

# Zygote structure enables pluripotent shape-transforming deployable structure

 Yu-Ki Lee<sup>a,1</sup>, Yue Hao<sup>b,1</sup>, Zhonghua Xi<sup>b</sup>, Woongbae Kim<sup>c,d</sup>, Youngmin Park<sup>a</sup>, Kyu-Jin Cho<sup>c,d</sup>, Jyh-Ming Lien<sup>id b,\*</sup> and In-Suk Choi<sup>id a,\*</sup>
<sup>a</sup>Department of Materials Science and Engineering, Research Institute of Advanced Materials (RIAM), Seoul National University, Seoul 08826, Republic of Korea

<sup>b</sup>Department of Computer Science, George Mason University, Fairfax, VA 22030, USA

<sup>c</sup>Soft Robotics Research Center, Seoul National University, Seoul 08826, Republic of Korea

<sup>d</sup>Department of Mechanical and Aerospace Engineering, Institute of Advanced Machines and Design, Seoul National University, Seoul, Republic of Korea

 \*To whom correspondence should be addressed: Emails: [jmlie@cs.gmu.edu](mailto:jmlie@cs.gmu.edu); [insukchoi@snu.ac.kr](mailto:insukchoi@snu.ac.kr)
<sup>1</sup>Y.-K.L. and Y.H. contributed equally to this work.

Edited By: Derek Abbott

## Abstract

We propose an algorithmic framework of a pluripotent structure evolving from a simple compact structure into diverse complex 3D structures for designing the shape-transformable, reconfigurable, and deployable structures and robots. Our algorithmic approach suggests a way of transforming a compact structure consisting of uniform building blocks into a large, desired 3D shape. Analogous to a fertilized egg cell that can grow into a preprogrammed shape according to coded information, compactly stacked panels named the zygote structure can evolve into arbitrary 3D structures by programming their connection path. Our stacking algorithm obtains this coded sequence by inversely stacking the voxelized surface of the desired structure into a tree. Applying the connection path obtained by the stacking algorithm, the compactly stacked panels named the zygote structure can be deployed into diverse large 3D structures. We conceptually demonstrated our pluripotent evolving structure by energy-releasing commercial spring hinges and thermally actuated shape memory alloy hinges, respectively. We also show that the proposed concept enables the fabrication of large structures in a significantly smaller workspace.

**Keywords:** shape-programmable structure, *origami*, zygote structure, deployable structure, pluripotent evolving structure

## Significance Statement

*Origami*-inspired structures have enabled shape-transformable or deployable structures such as expandable solar cells in space. A more interesting question for such shape-transformable structures is how we can design a pluripotent structure that transforms into diverse shapes, analogous to the zygote in nature that changes its shape and size into a huge, complex living organism. We propose a computational approach for an *origami*-based zygote structure. Like an amino acid sequence organizes a complex protein, our algorithm finds a coded sequence that guides the simple zygote structure consisting of uniform panels into desired complex shapes. Combined with recent 4D printing or *origami*-robot techniques, our concept of the zygote structure can enable portable, deployable, or 3D-shaped devices and robots.

## Introduction

Nature-inspired shape-programmable structures have generated substantial interest in mathematics, applied physics, computer science and graphics, materials science, robotics, and biological engineering (1–7). Diverse structures in nature are sometimes constructed by a combination of basic units under simple, rational algorithms (8, 9). For example, complex patterns on leaves are composed of a single geometry and its repetition called a fractal, e.g. veins on a leaf branch off in the form of a self-similar geometry (10). The living organism also grows and evolves based on the combination of four units (i.e. DNA consisting of combination of four nucleotides). This coded information determines how a small zygote (a fertilized egg) proliferates and grows into the

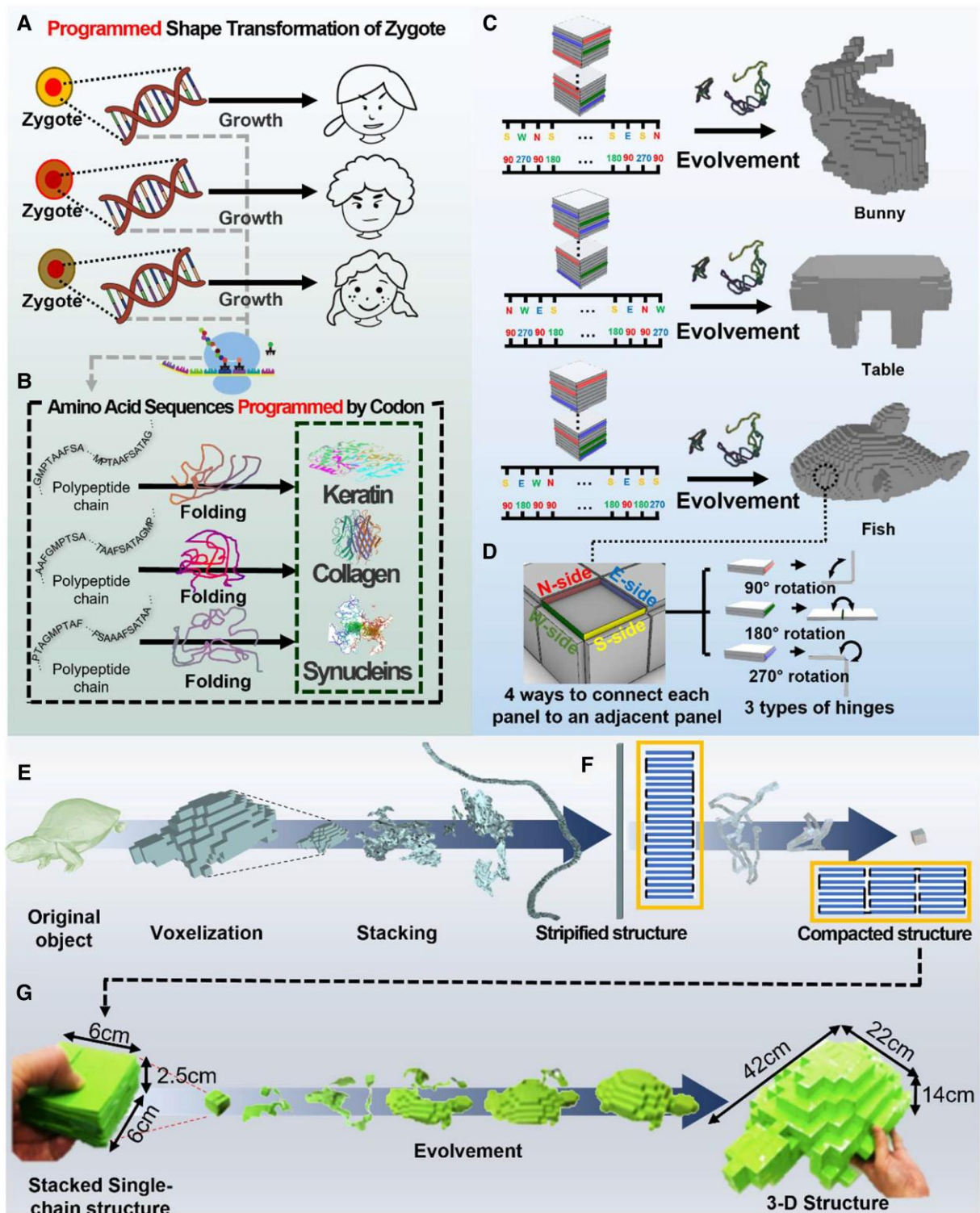
preprogrammed shape (Fig. 1A) (11–13). In addition, proteins with complex 3D structures are synthesized with a combination of only 20 amino acids. The preprogrammed information called codons in mRNA determines the sequence of the amino acids (Fig. 1B) (14–16). In other words, nature may consist of simple geometry and an algorithm determining their combination.

