Adaptive Mobile Manipulation for Articulated Objects In the Open World

Haoyu Xiong
CMU

Russell Mendonca
CMU

Kenneth Shaw
CMU

Deepak Pathak
CMU

Fig. 1: **Open-World Mobile Manipulation System**: We introduce a **full-stack** approach to tackle realistic articulated object operation, e.g. real-world doors, cabinets, drawers, and refrigerators in open-ended unstructured environments.

***Abstract*— Deploying robots in open-ended unstructured environments such as homes has been a long-standing research problem. However, robots are often studied only in closed-off lab settings, and prior mobile manipulation work is restricted to pick-move-place, which is arguably just the tip of the iceberg in this area. In this paper, we introduce Open-World Mobile Manipulation System, a full-stack approach to tackle realistic articulated object operation, e.g. real-world doors, cabinets, drawers, and refrigerators in open-ended unstructured environments. We propose an adaptive learning framework in which the robot initially learns from a small set of data through behavior cloning, followed by learning from online self-practice on novel variations that fall outside the BC training domain. We develop a low-cost mobile manipulation hardware platform capable of repeatedly safe and autonomous online adaptation in unstructured environments with a cost of around $20,000$ USD. We conducted a field test on 20 novel doors across 4 different buildings on a university campus. In a trial period of less than one hour, our system demonstrated significant improvement, boosting the success rate from 25% of BC pre-training to 70% of online adaptation without any human intervention. Video results at `https://door-open.github.io/`.**

I. Introduction

Deploying robotic systems in unstructured environments such as homes has been a long-standing research problem. In recent years, significant progress has been made in learning-based approaches [1]–[4]. However, this progress has been largely made independently either in mobility or in manipulation, while a wide range of practical robotic tasks require dealing with both aspects [5]–[8]. The joint study of mobile manipulation paves the way for generalist robots which can perform realistic and useful tasks in open-ended unstructured environments, as opposed to being restricted to controlled laboratory settings focused primarily on tabletop manipulation. However, developing and deploying such robot systems in the *open-world* with the capability of handling unseen scenarios is challenging for a variety of reasons, ranging from the lack of capable mobile manipulator hardware systems to the difficulty of operating in diverse variations. Consequently, most of the recent mobile manipulation results end up being limited to pick-move-place tasks [9], [10], which is arguably representative of only a small fraction of problems in this space.

In this project, we focus on mobile manipulation for everyday articulated objects involving *real* doors, drawers, refrigerators, or cabinets in unstructured open-world environments. Realistic articulated object operations, e.g. door opening, is a common and essential task encountered in everyday life and is a long-standing problem in the community [11]–[17]. The primary challenge for open-world scenarios is generalizing effectively across the diverse variety of such objects in unstructured environments rather than manipulation tasks in a controllable lab setup. Furthermore, we also need capable hardware, as real-world tasks, e.g. door opening, not only require a powerful and dexterous manipulator, but the base has to be stable enough to balance while the door is being opened and agile enough to walk through the door if required.

We take a **full-stack** approach to address the above challenges. To effectively mobile-manipulate objects in open-world settings, we adopt an *adaptive learning* framework. Our robot starts by initializing the policy using a limited set of training data through behavior cloning. The central challenge we face is operating with novel objects that fall outside the training domain. To address this, our robot keeps learning from online samples collected during self-practice via reinforcement learning. Hence even if the robot encounters a novel door with a different mode of articulation, or with different physical parameters, e.g. weight or friction, it can adapt using online practice data.

Furthermore, we develop a mobile manipulation system to jointly study the hardware, data collection, and adaptive learning algorithms. Learning via repeated online interaction in unstructured environments requires capable hardware. As shown in Figure 3, we provide a simple and intuitive solution to build a mobile manipulation hardware platform, followed by two main principles: (1) Affordability and Rapid-Prototyping. The platform employs off-the-shelf components, e.g. robot arms, and the total cost of the entire system is around $20,000$ USD, making it affordable for most research labs. This allows researchers to reproduce and quickly assemble the system with ease and unlocks the potential of large-scale open-world data collection in the future. (2) Versatile and agile. Our platform offers high payload capabilities via a strong arm and base, making it capable of articulating realistic objects, e.g. a heavy, spring-loaded door. Additionally, we sought to develop a human-sized, agile platform with an omnidirectional base capable of maneuvering across various doors and navigating around narrow and cluttered spaces. For such a real-world system to be effective, it is critical to be able to learn efficiently, since it is expensive to collect real-world data. Consider how people typically approach operating articulated objects such as doors. This generally first involves reaching towards a part of the object (such as a handle) and establishing a grasp. We then execute constrained manipulation like rotating, unlocking, opening, , where we apply arm or body movement to manipulate the object. In addition to this high-level strategy, there are also lower-level decisions made at each step regarding the exact direction of movement, extent of perturbation, and amount of force applied. Inspired by this, we use an action space for our controller, where the action sequence follows the grasp, constrained manipulation strategy. These primitives are parameterized by learned discrete and continuous values, which need to be adapted to operate diverse articulated objects.

