

# 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, for example, choosing a ladle over a flat spatula to serve meatballs. However, robustness under external disturbances remains underexplored in robotic tool-use planning. This paper presents a robustness-aware method that jointly selects tools and plans contact-rich manipulation trajectories, explicitly optimizing for robustness against disturbances. At the core of our method is an energy-based robustness metric that guides the planner toward 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 method across three representative tool-use tasks. Simulation and real-world results demonstrate that our method consistently selects robust tools and generates disturbance-resilient manipulation plans.

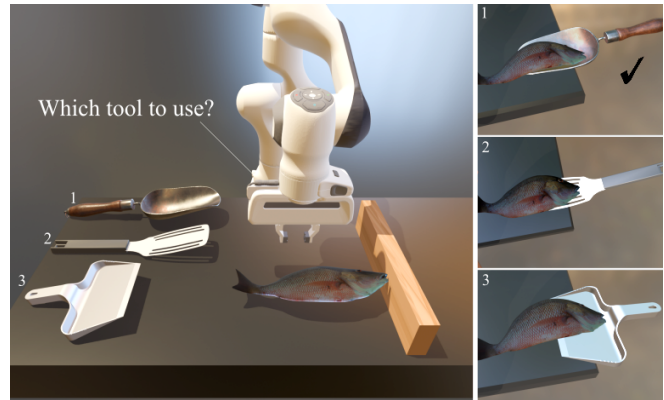
## I. INTRODUCTION

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. It also requires selecting the best tool from a diverse set, while accounting for complex contact dynamics in uncertain, contact-rich environments.

To address these challenges, we propose a unified method that jointly selects the most robust tool and plans a corresponding manipulation trajectory. The problem is formulated as a hierarchical optimization process: (i) selecting a tool and tool-object interaction that is the most robust,

<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. Funded by the European Commission under the Horizon Europe Framework Program project SoftEnable, grant number 101070600, <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.



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

and (ii) planning a manipulation trajectory passing through the selected interaction configuration while maintaining the robustness throughout task execution with the best-found tool. Since robustness evaluation is central but computationally demanding, we introduce an energy-informed metric that provides efficient robustness guidance at both stages of optimization. To further reduce online costs, we train a neural network offline on a dataset of tool-object configurations paired with robustness scores, enabling fast inference during planning. We evaluate our method on three representative tasks: tape pulling, scissors hooking, and fish scooping. The tasks encompass rigid, articulated, and deformable objects. Experimental results demonstrate that our method effectively selects a tool and manipulation trajectory robust against random force disturbances.

In summary, our contributions include: (i) a robustness-aware optimization method that jointly selects tools and plans contact-rich manipulation trajectories, prioritizing tools and configurations robust to disturbances, (ii) a learned energy-informed robustness metric derived from caging analysis, which enables 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 from noisy sensing, unmodeled dynamics, or unpredictable

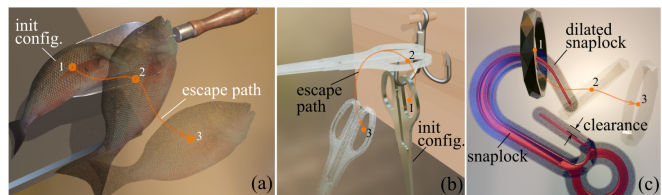
environmental interaction [1], [2]. Mason highlights the importance of robustness in tasks such as crank turning and peg-in-hole insertion, which succeed only if the system tolerates errors in object pose or robot motion [3]. Common sources of uncertainty include geometric ambiguity due to limited perception [4], [5] and dynamic uncertainty from contact interactions [6], [7]. Traditional methods 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 method 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 method that explicitly evaluates and optimizes for manipulation robustness during tool selection and motion planning. Additionally, our research is closely related to



**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 keying in a snaplock (c) can also be characterized by clearance, i.e., the minimal dilation needed to completely prevent escape.

