Leveraging Extrinsic Dexterity for Occluded Grasping on Undesirable Walls

Keita Kobashi[1] and Masayoshi Tomizuka[1]

***Abstract*— This study addresses an occluded grasping problem that requires physical interactions between a robot and an environment to grasp an object. A simple parallel gripper has the limitation of its dexterity. For instance, the gripper sometimes cannot grasp a flat box on a table due to the actuation limit of the gripper. However, even with the simple gripper, the robot can grasp the box by manipulating its pose by leveraging extrinsic contact with the environment, such as a vertical wall. Indeed, several previous works have discussed similar problems, such work assumes a short wall for manipulation. This assumption may not always be satisfied. If the wall that makes physical interactions with the robot is too large or too tall, the robot cannot grasp the object even after manipulating the pose. In that case, the robot is required to combine different types of actions. Then, in this work, we consider a hierarchical reinforcement learning framework to tackle this long-horizon manipulation problem. We adopt Q-learning to train the high-level policy, and this policy chooses the type of action that renders us the highest reward. Then, the selected low-level skill samples an actual robot action in a continuous space. During the training phase of the skills, we apply domain randomization so that the skills have generalizability.**

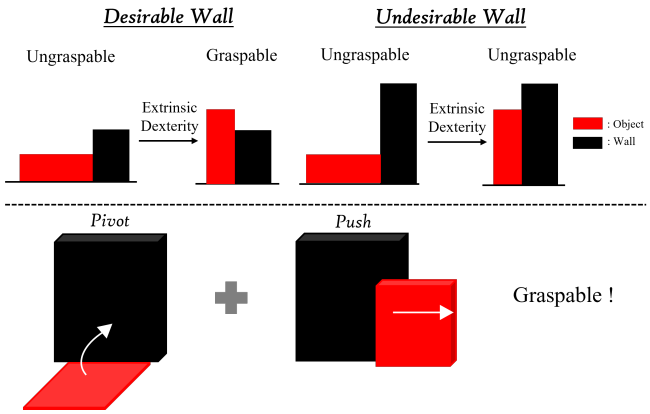

Fig. 1. The problem description. In this work, we address an occluded grasping problem on undesirable walls. Previous works implicitly assume desirable walls, where the robot can grasp an object after pivoting. However, if the wall is too large or too tall, the robot cannot grasp even after pivoting. To deal with this problem, we consider combining pivoting actions and pushing actions to make the object graspable even on undesirable walls.

I. INTRODUCTION

Robotic grasping is one of the fundamental tasks and is important to address. Typically, when the robot picks and places an object in another location or the robot performs further manipulation, such as robotic insertion, the robot needs to grasp the object [1]–[5]. Whether the object can be grasped or not can depend on the object pose. For instance, when grasping a flat box on a table (as shown in Fig. 1), the robot may fail due to the gripper's actuation limits. In this work, we address such an "Occluded grasping" problem [6], [7], which deals with grasping an object whose primary grasp configurations are occluded.

We can consider deploying more dexterous grippers to the robot, such as multi-finger hands [8], [9], to tackle this problem. However, such grippers generally involve complex structures, making sustainable deployment difficult due to difficulty in maintenance and simply its fabrication cost. Then, in this work, we discuss enhancing the dexterity and grasping the object of a simple parallel gripper by leveraging extrinsic contact with an environment.

Although some prior works address similar problems [2], [6], [10], [11], they implicitly assume a *desirable* environment, where the robot can grasp the object after pivoting motion as shown in Fig. 1. In this work, we consider an *undesirable* environment; the grasping configurations are still occluded even after pivoting the object. In such an environment, the robot needs to combine different types of actions in addition to pivoting to grasp the object.

The main difficulties of this problem lie in 1) handling complex physical interactions, 2) switching different types of actions automatically, 3) changing the contact location on the object, and 4) achieving generalized performance across various objects. To address this problem, we propose a hierarchical reinforcement learning framework.