To further bias the exploration of the system towards reasonable actions and avoid unsafe actions during online sampling, we collect a dataset of expert demonstrations on 12 training objects, including doors, drawers and cabinets to train an initial policy via behavior cloning. While this is not very performant on new unseen doors (getting around **25%** accuracy), starting from this policy allows subsequent learning to be faster and safer. Finally, we develop a mobile manipulation system capable of fully autonomous Reinforcement Learning (RL) adaptation, utilizing a Vision-Language Model (VLM) for reward function. We conducted a large-scale field test of 8 novel objects ranging across 4 buildings on a university campus. For each novel object, un a trial period of less than one hour, this system demonstrated significant improvement, boosting the success rate from 25% to 70% without any human intervention.

In this paper, we present **Open-World Mobile Manipulation System**, a **full stack** approach to tackle the problem of mobile manipulation of realistic articulated objects in the open world. We adopt an adaptive learning method that allows the robot to keep learning from self-practice data via online RL, boosting the success rate from 25% (pre-training) to 70% (adaptation) without any human intervention. We introduce a low-cost mobile manipulation hardware platform that offers a high payload, making it capable of repeated interaction with objects, e.g. a heavy, spring-loaded door, and a human-size, capable of maneuvering across various doors and navigating around narrow and cluttered spaces in the open world. We conducted a field test of 8 novel objects ranging across 4 buildings on a university campus and validated the efficiency and effectiveness of our system.

II. RELATED WORK

a) Real-world Policy Fine-Tuning: Autonomous fine-tuning policies in the real world has become popular in recent years, notably in manipulation [18] and locomotion [19]. Leveraging prior from human videos, and keep finetuning the policy with online robot interaction data is a promising avenue in manipulation [20]–[22]. The approach of pretraining in simulation and finetuning with a small amount of real-world data has shown significant improvement for legged robots [19]. In our study, we showcase the autonomous fine-tuning of real-world policies with mobile manipulation, particularly in challenging, unstructured open-world environments.

b) Learning-based Mobile Manipulation Systems. : In recent years, the setup for mobile manipulation tasks in both simulated and real-world environments has been a prominent topic of research [5], [23]–[31]. Notably, several studies have explored the potential of integrating Large Language Models into personalized home robots, signifying a trend towards more interactive and user-friendly robotic systems [31]–[33]. Tidybot [32] relies on external sensors, we only use on-board sensor, which enables in-the-wild systems.

Unlike the majority of prior research, which predominantly focuses on pick-move-place tasks [9], our work expands on these efforts by introducing an open-world mobile manipulation system capable of articulating everyday objects in unstructured environments, which present an increased level of difficulty.

c) Door Manipulation: The research area of door opening in robotics has a rich history [14]–[17], [34]. A significant milestone in the domain was the DARPA Robotics Challenge (DRC) finals in 2015. The accomplishment of the WPI-CMU team in door opening illustrated not only

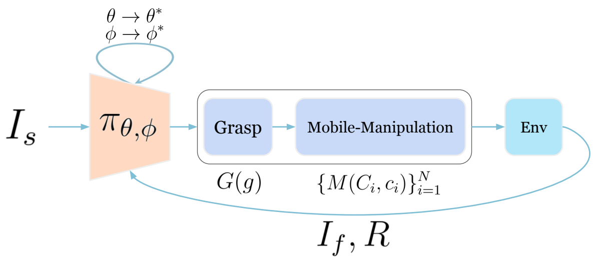

Fig. 2: **Adaptive Learning Framework**: We use a parametrized primitive action space, with grasping and mobile-manipulation primitives.

advances in robotic manipulation and control but also the potential of humanoid robots to carry out intricate tasks in real-world environments [11]–[13]. Nevertheless, prior to the deep learning era, the primary impediment was the robots' perception capabilities, which faltered when confronted with tasks necessitating visual comprehension of complex and unstructured environments. Wang et al. [35], leverages synthetic data to train keypoint representation for the grasping pose estimation. Qin et, al. [36], proposed an end-end point cloud RL framework for sim2real transfer. Other significant work in this field includes Doorgym [37], which provides a simulation benchmark for door opening tasks. Moreover, the prospect of large-scale RL combined with sim-to-real transfer holds great promise for generalizing to a diverse range of doors in real-world settings [36]–[38]. However, one major drawback of these methods lies in the lack of diversity in simulation environments, particularly concerning the contextual understanding of doors (e.g., discerning whether the door opens inward or outward). In our work, we design a mobile manipulation system to collect diverse robot demonstrations in the open world.

III. Adaptive Learning Framework

In this section, we describe our algorithm framework for training robots for adaptive mobile manipulation of everyday articulated objects. To achieve efficient learning, we use a structured hierarchical action space. This uses a fixed high-level action strategy and learnable low-level control parameters. Using this action space, we start from initailizing our polict via behavior cloning with a diverse dataset collected by ourselves. This provides a strong prior for exploration and decreases the likelihood of executing unsafe actions. However, the initalized BC policy might not generalize to every unseen object that the robot might encounter due to the large scope of variation of objects in the open-world environments. Hence we enable the robot to learn from the online samples it collects to continually learn and adapt. We describe the continual learning process as well as design considerations for online learning.