Analogous to these growing and evolving systems of nature, researchers have shown that the rational design of simple unit structures can realize shape-programmable structures (17, 18). One of popular strategies is *origami* or *kirigami* (art of paper folding and cutting) structures. For example, Cho and colleagues suggest a fractal cut pattern that can program the shape of a *kirigami*-inspired deployable structure with only

**Competing interest:** The authors declare no competing interest.

**Received:** September 27, 2022. **Revised:** January 12, 2023. **Accepted:** January 17, 2023

© The Author(s) 2023. Published by Oxford University Press on behalf of National Academy of Sciences. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



**Fig. 1.** Concept of the pluripotent evolving structure. A) A tiny zygote differentiates into a preprogrammed shape guided by coded information consisting of only four units (i.e. a DNA consisting of four types of nucleotides). Zygotes with different coded information grow into diverse shapes. B) Codons in mRNA program the shape of proteins, which is the sequential connection of only 20 amino acids. A polypeptide chain, which is a single chain in which twenty amino acids are connected in order of the preprogrammed sequence in the codon, transforms into its corresponding 3D structure (i.e. a protein). C) Analogous to these shape-programming systems in nature, our pluripotent evolving structure starts from stacked panels named the zygote structure, and it deploys into final structures guided by coded information. D) Our stacking algorithm inversely finds the connection path and folding angles of each panel in the zygote structure to transform it into diverse 3D shapes in a large space increasing as  $4^N \times 3^N$  according to the number of panels  $N$ . E) The stacking algorithm approximates a target 3D surface with square panels first and compactly stacks them with a single-strip structure by finding a Hamiltonian path on a dual graph on the target surface (see [Movie S1](#)). F) The single-chain structure can be further compacted by dividing the stripified panels into  $K$  components (see [Fig. S3](#)). G) By connecting paper panels according to this connection path, we could prepare a compacted structure and deploy it into a 3D turtle model, which has an almost 144 times larger scale.

two cut motifs and their hierarchical arrangement (19). Recent studies have focus on the computational design of such *origami* or *kirigami* structures (17, 20–24). For example, Choi and colleagues reported a computational inverse design method of *kirigami* tessellations by using an objective function that minimizes the deviations between *kirigami* tessellations and the target shapes under defined boundary constraints. With simple cut patterns on a planar sheet, they obtained various deployed 2D and 3D *kirigami* shapes (22). They further developed their inverse design algorithm with an additive approach that enables the creation of a large variety of rigid-deployable, compact reconfigurable *kirigami* patterns (24). Another strategy is developing rational algorithms that construct desired structure with building unit blocks, rather than starting from a planar sheet. However, most previous computational approaches (i) were based on the disassembly and re-assembly of unit blocks, (ii) allowed the variation of unit blocks (i.e. building blocks have different sizes or shapes), or (iii) had limited deployability (i.e. the ability to expand its scale) (25–29). For example, Luo and colleagues developed a computational framework named *legolization*, in which arbitrary 3D structures are constructed by assembling basic building blocks; however, their algorithm requires disassembly and re-assembly for shape reconfiguration that hardly incorporate with recent self-folding systems for automation (27). Zhou and colleagues proposed a computational design approach that can enable the reconfiguration of a structure consisting of singly connected boxes into another shape through a sequential folding process (i.e. without a disassembly process); however, they allowed variation of the unit blocks rather than uniform building blocks preferred in diverse and efficient shape programming (28). Yu and colleagues suggested an algorithm that can be used to construct arbitrary 3D shapes with a single-chain structure consisting of uniform building blocks. Their single-chain structure can transform into diverse 3D structures by means of layer-by-layer coiling and stacking under computational guidance (29). Nevertheless, their algorithm has limited deployability because their coiling algorithm fills extra blocks into the inside of the final structure.

In this study, we first propose an *origami* (the art of paper folding)-inspired algorithmic approach for both shape-programmable structures (i.e. a structure that can change its shape into the desired form) and deployable structures (i.e. a structure that can change its size from a compactly small scale to a significantly large scale) consisting of uniform building blocks. In contrast to *origami*-based modular robots, which have mostly focused on shape transformation from a 2D state to a 3D structure or from a 3D shape to another 3D shape of a similar size (30–32), we aim to develop a super-compact structure consisting of uniform building blocks that can transform into diverse large 3D shapes, analogous to a pluripotent shape-transformable system in nature. Similar to the natural system, our shape-programming framework of pluripotent evolving structure, named a *zygote* structure, consists of fully connected thin, uniform building blocks. Controlling its connection path with a computationally designed coded sequence, it can be deployed into the preprogrammed structure.

## Results and discussion

### Basic concept of *zygote* structure

Fig. 1C shows the basic concept of our pluripotent evolving structure called *zygote* structure. All the building blocks are uniform

square panels in our system. Starting from a given compactly stacked panels, we connect them according to the preprogrammed connection path (i.e. the computationally designed coded sequence) that can expand it into a desired 3D *origami* structure. A panel is physically connected with the upper or lower panel through one of its four sides by a rotational hinge. We constrained the rotation angles of the hinge to 90°, 180°, or 270° (Fig. 1D). In our simulation for the shape transformation, the hinges are designed to be stretchable to avoid the folding errors caused by thick panels (see Fig. S1). After connection, the stacked panels become a fully connected single-chain structure (which will be called a “stripified panels” hereinafter) that can be unfolded and reconfigured into a 3D *origami* structure. The compactly stacked structure can be transformed into diverse 3D structures according to the connection paths, as if the genetic codes represent certain physical features of creatures (11–13). In this framework, the number of possible connection paths increases exponentially with the number of panels  $N$  (proportional to  $4^N \times 3^N$ ) since each panel has to be connected to an upper or a lower panel through one of four sides and the hinge can have one of three angles (90°, 180°, or 270°). Then, the key question is how we can find the coded sequence for a specific shape efficiently in this large search space. The computer science community has developed answers to this problem (28, 29). Rather than searching for the stacked state in this large space, previous studies have focused on finding connection paths of panels tessellating the surface of 3D structures before compacting them. Similar to these previous approaches, our stacking algorithm finds the path that can compactly stack 3D structures to solve this inverse problem.

Fig. 1E shows our stacking algorithm that transforms a 3D structure into stacked panels in which all panels are connected in a single-strip structure. To achieve this, we first approximate a curved 3D surface with square panels via voxelization (33). Then, we represent the voxelized surface as a dual-graph  $G = \langle V, E \rangle$ , where each node in  $V$  corresponds to a square panel and each edge in  $E$  corresponds to a rotating hinge connecting the two neighboring panels. This inverse problem can be solved by finding the Hamiltonian path  $P$ , i.e. a path that visits all of the vertices on the given  $G$ , as illustrated in Fig. S2A. In general, the Hamiltonian path problem (HPP) is nondeterministic polynomial-time complete (NP-complete); however, Hamiltonian cycles must exist in the dual graph of the voxelized mesh as a four-regular graph (34). To solve this HPP, we used the traveling salesman problem (TSP) solver, which finds a Hamiltonian cycle, i.e. a path with identical start and end, with minimum cost (35). Once a Hamiltonian cycle is found, we break the cycle at an arbitrary position. This inverse programming can be operated in almost linear time by the number of panels (Fig. S2B shows the operation time for diverse 3D surfaces). Cut and folded along  $P$ , the tessellated 3D surface can be transformed into stripified panels and stacked into a columnar pile. The stripified panels can be more compactly stacked with multiple piles, as shown in Fig. 1F. The yellow boxes in Fig. 1F show the conceptual configuration of the compaction process (details for this compaction process are illustrated in Fig. S3). Finally, we inversely use  $P$  as the connection path for shape programming, i.e.  $P$  for stacking can inversely guide the compactly stacked panels to deploy into the target 3D structure. For example, Fig. 1G shows the experimental result that compactly stacked square sheets connected by the coded sequence can deploy into a 3D turtle structure that has a 144 times larger bounding box size. More importantly, we can deploy stacked panels into different 3D structures just by re-arranging their connection path by finding alternative or different Hamiltonian paths for diverse 3D



surfaces (Movie S1) as long as we can tessellate surface of a structure with the same number of panels.

In this aspect, we refer to given initial stacked panels as zygote structures to represent that this initial simple structure can deploy into diverse 3D structures under the guidance of the coded sequence, analogous to the zygote cell in an organism that is divided and grows into the preprogrammed shape. This computational design strategy can potentially contribute to the development of versatile, deployable, and shape-programmable structures, especially when combined with recent 4D-printing techniques or *origami* robots.