This framework possesses high-level policy and low-level skills. The high-level policy decides which skill to choose and the low-level skills generate actual robot actions. We consider three types of robot actions to deal with the occluded grasping problem, pivoting, pushing, and grasping. Each low-level skill corresponds to these robot actions. Since pivoting involves complex physical interactions, we apply domain randomization during training to improve robustness and generalizability. We also employ Conditional Variational Autoencoder (CVAE) to infer the contact location to successfully execute actions from the low-level skills. We would like to highlight that our framework does not require any human demonstrations.

Finally, we verify the effectiveness of our framework by conducting numerical simulations and physical experiments. The results of numerical simulations indicate that our framework achieves the highest success rate of task completion compared to other baselines. Besides, our framework requires fewer training epochs for training due to the hierarchical

[1]Keita Kobashi and Masayoshi Tomizuka are in Department of Mechanical Engineering, University of California, Berkeley, CA, 94704

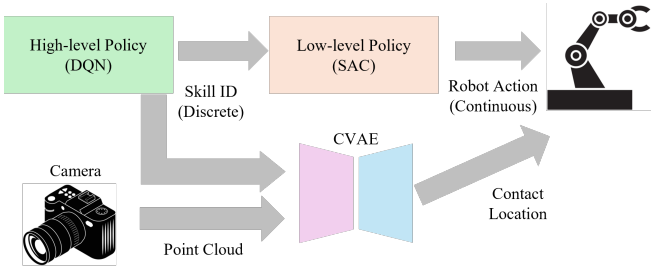

Fig. 2. The overview of our hierarchical reinforcement learning framework for occluded grasping tasks. The high-level policy decides which low-level skill to use based on the object pose and subtask completions. Then, the selected low-level skill generates a robot action. To guide the robot to an appropriate contact location to successfully execute the low-level skill, we adopt CVAE. The contact locations are collected through successful rollouts of the low-level skills and we do not need human demonstrations. The CVAE is conditioned on the skill ID and object point clouds, so that the CVAE can infer the contact location according to the object and the skill. The robot moves to the contact location inferred by the CVAE first, then the robot executes the action from the low-level skill.

structure. For physical experiments, we aim to do zero-shot sim-to-real transfer of the trained high-level and low-level policy. In this paper, we present a preliminary result that a robot executes pivoting, pushing, and grasping actions on a box.

II. PROBLEM FORMULATION

In this section, we address the problem statement we focus in this work, and introduce the fundamentals of reinforcement learning.

A. Problem Statement

In this work, we mainly focus on an occluded grasping problem on undesirable walls and a hierarchical reinforcement framework to decompose such a long-horizon non-prehensile manipulation problem into simple subproblems to easily handle the original problem. We handle an object on a flat surface like a table that is ungraspable at the initial state and a fixed proximity wall perpendicular to the surface. We assume the lateral direction of the wall is aligned with y-axis in the Cartesian coordinate. The undesirable walls are tall, i.e., the object does not become graspable even after the robot pivots the object.

To complete the manipulation task, we aim to learn a policy to manipulate the pose of the object by leveraging extrinsic contact between the wall and the object to make it graspable. The size and position of the wall vary, and our policy automatically manipulates the object by adapting to the different walls. We also aim to consider generalizing the policy to various objects with zero-shot sim-to-real transfer.

B. Fundamentals of Reinforcement Learning

An environment for reinforcement learning is described as a Markov Decision Process (MDP). MDP is a sequential stochastic process and can be modeled as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$ where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P}(s_{t+1}|s_t, a_t)$ denotes the transition probability, which describes the probability from the current state s_t to the next state s_{t+1} with the action a_t, and r denotes the reward. The primary objective of reinforcement learning is to find a policy $\pi(a|s)$ to maximize the expected return $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$ where γ is a discount factor.

