In Pursuit of Open-World Mobile Manipulation

Haoyu Xiong

CMU-RI-TR-24-05

April 24, 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor Deepak Pathak, *chair*
Professor Guanya Shi,
Kenneth Shaw

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

To my mother and father.

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. The robot utilizes an adaptive learning framework to initially learns from a small set of data through behavior cloning, followed by learning from online practice on novel objects that fall outside the training distribution. We also develop a low-cost mobile manipulation hardware platform capable of safe and autonomous online adaptation in unstructured environments with a cost of around $25,000$ USD. In our experiments we utilize 20 articulate objects across 4 buildings in an university campus. With less than an hour of online learning for each object, the system is able to increase success rate from 50% of BC pre-training to 95% using online adaptation.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Professor Deepak Pathak, whose guidance and support have been invaluable throughout this journey.

I extend my heartfelt appreciation to my committee members, Professor Guanya Shi and Kenneth Shaw, for their insightful feedback and encouragement.

I am deeply thankful to every member of my lab for their contributions and camaraderie. Special thanks to Ananye Agarwal, Dr. Shikhar Bahl, Lili Chen, Alex Li, Russell Mendonca, Mihir Prabhudesai, Dr. Unnat Jain, Alexandre Kirchmeyer, Shagun Uppal, Kexin Shi, Murtaza Dalal, Shivam Duggal, Ellis Brown, and Xuxin Cheng for their unwavering support and collaboration.

To my friends, Jianren Wang, Tianyi Zhang, Zipeng Fu, Ken Liu, Tianyuan Zhang, Heng Yu, and Quanting Xie, your encouragement and friendship have meant the world to me.

Last but not least, I am deeply grateful to my parents for their unwavering love, encouragement, and sacrifices.

Funding

This work was supported in part by CMU-AIST Bridge project, AFOSR research grant FA9550-23-1-0747 and Sony faculty award.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

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 deploying learning-based approaches [3, 8, 27, 50] towards this goal. 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 [7, 14, 49, 61]. The joint study of mobile manipulation paves the way for generalist robots which can perform 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 objects is challenging for a variety of reasons, ranging from the lack of capable mobile manipulator hardware systems to the difficulty of operating in diverse scenarios. Consequently, most of the recent mobile manipulation results end up being limited to pick-move-place tasks[20, 32, 53, 62], which is arguably representative of only a small fraction of problems in this space. Since learning for general-purpose mobile manipulation is challenging, we focus on a restricted class of problems, involving the operation of articulated objects, such as doors, drawers, refrigerators, or cabinets in open-world environments. This is a common and essential task encountered in everyday life, and is a long-standing problem in the community [2, 5, 10, 11, 21, 37, 40]. The primary challenge is generalizing effectively across the diverse variety of such objects in unstructured real-world environments

rather than manipulating a single object in a constrained lab setup. Furthermore, we also need capable hardware, as opening a door not only requires 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.

We take a **full-stack** approach to address the above challenges. In order to effectively manipulate objects in open-world settings, we adopt a *adaptive learning* approach, where the robot keeps learning from online samples collected during interaction. Hence even if the robot encounters a new door with a different mode of articulation, or with different physical parameters like weight or friction, it can keep adapting by learning from its interactions. For such a system to be effective, it is critical to be able to learn efficiently, since it is expensive to collect real world samples. The mobile manipulator we use as shown in Figure. 3.2 has a very large number of degrees of freedom, corresponding to the base as well as the arm. A conventional approach for the action space of the robot could be regular end-effector control for the arm and SE2 control for the base to move in the plane. While this is very expressive and can cover many potential behaviors for the robot to perform, we will need to collect a very large amount of data to learn control policies in this space. Given that our focus is on operating articulated objects, can we structure the action space so that we can get away with needing fewer samples for learning?

Consider the manner in which 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, unlatching, or unhooking, 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 exact direction of movement, extent of perturbation and amount of force applied. Inspired by this, we use a hierarchical action space for our controller, where the high-level action sequence follows the grasp, constrained manipulation strategy. These primitives are parameterized by learned low-level continuous values, which needs 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 50% accuracy), starting from this policy allows subsequent learning to be faster and safer.

Learning via repeated online interaction also requires capable hardware. As shown in Figure 3.2, we provide a simple and intuitive solution to build a mobile manipulation hardware platform, followed by two main principles: (1) versatility and agility - this is essential to effectively operate diverse objects with different physical properties in potentially challenging environments, for instance a cluttered office. (2) affordability and rapid-prototyping - Assembled with off-the-shelf components, the system is accessible and can be readily be used by most research labs.

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. Efficient learning is enabled by a structured action space with parametric primitives, and by pretraining the policy on a demonstration dataset using imitation learning. Adaptive learning allows the robot to keep learning from self-practice data via online RL. We introduce a low-cost mobile manipulation hardware platform that offers 1) a high payload, making it capable of repeated interaction with objects, e.g. a heavy, spring-loaded door, 2) and a human size, which is 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 to test the effectiveness of our system, and found adaptive earning boosts success rate from 50% from the pre-trained policy to 95% after adaptation.

Chapter 2

Related Works

2.1 Adaptive Real-world Robot Learning

There has been a lot of prior work that studies how robots can acquire new behavior by directly using real-world interaction samples via reinforcement learning using reward [22, 23, 29, 30], and even via unsupervised exploration [4, 33, 41]. More recently there have been approaches that use RL to fine-tune policies very efficiently that have been initialized via by imitating demonstrations [16, 17]. Other methods aim to do so without access to demonstrations on the test objects, and pretrain using other sources of data - either using offline robot datasets [28], simulation [51] or human video [18, 24, 34, 60] or a combination of these approaches [20]. We operate in a similar setting, without any demonstrations on test objects, and focus on demonstrating RL adaptation on mobile manipulation systems that can be deployed in open-world environments. While prior large-scale industry efforts also investigate this [20], we seek to be able to learn much more efficiently with fewer data samples. Prior research has been dedicated to utilizing real-world online data to infer environmental parameters or latent representations for adapting policies [9, 31, 36, 44, 46]. For example, Active Sys-Id is a promising direction, Jacky Liang [31] proposes a framework for training task-oriented exploration policies to identify system parameters.

2.2 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 [6, 13, 14, 35, 48, 52, 57, 58, 63, 64]. 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 [1, 6, 59]. While these systems display impressive long horizon capabilities using language for planning, these assume fixed low-level primitives for control. In our work we seek to learn low-level control parameters via interaction. Furthermore, unlike the majority of prior research which predominantly focuses on pick-move-place tasks [62], we consider operating articulated objects in unstructured environments, which present an increased level of difficulty.