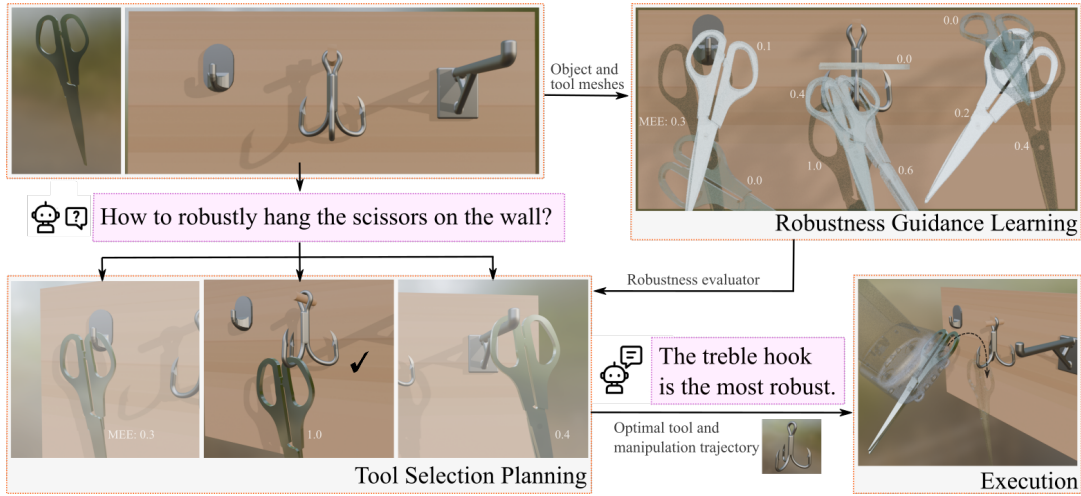
recent studies on the co-design of robot behavior and tool morphology [36]–[39]. Unlike tool design works that can tune or generate geometry to maximize performance, we operate under stricter constraints—selecting the best tool and usage strategy from a fixed set of available tools.

### III. PRELIMINARY

Here, we introduce an energy-informed metric derived from caging analysis, which guides robustness-aware manipulation planning [18]. Consider a rigid object’s configuration space ( $\mathcal{C}$ -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\* [40]. For a formal definition of  $Q_{\text{mee}}$  and details on how to estimate it using BIT\*, we refer interested readers to [18].

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  $E(s_{\text{obj}}) = m_{\text{obj}}gz_{\text{obj}}$ , 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:  $E(s_{\text{obj}}) = m_{\text{obj}}gz_{\text{obj}} + F_p \hat{v} \cdot (x_{\text{obj}}, y_{\text{obj}})$ , 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:  $E(s_{\text{obj}}) = m_{\text{obj}}gz_{\text{obj}} + \frac{1}{2}k\alpha_{\text{obj}}^2$ , 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



**Fig. 3:** Overview of the robustness-aware tool selection and manipulation planning method. The pipeline first performs keyframe optimization to jointly select the most robust tool and tool–object configuration. The best-found configuration then serves as a waypoint for full trajectory optimization, which plans a robustness-preserving manipulation strategy. A neural network trained offline on simulated robustness scores enables efficient robustness evaluation throughout the planning process.

caging tools. PCC  $Q_{\text{pcc}}$  is defined as the minimal tool’s 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  $s_{\text{obj}}^G$ :

$$\min_{o_{\text{tool}}, \tau_{\text{tool}}} \|s_{\text{obj}}^G - s_{\text{obj}}^T\|_2 - \beta \sum_{t=0}^T Q(s_{\text{obj}}^t, o_{\text{tool}}, s_{\text{tool}}^t), \quad (1a)$$

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

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

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  $o_{\text{tool}}$ .  $\beta$  is a tuning parameter that adjusts the balance between the terms of manipulation planning and robustness. In the following, we present how we solve the optimization problem in Section V. Section VI details our

data-driven method to learning the robustness metrics  $Q$ , essential for robustness-aware guidance and computational efficiency of the optimization process. A method overview is illustrated in Fig. 3.

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

The problem of robustness-aware tool use is inherently hybrid, involving discrete decisions over tool choice and continuous optimization of manipulation trajectories. To make this mixed-integer optimization tractable, we adopt a hierarchical method 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:

$$\min_{\tau_{\text{tool}}} \|s_{\text{obj}}^G - s_{\text{obj}}^T\|_2 + \|s_{\text{tool}}^* - s_{\text{tool}}^{T_k}\|_2 + \|s_{\text{obj}}^* - s_{\text{obj}}^{T_k}\|_2 - \beta' \sum_{t=0}^T Q(s_{\text{obj}}^t, o_{\text{tool}}^*, s_{\text{tool}}^t), \quad (3a)$$

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

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

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 simple data-driven method 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 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 ( $Q_{\text{mee}} = \text{inf}$ ), 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. Eq. 4 ensures assigning high robustness  $Q_{\text{max}}$  only when the object is confidently caged (above threshold). Otherwise, it is considered an energy-bounded caging configuration, relying on the learned continuous metric  $\hat{Q}_{\text{mee}}$ .