A. Action Space

For greater learning efficiency, we use a parametrized primitive action space. Concretely, we assume access to a grasping primitive $G(.)$ parameterized by g. We also have a constrained mobile-manipulation primitives $M(.)$, where

Algorithm 1 Adaptive Learning

Require: Grasping primitive $G(.)$ taking parameter g
Require: Constrained manipulation primitives $M(.)$, taking parameter C and c.
Initialize primitive classifier $\pi_\phi(\{C_i\}_{i=1}^N | I)$
Initialize conditional action policy $\pi_\theta(g, \{c_i\}_{i=1}^N | I, \{C_i\}_{i=1}^N)$
Collect a dataset D of expert demos $\{I, g, \{C_i\}_{i=1}^N, \{c_i\}_{i=1}^N\}$
Train π_ϕ and π_θ on D using Imitation Learning 2
for online RL iteration 1:N **do**
  Given image I_s, sample $\{C_i\}_{i=1}^N \sim \pi_\phi(.|I_s)$, sample $(g, \{c_i\}_{i=1}^N) \sim \pi_\theta(.|I_s)$
  Execute trajectory $\{G(g), \{M(C_i, c_i)\}_{i=1}^N\}$, observe reward R
  Update policies π_ϕ and π_θ using RL (Eqs. 5, 4, 2)
end for

primitive $M(.)$ takes two parameters, a discrete paramter C and a continuous paramter c. Trajectories are executed in an open-loop manner, a grasping primitive followed by a sequence of N constrained mobile-manipulation primitives:

$$\{I_s, G(g), \{M(C_i, c_i)\}_{i=1}^N, I_f, R\}$$

where I_s is the initial observed image, $G(g)$, $M(C_i, c_i))$ denote the parameterized grasp and constrained manipulation primitives respectively, I_f is the final observed image, and r is the reward for the trajectory. While this structured space is less expressive than the full action space, it is large enough to learn effective strategies for the everyday articulated objects we encountered, covering 20 different doors, drawers, and fridges in open-world environments. The key benefit of the structure is that it allows us to learn from very few samples, using only on the order of 20-30 trajectories. We describe the implementation details of the primitives in the next section.

B. Adaptive Learning

Given an inistial observation image I_s, we use a classifier $\pi_\phi(\{C_i\}_{i=1}^N | I)$ to predict the a sequence of N discrete parameters $\{C_i\}_{i=1}^N$ for constrained mobile-manipulation, and a conditional policy network $\pi_\theta(g, \{c_i\}_{i=1}^N | I, \{C_i\}_{i=1}^N)$ which produces the continuous parameters of the grasping primitive and a sequence of N constrained mobile-manipulation primitives. The robot executes the parameterized primitives one by one in an open-loop manner.

1) Imitation: We start by initializing our policy using a small set of expert demonstrations via behavior cloning. The imitation learning objective is to learn policy parameters $\pi_{\theta,\phi}$ that maximize the likelihood of the expert actions. Specifically, given a dataset of image observations I_s, and corresponding actions $\{g, \{C_i\}_{i=1}^N, \{c_i\}_{i=1}^N\}$, the imitation learning objective is:

$$\max_{\phi,\theta} \left[\log \pi_\phi(\{C_i\}_{i=1}^N \,|\, I_s) + \log \pi_\theta(g, \{c_i\}_{i=1}^N \,|\, \{C_i\}_{i=1}^N, I_s)\right] \tag{1}$$

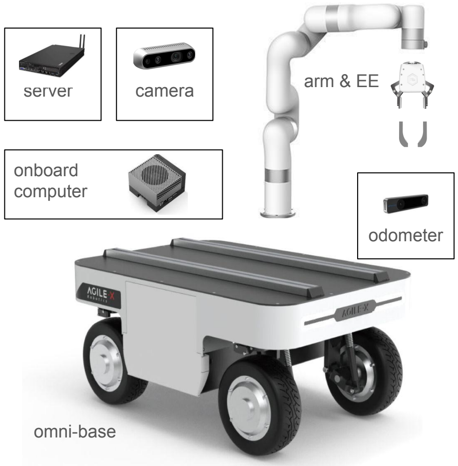

Fig. 3: **Mobile Manipulation Hardware Platform:** Hardware design is low-cost and easy-to-build with off-the-shelf components

2) Online RL: The central challenge we face is operating with new objects that fall outside the behavior cloning training domain. To address this, we develop the policy to keep improving using the online samples collected by the robot. This corresponds to maximizing the expected sum of rewards under the policy :

$$\max_{\theta,\phi} \mathbb{E}_{\pi_{\theta,\phi}} \left[\sum_{t=0}^{T} r(s_t, a_t) \right] \tag{2}$$

Since we utilize a highly structured action space as described previously, we can optimize this objective using a fairly simple RL algorithm. Specifically we use the REINFORCE objective [39]:

$$\nabla_{\theta,\phi} J(\theta,\phi) = \mathbb{E}_{\pi_{\theta,\phi}} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi(a_t|s_t) \cdot r_t \right] \tag{3}$$

$$= \mathbb{E}_{\pi_{\phi,\theta}} \left[(\nabla_\phi \log \pi_\phi(C_i|I) + \nabla_\theta \log \pi_\theta(g, c_i|C_i, I)) \cdot R \right] \tag{4}$$

where R is the reward provided at the end of trajectory execution. Note that we only have a single time-step transition, all actions are determined from the observed image I_s, and executed in an open-loop manner.

