

PaperBot: Learning to Design Real-World Tools Using Paper

Ruoshi Liu¹ Junbang Liang¹ Sruthi Sudhakar¹ Huy Ha^{1,2} Cheng Chi^{1,2} Shuran Song² Carl Vondrick¹

¹ Columbia University ² Stanford University

paperbot.cs.columbia.edu

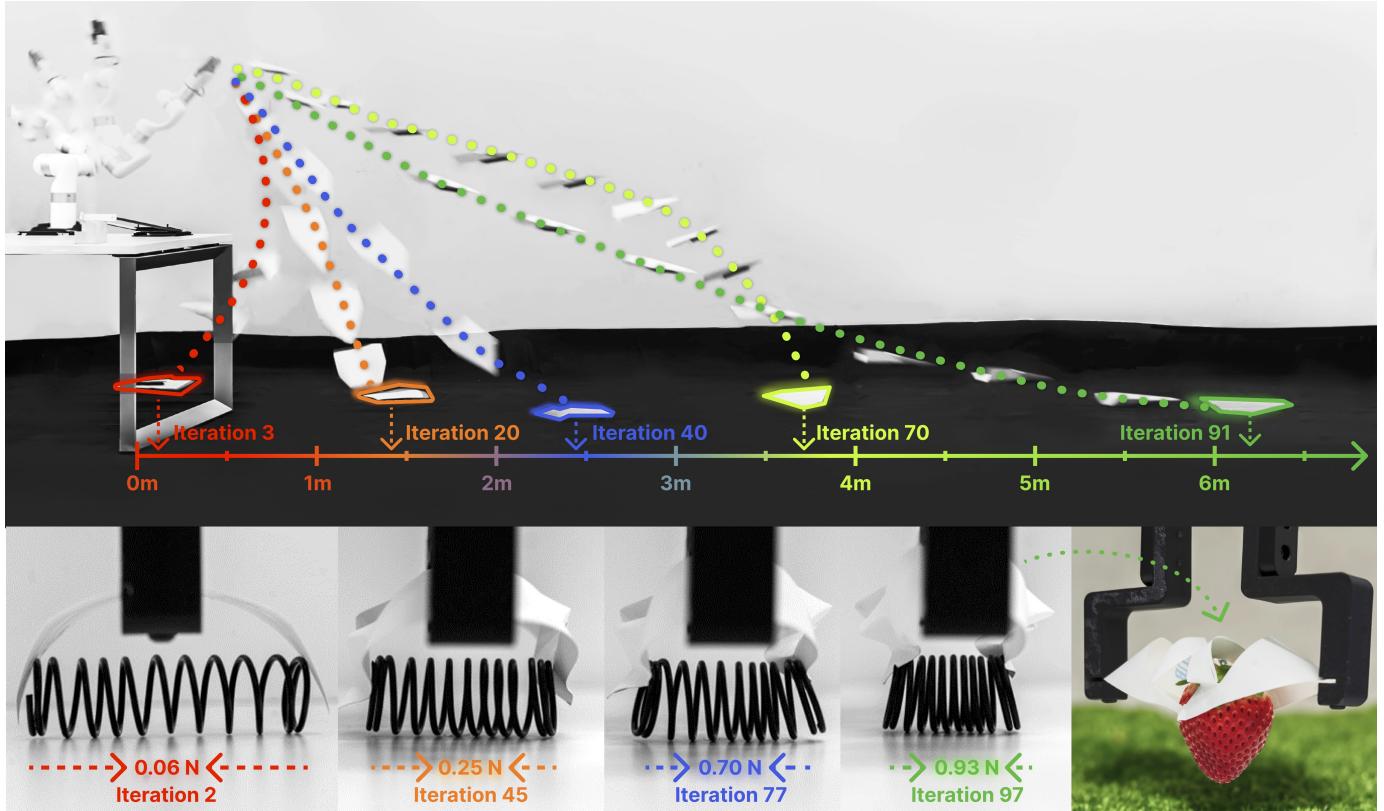


Fig. 1: **Self-supervised Design of Paper Tools.** Through trial-and-error, PaperBot autonomously learns how to design and use paper tools directly in the real world. Given only 100 trials (≈ 3 hours), our fully autonomous system discovers a paper airplane folding and throwing strategy that flew further than the best human design after the same number of trials (top), and learns how to cut and actuate a kirigami gripper that exerts 0.93N of force (bottom), equivalent to the weight of over four strawberries. For our system’s final design, please see Fig. 5.

Abstract—Paper is a recyclable¹, affordable, and widely accessible material, making it a popular medium for building practical tools. Traditional tool design either relies on simulation or physical analysis, which is often inaccurate and time-consuming. In this paper, we propose PaperBot, a framework that directly learns to design a tool out of paper then use it in the real world. We demonstrated the effectiveness and efficiency of PaperBot on two tool design tasks: a) learning to fold and throw paper airplanes for maximum travel distance, and b) learning to cut paper into grippers that exert maximum gripping force. We present a self-supervised learning framework that performs cutting, sequential-folding, and dynamic throwing and actuation actions to discover and optimize the design of paper tools. We deploy our system to a real-world bi-manual robotic system to solve challenging design tasks involving aerodynamics (paper airplane), friction and deformation (paper gripper) that are difficult to simulate accurately and efficiently with traditional tool design approaches.

I. INTRODUCTION

If you wanted to fold this paper (yes, the sheet you are reading) into an airplane that travels a maximum distance, how would you do it? By experimenting with paper as a child, you have learned multiple ways to cut and fold objects out of paper. In this work, we aim to automate this iterative design, deploy, and improve loop with machine learning methods in order to create real-world tools using paper (see Fig. 1). Paper is an attractive medium for design because it is an affordable, recyclable, and versatile commodity that can be repurposed and customized for different functions. Society has made many different tools out of paper – ranging from filters and fans

¹ All the paper used during experiments was properly recycled.

to packaging boxes and bowls, and it is beneficial to the environment to make more. Robots could also create and use its own paper tools for many applications, for example in medical settings [8] where the cost of a paper end-effector allows it to be discarded after use or in logistical settings where paper vessels can be customized to each object.

Prior work in robotics has largely studied tool design in the context of simulation [31, 14, 45, 49, 25], where a reinforcement learning model learns the design of hardware jointly with a policy to use the tool. Despite their impressive results, these prior works typically assume an abundance of data, which restricts their application to efficiently simulatable physical phenomena (i.e., rigid objects). In contrast, paper are deformable, thin-shell objects with relatively low elasticity and high plasticity. Further, its applications typically involve interaction with fluids and air, which are expensive to simulate accurately. Instead of scaling up high-quality synthetic data, how else could we approach automatic paper tool design?

In this paper, we introduce PaperBot, a robot system that autonomously performs experiments in the real world in order to learn the design of paper tools with different functions. Given a reward function (such as travel distance or gripping force), the system learns a series of cuts or folds in order to create a paper tool that optimizes a reward. Our system finds these solutions through experimentation and learning:

- **An automation pipeline that creates tools from sampled design parameters.** We build a system composed of bi-manual robot arms and several paper-related machines including a Cricut machine to perform a sequence of folding and cutting actions on paper. These actions are composed of a set of learnable parameters.
- **Automatic reward evaluation.** We use various sensors, including load cells and RGBD cameras to automatically evaluate the effectiveness of a tool created from the aforementioned automated pipeline and generate a reward.
- **An optimization framework that learns to optimize the design from trial-and-error.** We use a neural network as a surrogate model to learn the non-linear correlation between design parameters and reward. After training, optimal design parameters based on prior experimental data are searched in the surrogate model’s parameter space. We adopt an epsilon-greedy strategy to search for the optimal parameters.

