

# Robustness-Aware Tool Selection and Manipulation Planning with Learned Energy-Informed Guidance

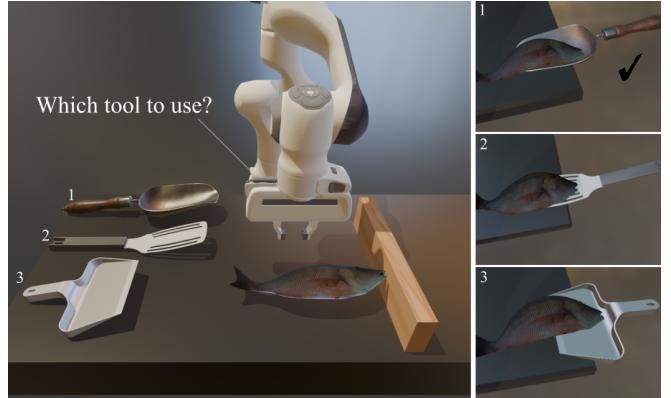
Yifei Dong<sup>1,\*</sup>, Yan Zhang<sup>2,\*</sup>, Sylvain Calinon<sup>2</sup>, Florian T. Pokorny<sup>1</sup>

**Abstract**— Humans subconsciously choose robust ways of selecting and using tools, based on years of embodied experience—for example, choosing a ladle instead of a flat spatula to serve meatballs. However, robustness under uncertainty remains underexplored in robotic tool-use planning. This paper presents a robustness-aware framework that jointly selects tools and plans contact-rich manipulation trajectories, explicitly optimizing for robustness against environmental disturbances. At the core of our approach is a learned, energy-based robustness metric, which guides the planner towards robust manipulation behaviors. We formulate a hierarchical optimization pipeline that first identifies a tool and configuration that optimizes robustness, and then plans a corresponding manipulation trajectory that maintains robustness throughout execution. We evaluate our approach across three representative tool-use tasks. Simulation and real-world results demonstrate that our approach consistently selects robust tools and generates disturbance-resilient manipulation plans.

## I. INTRODUCTION

Humans develop an intuition for robustness in tool use from an early age. Even with limited precision, babies learn to complete tasks under uncertainty, and adults subconsciously leverage experience to select and use tools in robust ways. As shown in Fig. 1, different tools can lead to significantly different outcomes under disturbance, highlighting the importance of robustness in both selection and use. Despite this, robustness in robotic tool use remains underexplored, as most prior work prioritizes task completion over disturbance resilience. Achieving human-like robustness in robots is challenging for two reasons. First, tools vary widely in geometry and physical properties, making manipulation outcomes highly sensitive to the combined configuration of tool, robot, and object. Second, robustness requires not only planning effective tool-use strategies, but also selecting the best tool from a diverse set—accounting for complex contact dynamics in uncertain, contact-rich environments.

To address these challenges, we propose a unified framework that jointly selects the most robust tool and plans a corresponding manipulation trajectory. The problem is



**Fig. 1:** How can the robot choose the best tool to scoop and lift a fish so it does not fall out of the tool during manipulation? This work introduces a robustness-aware planner that selects and uses tools effectively under uncertainty.

formulated as a hierarchical optimization process: (i) selecting a tool and tool-object interaction that is the most robust, and (ii) planning a manipulation trajectory passing through the selected interaction configuration while maintaining the robustness throughout task execution with the best-found tool. A learned, energy-informed metric enables efficient robustness evaluation and guidance at both stages of optimization. To offload the computationally expensive robustness labeling process, we train a neural network offline on a dataset of tool-object configurations and corresponding robustness scores. We evaluate our framework on three representative tasks—tape pulling, scissors hooking, and fish scooping—encompassing rigid, articulated, and deformable objects. Experimental results demonstrate that our method effectively selects a tool and manipulation trajectory that is most robust to random force disturbances.

In summary, our contributions include: (i) a robustness-aware optimization framework that jointly selects tools and plans contact-rich manipulation trajectories, prioritizing tools and configurations robust to uncertainties, (ii) a learned energy-informed robustness metric derived from caging analysis, enabling efficient online robustness guidance, and (iii) simulation and real-world experiments demonstrating improved robustness and reliability compared to a baseline method.

## II. RELATED WORK

### A. Robust Manipulation

Robust manipulation aims to ensure consistent and reliable task execution in the presence of uncertainty—whether

<sup>1</sup>The authors are with the division of Robotics, Perception and Learning, KTH Royal Institute of Technology, 10044 Stockholm, Sweden.

<sup>2</sup>The authors are with the Idiap Research Institute, CH-1920 Martigny, Switzerland and also with the EPFL, 1015 Lausanne, Switzerland. Authors with \* contributed equally. This work was supported by the European Commission’s Horizon Europe Program SoftEnable project (<https://softenable.eu/>), and the State Secretariat for Education, Research and Innovation in Switzerland for participation in the European Commission’s Horizon Europe Program through the INTELLIMAN project (<https://intelliman-project.eu/>) and the SESTOSENSE project (<http://sestosenso.eu/>). Contact: yifeid@kth.se, yan.zhang@idiap.ch.

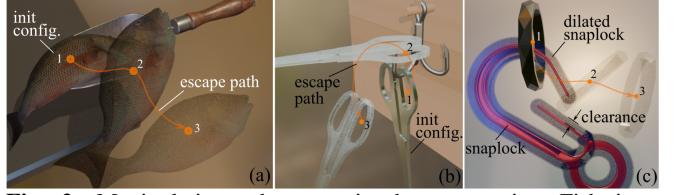
from noisy sensing, unmodeled dynamics, or unpredictable environmental interaction [1], [2]. Mason highlights the importance of robustness in classical tasks such as crank turning and peg-in-hole insertion, which succeed only if the system tolerates errors in object pose or robot motion [3]. Common sources of uncertainty include geometric ambiguity due to limited perception [4], [5] and dynamic uncertainty from contact interactions [6], [7]. Traditional approaches focus on prehensile manipulation, using analytic grasp metrics [8], [9] or learning-based estimators [10]–[12] to select robust grasps. Environmental contacts are often exploited by humans to reduce uncertainties during manipulation—for example, sliding an object along a surface to correct pose uncertainty before grasping it. Such uncertainty reduction strategies with action rather than sensing are formalized in funnels [13] and applied in in-hand robust manipulation [2].