3) Overall Finetuning Objective: To ensure that the policy doesn't completely forget the initialization from the imitation dataset, we use a weighted objective while finetuning, where the overall loss is :

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{online}} + \alpha * \mathcal{L}_{\text{offline}} \tag{5}$$

where loss on online sampled data is optimized via Eq.4 and loss on the batch of offline data is optimized via BC as in Eq.2. We use equal sized batches for online and offline data while performing the update.

IV. Open-world Mobile Manipulation Systems

In this section, we introduce the detail of our *full-stack* approach encompassing hardware, offline and online data collection, and algorithms to address the challenges of enabling mobile manipulation robots adaptive learning in open-world environments, particularly with everyday articulated objects like cabinets, drawers, refrigerators, and real doors.

A. Hardware

The transition from tabletop manipulation to mobile manipulation is challenging not only from algorithmic studies but also from the perspective of hardware, guided by the following principles. (1) Affordability and Rapid-Prototyping. The platform is designed to be low-cost for most robotics labs and employs off-the-shelf components. This allows researchers to reproduce and quickly assemble the system with ease, and unlocks the potential of large-scale open-world data collection in the future. (2) Versatile and agile. The platform is designed to perform articulating everyday objects, like opening a heavy, spring-loaded door. To achieve this, the platform must offer high payload capabilities via a strong arm and base. Additionally, we sought to develop a human-sized, agile platform capable of maneuvering across various real-world doors and navigating unstructured and narrow environments, such as cluttered office spaces.

In this project, we provide a simple and intuitive solution to build a mobile manipulation the hardware platform. As shown in Figure 3, among the commercially available options, we found the Ranger Mini 2 from AgileX to be an ideal choice for robot base due to its stability, omni-directional velocity control, and high payload capacity. The system uses an xArm for manipulation, which is an effective low-cost arm with a high payload (5kg), and is widely avaible in research labs. The system uses a Jetson computer to support real-time communication between sensors, the base, the arm, as well as a server that hosts large models. We use a D435 Intel Realsense camera mounted on the frame to collect RGBD images as ego-centric observations and a T265 Intel Realsense camera to provide visual odometry which is critical for RL reset in mobile manipulation setup. The gripper is equipped with a 3d-printed hooker and an anti-slip tape to ensure a secure and stable grip. The overall cost of the entire system is around $20,000$ USD, making it an affordable solution for most robotics labs.

B. Primitive Implementation

In this subsection, we introduce the implementation details of our paramterized primitive action space.

1) Grasping: Given the RGBD image of the scene obtained from the realsense camera, we use off-the-shelf visual models [40], [41] to obtain the mask of the door and handle given just text prompts. Furthermore, since the door is a flat plane, we can estimate the surface normals of the door using the corresponding mask and the depth image. This is used to move the base close to the door and align it to be perpendicular, and also to set the orientation angle for grasping the handle. The center of the 2d mask of the handle is projected into 3d coordinates using camera calibration, and this is the nominal grasp position. The low-level control parameters to the grasping primitive indicate an offset for this