The optimization of the paper tool is driven by real-world feedback and operates entirely in the physical world without simulation and prior knowledge such as physical or material properties, making it generalizable to different tasks. Our approach learns strong designs with just 100 trials, requiring less than 3 hours for both tasks we evaluate. Experiments show that our learned design significantly outperforms baselines, including state-of-the-art methods based on evolutionary algorithms.

A key advantage of PaperBot is its ability to automatically adapt to new situations. In the real world, we often want customized tools for specific task definitions, which are expensive and time consuming to create if we need to hand-



Fig. 2: **Paper Tools.** Paper is an affordable and versatile medium for constructing a variety of different paper tools.

design every customization. PaperBot provides a viable way to automate this customization process. Our experiments show that the surrogate model trained on one task can quickly adapt to different reward definitions.

The primary contribution of this paper is an approach for real-world tool design using paper as a material, and the rest of the paper will present and analyze this system in detail. Section II provides a brief overview of related work. Section III introduces the approach to solve the design tasks of paper airplanes and kirigami grippers. Section IV provides the experimental analysis of the system in terms of efficiency of the learning algorithm and the effectiveness of the designed tools. We believe the ability to perform real world tool design will significantly improve robot’s ability to adapt and solve various real world tasks.

II. RELATED WORK

A. Origami-Inspired Robot and Tool Design

The use of paper as a material in tool and robotic design has been explored in various fields. For instance, altering paper structures through cuts has been shown to modify mechanical properties such as stiffness [20] and resonant frequency [52]. Notably, these flat, 2D sheets can transform into complex 3D shapes, offering diverse design possibilities [21, 7, 30]. For example, kirigami structures have been applied in optical tracking and enhancing solar panel power generation by controlling their deformation [26]. Similarly, origami principles have enabled the creation of transformable wheels capable of bearing significant loads [27]. Moreover, these deformable structures can imitate biological movements, such as crawling [36] and aquatic swimming [54]. In more conventional applications, paper has been made into robotic manipulators [24, 22, 48] and grippers [28, 15, 16, 2, 23] for specialized object handling.

Previous studies have underscored the potential of paper-based designs, yet they typically require intricate human-driven analysis for optimization. This is attributed to the

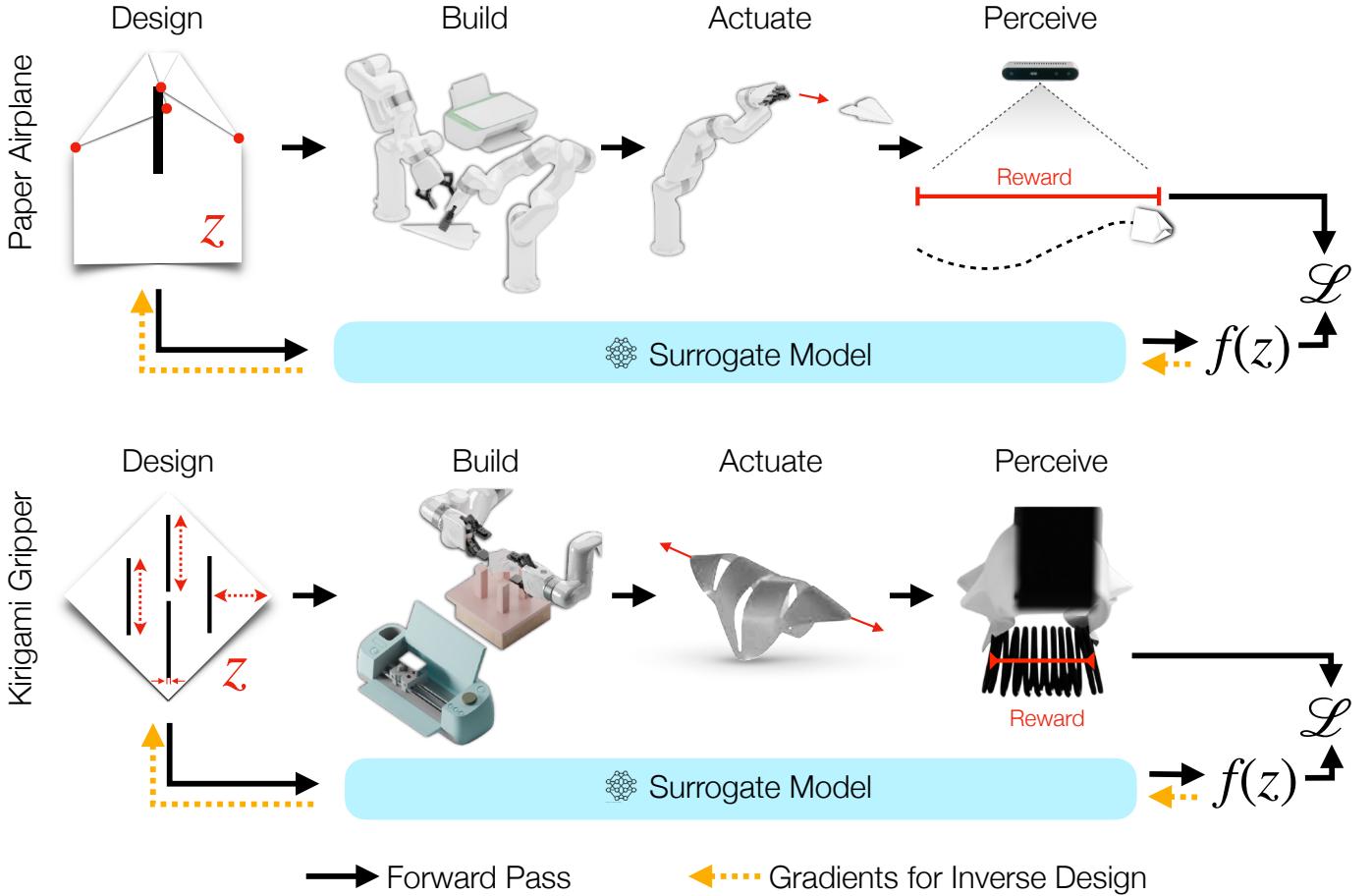


Fig. 3: **Approach Overview.** Our framework samples paper designs for a tool, builds them, actuates a robot to perform on a task with the tool, and perceives its performance. By learning a surrogate model to predict the utility of a design, we obtain a differentiable model that allows us to solve inverse design tasks with gradient-based optimization. The above figure shows how our framework applies to two design tasks of paper airplanes (top) and kirigami grippers (bottom).

inherent challenge in simulating complex deformations and multiple buckling modes of thin-shell structures, making accurate predictions difficult [35, 51, 34, 32]. The complexity of accurately predicting these behaviors necessitates advanced analysis and expertise.

B. Hardware-Software Co-Design

Real-world robots are often limited by their own morphologies as well as other physical constraints, posing many challenges when attempting to accomplish a task with policy learning alone. Therefore in recent years, hardware-software co-design has emerged as a field that studies how hardware design can be jointly learned with policy to create better tools for accomplishing certain tasks.

A line of work has explored solving manipulation tasks relying on simulation environments [49, 25], including [45] that optimizes the shape of end-effector and motion jointly for dynamic planar manipulation. Other studies [55, 49, 17, 47, 18, 46] have investigated the hardware-software co-design problem using modular locomotion robots with data-driven approaches. With the rise of deep reinforcement learning and

GPU-based large-scale simulation, a line of work [38, 5, 31] has focused on learning "hardware-as-policy" in simulated environments. These works highlight the importance of integrating hardware and software in the design process. However, all prior works have approached the problem in a simulated environment, which is often hand-crafted for a specific application and difficult to tune accurately. In this work, we explore the possibility of learning hardware-software co-design directly in the real world by using paper as a material.