2.3 Door Manipulation

The research area of door opening has a rich history in the robotics community [10, 21, 37, 40, 47]. 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 advances in robotic manipulation and control but also the potential of humanoid robots to carry out intricate tasks in real-world environments [2, 5, 11]. 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. Approaches using deep learning to address vision challenges include Wang et al. [55], which leverages synthetic data to train keypoint representation for the grasping pose estimation, and Qin et, al. [42], which proposed an end-end point cloud RL framework for sim2real transfer. Another approach is to use simulation to learn policies, using environments such as Doorgym [54], which provides a simulation benchmark for door opening tasks. 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 [15, 42, 54]. However, one major drawback is that the system can only generalize to the space of assets

already present while training in the simulation. Such policies might struggle when faced with a new unseen door with physical properties, texture or shape different from the training distribution. Our approach can keep on learning via real-world samples, and hence can learn to adapt to difficulties faced when operating new unseen doors.

Chapter 3

Method

3.1 Adaptive Learning Framework

In this section, we describe our algorithmic 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 initialize our policy via behavior cloning (BC) with a diverse dataset of teleoperated demonstrations. This provides a strong prior for exploration and decreases the likelihood of executing unsafe actions. However, the initialized BC policy might not generalize to every unseen object that the robot might encounter due to the large scope of variation of objects in open-world environments. To address this, 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.

3.1.1 Action Space

For greater learning efficiency, we use a parameterized primitive action space. Concretely, we assume access to a grasping primitive $G(.)$ parameterized by g. We also have a constrained mobile-manipulation primitives $M(.)$, which primitive $M(.)$ takes two parameters, a discrete parameter C and a continuous parameter c. Trajectories are executed in an open-loop manner, a grasping primitive followed by a sequence of

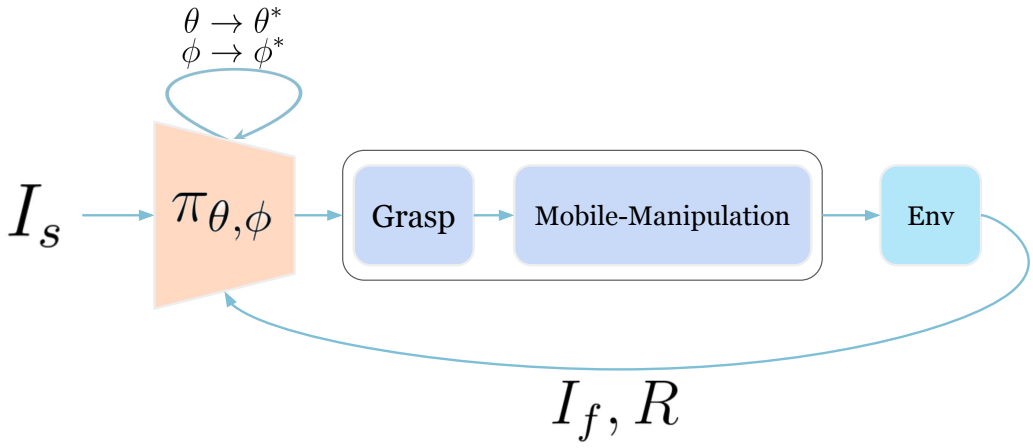

Figure 3.1: **Adaptive Learning Framework**: The policy outputs low-level parameters for the grasing primitive, and chooses a sequence of manipulation primitives and their parameters.

Algorithm 1 Adaptive Learning

Grasping primitive $G(.)$ taking parameter g 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 3.2 online RL iteration 1:N_{iter} sampling rollout 1:N_{rol} 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. 3.5, 3.4, 3.2)

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 section 4.1.2.

3.1.2 Adaptive Learning

Given an initial 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.

Imitation

We start by initializing our policy using a small set of expert demonstrations via behavior cloning. The details of this dataset are described in section 4.1.4. 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:

$$\max_{\phi,\theta} \left[\log \pi_\phi(\{C_i\}_{i=1}^N \mid I_s) + \log \pi_\theta(g, \{c_i\}_{i=1}^N \mid \{C_i\}_{i=1}^N, I_s)\right] \tag{3.1}$$

Online RL

The central challenge we face is operating new articulated objects that fall outside the behavior cloning training data distribution. To address this, we enable 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 :

$$\max_{\theta,\phi} \mathbb{E}_{\pi_{\theta,\phi}} \left[\sum_{t=0}^{T} r(s_t, a_t)\right] \tag{3.2}$$

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 [56]:

$$\nabla_{\theta,\phi} J(\theta,\phi) = \mathbb{E}_{\pi_{\theta,\phi}} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi(a_t|s_t) \cdot r_t \right] \tag{3.3}$$

$$= \mathbb{E}_{\pi_{\phi,\theta}} \left[(\nabla_\phi \log \pi_\phi(C_i|I) + \nabla_\theta \log \pi_\theta(g, c_i|C_i, I)) \cdot R \right] \tag{3.4}$$

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. Further details for online adaptation such as rewards, resets and safety are detailed in section 4.1.5.

Overall Finetuning Objective

To ensure that the policy doesn't deviate too far from the initialization of the imitation dataset, we use a weighted objective while finetuning, where the overall loss is :

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{online}} + \alpha * \mathcal{L}_{\text{offline}} \tag{3.5}$$

where loss on online sampled data is optimized via Eq.3.4 and loss on the batch of offline data is optimized via BC as in Eq.3.2. We use equal sized batches for online and offline data while performing the update.

