles (tiles containing at least one vertex on the outer loop) and tensioned, the side faces of these boundary tiles come into direct contact, constraining their relative motion. Under these conditions, the entire structure admits exactly the six trivial rigid-body motions and no internal infinitesimal flexes.

THEOREM 5.1. *A linkage formed of eight-vertex quadrilateral frustum rigid tiles in \mathbb{R}^3 , each joined to its neighbors at a single corner and arranged so that every interior void is a four-vertex rhomboid, when enclosed by and tensioned with a single continuous boundary string, is isostatic, admitting exactly the six trivial rigid-body motions and no internal infinitesimal flexes.*

See the detailed proof of the theorem in Supplement, Appendix B, where we provide a general theory for $N \times N$ and $N \times M$ auxetic topology and subsequently for irregular topology. In other words, the main degrees of freedom of our linkage are controlled by the boundary tiles. As the boundary starts shrinking to its target configuration, the interior linkage starts to close and locks once all the gaps are closed. Due to the quad auxetic topology of the linkage, absence of singularities, and constraining motion with dihedral angles, our linkage can be assembled simply by closing the boundary tiles when the weight of the structure is relatively small. However, as the structure becomes heavier, introducing a lifting point is beneficial to counteract gravitational loads during assembly. In the following section, we present an optimization strategy to choose these lifting points.

5.2 Selection of Minimum Lift Points

Since the string path must traverse all boundary tiles to constrain them, the next step is to determine the location of the lift points, where the string is being tightened and the structure lifted during the assembly. Any valid string path must therefore traverse all boundary tiles and the subset of interior tiles required to connect the lift point to the boundary, thereby forming a single connected route. The location of the lift point depends on the geometry of the structure as well as its size and mass distribution. Intuitively, the lift points are typically selected locally as the highest points on the surface. However, as the number of lift points increases, the string path becomes longer and more complex, introducing additional friction. Minimizing their number is therefore a key design strategy for smooth actuation. In the following, we provide a detailed algorithm for selecting the minimum set of lift points.

A naive solution would be to choose all local maxima throughout the structure as lift points. However, this ignores how the pull forces are distributed among those lifting points and can lead to wasted resources. Rather than assigning lift points at every geometric local maximum, we employ the discrete Morse–Smale theory [Forman 1998; Gyulassy et al. 2008] in conjunction with techniques from computational topology [Zomorodian and Carlsson 2004] to identify the minimum set of lift points required during actuation. We model each gap as a discrete energy field, with the gravitational potential energy (GPE) of each gap quantifying its load-bearing significance. First, we apply discrete Morse–Smale segmentation to identify the true energy maxima (peaks) and their associated regions, along with the energetic interaction information. Next, we build a topologically informed coupling directed graph among these peaks based on their energetic interactions and connected component of this graph clusters the peaks into minimum number of sets. This two-stage process identifies the minimum set of lift points needed to bear the load efficiently during assembly.

For each gap G_i , let $\mathcal{T}(G_i)$ denote the set of its surrounding tiles. If each tile $t_k \in \mathcal{T}(G_i)$ has mass m_k and center-of-mass height z_k , and if z_{\min} denotes the lowest point in the structure, then the gravitational potential energy of the gap (GPE), g_i , is given by

$$g_i = \sum_{t_k \in \mathcal{T}(G_i)} \frac{1}{4} m_k g (z_k - z_{\min})$$

We then identify the dominant energy peaks and their corresponding basins in this scalar field by applying Morse–Smale segmentation (see Supplement, Appendix C).

We build an undirected graph G where each node represent a gap and each edge (u, v) connect two gaps G_u and G_v if they share at least one surrounding tile, i.e. $\mathcal{T}(G_u) \cap \mathcal{T}(G_v) \neq \emptyset$. To quantify peak-to-peak coupling, we assign each edge (u, v) a weight $w(u, v) = \min\{g_u, g_v\}$, so that the strength of each connection is limited by its weaker end. We then compute the maximum-spanning tree (MST) T of the weighted graph G using the Kruskal algorithm [Kruskal 1956]. The unique path between two peaks i and j , denoted $\text{Path}_T(i, j)$, is formed during MST construction: Kruskal's algorithm sorts all edges in descending order of weight and adds them one by one, skipping any edge that would introduce a cycle.

At some stage, adding a particular edge connects two previously disconnected components containing peaks i and j . Since heavier edges are always considered first, any stronger possible connections between these peaks have already been evaluated. Hence, the edge that eventually joins these peaks represents the weakest (lightest) link along their path in the MST. Consequently, no alternative path between these two peaks in the original graph can have a stronger weakest edge.

Since each edge weight reflects the minimum GPE of its two connected nodes, this weakest link on the unique path $\text{Path}_T(i, j)$ corresponds precisely to the highest energy that must be overcome before the basins of peaks i and j can be merged. Thus, $\text{barrier}(i, j)$ can be defined as:

$$\text{barrier}(i, j) = \min_{(u,v) \in \text{Path}_T(i,j)} w(u, v)$$

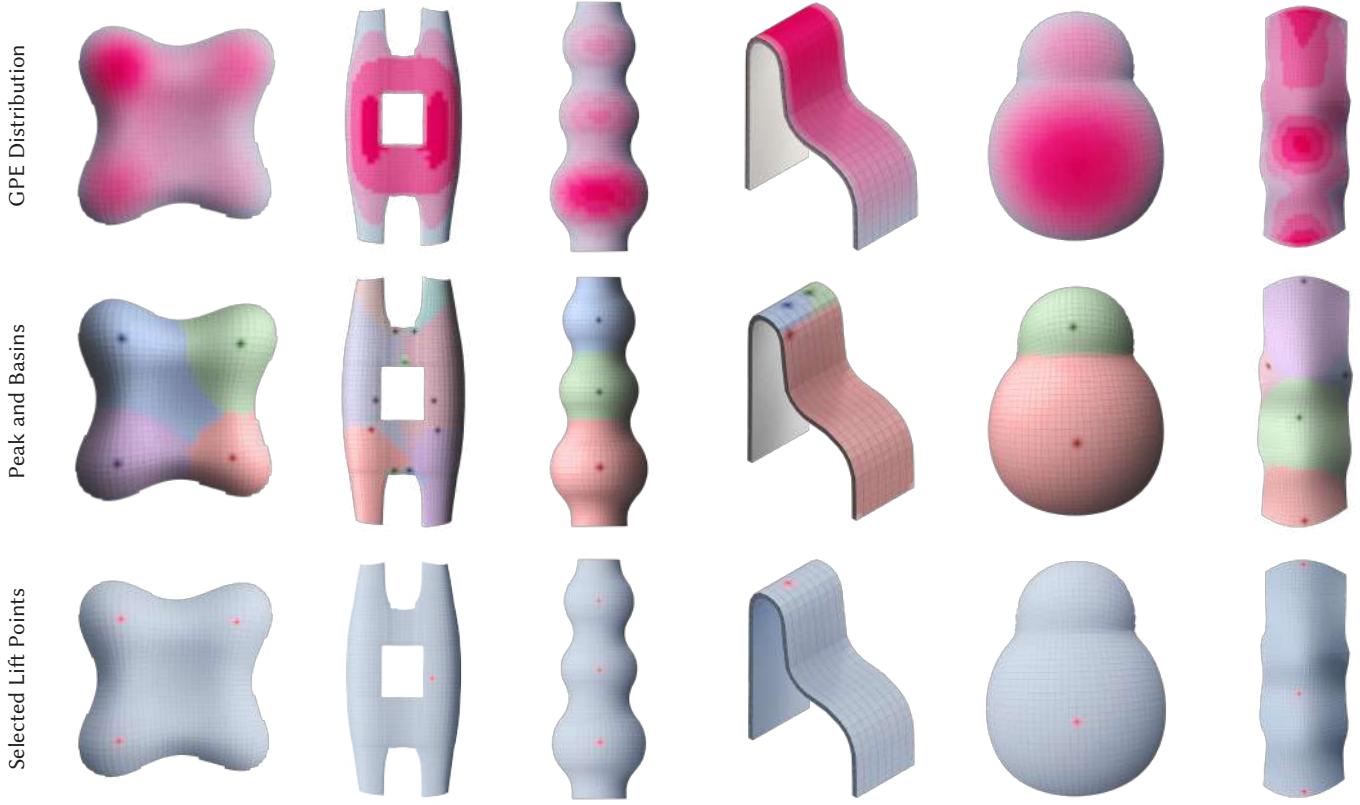


Fig. 7. Minimum lift points selection. *Top*: Per-vertex gravitational potential energy (GPE) distribution (red = high); *Middle*: Identified peaks (dark dots) and their Morse-Smale basins (shaded regions); *Bottom*: Final minimum lift points (red) selected via coupling analysis for string actuation (details in Section 5.2).

Higher energy indicates that the peaks i and j are more highly coupled with each other. Therefore, we define the coupling coefficient

$$c(i, j) = \frac{\text{barrier}(i, j)}{g_j}, i, j \in \{1, \dots, P\}, g_i \geq g_j, P \geq 1$$

where, the coupling coefficient $c \approx 1$ indicates that the peak j begins to rise almost simultaneously with the peak i , which indicates a strong energetic coupling. In contrast, $c \approx 0$ implies minimal coupling, which means that the two peaks remain effectively independent and transfer little to no energy [Zomorodian and Carlsson 2004].

We begin by constructing a Directed Acyclic Graph (DAG) whose vertices represent peaks. We use a coupling threshold τ to decide whether the energetic interaction between two peaks is strong enough to consider them coupled. We add an edge from peaks i to j if:

- $c(i, j) \geq \tau$
- $c(i, j) = \max_k c(k, j)$ (no other peak k has a larger coupling to j)

In the resulting DAG, we extract the connected components which correspond to a cluster of strongly coupled peaks. The number of total clusters denotes the number of minimum lifting points required.

Within each cluster, the unique node with zero in-degree, i.e. the cluster's highest energy peak, is selected as that cluster's lift point.

Determining an optimal threshold τ for identifying strong coupling is challenging, as it typically requires extensive testing with various physical structures. To mitigate sensitivity to threshold selection, we use a threshold-sweeping strategy. Initially, we construct the coupling DAG and select lift points using a starting threshold ($\tau_0 = 0.8$). We then simulate the actuation process with these selected lift points. If the deployment fails during transition (i.e., the structure is stuck in an intermediate configuration during simulation), we incrementally raise the threshold by 0.1, rebuild the coupling graph and corresponding lift-point clusters, and rerun the simulation. This sequential procedure continues until the structure successfully deploys from its flat configuration.

Figure 7 shows the result of the selection of minimum lifting points in multiple structures with various curvature and resolutions. Our method also addresses discretization artifacts that would otherwise inflate the peak count, especially in lower mesh resolution. For instance, the chair model in Figure 7 shows two closely spaced GPE maxima at its top, an artifact of low mesh resolution that would merge into a single peak at higher resolution. By analyzing energetic interactions, our algorithm automatically fuses such spurious duplicates and returns the minimum set of lift points.

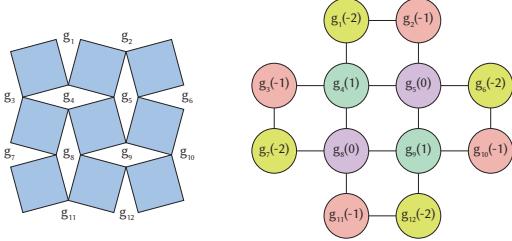


Fig. 8. Auxetic representation to gap-graph formulation: empty spaces marked as g_i (left) are represented as nodes in the gap graph (right). In the gap graph, node value 0 denotes vertical gap (purple), 1 denotes horizontal gap (deep green), -1 denotes a virtual entrance point on the boundary which are not physical gaps (red), -2 denotes split boundary gaps created when the mesh is segmented into multiple segments (light green).

5.3 Minimal Friction String Route

Having identified the minimum set of lift points, we next compute a valid closed string route that minimizes friction between the string and string channel. We model channel energy, E_{Channel} using the Capstan equation [Stuart 1961]. If the string enters under tension T_1 and wraps through a cumulative angle θ_{Total} , and μ_c is the channel wall friction coefficient then, $E_{\text{Channel}} = T_1 (e^{\mu_c \theta_{\text{Total}}} - 1)$ (see Supplement, Appendix D).

We represent the *flat* configuration as an undirected graph $G = (V, E)$, where each node $g_i \in V$ corresponds to a gap and each edge $(g_i, g_j) \in E$ indicates that the two gaps share a vertex (Figure 8). Owing to the regular quadrilateral connectivity, G forms a grid-like structure. Each gap node g_i carries an integer label x_i specifying its role: $x_i = 0$ and $x_i = 1$ denote vertical and horizontal gaps, $x_i = -1$ denotes a virtual boundary entrance (allowing entry into an interior lift-point gap), and $x_i = -2$ denotes a split boundary gap resulting from mesh splits. A valid route R must form a closed walk C on G , such that $C \subseteq N \subseteq V$, where N is the set of visited nodes.

While more elaborate models of friction along curved string paths exist, we draw inspiration from prior work in robotics [Choi et al. 2023; Heap et al. 2022; Suthar et al. 2018] and graph theory [Demaine et al. 2024], which effectively employ the Capstan equation to capture frictional effects. Due to the earlier stages of our pipeline, which guarantee a uniform quadrilateral topology and select minimum lift points to add to the string path, the path space is already well-structured and yields simple candidate routes. At this stage, large directional turns as the string weaves through the tiles is the remaining contributor of friction. Rather than estimating the absolute magnitude of friction, we focus on minimizing the cumulative sharp bends in the string path. This approach is computationally lightweight and practically effective, as demonstrated by the successful deployment of our prototypes across a range of scales.