C. Dynamic Manipulation of Deformable Objects

The manipulation of deformable objects, such as paper, has been studied in the robotics and sciences community. The two main approaches to learning to manipulate an object are: model-based and model-free. In model-based works, a dynamics model of the real-world object enables the agent to simulate the outcomes of various forces acting upon the object and use this model to choose the next action. This dynamics model is either learned through particle-based dynamics network [43, 44], or via a differentiable simulator [29, 19]. Such approaches either require large amounts of training data due

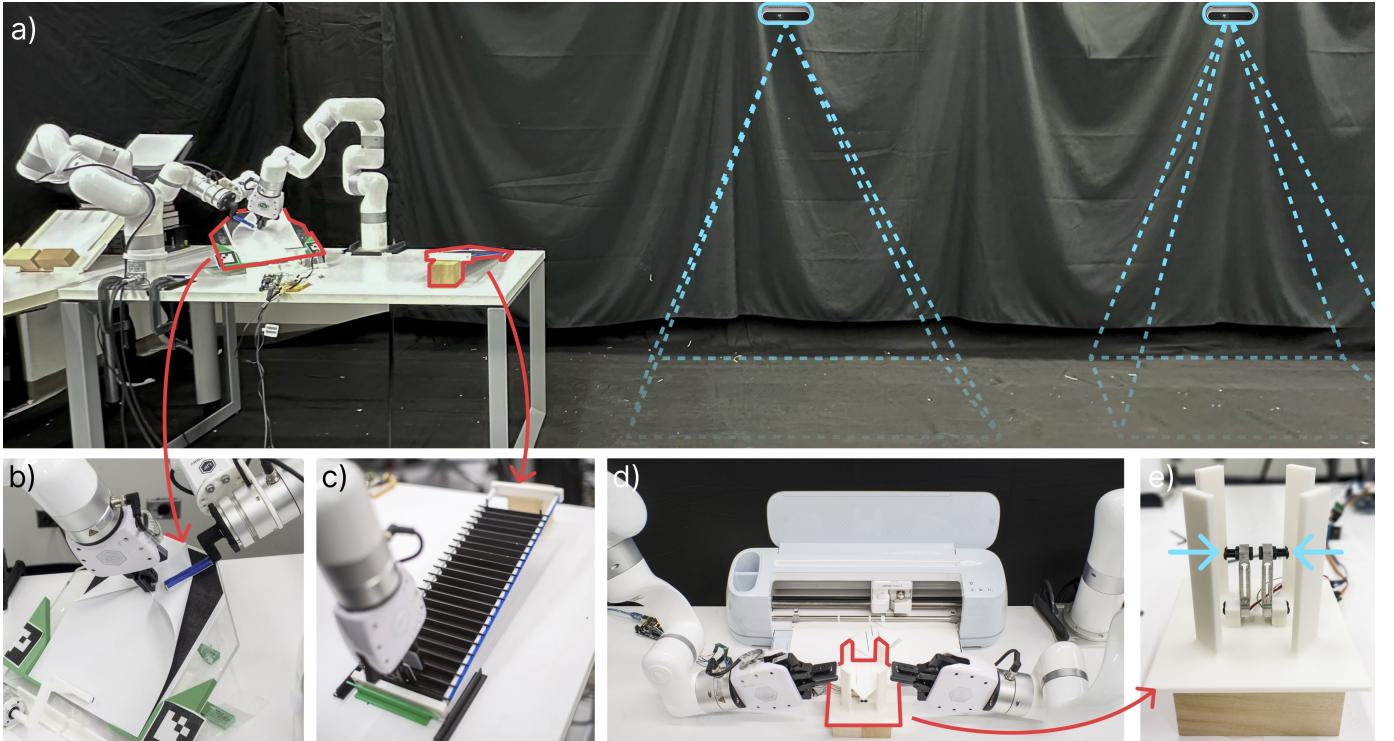


Fig. 4: System Setup. We visualize the workspace of PaperBot while designing paper airplanes (a-c) and kirigami grippers (d-e). (a) shows the full workspace, including two RealSense D435i cameras to measure travel distance over the runway. (b) shows two xArm7s folding a paper airplane. (c) shows the sticky holders. (d) shows the xArm 7s retrieving a gripper cut by the Cricut Maker 3 and actuating it. (e) shows load cells that we use to measure the force of the candidate gripper design.

to the requirement of explicitly modeling dynamics, or assume good simulators.

In model-free approaches, the relationship between input actions and output states is learned through interaction, with the dynamics of the interaction implicitly modeled. The concept of transporter networks, crucial in the manipulation of complex deformable objects, is investigated in [41, 42]. In dynamic manipulation [12, 3, 6, 50], the action space is parameterized for efficiently learning a mapping to the goal state. PaperBot simultaneously learns to design *and* dynamically manipulate a deformable object to achieve a goal state with a neural surrogate model to learn the non-linear correlation between design and manipulation parameters and reward through experimentation, then performing inverse design.

III. APPROACH

A. Inverse Design Framework

PaperBot is a self-supervised framework for learning to design objects out of paper. Our approach performs real-world experiments with paper (cutting, folding) and automatically measures its performance on a task with a perception system. Given a parameterization of the design (such as where to cut or fold), we want to find a design $z \in \mathcal{Z} \subset \mathbb{R}^D$ that maximizes the reward function:

$$z^* = \operatorname{argmax}_z R(z) \quad (1)$$

where the reward $R(z) \in \mathbb{R}$ is a perceivable quantity, such as travel distance or amount of force and \mathcal{Z} is a bounded design parameter space. In real-world settings, calculating the reward R requires building the paper tool, actuating a robot to use it, and measuring the reward with a perception system, i.e. $R(z) = \text{Perceive}(\text{Actuate}(\text{Build}(z), z))$. This process is neither convex nor differentiable, which makes optimizing the design challenging.

Our framework learns a differentiable surrogate model $f_\theta(z_t)$, represented by a neural network parameterized by θ , that is trained to predict the reward of a candidate design z_t :

$$\min_{\theta} \mathbb{E}_{z_t} [\mathcal{L}(f_\theta(z_t), r_t)] \quad \text{for } r_t = R(z_t) \quad (2)$$

where z_t is the design configuration sampled at iteration t and \mathcal{L} is the loss function. When sampling a new z_{t+1} , we use a ϵ -greedy sampling strategy in order balance exploration and exploitation in the design space:

$$z_{t+1} = \begin{cases} \operatorname{argmax}_z f_\theta(z) & \text{with probability } 1 - \epsilon \\ z \sim U & \text{with probability } \epsilon \end{cases} \quad (3)$$

where U is a uniform distribution over \mathcal{Z} .