In contrast, robustness in non-prehensile manipulation remains underexplored. Caging [14]–[16] provides valuable insights into addressing these challenges. Caging is a strategy that prevents objects from escaping through geometric constraints without relying on force or form closure, allowing for the tolerance of geometric uncertainties. Energy-bounded caging and its variants [17]–[21] extend caging by incorporating energy constraints on the object. In this work, we integrate an energy-bounded caging metric as robustness guidance into a tool-use manipulation planner. This data-driven approach improves online efficiency by offloading heavy computations of robustness labeling offline. It identifies robust tool-object configurations from an energy perspective and guides robust manipulation synthesis.

### B. Manipulation with Tools

Tools can significantly extend the dexterity of robotic end-effectors, enabling more complicated manipulation tasks such as flipping pancakes [22], rolling dough [23], [24], scooping tofu with a spoon [25], pouring water with a box [26], throwing with a ball [27], cutting food with a knife [28], cleaning with a broom [29], hammering nails [30], or performing brain surgery using neurosurgical instruments [31]. Robotic tool use often involves one or more tools, multiple sequential actions, and goals that consist of several sub-goals [32]. In most existing robotic tool-use applications, tools are typically assumed to be either rigidly attached to the robot arm or grasped and held during use [33].

Tool selection is defined as the process of identifying the most appropriate tool from a set of candidates to complete a specific task [32]. This is a challenging problem due to the large combinatorial search space involved. Levihn et al. [34] propose an approach that reasons over physical constraints between objects to reduce computational complexity, while Xu et al. [35] utilize language models to guide tool selection or creation. Existing tool selection approaches often focus on semantic reasoning or functional feasibility, without modeling the geometry-dependent robustness of tool-object interactions. In contrast, we introduce a unified framework that explicitly evaluates and optimizes for manipulation robustness during tool selection and motion



**Fig. 2:** Manipulation robustness in three scenarios. Fish in a shovel (a) and scissors on a treble hook (b) remain secured unless large disturbances occur; keyframes (1–3) illustrate minimal-energy escape paths used to quantify robustness. The robustness of a keyring in a snaplock (c) can also be characterized by clearance, i.e., the minimal dilation needed to completely prevent escape.

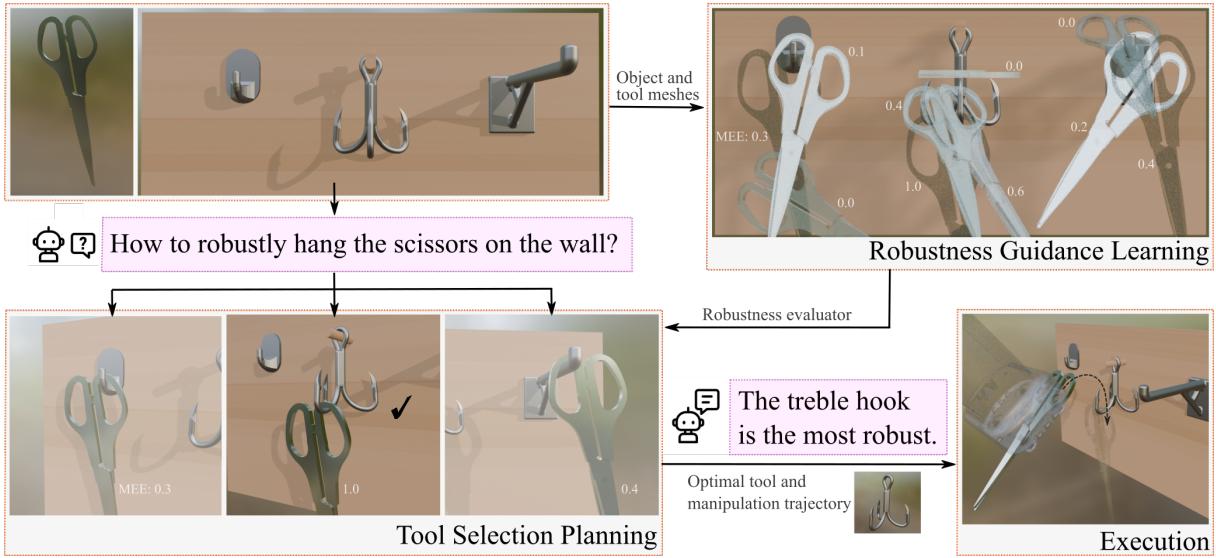
planning. Additionally, our research is closely related to recent studies on the co-design of robot behavior and tool morphology [36]–[39]. Unlike tool design works that can tune or generate geometry to maximize performance, we operate under stricter constraints—selecting the best tool and usage strategy from a fixed set of available tools.

### III. PRELIMINARY

Here, we introduce an energy-informed metric derived from caging analysis, which guides robustness-aware manipulation planning. Consider a rigid object’s configuration space (C-space), denoted as  $\mathcal{S}_{\text{obj}}$ , with  $\mathcal{S}_{\text{free}} \subset \mathcal{S}_{\text{obj}}$  representing its collision-free subset. In the presence of the force field, an object at configuration  $s_{\text{obj}} \in \mathcal{S}_{\text{free}}$  is associated with an energy value  $E(s_{\text{obj}}) : \mathcal{S}_{\text{free}} \rightarrow \mathbb{R}$ . The configuration’s robustness is characterized by the *Minimum Escape Energy* (MEE), denoted  $Q_{\text{mee}}$ . Intuitively, MEE  $Q_{\text{mee}}$  measures how much additional energy the object needs to “escape” from its current configuration  $s_{\text{obj}}$ . Take the fish in the shovel as an example (Fig. 2a). It has non-zero escape energy  $Q_{\text{mee}} > 0$ , meaning that it is secured by the shovel within a local subset of its configuration space and cannot escape unless external energy greater than  $Q_{\text{mee}}$  is applied. In practice, we estimate the MEE by searching for the corresponding energy-efficient collision-free escape paths using sampling-based algorithms, such as BIT\* [18], [40].