We minimize E_{Channel} to reduce overall friction in the string channel. Previous work on string routing [Demaine et al. 2024] emphasized minimizing the turn cost over total string length, noting that although shorter strings reduce material cost and manufacturing time, friction during the deployment of a structure is primarily determined by string turns within the channel. In a valid string path

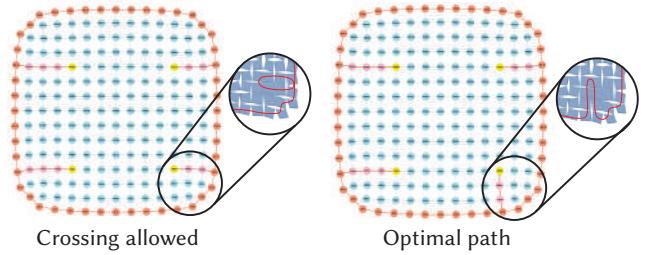


Fig. 9. If a string path from boundary enters the interior of the linkage from a joint rather than a gap, the string path can avoid making a crossing on each other, which significantly reduce the friction. Left: minimize crossing is taken off. Right: Showing the optimal path with minimum friction.

within the graph, each node corresponds to a gap in the flat configuration. The turn angle at a node is given by the angle formed between the centroids of its predecessor, itself, and its successor. Although the path is extracted from the graph, the angles are measured using the geometric positions of the gaps in the flat configuration rather than the graph edges. The total turn angle of the path, θ_{Total} , is obtained by summing these local contributions. Assuming uniform entry tension T_1 and channel friction coefficient μ_c , our objective is therefore to identify a valid string route $R \subseteq \{g_1, g_2, \dots, g_{\text{lift}}\}$ that minimizes the cumulative turn angle $\min_R \theta_{\text{Total}}$ under the following constraints:

Minimize Crossing. Strings crossing each other in a gap or channel introduces additional friction as they slide against each other (Figure 9, *Left*) and hinder smooth deployment. To minimize crossing, we need to constrain each gap $g_i \in N_i$ to be traversed by exactly one string, thereby minimizing the number of paths intersecting any tile. Formally, in the graph $G = (V, E)$, when a string path transitions from a boundary node $g \in V_b$ to an internal node which is a lift point gap, it should try to enter from a gap with $x_b = -1$, see Figure 9.

Connecting Splits. A valid path must also preserve component connectivity by traversing all boundary tiles while avoiding routes that separate adjacent parts of the structure. In practice, this means the string may traverse boundary nodes split by prior cuts, but it cannot enter the interior through them. Formally, entry from a gap with $x_b = -2$ is disallowed.

5.4 Simulation and Implementation

Using the generated string path, we simulate the structures from flat to assembled configuration (see Supplementary video, 00:19) to evaluate our method before proceeding to physical fabrication. For the simulation environment, we use Projective Dynamics [Bouaziz et al. 2014] for computational efficiency and compatibility with our design pipeline, which allows us to develop an end-to-end workflow for the design and actuation of our structures. Our actuation simulation includes dynamic effects (e.g., gravity) through position and velocity updates (see Supplement, Appendix E).

During this simulation, we need the tiles to be rigid and non overlapping with each other. Additionally, we need to model the actuation behavior of the string. When a single string is routed

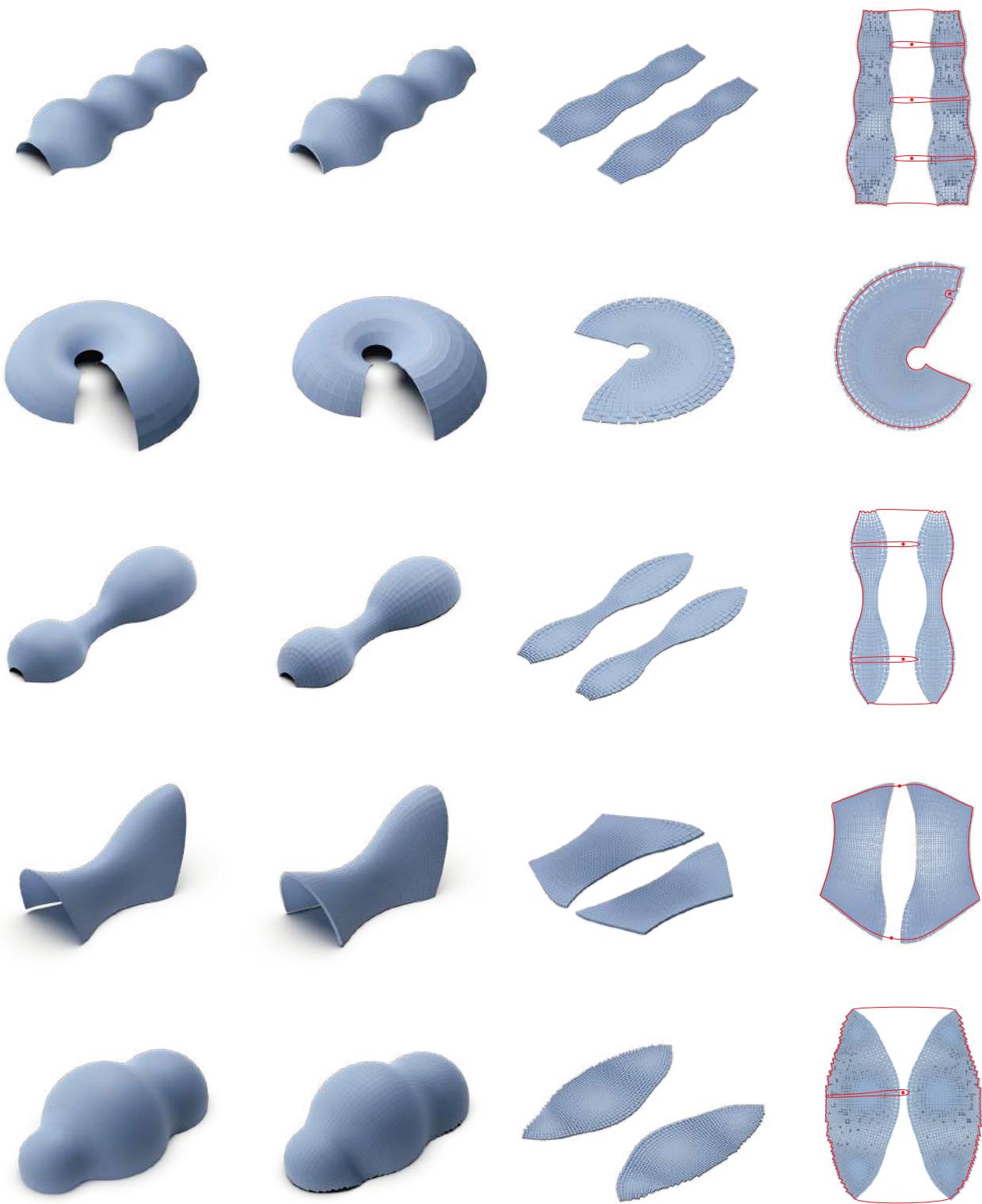


Fig. 10. From left to right: target surface, assembled 3D structure, flat layout, optimized string path on flat linkage. From top to bottom: ThreeShell, Seashell, Peanut, Saddle, Cloud.