After a sufficient number of trials (we found 100 trials to be reasonable), the final design can be estimated with gradient descent by solving:

$$\hat{z} = \operatorname{argmax}_z f_\theta(z) \quad (4)$$

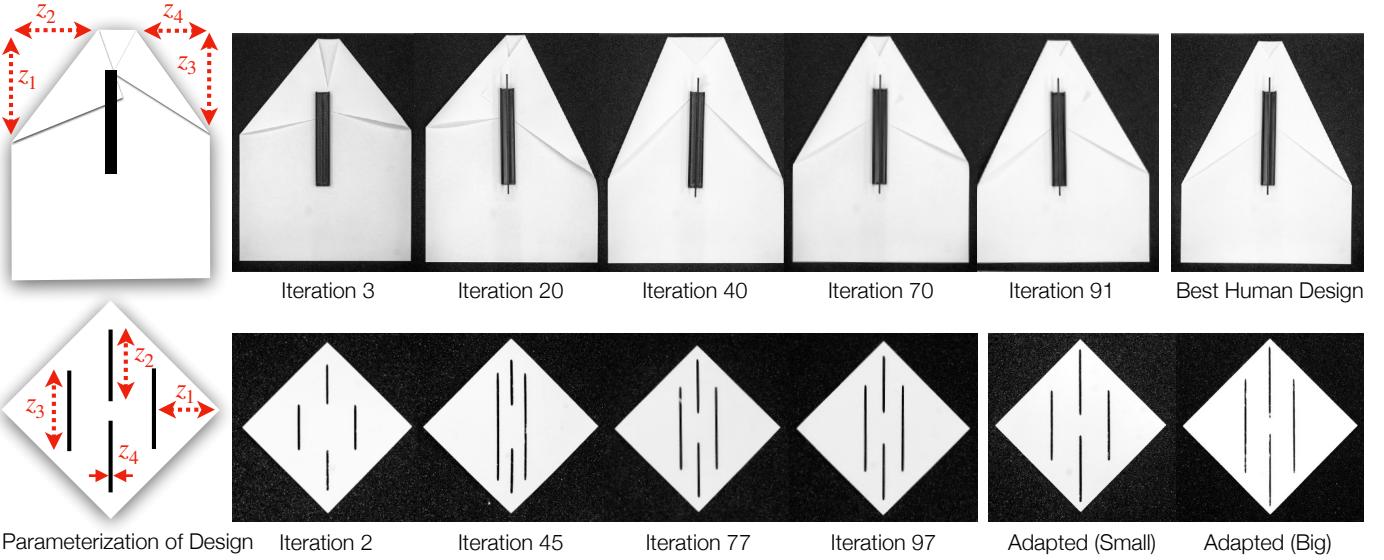


Fig. 5: Parameterization and Learning Process. Left shows the parameterization of the tools. Right shows the intermediate designs during the learning process as well as the best human-fold plane and adapted gripper designs for smaller and larger objects. These iterations correspond to the teaser figure. See Fig. 1 for the actual performance of these designs.

which approximates the optimal design z^* in Equation 1.

In this work, we instantiate this framework on two paper design tasks: building paper airplanes that maximize their travel distance, and building kirigami grippers that exert a desired gripping force. In the remainder of this section, we describe how we apply this framework to these two tasks.

B. Paper Airplane

We first consider the design of a paper airplane that travels a maximum distance. Prior works have explored similar tasks [33, 37] for different purposes, but here we use it as a motivating task for paper tool design as it involves a sequence of folding and throwing action as well as the interaction between deformable object and air, which is notably difficult to simulate.

Task Configuration. The objective of the task is to fold and throw a paper airplane that maximizes the flying distance. The workspace is a 9m x 1.5m runway with curtains and wall on the side to prevent plane from deviating. A carpet was placed on the ground to prevent a paper airplane from sliding upon landing. We use the letter-size printer paper (216mm x 279mm) with 75g/mm² as material. The distance traveled by the plane is monitored by two RealSense D435i cameras mounted on the ceiling looking down with enough field of view to cover the entire runway.

Design and Action Space. The parameter space of the task is composed of 4 plane design parameters as described in Fig. 5. We also learn 1 dynamic manipulation parameter of the throwing motion. Parameters 1 and 2 define the left fold position and orientation and parameters 3 and 4 define the right fold position and orientation. Parameters 5 is the throwing angle of the robot arm defined by when the gripper is released while the robot arm performs the throwing motion.

Automation. We automate paper airplane folding with two xArm 7 robot arms, one equipped with a parallel gripper and the other equipped with a 3D-printed soft presser tool and a suction cup. During folding, the corner of the paper is grasped by the parallel gripper and moved to the parameterized position, and the other arm will press down along the crease. The soft design of the presser tool exempts us from impedance control and did not fail once during more than a thousand trials we have run. To robustly load a single page of paper, we used a cheap inkjet printer that prints a blank page. To simplify the problem, after two folds, a 3D printed holder with pre-applied double-sided tape at the bottom is used to stick to the paper as a grasping point for throwing.

The pipeline for folding and throwing a paper airplane is fully automated without any human intervention. Each trial takes around 100 seconds, allowing 100 trials to be performed within 3 hours.

C. Kirigami Gripper

Kirigami grippers are inspired by the traditional art of paper cutting and folding, and they have recently been shown to be a novel approach for grasping in robotics [53]. Constructed from a single flat sheet, these grippers are not only cost-effective and lightweight but also environmentally friendly, making them ideal for a variety of object grasping tasks in dynamic and unstructured environments [15, 16, 2, 23]. Nevertheless, the analysis and simulation of kirigami grippers pose significant challenges due to their complex behavior, including multiple buckling modes and reconfigurations upon contact with objects [4, 1]. This complexity requires sophisticated analysis and human expertise for kirigami gripper design in current methodologies. To resolve this, we propose to directly learn a surrogate model through experimentation in the real world

Algorithm 1: ϵ -Greedy Tool Design Optimization

Input: Neural network f with parameters θ and initialization σ , automation pipeline R , explore probability ϵ , loss function \mathcal{L} , number of total trials T , optimization step sizes $\lambda_z, \lambda_\theta$, number of iterations N, M

Output: Estimated design parameters \hat{z}

```

for  $i = 1, \dots, T$  do
    if  $\text{rand}() \leq \epsilon$  then
         $z_0 \leftarrow$  randomly sampled design parameters
         $\{z\}_i = \{z\}_{i-1} \cup z_0$ 
         $\{r\}_i = \{r\}_{i-1} \cup R(z_0)$ 
    else
         $\theta_0 \sim \mathcal{N}(0, \sigma)$ 
        // Fitting surrogate model with data
        for  $j = 1, \dots, N$  do
             $B \leftarrow$  sample mini-batch from  $\{z\}_i, \{r\}_i$ 
             $L \leftarrow \sum_{z, r \in B} \mathcal{L}(f_{\theta_{j-1}}(z), r)$ 
             $\theta_j \leftarrow \theta_{j-1} - \lambda_\theta \frac{\partial}{\partial \theta_{j-1}} L$ 
        end
        // Inverse design
         $z_0 \leftarrow$  randomly sampled design parameters
        for  $j = 1, \dots, M$  do
             $z_j \leftarrow z_{j-1} + \lambda_z \frac{\partial}{\partial z_{j-1}} f_{\theta_N}(z_{j-1})$ 
        end
         $\{z\}_i = \{z\}_{i-1} \cup z_M$ 
         $\{r\}_i = \{r\}_{i-1} \cup R(z_M)$ 
    end
end
 $t^* \leftarrow \text{argmax } \{r\}_T$ 
return  $z_{t^*}$ 

```

and invert it for design.

Task Configuration. The objective of this task is to design a gripper that exerts maximum gripping force for a given object geometry. The created gripper is placed on top of a measuring device with two parallel load cells measuring the horizontal gripping force exerted on each side of the gripper. A sequence of 240 measurements ($80 \text{ Hz} \times 3\text{s}$) is collected and applied with median filtering with a window size of 5. The peak value among a 10-step moving average is taken as the maximum force exerted by the Kirigami gripper. A sublimation paper (120 gsm) is used as the material².

Design and Action Space. For simplicity and practicality, we keep the size of each gripper at 3x3 inches. We use 4 parameters to control the design of the gripper as described in Fig. 5. This parameterization is functionally equivalent to the original design in [53], but more convenient for cutting. We limit the parameter space so that the cuts do not cross each other and do not get too close to the edge of the paper. Please see the detailed parameterization in Fig. 5.