### Implementation of tree-stacking algorithm into zygote structure

In practical realization, stripified panels can result in significant inaccuracy for transforming the compactly stacked structure into the 3D surface (or vice versa) because even small noises in the motion control of the folding hinges can be cumulated along the connection-path and cause large gaps or overlaps in the final structure (see Fig. S1C) (36). This problem can be reduced by representing the zygote structure as a spanning tree  $T$  in the graph  $G$  rather than a single Hamiltonian path  $P$ . While all vertices in  $P$  are  $<2$  degree nodes (i.e. every panel can be physically connected with only two adjacent panels except the first and the last panel), the spanning tree  $T$  allows high-degree nodes (HDNs) with greater than three connections. Fig. 2A shows the conceptual configuration of the zygote structure in which panels are connected with a tree-path  $T$  rather than a single Hamiltonian path  $P$ ; panels in a pile are connected with a Hamiltonian path, and each pile is connected with hinges called bridges. The panels having bridges correspond to the HDNs in  $T$ .

Fig. 2B details this tree stacking using an example of the plane sheet approximated with  $12 \times 12$  quads (note that our stacking algorithm is valid for not only closed 3D structures but also unclosed structures such as a plane sheet). We first divide  $G$  with  $K$  subgraphs  $G_s$ . A Hamiltonian path  $P$  in  $G_s$  can potentially stack the corresponding panels into a single pile, i.e. the number of partitions determines the number of piles in the zygote structure. Multiple  $P$  in  $G_s$  increase the pluripotency of the zygote structure. One way to achieve this is to increase the number of edges,  $|E(G_s)|$ , within  $G_s$  since the diversity of finding  $P$  also increases due to higher inner connectivity. We also aim that all the piles have the same heights, i.e. each subgraph has the same number of vertices. This increases the deployability (i.e. volume expandability) of the pluripotent evolving structure. This partitioning problem can be considered as a balanced graph partitioning problem (37), which divides a graph into  $K$  components having an equal or similar size while minimizing the total weight of the edges connecting different components. In our system, all the weights of edges are set to 1. In this case, the generalized Fiduccia–Mattheyses algorithm can easily minimize the number of edges in the cut (38). However, the Fiduccia–Mattheyses algorithm starts from a random partition and optimizes it for the minimal cut length, so it does not guarantee an exactly equal number of vertices in  $K$  components and may result in unbalanced stacked panels (Fig. 2C) (39). To address this limitation, we repeat the Fiduccia–Mattheyses algorithm until it finds a balanced partition in which all components have the same number of nodes (Fig. 2D). Although this approach is a quite simple and brute-force alternative, this method was sufficient for partitioning models with few hundreds to thousands of meshes into  $<10$  components.

After partitioning, we consider the placement of piles to maximize the connectivity between  $G_s$ . We set a hypergraph  $G_{\text{hyper}}$  as a 2D grid. Each node  $v_{\text{hyper}}$  in  $G_{\text{hyper}}$  corresponds to a subgraph  $G_s^i$  and occupies a single grid cell of the 2D grid, as shown in Fig. 2E. When the number of edges in  $G$  between two subgraphs  $E(G_s^i, G_s^j) > 0$ , an edge  $e_{\text{hyper},i,j}$  connects  $v_{\text{hyper},i}$  and  $v_{\text{hyper},j}$  with the weight of the edge  $W(\text{hyper}_{i,j}) = E(G_s^i, G_s^j)$ . Then, we can find the optimal hypergraph  $G_{\text{hyper}}$  as

$$G'_{\text{hyper}} = \arg \max_{G_{\text{hyper}} \in \mathcal{g}} \sum W(e_{\text{hyper}})$$

This optimization problem can be solved by a dynamic-programming solver designed for subset sum problem (40).

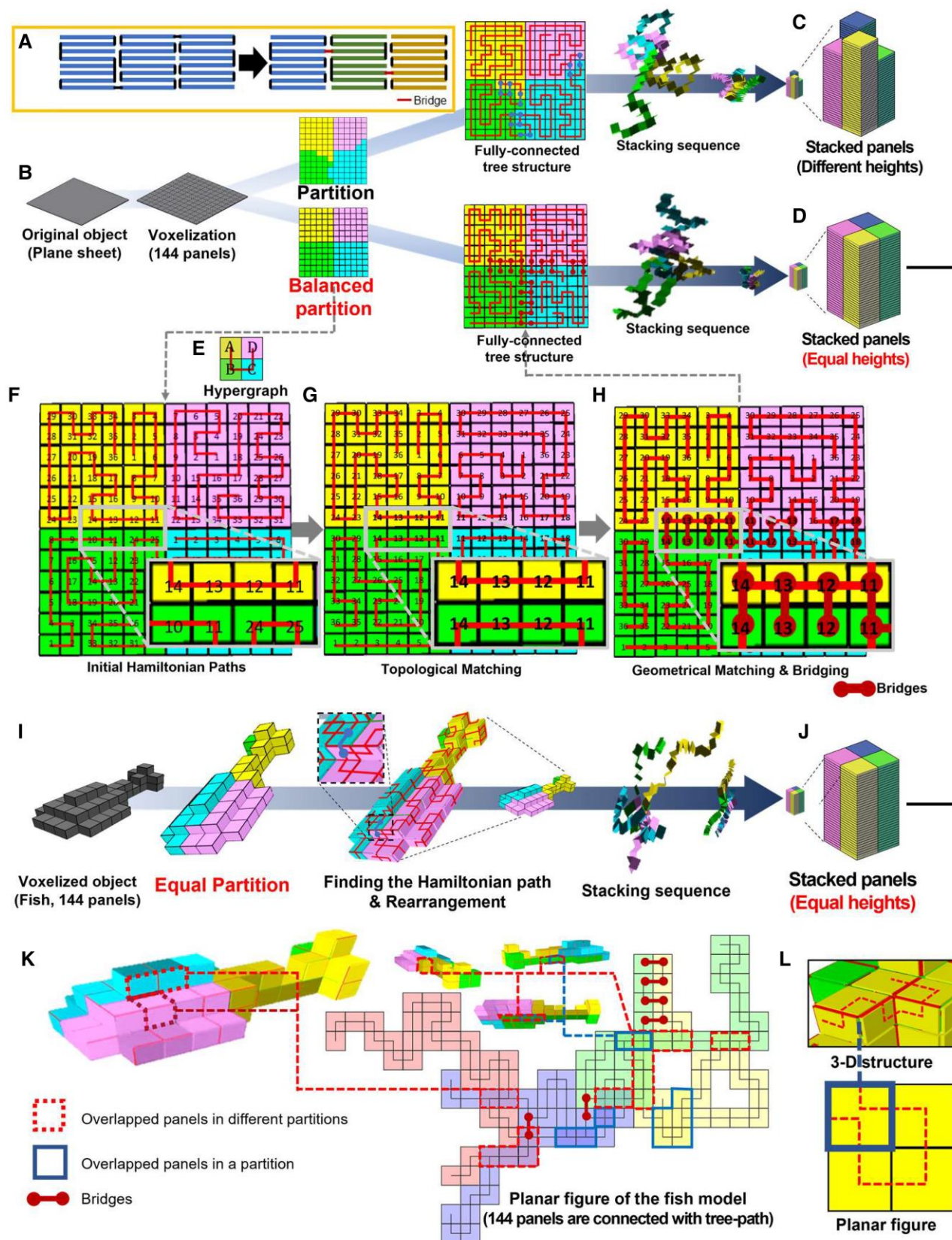
Then, each  $K$  component finds an arbitrary Hamiltonian path  $P^i$  (Fig. 2F). To construct a valid tree path, a Hamiltonian path  $P^i$  should be adjusted to be connectable with its adjacent Hamiltonian path  $P^j$  on both the 3D structure and zygote structure, i.e. at least two adjacent nodes in each neighboring Hamiltonian path on the 3D structure should also be adjacent in the stacked state. To enforce this requirement, we optimize  $P^i$  locally in each pile  $G_s^i$  to maximize connections between piles: (i) we select an arbitrary reference node  $v_{\text{hyper},r}$  in  $G'_{\text{hyper}}$ . Depending on their topological distance from  $v_{\text{hyper},r}$  in  $G_{\text{hyper}}$ , any child node  $v_{\text{hyper},i}$  is linked to its parent node  $v_{\text{hyper},j}$ . (ii) Let  $v_x^i$  be the  $x$ th node in  $P^i$ ; we place the stacked pile of  $P^i$ , which corresponds to  $v_{\text{hyper},i}$ , next to  $P^j$  with  $v_x^i$  and  $v_x^j$  being the bottom. Then, we recursively update the  $P^i$  based on  $P^j$  until there are some  $v_x^i$  in  $P^i$  that potentially be connected to node  $v_x^j$  in  $P^j$  (see Fig. S4). This matching function is defined as