We consider three forms of the energy function  $E$ , depending on the task setting. (1) Under a gravitational field (Fig. 2b), the energy is given by  $E(s_{\text{obj}}) = m_{\text{obj}}g z_{\text{obj}}$ , where  $m_{\text{obj}}$  is the object’s mass,  $g$  is gravitational acceleration, and  $z_{\text{obj}}$  is its vertical position of its center of mass. (2) For planar pushing tasks, we consider an additional pushing energy field:  $E(s_{\text{obj}}) = m_{\text{obj}}g z_{\text{obj}} + F_p \hat{v} \cdot (x_{\text{obj}}, y_{\text{obj}})$ , where  $F_p$  is the pushing force magnitude,  $\hat{v} \in \mathcal{S}^1$  is a horizontal unit vector in the pushing direction, and  $(x_{\text{obj}}, y_{\text{obj}})$  is the object’s planar position [17]. (3) For deformable objects (Fig. 2a), the energy includes an elastic term:  $E(s_{\text{obj}}) = m_{\text{obj}}g z_{\text{obj}} + \frac{1}{2}k\alpha_{\text{obj}}^2$ , where  $k$  is the stiffness coefficient and  $\alpha_{\text{obj}}$  is the deformation angle [18].

We use *Partial Caging Clearance* (PCC) [41] as a baseline to compare against MEE. Unlike MEE, which defines robustness from an energy perspective, PCC quantifies robustness based on the clearance between the object and the caging tools. PCC  $Q_{\text{pcc}}$  is defined as the minimal tool’s



**Fig. 3:** Overview of the robustness-aware tool selection and manipulation planning framework.

geometric dilation offset  $\epsilon_{\min}$  required to prevent the object from escaping, provided no collision occurs (Fig. 2c). If such a collision-free offset  $\epsilon_{\min} \in [0, \epsilon_{\text{col}}]$  exists,  $Q_{\text{pcc}} = -\epsilon_{\min}$  and the object is “partially caged”, denoted  $q_{\text{pc}} = 1$ ; otherwise it is not partially caged  $q_{\text{pc}} = 0$  and  $Q_{\text{pcc}}$  is not defined. Practically, we approximate collision offset  $\epsilon_{\text{col}}$  using mesh distance and employ BIT\* to detect escape paths, determining the partial cage status  $q_{\text{pc}}$  accordingly.

#### IV. PROBLEM STATEMENT AND FORMULATION

Consider the problem of manipulating a target object  $o_{\text{obj}}$  from an initial configuration  $s_{\text{obj}}^0$  to a desired goal configuration  $s_{\text{obj}}^G$ , using a tool selected from  $N$  available ones  $\mathcal{O}_{\text{tool}} = \{o_i\}_{i=1}^N$ . The objective is to determine the best tool  $o_{\text{tool}}^* \in \mathcal{O}_{\text{tool}}$  and corresponding tool trajectory  $\tau_{\text{tool}}^* = \{s_{\text{tool}}^t\}_{t=0}^T$  that maximize robustness  $Q$  while manipulating target object to its goal configuration  $s_{\text{obj}}^G$ :

$$\min_{o_{\text{tool}}, \tau_{\text{tool}}} \|s_{\text{obj}}^G - s_{\text{obj}}^T\|_2 - \int_0^T Q(s_{\text{obj}}^t, o_{\text{tool}}, s_{\text{tool}}^t) dt, \quad (1a)$$

$$\text{s.t. } s_{\text{obj}}^{t+1}, s_{\text{tool}}^{t+1} = f(s_{\text{obj}}^t, o_{\text{tool}}, s_{\text{tool}}^t), \quad (1b)$$

$$g(s_{\text{obj}}^t, s_{\text{tool}}^t) \leq 0, \quad c(s_{\text{tool}}^t, r) \leq 0, \quad (1c)$$

where  $s_{\text{obj}}^T$  denotes the final configuration of the target object at the terminal time step  $T$ . The metric  $Q(s_{\text{obj}}^t, o_{\text{tool}}, s_{\text{tool}}^t)$  quantifies the robustness of the manipulation at each time step  $t$ , which depends on the selected tool  $o_{\text{tool}}$ , target object, and their configurations  $s_{\text{tool}}^t, s_{\text{obj}}^t$ . The constraint  $g(s_{\text{tool}}^t, s_{\text{obj}}^t)$  encodes the bounded free space  $\mathcal{S}_{\text{free}}$  between the tool and target object.  $c(s_{\text{tool}}^t, r)$  captures the kinematic constraints while the robot  $r$  manipulating the target object with tool  $o_{\text{tool}}$ . In the following, we present how we solve the optimization problem in Section V. Section VI details our data-driven approach to learning the robustness metrics  $Q$ , essential for robustness-aware guidance and computational efficiency of the optimization process. A framework overview is illustrated in Fig. 3.

#### V. ROBUSTNESS-AWARE TOOL SELECTION AND MANIPULATION PLANNING

We propose a hierarchical approach to solve Eq. (1). First, we perform keyframe optimization to identify the best tool  $o_{\text{tool}}^*$  and corresponding object and tool configurations  $s_{\text{obj}}^*, s_{\text{tool}}^*$  that maximize robustness  $Q$ . Given the best-found tool, we subsequently compute the manipulation trajectory  $\tau_{\text{tool}}^*$ , aiming to maintain robustness while passing through the configurations  $s_{\text{obj}}^*, s_{\text{tool}}^*$  at the keyframe timestep  $T_k$ .

##### A. Keyframe Optimization

We formulate the keyframe optimization problem as a mixed-integer program defined at a pre-specified keyframe timestep  $T_k \in [0, T]$ . The goal is to jointly select the most robust tool from the candidate set  $\mathcal{O}_{\text{tool}} = \{o_i\}_{i=1}^N$  and determine the corresponding tool and object configurations  $(s_{\text{tool}}, s_{\text{obj}})$  that maximizes the robustness metric  $Q$ :

$$\max_{o_{\text{tool}}, s_{\text{tool}}, s_{\text{obj}}} Q(s_{\text{tool}}, o_{\text{tool}}, s_{\text{obj}}), \quad (2a)$$

$$\text{s.t. } g(s_{\text{tool}}, s_{\text{obj}}) \leq 0, \quad c(s_{\text{tool}}, r) \leq 0. \quad (2b)$$

This yields the best-found tool  $o_{\text{tool}}^*$  and the corresponding configurations  $(s_{\text{tool}}^*, s_{\text{obj}}^*)$  at the keyframe timestep  $T_k$ . The resulting configurations serve as an intermediate waypoint that the robot aims to pass through during manipulation. Once established, the tool is used to guide the object toward its goal configuration  $s_{\text{obj}}^G$ , maintaining high robustness throughout the trajectory.