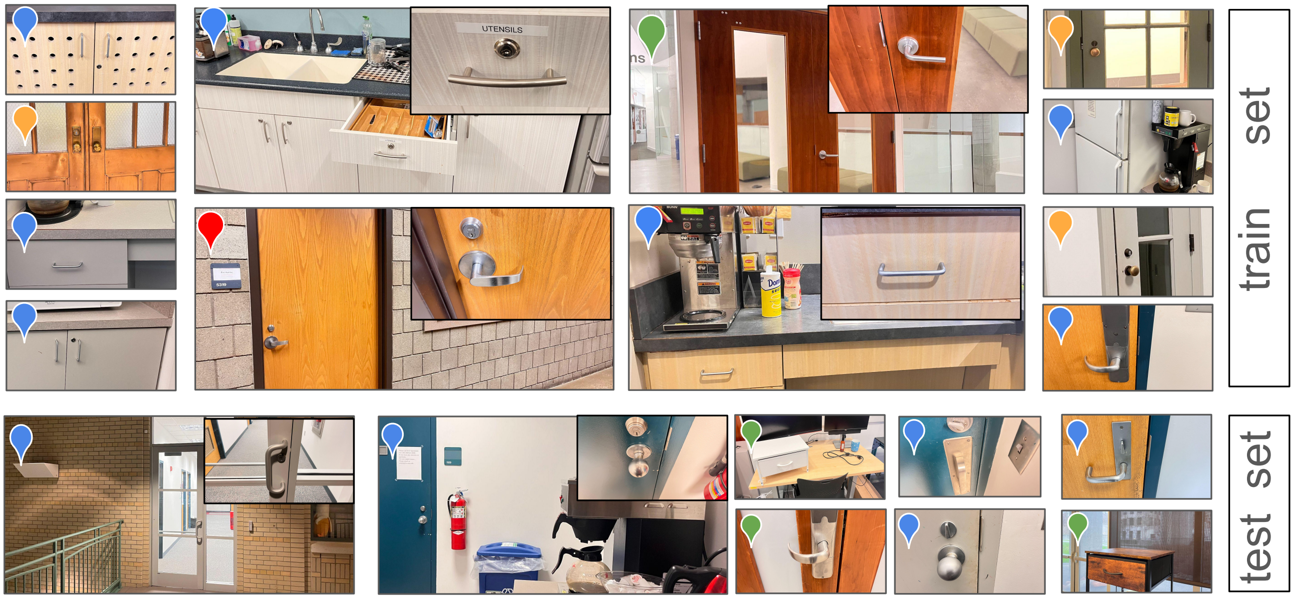

Fig. 4: **Large field study on CMU campus.** We conduct an extensive field study involving 12 training objects and 8 testing objects. Our training objects and testing objects are significantly different from each other, in terms of different visual appearances, different modes of articulation, or different physical parameters, e.g. weight or friction.

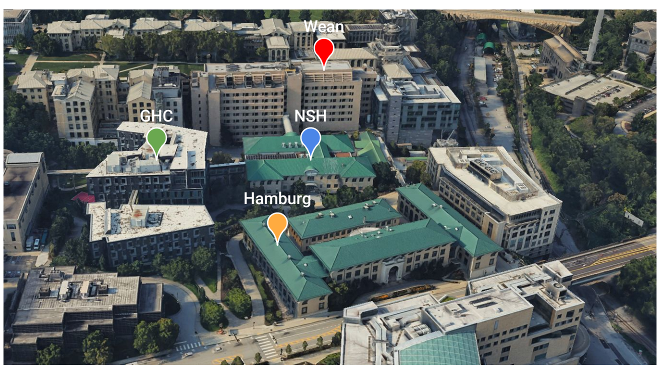

Fig. 5: **Field Test on CMU Campus**We conduct an extensive field study across four distinct buildings on the Carnegie Mellon University campus to test the efficacy of our system.

position at which to grasp. This is beneficial since depending on the type of handle the robot might need to reach a slightly different position which can be learned via the low-level continuous valued parameters. After the robot reaches the desired grasping pose, we commend the gripper to close.

2) Constrained Mobile-Manipulation: **todo:** specify the coordinate in Fig 3

We use velocity control for the robot arm end-effector and the robot base.

$$\text{Control}: \quad (v_x, v_y, v_z, v_{\text{yaw}}, v_{\text{pitch}}, v_{\text{roll}}, V_{\text{x}}, V_{\text{y}}, V_{\omega})$$

Specifying constraints on this 9-dimensional vector, we have three velocity controller with fixed velocities.

$$\begin{aligned} \text{Unlock}: &\quad (0,0,v_z,v_{\text{yaw}},0,0,0,0,0) \\ \text{Rotate}: &\quad (0,0,0,v_{\text{yaw}},0,0,0,0,0) \\ \text{Open}: &\quad (0,0,0,0,0,0,V_{\text{x}},0,0) \end{aligned}$$

Given the control parameters, a discrete parameter C and a continuous parameter c. C is to choose which velocity controller to use and the continuous parameter c is to determine duration to execute the velocity control. Here, the parameter c is continuous valued from -1 to 1. The absolute value of c is adjusted to a scale of 2.5 seconds, which is used to determine the duration of the velocity control execution. The sign of c (positive or negative) dictates the direction of the velocity control, either clockwise or counter-clockwise for unlock and rotate, forward or backward for open.

C. Pre-training Dataset

In this project, we consider a set of articulated objects, e.g. doors, cabinets, drawers, which consist of three rigid parts: a base part, a frame part, and a handle part. The base and frame are connected by either a revolute joint (as in a cabinet) or a prismatic joint (as in a drawer). The frame is connected to the handle by either a revolute joint or a fixed joint. We identify four major types of the articulated objects, which relate to the type of handle, and the joint mechanisms. Handle articulations commonly include levers (Type A) and knobs (Type B). For cases where handles are not articulated, the body-frame can revolve about a hinge using a revolute joint (Type C), or slide back and forth along a prismatic joint, for example, drawers (Type D). While not exhaustive, this categorization covers a wide variety of

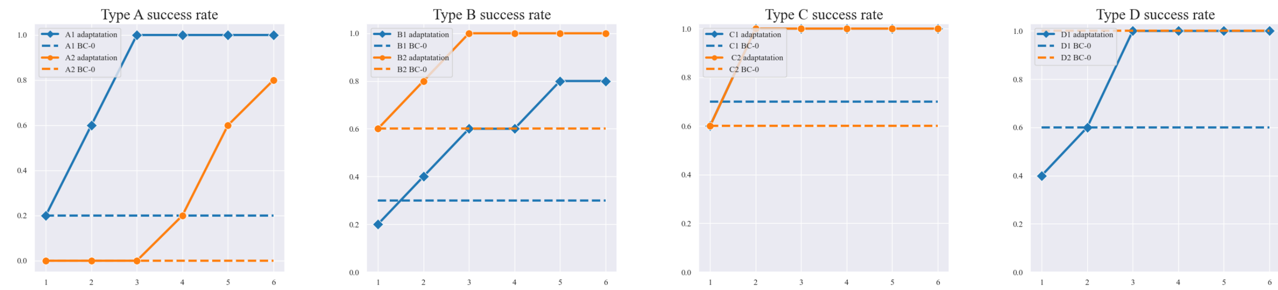

Fig. 6: **Online Adaptation Improvement.** We demonstrate enhanced online adaptation for four types of articulated objects. The plots reveal that the adapted policies significantly outperform the initial behavior cloning policies.