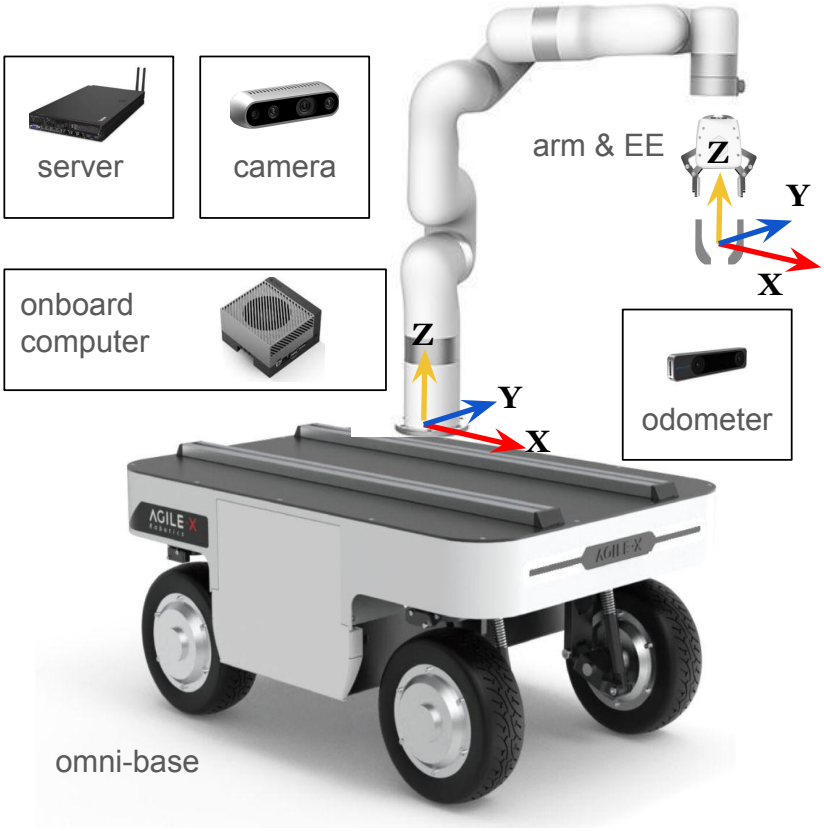

Figure 3.2: **Mobile Manipulation Hardware Platform:** We design a mobile manipulation hardware platform that is cost-effective and user-friendly using off-the-shelf components. Our mobile manipulation system consists of a base and a robotic arm. As shown in the figure, the kinematic coordination includes the base frame and the arm end-effector frame. The end-effector frame is defined relative to (i.e. with respect to) the base frame.

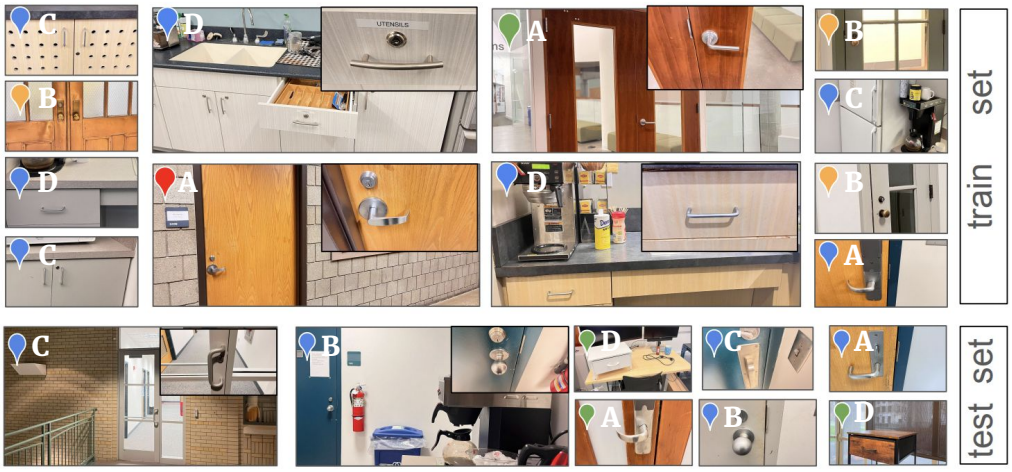

Figure 3.3: **Articulated Objects**: Visualization of the 12 training and 8 testing objects used, with type labeled, and with location indicators corresponding to the buildings in the map below. The training and testing objects are significantly different from each other, in terms of different visual appearances, different modes of articulation, or different physical parameters, e.g. weight or friction.

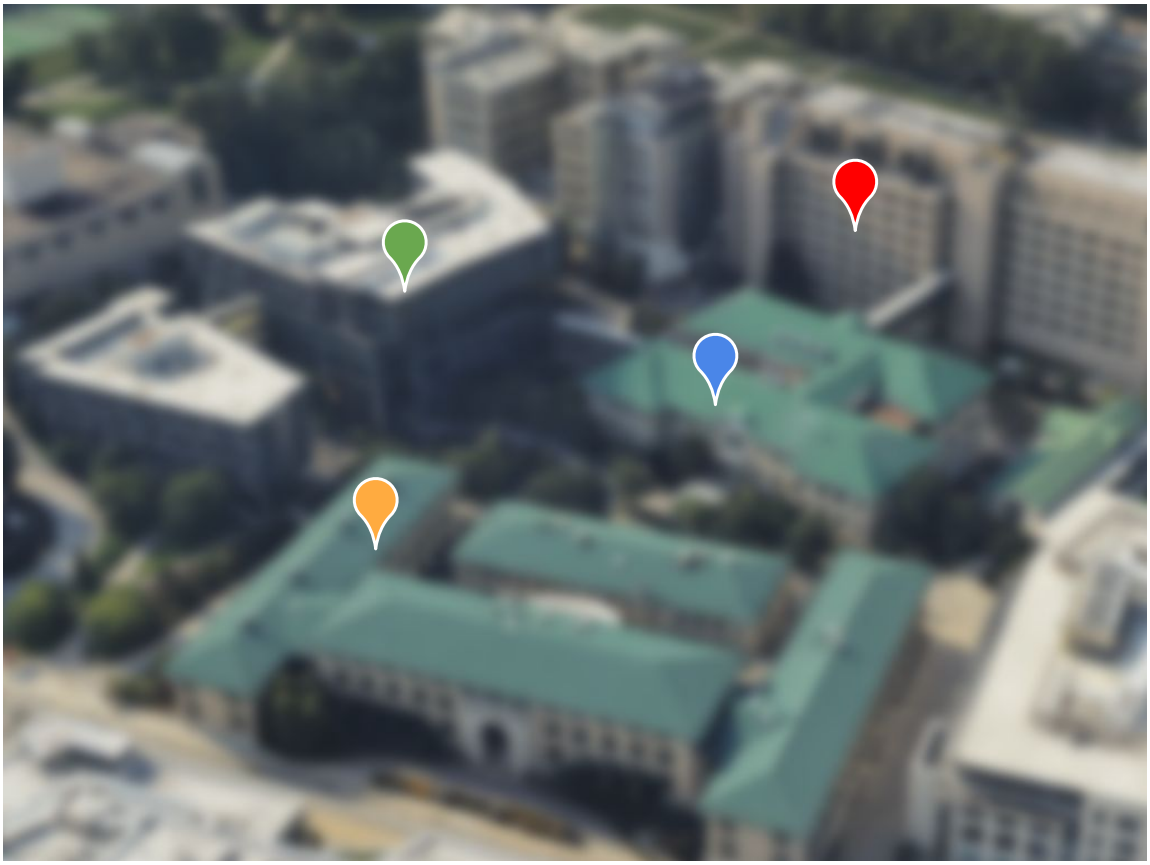

Figure 3.4: **Field Test on University Campus**: The system was evaluated on articulated objects from across four distinct buildings on the university campus.