##### B. Robust Tool Manipulation Planning

Given the selected tool  $o_{\text{tool}}^*$  and keyframe configurations  $(s_{\text{tool}}^*, s_{\text{obj}}^*)$ , we plan the manipulation trajectory  $\tau_{\text{tool}}^*$  that maintains robustness while guiding the object to its goal

configuration. This is formulated as:

$$\min_{\tau_{\text{tool}}} \|s_{\text{obj}}^G - s_{\text{obj}}^T\|_2 + \|s_{\text{tool}}^* - s_{\text{tool}}^{T_k}\|_2 + \|s_{\text{obj}}^* - s_{\text{obj}}^{T_k}\|_2 - \int_0^T Q(s_{\text{obj}}^t, o_{\text{tool}}^*, s_{\text{tool}}^t) dt, \quad (3a)$$

$$\text{s.t. } s_{\text{obj}}^{t+1}, s_{\text{tool}}^{t+1} = f(s_{\text{obj}}^t, o_{\text{tool}}^*, s_{\text{tool}}^t), \quad (3b)$$

$$g(s_{\text{tool}}^t, s_{\text{obj}}^t) \leq 0, \quad c(s_{\text{tool}}^t, r) \leq 0. \quad (3c)$$

This formulation extends Eq. (1) by enforcing that the trajectory passes through the best-found configurations  $s_{\text{tool}}^*$ ,  $s_{\text{obj}}^*$  at  $T_k$ . By focusing the optimization on the selected tool  $o_{\text{tool}}^*$ , we avoid evaluating contact dynamics and robustness for all  $N$  candidates, reducing computational complexity.

### C. Optimizer

To solve the keyframe optimization problem in Eq. (2), we employ Covariance Matrix Adaptation Evolution Strategy (*CMA-ES*) [42]. We normalize all continuous optimization variables to the range  $[0, 1]$ . Discrete tool selection is handled by partitioning this range into  $N$  segments, each corresponding to a specific tool, enabling joint optimization of both tool identity and continuous tool-object configurations. For the full trajectory optimization problem described in Eq. (3), we utilize the Via-Point-Based Stochastic Trajectory Optimization (*VPSTO*) algorithm [43], a variant of CMA-ES. *VPSTO* introduces via-points as a low-dimensional, time-continuous representation of trajectories, which facilitates efficient optimization across the full planning horizon in complex, high-dimensional input spaces.

## VI. LEARNING ENERGY-INFORMED ROBUSTNESS GUIDANCE

To practically estimate the robustness guidance  $Q$  in Eq. (1), we adopt a data-driven approach to learn the energy-based metric  $Q_{\text{mee}}$  and the clearance-based metric  $Q_{\text{pcc}}$ , as detailed below.

### A. Data Collection

We curate a dataset of  $N_s$  samples for each object-tool pair by systematically randomizing their configurations within task-relevant bounds. For each sampled configuration, we use the *BIT\** algorithm [40] to approximate the underlying Minimum Escape Energy (MEE) in a physical simulator, which serves as our robustness metric under gravitational, elastic, or pushing force fields. For example, in the scissor hanging task with hooks (Fig. 3, top-right), we randomize the poses of the scissors  $s_{\text{obj}}$  for each hook. If it is collision-free, we compute the scissors' escape trajectory and corresponding robustness value  $Q_{\text{mee}}$  using *BIT\**. If an escape path cannot be found ( $Q_{\text{mee}} = \inf$ ), we consider the scissors completely caged and define a binary cage status label  $q_c = 1$ . Otherwise, the scissors are either energy-bounded caged or not caged, with  $q_c = 0$ . The resulting dataset consists of tuples  $(s_{\text{obj}}, s_{\text{tool}}, q_c, Q_{\text{mee}})$ , thus capturing diverse configurations and their robustness metrics, enabling effective supervised learning of manipulation robustness. Similarly, a dataset

$(s_{\text{obj}}, s_{\text{tool}}, q_{\text{pc}}, Q_{\text{pcc}})$  is created for the baseline robustness metric.

### B. Network Structure

We employ a Multi-Layer Perceptron (MLP) to learn the robustness metric in a supervised manner. Specifically, the input consists of the object's and the tool's configurations  $s_{\text{obj}}$ ,  $s_{\text{tool}}$ . The network comprises three fully-connected hidden layers with dimensions of 128, 128, and 64 neurons, respectively, each followed by ReLU activation functions and batch normalization. Furthermore, the output layer simultaneously predicts the cage status  $\hat{q}_c$  via a sigmoid-activated neuron and the MEE  $\hat{Q}_{\text{mee}}$  through a linear regression neuron. Training is performed using a combined loss of binary cross-entropy for classification and Smooth L1 loss for regression, with the latter computed only on samples with  $q_c = 0$  (i.e., not completely caged). In the manipulation planner (Section V), we define a unified robustness metric  $\hat{Q}_{\text{mee}}$  as:

$$\hat{Q}_{\text{mee}} = \mathbb{1}[\hat{q}_c \geq q_{\text{thres}}] \cdot Q_{\text{max}} + \mathbb{1}[\hat{q}_c < q_{\text{thres}}] \cdot \hat{Q}_{\text{mee}}, \quad (4)$$

where  $\mathbb{1}[\cdot]$  denotes the indicator function,  $q_{\text{thres}} \in [0, 1]$  is a cage status confidence threshold, and  $Q_{\text{max}} > 0$  is a predefined high robustness value.

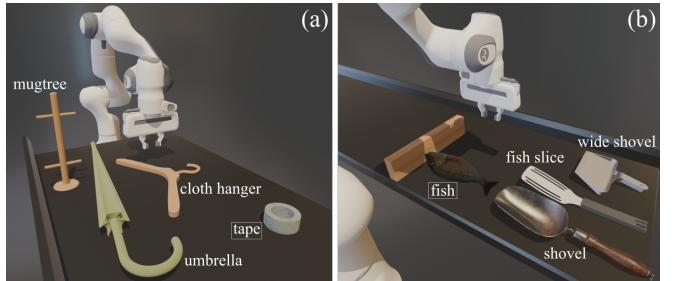
For the PCC baseline robustness metric, we employ the same network structure, which outputs partial caging status  $\hat{q}_{\text{pc}}$  and clearance  $\hat{Q}_{\text{pcc}}$ . These predictions are integrated in the planner through a unified partial caging clearance metric  $\hat{Q}_{\text{pcc}}$ :

$$\hat{Q}_{\text{pcc}} = \mathbb{1}[\hat{q}_{\text{pc}} \geq q'_{\text{thres}}] \cdot (Q'_{\text{max}} + \hat{Q}_{\text{pcc}}), \quad (5)$$

where  $q'_{\text{thres}} \in [0, 1]$  is the partial caging confidence threshold and  $Q'_{\text{max}} > 0$  is a predefined clearance offset constant.