erveryday articulated objects a robot system might encounter. To provide generalization benefits in operating unseen novel articulated objects, we first collect a offline demonstration dataset. We include 3 objects from each category in the BC training dataset, collecting 10 demonstrations for each object, producing a total of 120 trajectories. For offline dataset collection, like video games, we use keyboard to map button presses to both discrete (button press down) and continuous parameters (long press), allowing us to teleoperate the robot for data collection. To execute discrete actions, we use different buttons.

We also have 2 held-out testing objects from each category for generalization experiments. The training and testing objects differ significantly in visual appearance (eg. texture, color), physical dynamics (eg. if spring-loaded), and actuation (e.g. the handle joint might be clockwise or counter-clockwise). We include visualizations of all objects used in train and test sets in Fig. 4, along with which part of campus they are from as visualized in Fig. 5.

D. Autonomous and Safe Online Adaptation

The key challenge we face is operating with new objects that fall outside the BC training domain. To address this, we develop a system capable of fully autonomous Reinforcement Learning (RL) online adaptation, utilizing a Vision-Language Model (VLM) for reward function. In this subsection, we demonstrate the details of the autonomy and safety of our system.

1) Safety Aware Exploration: It is crucial to ensure that the actions the robot takes for exploring are safe for its hardware, especially since it is interacting with objects under articulation constraints. Ideally, this could be addressed for dynamic tasks like door opening using force control. However, low-cost arms like the xarm-6 we use do not support precise force sensing. For deploying our system, we use a safety mechanism based which reads the joint current during online sampling. If the robot samples an action that causes the joint current to meet its threshold, we terminate the episode and reset the robot, to prevent the arm from potentially damaging itself, and this also gives the robot a negative reward as a penalty.

2) Vision-Language Model as a Reward Function: To distinguish whether the door is open or not, We give the CLIP [42] text encoder two prompts *"door that is closed"* and *"door that is open"*, and give the CLIP image encoder an RGB image I. We calculate the similarity scores between the image and each text description. Given an RGB image I_s before online sampling starts, and an RGB image I_f when online sampling ends from the same camera viewpoint after the robot is programmed to return to the initialized base pose. We define the reward function as a binary signal: if the similarity score between I_f and the text "door that is open" is larger than the similarity score between I_s and the text "door that is open", the reward is 1, else if, the safety protection is triggered, the reward is minus 1, otherwise, the reward is 0.

3) Reset Mechanism: The robot employs visual odometry, utilizing the T265 tracking camera mounted on its base, enabling it to navigate back to its initial position. At the end of every episode, the robot releases its gripper, and moves back to the original SE2 base position, and take an image of I_f for reward computing, and then a perturbation of base position is added for robustness. Furthermore, if the reward is 1, where the door is opened, the robot will be programmed to close the door.

V. Results

We conduct an extensive field study involving 12 training objects and 8 testing objects across four distinct buildings on the Carnegie Mellon University campus to test the efficacy of our system. In our experiments, we seek to answer the following questions:

1) Can the system adapt to operate various unseen objects via online adaptation?
2) How does this compare to simply using imitation learning on provided demonstrations?
3) Can we automate providing rewards using off-the-shelf vision-language models?
4) How does the hardware design compare with other platforms?

	online adapt success rate							
	lever A	knob A	hinge A	drawer A	lever B	knob B	hinge B	drawer B
BC-0	20%	30%	70%	60%	0%	60%	60%	100%
Adapt-GT	70%	80%	90%	80%	100%	100%	80%	100%

TABLE I: In this table, we present improvements in online adaptation with human-labeled ground truth reward.

A. Online adaptation improvement

In this subsection, we describe the details of the online adaptation. The policy is initialized using the training dataset using behavior cloning (BC). We deploy this model in a zero-shot fashion, testing it on a set of 8 unseen testing objects for evaluation, and then adapt the policy via online interactions. For each test object, we conduct 10 trials and document the success rates in Table I. Our observations indicate that the initial success rate of the BC model (denoted as BC-0) is relatively low, especially for challenging tasks like opening lever doors. To continuously improve the performance of testing objects, we developed a pipeline that enables robots to learn how to manipulate new articulated objects by RL adaptation with a few online samples. We use the BC policy to sample 5 trajectories on testing objects and update the model, then we use the latest updated model to sample another 5 trajectories, see 1 for more details for the online sampling. We use human-labeled rewards to update the model. Human expert follows the rule: if the door is open, the reward is 1, if the door is not open, the human expert gives a score from 0 to 1 to evaluate the whole sampling trajectory. The whole reward function is combined with the human-labeled reward and the safety reward, see 1 for more details on the reward function definition.