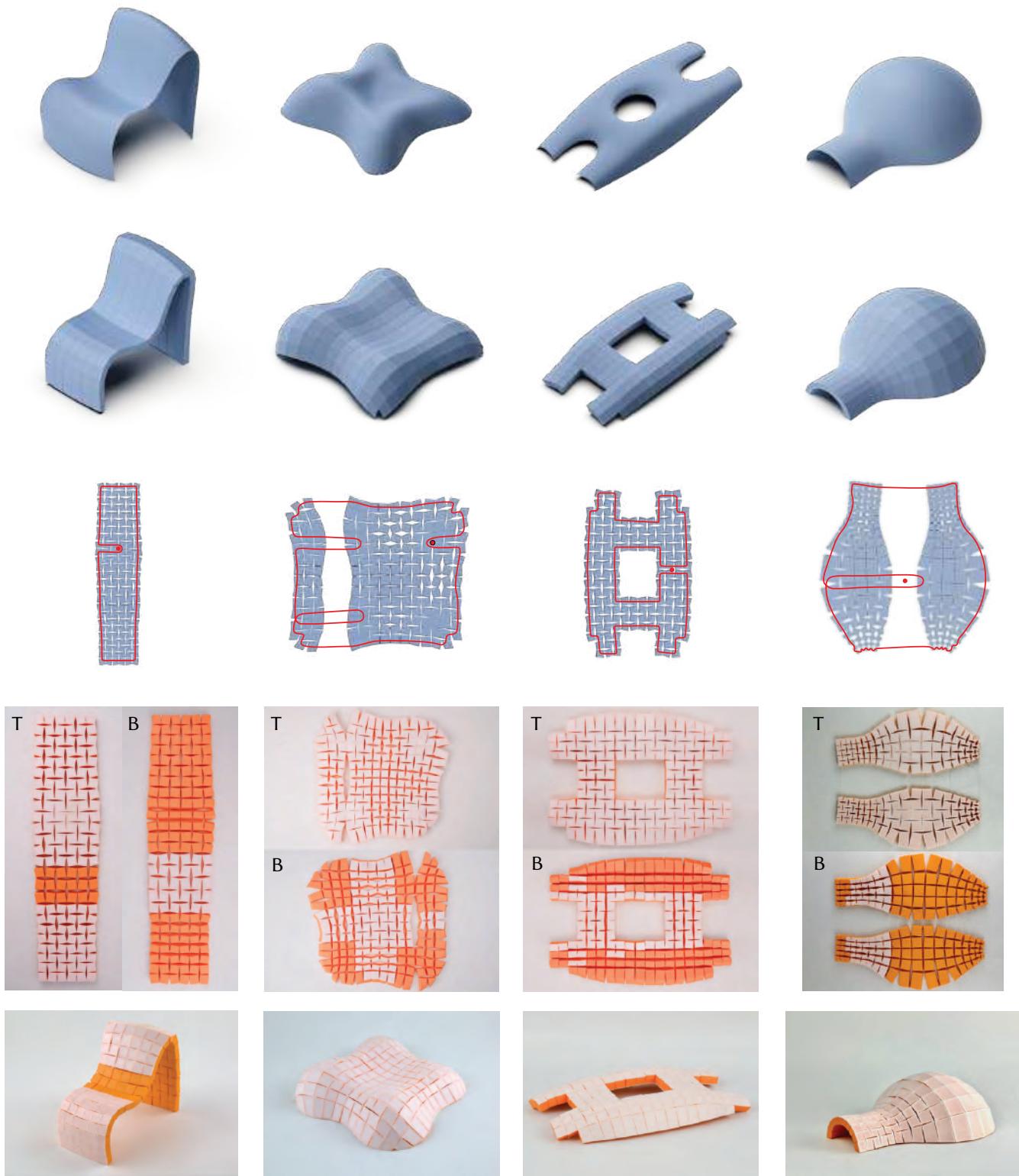


Fig. 11. From top to bottom: target surface, assembled 3D structure, optimized string path using the minimum set of lift points, fabricated flat layout (T: top, B: bottom), deployed structure. All models are manufactured via multi material 3D printing (FDM), using orange PLA for the rigid tiles and white TPU for the hinges.



Fig. 12. Application in the furniture industry, a functional human-scale chair: assembled configuration (left), top and side-view of flat configuration (right). This chair can be manufactured, stored, and shipped flat, while being deployed quickly and reliably by the consumer with a single string pull.



Fig. 13. Timelapse of deploying and disassembling a chair, demonstrating the reversible actuation capability of our method.

through a sequence of tiles, it performs two kinematic tasks: (a) it brings the mid-points of adjacent side faces into coincidence, thereby closing each gap, a *snap* action and (b) at its exit, it lifts the associated gap to a prescribed height, a *lift* action. Because both effects are purely geometric, we can encode them as positional constraints on those mid-points, rather than modeling a discretized rope model, for a faster simulation.

Additionally, to model the string actuation behavior maintaining the rigidity of the tiles and actuate them from a flat to an assembled configuration, we minimize the energy $E_{\text{Simulation}}$, defined as:

$$E_{\text{Simulation}}(v) = \omega_1 E_{\text{Rigid}}(v) + \omega_2 E_{\text{Collision}}(v) + \omega_3 E_{\text{Actuation}}(v),$$

where E_{Rigid} represents deviation from rigidity, $E_{\text{Collision}}$ represents collision, $E_{\text{Actuation}}$ represents deviation from string pulling action respectively, and ω_1 , ω_2 and ω_3 are their corresponding weights. Here, E_{Rigid} and $E_{\text{Collision}}$ are formulated identical to Equation 6. Additionally, we form two constraints to abstract string-path actuation from the flat rest state to the lifted configuration. First, we penalize the squared distance between each paired side-face midpoint along the string path and its partner's position, defining the energy E_{Snap} . Second, at each designated lift gap, we penalize the squared distance between each vertex's current position and its prescribed lift target, capturing the string's pull along its exit

direction and thereby define the energy E_{Lift} . Detailed formulations of E_{Snap} and E_{Lift} are provided in Supplement, Appendix E. Together, the snap and lift constraints model the motion of how the auxetic mechanism responds to the string pull by snapping closed the gaps between tiles and lifting the tiles at the designated lift points.