$$f_{\text{match}}(x) = \begin{cases} 1, & \text{if the edge } v_x^i, v_x^j \text{ exists in } G \\ 0, & \text{otherwise} \end{cases}$$

which checks every pair of nodes  $v_x^i$  and  $v_x^j$  in  $P^i$  and  $P^j$  that are on the same height  $x$  in the stacked configuration. (iii) We update  $P^i$  using

$$P^i = \arg \max_{P^i \text{ in } G_s^i} \sum_{x=1}^{\min(H(P^i), H(P^j))} f_{\text{match}}(x) \cdot f_{\text{geo match}}(x)$$

where  $f_{\text{geo match}}(x)$  is the geometric matching function (Fig. 2G shows that  $P^B$  is updated to topologically match  $P^A$ . All  $P^i$  are recursively updated in this way.). If there is already a hinge connecting  $v_x^i$  and  $v_{x+1}^j$  where the potential hinge between  $v_x^i$  and  $v_x^j$  should be in the stacked configuration, they cannot be linked since connecting them will result in nonmanifold geometries, as shown in Fig. S5. If  $f_{\text{match}}(x)$  violates this constraint,  $f_{\text{geo match}}(x) = 0$ ; otherwise,  $f_{\text{geo match}}(x) = 1$ . (iv) To ensure this geometric matching constraint, we locally cut the  $P^i$  and change the orientation of  $v_x^i$ . For this process, we record all the  $x$ s that satisfy  $f_{\text{match}}(x) = 1$  and put in a sorted vector  $\vec{x}$  and ensure  $x_i < x_{i+1}$ . If  $|\vec{x}| = 1$ , which means that there is only one matching component, we can simply rotate the entire pile along with  $v_x^i$ . However, if  $|\vec{x}| > 1$ ,  $v_x^i$  has to be detached from  $v_{x-1}^i$  or  $v_{x+1}^i$ , i.e. we break  $P^i$ . We denote such a broken edge as  $e_{\text{break}}^i$ . Then,  $v_x^i$  and  $v_x^j$  can be connected by an edge  $e_{\text{match}}^i$  connecting the neighboring piles (Fig. 2H). We call such  $e_{\text{match}}^i$  as a “bridge.” In our pluripotent evolving structure, these HDNs having three neighbors, including the bridge, correspond to the high-degree panels (i.e. panels corresponding to the HDNs) having three hinges (i.e. a hinge corresponding to the bridge connects panels in two adjacent piles, while the others connect the upper or lower panels). Finally, we obtain a zygote structure in which all panels are con-



**Fig. 2.** The tree stacking algorithm with balanced partition. A) Rather than stripified panels corresponding to a Hamiltonian path, the tree-path structure reduces cumulative folding errors for nonzero thick panels and provides better load-bearing capabilities for practical application. B) For a given plane sheet structure approximated with  $12 \times 12$  square meshes, our tree-stacking algorithm partitions the graph on it using the Fiduccia–Mattheyses algorithm. C) However, the Fiduccia–Mattheyses algorithm does not guarantee an equal number of panels in each partition and results in nonuniform heights of piles in the zygote structure. D) To obtain the zygote structure in which all piles have the same heights for maximized pluripotency and

(continued)



nected in the tree path as

$$T_s = \cup P^i + \cup E_{\text{match}}^i - \cup E_{\text{break}}^i \text{ for all } i \text{ in } \{1, k\}$$

where  $E_{\text{match}}^i$  and  $E_{\text{break}}^i$  are the sets of all matching and broken edges in pile  $P^i$ . This tree-stacking algorithm effectively reduces the large gaps or overlaps in the final structure by preventing the accumulation of folding errors (see Fig. S6) (36).

The tree-path  $T_s$  generated by the tree-stacking process can inversely guide the zygote structures having the same configuration of panels into an arbitrary target 3D structure. For example, Fig. 2I shows that a voxelized 3D fish model approximated with 144 square meshes is stacked into a zygote structure with 4 piles, which is the same as the zygote structure corresponding to the plane model approximated with  $12 \times 12$  meshes (Fig. 2J and Movie S2). This implies that we can develop diverse structures from a single zygote structure by controlling only the connection path and folding angles of hinges guided by the coded sequences, i.e. the  $T_s$  can be used as a coded sequence for our pluripotent evolving structure.

It should be noted that at this stage, our tree-stacking algorithm does not include collision-free motion planning of the panels during transformation since our main objective is to find the connection paths that can guide the zygote structure into diverse 3D shapes. However, motion planning algorithm can be incorporated into our tree-stacking algorithm based on the previous works (36, 41). In addition, our tree-stacking algorithm aims for a different objective than recent polyhedron unfolding algorithms. In contrast to a polyhedron unfolding algorithm that flattens a 3D-meshed structure onto a 2D plane without self-overlapping, the panels may overlap if we flatten the zygote structure onto a 2D plane along the tree path (42, 43). If it is desired to create the flattened configuration and net a planar structure, we can require a 2D grid such that each node occupies a single grid cell, and all nodes are connected along  $T_s$ . For instance, in case of the  $T_s$  corresponding to the 3D fish approximated with 144 panels, there are grid cells that are occupied by more than two nodes, as shown in Fig. 2K (i.e. more than two panels can overlap in the flattened configuration). Even a single Hamiltonian path  $P$  can generate its self-overlapped planar figure, as shown in Fig. 2L. In this study, we do not consider self-collision or overlapped panels in flattened planar figures; therefore, all the possible connection paths obtained in the above process can be regarded as possible solutions. However, a global optimization problem with more restrictions could be established, such as minimizing the folding angles of hinges and the number of overlapping panels in flattened planar figure or avoiding self-collision during transformation.

## Demonstration of algorithmic zygote structure

Fig. 3 shows examples of our zygote structure. According to the coded sequence generated by the tree-stacking algorithm, a