III. METHODOLOGY

In this section, we introduce the proposed framework to deal with this occluded grasping problem. The hierarchical reinforcement learning framework comprises a high-level policy to decide which skill to use based on the observation and a selected low-level skill to generate an action for the robot. Before executing robot actions from the low-level skills, we guide the robot to a desired endeffector pose corresponding to the skills by following a linearly interpolated trajectory. To infer the desired endeffector pose for pivoting and pushing, we adopt a CVAE conditioned on object point clouds and the skill ID. Since our main focus is not inferring grasp candidates, we calculate the desired grasping pose by using object sizes. In this work, we use Euler angles as orientations of both objects and the robot. Figure 2 summarizes the overall proposed framework. We further discuss the details of the framework in the following subsections.

A. High-level Policy

The role of the high-level policy is to choose an appropriate skill based on the observation, thus, the action space is discrete. Hence, we employ Deep Q Network (DQN) [12] to train the high-level policy. The action $a_{\text{high}} \in \{0, 1, 2\}$ corresponds to the low-level skill ID where 0 is pivoting, 1 is pushing, and 2 is grasping. The observations are the position and orientation of the object $x_{\text{obj}} \in \mathbb{R}^3$, $o_{\text{obj}} \in \mathbb{R}^3$, the lateral length of the wall $l \in \mathbb{R}$, and subtask (pivoting and pushing) completion flags $v_{\text{pivot}} \in \{0, 1\}$, $v_{\text{push}} \in \{0, 1\}$. For pivoting, the task completion criterion is the same as that in the reward calculation we mentioned later in this section. For pushing, if the distance from the current object position $x_{\text{obj}} \in \mathbb{R}^3$ to the goal position $x_{\text{goal}} \in \mathbb{R}^3$ is less than 0.05 [m], we regard this as completion. We calculate x_{goal} using l so that the object becomes graspable. The high-level policy can adapt to different goal positions by taking l as an observation.

We construct the following reward to train the policy

$$
\begin{aligned}
r &= -\|x_{\text{obj}} - x_{\text{goal}}\|_2 + r_{\text{bonus}} + r_{\text{penalty}} + r_{\text{done}} \\
r_{\text{bonus}} &= \begin{cases} 0.05 & \text{if } d \leq \frac{5\pi}{180} \\ 0 & \text{otherwise} \end{cases} \\
r_{\text{penalty}} &= \begin{cases} -0.02 & \text{if } a_{\text{high}} = 0 \text{ and } d \leq \frac{5\pi}{180} \\ 0 & \text{otherwise} \end{cases} \\
r_{\text{done}} &= \begin{cases} 0.05 & \text{if grasping} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{1}
$$

where $d \in \mathbb{R}$ denotes the distance between the current object rotation matrix $R \in \mathbb{R}^{3\times3}$ and the rotation matrix when the object orientation is perpendicular to the table $R_{\text{perp}} \in \mathbb{R}^{3\times3}$,

$d = \cos^{-1}(\frac{1}{2}(\mathrm{tr}(R_{\mathrm{perp}}R^{\top}) - 1))$. When $d = 0$, the object stands on the table, hence the robot completes the pivoting task. We assign a buffer of 5 [deg] for a pivoting success criterion and use this criterion to feed the bonus r_{bonus} and penalty r_{penalty} to the high-level agent. When the robot succeeds in pivoting an object, the high-level policy receives a bonus reward of 0.05, however, if the high-level agent chooses a pivoting action of 0, we feed a penalty of -0.02 to the high-level agent. Although the actual values of the bonus and the penalty need not to be fixed, we indicate that adding such bonus and penalty terms accelerates the training procedure of the high-level policy. We feed the done reward of 0.05 to the agent when succeeding in grasping the object.

B. Low-level Skill

The lower-level skills generate an actual robot action a_{robot}. In this work, we consider the following three skills: pivoting, pushing, and grasping.