Implementation. We developed a computational design tool with an interactive user interface to design and optimize configurations. The implementation of constraints for both assembled and flat configurations builds on the open-source libraries ShapeOp [Deuss et al. 2015] and libigl [Jacobson and Panozzo 2017]. For a mesh containing $\sim 10k$ triangles, optimization of the assembled configuration takes ~ 1.2 minutes, while optimization of the flat configuration takes ~ 0.6 minutes on a Mac M1 Pro computer with a 3.2 GHz, eight-core processor. Actuation simulation requires an additional 5.5–6 minutes to complete the assembly. Although our design approximation (Section 4) employs soft constraints, we find that intuitive choices of relative weights consistently yield good results across different structures. For reproduction of our method we provide the weights used in minimizing $E_{\text{Assembled}}$, E_{Flat} , and E_{Hinge} (see Section 4). In Equation 1, we set $\omega_1 = 10000$, $\omega_2 = 10$, and $\omega_3 = 0.1$. A low weight on surface matching (ω_3) allows deviations from the target geometry to accommodate planar elements and coarse approximations, while a high weight on planarity ensures physical fabricability. In



Fig. 14. Comparison of two fabrication methods and hinge placement strategies: Top: dual-material FDM printing with TPU linkages and PLA tiles. Bottom: laser-cut EPDM rubber hinges adhered to PLA tiles. Left: top-surface-only hinge placement. Right: hinges placed on the top or the bottom using the design optimization strategy presented in Section 4.4. Prototypes with dual surface hinge placement show improved quality over top surface only hinges.

Equation 5, we set $\omega_1 = 1$, $\omega_2 = 1$, and $\omega_3 = 0.001$, balancing edge matching and collision avoidance equally, while allowing ω_3 to be tuned to control gap size for ease of fabrication. Finally, in Equation 6, we set $\omega_1 = 100$, $\omega_2 = 100$, and $\omega_3 = 1$, emphasizing rigidity and collision avoidance over alignment of the hinge vertices.

6 Results and Discussion

We evaluate our method on a wide range of doubly curved surfaces (Figure 10), complemented by physical fabrications (Figure 11) and a large-scale prototype (Figure 12), which showcase a variety of fabrication methods and the effectiveness of our hinge placement strategy. Finally, we highlight application case studies spanning biomedical devices, consumer products, space exploration, robotics, and architecture.

6.1 Fabrication and Deployment

Fabrication methods. The auxetic mechanism consists of two components: the rigid tiles and the flexible hinges that connect them. In Figure 14, we present two possible fabrication techniques: 1) dual-material 3D printing with Flexible Thermoplastic Polyurethane (TPU) for the linkages and Polylactic Acid (PLA) for the tiles, and 2) laser-cut Ethylene propylene diene monomer (EPDM) rubber hinges adhered to PLA tiles. A thickness of 0.6mm for the TPU joints is found to provide sufficient flexibility and resistance to tearing, while a width of 0.8mm prevents plastic deformation caused by excessive stiffness. To create a tight bond between TPU and PLA, an interlocking structural design is used between the material layers. By contrast, laser-cut EPDM rubber offers lower resistance to rotation but requires an additional fabrication step of manual adhesion. This

highlights a trade-off between quality and ease of production, illustrating that fabrication choices for our designs can be tailored to user requirements and available resources.

Hinge placement ablation. We compare the hinge placement strategies of top-surface hinges and the dual-hinge method in Figure 14. Top surface hinges allow the prototype to sag in regions with concave features, since tile overlap in the flat layout was trimmed. The dual-hinge method automatically places hinges on the bottom in these regions, which enables full side-face contact between tiles and prevents sagging.

Small-scale prototypes. We show several physical prototypes fabricated using dual-material 3D printing in Figure 11. We use an Ultimaker S5 3D printer with a build plate volume of $330 \times 240 \times 300$ mm to fabricate the prototypes. The assembled configurations of these structures have an average size of $164 \times 121 \times 56$ mm and an average mass of 135 g (see Supplement, Table 1). In structures having splits, we embed magnets on the side faces of the tiles along the split zone to precisely align the adjacent tiles in the assembled state. For different applications and scales, tiles can also be aligned with other engineering solutions, such as pins or snapping joints.

Large-scale prototype. We fabricate a life-size, functional chair (Figure 12), measuring approximately 3 ft in height once assembled (see Supplement, Table 1). The structure is made of laser-cut plywood boxes that are connected with rubber strips as hinges. A vinyl-coated stainless steel cable is routed through plastic channels glued within the tiles. The chair deploys smoothly from the flat state, remains stable after tension release, and can be flattened and folded into a compact form. (Figure 13). For sitting, the cable is knotted at the lift point to maintain tension, and the chair is placed within a wooden frame that constrains the boundary tiles in contact with the ground.

6.2 Discussion on Design and Optimization Choices

We optimize the assembled configuration first, followed by the flat configuration through two distinct steps as discussed in Section 4. Although joint co-optimization is a natural alternative, in our setting it is unnecessary. Conformal mapping with a bounded scale factor ensures that a valid flat embedding always exists. Thus, edge lengths of the flat configuration can always match those already optimized for the assembled configuration. Across the pipeline, we choose the simplest approach that achieves the required accuracy. We enforce planarity by computing a best-fit plane via SVD for each face and projecting its four vertices onto that plane, with per-face operations executed in parallel. For surface matching, we minimize a point-to-point energy using an AABB tree for closest-point queries, which directly reduces Euclidean distances and preserves sharp features without relying on normals or offset surfaces. While other choices are possible, this integrates cleanly with ShapeOp and has proven reliable in practice.

Planarity of the Contact Faces. Although perfect planarity is not assured, our method significantly minimizes deviations by prioritizing planarity over surface approximation, since this is crucial for valid fabrication and assembly. We quantify planarity error as the

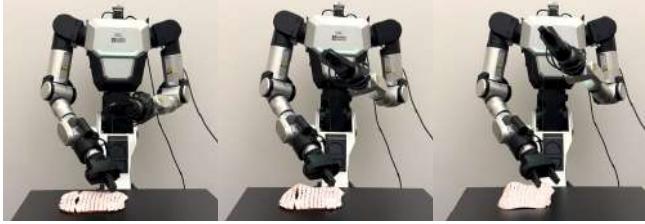


Fig. 15. Timelapse of a humanoid robot assembling Lilium containing 193 tiles with one continuous string pull (see Supplementary Video, 01:23). Our method can facilitate autonomous robotic assembly of structures, as the single string pull actuation mechanism is a much simpler motion planning task than assembling individual modules.

maximum vertex to best-fit plane distance, with 0 indicating perfect planarity. Upon optimization, 3D printed Lilium ($211 \times 212 \times 169$ mm) although not a conical mesh, demonstrates reductions in maximum planarity error in absolute scale from 0.0074 mm to 0.00036 mm (top faces), 0.1729 mm to 0.00038 mm (bottom faces), and 0.587 mm to 0.00039 mm (contact faces). While we observe larger errors at lower resolutions and smaller ones at higher resolutions, as expected due to the discretization error, our optimization consistently drives deviations to negligible values, typically by factors of 100 to 1000.