Automation. We use a Cricut Maker 3 machine, which takes

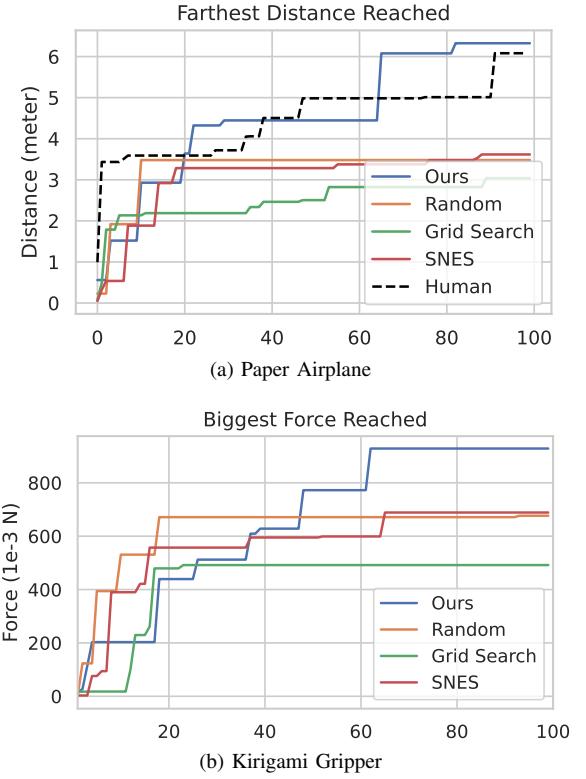


Fig. 6: **Performance** of paper airplanes (top) and kirigami grippers (bottom) per optimization iteration, designed by our method (blue) versus baselines.

a PNG image of the designed gripper and cuts a sheet of paper into the exact shape. The cut gripper is placed to the force-measuring device shown in Fig. 4. Two xArm 7s with parallel grippers grasp the two opposite corners of the gripper and pull towards the opposite direction with 10mm/s speed for 30mm, during which both sides of the gripper press against the load cells. Different-sized 3D printed objects are mounted at the end of the load cells to mimic different-sized objects during adaptation experiments.

D. Design Optimization

After automating the construction and perception of the reward, we want to find a design z that maximizes the reward function R for a task. We simultaneously learn a surrogate model $f_\theta(z)$ and optimize designs against it.

Learning the Surrogate Model. We instantiate the surrogate model f as a 4-layer MLP with an input dimension matching the corresponding parameter space, hidden dimensions of 512, and a scalar output for reward. Given a dataset of pairs of input design parameters and output reward, we train the network for 1000 iterations with a batch size of 8. We use an AdamW optimizer with learning rate of 0.01 and weight decay of 0.1 to optimize for the Huber loss (smoothed L1 loss) between the predicted reward and ground truth.

Since we are operating in the real world, which is expensive

²<https://a.co/d/2WK65p7>

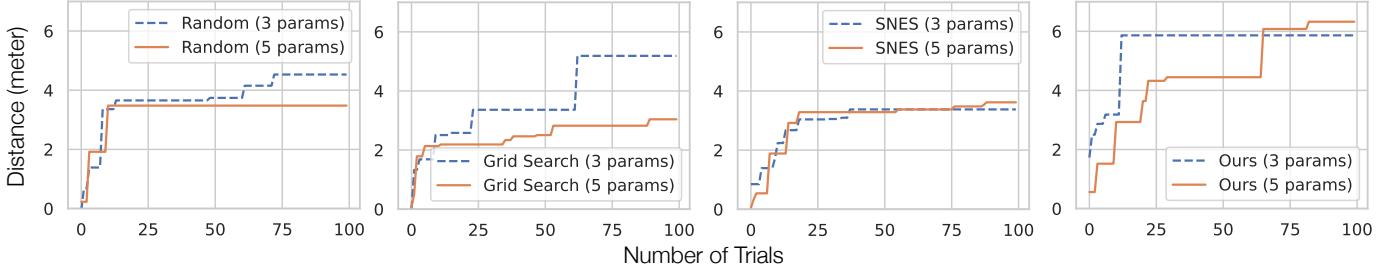


Fig. 7: **Exploring Larger Design Spaces.** When the paper airplane design space increases from 3 to 5, random and grid search performs significantly worse. In contrast, our approach can efficiently navigate the larger design space to find better design/throwing parameters. Further, regardless of the design space, our approach significantly outperforms all baselines.

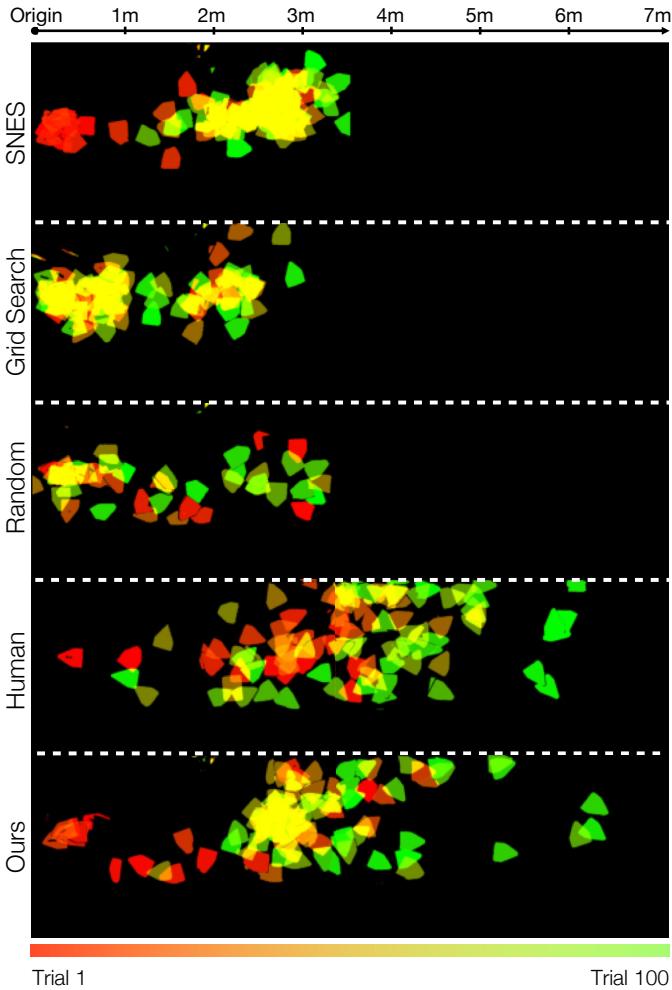


Fig. 8: **Paper Airplanes Landing Positions.** We visualize the top-down paper airplane mask over the course of training. Compared to SNES, PaperBot was able to better balance exploration and exploitation, escaping the local minima at around 2.5 meters, and eventually beating the human baseline.

in terms of time and resources, we cannot afford to massively randomly sample design parameters as training data for the surrogate model. For this, we adopt an ϵ -greedy exploration strategy. At each iteration, there is an ϵ probability that we

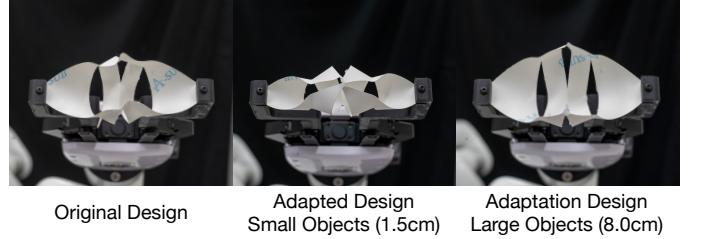


Fig. 9: **Adapted Grippers Under Actuation.** Changing the distance between load cells changes where gripping force is measured, and thus, the optimal bending point for the gripper designs. Given only 50 adaptation trials, PaperBot discovered subtle design changes which significantly impacted the mechanical structure of the paper, leading to grippers tailored to different object sizes.

