

# Scalable sim-to-real transfer of soft robot designs

Sam Kriegman<sup>1</sup>, Amir Mohammadi Nasab<sup>2</sup>, Dylan Shah<sup>2</sup>, Hannah Steele<sup>2</sup>, Gabrielle Branim<sup>2</sup>,  
Michael Levin<sup>3</sup>, Josh Bongard<sup>1</sup>, Rebecca Kramer-Bottiglio<sup>2</sup>

<sup>1</sup>University of Vermont, <sup>2</sup>Yale University, <sup>3</sup>Tufts University

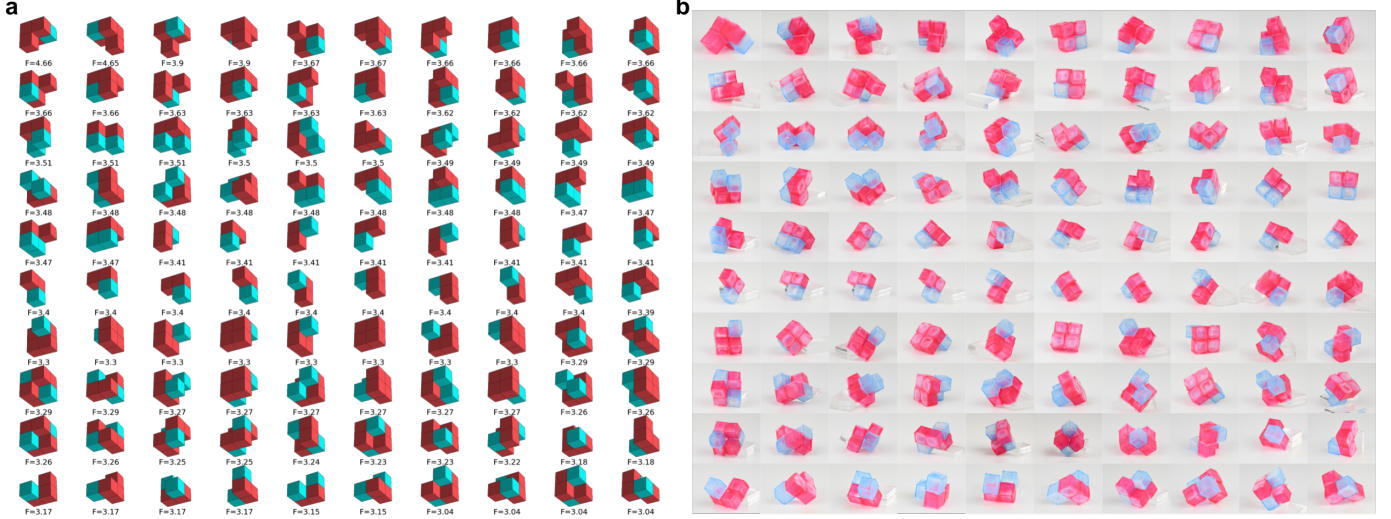


Fig. 1. The top 100 simulated 2-by-2-by-2 configurations of passive (cyan) and volumetrically-actuating (red) voxels (a) were manufactured in reality (b).

**Abstract**—The manual design of soft robots and their controllers is notoriously challenging, but it could be augmented—or, in some cases, entirely replaced—by automated design tools. Machine learning algorithms can automatically propose, test, and refine designs in simulation, and the most promising ones can then be manufactured in reality (sim2real). However, it is currently not known how to guarantee that behavior generated in simulation can be preserved when deployed in reality. Although many previous studies have devised training protocols that facilitate sim2real transfer of control policies, little to no work has investigated the simulation-reality gap as a function of morphology. This is due in part to an overall lack of tools capable of systematically designing and rapidly manufacturing robots. Here we introduce a low cost, open source, and modular soft robot design and construction kit, and use it to simulate, fabricate, and measure the simulation-reality gap of minimally complex yet soft, locomoting machines. We prove the scalability of this approach by transferring an order of magnitude more robot designs from simulation to reality than any other method. The kit and its instructions can be found here: [github.com/skriegman/sim2real4designs](https://github.com/skriegman/sim2real4designs)

## I. INTRODUCTION

The simulation-reality gap<sup>1</sup> for rigid-bodied robots is steadily closing. Computational models of rigid body dynamics can now be regularized and tuned so that control policies optimized in simulation are just as successful when tested on the physical system [2, 11]. The reality gap for soft robots, on the other hand, remains uncharted. It could be wider than the gap for rigid bodies, or not. Soft bodies are more challenging to accurately simulate, design, and precisely

control. But, they are also, by definition, more permissive to simulation inaccuracies, design flaws, and control precision: A soft gripper or foot will passively conform to complex objects and terrain, reducing the burden on the simulator to perfectly capture any single, “true” surface contact geometry.

Quantifying which soft robot designs, policies and behaviors can be faithfully simulated is critical not only for robotics, but also synthetic approaches to understand functional plasticity of biological systems during development and regeneration. For both domains, testing candidate hypotheses in reality is expensive, time consuming, and, in some cases, dangerous. With the recent development of several high-space, many-body, GPU-accelerated soft body simulators [10, 18], sim2real for soft robotics and synthetic biology has become more feasible. However, because these simulators have yet to be employed to design physical systems, their transferability is currently unknown.

Previous work has demonstrated methods that promote successful sim2real transfer of soft object manipulation but not soft robot behavior. For example, a rigid-bodied robot arm was successfully trained in simulation to fold towels and drape pieces of cloth over a hanger [20]. However, the reality gap was not quantified beyond a binary success rate for each task. Additionally, the robot’s geometry was fixed and controllers were then optimized for it, whereas in the work reported here, the robot’s geometry is part of the solution space.