Quad Remeshing. We initialize our optimization with conformal mapping, since quad remeshing through alignment of Conjugate Direction Field (CDF) often introduce points with five valances [Dong et al. 2006; Eigensatz et al. 2010; Liu et al. 2006, 2011], i.e., singularities. Such points cannot be represented in a quad linkage, which always requires four vertices. Similarly, aligning with a Rotational Symmetry Field (n-RoSy) [Ray et al. 2006] does not eliminate the issue and singularities still arise, which is undesirable for the quad auxetic topology we adopt. We can observe the same issue in [Jiang et al. 2022] proposing a quad-auxetic mechanism, where initial remeshing using alignment of the curve network produces quadrilateral tiles but non-quadrilateral gaps. By contrast, we prioritize preserving quadrilateral gaps over exact curve alignment, yielding a grid-like pattern that enables simpler string routing and assembly.

Absence of Singularities. Approximating surfaces without allowing for singularities limits achievable shapes. We mitigate this by introducing splits (see Section 4.5), which expands the achievable shape space without introducing singularities that can cause irregular non-quad gaps and disrupt auxetic linkage connectivity. Our method relies on consistent gap geometries and uniform tile configurations to ensure effective cumulative gap closure and smooth force propagation via string tension.

6.3 Application Case Studies

Our method can be utilized to design structures for a variety of applications, including consumer goods, medical devices, autonomous robotic assembly, and large-scale architectures.

Consumer Items. We present the physical case study of a chair in Figure 12. Our algorithm encodes the ergonomic curves of a chair into a modularized planar state, allowing it to be manufactured from planar materials and transported flat to reduce time and cost. Once purchased, the ease of a single string pull eliminates any assembly challenges for the consumer. The flat configuration can also be folded compactly for storage and transportation (Figure 13).

Personalized Medical Devices. We explore applications in personalized medical devices in Figure 2. The ability to approximate doubly curved geometries allows us to create medical devices that conform to the complex contours of the body. Manufacturing in the flat state enables quick and simple fabrication of personalized geometries. We present the examples of a splint and posture corrector, both of which can be custom-fit to a user and stored flat while not in use, and then are easily assembled at the time of a user’s need by tightening a string.

Fast Robotic Assembly. Robotic assembly is expected to be at the forefront of future manufacturing. However, assembling a complex freeform structure from modules remains a challenging problem for robotic systems, requiring complex motion planning and often multiple robotic arms to hold the intermediate configurations in place. Our method facilitates fast and compute-efficient assembly, since the robot only needs to detect and pull a single string. This reduces the task to a simple motion while our pipeline ensures a minimal-friction string path that can be easily actuated, making the process well suited for robotic systems. In Figure 15, we demonstrate the effortless assembly of a 3D-printed prototype with 193 modules using a humanoid robot (see Supplementary Video, 01:23).

Space Exploration and Architectural Structures. Compact, easily maneuverable and deployable habitats are desirable in space missions. Our method could offer a solution that addresses these requirements. The habitats could be designed as flat-manufactured modular systems that compactly pack for storage and transit and



Fig. 16. Modular space habitat. Our method is convenient for designing flat-manufactured modular habitats that are compact to store and transport, and can be rapidly deployed, disassembled, and reassembled through robotic assembly in hostile extraterrestrial environments, potentially saving critical time and energy in future space missions. While similar structures have been evaluated at small scale, application-specific studies on hardware and large-scale deployment remain as future work.



Fig. 17. Rapid assembly of architectural structure. Our geometric approximation could provide the frames around the glass panels which can be fully manufactured off-site, transported flat, and deployed onsite using only two cranes. While our design method is scalable, realizing full architectural deployment requires future engineering studies.

can be rapidly deployed, disassembled, and reassembled via robotic assembly (See Figure 16).

Similarly, architectural structures benefit from simpler and faster deployment mechanisms, as well as manufacturing and transporting flat structures. A vast majority of architectural designs rely on building a structure from modules. Our method could facilitate the creation of both final modular structures or on-site scaffolding for freeform designs. Note that the modules are not required to be solid blocks of material. They can also be realized as hollow frames representing the edges of tiles computed by our optimization system. Then, for example, glass panels could be mounted only on the top faces of the frames. These frames can be deployed rapidly with minimal crane operations and limited human intervention (see Figure 17). Although related concepts have been demonstrated at small scale, rigorous evaluation for specific hardware and large-scale deployment remain areas for future work.

7 Limitations and Future Work

Our current implementation focuses on input surfaces with disk topology. More work is required to generalize our initial solution to other topologies, which can be an interesting avenue for future exploration. Exploring fabrication of a sphere-like topology is mainly constrained by conformal mapping and its inability to always align with the quadrilateral pattern. However, this is an interesting venue for future research.

For very complex geometries, maintaining a conformal scale factor below our threshold requires many splits, making the fabrication and actuation challenging. For complex structures, like the Stanford Bunny, numerous splits would be needed to keep the conformal scale factor below our threshold (≥ 2). Optimally placing numerous splits in the right location and orientation poses a separate and complex research problem, beyond the capabilities of our current algorithm.

Although our theory shows that constraining the boundary with the string guarantees rigidity of the assembled configuration, we observe that all of our fabricated structures remain stable under

self-weight once the string tension is released after deployment (see *Lilium* at 01:12, *small-scale chair* at 02:00, and *human-scale chair* at 02:38 in Supplementary Video). However, this behavior was neither rigorously tested nor enforced during optimization. Nevertheless, it indicates a strong potential to extend our method to create single-string actuated self-supporting structures. This would be further beneficial for a range of applications and particularly interesting for large-scale architectures that could stand independently after actuation.

Though we demonstrate building a human-size chair, we do not guarantee a solution to every scale-specific challenge. The string paradigm that assembles a human-scale chair can, in principle, deploy a building-scale architectural structure (see Figure 17) using a minimal number of cranes. The primary engineering hurdle is hardware specifications, such as identifying cable diameters and hinge strength. We leave this scale-specific engineering challenge aimed at architectural-scale deployment for future exploration.

Finally, despite its promise for reversible deployment in space habitats and disaster-relief scenarios, our current simulation system does not incorporate environmental loads such as wind or thermal stresses. A valuable direction for future work is to integrate these loads into the simulation framework to assess linkage performance in diverse operating environments and fabrication scenarios.

8 Conclusion

We present a novel approach for assembling a 3D structure from a flat linkage via a single string pull. Our key contribution abstracts the interaction between the auxetic linkage and the actuation string solely through tile connectivity, making the method agnostic to scale, material, and fabrication technique. The pipeline unifies geometry-driven quad tiling, collision-free flattening, lift-point minimization, and friction-aware string routing into a coherent design tool. Designed and fabricated prototypes exhibit diverse curvatures and geometric complexities, demonstrating that the method produces deployable, rigid, and reversible structures without specialized equipment. By encoding kinematics, structural integrity, and surface closure directly in a flat rest state, the approach establishes an efficient, scalable, and reversible assembly paradigm. Our method opens new avenues for compact and adaptive consumer products, personalized medical devices, fast and autonomous robotic assembly, transportable and rapidly deployable space habitats, emergency shelters, and architectural applications.