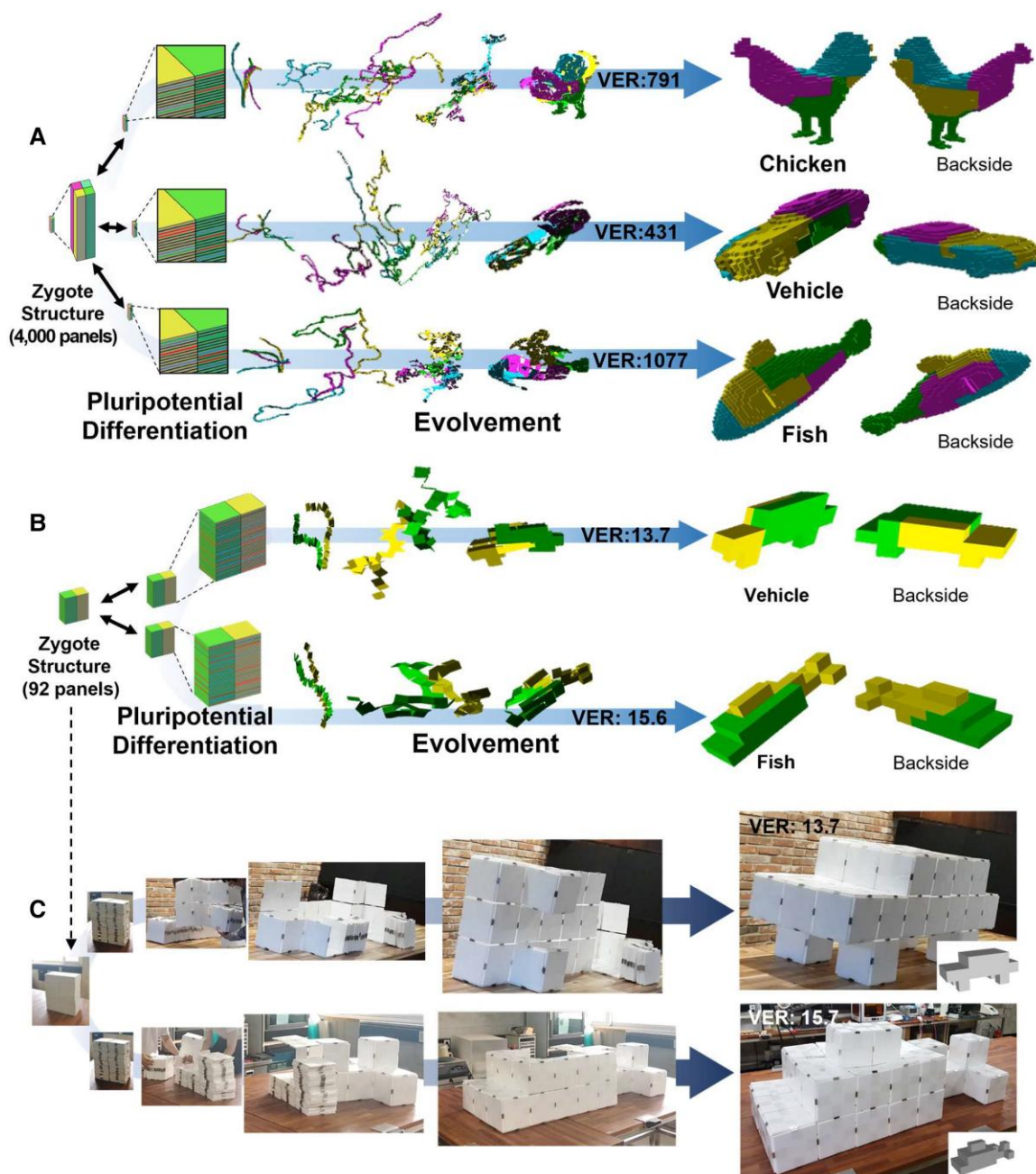
zygote structure having 4,000 panels with 4 piles can evolve (i.e. deploy) into a chicken, a vehicle, and a fish model (Fig. 3A and Movie S3. Fig. S7 shows another case using 498 panels). Besides pluripotency, scale expandability is another advantage of the pluripotent evolving structure. It is obvious that the size of the evolved structure is related to the thickness of each panel. To quantitatively compare the volume compaction/expansion ratio, we used package volume, which is widely used in the shipping industry (44). Package volume measures the cuboid box size to compactly pack an object. In other words, the package volume expansion ratio (referred to as “VER” hereinafter), defined as  $\frac{W_e \times L_e \times H_e}{W_s \times L_s \times H_s}$ , measures the ratio of bounding box volume (Length  $\times$  Width  $\times$  Height) between the zygote and evolved 3D structures (see Fig. S8). When the length of square panel  $L$  is 100 times the thickness  $t$  (i.e.  $L = 100 \times t$ ), the VER is 791, 431, and 1,077 for each chicken, vehicle, and fish model, respectively.

The number of panels (i.e. resolution) also affects VER. In general, the feasibility of tree stacking for high-resolution models (i.e. 3D models consisting of many panels) is determined by the number of partitions rather than the number of panels. More partitions are much more expensive to compute, as each pile must have at least one HDN, or the entire algorithm must restart. In addition, the equal size partitioning process is also expensive for more partitions since we must repeat the Fiduccia–Mattheyses algorithm. When we increased the number of piles to  $>6$  for the 4,000 panels case, the algorithm fails to find the solution. Meanwhile, our implementation using the state-of-the-art TSP solver Concorde (35) could find  $P_s$  in each partition efficiently so that our pluripotent evolving structure can have high-resolution 3D models if we restrict the number of partitions.

For experimental demonstration, we generate a pluripotent evolving structure with 92 panels (Fig. 3B and Movie S4). A zygote structure with 2 piles can evolve into a vehicle having 13.7 VER or a fish having 15.6 VER when the length of square panel  $L$  is 20 times the thickness  $t$  (i.e.  $L = 20 \times t$ ). We prepared 92 plastic panels (20 cm  $\times$  20 cm  $\times$  1 cm size), triangular magnets for  $90^\circ$  and  $270^\circ$  rotating hinges, and cuboid magnets for  $180^\circ$  rotating hinges (Fig. S9 for detailed information). We attached the magnets on the side faces of the plastic panels according to the coded sequence of the 3D vehicle model. Then, we could easily transform it into the 3D vehicle guided by the coded sequence generated by the tree-stacking algorithm. Although only four panels meet the ground (wheels of the vehicle) and there are no internal supports, the final vehicle structure stably maintains its shape without collapse. The 3D vehicle can be folded back to the initial zygote state, and we can transform it again into the 3D fish model just by rearranging the magnets (i.e. simply detaching the magnets and reattaching them) according to its coded sequence (Fig. 3C and Movie S5). Fig. S9 shows a further demonstration with magnetic toy blocks or paper boxes.

**Fig. 2.** (Continued)

deployability (i.e. volume expandability), we simply repeated this partition process until it generated a uniformly balanced partition. E) Each partition is stacked into a single pile, and each pile is connected with one of its adjacent piles to construct a fully connected tree-path. For this process, we set a hypergraph that determines the placement and connection between piles on a 2D grid. F) Then, we find a Hamiltonian path on each partition. The number on panels shows the sequence of each panel in the Hamiltonian path (corresponding to the height in the stacked configuration), i.e. the numbers on the panels indicate the sequence of vertices  $v_x$ , i.e. a panel having number  $X$  corresponds to the  $x$ th node in  $P^i$ . G) The Hamiltonian paths are adjusted to make a bridge that can connect two neighboring piles in both unfolded 3D and stacked configuration (Hamiltonian path on partition B is adjusted and the 11th to 14th panels in A and B partitions become neighbors in both 3D and stacked configuration). They are connectable with hinges called bridges. H) Finally, the geometric matching process checks potential panels having double hinges on a single side caused by the bridge and locally adjusts it to a valid connection path (see Figs. S4 and S5 for further description). I) When we apply this process for the 3D fish model having the same number of panels (144 panels), (J) it generates the same zygote structure, which implies that a zygote structure can deploy in both the plane sheet structure or the fish structure just by controlling their connection path. (Movie S2 shows the tree stacking sequence). In our tree-stacking algorithm, we allow the overlapping of its unfolded figure. Both overlapped panels (K) between different partitions or (L) in a partition can be seen in the flattened configuration of the tree-path.