Chapter 4

System

4.1 Open-world Mobile Manipulation Systems

In this section, we describe details of our *full-stack* approach encompassing hardware, action space for efficient learning, the demonstration dataset for initialization of the policy and crucially details of autonomous, safe execution with rewards. This enables our mobile manipulation system to adaptively learn in open-world environments, to manipulate everyday articulated objects like cabinets, drawers, refrigerators, and doors.

4.1.1 Hardware

The transition from tabletop manipulation to mobile manipulation is challenging not only from algorithmic studies but also from the perspective of hardware. In this project, we provide a simple and intuitive solution to build a mobile manipulation hardware platform. Specifically, our design addresses the following challenges -

- *versatility and agility*: Everyday articulated objects like doors have a wide degree of variation of physical properties, including weight, friction and resistance. To successfully operate these, 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.

- *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 quickly assemble the system with ease, allowing the possibility of large-scale open-world data collection in the future.

We show the different components of the hardware system in Figure 3.2. 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 accessible for research labs. The system uses a Nvidia 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 resetting the robot when performing trials for RL. 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 $25,000$ USD, making it an affordable solution for most robotics labs.

We compare key aspects of our modular platform with that of other mobile manipulation platforms in Table 4.1.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.

4.1.2 Structured Action Space with Primitives

Inspired by the recent works of efficient policy learning with manipulation primitives [38], we pre-built four expressive primitives, including grasp, unlock, rotate, open. Each primitive is a functional API that takes continuous low-level parameters as the input to instantiate an action execution. A sequential combination of these primitives effectively handles mobile manipulation tasks of a diverse set of articulated objects. We detail the implementation of our parameterized primitives in this section.

Hardware features comparison						
	Arm payload	DoF arm	omni-base	footprint	base max speed	price
Stretch RE1 [61]	1.5kg	2	53	34 cm, 33 cm	0.6 m/s	20k USD
Go1-air + WidowX 250s [13]	0.25kg	6	51	59 cm, 22 cm	2.5 m/s	10k USD
Franka + Clearpath Ridgeback [25]	3kg	7	51	96 cm, 80 cm	1.1 m/s	75k USD
Franka + Omron LD-60 [45]	3kg	7	53	70 cm, 50 cm	1.8 m/s	50k USD
Xarm-6 + Agilex Ranger mini 2 (ours)	5kg	6	51	74 cm, 50 cm	2.6 m/s	25k USD

Table 4.1: Comparison of different aspects of popular hardware systems for mobile manipulation

Grasp Primitive

At the testing time, the robot is initialized randomly in front of the objects. Given the RGBD image of the scene obtained from the realsense camera, we use off-the-shelf visual models [26, 65] to obtain the mask of the door frame using 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 perpendicularly.

We further obtain the grasp pose of the handles from the detection and segmentation models [26, 65]. As shown in Fig: 4.1, given a text prompt of "handle", the open-vocabulary detection model [65] returns a 2D bounding box of the handle. As shown in the left image of the grasp examples in Fig: 4.1, if the width of the 2D bounding box is smaller than the length of the bounding box, we determine it is a vertical handle. Otherwise, it is a horizontal handle. The grasp orientation is determined by the surface normal of the door frame, the vertical-horizontal type of the handle, and the direction of gravity. We draw a dotted middle line to find the center point of the segmentation mask of the handle, and then it is projected into 3d coordinates using camera calibration and depth. However, passive detection and segmentation models are insufficient to predict a robust grasp pose for all types of handles. For the grasp primitve, we introduce a 3-dimension continuous low-level parameter ranging from -1 to 1 as the grasp primitive input, which is then rescaled to the grasp offset ranging from $-d$ to d. This is beneficial since our residual grasp can be adapted to diverse handles via online adaptation.

Hyperparameters Table		
Hyperparameter	Symbol	Value
Mobile Manipulation Primitives Sequence	N	2
Number of Mobile Manipulation Primitives	N_p	4
Number of parameter dimension	M	3
BC Learning Rate	lr_{BC}	$1e-3$
Online Adaptation Learning Rate	lr_{Adp}	$1e-4$
Constrained mobile manipulation primitive execution Time	T	$2.5s$
Unlock velocity	v_{z}	$10cm/s$
Rotate velocity	v_{yaw}	$25 \circ /s$
Open velocity	V_{x}	$20cm/s$
Grasp offset	d	$2.5cm$
Batch size	B	16
Number of iteration during sample	N_{iter}	5
Number of rollout during sample	N_{rol}	5
Overall loss function hyperparameter	α	0.2

Table 4.2: The hyperparameters that are used in our system are listed in this table.

Constrained Mobile-Manipulation Primitives

We introduce three primitives, including unlock, rotate, and open. Each primitive is a functional API that takes a low-level parameter as the input to instantiate constrained mobile-manipulation action executions. As shown in Fig.3.2, we define two coordinate frames in the mobile manipulation system. We have a base frame, and an arm end-effector frame. The end-effector frame is defined relative to (i.e. with respect to) the base frame. With a 3-DOF motion for the base (in the SE(2) plane), and a 6-DOF arm (with respect to the base frame), we have a 9-dimensional vector -

$$(v_x, v_y, v_z, v_{\mathrm{yaw}}, v_{\mathrm{pitch}}, v_{\mathrm{roll}}, V_{\mathrm{x}}, V_{\mathrm{y}}, V_{\omega})$$

The first 6 dimensions correspond to velocity control for the arm end-effector, and the last three are the velocity control for the base. The primitives we use impose

contraints on this space as follows -

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

The velocities of these primitives are a fixed value, and the low-level parameters are continuous one-dimension values ranging from -1 to 1, which is then rescaled to the primitive execution time ranging from $-T$ to T. The sign of the low-level parameters dictates the direction of the velocity control, either clockwise or counter-clockwise for unlock and rotate, and forward or backward for open.

4.1.3 Task Definition

In this project, we consider a set of articulated objects that 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. This covers objects such as doors, cabinets, drawers, and fridges. 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 everyday articulated objects a robot system might encounter.