Acknowledgments

We are grateful to Pavle Konaković for his help with design, modeling and rendering. We thank Ryan Mei and Aylin Baca for their generous help with fabrication. We also thank the anonymous reviewers for their valuable feedback. This work was supported by the MIT Research Support Committee Award.

References

- Hassan Bahrami, Michal Piovarci, Marco Tarini, Bernd Bickel, and Nico Pietroni. 2025. Fabricable Discretized Ruled Surfaces. *ACM Transactions on Graphics* 44, 3 (2025), 1–15.
 Quentin Becker, Seiichi Suzuki, Yingying Ren, Davide Pellis, Francis Julian Panetta, and Mark Pauly. 2023. C-shells: Deployable Gridshells with Curved Beams. *ACM Transactions on Graphics* 42, 6 (2023), 1–17.

- Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. 2012. Shape-up: Shaping discrete geometry with projections. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1657–1667.
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph.* 33, 4, Article 154 (jul 2014), 11 pages. doi:10.1145/2601097.2601116
- Siheng Chen, Gary PT Choi, and L Mahadevan. 2020. Deterministic and stochastic control of kirigami topology. *Proceedings of the National Academy of Sciences* 117, 9 (2020), 4511–4517.
- Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. 2021. Bistable auxetic surface structures. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–9.
- Tianyu Chen, Xudong Yang, Bojian Zhang, Junwei Li, Jie Pan, and Yifan Wang. 2024. Scale-inspired programmable robotic structures with concurrent shape morphing and stiffness variation. *Science Robotics* 9, 92 (2024), eadl0307.
- Gary PT Choi, Levi H Dudte, and Lakshminarayanan Mahadevan. 2019. Programming shape using kirigami tessellations. *Nature materials* 18, 9 (2019), 999–1004.
- Gary PT Choi, Levi H Dudte, and L Mahadevan. 2021. Compact reconfigurable kirigami. *Physical Review Research* 3, 4 (2021), 043030.
- Myeongjin Choi, Sahooh Ahn, Hwa Sook Kim, and Taewon Seo. 2023. Examining the mechanics of rope bending over a three-dimensional edge in ascending robots. *Scientific Reports* 13, 1 (2023), 20663.
- Erik D Demaine, Yael Kirkpatrick, and Rebecca Lin. 2023. Graph threading. *arXiv preprint arXiv:2309.10122* (2023).
- Erik D Demaine, Yael Kirkpatrick, and Rebecca Lin. 2024. Graph Threading with Turn Costs. *arXiv preprint arXiv:2405.17953* (2024).
- Mario Deus, Anders Holden Deleuran, Sofien Bouaziz, Bailin Deng, Daniel Piker, and Mark Pauly. 2015. ShapeOp—a robust and extensible geometric modelling paradigm. In *Modelling Behaviour: Design Modelling Symposium 2015*. Springer, 505–515.
- Shen Dong, Peer-Timo Bremer, Michael Garland, Valerio Pascucci, and John C Hart. 2006. Spectral surface quadrangulation. In *Acm siggraph 2006 papers*. 1057–1066.
- Levi H Dudte, Gary PT Choi, Kaitlyn P Becker, and L Mahadevan. 2023. An additive framework for kirigami design. *Nature Computational Science* 3, 5 (2023), 443–454.
- Soya Eguchi, Claire Okabe, Mai Ohira, and Hiroya Tanaka. 2022. Pneumatic Auxetics: Inverse design and 3D printing of auxetic pattern for pneumatic morphing. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–7.
- Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. 2010. Paneling architectural freeform surfaces. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) (SIGGRAPH '10). Association for Computing Machinery, New York, NY, USA, Article 45, 10 pages. doi:10.1145/1833349.1778782
- Robin Forman. 1998. Witten–Morse theory for cell complexes. *Topology* 37, 5 (1998), 945–979.
- Jianzhu Gu, David E Breen, Jenny Hu, Lifeng Zhu, Ye Tao, Tyson Van de Zande, Guanyun Wang, Yongjie Jessica Zhang, and Lining Yao. 2019. Geodesy: Self-rising 2.5 d tiles by printing along 2d geodesic closed path. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.
- Jianzhe Gu, Vidya Narayanan, Guanyun Wang, Danli Luo, Harshika Jain, Kexin Lu, Fang Qin, Sijia Wang, James McCann, and Lining Yao. 2020. Inverse design tool for asymmetrical self-rising surfaces with color texture. In *Proceedings of the 5th Annual ACM Symposium on Computational Fabrication*. 1–12.
- Ruslan Guseinov, Connor McMahan, Jesús Pérez, Chiara Daraio, and Bernd Bickel. 2020. Programming temporal morphing of self-actuated shells. *Nature communications* 11, 1 (2020), 237.
- Ruslan Guseinov, Eder Miguel, and Bernd Bickel. 2017. CurveUps: shaping objects from flat plates with tension-actuated curvature. *ACM Trans. Graph.* 36, 4 (2017), 64–1.
- Attila Gyulassy, Peer-Timo Bremer, Bernd Hamann, and Valerio Pascucci. 2008. A practical approach to Morse–Smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1619–1626.
- Siyuan He, Mengran Wu, Arthur Lebée, and Mélina Skouras. 2024. MatAIIrals: Inflatable Metamaterials for Freeform Surface Design. (2024).
- William E Heap, Chris T Keeley, Elvy B Yao, Nicholas D Naclerio, and Elliot W Hawkes. 2022. Miniature, lightweight, high-force, capstan winch for mobile robots. *IEEE Robotics and Automation Letters* 7, 4 (2022), 9873–9880.
- Alec Jacobson and Daniele Panozzo. 2017. Libig: Prototyping geometry processing research in c++. In *SIGGRAPH Asia 2017 courses*. 1–172.
- Caigui Jiang, Dmitry Lyakhov, Florian Rist, Helmut Pottmann, and Johannes Wallner. 2024. Quad mesh mechanisms. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–17.
- Caigui Jiang, Florian Rist, Helmut Pottmann, and Johannes Wallner. 2020a. Freeform quad-based kirigami. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–11.
- Caigui Jiang, Florian Rist, Hui Wang, Johannes Wallner, and Helmut Pottmann. 2022. Shape-morphing mechanical metamaterials. *Computer-Aided Design* 143 (2022), 103146.
- Caigui Jiang, Cheng Wang, Eike Schling, and Helmut Pottmann. 2020b. Computational design and optimization of quad meshes based on diagonal meshes. *Advances in Architectural Geometry 2020* (2020).
- David Jourdan, Pierre-Alexandre Hugron, Camille Schreck, Jonás Martínez, and Sylvain Lefebvre. 2023. Shrink & Morph: 3D-printed self-shaping shells actuated by a shape memory effect. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–13.
- David Jourdan, Mélina Skouras, Etienne Vouga, and Adrien Bousseau. 2022. Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 493–506.
- Martin Kilian, Aron Monszpart, and Niloy J. Mitra. 2017. String actuated curved folded surfaces. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 1–13.
- Mattis Koh, Yue Wang, Kristina Shea, and Tian Chen. 2023. Shape reconfiguring bistable structures using heat activated fibers. *Engineering Structures* 295 (2023), 116792.
- Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond developable: computational design and fabrication with auxetic materials. *ACM Transactions On Graphics (TOG)* 35, 4 (2016), 1–11.
- Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid deployment of curved surfaces via programmable auxetics. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.
- Joseph B Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7, 1 (1956), 48–50.
- Qingkun Liu, Wei Wang, Himani Sinhmar, Itay Griniasty, Jason Z Kim, Jacob T Pelster, Paragkumar Chaudhari, Michael F Reynolds, Michael C Cao, David A Muller, et al. 2024. Electronically configurable microscopic metasheet robots. *Nature Materials* (2024), 1–7.
- Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. 2006. Geometric modeling with conical meshes and developable surfaces. In *ACM SIGGRAPH 2006 Papers*. 681–689.
- Yang Liu, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. 2011. General planar quadrilateral mesh design using conjugate direction field. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–10.
- Leixin Ma, Mrunmayi Munegowda, Kvani Roychowdhury, and Mohammad Khalid Jawed. 2024. Rapid design of fully soft deployable structures via kirigami cuts and active learning. *Advanced Materials Technologies* 9, 5 (2024), 2301305.
- Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. 2018. FlexMaps: computational design of flat flexible shells for shaping 3D objects. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.
- Matthew K McBride, Alina M Martinez, Lewis Cox, Marvin Alim, Kimberly Childress, Michael Beiswinger, Maciej Podgorski, Brady T Worrell, Jason Killgore, and Christopher N Bowman. 2018. A readily programmable, fully reversible shape-switching material. *Science advances* 4, 8 (2018), eaat4634.
- Mariam Mir, Murtaza Najabat Ali, Javarria Sami, and Umar Ansari. 2014. Review of mechanics and applications of auxetic structures. *Advances in Materials Science and Engineering* 2014, 1 (2014), 753496.
- Koya Narumi, Kazuki Koyama, Kai Suto, Yuta Noma, Hiroki Sato, Tomohiro Tachi, Masaaki Sugimoto, Takeo Igarashi, and Yoshihiro Kawahara. 2023. Inkjet 4D print: Self-folding tessellated origami objects by inkjet UV printing. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–13.
- Lauren Niu, Xinyi Yang, Martin Nisser, and Stefanie Mueller. 2023. PullupStructs: Digital Fabrication for Folding Structures via Pull-up Nets. In *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction*. 1–6.
- Amirali Nojoomi, Junha Jeon, and Kyungsuk Yum. 2021. 2D material programming for 3D shaping. *Nature communications* 12, 1 (2021), 603.
- Yuta Noma, Koya Narumi, Fuminori Okuya, and Yoshihiro Kawahara. 2020. Pop-up print: Rapidly 3D printing mechanically reversible objects in the folded state. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 58–70.
- Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoit Roman, and Mark Pauly. 2021. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.
- Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.
- Helmut Pottmann, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph.* 26, 3 (July 2007), 65–es. doi:10.1145/1276377.1276458
- Ahmad Rafsanjani and Damiano Pasini. 2016. Bistable auxetic mechanical metamaterials inspired by ancient geometric motifs. *Extreme Mechanics Letters* 9 (2016), 291–296.
- Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. 2006. Periodic global parameterization. *ACM Transactions on Graphics (TOG)* 25, 4 (2006), 1460–1485.
- Yingying Ren, Uday Kusupati, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, and Mark Pauly. 2022. Umbrella meshes: elastic mechanisms for freeform shape deployment. *ACM Transactions on Graphics* 41, 4 (2022), 1–15.
- Yingying Ren, Julian Panetta, Seiichi Suzuki, Uday Kusupati, Florin Isvoranu, and Mark Pauly. 2024. Computational Homogenization for Inverse Design of Surface-based Inflatables. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–18.