	Small (1.5cm)	Large (8cm)
Original (optimized for 5cm)	0.302 ± 0.012	0.055 ± 0.027
Adapted Gripper	0.442 ± 0.080	1.131 ± 0.235

TABLE I: Force measurement of original gripper optimized for grasping 5cm-sized objects compared with adapted grippers optimized for different object sizes.

will perform random exploration, and otherwise, we perform a greedy search by finding the best design parameters defined by the neural surrogate model. Please see Algorithm 1 for a detailed description.

Inverse Design. Once the surrogate model is trained, we randomly sample a set of initial design parameters z and perform gradient-descent on Equation 4. During each iteration, we calculate $z \leftarrow z + \lambda \frac{\delta f_\theta(z)}{\delta z}$ for a scalar step size $\lambda \in \mathbb{R}$, which can be efficiently computed with back-propagation. The gradient contains information regarding which directions to change the design parameters to increase the reward. If the gradient update causes z to deviate outside of physically feasible actions (such as cutting off the paper surface), we project z back to the nearest feasible region.

Because the surrogate model training and inverse design combined take less than 2 seconds on an NVIDIA RTX 3090 GPU, far from being the bottleneck during experiments, we perform these two steps before each trial as detailed in

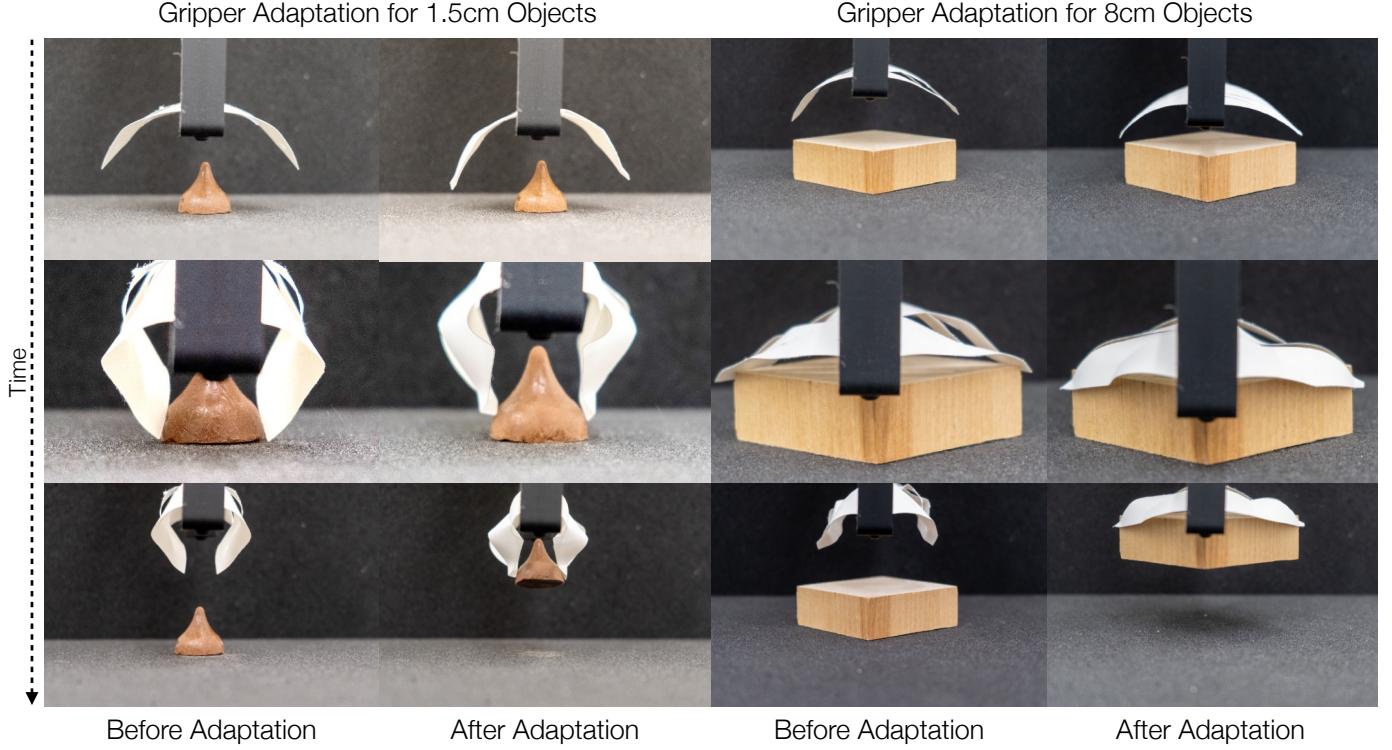


Fig. 10: **Grasping with Adapted Gripper Design.** We compare the grasping process of our gripper design before adaptation and after adaptation for larger and smaller objects. Note that the “before adaptation” gripper is optimized for 5cm objects.

Algorithm 1.

It is worth mentioning that evolutionary algorithms [11, 10, 39, 9, 13] is another class of algorithms that also attempts to address the problem of gradient-free non-convex optimization. However, this class of methods is known to be sample-inefficient thus is not feasible in our real-world setting. We show the experimental comparison in section IV-A.

IV. EVALUATION

The goal of this section is to systematically evaluate our system with both quantitative and qualitative analysis. In section IV-A, we compare our method against several baselines to validate the effectiveness and efficiency of our optimization algorithm. In section IV-B, we show the effect of increasing the size of parameter space on ours and baseline methods. In section IV-C, we show how our pre-trained model in one environment can adapt to a new environment with much less data. Finally, in section IV-D, we show applications of our optimized tool design.

A. Quantitative Experiments

Baselines. For both tasks, we selected several baselines for comparison.

- *Grid Search:* A typical approach for parameter optimization is grid search. Given 100 trials as a budget, we used 5-5-4 for 3-parameter problems, 3-3-3-4 for 4-parameter problems, and 3-3-3-2-2 for 5-parameter problems.

- *Random Sampling:* We perform random sampling of design parameters for each trial. Random and grid search provide a holistic view of the design problems’ difficulty.
- *SNES [40]:* Evolutionary algorithms are common choices for solving gradient-free non-convex optimization problems. It has also been used in prior works for tool design based on simulation. We ran the algorithm with a population size of 5 for 20 iterations. We adopted the implementation provided by *EvoTorch*.

Paper Airplane. In the paper airplane task, as shown in Fig. 6, our approach significantly outperforms other baselines by a large margin, outperforming the second best baseline, SNES, by 87%.

To further understand the results, we visualize the landing positions of all folded paper airplanes during all experiment runs. This is collected by the ceiling-mounted RealSense cameras. The masks of planes of all trials are combined to visualize the distances traveled in Fig. 8. In both grid search and random exploration experiments, the majority of the folded paper airplanes landed near the origin, and overall show similar distance distribution. In SNES [40], most of the planes landed between 2.5-3.5 meters, while in comparison, our method quickly breaks out of that region and reaches further and further distances. During experiments, we observed that the evolutionary algorithm can quickly learn the throwing angle, which has the highest correlation with the travel distance and is relatively orthogonal to other design parameters. However, during the 100 trials, SNES struggled to find good folding



Fig. 11: **Grasping of everyday objects using the optimized gripper design.** We used the grippers with a design optimized by our method to perform grasping of objects with various sizes, surface materials, weights, and rigidity, as well as fresh fruits as shown in the bottom row.

parameters, which are necessary for folding a paper airplane that flies further than 3.5 meters. This explains what we see in the visualization and shows that our method is significantly more sample-efficient than evolutionary algorithm for this problem in a low-data setting.