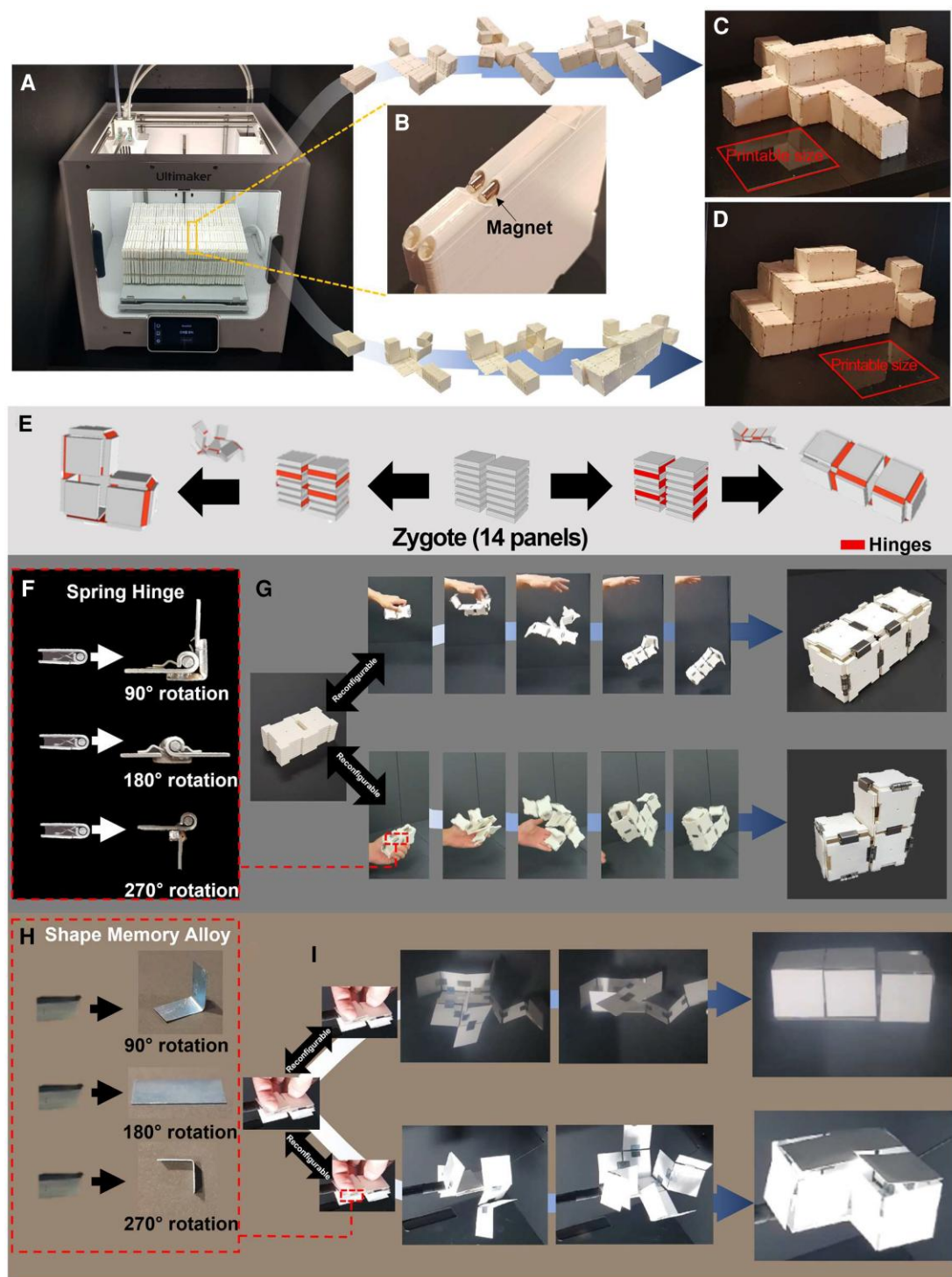


**Fig. 3.** Algorithmic results of the pluripotent evolving structure and its demonstration with magnets. A) The coded sequences generated by the tree-stacking algorithm can guide the zygote structure (i.e. compactly stacked panels) into diverse 3D structures, such as a chicken, a vehicle, and a fish. When the length of each panel is 100 times the thickness ( $L = 100 \times t$ ), the package volume expansion ratio (VER) is 791, 437, and 1,077 for each model (see Fig. S8 for a further explanation of VER). B) We simply demonstrated our pluripotent evolving structure with 92 panels. A zygote structure can evolve (i.e. deploy) into a vehicle and fish model, each having 13.7 and 15.6 of VER ( $L = 20 \times t$ ) according to the coded sequences. C) We could simply demonstrate it with plastic panels by attaching triangular or rectangular magnets on the side of the panels (see Movie S5). The final structures were robust and reliable without collapse, especially even for the vehicle structure in which only four panels were placed on the ground.

### Practical implications of zygote structure

We conceptually show that our zygote structure can be used as the fabrication platform for overcoming the limited working space. Fabrication of a large structure may require a large area larger than the final product, e.g. making a large structure difficult and sometimes impossible without assembly due to limited workspace area or volume. Deployable structures that can overcome these limitations are suggested in diverse research fields involving architecture and robotics; however, most of them consist of bars

and construct open scaffold structures (45–47). In contrast to previous deployable structures, our pluripotent evolving structure can represent the external shape of the desired structures with thin panels. Fig. 4A–D illustrates that a 3D printed zygote structure having 92 panels with 2 piles can evolve into objects that are significantly larger than the printable size of the 3D printer. Our 3D printed panel has a size of  $9 \text{ cm} \times 9 \text{ cm} \times 0.48 \text{ cm}$  and has holes in its rounded side so that it can embed a cylindrical magnet with a diameter of 2 mm and length of 3 cm, as shown in



**Fig. 4.** Self-transformable pluripotent evolving structure and a conceptual demonstration of overcoming the limited fabrication space using the pluripotent evolving structure. A) Our pluripotent evolving structure enables overcoming the limited workspace since it can be fabricated in a compacted zygote structure and evolve (i.e. deploy) into huge structures. For example, we printed a zygote structure with 92 panels  $23\text{ cm} \times 18\text{ cm} \times 9\text{ cm}$  in size. B) Each panel ( $9\text{ cm} \times 9\text{ cm} \times 0.48\text{ cm}$  size) can embed cylindrical magnets in its holes in its rounded sides (see Fig. S10). Without an additional reconfiguration process (i.e. detaching and reattaching hinges), (C) the printed pluripotent evolving structure evolves into an airplane model ( $81\text{ cm} \times 72\text{ cm} \times 18\text{ cm}$ ,  $\text{VER} = 31.3$ ) is folded back into the zygote structure, and (D) then evolves again into a fish model ( $63\text{ cm} \times 27\text{ cm} \times 27\text{ cm}$ ,  $\text{VER} = 15.6$ ). Both 3D structures have more than 12 times or 5 times larger package volume sizes than the printable size of the commercial 3D print ( $23\text{ cm} \times 19\text{ cm} \times 20\text{ cm}$ ). E) We also demonstrate self-deployable pluripotent evolving structure with a zygote structure having 14 panels with two piles that can evolve into L- or I-shaped 3D structures without self-collision. F) We prepared three types of mechanical hinges that rotate  $90^\circ$ ,  $180^\circ$ , or  $270^\circ$  by welding a steel bar on commercial spring hinges. G) Attached the spring hinges according to the coded sequences, the zygote structure rapidly evolves into the L-shaped structure when we simply throw it into the air. It is also possible to fold back into the zygote state and reconfigure it by rearranging hinges. Then, it transforms it into an I-shaped structure in the same manner (see Fig. S12 and Movie S6). H) We also showed that stimuli-responsive materials can be introduced in our pluripotent evolving structure. The SMA sheets are preprogrammed to be folded at  $90^\circ$ ,  $180^\circ$ , or  $270^\circ$ . I) By attaching them to the stacked papers according to the coded sequences, we could realize a stimuli-responsive self-evolving structure (see Fig. S13 and Movie S7).



Fig. 4B. These cylindrical magnets are used as universal hinges for all 90°, 180°, or 270° rotations, which enables shape reconfiguration without rearranging the hinges (i.e. without detaching and reattaching the hinges in the zygote state). In addition, our panel design enables inserting magnets even inside of the printed zygote structure (Fig. S10). Since all holes aligned along the four sides of each panel are exposed in the stacked configuration, we can insert magnets as the compactly printed configuration. According to the algorithmic result, we printed a zygote structure with 92 panels and 2 piles (Fig. 4A). For printing, we introduced small gaps between panels (0.2 mm) to prevent their adhesion. The final printed zygote structure has a size of 23 cm × 18.2 cm × 9 cm. Then, we inserted the cylindrical magnets in all of the holes of the panels (total 368 magnets). As shown in Fig. 4C, it could evolve (i.e. deploy) into an 81 cm × 72 cm × 18 cm airplane model with 31.3 VER by folding panels along its corresponding coded sequence. It could be folded back into the zygote state without an additional detaching process and could evolve again into a 63 cm × 27 cm × 27 cm size fish model (Fig. 4D). Both 3D structures have larger dimensions than the printable size (23 cm × 19 cm × 20 cm, marked in red in Fig. 4C and D). We acknowledge that the pluripotent evolving structure with higher resolution may practically result in unstable or floppy structures because it deploys into hollow 3D structures; however, this issue can practically be mitigated by adding additional hinges connecting the cut edges only at the target state or adding internal support. For robust final structures, it may potentially be integrated with internal folding struts inside the unit panels.