- Rohan Sawhney and Keenan Crane. 2017. Boundary first flattening. *ACM Transactions on Graphics (ToG)* 37, 1 (2017), 1–14.
- Emmanuel Siéfert, Etienne Reysat, José Bico, and Benoît Roman. 2019. Bio-inspired pneumatic shape-morphing elastomers. *Nature materials* 18, 1 (2019), 24–28.
- IM Stuart. 1961. Capstan equation for strings with rigidity. *British Journal of Applied Physics* 12, 10 (1961), 559.
- Bhivraj Suthar, Muhammad Usman, Hyunseok Seong, Igor Gaponov, and Jee-Hwan Ryu. 2018. Preliminary study of twisted string actuation through a conduit toward soft and wearable actuation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2260–2265.
- Seiichi Suzuki, Alison Martin, Yingying Ren, Tzu-Ying Chen, Stefana Parascho, and Mark Pauly. 2023. BamX: Rethinking Deployability in Architecture through Weaving. *Advances in Architectural Geometry 2023* (2023).
- Shinji Umeyama. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13, 04 (1991), 376–380.
- Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 2018. 4DMesh: 4D printing morphing non-developable mesh surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 623–635.
- Guanyun Wang, Yue Yang, Mengyan Guo, Kuangqi Zhu, Zihan Yan, Qiang Cui, Zihong Zhou, Junzhe Ji, Jiaji Li, Danli Luo, et al. 2023b. ThermoFit: thermoforming smart orthoses via metamaterial structures for body-fitting and component-adjusting. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–27.
- Liwei Wang, Yilong Chang, Shuai Wu, Ruike Renee Zhao, and Wei Chen. 2023a. Physics-aware differentiable design of magnetically actuated kirigami for shape morphing. *Nature Communications* 14, 1 (2023), 8516.
- Xudong Yang, Yu Chen, Tianyu Chen, Junwei Li, and Yifan Wang. 2024. Active Fabrics with Controllable Stiffness for Robotic Assistive Interfaces. *Advanced Materials* (2024), 2404502.
- Chao Yuan, Nan Cao, Xuejiao Ma, and Shengqi Dang. 2024. FreeShell: A Context-Free 4D Printing Technique for Fabricating Complex 3D Triangle Mesh Shells. *arXiv preprint arXiv:2407.19533* (2024).
- Zheng-Yu Zhao, Mo Li, Zheng Zhang, Qing Fang, Ligang Liu, and Xiao-Ming Fu. 2023. Evolutionary piecewise developable approximations. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14.
- Changxi Zheng, Timothy Sun, and Xiang Chen. 2016. Deployable 3D linkages with collision avoidance. In *Symposium on Computer Animation*, Vol. 179. 188.
- Afra Zomorodian and Gunnar Carlsson. 2004. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*. 347–356.