Kirigami Gripper. For the kirigami gripper design task, we adopted the same set of baselines. We optimize the 4 design parameters as shown in Fig. 5. Similar to the paper airplane task, our method outperforms the second-best baseline – SNES by a large margin of 36.4%. Our best-designed gripper can exert a gripping force equivalent to the gravity of 92.8g, which is more than the weight of four strawberries.

B. Size of Design Space

Since real-world tools are complicated and contain many design parameters, we want to understand the scaling performance of different methods against the number of parameters. To investigate this, we perform an ablation study on the paper airplane task. We run our experiments to learn 3 and 5 parameters independently. For the 3 parameter experiments, we cut the two parameters governing the right fold, and

replaced them with the parameters of the left fold, enforcing a symmetry of wing angles to every plane – reducing the complexity of the design task.

As shown in figure 7, the performance of baseline methods drops significantly as the number of learnable parameters increases. All of these methods struggle to learn to fold symmetrically, which causes most of the folded planes to crash on the side. However, for our method, even though the rate of learning in the 5-parameter experiment is much slower than the 3-parameter one, it eventually found a better design. We observed that this is because our enforced symmetry does not guarantee the plane travel in a straight line due to both calibration precision and environmental influence (such as small airflow on the paper runway). When given 5 learnable parameters, our method can discover a slightly asymmetric design that counteracts the systematic asymmetry, allowing the plane to travel in a straight line to the furthest distance.

C. Adaptation of Kirigami Gripper to Different Objects

Since the entire learning process happens in the real world and we do not assume prior knowledge about the physical world, our system is highly adaptive. In the kirigami gripper

design task, we conduct experiments where we vary the distance between the load cells to mimic objects of different sizes. In these experiments, we initialized the parameters of the neural surrogate model with pretrained weights from previous experiments conducted on a different object size. With fewer experiments (50 trials), our method can quickly adapt its design for different object sizes.

Table I shows the force measurement of the original gripper vs. adapted gripper exerted on new objects. For the small object experiment, the adapted gripper exerts 46% more gripping force than before. For larger objects, the adapted gripper obtains a more than 20-fold increase in gripping force.

The effect of adaptation of gripper design can be qualitatively seen in Fig. 9, where we mount different grippers on a parallel robot gripper and actuate with the same stretching distances. The difference in design changes the mechanical structure of the paper during its deformation, causing it to enclose objects of different sizes much better. This effect directly causes the success rate of grasping objects of different sizes as shown in Fig. 10, illustrating the importance of customization. These experiments showcase the advantage of our method to quickly adapt to different reward definitions and potentially different environmental conditions. It has also shown that prior knowledge obtained from a related task with a shared structure can be effectively transferred to improve sample efficiency.

D. Grasping with Designed Gripper

In Fig. 11, the robot performs grasping on a variety of objects with different sizes and geometry to showcase the effectiveness of the emergent gripper design. In each example, we mount the kirigami gripper on a normal robot parallel gripper using a 3D printer tool and perform a predefined primitive grasping action. These objects include small, high-density objects such as the metal nut, as well as large objects with a smooth slippery surface such as the box, highlighting that our optimized design can exert high gripping force for grasping. The bottom row shows our designed gripper grasping various fruits, which motivates paper as a material for making cheap, clean, recyclable, and customizable grippers for handling food.

V. CONCLUSION

Paper is an affordable and versatile medium that can be used to construct many different types of tools. We demonstrated PaperBot, a system that autonomously performs experiments to simultaneously learn the a) design and b) use of paper tools for different tasks. Without requiring a simulator, experiments show that our approach is able to learn to design, build, and use real-world paper tools in just a few hours. Our system is reproducible in any robotics research lab and we will open-source all software, hardware, models, and data.

ACKNOWLEDGMENTS

This research is based on work partially supported by NSF NRI Awards #1925157 and #2132519, and the Toyota Research Institute. S.S. is supported by the NSF Graduate Research Fellowship.

REFERENCES

- [1] Joao Buzzatto, Mojtaba Shahmohammadi, Junbang Liang, Felipe Sanches, Saori Matsunaga, Rintaro Haraguchi, Toshisada Mariyama, Bruce MacDonald, and Minas Liarokapis. Soft, multi-layer, disposable, kirigami based robotic grippers: On handling of delicate, contaminated, and everyday objects. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5440–5447. IEEE, 2022. 5
- [2] Joao Buzzatto, Junbang Liang, Mojtaba Shahmohammadi, Saori Matsunaga, Rintaro Haraguchi, Toshisada Mariyama, Bruce A MacDonald, and Minas Liarokapis. A soft, multi-layer, kirigami inspired robotic gripper with a compact, compression-based actuation system. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4488–4495. IEEE, 2023. 2, 5
- [3] Alper Canberk, Cheng Chi, Huy Ha, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5872–5879. IEEE, 2023. 4
- [4] G Chaudhary, L Niu, Q Han, M Lewicka, and L Mahadevan. Geometric mechanics of ordered and disordered kirigami. *Proceedings of the Royal Society A*, 479(2274):20220822, 2023. 5
- [5] Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning. *arXiv preprint arXiv:2008.04460*, 2020. 3
- [6] Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Iterative residual policy for goal-conditioned dynamic manipulation of deformable objects. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022. 4
- [7] Gary PT Choi, Levi H Dudte, and Lakshminarayanan Mahadevan. Programming shape using kirigami tessellations. *Nature materials*, 18(9):999–1004, 2019. 2
- [8] Matteo Cianchetti, Cecilia Laschi, Arianna Menciassi, and Paolo Dario. Biomedical applications of soft robotics. *Nature Reviews Materials*, 3(6):143–153, 2018. 2
- [9] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR, 2016. 8
- [10] Tobias Glasmachers, Tom Schaul, Sun Yi, Daan Wierstra, and Jürgen Schmidhuber. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 393–400, 2010. 8
- [11] Faustino Gomez, Jürgen Schmidhuber, Risto Miikkulainen, and Melanie Mitchell. Accelerated neural evolution through cooperatively coevolved synapses. *Journal*