We define success rate as the major metric for our task. If there is a resultant gap between the door frame and base, large enough to ensure that 1) human experts can visually identify the gap 2) the robot base is able to traverse through the door by human expert teleopration, it is a success.

4.1.4 BC Pretraining

We start with Behavior Cloning (BC) pre-training to initialize the policy. We first collect an offline demonstration dataset by teleoperating the mobile manipulation robot in the open world. We type the keyboard to select the primitives and long-press the keyboard bottom to instance the low-level parameters. We include 3 objects from each category in the BC training dataset, collecting 10 demonstrations for each object, producing a total of 120 trajectories. 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. 3.3, along with which part of campus they are from as visualized in Fig. 3.4.

4.1.5 Online Adaptation

The key challenge we face is operating with new objects that fall outside the BC training domain. For example, it is extremely difficult to generalize to "push" doors if we only initialize the policy with BC on "pull" doors. To address this, we develop a system capable of fully autonomous Reinforcement Learning (RL) online adaptation. In this subsection, we demonstrate the details of the autonomy and safety of our system.

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 also provide negative reward to disincentivize such actions.

Reward Specification

In our main experiments, a human operator provides rewards- with +1 if the robot succesfully opens the doors, 0 if it fails, and -1 if there is a safety violation. Manual reward annotation is feasible since the system requires very few samples for learning. For autonomous learning however, we would like to remove the bottleneck of relying on humans to be present in the loop. We investigate using large vision language models (VLMs) as a source of reward. Specifically, we use CLIP [43] to compute the similarity score between two text prompts and the image observed after robot execution. The two prompts we use are - *"door that is closed"* and *"door that is open"*. We compute the similarity score of the final observed image and each of these prompts and assign a reward of +1 if the image is closer to the prompt indicating the door is open, and 0 in the other case. If a safety protection is triggered the reward is -1.

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 takes an image of I_f for computing reward. We then apply a random perturbation to the SE2 position of the base so that the policy learns to be more robust. Furthermore, if the reward is 1, where the door is opened, the robot has a scripted routine to close the door.

4.1.6 Model Parameterization Details

In this section, we discuss the details of the policy model parameterization and the primitive execution. We also list all the related hyperparameters in Table 4.2.

Hybrid Open-loop Policy

In our setup, We introduce a hybrid policy that incorporates a high-level policy and a low-level policy. The high-level policy takes a visual input and outputs discrete actions to determine a sequence of primitive types. The low-level policy takes a visual

input and the actions output of the high-level policy, and outputs the continuous parameters for the corresponding primitives. The discrete actions of the high-level policy and the continuous parameters of the low-level policy instantiate a sequence of primitives. The robot executes the primitives sequence in an open-loop manner. To make sure the open-loop policy output action sequence has a fixed horizon, we introduce a blank primitive in policy learning, skipping the action execution.

Network Architecture and Policy Parameterization

The high-level and low-level policy shares a frozen visual backbone, which is a ResNet-18 [19] pre-trained on ImageNet [12]. The visual backbone takes a cropped "door handle" RGB image as the input, and outputs encoded visual features.

The high-level policy takes the encoded visual features as the input and outputs a sequence of H indicates that represent the primitive types. For instance, $[0, 1, 3]$ represents [$Grasp$, $Unlock$, $Open$. In our implementation, the high-level policy head is a three-layer of multi-layer perception (MLP), it outputs action logits as a size of [B, N, H], where B is the batch size, N is the number of the primitives, and H is the horizon of the primitive sequence. The values of these hyperparameters are listed in Table: 4.2. We use a sofmax layer to get the action probabilities of the categorical distribution from the action logits. The discrete high-level actions are sampled from the categorical distribution by a simple greedy sampling.

The low-level policy head takes the encoded visual features and the sampled action of the high-level policy as the input and outputs a sequence of $H*M$, where H is the horizon of the primitive sequence, M is the low-level parameter dimension. In our setup, the grasp primitive takes a M-dimension parameter, where M is 3, While the other primitives only take a 1-dimension parameter. To allow batch tensor computations across primitives with different parameter dimensions, The low-level policy outputs a "one size fits all" distribution over the parameters. For execution, similar to Maple [38], the grasp primitive takes the M-dimension parameter, but the other primitives only take the first dimension of the M-dimension parameter. In our implementation, the low-level policy head outputs the mean and the standard deviation of Gaussian distributions for the low-level parameters of the primitives. The low-level policy head has a shared two-layer MLP, a fully connected layer of

the mean, and another fully connected layer of the standard deviation. For mean, the output of the two-layer MLP is passed through the third fully connected layer of mean, and a tanh activation function is applied. For the standard deviation, the output of the two-layer MLP is passed through the third fully connected layer of std, and a sigmoid activation function is applied. We sample low-level actions from the Gaussian distributions and clip the actions from -1 to 1.