Hiller and Lipson [8] evolved the overall geometry and distribution of hard and soft materials in simulation, and transferred the structures and passive dynamics of various

<sup>1</sup>Henceforth, “the reality gap”—as coined by Jakobi et al. [12].

cantilever beams. In a separate experiment that included actuating materials, Hiller and Lipson evolved the morphology and behavior of soft robots in simulation, and then built one of the evolved designs physically. However, in order to transfer the simulated behavior of this one design, the physical robot needed to be placed in a pressure and vacuum chamber, whereas the hundreds of soft robot designs built here can be internally pressurized and actuated.

More recently, Kriegman et al. [14] subjected a simulated soft robot (composed of elastic voxels) to a series of damage scenarios that removed increasingly more of the robot’s structure. In each scenario, the robot was challenged to recover function (locomotion) by deforming its remnant structure, without changing its predamage control policy. A pair of recovery strategies, automatically discovered by an evolutionary algorithm in simulation, were transferred to reality (using silicone “voxels”), but function was not: The physical system could deform its resting structure as dictated by the recovery strategy, but it could not locomote, before or after damage. The physical robot was heavy, had high friction feet, and was symmetrically actuated in phase, so it just oscillated in place.

To determine the particular challenges and opportunities of soft robot transferal, it would be useful to greatly scale up the number of design/policy pairs transferred. To this end, we present a soft robot design and construction kit based on the silicone voxel modules used in [14], but miniaturized to increase stability, simplified to improve reproducibility, and arbitrarily actuated to permit the transferal of locomotion.

Other modular yet rigid-bodied robot design and construction kits exist, such as Molecubes [30]. However, our kit is easier, faster, cheaper, and safer to use. In short, silicone is molded into hollow voxels, and tubing is attached to supply low pressure actuation from a hand pump, causing volumetric changes in one or more of the voxels (Figs. 2 and 3). For simple behaviors robust to actuation noise, there is no need to use a highly-pressurized air supply or program microcontrollers for open-loop control. There are also no expensive motors or power supplies.

Here, we employ the kit as an instrument to measure the reality gap as a function of morphology (Table I). To do so, we fabricated 108 morphologies (transferal of structure) and compared the behavior of nine simulated designs to their silicone equivalents (transferal of behavior). We hope that the kit’s affordability, safety, speed, and simplicity will generate increasingly more, and more reproducible, data about the automated design of increasingly competent soft machines.

## II. METHODS

### A. The design space.

Following [8] and [14], our kit uses elastic voxels as building blocks of structure. Here, we considered a 2-by-2-by-2 cartesian lattice workspace, within which voxels were connected together to form a robot. At each x,y,z coordinate, voxels could either be passive, volumetrically actuated, or absent, yielding a total of  $3^8 = 6561$  different configurations. We evaluated each configuration in simulation.

TABLE I  
SUMMARY OF PUBLISHED SIM2REAL TRANSFERENCE.

<i>Author/citation</i>	<i>Year</i>	<i>Controllers</i>	<i>Morphologies</i>
Miglino et al. [21]	1994	1	1
Jakobi et al. [12]	1995	2	1
Harvey et al. [7]	1997	4	1
Lipson and Pollack [17]	2000	3	3
Bongard et al. [2]	2006	34	2
Hiller and Lipson [8]	2011	1	5
Koos et al. [13]	2012	2	2
Moeckel et al. [22]	2013	1	1
Caluwaerts et al. [3]	2014	2	1
Cully et al. [5]	2015	10	10
Cellucci et al. [4]	2017	1	3
Tobin et al. [29]	2017	1	1
Rusu et al. [27]	2017	1	1
Peng et al. [24]	2018	1	1
Pinto et al. [25]	2018	3	1
Tan et al. [28]	2018	2	1
Golemo et al. [6]	2018	1	1
Matas et al. [20]	2018	3	1
Kwiatkowski and Lipson [15]	2019	2	2
Hwangbo et al. [11]	2019	3	1
Kriegman et al. [14]	2019	1	5
Nachum et al. [23]	2019	3	1
Akkaya et al. [1]	2019	1	1
Rosser et al. [26]	2019	1	16
<b>The results presented here</b>	<b>2019</b>	<b>1</b>	<b>108</b>

### B. The simulation.

We used the soft-body physics engine Voxelyze [9] to simulate robots composed of actuating and/or passive, elastic voxels. The simulator models the distance between adjacent voxels as Euler-Bernoulli beams (critically damped;  $\zeta = 1$ ). Additionally, a collision detection system monitors the distance between the voxels on the surface of the robot at each timestep. If a pair of surface voxels are detected to collide (intersect), a temporary beam (underdamped;  $\zeta = 0.8$ ) is constructed between the two until the collision is resolved.

Designs were simulated with a gravitational acceleration of  $-9.81 \text{ m/s}^2$ , and initialized on top of an infinite surface plane at  $z = 0$ . Coulomb friction is applied to voxels in contact with the surface plane. Voxels were simulated to have  $1 \text{ cm}^3$  resting volume (resting beam lengths), with Young’s modulus  $10^7 \text{ Pa}$ , Poisson’s ratio 0.35, and coefficients of static and kinetic friction of 1 and 0.5, respectively. These hyperparameters were adopted from [14]. For more details about how the physics are actually modeled, see [9].

Volumetric actuation was implemented by varying the rest length between voxels, in all three dimensions, when computing the elastic force between them. Volumetric expansion

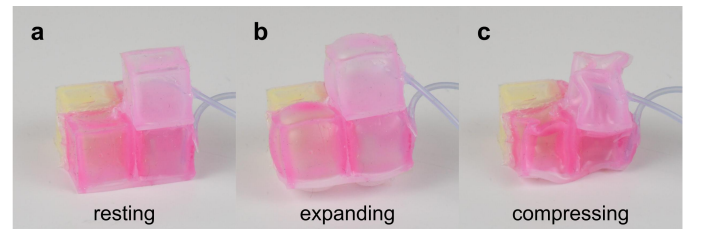


Fig. 2. A random morphology in the design space shown at atmospheric (resting; **a**), positive (expanding; **b**), and negative (compressing; **c**) pressure.