Our demonstration with a 3D printer further implies that our zygote structures can be incorporated with recent 4D-printing techniques (48–50) or *origami* robots (51–53). The zygote structure can easily incorporate these self-transformation systems because thin panels as building blocks have advantages in terms of light weight and integration with novel stimuli-responsive materials than volumetric building blocks (17, 52–56). To integrate the *origami*-robot techniques in our zygote structure, we introduce energy-releasing spring hinges and thermally actuated shape memory alloy (SMA) hinges in our pluripotent evolving structure. We are the first to demonstrate a shape-reconfigurable structure that can transform into multiple targets starting from a compact structure consisting of a thin, uniform panel. Figure 4E shows a simple zygote structure having 14 panels with 2 piles that can be deployed into an L- or an I-shaped 3D structure (Fig. S11 shows the flattened planar figures of both 3D structures with mountain-valley fold lines). We experimentally demonstrate the deployable zygote structure by assembling 3D printed 14 panels with commercial spring hinges; their rotation angles were controlled to be 90°, 180°, or 270° by welding steel bars on the spring hinges (Fig. 4F). The spring hinges can be inserted into one of four sides of the 3D printed panels (Fig. S12). First, we connected stacked panels with the spring hinges according to the coded sequence for the I-shape structure. Simply thrown into the air, it rapidly deploys into the final 3D structure, and the unfolded structure maintains its shape without additional supports. Then, we could transform it to the zygote structure easily by folding back guided by the hinges and detaching them and reconfiguring it into the L-shape structure in the same way (Fig. 4G and Movie S6). This demonstration using spring hinges enables fast and robust shape transformation, while recent stimuli-responsive self-folding systems experience delays caused by phase transformation or alignment of materials (17).

We further demonstrate that our pluripotent evolving structure can be combined with recent self-folding systems based on novel stimuli-responsive materials. Most of these systems are

composed of functionalized or layered thin sheet-like structures (54–56). In other words, whether incorporating thin sheet-like hinges is possible is the key criterion to test the potential feasibility of introducing stimuli-responsive hinges. To demonstrate this, we simply introduce SMA sheets into our pluripotent evolving structure. We prepared 5 mm × 10 mm thin SMA sheets. They are folded to 90° and fixed, annealed at 450°C for an hour, and quenched in water. This process preprograms the shape of SMA sheets. Overlapping the programmed SMA by folding it in half with either 90° or 270°, we could prepare SMA hinges that can rotate and stop either at 90° or at 270°. Hinges for 180° rotation are prepared in the same manner (Fig. 4H). Detailed process is described in Fig. S13). Then, we prepared two zygote structures consisting of 14 pieces of paper and attached SMA hinges to each zygote structure guided by the coded sequence for each I- or L-shaped structure. Fig. 4I and Movie S7 show that the pluripotent evolving structure can deploy into the two 3D structures when we apply heat by a commercial heat gun. Stimuli-responsive materials have enabled diverse shape-transformable structures, but its integration with a computationally guided design system as in our study has been rarely reported.

In this study, we simply demonstrate a pluripotent evolving structure consisting of 14 panels with spring hinges or SMA hinges. Although physically realizing a zygote structure consisting of many panels with such stimulus-responsive hinges is currently challenging, further development of mechanical and electrical dynamic systems for hinges and panels, such as recent hydraulic- or motor-based moving hinges and MEMS panels (57–59), may enhance the structural stability during shape transformation. In addition, we believe that recent *origami* robots that embed circuit boards, sensors, or batteries can be applied to realize a functionalized shape-programmable *origami* robot (51–53).

## Summary

We proposed the concept of the pluripotent evolving structure for designing a shape-transformable, reconfigurable, and deployable structure. Computational inverse design using our tree-stacking algorithm enables finding the connection paths that inversely guide compactly stacked uniform panels called the zygote structure into an arbitrary 3D structure. We demonstrated that our framework provides a shape-programmable structure with both high pluripotency and deployability. For example, we showed that this inverse design approach could transform a compact zygote structure having 4,000 panels into a chicken, a vehicle, and a fish model having 791-, 431-, and 1,077-times package volume sizes than their initial state. Thin building blocks in our zygote structure not only lead to extremely high deployability (i.e. volume expandability) but also potentially bring better feasibility for self-transformation combined with diverse self-folding mechanisms based on thin-film materials. We experimentally demonstrated the self-transformation of our pluripotent evolving structure with commercial spring hinges and SMA sheets. This further implies that our self-transformation system can be further incorporated with recent 4D-printing systems. In addition, it has versatile application rather than previous shape-programmable structures since conventional functionalized thin materials or fabrication processes for thin-film materials can be easily incorporated into the thin, uniform unit panels of our pluripotent evolving structure. One potential application is the design of wearable devices and robots, in which our pluripotent evolving structure provides tight conformability to the underlying shapes or human body. Recent autonomous *origami* robots embedding

circuit boards or microcontroller can be applied to achieve it (51–53). In these aspects, our concept for pluripotent evolving structures not only contributes to the development of rational algorithms for shape-programmable structures in computer science and graphics, materials engineering, and architectures but also provides new insights into the development of portable, deployable, or 3D-shaped devices and robots in engineering fields involving materials engineering, electronic engineering, aerospace engineering, and robotics.

## Materials and methods

We implemented the proposed stacking algorithm in C++. All data, including the running time illustrated in the [supplementary materials](#), are collected on a MacBook Pro with a 2.5 GHz Intel Core i7 CPU with 16 GB memory running macOS 10.12. For the 3D-printed panels, we used commercial polylactic acid filaments. For the experimental demonstration with the SMA sheets, we used 0.125 mm thick nitinol foil (purchased from AVENTION Co., Ltd.)

## Supplementary material

[Supplementary material](#) is available at PNAS Nexus online.

## Funding

This work was supported by the National Research Foundation of Korea (NRF; grant numbers 2019R1A2C2003430, 2018M3A7B408 9670, and 2016R1A5A1938472), the Creative-Pioneering Researchers Program through Seoul National University, and LG Display under the LGD–Seoul National University Incubation Program.

## Author contributions

I.-S.C. conceived the concept. Y.H, Z.X, and J.-M.L. developed the stacking algorithms. Y.-K.L. and Y.H. generated the 3D models used in this paper and manipulated and implemented the algorithm. Y.-K.L., Y.P., and W.K. carried out experimental demonstrations. J.-M.L. and I.-S.C. supervised the project. Y.-K.L., Y.H., Z.X., K.-J.C., J.-M.L., and I.-S.C. wrote the paper.

## Data availability

All data needed to evaluate the conclusions in the paper are presented in the paper and/or the [supplementary materials](#). This manuscript was posted on a preprint: <https://doi.org/10.48550/arXiv.2208.04204>