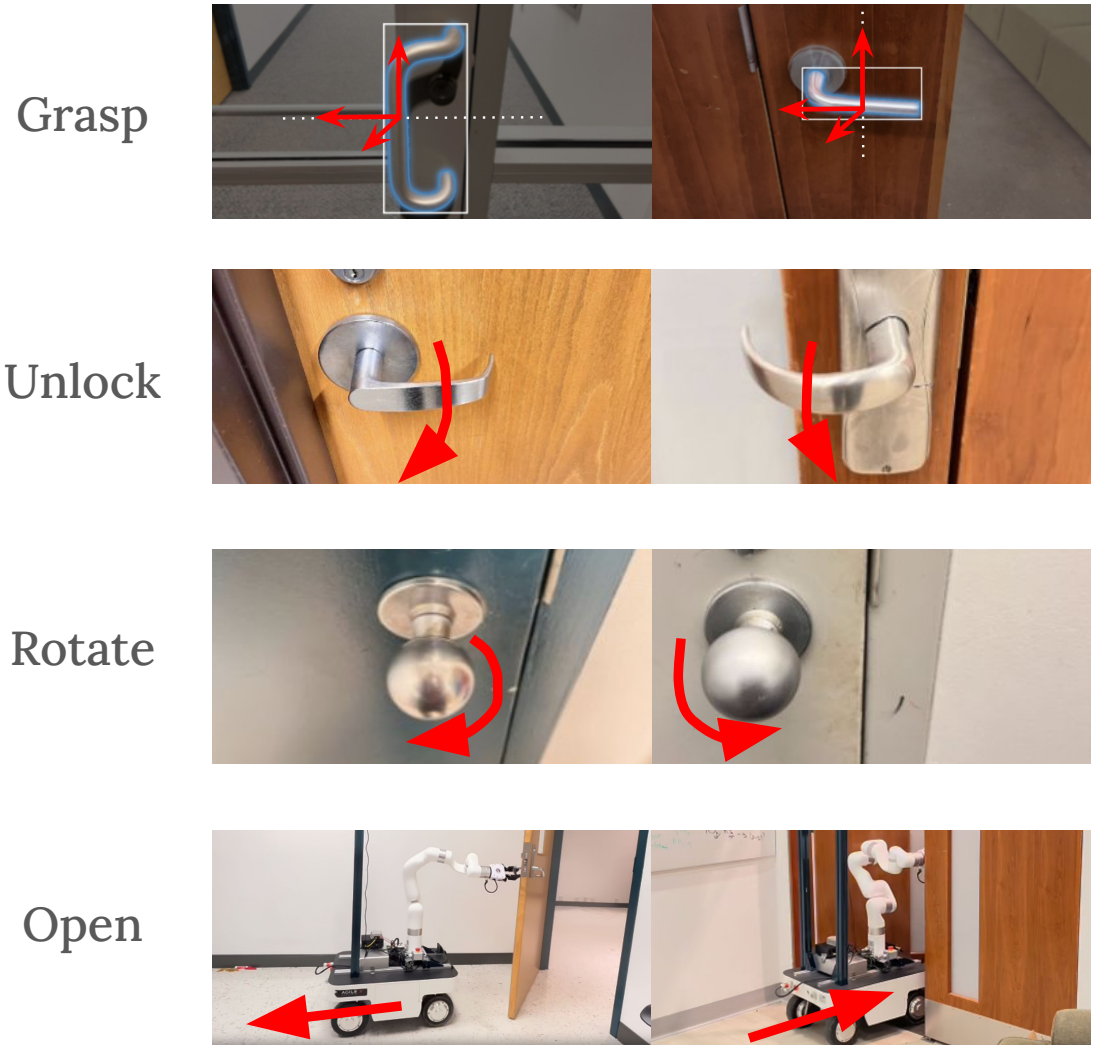

Figure 4.1: **Primitives.** We design a set of primitives to articulate a diverse set of everyday objects. Each primitive serves as a functional API that take low-level parameters to instantiate action executions.

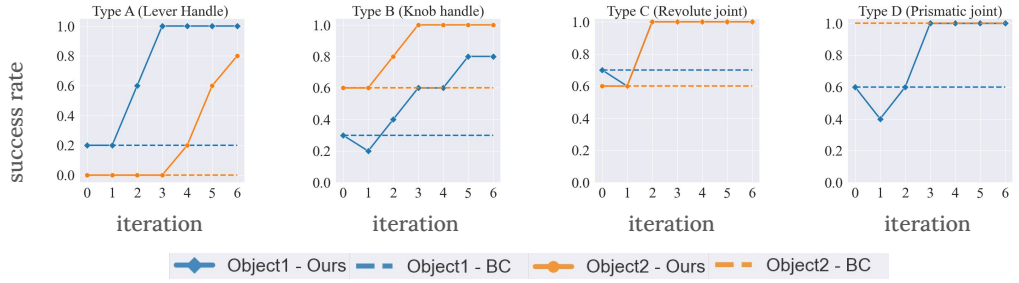

Figure 4.2: **Online Improvement**: Comparison of our approach to the imitation policy on 4 different categories of articulated objects, each consisting of two different objects. Our adaptive approach is able to improve in performance, while the imitation policy has limited generalization.

Chapter 5

Result

5.1 Results

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

1. Can the system improve performance on unseen objects via online adaptation across diverse object categories?
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?

CLIP-reward comparison			
	BC-0	Adapt-GT	Adapt-CLIP
Success Rate A1 (lever)	20%	100%	80%
Success Rate B1 (knob)	30%	80%	80%

Table 5.1: In this table, we present improvements in online adaptation with CLIP reward.

5.1.1 Online Improvement

Diverse Object Category Evaluation

: We evaluate our approach on 4 categories of held-out articulated objects. As described in section 4.1.4, these are determined by handle articulation and joint mechanisms. This categorization is based on types of handles, including levers (type A) and knobs (type B), as well as joint mechanisms including revolute (type C) and prismatic (type D) joints. We have two test objects from each category. We report continual adaptation performance in Fig. 4.2 over 5 iterations of fine-tuning using online interactions, starting from the behavior cloned initial policy. Each iteration of improvement consists of 5 policy rollouts, after which the model is updated using the loss in Equation 3.5.

From Fig. 4.2, we see that our approach improves the average success rate across all objects from 50 to 95 percent. Hence, continually learning via online interaction samples is able to overcome the limited generalization ability of the initial behavior cloned policy. The adaptive learning procedure is able to learn from trajectories that get high reward, and then change its behavior to get higher reward more often. In cases where the BC policy is reasonably performant, such as Type C and D objects with an average success rate of around 70 percent, RL is able to perfect the policy to 100 percent performance. Furthermore, RL is also able to learn how to operate objects even when the initial policy is mostly unable to perform the task. This can be seen from the Type A experiments, where the imitation learning policy has a very low success rate of only 10 percent, and completely fails to open one of the two doors. With continual practice, RL is able to achieve an average success of 90 percent. This shows that RL can explore to take actions that are potentially out of distribution from the imitation dataset, and learn from them, allowing the robot to learn how to operate novel unseen articulated objects.

Action-replay baseline

: There is also another very simple approach for utilizing a dataset of demonstrations for performing a task on a new object. This involves replaying trajectories from the closest object in the training set. This closest object can be found using k-

Action-Replay Comparison				
	KNN-open	KNN-close	BC-0	Adapt-GT
Success Rate B1 (knob)	10%	0%	30%	80%
Success Rate A2 (lever)	0%	0%	0%	80%

Table 5.2: We compare the performance of our adaptation policies and initialized BC policies with KNN baselines.

nearest neighbors with some distance metric [39]. This approach is likely to perform well especially if the distribution gap between training and test objects is small, allowing the same actions to be effective. We run this baseline for two objects that are particularly hard for behavior cloning, one each from Type A and B categories (lever and knob handles respectively). The distance metric we use to find the nearest neighbor in the training set is Euclidean distance of the the CLIP encoding of observed images. We evaluate this baseline both in an open-loop and closed-loop manner. In the former case, only the first observed image is used for comparison and the entire retrieved action sequence is executed, and in the latter we search for the closest neighbor after every step of execution and perform the corresponding action. From Table 5.2 we see that this approach is quite ineffective, further underscoring the distribution gap between the training and test objects in our experiments.