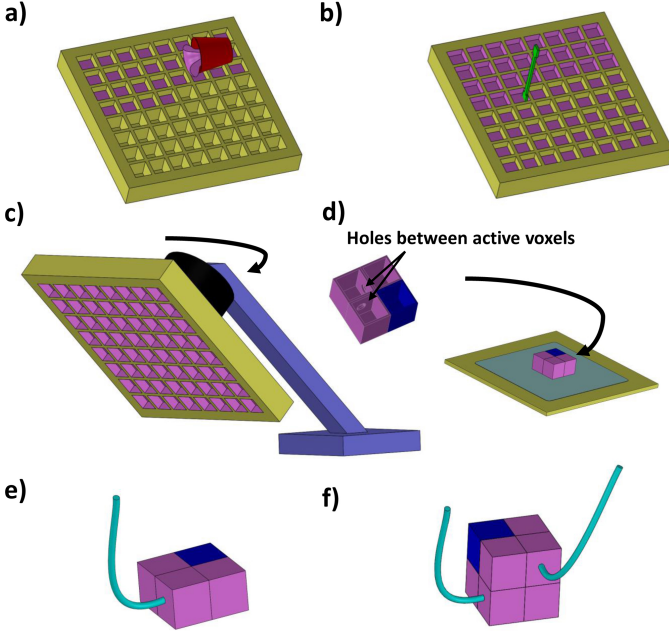


Fig. 3. **Manufacturing modular soft robots.** Hollow, silicone voxels were created by partially filling an open-face mold with silicone (a), using a spatula to spread it along the interior walls (b), and then securing the mold to a 1-axis rotational molding machine (c). This process allowed excess silicone to drip out of the mold, while spreading the remaining silicone into a thin uniform layer. The cured, bottomless voxels were then appropriately arranged and connected for each  $x,y$  slice of the design, and bonded with a shared bottom layer (d). Finally, tubing was attached (e), and the slices were stacked and bonded to form the design (f). Video: [youtu.be/jbQ2T7j1YRU](https://youtu.be/jbQ2T7j1YRU).

in simulation and reality are both roughly spherical (Fig. 2b), but compression in reality is more complex and difficult to simulate: the voxels buckle (Fig. 2c). So volumetric actuation was here limited to expansion only (+90% rest volume). The active voxels expand in phase with each other as dictated by a central pattern generator: a sine wave with frequency 4 Hz and amplitude  $1.9 \text{ cm}^3$ . When the sine wave is at or below zero, the active voxels remain at their resting volume ( $1 \text{ cm}^3$ ). This produced quasistatic dynamics.

Each design was simulated for 8831 timesteps, with a stepsize of 0.000453 seconds, resulting in a total simulation time of 4 seconds. During the first 552 times steps (0.25 sec), the design was allowed to settle under gravity before actuation begins, ensuring that movement (if any) is a result of actuation, rather than passively falling forward. Just before actuation, the design's initial center of mass is recorded as  $(x_0, y_0, z_0)$ . The active voxels are then actuated for 3.75 sec at 4 Hz, or 15 actuation cycles.

An exhaustive search of all 6561 designs (in batches of 50) took 58 CPU hours (1.8 wall-clock hours) on a single AMD Ryzen threadripper 1950X 16-core/32-thread processor. Fitness was taken to be the net displacement (away from the origin in any direction) of the design's center of mass, in terms of euclidean distance in the plane, where the origin is defined by the  $x, y$  components of the design's initial center of mass  $(x_0, y_0)$ . Fitness is thus defined as:

$$F = \sqrt{(x_t - x_0)^2 + (y_t - y_0)^2}, \quad (1)$$

where  $x_t, y_t$  are the final coordinates of the design at the end of the evaluation period.

### C. Reality.

Following Kriegman et al. [14], simulated voxels were realized physically as pneumatically-actuated, hollow silicone voxels. The physical robot in [14] was constructed to transfer symmetrical shape change, so its actuated voxels were distributed symmetrically and hooked into a single pressure inlet. Thus, pressure oscillations occurred symmetrically in phase, and the robot could only pulse in place. Moreover, due to thin voxel walls relative to overall voxel size, and the tubing and glue used to bond them together, the robot in [14] could not fully support its own weight. The robot was lifted off the ground by placing it on top of a small petri dish, positioned underneath a segment of entirely passive voxels in the center of the robot's ventral surface. This permitted ventral (and more extreme global) changes in surface curvature, yielding successful sim2real transfer of shape change, but not locomotion.

The construction kit presented here rectifies the weight issue by miniaturizing the voxels—voxel length was halved (from 3cm to 1.5cm) and the wall thickness remained the same (1mm), reducing voxel mass from 4.3g to 1.2g (including tubing but not pneumatic connectors). Further, the inter-voxel tubing and glue was replaced with holes punched through the walls of adjacent active voxels in the same  $x,y$  slice, before attaching them with a shared bottom layer (Fig. 3d). Finally, locomotion is now possible because separate contiguous sections of voxels in each slice can be arbitrarily actuated in or out of phase with other sections across the body.

### D. The build protocol.

The voxels were manufactured using a single-axis rotational molding machine.<sup>2</sup> First, an open-face mold was fabricated by interlacing 26 acrylic strips into a flat base, to form a lattice of cubic concavities, resembling an ice-cube tray (Fig. 3a). Mold components were laser-cut (VLS2.30, Universal Laser System) from a flat acrylic sheet with a thickness of 0.025 inch. Next, silicone (Dragon Skin 10 Fast; Smooth-On, Inc.) was poured into the acrylic mold (Fig. 3a), and a spatula was used to spread the silicone along the interior walls of each cavity (Fig. 3b). Colored pigment was added to each batch of silicone to indicate whether the voxel was active or passive, simplifying the assembly process. Here we used pink for active voxels and blue or yellow for passive voxels.