We observed an increasing improvement of success rate during online sampling, in Fig. 6, we record the succes rate of 5 trajectories sampling of each round. After six rounds of sampling and updating, we conduct an additional set of 10 trials on each test object. This process leads to a noticeable improvement in the success rates with the updated policies as shown in Table I.

B. autonomous online adaptation with VLM reward

To achieve fully autonomous online adaptation, we replace the human-labeled reward with an auto-reward method using the Vision-Language Model, CLIP [42]. In this subsection, we discuss the details of this method.

We conducted online adaptation with VLM reward on two objects, lever A and knob A, and reported a success rate improvement graph in Fig. 7. In Table II, we observed that online adaptation with VLM reward achieves a similar performance as using ground-truth human-labeled reward.

C. Baseline comparisons

In this subsection, we aim to evaluate whether simply replay trajectory of training objects on unseen testing objects still remain effective. We investigate an intuitive and simple baseline, Nearest-Neighbor-based replay method. We implemented two methods, an open-loop and a closed-loop manner. In the training dataset, for all image-action pairs,

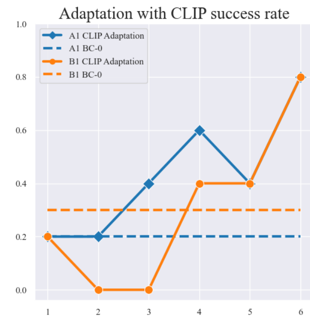

Fig. 7: **Online Adaptation with CLIP reward.** We present improvements in online adaptation with CLIP reward. The result illustrates that adapted policies exhibit substantial enhancements compared to the initialized behavior cloning policies.

online adapt success rate			
	BC-0	Adapt-GT	Adapt-CLIP
Success Rate lever A	20%	70%	70%
Success Rate knob A	30%	80%	70%

TABLE II: In this table, we present improvements in online adaptation with CLIP reward.

we use the CLIP [42] image encoder to encode all the images. For open-loop method, to evaluate the performance of Nearest-Neighbor-based method on unseen testing objects, we take an image from the initialization, and compute the CLIP encoded embedding euclidean distance between the current image and the first images of each trajectories in the training dataset. We then replay the actions in the trajectory whoes first image embedding is closest to the current inital image. For closed-loop, similar to open-loop, at each time-step, we take an image from the head camera and compute the euclidean distances between the current image and the images in the training dataset, find the nearest one, and replay its corresponding action. Based on the results presented in Table III, it is evident that the Nearest-Neighbor-based replay method demonstrates a significantly poor success rate. This indicates that merely replaying trajectories from training set objects on testing set objects is ineffective.

D. Hardware Platform Discussion

Prior work has discussed mobile manipulation systems for everyday household tasks [43], [44], such as opening a door using sim2real approaches [8] or whole-body control

baseline				
	KNN-open	KNN-close	BC-0	Adapt-GT
success rate knob A	10%	0%	30%	80%
success rate lever B	0%	0%	0%	100%

TABLE III: We compare the performance of our adaptation policies and initialized BC policies with KNN baselines.

Hardware features comparison						
	Arm payload	DoF arm	omni-base	footprint	base max speed	price
Stretch RE1 [8]	1.5kg	2	×	34 cm, 33 cm	0.6 m/s	20k USD
Go1-air + WidowX 250s [30]	0.25kg	6	✓	59 cm, 22 cm	2.5 m/s	10k USD
Franka + Clearpath Ridgeback [43]	3kg	7	✓	96 cm, 80 cm	1.1 m/s	75k USD
Franka + Omron LD-60 [44]	3kg	7	×	70 cm, 50 cm	1.8 m/s	50k USD
Xarm-6 + Agilex Ranger mini 2 (ours)	5kg	6	✓	74 cm, 50 cm	2.6 m/s	20k USD

TABLE IV: hardware spec sheet.

expert teleoperation success rate		
	lever B	knob A
Stretch RE1	0/5	0/5
Ours	5/5	5/5

TABLE V: We performed experiments using the Stretch RE1 robot and our platform, where a human expert teleoperated the systems to manipulate two test objects, namely lever B and knob A.

for a quadruped robot with an arm [30]. However, these systems might not fully address the requirements for realistic household applications.

We compare key aspects of our modular platform with that of other mobile manipulation platforms in Table IV.This comparison highlights advantages of our system such as cost-effectiveness, reactivity, ability to support a high-payload arm, and a base with omnidirectional drive.

Furthermore, the robot needs to be strong enough to open and move through various doors, some of which can be quite heavy. We empirically compare against a different popular mobile manipulation system, namely the Stretch RE1 (Hello Robot). We test the ability of the robots to be teleoperated by a human expert to open two doors from different categories, specifically lever and knob doors. Each object was subjected to five trials. As shown is Table V, the outcomes of these trials revealed a significant limitation of the Stretch RE1: its payload capacity is inadequate for opening a real door, even when operated by an expert, while our system succeeds in all trials.

ACKNOWLEDGMENT

We thank Shikhar Bahl, Tianyi Zhang, Xuxin Cheng, Shagun Uppal, and Shivam Duggal for the helpful discussions.