Autonomous reward via VLMs

We investigate whether we can replace the human operator with an automated procedure to provide rewards. The reward is given by computing the similarity score between the observed image at the end of execution, and two text prompts, one of which indicate that the door is open, and the other that says the doors is closed, as described in section 4.1.5.

As with the action-replay baseline, we evaluate this on two test doors, on each from the handle and knob categories. From Table 5.1, we see that online adaptation with VLM reward achieves a similar performance as using ground-truth human-labeled reward, with an average of 80 percent compared to 90 percent. We also report the performance after every iteration of training in Fig. 5.1. Removing the need for

a human operator to be present in the learning loop opens up the possiblity for autonomous training and improvement.

5.1.2 Hardware Teleop Strength

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

Table 5.3: Human expert teleoperation success rate using stretch and our system for opening doors

In order to successfully operate various doors the robot needs to be strong enough to open and move through them. 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 5.3, 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.

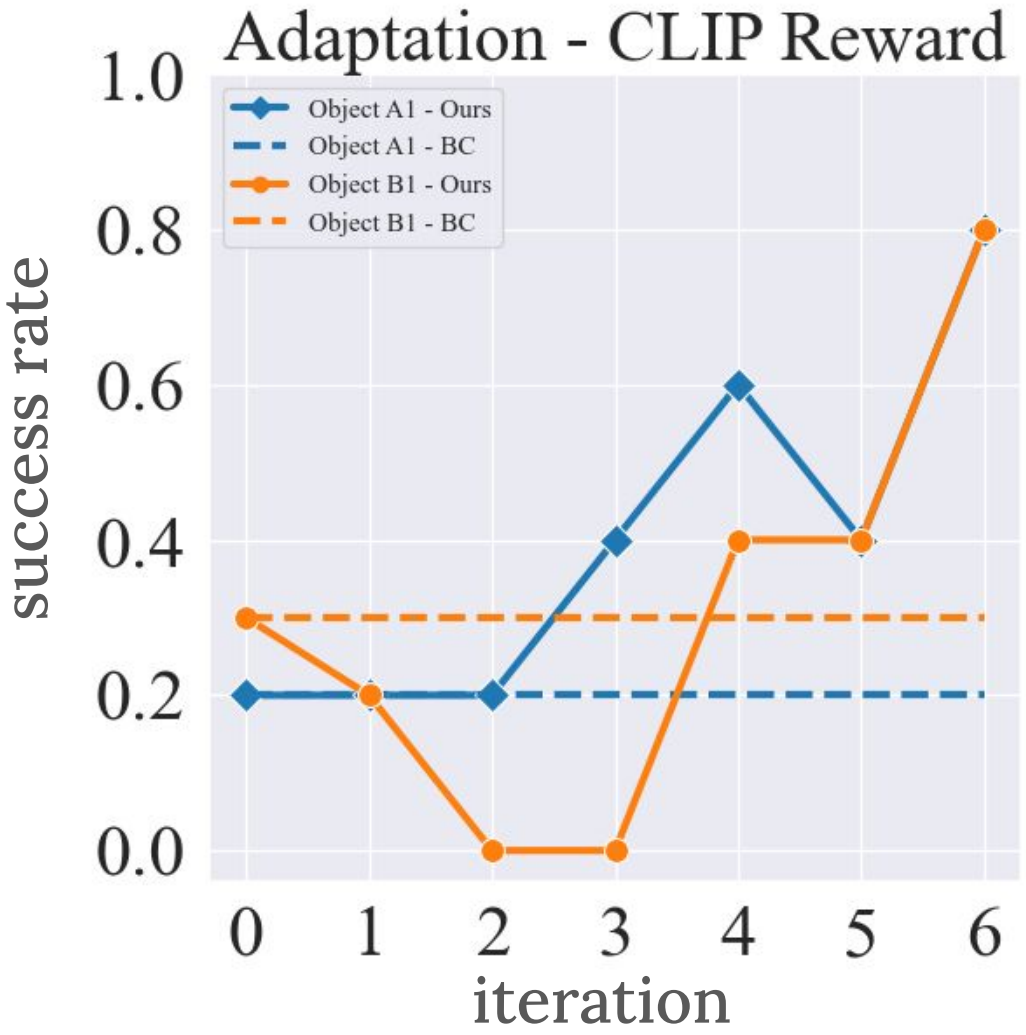

Figure 5.1: **Online Adaptation with CLIP reward.** Adaptive learning using rewards from CLIP, instead of a human operator, showing our system can operate autonomously.

Chapter 6

Conclusions

We present a full-stack system for adaptive learning in open world environments to operate various articulated objects, such as doors, fridges, cabinets and drawers. The system is able to learn from very few online samples since it uses a highly structured action space, which consists of a parametric grasp primitive, followed by a sequence of parametric constrained mobile manipulation primitives. The exploration space is further structured via a demonstration dataset on some training objects. Our approach is able to improve performance from about 50 to 95 percent across 8 unseen objects from 4 different object categories, selected from buildings across the university campus. The system can also learn using rewards from VLMs without human intervention, allowing for autonomous learning. We hope to deploy such mobile manipulators to continuously learn a broader variety of tasks via repeated practice.

Bibliography

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2.2

[2] Christopher G Atkeson, PW Babu Benzun, Nandan Banerjee, Dmitry Berenson, Christoper P Bove, Xiongyi Cui, Mathew DeDonato, Ruixiang Du, Siyuan Feng, Perry Franklin, et al. What happened at the darpa robotics challenge finals. *The DARPA robotics challenge finals: Humanoid robots to the rescue*, pages 667–684, 2018. 1, 2.3

[3] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *RSS*, 2022. 1

[4] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 2.1

[5] Nandan Banerjee, Xianchao Long, Ruixiang Du, Felipe Polido, Siyuan Feng, Christopher G Atkeson, Michael Gennert, and Taskin Padir. Human-supervised control of the atlas humanoid robot for traversing doors. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 722–729. IEEE, 2015. 1, 2.3

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2.2

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath,

Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. 1

[8] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 1

[9] Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects. *The International Journal of Robotics Research*, page 02783649231201201, 2022. 2.1

[10] Sachin Chitta, Benjamin Cohen, and Maxim Likhachev. Planning for autonomous door opening with a mobile manipulator. In *2010 IEEE International Conference on Robotics and Automation*, pages 1799–1806. IEEE, 2010. 1, 2.3