The mold was then flipped upside down and secured to a 1-axis rotational molding machine. The machine was clamped to a table with binder clips, angled  $45^\circ$  relative to horizontal, and set to rotate  $90^\circ$  every 45 seconds (Fig. 3c). This allowed the silicone to flow and evenly coat the walls of the mold, as excess silicone dripped out. After the voxels partially cured for 25 minutes at room temperature, the mold was moved to an incubator, with a temperature of  $60^\circ\text{C}$  for another 20 minutes. (Without an incubator, the silicone will take 75 minutes to fully cure at room temperature.)

<sup>2</sup>The required materials are listed at the end of the manuscript in Table II.



The above steps were then repeated to add an additional layer of silicone. Once the second layer cured, the bottomless voxels were removed from the mold using an X-Acto knife, and excess silicone around their edges was trimmed.

In the next step, each x,y slice (or dorsal plane) of the design was assembled by using Sil-Poxy (Smooth-On, Inc.) to bond adjacent voxels and prevent the slice from shifting. Holes were then punched between adjacent active voxels so that contiguous collections of voxels could be actuated together in phase. Each actuator group needed to contain at least one voxel on the surface of the design so that it could be controlled by an external pressure inlet. To create the bottom layer, two 1mm-thick rulers were attached to an acrylic substrate using double-sided tape and silicone was poured in the space between them. Then, the slice of bottomless voxels was flipped, open-side down, onto this uncured silicone layer (Fig. 3d).

After the bottom layer cured, a thin layer of silicone was applied with a popsicle stick along the outermost portions of the interstices of the voxels, bonding adjacent voxels (without gluing over inter-voxel holes). Then, the slice was cut from the silicone sheet and a hole was poked into the side of one exterior voxel from each group of active voxels. Next, a 1/32" ID silicone tube was inserted into the hole, and glued in place with Sil-Poxy, applied with a Q-tip (Fig. 3e). The end of this tube was then connected to a straight pneumatic connector, which was connected to 1/16" ID silicone tubing.

Occasional imperfections in alignment, silicone thickness, or inter-voxel hole sizes would result in leaky structures. Leaks were detected by filling a beaker with water, submerging the voxels, and inflating them. Bubbles would emanate from leaks, which were repaired with Sil-Poxy. After repairing any leaks, the slices were stacked on top of each other and bonded together using a thin layer of silicone (Fig. 3e). Finally, these layers were connected pneumatically with assorted pneumatic connectors, attached to 1/16" ID silicone tubes.

### III. RESULTS

To test the effects of morphology on fitness and sim2real transfer success, it is useful to first visualize the design space.

However, because there are eight cartesian voxel coordinates in the chosen workspace, the design space here is eight dimensional, which is difficult to draw (let alone conceptualize) without dimensionality reduction. By nesting the dimensions of a search space onto a single plot (Fig. 4), the entire space can be visualized as a 2D heatmap. This strategy was used by Cully et al. [5] to

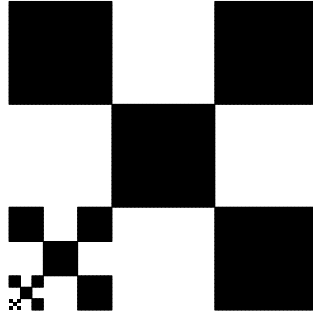


Fig. 4. The 2D tessellation of 8D ternary vector space used in Fig. 5.

neatly visualize the predicted fitness of a very large library of control policies, as a function of the time a robot's six limbs were in contact with the simulated ground plane: 6D

quinary control space was mapped to 2D, by nesting pairs of dimensions within each other.

Here, the 8D ternary morphology space was reduced to 2D by plotting pairs of dimensions nested within each other (Fig. 5). The pixel in the exact center of Fig. 5, for instance, represents the configuration consisting entirely of passive voxels, and thus cannot locomote ( $F = 0$ ). Likewise, the pixel in the top right-hand corner of the heatmap represents the configuration of all active voxels (Fig. 6d), which actuated symmetrically in phase, and thus (given its flat ventral surface) could not locomote across the flat ground plane ( $F = 0$ ). Finally, the pixel in the bottom left-hand corner contains no voxels at all, and thus  $F = 0$ .

For locomotion, a good design obviously needs to have a body, rather than none at all. With open-loop, in-phase actuation, designs also need to have asymmetrical mass and/or actuator distributions, or they will not generate any forward movement. However it is not clear, even for this minimal design space, exactly which asymmetrical designs will yield the highest fitness. Yet we can see small clusters and lines of similarly colored pixels in Fig. 5, representing morphologically similar designs with similar fitness. This suggests that these configurations and substructures would be relatively stable under random mutations or errors in fabrication.

Because fitness was measured by displacement in any direction away from the origin (Eq. 1), there are four configurations—rotations, in the x,y plane, of a single geometry and distribution of passive and active voxels—with different