## VII. EXPERIMENTS

We evaluate our approach on three representative tool-use tasks—tape pulling, fish scooping, and scissors hanging—each involving selection from a set of  $N=3$  candidate tools. These tasks span rigid, articulated and deformable objects, covering both non-prehensile and prehensile manipulation scenarios. Our method consistently selects tools and robust tool-object configurations, outperforming the baseline under force disturbances. Experiments on the scissors hanging task validate the real-world practicality of our approach.



**Fig. 4:** Our planner identifies the most robust tool from a cluttered table and plans a trajectory to pull a tape (a) or scoop a fish (b).

### A. Task Description

**Tape Pulling:** A Franka Panda robot arm pulls a Tape Roll ( $s_{\text{obj}} \in \text{SE}(2)$ ) from outside its workspace into a reachable goal region using one of the candidate tools—*Cloth Hanger*, *Umbrella*, and *Mugtree*, shown in Fig. 4a.

**Fish Scooping:** A deformable Fish ( $s_{\text{obj}} \in \text{SE}(3) \times \mathbb{R}^9$ ), modeled as 10 links connected by 9 compliant revolute joints, is scooped from a surface near a fixed block. Available tools are a *Fish Slice*, *Wide Shovel*, and *Shovel* (Fig. 4b).

**Scissors Hanging:** A robot aims to hang a pair of Scissors ( $s_{\text{obj}} \in \text{SE}(3) \times \mathbb{R}$ ) with an articulated joint onto a hook. Candidate hooks are a *Coat Hook*, *Treble Hook*, and *Slatwall Hook* (Fig. 7).

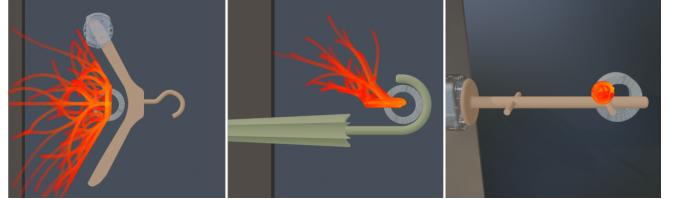
### B. Robustness-Aware Tool Selection

#### 1) Impact of tool geometry on manipulation robustness:

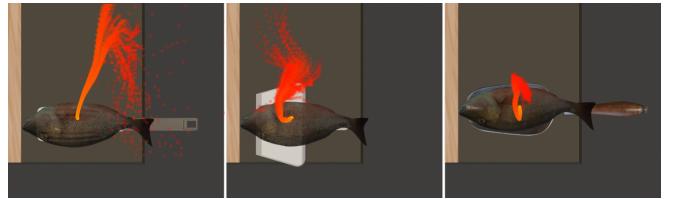
We first evaluate whether our learned robustness metric  $\dot{Q}_{\text{mee}}$  leads to meaningful tool choices by examining how different tool geometries affect manipulation robustness. For each tool  $o_{\text{tool}}$ , we optimize the tool-object configuration  $(s_{\text{tool}}, s_{\text{obj}})$  to maximize  $\dot{Q}_{\text{mee}}$  using CMA-ES (20 iterations, 100 candidates per iteration). Fig. 5 visualizes the disturbed object positions for each tool, offering a direct comparison of robustness across tools.

In the tape pulling and fish scooping tasks, tools with stronger caging affordances consistently lead to better robustness to disturbances. For the tape pulling task, the *Mugtree* offers the best robustness by caging the Tape Roll within its central loop. The *Umbrella* partially surrounds the tape using its handle, while the *Cloth Hanger* fails to constrain the tape, resulting in frequent displacements. For the fish scooping task, the *Shovel* secures the Fish more effectively than the *Fish Slice* and *Wide Shovel*, leading to fewer deviations in response to disturbances. These results demonstrate that our robustness metric reliably identifies tool-object configurations that maximize the object’s robustness against external forces. They also highlight the importance of the escape energy metric in achieving robust non-prehensile manipulation.

**2) Tool selection under constraints:** Robotic tool use is subject to constraints like kinematic reachability,  $c(s_{\text{tool}}, r)$  in Eq. (2). We consider these constraints during keyframe optimization, assuming fixed grasp poses and workspace limits. Under these constraints, the robot selects: (i) the *Shovel* for robust fish scooping, and (ii) the *Umbrella* for tape pulling. Although the *Mugtree* yields the highest robustness, it is not selected since the robot cannot reach the tape using the *Mugtree*. The resulting *Umbrella-Tape Roll* configuration (Fig. 5a, center) is planned as the best tradeoff between robustness and feasibility. Therefore, robustness-aware keyframe optimization enables the planner to select not just feasible tools, but ones that are geometrically optimized to tolerate disturbances. It validates the effectiveness of our learned metric in guiding robustness-aware tool use.