[11] Mathew DeDonato, Felipe Polido, Kevin Knoedler, Benzun PW Babu, Nandan Banerjee, Christoper P Bove, Xiongyi Cui, Ruixiang Du, Perry Franklin, Joshua P Graff, et al. Team wpi-cmu: achieving reliable humanoid behavior in the darpa robotics challenge. *Journal of Field Robotics*, 34(2):381–399, 2017. 1, 2.3

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. 2.1

[13] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023. 2.2, **??**

[14] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *arXiv*, 2024. 1, 2.2

[15] Arjun Gupta, Max E Shepherd, and Saurabh Gupta. Predicting motion plans for articulating everyday objects. *arXiv preprint arXiv:2303.01484*, 2023. 2.3

[16] Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and match: Supercharging imitation with regularized optimal transport. *CoRL*, 2022. 2.1

[17] Siddhant Haldar, Jyothish Pari, Anant Rai, and Lerrel Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *arXiv preprint arXiv:2303.01497*, 2023. 2.1

[18] haoyu Xiong, Haoyuan Fu, Jieyi Zhang, Chen Bao, Qiang Zhang, Yongxi Huang, Wenqiang Xu, Animesh Garg, and Cewu Lu. Robotube: Learning household manipulation from human videos with simulated twin environments. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=VD0nXUG5Qk. 2.1

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2.1

[20] Alexander Herzog, Kanishka Rao, Karol Hausman, Yao Lu, Paul Wohlhart, Mengyuan Yan, Jessica Lin, Montserrat Gonzalez Arenas, Ted Xiao, Daniel Kappler, et al. Deep rl at scale: Sorting waste in office buildings with a fleet of mobile manipulators. *arXiv preprint arXiv:2305.03270*, 2023. 1, 2.1

[21] Advait Jain and Charles C Kemp. Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator. Georgia Institute of Technology, 2008. 1, 2.3

[22] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018. 2.1

[23] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021. 2.1

[24] Aditya Kannan, Kenneth Shaw, Shikhar Bahl, Pragna Mannam, and Deepak Pathak. Deft: Dexterous fine-tuning for real-world hand policies. *CoRL*, 2023. 2.1

[25] Julien Kindle, Fadri Furrer, Tonci Novkovic, Jen Jen Chung, Roland Siegwart, and Juan Nieto. Whole-body control of a mobile manipulator using end-to-end reinforcement learning, 2020. **??**

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 4.1.2

[27] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021. 1

[28] Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022. 2.1

[29] Sergey Levine and Vladlen Koltun. Guided policy search. In *ICML*, 2013. 2.1

[30] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 2016. 2.1

[31] Jacky Liang, Saumya Saxena, and Oliver Kroemer. Learning active task-oriented exploration policies for bridging the sim-to-real gap. *arXiv preprint arXiv:2006.01952*, 2020. 2.1

[32] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024. 1

[33] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Alan: Autonomously exploring robotic agents in the real world. In *ICRA*, 2023. 2.1

[34] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. 2023. 2.1

[35] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1647–1654. IEEE, 2022. 2.2

[36] Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2):911–918, 2021. 2.1

[37] K. Nagatani and S.I. 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*, volume 2, pages 45–50 vol.2, 1995. doi: 10.1109/IROS.1995.526137. 1, 2.3

[38] Soroush Nasiriany, Huihan Liu, and Yuke Zhu. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 4.1.2, 2.1

[39] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021. 5.1.1

[40] 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)*, volume 3, pages 2333–2338 vol.3, 2000. doi: 10.1109/IROS.2000.895316. 1, 2.3

[41] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning.

arXiv preprint arXiv:1903.03698, 2019. 2.1

[42] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Conference on Robot Learning*, pages 594–605. PMLR, 2023. 2.3

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4.1.5

[44] Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019. 2.1

[45] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *arXiv preprint arXiv:2307.06135*, 2023. **??**

[46] Allen Z Ren, Hongkai Dai, Benjamin Burchfiel, and Anirudha Majumdar. Adaptsim: Task-driven simulation adaptation for sim-to-real transfer. *arXiv preprint arXiv:2302.04903*, 2023. 2.1

[47] R.B. Rusu, W. Meeussen, Sachin Chitta, and Michael Beetz. Laser-based perception for door and handle identification. pages 1 – 8, 07 2009. 2.3

[48] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 2.2

[49] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023. 1

[50] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation, 2023. 1

[51] Laura Smith, J. Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world, 2021. 2.1

[52] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen

Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022. 2.2

[53] Charles Sun, Jedrzej Orbik, Coline Devin, Brian Yang, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully autonomous real-world reinforcement learning with applications to mobile manipulation, 2021. 1

[54] Yusuke Urakami, Alec Hodgkinson, Casey Carlin, Randall Leu, Luca Rigazio, and Pieter Abbeel. Doorgym: A scalable door opening environment and baseline agent. *arXiv preprint arXiv:1908.01887*, 2019. 2.3

[55] Jiayu Wang, Shize Lin, Chuxiong Hu, Yu Zhu, and Limin Zhu. Learning semantic keypoint representations for door opening manipulation. *IEEE Robotics and Automation Letters*, 5(4):6980–6987, 2020. doi: 10.1109/LRA.2020.3026963. 2.3

[56] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 3.1.2

[57] Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and Roberto Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation. In *Conference on Robot Learning*, pages 1367–1378. PMLR, 2022. 2.2

[58] Bohan Wu, Roberto Martin-Martin, and Li Fei-Fei. M-ember: Tackling long-horizon mobile manipulation via factorized domain transfer. *arXiv preprint arXiv:2305.13567*, 2023. 2.2

[59] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *arXiv preprint arXiv:2305.05658*, 2023. 2.2

[60] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834, 2021. doi: 10.1109/IROS51168.2021.9636080. 2.1

[61] Ruihan Yang, Yejin Kim, Aniruddha Kembhavi, Xiaolong Wang, and Kiana Ehsani. Harmonic mobile manipulation, 2023. 1, **??**

[62] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*, 2023. 1, 2.2

[63] Naoki Yokoyama, Alexander William Clegg, Eric Undersander, Sehoon Ha,

Dhruv Batra, and Akshara Rai. Adaptive skill coordination for robotic mobile manipulation. *arXiv preprint arXiv:2304.00410*, 2023. 2.2

[64] Yizhou Zhao, Qiaozi Gao, Liang Qiu, Govind Thattai, and Gaurav S Sukhatme. Opend: A benchmark for language-driven door and drawer opening. *arXiv preprint arXiv:2212.05211*, 2022. 2.2

[65] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan 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*, pages 350–368. Springer, 2022. 4.1.2