For the pivoting skill, we handle a continuous action space to complete a complex contact-rich manipulation task. We employ Soft Actor-Critic (SAC) [13] and design the following reward to train the pivoting skill.

$$r_{\mathrm{pivot}} = \frac{\pi}{2} - d \tag{2}$$

The observations for this skill are the position and orientation of the endeffector $x_{\mathrm{eef}} \in \mathbb{R}^3$, $o_{\mathrm{eef}} \in \mathbb{R}^3$, and the external contact force $f_{\mathrm{ext}} \in \mathbb{R}^6$. The pivoting skill generates an action of an endeffector velocity in the Cartesian space and we do not consider any rotation of the endeffector during the pivoting task. Thus, the action from the pivoting skill $a_{\mathrm{pivot}} \in \mathbb{R}^3$ should be in a three dimensional space. To acquire the generalizability, we apply domain randomization. We add a zero-mean Gaussian noise $\mathcal{N}(0, 0.2I)$ to f_{ext}, x_{eef}, and o_{eef}.

For pushing and grasping, we utilize hand-crafted skills due to the simplicity of tasks. The pushing skill feeds a constant action of the endeffector velocity $a_{\mathrm{push}} = [0, -0.005, 0]$ to the robot and the robot pushes the object after moving to the contact location inferred by CVAE. The grasping skill feeds a constant action of the endeffector velocity $a_{\mathrm{grasp}} = [0, 0, -0.01]$. When the robot grasps the object, the robot first moves above the grasp location estimated by CVAE, then gradually approaches to the object to grasp with a_{grasp}.

C. Conditional Variational Autoencoder

To estimate the desired contact location for execution of each skill, we adopt CVAE. We feed an object point cloud and a skill ID provided as conditions, and feed desired contact locations as data to train CVAE. The desired contact locations are collected by the successful rollouts of the skills. When we run CVAE online, the point cloud is provided by a depth camera and the skill ID is provided by the high-level agent. When we train CVAE, we add a zero-mean Gaussian noise $\mathcal{N}(0, 0.003I)$ to the object point cloud to simulate the noise and acquire robustness.

IV. EXPERIMENTS

In this section, we conduct numerical simulations and physical experiments to verify the effectiveness of the proposed framework.

A. Numerical Simulation

To begin with, we conduct numerical simulations to verify the effectiveness of our proposed framework. As mentioned in Section III, we consider an environment where there is an object and a fixed wall, and the fixed wall is undesirable. For the manipulated object, we consider different sizes of boxes. To evaluate the performance, we adopt the following baseline methods for comparisons.

- SAC [13]: This is the most basic approach that does not have any skills and contact location estimation modules.
- Proposed w/o CVAE: This approach uses the same architecture of the proposed method without CVAE.
- Proposed w/o skills: This approach uses a SAC-based framework to obtain a continuous action. The robot randomly picks up a contact location from the outputs of the CVAE.
- HACMAN [14]: This method is named Hybrid Actor-Critic Maps for Non-prehensile Manipulation (HACMAN). HACMAN has a discrete-continuous action space to manipulate an object to a desired pose without grasping. In this framework, a point cloud of an object is the observation. HACMAN firstly chooses a contact location on an object by choosing one of the points in a point cloud, then executes the continuous poking action to manipulate the object pose calculated by per-point flow from the current pose to the goal pose. The main difference from ours is HACMAN does not handle manipulation with extrinsic contacts. We examine how this affects the performance for manipulating an object to the graspable (goal) pose.
- HACMAN++ [15]: This approach is an extension of HACMAN, which integrates action primitives into the original framework. We apply the pivoting, pushing, and grasping primitives to this framework for a fair comparison.
- Ungraspable [6]: This approach leverages extrinsic dexterity to grasp an ungraspable object. The extrinsic dexterity is an emergent behavior of the robot (not explicitly designed) and the robot tilts or pivots (not including pushing actions) the object to make it graspable. This work implicitly assumes a desirable environment for manipulation, thus we aim to verify the performance in an undesirable environment.