## References

- Baek S-M, Yim S, Chae S-H, Lee D-Y, Cho K-J. 2020. Ladybird beetle-inspired compliant origami. *Sci Robot*. 5:eaa26262.
- Lendlein A, Balk M, Tarazona NA, Gould OEC. 2019. Bioprospectives for shape-memory polymers as shape programmable, active materials. *Biomacromolecules* 20:3627–3640.
- Baik S, et al. 2019. Bioinspired adhesive architectures: from skin patch to integrated bioelectronics. *Adv Mater*. 31:e1803309.
- Liu Y, He K, Chen G, Leow WR, Chen X. 2017. Nature-inspired structural materials for flexible electronic devices. *Chem Rev*. 117:12893–12941.
- Rafsanjani A, Zhang Y, Liu B, Rubinstein SM, Bertoldi K. 2018. Kirigami skins make a simple soft actuator crawl. *Sci Robot*. 3: eaar7555.
- Yang Y, et al. 2016. Development of a bio-inspired transformable robotic fin. *Bioinspir Biomim*. 11:056010.
- Hamley IW. 2019. Protein assemblies: nature-inspired and designed nanostructures. *Biomacromolecules* 20:1829–1848.
- Samadikuchaksaraei A, Lecht S, Lelkes PI, Mantalaris A, Polak JM. 2014. Stem cells as building blocks. In: *Principles of tissue engineering*. Academic Press. p. 41–55.
- Kalve S, De Vos D, Beemster GT. 2014. Leaf development: a cellular perspective. *Front Plant Sci*. 5:362.
- Yi Peng WL, Wang N, Tian Y, Chen X. 2013. A novel wick structure of vapor chamber based on the fractal architecture of leaf vein. *Int J Heat Mass Transf*. 63:120–133.
- Ahmad M, Jung LT, Bhuiyana A-A. 2017. From DNA to protein: why genetic code context of nucleotides for DNA signal processing? A review. *Biomed Signal Process and Control*. 34:44–63.
- Soman KP. 2010. *Insight into wavelets: from theory to practice*. Boston: Academic Press.
- Sarkar S. 1996. Decoding coding: information and DNA. *Bioscience* 46:857–864.
- Wei G-W. 2019. Protein structure prediction beyond AlphaFold. *Nat Mach Intell*. 1:336–337.
- Strachan T, Read A. 1999. An overview of mutation, polymorphism, and DNA repair. In: *Human molecular genetics*, Vol. 3. New York: GarlandScience.
- Stanke SWM. 2003. Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics* 19:ii215–ii225.
- Lee Y-K, Kim J, Lien J-M, Lee Y-J, Choi I-S. 2021. Recent progress in shape-transformable materials and their applications. *Electron Mater Lett*. 18:215–231.
- Bertoldi K, Vitelli V, Christensen J, van Hecke M. 2017. Flexible mechanical metamaterials. *Nat Rev Mater*. 2:1–11.
- Cho Y, et al. 2014. Engineering the shape and structure of materials by fractal cut. *Proc Natl Acad Sci U S A*. 111:17390–17395.
- Walker A, Stankovic T. 2022. Algorithmic design of origami mechanisms and tessellations. *Commun Mater*. 3(1):1–8.
- Xiao K, Liang Z, Zou B, Zhou X, Ju J. 2022. Inverse design of 3D reconfigurable curvilinear modular origami structures using geometric and topological reconstructions. *Nat Commun*. 13(1):1–9.
- Choi G, et al. 2019. Programming shape using kirigami tessellations. *Nat Mater*. 18(9):999–1004.
- Choi GP, Dudte LH, Mahadevan L. 2021. Compact reconfigurable kirigami. *Phys Rev Res*. 3(4):043030.
- Dudte LH, et al. 2022. An additive framework for kirigami design. *arXiv*. 01810. <https://doi.org/2207.01810>, preprint: not peer reviewed.
- Duncan N, Yu L-F, Yeung S-K. 2016. Interchangeable components for hands-on assembly based modelling. *ACM Trans Graph*. 35:1–14.
- Song P, Fu C-W, Cohen-Or D. 2012. Recursive interlocking puzzles. *ACM Trans Graph*. 31:1–10.
- Luo S-J, et al. 2015. Legolization. *ACM Trans Graph*. 34:1–12.
- Zhou Y, Sueda S, Matusik W, Shamir A. 2014. Boxelization: folding 3D objects into boxes. *ACM Trans Graph*. 33(4):1–8.
- Yu M, Ye Z, Liu Y-J, He Y, Wang CCL. 2019. Lineup: computing chain-based physical transformation. *ACM Trans Graph*. 38:1–16.
- Rus D, Vona M. 2001. Crystalline robots: self-reconfiguration with compressible unit modules. *Auton Robots*. 10(1):107–124.
- Yim M, Duff DG, Roufas KD. 2000. Polybot: a modular reconfigurable robot. *IEEE International Conference on Robotics and Automation*; IEEE, San Francisco.
- Belke CH, Paik J. 2017. Mori: a modular origami robot. *IEEE/ASME Trans Mechatron*. 22(5):2153–2164.

- 33 Cohen-Or D, Kaufman A. 1995. Fundamentals of surface voxelization. *Graphical Models Image Process.* 57:453–461.
- 34 Lemus E, Bribiesca E, Garduño E. 2014. Representation of enclosing surfaces from simple voxelized objects by means of a chain code. *Pattern Recognit.* 47(4):1721–1730.
- 35 Applegate D, Bixby R, Chvatal V, Cook W. 2006. Concorde TSP solver. <http://www.math.uwaterloo.ca/tsp/concorde.html>.
- 36 Hao Y, Lien J-M. 2019. Compacting voxelized polyhedra via tree stacking. *Comput Graph Forum.* 38:323–333.
- 37 Andreev K, Racke H. 2006. Balanced graph partitioning. *Theory Comput Syst.* 39(6):929–939.
- 38 Sanchis LA. 1989. Multiple-way network partitioning. *IEEE Trans Comput.* 38(1):62–81.
- 39 Sheblaev MV, Sheblaeva AS. 2018. A method of improving initial partition of Fiduccia–Mattheyses algorithm. *Lobachevskii J Math.* 39:1270–1276.
- 40 Schnorr C-P, Euchner M. 1994. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math Program.* 66(1):181–199.
- 41 Xi Z, Lien J-M. 2015. Continuous unfolding of polyhedra—a motion planning approach. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany.
- 42 Lee Y-K, et al. 2020. Computational wrapping: a universal method to wrap 3D-curved surfaces with nonstretchable materials for conformal devices. *Sci Adv.* 6(15):eaax6212.
- 43 Xi Z, Kim Y-H, Kim YJ, Lien J-M. 2016. Learning to segment and unfold polyhedral mesh from failures. *Comput Graph.* 58:139–149.
- 44 Garber LL, Hyatt EM, Boya ÜÖ. 2009. The effect of package shape on apparent volume: an exploratory study with implications for package design. *J Mark Theory Pract.* 17:215–234.
- 45 Coelho MT, Linder S, Reches N. 2013. Hyperform—folding strategies for 3D printing. <http://www.cmarcelo.com/hyperform>
- 46 Ryan KR, Down MP, Banks CE. 2021. Future of additive manufacturing: overview of 4D and 3D printed smart and advanced materials and their applications. *Chem Eng J.* 403:126162.
- 47 Ioan D, Ioan-Alexandru R. 2014. Deployable structures for architectural applications—a short review. *Appl Mech Mater.* 658: 233–240.
- 48 Kuang X, et al. 2019. Advances in 4D printing: materials and applications. *Adv Funct Mater.* 29:1805290.
- 49 Momeni F, Hassani N, SMM, Liu X, Ni J. 2017. A review of 4D printing. *Mater Des.* 122:42–79.
- 50 Tibbits S. 2014. 4D Printing: multi-material shape change. *Archit Des.* 84:116–121.
- 51 Zhakypov Z, Mori K, Hosoda K, Paik J. 2019. Designing minimal and scalable insect-inspired multi-locomotion millirobots. *Nature* 571:381–386.
- 52 Nisser MEW, Felton SM, Tolley MT, Rubenstein M, Wood RJ. 2016. Feedback-controlled self-folding of autonomous robot collectives. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, IEEE.
- 53 Lee D, Saito K, Umedachi T, Ta TD, Kawahara Y. 2018. Origami robots with flexible printed circuit sheets. Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, Singapore.
- 54 Santangelo CD. 2017. Extreme mechanics: self-folding origami. *Ann Rev Condens Matter Phys.* 8:165–183.
- 55 Koh J-S, Kim S-R, Cho K-J. 2014. Self-folding origami using torsion shape memory alloy wire actuators. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 46377; American Society of Mechanical Engineers, Stockholm, Sweden.
- 56 Na J-H, et al. 2015. Programming reversibly self-folding origami with micropatterned photo-crosslinkable polymer trilayers. *Adv Mater.* 27:79–85.
- 57 Rus D, Tolley MT. 2018. Design, fabrication and control of origami robots. *Nat Rev Mater.* 3:101–112.
- 58 Liu C, Orlofsky A, Kitcher CD, Felton SM. 2019. A self-folding pneumatic piston for mechanically robust origami robots. *IEEE Robot Autom Lett.* 4:1372–1378.
- 59 Sun X, Felton SM, Niiyama R, Wood RJ, Kim S. 2015. Self-folding and self-actuating robots: A pneumatic approach. IEEE International Conference Robotics Automation (ICRA), Seattle, Washington, USA, IEEE.