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 method 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. Experiments on the scissors hanging task validate the real-world practicality of our method.

### A. Task and Method Description

We evaluate our method on three representative tool-use tasks (Fig. 4, 7): (i) **Tape Pulling**: A Franka Panda robot pulls a Tape Roll ( $s_{\text{obj}} \in SE(2)$ ) into a reachable region using one of three tools—Cloth Hanger, Umbrella, or Mugtree. (ii) **Fish Scooping**: A deformable Fish ( $s_{\text{obj}} \in SE(3) \times \mathbb{R}^9$ ), modeled as 10 links connected by 9 compliant

TABLE I: VLM confidence scores (mean±std) for tool selection.

Task	Tool 1	Tool 2	Tool 3
Tape pulling	0.13±0.03 (mugtree)	0.41±0.15 (cloth hanger)	<b>0.46±0.17</b> (umbrella)
Fish scooping	0.22±0.06 (wide shovel)	0.12±0.04 (fish slice)	<b>0.66±0.03</b> (shovel)
Scissors hanging	0.18±0.07 (coat hook)	<b>0.64±0.01</b> (treble hook)	0.18±0.06 (slatwall hook)

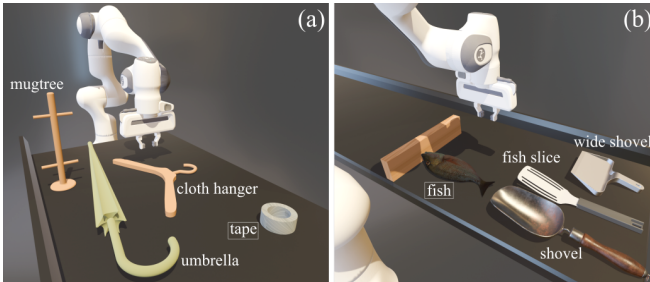


Fig. 4: Our planner identifies the most robust tool from a cluttered table and plans an optimal trajectory to robustly pull a tape (a) or scoop a fish (b).

revolute joints, is scooped from a surface near a fixed block using a Fish Slice, Wide Shovel, or Shovel. (iii) **Scissors Hanging:** A pair of articulated Scissors ( $s_{\text{obj}} \in SE(3) \times \mathbb{R}$ ) is hung onto a Coat Hook, Treble Hook, or Slatwall Hook.

We compare four methods: MEE (our proposed learned escape-energy metric), PCC (clearance-based baseline), VLM (Vision–Language Model), and NoRob (planning without any robustness guidance,  $\beta = 0$ ).

1) *VLM-based baseline:* We choose VLMs as a baseline for tool selection because pre-trained foundation models have shown strong capabilities in scene understanding and decision-making for robotic manipulation tasks. Recent works demonstrate that VLMs can generate robotic tool designs [44]. Our work extends the idea to tool selection using VLMs. We use GPT-5 across the three tasks. For each task, the input image (Fig. 4a, 4b, 7a) includes geometry, size, and pose information of three candidate tools, the target object, and the robot arm. The input text contains the task description, robustness definition, and other contextual details. We instruct the VLMs to reason step-by-step and output: the chosen tool, confidence scores for each tool, the reasoning process, and a sketch of the low-level action plan using the selected tool. More details can be found in the [project website](https://sites.google.com/view/robust-tool-use/)<sup>1</sup>.

## B. Robustness-Aware Tool Selection

1) *How tool geometry affects robustness:* We evaluate whether the learned robustness metric  $\hat{Q}_{\text{mee}}$  produces meaningful tool–object configurations by analyzing their resistance to disturbances. For each tool, the tool–object configuration is optimized using CMA-ES (20 iterations, 100 candidates/iteration) with MEE, and the resulting disturbed object positions are visualized in Fig. 5. Across all tasks, tools with stronger caging affordances yield higher robustness: in Tape Pulling, Mugtree fully secures Tape Roll by enclosing it in its core, Umbrella provides partial

enclosure, and Cloth Hanger offers minimal constraint; in Fish Scooping, Shovel holds Fish more securely than Fish Slice or Wide Shovel; in Scissors Hanging, disturbance tests show scissors typically fall from Coat Hook at around 18 N, from Slatwall Hook at 30 N, and remain on Treble Hook up to 40 N, confirming Treble Hook as the most robust choice. These results demonstrate that tools offer varying levels of robustness and that robustness guidance enables the planner to reliably identify configurations that maximize disturbance resistance for a given tool.