We train both the high-level policy and the low-level pivoting skill using the implementations from RLkit with a batch size of 256 for the high-level agent and 4096 for the low-level pivoting skill. The CVAE consists of an encoder with two-layer ReLU networks with 512 units, a dropout layer, and a decoder with the same architecture as the encoder. The batch size is 256 and the learning rate is 10^{-4}. We use 3,500 contact locations and object point clouds for

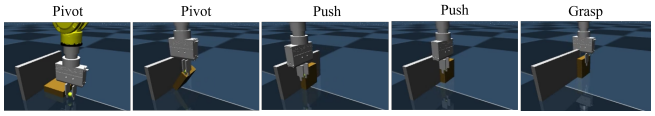

Fig. 3. An example of a successful action sequence of the robot in MuJoCo simulation. The proposed framework can manipulate the object to the graspable pose by combining pivoting, pushing, and grasping actions.

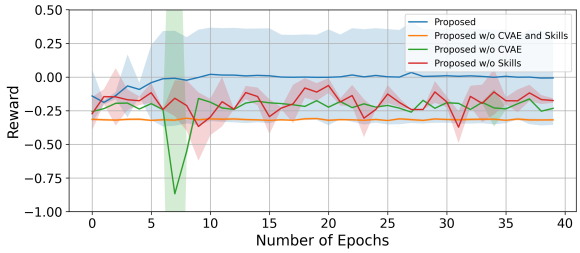

Fig. 4. Reward profile for training of the high-level agent. The proposed framework efficiently learns the appropriate action compared with the other baselines and achieves the highest reward. For HACMAN, HACMAN++, and Ungraspable, the definition of the reward itself is different, and we do not plot in this figure for a fair comparison.

pivoting and pushing to train the CVAE, and run 1,000,000 steps to train.

For simulation settings, we use a $0.05 \times 0.05 \times 0.015$ $[\mathrm{m}^3]$ box for a manipulated object and vary the size of object to examine the generalizability of the proposed framework and baselines. The range is $\mathrm{size}_x \in [0.04\ [\mathrm{m}], 0.06\ [\mathrm{m}]]$, $\mathrm{size}_y \in [0.04\ [\mathrm{m}], 0.06\ [\mathrm{m}]]$. We also vary the wall lateral length to verify whether the proposed framework can adapt to different environments. In this work, we only consider task success rate as an evaluation metric. We allow the high-level policy to choose the skills within 15 trials and if the robot completes the occluded grasping task, we regard this as success. We run the proposed framework and the baselines 10 times. Figure 3 depicts the successful action sequence in the simulation environment. We execute the proposed framework and the baselines on desirable and undesirable walls to demonstrate the effectiveness of the proposed method.

Table. I exhibits the simulation results. Note that for desirable walls, we regard success if the robot pivots the object because the pose becomes graspable. We observe that all baselines demonstrate high success rates on desirable walls. However, on undesirable walls, the performance of all baselines degrades significantly. The reason is that without skills or contact location estimator, finding an appropriate contact location and an effective action to complete the task is difficult. The performance of HACMAN and HACMAN++ also becomes worse. Since these baselines require a perfect

Table I. Success rates on desirable and undesirable walls

	Desirable	Undesirable
SAC	80%	0%
Proposed w/o CVAE	80%	0%
Proposed w/o skills	70%	0%
HACMAN	0%	0%
HACMAN++	70%	0%
Ungraspable	100%	0%
Proposed	100%	100%

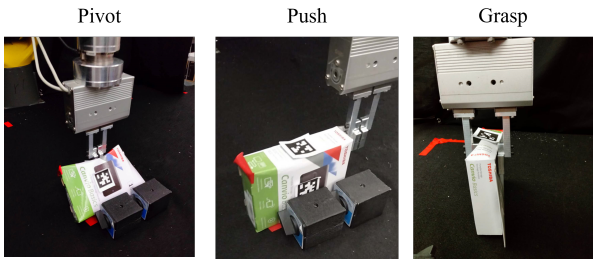

Fig. 5. An example of a manipulation scenario of physical experiments. The proposed framework realizes sim-to-real transfer in pivoting, pushing, and grasping.

object point cloud and the point cloud observation should be updated at each step, the robot cannot execute a long-horizon action. Based on these reasons, these frameworks get worse. When the robot executes the Ungraspable framework, the robot tends to grasp the object after pivoting the object, though the robot motion is emergent behavior and we do not explicitly design the . Hence, when the wall becomes tall, the robot fails to grasp the object. These results demonstrate the effectiveness of our framework, particularly in undesirable environments.