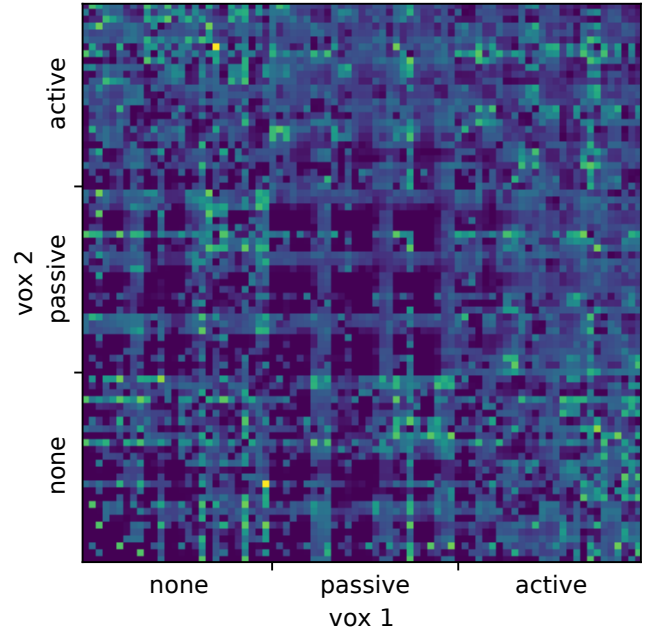


Fig. 5. **Simulating modular soft robots.** The design space is plotted as a heatmap, containing one cell for each of the 6561 possible configurations. Lighter colored cells are fitter designs (Eq. 1). Each design is defined by a vector of eight ternary values, indicating what kind of voxel (none, passive, or active) the design contains at the eight lattice points in the  $2 \times 2 \times 2$  workspace. The 8D ternary vector is reduced to a 2D heatmap by nesting pairs of dimensions within each other: four, nested  $3 \times 3$  grids result in a  $3^4 \times 3^4 = 81 \times 81$  overall heatmap.

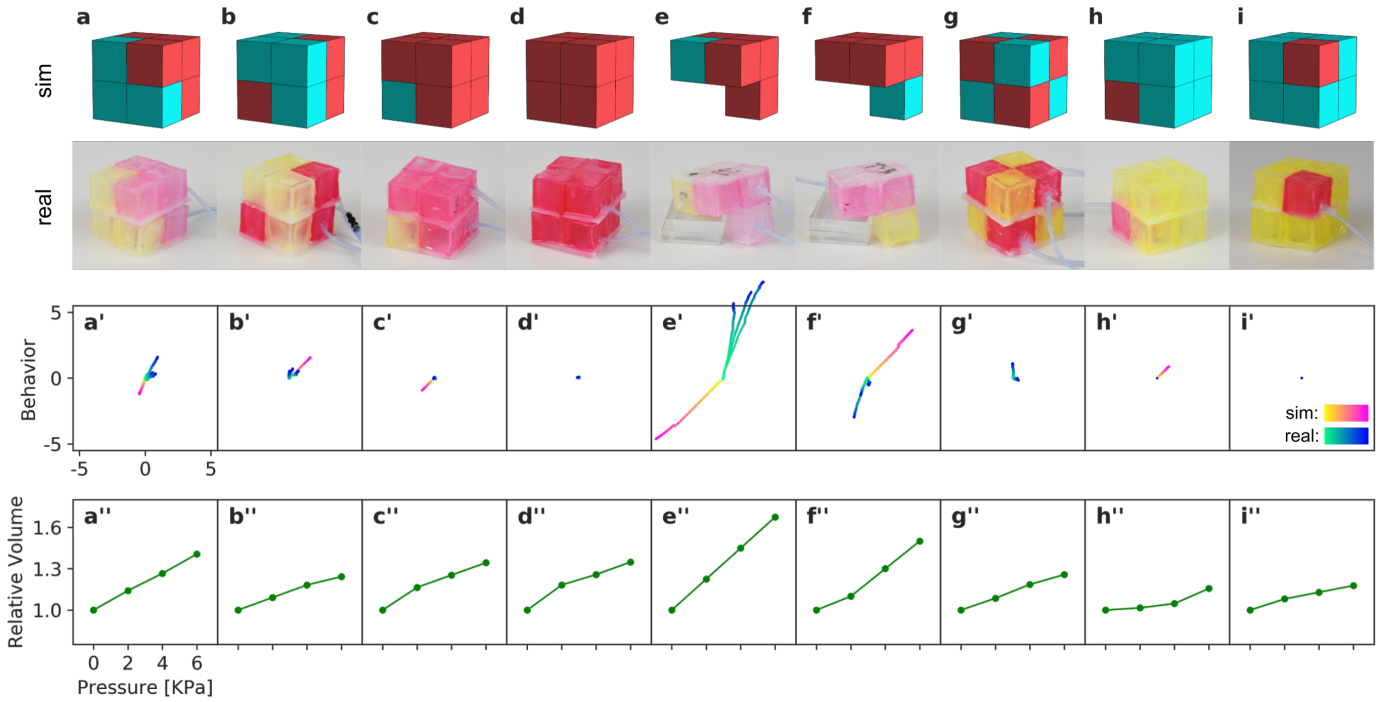


Fig. 6. **Measuring transfer from simulation to reality.** Nine designs (a-i) were evaluated three times each in reality (green-to-blue gradient colored curves in a'-i'). The behavioral trajectories start at the origin (green) and end at the robot's final XY destination (blue) (in centimeters). The simulated movement tracks (yellow-to-pink curves) are superimposed on top of the real ones. The relative volume (normalized by rest volume) was also recorded for each design at four points during actuation under water (a''-i''). The simulated and real behavior of designs e and f can be observed here: [youtu.be/UqjvmkYa9u4](https://youtu.be/UqjvmkYa9u4).

behaviors (they moved in different directions) but very similar (if not identical) fitness. There were also some configurations that, when rotated upward (in the  $x,z$  or  $y,z$  plane) fell into the same basic orientation and behavior but with a slightly different heading. Thus, configurations with similar fitness (similarly colored pixels) are reflected across multiple, nested planes of symmetry in Fig. 5. These symmetries can also be seen in the manufactured robots (Fig. 1b). The uniqueness of designs (i.e., the size of the search space of morphologies) is therefore a function of how behavior is measured.

Fig. 6 shows the behavior of nine different designs in simulation and reality. The real robot was actuated 90 times at 6 kPa pressure on a surface covered with cornstarch (Argo®, ACH Food Companies, Inc.) to reduce friction, and is compared to 23 simulated actuation cycles. Seven of the nine designs filled the cubic workspace with passive and active voxels, while the other two share a more complex geometry: a single-voxel limb attached to the face of a 2-by-2 plane of voxels (Fig. 6e,f). In one, the limb is active (Fig. 6e), in the other it is passive (Fig. 6f). These two designs achieved the two highest fitness scores (Eq. 1), in both simulation and reality.