2) *Tool selection across methods:* We compare tool selection outcomes from MEE, PCC, NoRob, and VLM under robot kinematic constraints. Results for MEE, PCC, and NoRob are averaged over 20 keyframe optimization runs, while VLM results (Table I) are averaged over 5 runs.

In Tape Pulling, all methods select Umbrella as it is the only tool satisfying the reachability constraint, although Fig. 5 shows that Mugtree can provide the most robust configuration by fully caging Tape Roll through its core. Compared to PCC and NoRob, MEE produces more robust Umbrella–Tape Roll keyframe configurations. VLM fails to identify the most robust configuration for Umbrella or Mugtree, reflecting limited capability to repurpose everyday objects for novel tool-use strategies, despite showing some zero-shot ability to infer simple geometric properties.

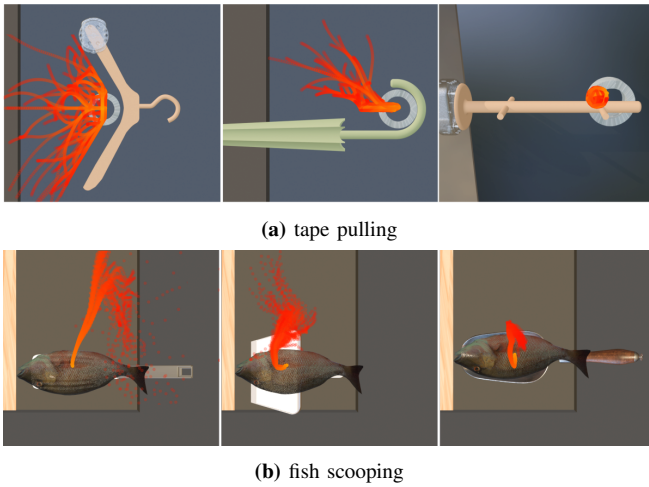
In Fish Scooping, MEE and VLM consistently select Shovel, which, as confirmed earlier, provides the highest disturbance resistance. In contrast, PCC and NoRob vary in their selections and often choose less robust tools and suboptimal keyframe configurations.

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

We assess the effectiveness of our method MEE for robustness-aware manipulation planning by comparing it with the baselines PCC and NoRob. The evaluation measures positional deviations of the manipulated objects under random force disturbances in open-loop execution. Results are summarized in Fig. 6 and Table II.

Fig. 6 shows that plans generated with MEE exhibit greater robustness against disturbances than those from the baselines. In Tape Pulling, all methods select Umbrella as the optimal tool due to kinematic constraints. However, MEE produces keyframe configurations and trajectories in which the Umbrella’s handle maintains close, consistent contact with Tape Roll (Fig. 6a). This results in higher escape energy along the trajectory, making Tape Roll more resistant to external forces. In contrast, PCC fails to find any partial caging configurations between Umbrella and Tape Roll. As a result, its trajectories start from arbitrary collision-free pulling configurations. Lacking robustness guidance, the robot pulls Tape Roll with Umbrella using excessive force,

<sup>1</sup><https://sites.google.com/view/robust-tool-use/>



**Fig. 5:** Robustness of different tools under disturbances, evaluated in their best-found configurations ( $s_{obj}^*$ ,  $s_{tool}^*$ ) with MEE. For each tool, 100 simulated episodes of object trajectories under random force disturbances are shown in red. Tools are ordered from left to right by increasing manipulation robustness to external disturbances.

causing it to slide out of the handle (Fig. 6b) and making the system less robust to external disturbances.

In Fish Scooping, MEE enables Shovel to capture a substantial portion of Fish, align with it, and push it deep into Shovel’s belly using the table fixture (Fig. 6c). The resulting keyframe configuration effectively constrains Fish between Shovel and the table fixture. Without effective robustness guidance, PCC fails to achieve this outcome (Fig. 6d), producing configurations that are far more susceptible to external disturbances.