REFERENCES

[1] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," *RSS*, 2022. 1

[2] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021. 1

[3] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra *et al.*, "Goat: Go to any thing," *arXiv preprint arXiv:2311.06430*, 2023. 1

[4] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," 2023. 1

[5] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *arXiv*, 2024. 1, 2

[6] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," 2023. 1

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-1: Robotics transformer for real-world control at scale," 2023. 1

[8] R. Yang, Y. Kim, A. Kembhavi, X. Wang, and K. Ehsani, "Harmonic mobile manipulation," 2023. 1, 7, 8

[9] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv preprint arXiv:2306.11565*, 2023. 1, 2

[10] C. Sun, J. Orbik, C. Devin, B. Yang, A. Gupta, G. Berseth, and S. Levine, "Fully autonomous real-world reinforcement learning with applications to mobile manipulation," 2021. 1

[11] C. G. Atkeson, P. B. Benzun, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin *et al.*, "What happened at the darpa robotics challenge finals," *The DARPA robotics challenge finals: Humanoid robots to the rescue*, pp. 667–684, 2018. 1, 3

[12] M. DeDonato, F. Polido, K. Knoedler, B. P. Babu, N. Banerjee, C. P. Bove, X. Cui, R. Du, P. Franklin, J. P. Graff *et al.*, "Team wpi-cmu: achieving reliable humanoid behavior in the darpa robotics challenge," *Journal of Field Robotics*, vol. 34, no. 2, pp. 381–399, 2017. 1, 3

[13] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir, "Human-supervised control of the atlas humanoid robot for traversing doors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 722–729. 1, 3

[14] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1799–1806. 1, 2

[15] A. Jain and C. C. Kemp, "Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator." Georgia Institute of Technology, 2008. 1, 2

[16] K. Nagatani and S. Yuta, "An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 2, 1995, pp. 45–50 vol.2. 1, 2

[17] L. Peterson, D. Austin, and D. Kragic, "High-level control of a mobile manipulator for door opening," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 3, 2000, pp. 2333–2338 vol.3. 1, 2

[18] J. Yang, M. S. Mark, B. Vu, A. Sharma, J. Bohg, and C. Finn, "Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning," 2023. 2

[19] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," 2021. 2

[20] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 778–13 790. 2

[21] R. Mendonca, S. Bahl, and D. Pathak, "Structured world models from human videos," 2023. 2

[22] A. Kannan, K. Shaw, S. Bahl, P. Mannam, and D. Pathak, "Deft: Dexterous fine-tuning for real-world hand policies," *arXiv preprint arXiv:2310.19797*, 2023. 2

[23] B. Wu, R. Martin-Martin, and L. Fei-Fei, "M-ember: Tackling long-horizon mobile manipulation via factorized domain transfer," *arXiv preprint arXiv:2305.13567*, 2023. 2

[24] N. Yokoyama, A. W. Clegg, E. Undersander, S. Ha, D. Batra, and A. Rai, "Adaptive skill coordination for robotic mobile manipulation," *arXiv preprint arXiv:2304.00410*, 2023. 2

[25] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on Robot Learning*. PMLR, 2022, pp. 477–490. 2

[26] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347. 2

[27] J. Wong, A. Tung, A. Kurenkov, A. Mandlekar, L. Fei-Fei, S. Savarese, and R. Martín-Martín, "Error-aware imitation learning from teleoperation data for mobile manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 1367–1378. 2

[28] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1647–1654. 2

[29] Y. Zhao, Q. Gao, L. Qiu, G. Thattai, and G. S. Sukhatme, "Opend: A benchmark for language-driven door and drawer opening," *arXiv preprint arXiv:2212.05211*, 2022. 2

[30] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149. 2, 8

[31] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022. 2

[32] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023. 2

[33] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022. 2

[34] R. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based perception for door and handle identification," 07 2009, pp. 1 – 8. 2

[35] J. Wang, S. Lin, C. Hu, Y. Zhu, and L. Zhu, "Learning semantic keypoint representations for door opening manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6980–6987, 2020. 3

[36] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 594–605. 3

[37] Y. Urakami, A. Hodgkinson, C. Carlin, R. Leu, L. Rigazio, and P. Abbeel, "Doorgym: A scalable door opening environment and baseline agent," *arXiv preprint arXiv:1908.01887*, 2019. 3

[38] A. Gupta, M. E. Shepherd, and S. Gupta, "Predicting motion plans for articulating everyday objects," *arXiv preprint arXiv:2303.01484*, 2023. 3

[39] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992. 4

[40] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer, 2022, pp. 350–368. 4

[41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” *arXiv preprint arXiv:2304.02643*, 2023. 4

[42] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763. 6, 7

[43] J. Kindle, F. Furrer, T. Novkovic, J. J. Chung, R. Siegwart, and J. Nieto, “Whole-body control of a mobile manipulator using end-to-end reinforcement learning,” 2020. 7, 8

[44] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, “Sayplan: Grounding large language models using 3d scene graphs for scalable task planning,” *arXiv preprint arXiv:2307.06135*, 2023. 7, 8