By this measure, the reality gap appears small. However, these simulated designs move very differently from their manufactured equivalents. The simulated morphology in Fig. 6e pushes off its active limb, whereas in reality the design uses its limb to pull itself forward, in the opposite direction. Likewise, the simulated morphology in Fig. 6f pushes off its active 2-by-2 torso, whereas in reality the design uses its torso to pull itself forward, in the opposite direction.

Majidi et al. [19] showed that the interfacial shear strength

and coefficient of friction of the surface on which their soft robot undulated determined the direction of locomotion. They decomposed friction into load- and area controlled terms for point and surface contacts, respectively. On slippery surfaces with low interfacial shear resistance, the robot anchored about the point contact (expanded section) for locomotion and pulled its surface contact (passive segment). However, on surfaces with high interfacial shear resistance, the robot anchored about the surface contact and pulled the point contact toward it. We hypothesize that such differences in tribological properties could have caused our designs to move in opposite directions in simulation and reality.

In an attempt to test this hypothesis and reduce the simulation-reality discrepancies that cause the virtual configurations in Fig. 6 to move differently than their physical realizations, we performed a grid search of various simulation hyperparameters, including the coefficients of static and kinetic friction. However, we could not identify a pair of friction coefficients that resulted in correct movement heading for all nine of the behaving designs (Fig. 6a'-i'). This could be due to either low precision or low accuracy of the model. To isolate and test the former possibility, we increased the resolution of the simulated surface contact geometry by modeling each silicone voxel as a 3-by-3-by-3 group of simulated "subvoxels" (Fig. 7), and then re-ran the parameter sweep. Still, we could not find friction settings in which the simulated movement direction matched the ground truth across all designs simultaneously. This suggests that the accuracy of Coulomb friction model may be insufficient to model this type of movement.

The Coulomb approximation assumes that friction is simply

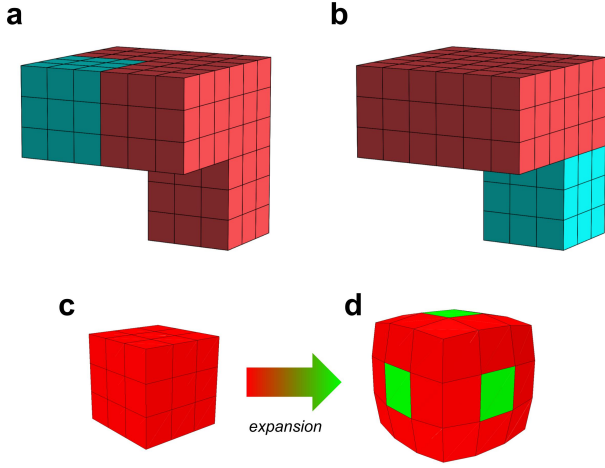


Fig. 7. A higher resolution model in which each silicone voxel is approximated by a 3-by-3-by-3 group of simulated subvoxels: a high-res voxel. The design in **a** and **b** are high-res instantiations of those in Figs. 6e and 6f, respectively. Spherical volumetric expansion in a high-res voxel (**c**) was approximated by increasing the rest length between the centermost subvoxel and the subvoxels at center of each face (green subvoxels in **d**).

proportional to the vertical component  $N$  of the reaction force, and independent of the contact area. However, friction is also a function of the surface area and interfacial shear strength  $\tau$ , a fixed constant which is mostly governed by adhesion or mechanical interlocking between the contacting surfaces. A better model would thus consider friction as a function of both the normal force and the interfacial shear strength. However, before fundamentally changing the simulator, we plan to evaluate designs in noisy environments with imperfect control over actuation characteristics to avoid ascribing high fitness to designs that exploited unrealistic properties of the simulation [12]. Additionally, data from reality could be used to automatically tune the geometry and resolution of the simulated finite elements [2], or to predict the kinds of behaviors that are more likely to successfully transfer [13], and which should be tested next [2]. Concurrently, we are investigating additional physical surfaces with varied tribological properties in an attempt to match reality to simulation.

#### IV. DISCUSSION

In this paper, we introduced a low cost, open source platform for designing and rapidly building soft robots, and used it to transfer 108 different morphologies (voxels on a cartesian grid) from simulation to reality. We then measured the reality gap as function of the robot’s design (geometry and distribution of passive and actuating voxels) by tracking the behavior of nine transferred morphologies. Under one measure (net displacement) the reality gap appeared rather small, but under another (velocity) the gap was much wider, likely due to oversimplified tribological contacts between the simulated design plane and the robot’s ventral surface [19].

Although most of the transferred designs (99 out of the 108) were not actuated in reality, they nevertheless served an important function: they were *sketches*. Sketches let us think more clearly about the behavior or properties (e.g., stability)

of a design without investing the additional time and resources required to fully build and examine the design itself. Sketches, in other words, greatly increase the breadth of exploration in design space. All sim2real methods embrace this utility of simplifying sketches. Simulation, after all, is also a sketch.

However, there is a tacit assumption in robotics about Depth First. A typical sim2real experiment begins by sending a complicated robot design across the reality gap, and then endeavors to learn transferable policies that control the morphology in all its complexity. But this is not how most design proceeds. An architect first roughly sketches a structure, say, a bridge, on the back of a napkin. A diversity of designs are then generated, tweaked, discarded or provisionally accepted—at this shallow level of napkin realism—before more detailed blueprints are drawn under more stringent constraints. Blueprints, too, undergo this breadthwise evolution, before the most promising are realized physically, first as scale models (built from matchsticks and glue instead of concrete and steel), then, finally, at full scale and cost. This incrementally weeds out nontransferable features and adds mechanical complexity only when and where it is necessary to do so, rather than globally from the outset.