(a) tape pulling



(b) fish scooping

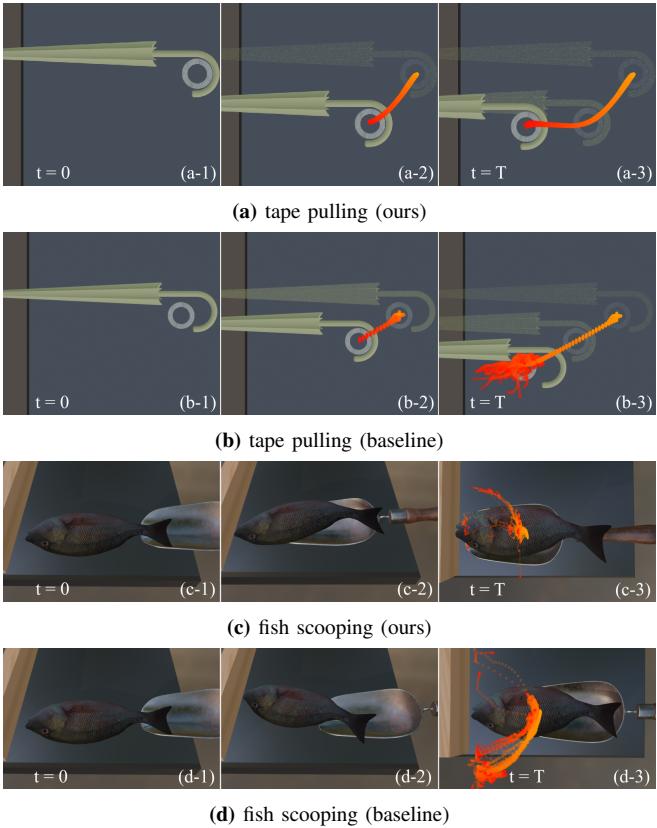
**Fig. 5:** Robustness of different tools under uncertainties, evaluated in their best-found configurations  $s_{\text{obj}}^*, s_{\text{tool}}^*$ . 100 episodes of object trajectories under random force disturbances are simulated and visualized in red. From left to right: tools with increasing manipulation robustness to external disturbances.

### C. Robustness-Aware Tool-Use Manipulation Planning

We next examine whether our learned robustness metric  $\dot{Q}_{\text{mee}}$  improves manipulation trajectories compared to the baseline (PCC) metric  $\dot{Q}_{\text{pcc}}$ . Specifically, we focus on how the choice of robustness metric impacts the quality of both the selected keyframe configurations  $(s_{\text{tool}}^*, s_{\text{obj}}^*)$  and subsequent motion trajectories  $\tau_{\text{tool}}^*$ . Our results demonstrate that the energy-informed robustness metric better guides robust tool-use planning, highlighting the importance of incorporating it into the framework for robustness-aware manipulation.

**1) Tape pulling:** Both our method and the baseline PCC metric take the *Umbrella* as the best-found tool  $o_{\text{tool}}^*$  under reachability constraints. However, our robustness-aware method, guided by the  $\dot{Q}_{\text{mee}}$  metric, produces a trajectory  $\tau_{\text{tool}}^*$  that maintains close contact with the Tape Roll, ensuring its high escape energy throughout the entire motion (Fig. 6a). This configuration significantly enhances resistance to external disturbances. In contrast, the PCC-based planner fails to plan a robust initial keyframe configuration, as no valid partial caging configuration exists for the selected tool-object pair (Fig. 6b). Consequently, the baseline trajectory starts from a suboptimal, less robust configuration. The umbrella initially pulls the tape using its handle but prematurely loses contact. It relies solely on the tape’s momentum to reach the goal, which makes it more prone to external disturbances. This difference is evident when external force disturbances are applied: our method yields highly consistent tape trajectories, whereas the baseline method results in substantial deviations, especially at the beginning and the end of the motion.

**2) Fish scooping:** Our approach with  $\dot{Q}_{\text{mee}}$  identifies the *Shovel* as the best-found tool, selecting a configuration near the wall as the most robust keyframe  $(s_{\text{tool}}^*, s_{\text{obj}}^*)$  (see Fig. 6c-3). In contrast, the baseline metric  $\dot{Q}_{\text{pcc}}$  can not provide

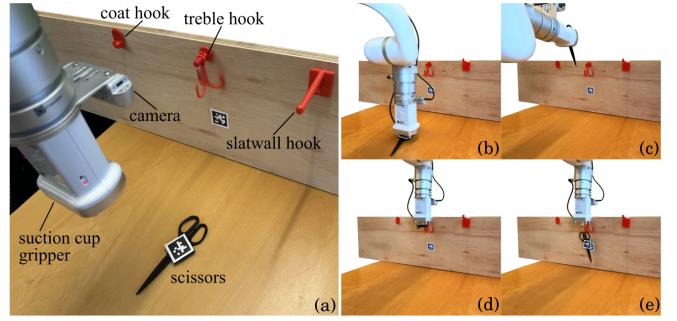


**Fig. 6:** Manipulation trajectories planned with our robustness-aware approach (escape energy metric  $\dot{Q}_{\text{mee}}$ ) compared to the baseline approach (clearance metric  $\dot{Q}_{\text{pcc}}$ ) for the tape pulling and fish scooping tasks. Snapshots show initial ( $t=0$ , left), intermediate (middle), and final ( $t=T$ , right) configurations. Under external disturbances, the intermediate and final tape/fish configurations planned with our approach clearly outperform the baseline in disturbance resistance (red trajectories).

meaningful guidance, as no partial caging configurations exist between the fish and available tools. Consequently, the planner with the PCC metric results in an arbitrary selection of tool and keyframe configurations. To fairly evaluate trajectory quality, we employ the *Shovel* as the tool for both metrics, using their respective optimized keyframe configurations.

Our robustness-aware planner generates a robust and effective scooping trajectory  $\tau_{\text{obj}}^*$ . Specifically, the robot first scoops a substantial portion of the fish, reorients it to achieve better alignment, and finally positions the fish deeper into the shovel. It results in high escape energy barriers and robust configurations. Under random force disturbances applied during open-loop execution, our method reliably secures the fish in the shovel in 92 out of 100 trials (Fig. 6c-3). In comparison, the baseline frequently loses control of the fish under disturbances (Fig. 6d-3), due to the lack of robustness guidance of the metric  $\dot{Q}_{\text{pcc}}$ . The results from the two tasks demonstrate the enhanced manipulation robustness by incorporating the escape energy metric into the planner.

3) *Implementation details:* We employ VP-STO for full-horizon trajectory optimization using 5 via-points, 100 can-



**Fig. 7:** Real-world experiment setup of the scissors hanging task (a) with a successful manipulation trajectory (b-e). The robot picks up the scissors on the table and places them on the selected *Treble Hook* in the most robust way.