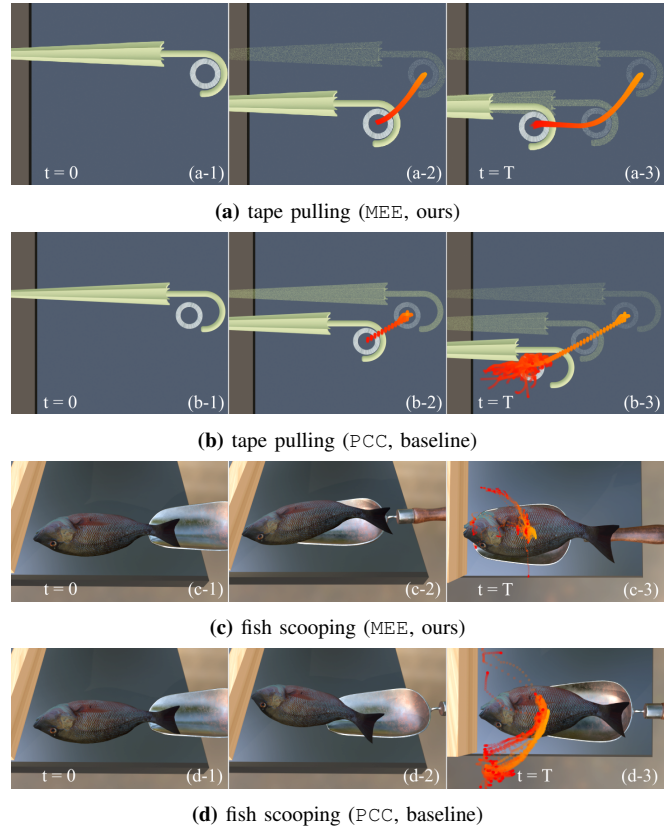
Quantitative results in Table II confirm the performance gap: MEE consistently yields higher robustness than PCC, while NoRob suffers from substantial positional deviations due to the absence of robustness guidance. These findings validate that incorporating robustness metrics into manipulation planning can significantly improve the disturbance resistance of tool-use strategies.

Tasks	MEE (Ours)	PCC	NoRob
Tape Pulling	<b>0.054 ± 0.021</b>	0.237 ± 0.136	0.261 ± 0.136
Fish Scooping	<b>0.190 ± 0.267</b>	0.761 ± 0.649	1.463 ± 0.322

**TABLE II:** Average positional deviations of target object under random force disturbances in open-loop execution of planned manipulation trajectory. Lower values indicate higher robustness.

#### D. Real World Experiments

We evaluate the practical applicability of our method in Scissors Hanging. The real-world experimental setup 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 the planned Cartesian trajectory, transporting the



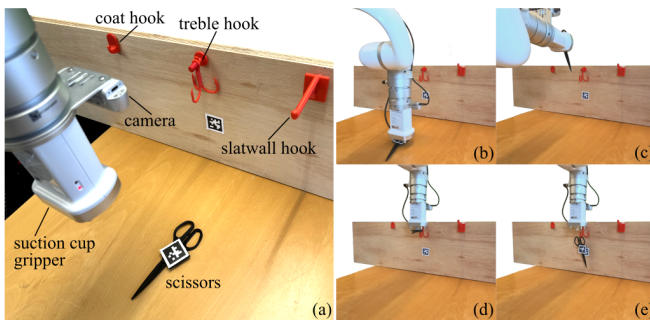
**Fig. 6:** Manipulation trajectories planned with our method MEE compared to the baseline PCC in Tape Pulling and Fish Scooping. Snapshots show initial ( $t=0$ , left), intermediate (middle), and final ( $t=T$ , right) configurations. Under random force disturbances, the intermediate and final Tape Roll/Fish configurations from MEE exhibit greater disturbance resistance than those from PCC (red trajectories).

scissors to the designated pre-release pose above the best-found tool  $o_{tool}^*$ , i.e. Treble Hook. The suction is subsequently disengaged, and Scissors drop onto the hook.

Table III showcases preliminary results of our method’s transferability to the real world. The results indicate the benefit of explicitly optimizing for robustness by maximizing escape energy during keyframe selection. Failure cases reveal that the primary reasons for unsuccessful attempts were possibly errors in pose estimation and slight physical misalignments. More examples of failure, such as misaligned suction positions or collisions between the scissors and the hooks or wall along the executed trajectory, can be found on our [website](#). We also evaluated in the real-world experiments the most promising baseline method PCC from the simulation results (over VLM and NoRob). It 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. The results demonstrate the practical utility of our proposed robustness metric and its potential in addressing real-world manipulation uncertainties.