The assumption that the reality gap can be bridged by policy search alone, with a single robot design, is groundless. The desired behavior of a robot is typically much more complicated than that of architecture. This suggests the necessity of more, not less, sketches. Soft robots are more complicated still. This makes their automated design that much more appealing, but implies the need for even greater breadth of sketches, at more intermediate levels of realism. Though not every experiment will need to start from a blank slate. Instead, designers (whether human or AI) could leverage prior knowledge to reject truly awful designs before sketching them in the first place. The designs transferred here add to a growing database (prior probabilities) about which and how well different soft robot designs and behaviors can be realized physically. Our construction kit has the potential to further increase this data by lowering not only cost and build times but also the barrier of entry to soft robotics for non-experts.

The generality of such data beyond robotics is currently not known, but it could also have important implications for developmental biology and regenerative medicine. The bio-electric and genetic control policies that orchestrate adaptive remodeling of growth and form in organisms are not yet understood, but could, in future, be reverse-engineered by machine learning, and then controlled externally to induce regeneration in otherwise non-regenerative organisms (such as adult humans), or to reprogram otherwise unbounded cancerous growth toward functional organogenesis [16]. However, such advances in regenerative medicine and synthetic morphology will only be possible if hypotheses generated in simulation are transferable and testable in reality.

#### ACKNOWLEDGEMENTS

This work was supported by NSF award EFRI-1830870 and DARPA contract HR0011-18-2-0022.

TABLE II  
THE CONSTRUCTION KIT FOR MAKING FIFTY 2-BY-2-BY-2 DESIGNS.

No.	Part	Cost	Notes
1	Dragon Skin 10 FAST, 2 lbs kit (Smooth-On, Inc.)	\$32.21	Created voxel bodies and connected x,y slices.
2	Sil-Poxy, 3 ounce tube (Smooth-On, Inc.)	\$30.72	Secured tubing and repaired air leaks in six-sided voxels.
3	1/32" ID silicone tubing, 52 ft	\$34.32	50A shore hardness; connected voxels to actuation system.
4	1/16" pneumatic plastic connectors, 200 pieces	\$100	Straight and T shaped.
5	1/16" ID silicone tubing, 65.6 ft	\$62.98	50A shore hardness; part of actuation system.
6	60 mL Plastic Syringe (McMasterCarr, Inc.)	\$3.13	Used to hand-actuate voxels while checking for leaks.
7	1000 mL beaker (PYREX VISTA Griffin, Fisher Scientific International, Inc.)	\$14	Voxels were inflated in this beaker filled with water to detect leaks.
8	Hole punch 1/4 rectangle (Fiskars)	\$10.35	Created holes between voxels within a slice so they actuated as a unit.
9	X-Acto Knife (McMasterCarr, Inc.)	\$4.11	Cut bottomless voxels out of acrylic mold, and trim the edges.
10	100 mL mixing cups (VWR international), pack of 100	\$66	Where silicone was mixed.
11	Spatula (McMasterCarr, Inc.)	\$7.50	Spread the Dragon Skin on the edges of the acrylic mold.
12	2 Scrap acrylic sheets, 12"×12"×1/8"	\$10.58	Collected scrap Dragon Skin; the surface where voxel bottom were created.
13	2-1mm thick 30 cm metal rulers	\$6.99	Used to set a thickness for sixth voxel side (bottom layer).
14	Double-sided tape	\$4.99	Adhered the rulers to a scrap acrylic sheet.
15	Simple 30 cm metal ruler	\$3.99	Spread a thin sheet of Dragon Skin onto scrap acrylic sheet for sixth side.
16	Popsicle stick (11.3 cm × 1 cm), box of 1000	\$13.49	Applied thin layer of silicone to bond adjacent voxels and x,y slices.
17	Cotton-tipped applicators, 6 inch, box of 1000	\$8.99	(McKesson Corp.) Used to spread Sil-Poxy.
18	Disposable Gloves (Halyard Inc.), box of 100	\$8.95	Wore when handling uncured silicone.
19	2 Acrylic sheets, 12"×12"×1/8" (McMasterCarr, Inc.)	\$18.30	Used to manufacture the open-face acrylic mold with laser-cut.
20	41 mm binder clips, pack of 12	\$7.99	Held the acrylic mold onto the rotational molding machine.
<b>The rotational molding machine:</b>			
21	Acrylic sheet, 12"×12"×1/4" (McMasterCarr, Inc.)	\$17.34	2× triangular plates, 3× motor mount - supports for rotational machine.
22	Acrylic sheet, 12"×12"×1/4" (McMasterCarr, Inc.)	\$17.34	Mounting plate; holes were cut to minimize weight.
23	Pololu 4756 DC rotational motor	\$39.95	Used for rotational molding machine.
24	Pololu 1999 mounting hum	\$7.95	Used to mount rotational molding machine.
25	Arduino Uno Microcontroller	\$22	Controlled rotation timing and degree.
25	Arduino Motor Shield	\$19.95	Controlled rotation motor.
26	AC charger	\$10.42	12mm×2.1×5.5mm barrel jack, 12V, Supplies power to the Arduino.
27	8020 T-Slotted Solid 1" beams (McMaster-Carr, Inc.)	\$12.31	2×10 cm, 2×40 cm, Supports for rotational molding machine.
28	8020 screws and T-nuts	\$7.92	Connected 8020 beams.
29	"M3×10" screws	\$1.24	Used to mount rotational molding machine.
30	Irwin QuickGrip 12"×2.75" Clamp	\$15.99	Held rotational molding machine to table.
<b>Grand total:</b>		<b>\$622</b>	