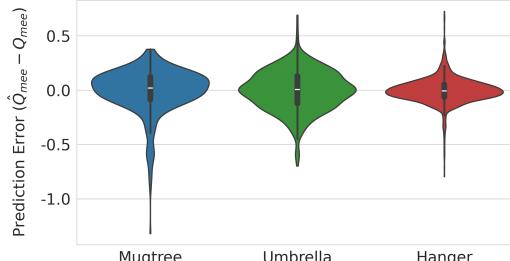
dicates per iteration, and 50 iterations across both tasks. In the tape pulling task, the motion is divided into two phases: (i) reaching the keyframe configuration ( $s_{\text{tool}}^*, s_{\text{obj}}^*$ ) while keeping the tape stationary, followed by (ii) executing a robustness-aware manipulation trajectory  $\tau_{\text{tool}}^*$  guided by the learned metrics. In the fish scooping task, the *Shovel* is initially positioned under the fish’s tail. The robot then uses it to scoop the fish, guiding the motion toward the keyframe configuration. During evaluation, random external disturbances are applied to the objects  $o_{\text{obj}}$  with magnitudes matching their gravity.

#### D. Scissors Hanging: From Simulation to Real World

We evaluate the practical applicability of our framework in the scissors hanging task. We first determine the best hook  $o_{\text{tool}}^*$  and keyframe configuration  $s_{\text{obj}}^*$  by maximizing the learned robustness metric (Eq. (2)). We set the keyframe timestep  $T_k = T$  and optimize the hanging motion before reaching a pre-release configuration. We apply random disturbances to the scissors in their best-found configurations with respect to each hook. The scissors typically fell from the *Coat Hook* at around 18 N, the *Slatwall Hook* at 30 N, and remained stable on the *Treble Hook* up to 40 N. These results confirm the *Treble Hook* as the best choice.

The real-world experimental setup closely replicates the simulation environment, utilizing a UFactory xArm 7-DoF robotic arm equipped with a suction-cup gripper and an in-hand camera. In each trial, the scissors are randomly placed on the workspace, after which the robot estimates their pose and grasps them by suction. The robot then executes a pre-computed Cartesian trajectory, transporting the scissors to the designated pre-release pose above the best-found tool  $o_{\text{tool}}^*$ , i.e. *Treble Hook*. The suction is subsequently disengaged, allowing the scissors to drop onto the hook.

Several practical uncertainties impact the experiment outcomes, including discrepancies between the simulation and real-world pose and geometry (e.g., inaccuracies in 3D-printed components), imperfect pose estimations of the scissors from the perception module, and vibrations from the experimental platform. Despite these uncertainties, our method achieves successful scissors placement onto the selected *Treble Hook* in 6 out of 8 trials, outperforming the



**Fig. 8:** Prediction error of the learned robustness guidance across three object-tool pairs in the tape pulling task. The violin plot shows the distribution of absolute errors between predicted  $\hat{Q}_{\text{mee}}$  and BIT\*-computed  $Q_{\text{mee}}$ . White dashes: medians; black bars: interquartile ranges (IQR); whiskers: extrema within 1.5 IQR.

baseline approach, which succeeded in 4 out of 8 trials. This improvement highlights the benefit of explicitly optimizing for robustness by maximizing escape energy during keyframe selection. Failure cases reveal that the primary sources of unsuccessful attempts were possibly errors in pose estimation and slight physical misalignments. The baseline method is more prone to uncertainties, since the partial caging metrics fail to effectively characterize manipulation robustness in the presence of physical contacts and gravitational potential energy. Overall, these results demonstrate the practical utility of our proposed robustness metric and its effectiveness in addressing real-world manipulation uncertainties.

### VIII. DISCUSSION

Our approach relies on a neural network to predict the robustness of tool-object configurations, significantly improving computational efficiency. Directly evaluating each configuration requires approximately 3 to 5 seconds using BIT\*, making intensive online planner queries impractical. The trained model offers an efficient alternative and achieves good regression performance, as shown in Fig. 8. However, the robustness evaluation quality is inherently bounded by the limitations of BIT\*, which only provides sub-optimal escape paths given finite time, resulting in an upper-bound estimate of robustness.

While our framework incorporates random external forces as a proxy for unmodeled dynamics, this is a simplified model of real-world uncertainty. Future extensions could consider a broader range of uncertainty sources, such as perception noise, object pose uncertainty, contact model errors, and actuation imprecision. Furthermore, in simulation and real-world experiments, we assume fixed pregrasp points or suction positions on the tools. This simplification avoids the complexity of grasp planning but may constrain robustness in practice.

Finally, the current data generation pipeline requires rebuilding a BIT\*-based tree for each new configuration, which significantly limits scalability. Future work could explore reusing and adapting existing tree structures across similar configurations to accelerate robustness estimation and support larger-scale training datasets.

### IX. CONCLUSION

Tool geometry significantly influences robustness in object manipulation. Building on this observation, we proposed a hierarchical optimization framework that jointly selects the most robust tool and plans its corresponding manipulation trajectory. By integrating a learned energy-informed robustness metric into the planning loop, our method consistently improved performance across three representative tasks, enabling robust, contact-rich manipulation under uncertainty.