We also analyze the reward profile of the proposed framework and ablations. As shown in Fig. 4, the reward of our framework quickly increases and maintains the highest reward compared to other ablations. For other baselines such as HACMAN, HACMAN++, and Ungraspable, we do not make comparisons since the reward is different.

B. *Physical Experiments*

We also conduct physical experiments in addition to numerical simulations. In these experiments, we aim for sim-to-real zero-shot transfer of the trained policy. We use an RGB-D camera to obtain an object point cloud and apply a color-based filter to segment the point cloud. To realize this, we cover the environment with black curtains. We also use AprilTag [16] to estimate the object pose. We adopt a 6-DoF robotic manipulator with a two-finger parallel gripper that makes it difficult to grasp a large flat object without manipulating its pose.

Figure 5 displays an example of a manipulation scenario that the robot manipulates a box and grasps the box. Although we still use a box object, the shape and the weight of the box are out of distribution. The physical experiments are currently ongoing and this is a preliminary result, however, we plan to apply our framework to different kinds of objects such as bottles for further verifications of generalizability.

V. CONCLUSION

In this study, we present a hierarchical reinforcement learning framework for occluded grasping problems on undesirable walls. The proposed framework performs well particularly in undesirable environments compared with other baselines even though the task comprises complex physical interactions between the environment and the robot.

As future work, we plan to further verify the generalization performance on the real robot.

REFERENCES

[1] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka, "Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks," in *2022 International conference on robotics and automation (ICRA)*. IEEE, 2022, pp. 9881–9887.

[2] X. Zhang, C. Wang, L. Sun, Z. Wu, X. Zhu, and M. Tomizuka, "Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1621–1639.

[3] Y. Fuchioka, C. C. Beltran-Hernandez, H. Nguyen, and M. Hamaya, "Robotic object insertion with a soft wrist through sim-to-real privileged training," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 9159–9166.

[4] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 819–825.

[5] L. Wang, Y. Xiang, and D. Fox, "Manipulation trajectory optimization with online grasp synthesis and selection," *arXiv preprint arXiv:1911.10280*, 2019.

[6] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *Conference on Robot Learning*. PMLR, 2023, pp. 150–160.

[7] D. Wang, C. Liu, F. Chang, H. Huan, and K. Cheng, "Multi-stage reinforcement learning for non-prehensile manipulation," *IEEE Robotics and Automation Letters*, 2024.

[8] K. Shaw, A. Agarwal, and D. Pathak, "Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning," *arXiv preprint arXiv:2309.06440*, 2023.

[9] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. Jin, Y. Liu, S. Fan, T. Lan, and Z. Chen, "Multisensory five-finger dexterous hand: The dlr/hit hand ii," in *2008 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2008, pp. 3692–3697.

[10] X. Zhang, S. Jain, B. Huang, M. Tomizuka, and D. Romeres, "Learning generalizable pivoting skills," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5865–5871.

[11] S.-M. Yang, M. Magnusson, J. A. Stork, and T. Stoyanov, "Learning extrinsic dexterity with parameterized manipulation primitives," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5404–5410.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[13] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[14] W. Zhou, B. Jiang, F. Yang, C. Paxton, and D. Held, "Hacman: Learning hybrid actor-critic maps for 6d non-prehensile manipulation," *arXiv preprint arXiv:2305.03942*, 2023.

[15] B. Jiang, Y. Wu, W. Zhou, C. Paxton, and D. Held, "Hacman++: Spatially-grounded motion primitives for manipulation," *arXiv preprint arXiv:2407.08585*, 2024.

[16] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.