## REFERENCES

- [1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chocie, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint*, 2019. URL <https://arxiv.org/abs/1910.07113>.
- [2] Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006. URL <https://doi.org/10.1126/science.1133687>.
- [3] Ken Caluwaerts, Jérémie Despraz, Atıl Işçen, Andrew P Sabelhaus, Jonathan Bruce, Benjamin Schrauwen, and Vytas SunSpiral. Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of the royal society interface*, 11(98):20140520, 2014. URL <https://doi.org/10.1098/rsif.2014.0520>.
- [4] Daniel Cellucci, Robert MacCurdy, Hod Lipson, and Sebastian Risi. 1d printing of recyclable robots. *IEEE Robotics and Automation Letters*, 2(4):1964–1971, 2017. URL <https://doi.org/10.1109/LRA.2017.2716418>.
- [5] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521:503–507, 2015. URL <https://doi.org/10.1038/nature14422>.
- [6] Florian Golemo, Adrien Ali Taiga, Aaron Courville, and Pierre-Yves Oudeyer. Sim-to-real transfer with neural-augmented robot simulation. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 817–828. PMLR, 2018. URL <http://proceedings.mlr.press/v87/golemo18a.html>.
- [7] I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi. Evolutionary robotics: the sussex approach. *Robotics and Autonomous Systems*, 20(2):205 – 224, 1997. ISSN 0921-8890. URL [https://doi.org/10.1016/S0921-8890\(96\)00067-X](https://doi.org/10.1016/S0921-8890(96)00067-X).
- [8] Jonathan Hiller and Hod Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466, 2011. URL <https://doi.org/10.1109/TRO.2011.2172702>.
- [9] Jonathan Hiller and Hod Lipson. Dynamic simulation of soft multimaterial 3D-printed objects. *Soft Robotics*, 1(1):



- 88–101, 2014. URL <https://doi.org/10.1089/soro.2013.0010>.
- [10] Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. Subspace neural physics: fast data-driven interactive simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 6. ACM, 2019. URL <https://doi.org/10.1145/3309486.3340245>.
- [11] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019. URL <https://doi.org/10.1126/scirobotics.aau5872>.
- [12] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer, 1995. URL [https://doi.org/10.1007/3-540-59496-5\\_337](https://doi.org/10.1007/3-540-59496-5_337).
- [13] Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2012. URL <https://doi.org/10.1109/TEVC.2012.2185849>.
- [14] Sam Kriegman, Stephanie Walker, Dylan Shah, Michael Levin, Rebecca Kramer-Bottiglio, and Josh Bongard. Automated shapeshifting for function recovery in damaged robots. In *Proceedings of Robotics: Science and Systems*, 2019. URL <http://www.roboticsproceedings.org/rss15/p28.html>.
- [15] Robert Kwiatkowski and Hod Lipson. Task-agnostic self-modeling machines. *Science Robotics*, 4(26), 2019. URL <https://doi.org/10.1126/scirobotics.aau9354>.
- [16] Michael Levin. Reprogramming cells and tissue patterning via bioelectrical pathways: molecular mechanisms and biomedical opportunities. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 5(6):657–676, 2013. URL <https://doi.org/10.1002/wsbm.1236>.
- [17] Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974, 2000. URL <https://doi.org/10.1038/35023115>.
- [18] Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviy-chuk. Non-smooth newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics (TOG)*, 38(5):140, 2019. URL <https://doi.org/10.1145/3338695>.
- [19] Carmel Majidi, Robert F Shepherd, Rebecca K Kramer, George M Whitesides, and Robert J Wood. Influence of surface traction on soft robot undulation. *The International Journal of Robotics Research*, 32(13):1577–1584, 2013. URL <https://doi.org/10.1177/0278364913498432>.
- [20] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 734–743. PMLR, 2018. URL <http://proceedings.mlr.press/v87/matas18a.html>.
- [21] Orazio Miglino, Kourosh Nafasi, and Charles E. Taylor. Selection for wandering behavior in a small robot. *Artificial Life*, 2(1):101–116, 1994. URL <https://doi.org/10.1162/artl.1994.2.1.101>.
- [22] Rico Moeckel, Yura N Perov, Anh The Nguyen, Massimo Vespignani, Stéphane Bonardi, Soha Pouya, Alexander Sproewitz, Jesse van den Kieboom, Frédéric Wilhelm, and Auke Jan Ijspeert. Gait optimization for roombots modular robotmatching simulation and reality. In *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3265–3272, 2013. URL <https://doi.org/10.1109/IROS.2013.6696820>.
- [23] Ofir Nachum, Michael Ahn, Hugo Ponte, Shixiang Gu, and Vikash Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. In *Proceedings of The 3rd Conference on Robot Learning*, 2019. URL <https://arxiv.org/abs/1908.05224>.
- [24] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *The IEEE International Conference on Robotics and Automation (ICRA)*, 2018. URL <https://doi.org/10.1109/ICRA.2018.8460528>.
- [25] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Proceedings of Robotics: Science and Systems*, 2018. URL <https://doi.org/10.15607/RSS.2018.XIV.008>.
- [26] Kent Rosser, Jia Kok, Javaan Chahl, and Josh Bongard. Sim2real transfer degrades non-monotonically with morphological complexity for flapping wing design. *arXiv preprint*, 2019. URL <https://arxiv.org/abs/1910.13790>.
- [27] Andrei A. Rusu, Matej Veerk, Thomas Rothl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 262–270. PMLR, 2017. URL <http://proceedings.mlr.press/v78/rusu17a.html>.
- [28] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Proceedings of Robotics: Science and Systems*, 2018. URL <https://doi.org/10.15607/RSS.2018.XIV.010>.
- [29] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. URL <https://doi.org/10.1109/IROS.2017.8202133>.
- [30] Victor Zikov, Andrew Chan, and Hod Lipson. Molecubes: An open-source modular robotics kit. In *IROS Self-Reconfigurable Robotics Workshop*, 2007. URL <http://molecubes.org>.