Method	MEE	PCC
SR	0.83	0.50

**TABLE III:** Success rates in real-world Scissors Hanging executions under uncertainties. 20 runs in total.



**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.

### E. Implementation Details

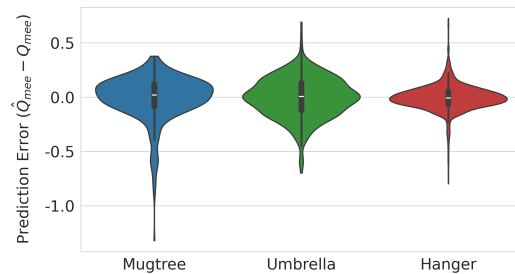
For robustness guidance learning, the number of training samples  $N_s$  per object-tool pair ranges from 1000 to 5000, depending on the dimensionality of the target object’s configuration space and the complexity of its collision geometry. Key hyperparameters include learning rate of  $4 \times 10^{-5}$ , batch size of 32, 200 training epochs, and the Adam optimizer with a weight decay of  $1 \times 10^{-4}$ . All models are trained on an Intel i9-12900H CPU. We publicly release the robustness guidance dataset for our three tasks on our [website](#). It comprises tool-object configurations with corresponding MEE/PCC values, along with tool and object model files.

For manipulation planning, we use VP-STO for full-horizon trajectory optimization with 5 via-points, 100 candidates per iteration, and 50 iterations. In Tape Pulling, planning involves (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 robustness metric. In Fish Scooping, the shovel is first positioned under the fish’s tail, after which the robot scoops the fish toward the keyframe configuration.

During evaluation, random force disturbances are applied at each of the 100 execution steps, with magnitudes matching the object’s weight. In Tape Pulling, disturbances are applied at four equidistant points (top, bottom, left, and right) along the outer circumference of Tape Roll, directed radially outward from the center toward the contact point. In Fish Scooping, similar planar disturbances are applied at the center of mass of Fish in the  $x$ - $y$  plane. Additionally, a randomized gravity-compensation force, ranging from 50% to 75% of the fish’s weight, is introduced to assess the robustness of the generated trajectories.

## VIII. DISCUSSION

We leverage a simple neural network to predict the robustness of tool-object configurations, significantly improving



**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.

computational efficiency. Each configuration requires approximately 3 seconds to evaluate using BIT\*, making direct evaluation during planning prohibitively expensive. Instead, our learned robustness module enables fast forward inference on a CPU with an average time of  $0.057 \pm 0.019$  ms over 1000 trials. To plan a full open-loop trajectory, our hierarchical planner queries the robustness model approximately  $50 \times 100 \times 100 = 5 \times 10^5$  times, resulting in a total inference time of about 29 seconds—orders of magnitude faster than the  $\sim 17$  CPU-days that would be needed without the neural network. This efficiency enables practical real-time planning while maintaining strong regression accuracy, as shown in Fig. 8.

One limitation of our method is that the robustness metric is trained only on the tools and objects present in the evaluated tasks. Future work will extend the dataset to include a broader variety of tools and objects, enabling a network with zero-shot generalization capabilities to novel tools and objects. In particular, a large-scale, data-driven model that maps raw camera inputs directly to robustness values would be especially useful. With such a metric, the proposed planner could be applied more broadly for tool selection and manipulation planning under disturbances. On the other hand, our method currently leverages random force disturbances as a proxy for hard-to-model system dynamic mismatches. A promising direction for future work is to integrate belief-space planning approaches [5], [7] that explicitly account for robustness against system uncertainties.

## IX. CONCLUSION

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

## 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. Bruder Müller, 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. Kyrki, “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, “Cagecoopt: enhancing manipulation robustness through caging-guided morphology and policy co-optimization,” *Proc. Int. Conf. Intell. Robot. Syst.*, pp. 21 795–21 802, 2025.
- [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. Bruder Müller, 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. 10 125–10 131.
- [44] G. J. Gao, T. Li, J. Shi, Y. Li, Z. Zhang, N. Figueroa, and D. Jayaraman, “Vlmgineer: Vision language models as robotic toolsmiths,” *arXiv preprint arXiv:2507.12644*, 2025.