- of Machine Learning Research*, 9(5), 2008. 8
- [12] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robotic Learning (CoRL)*, 2021. 4
- [13] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001. 8
- [14] Zhanpeng He and Matei Ciocarlie. Morph: Design co-optimization with reinforcement learning via a differentiable hardware model proxy. *arXiv preprint arXiv:2309.17227*, 2023. 2
- [15] Yaoye Hong, Yinding Chi, Shuang Wu, Yanbin Li, Yong Zhu, and Jie Yin. Boundary curvature guided programmable shape-morphing kirigami sheets. *Nature Communications*, 13(1):530, 2022. 2, 5
- [16] Yaoye Hong, Yao Zhao, Joseph Berman, Yinding Chi, Yanbin Li, He Huang, and Jie Yin. Angle-programmed tendril-like trajectories enable a multifunctional gripper with ultradelicacy, ultrastrength, and ultraprecision. *Nature Communications*, 14(1):4625, 2023. 2, 5
- [17] Jiaheng Hu, Julian Whitman, Matthew Travers, and Howie Choset. Modular robot design optimization with generative adversarial networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4282–4288. IEEE, 2022. 3
- [18] Jiaheng Hu, Julian Whitman, and Howie Choset. Glso: Grammar-guided latent space optimization for sample-efficient robot design automation. In *Conference on Robot Learning*, pages 1321–1331. PMLR, 2023. 3
- [19] Zhiao Huang, Yuanming Hu, Tao Du, Siyuan Zhou, Hao Su, Joshua B. Tenenbaum, and Chuang Gan. Plasticinelab: A soft-body manipulation benchmark with differentiable physics, 2021. 3
- [20] Doh-Gyu Hwang and Michael D Bartlett. Tunable mechanical metamaterials through hybrid kirigami structures. *Scientific reports*, 8(1):3378, 2018. 2
- [21] Midori Isobe and Ko Okumura. Initial rigid response and softening transition of highly stretchable kirigami sheet materials. *Scientific reports*, 6(1):24758, 2016. 2
- [22] Donghwa Jeong and Kiju Lee. Design and analysis of an origami-based three-finger manipulator. *Robotica*, 36(2):261–274, 2018. 2
- [23] Gyeongji Kang, Young-Joo Kim, Sung-Jin Lee, Se Kwon Kim, Dae-Young Lee, and Kahye Song. Grasping through dynamic weaving with entangled closed loops. *Nature Communications*, 14(1):4633, 2023. 2, 5
- [24] Suk-Jun Kim, Dae-Young Lee, Gwang-Pil Jung, and Kyu-Jin Cho. An origami-inspired, self-locking robotic arm that can be folded flat. *Science Robotics*, 3(16):eaar2915, 2018. 2
- [25] Milin Kodnongbua, Ian Good Yu Lou, Jeffrey Lipton, and Adriana Schulz. Computational design of passive grippers. *arXiv preprint arXiv:2306.03174*, 2023. 2, 3
- [26] Aaron Lamoureux, Kyusang Lee, Matthew Shlian, Stephen R Forrest, and Max Shtain. Dynamic kirigami structures for integrated solar tracking. *Nature communications*, 6(1):8092, 2015. 2
- [27] Dae-Young Lee, Jae-Kyeong Kim, Chang-Young Sohn, Jeong-Mu Heo, and Kyu-Jin Cho. High-load capacity origami transformable wheel. *Science Robotics*, 6(53):eabe0201, 2021. 2
- [28] S Li, JJ Stampfli, and HJ Xu. A vacuum-driven origami “magic-ball” soft gripper” in 2019 international conference on robotics and automation (icra). *IEEE, New York*, pages 7401–7408, 2019. 2
- [29] Xingyu Lin, Zhiao Huang, Yunzhu Li, Joshua B. Tenenbaum, David Held, and Chuang Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. 2022. 3
- [30] Ke Liu, Tomohiro Tachi, and Glaucio H Paulino. Invariant and smooth limit of discrete geometry folded from bistable origami leading to multistable metasurfaces. *Nature communications*, 10(1):4238, 2019. 2
- [31] Ziang Liu, Stephen Tian, Michelle Guo, Karen Liu, and Jiajun Wu. Learning to design and use tools for robotic manipulation. In *Conference on Robot Learning*, pages 887–905. PMLR, 2023. 2, 3
- [32] Rahul Narain, Tobias Pfaff, and James F O’Brien. Folding and crumpling adaptive sheets. *ACM Transactions on Graphics (TOG)*, 32(4):1–8, 2013. 3
- [33] Nana Obayashi, Kai Junge, Stefan Ilić, and Josie Hughes. Robotic automation and unsupervised cluster assisted modeling for solving the forward and reverse design problem of paper airplanes. *Scientific Reports*, 13(1), Mar 2023. doi: 10.1038/s41598-023-31395-0. 5
- [34] Nikhil Padhye and Subodh Kalia. Isogeometric analysis of elastic sheets exhibiting combined bending and stretching using dynamic relaxation. *arXiv preprint arXiv:2206.10406*, 2022. 3
- [35] Chuan Qiao, Lu Liu, and Damiano Pasini. Elastic thin shells with large axisymmetric imperfection: From bifurcation to snap-through buckling. *Journal of the Mechanics and Physics of Solids*, 141:103959, 2020. 3
- [36] Ahmad Rafsanjani, Yuerou Zhang, Bangyuan Liu, Shmuel M Rubinstein, and Katia Bertoldi. Kirigami skins make a simple soft actuator crawl. *Science Robotics*, 3(15):eaar7555, 2018. 2
- [37] ABB Robotics, Sep 2015. URL <https://www.youtube.com/watch?v=KWMtx9QotGk>. 5
- [38] Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9798–9805. IEEE, 2019. 3
- [39] Tom Schaul, Tobias Glasmachers, and Jürgen Schmidhuber. High dimensions and heavy tails for natural evolution strategies. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 845–852, 2011. 8
- [40] Tom Schaul, Tobias Glasmachers, and Jürgen Schmidhuber. High dimensions and heavy tails for natural

- evolution strategies. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, page 845–852, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450305570. doi: 10.1145/2001576.2001692. URL <https://doi.org/10.1145/2001576.2001692>. 8
- [41] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4568–4575. IEEE, 2021. 4
- [42] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks, 2023. 4
- [43] Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. *arXiv preprint arXiv:2205.02909*, 2022. 3
- [44] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *arXiv preprint arXiv:2306.14447*, 2023. 3
- [45] Orion Taylor and Alberto Rodriguez. Optimal shape and motion planning for dynamic planar manipulation. *Autonomous Robots*, 43:327–344, 2019. 2, 3
- [46] Tsun-Hsuan Wang, Juntian Zheng, Pingchuan Ma, Yilun Du, Byungchul Kim, Andrew Spielberg, Joshua Tenenbaum, Chuang Gan, and Daniela Rus. Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. *arXiv preprint arXiv:2311.17053*, 2023. 3
- [47] Julian Whitman, Raunaq Bhirangi, Matthew Travers, and Howie Choset. Modular robot design synthesis with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10418–10425, 2020. 3
- [48] Shuai Wu, Qiji Ze, Jize Dai, Nupur Udipi, Glaucio H Paulino, and Ruike Zhao. Stretchable origami robotic arm with omnidirectional bending and twisting. *Proceedings of the National Academy of Sciences*, 118(36):e2110023118, 2021. 2
- [49] Jie Xu, Andrew Spielberg, Allan Zhao, Daniela Rus, and Wojciech Matusik. Multi-objective graph heuristic search for terrestrial robot design. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 9863–9869. IEEE, 2021. 2, 3
- [50] Zhenjia Xu, Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Dextairity: Deformable manipulation can be a breeze. *arXiv preprint arXiv:2203.01197*, 2022. 4
- [51] Kshitij Kumar Yadav, Nicholas L Cuccia, Emmanuel Virot, Shmuel M Rubinstein, and Simos Gerasimidis. A nondestructive technique for the evaluation of thin cylindrical shells' axial buckling capacity. *Journal of Applied Mechanics*, 88(5):051003, 2021. 3
- [52] Wenzhong Yan and Ankur Mehta. A cut-and-fold self-sustained compliant oscillator for autonomous actuation of origami-inspired robots. *Soft Robotics*, 9(5):871–881, 2022. 2
- [53] Yi Yang, Katherine Vella, and Douglas P. Holmes. Grasping with kirigami shells. *Science Robotics*, 6(54):eabd6426, 2021. doi: 10.1126/scirobotics.abd6426. URL <https://www.science.org/doi/abs/10.1126/scirobotics.abd6426>. 5, 6
- [54] Qiji Ze, Shuai Wu, Jize Dai, Sophie Leanza, Gentaro Ikeda, Phillip C Yang, Gianluca Iaccarino, and Ruike Renee Zhao. Spinning-enabled wireless amphibious origami millirobot. *Nature communications*, 13(1):3118, 2022. 2
- [55] Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Robogrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020. 3