### REFERENCES

- [1] C. Wang, X. Zhang, X. Zang, Y. Liu, G. Ding, W. Yin, and J. Zhao, “Feature sensing and robotic grasping of objects with uncertain information: A review,” *Sensors*, vol. 20, no. 13, p. 3707, 2020.
- [2] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, “Surprisingly robust in-hand manipulation: An empirical study,” *Proc. Rob. Sci. Syst.*, 2017.
- [3] M. T. Mason, “Toward robotic manipulation,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 1–28, 2018.
- [4] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—a survey,” *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, 2013.
- [5] A. Daniels, S. Kerz, S. Bari, V. Gabler, and D. Wollherr, “Grasping in uncertain environments: A case study for industrial robotic recycling,” in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2023, pp. 3514–3521.
- [6] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, “Learning dexterous in-hand manipulation,” *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.
- [7] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, “Planning for robust open-loop pushing: Exploiting quasi-static belief dynamics and contact-informed optimization,” *Int. J. Robot. Res.*, 2024.
- [8] N. S. Pollard, “Synthesizing grasps from generalized prototypes,” in *Proc. Int. Conf. Robot. Automat.*, vol. 3. IEEE, 1996, pp. 2124–2130.
- [9] M. A. Roa and R. Suárez, “Grasp quality measures: review and performance,” *Autonomous robots*, vol. 38, pp. 65–88, 2015.
- [10] A. Saxena, L. L. Wong, and A. Y. Ng, “Learning grasp strategies with partial shape information.” in *AAAI*, vol. 3, no. 2, 2008, pp. 1491–1494.
- [11] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *Proc. Rob. Sci. Syst.*, 2017.
- [12] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, “Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning,” in *Proc. Int. Conf. Robot. Automat.*. IEEE, 2018, pp. 5620–5627.
- [13] M. Mason, “The mechanics of manipulation,” in *Proc. Int. Conf. Robot. Automat.*, vol. 2. IEEE, 1985, pp. 544–548.
- [14] W. Kuperberg, “Problems on polytopes and convex sets,” in *DIMACS Workshop on polytopes*, 1990, pp. 584–589.
- [15] E. Rimon and A. Blake, “Caging planar bodies by one-parameter two-fingered gripping systems,” *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 299–318, 1999.
- [16] A. Rodriguez, M. T. Mason, and S. Ferry, “From caging to grasping,” *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 886–900, 2012.
- [17] J. Mahler, F. T. Pokorny, S. Niyaz, and K. Goldberg, “Synthesis of energy-bounded planar caging grasps using persistent homology,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 908–918, 2018.
- [18] Y. Dong and F. T. Pokorny, “Quasi-static soft fixture analysis of rigid and deformable objects,” in *Proc. Int. Conf. Robot. Automat.*. IEEE, 2024, pp. 6513–6520.
- [19] Y. Dong, X. Cheng, and F. T. Pokorny, “Characterizing manipulation robustness through energy margin and caging analysis,” *IEEE Robot. Automat. Lett.*, 2024.
- [20] A. Shirizly and E. D. Rimon, “Selection of secure gravity based caging grasps of planar objects: Robustness and experimental validation,” *IEEE Trans. Robot.*, 2024.

- [21] G. Wang, K. Ren, A. S. Morgan, and K. Hang, “Caging in time: A framework for robust object manipulation under uncertainties and limited robot perception,” *Int. J. Robot. Res.*, 2025.
- [22] S. Calinon, P. Kormushev, and D. G. Caldwell, “Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning,” *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 369–379, April 2013.
- [23] S. Calinon, T. Alizadeh, and D. G. Caldwell, “On improving the extrapolation capability of task-parameterized movement models,” in *Proc. Int. Conf. Intell. Robot. Syst.*, Tokyo, Japan, November 2013, pp. 610–616.
- [24] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, “Robocook: Long-horizon elasto-plastic object manipulation with diverse tools,” in *Proc. Conf. Robot Learn.* PMLR, 2023, pp. 642–660.
- [25] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh, “Learning bimanual scooping policies for food acquisition,” in *Proc. Conf. Robot Learn.* PMLR, 2023, pp. 1510–1519.
- [26] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held, “Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds,” in *Proc. Conf. Robot Learn.* PMLR, 2023, pp. 1038–1049.
- [27] Y. Liu and A. Billard, “Tube acceleration: robust dexterous throwing against release uncertainty,” *IEEE Trans. Robot.*, 2024.
- [28] E. Heiden, M. Macklin, Y. Narang, D. Fox, A. Garg, and F. Ramos, “Disect: A differentiable simulation engine for autonomous robotic cutting,” *Robotics: Science and Systems XVII*, 2021.
- [29] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell, “Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems,” in *Proc. Int. Conf. Intell. Robot. Syst.*, Hamburg, Germany, Sept.-Oct. 2015, pp. 464–470.
- [30] B. Ti, Y. Gao, J. Zhao, and S. Calinon, “An optimal control formulation of tool affordance applied to impact tasks,” *IEEE Trans. Robot.*, vol. 40, pp. 1966–1982, 2024.
- [31] C. He, R. Nguyen, H. Mayer, L. Cheng, P. Kang, D. A. Aubeeluck, G. Thiong, E. Fredin, J. Drake, T. Looi, *et al.*, “Magnetically actuated dexterous tools for minimally invasive operation inside the brain,” *Science Robotics*, vol. 10, no. 100, p. eadk4249, 2025.
- [32] M. Qin, J. Brawer, and B. Scassellati, “Robot tool use: A survey,” *Frontiers in Robotics and AI*, vol. 9, p. 1009488, 2023.
- [33] M. Suomalainen, Y. Karayiannidis, and V. Kyriki, “A survey of robot manipulation in contact,” *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022.
- [34] M. Levihn and M. Stilman, “Using environment objects as tools: Unconventional door opening,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2502–2508.
- [35] M. Xu, P. Huang, W. Yu, S. Liu, X. Zhang, Y. Niu, T. Zhang, F. Xia, J. Tan, and D. Zhao, “Creative robot tool use with large language models,” *arXiv preprint arXiv:2310.13065*, 2023.
- [36] M. Li, R. Antonova, D. Sadigh, and J. Bohg, “Learning tool morphology for contact-rich manipulation tasks with differentiable simulation,” in *Proc. Int. Conf. Robot. Automat.* IEEE, 2023, pp. 1859–1865.
- [37] Z. Liu, S. Tian, M. Guo, K. Liu, and J. Wu, “Learning to design and use tools for robotic manipulation,” in *Proc. Conf. Robot Learn.*, 2023.
- [38] R. Liu, J. Liang, S. Sudhakar, H. Ha, C. Chi, S. Song, and C. Vondrick, “Paperbot: Learning to design real-world tools using paper,” *arXiv preprint arXiv:2403.09566*, 2024.
- [39] Y. Dong, S. Han, X. Cheng, W. Friedl, R. I. C. Muchacho, M. A. Roa, J. Tumova, and F. T. Pokorny, “Co-designing tools and control policies for robust manipulation,” *arXiv preprint arXiv:2409.11113*, 2024.
- [40] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Batch informed trees (bit\*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,” in *Proc. Int. Conf. Robot. Automat.* IEEE, 2015, pp. 3067–3074.
- [41] M. C. Welle, A. Varava, J. Mahler, K. Goldberg, D. Kragic, and F. T. Pokorny, “Partial caging: a clearance-based definition, datasets, and deep learning,” *Autonomous Robots*, vol. 45, pp. 647–664, 2021.
- [42] N. Hansen, “The cma evolution strategy: A tutorial,” *arXiv preprint arXiv:1604.00772*, 2016.
- [43] J. Jankowski, L. Brudermüller, N. Hawes, and S. Calinon, “VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior,” in *Proc. Int. Conf. Robot. Automat.*, 2023, pp. 10125–10131.