# Kitchen Robot Case Studies: Learning Manipulation Tasks from Human Video Demonstrations

Dingkun Guo

CMU-RI-TR-23-87

December 20

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

**Thesis Committee:**
Professor Christopher G. Atkeson, *co-chair*
Professor Jeffrey Ichnowski, *co-chair*
Professor David Held
Jianren Wang

*Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.*

*For a better world.*

# Abstract

The vision of integrating a robot into the kitchen, capable of acting as a chef, remains a sought-after goal in robotics. Current robotic systems, mostly programmed for specific tasks, fall short in versatility and adaptability to a diverse culinary environment. While significant progress has been made in robot learning, with advancements in behavior cloning, reinforcement learning, and recent strides in diffusion policies and transformers, the challenge remains to develop a robot that matches human capabilities in learning and generalizing across tasks, particularly in complex, unstructured real-world scenarios.

In the thesis, I focus on enabling robots to learn manipulation tasks from a single human demonstration, with predefined primitives that are generalizable across similar objects and environments. We developed a system that can process RGBD video demonstrations to identify task-relevant key poses and frames using Segment Anything. We then addressed challenges for robots replicating human actions, such as collision and robot configuration limitations. To validate the effectiveness of our approach, we conducted experiments focusing on manual dishwashing. With one human demonstration in a lab kitchen, the method was tested under varied conditions in a standard home kitchen, differing in geometry and appearance from the learning environment.

Further, we broaden the scope of learning to more generalized data sources, particularly focusing on videos from unstructured environments like YouTube. By enabling the use of unseen videos as a source for specific robot learning tasks, we translated visual elements into physical constraints and goals in simulation, inferring physics of the tasks. We demonstrated the transferability of this learning methods to real-world scenarios with actual robots, on tasks including fruit cutting, dough manipulation, and pouring liquids.

# Acknowledgments

My heart is filled with immense gratitude for the extraordinary support and guidance I have received from a remarkable group of individuals. Each has played a unique and important role in my academic and personal growth, shaping the journey of my Master's degree at CMU.

Foremost, I extend my deepest gratitude to my parents. Their unwavering and limitless support throughout my life has been the bedrock of my achievements. Their emotional and financial backing, which enabled me to pursue my dreams in robotics, coupled with their encouragement during challenging times, has been a beacon of motivation and strength, inspiring me to persevere and strive for excellence.

I am greatly thankful to Professor Chris Atkeson for his invaluable mentorship. His approach of granting me the freedom to explore and experiment in my research taught me how to define research problems, a skill more important than solving the problem itself. He provided a wealth of inspiring and critical ideas that have significantly shaped my thinking. The opportunity to have exclusive access to a Franka robot arm in his lab has been a pivotal element in my learning and development.

My other advisor, Professor Jeff Ichnowski, has been a foundational pillar of strength, endowing me with the confidence to successfully complete my thesis. His patience and expert guidance through technical challenges have been instrumental in my evolution as a researcher.

I am also thankful to my committee member Professor David Held. His instruction in the 'Deep Reinforcement Learning' course and his insights about robot learning have deeply enriched my understanding and approach to the field.

Special thanks to Professor Zackory Erickson, who offered me the chance to teach during the summer and to create assignments for his course, 'Mechanics of Manipulation.' This experience laid the foundation for my understanding of robot manipulation, an area that has become central to my academic pursuits.

Jianren Wang, as a committee member and coauthor, has been indispensable to the success of my projects. His support during challenging phases, with his comprehensive knowledge of both technical details and the broader theoretical aspects of robotics, has been invaluable.

My labmate Arkadeep Narayan Chaudhury has provided valuable insights into the world of robotics and effective collaboration with advisors. The positive and collaborative atmosphere he fostered in the lab has greatly enhanced my research experience.

Finally, I am immensely grateful to my friends at CMU, including Ziqi Wang, Yulun Zhang, Yuxin Guo, Jianzhe Gu, Zilin Si, Gaoyue Zhou, Yufei Wang (and his dog Mars), Yuanzhui, and many others. They have made this journey not only academically fulfilling but also personally enriching. Their friendship, support, and shared moments have been a source of joy and motivation.

This thesis stands as a testament to the collective support, wisdom, and encouragement I received from each of these individuals. Their contributions to my journey have been invaluable, and I am forever grateful.

# 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

The aspiration of integrating a robot into the kitchen to assist in meal preparation has long been a dream in robotics. The landscape of culinary robotics has seen notable developments, with inventions like Miso Robotics' Flippy robot [1] adept at flipping hamburgers, Hyper-Robotics' pizza-making robot [3], and the salad-preparing robot named Sally [2]. Additionally, the Moley kitchen [4] presents a prototype that adapts an entire kitchen for robotic cooking. However, these innovations, while groundbreaking, exhibit limitations in their intelligence and versatility. They are typically programmed for specific, repetitive tasks, lacking the adaptability required for a diverse culinary repertoire and are not designed to learn new recipes or adapt to varying kitchen environments.

I envision a robot that not only assists in the kitchen but acts as a culinary connoisseur, capable of recommending and preparing a variety of dishes based on the available ingredients at home. It would be able to simultaneously cook multiple dishes, learn new recipes, and adapt to personal taste preferences, akin to having a 3-star Michelin chef in one's own kitchen. The concept extends beyond mere automated machines; it is about creating a robot that learns and adapts, similar to how humans learn cooking through video tutorials and textual instructions. By combining these learning modalities, the goal is to develop a generalized robot that can **acquire and refine skills from video demonstrations**, thereby significantly enhancing its utility and intelligence in a domestic setting. Our research focuses on Learning from Demonstrations (LfD), aiming to imbue robots with the capability to master and perform complex manipulation tasks.

Admittedly, the field of robot learning has witnessed significant advancements. Traditional behavior cloning [74], for instance, optimizes policies to replicate actions from demonstration data through extensive network engineering; visual behavior cloning [89] has shown the capability to learn simple tasks in unseen environments by converting tasks into specific rewards aligned with vision representations; reinforcement learning [79] has been successfully applied to generate solutions for kitchen tasks within simulated environments. Moreover, recent developments in Diffusion Policies [23] and Transformers [77] have introduced more efficient ways of learning from multi-modal data, eliminating the need for designing specific primitives.

However, achieving the level of robotic proficiency where it can learn from demonstrations and generalize learning across tasks to match human capabilities remains an unfulfilled goal. Current advancements have not yet bridged the gap in fully addressing the complexity and variability inherent in real-world diverse and unstructured scenarios.

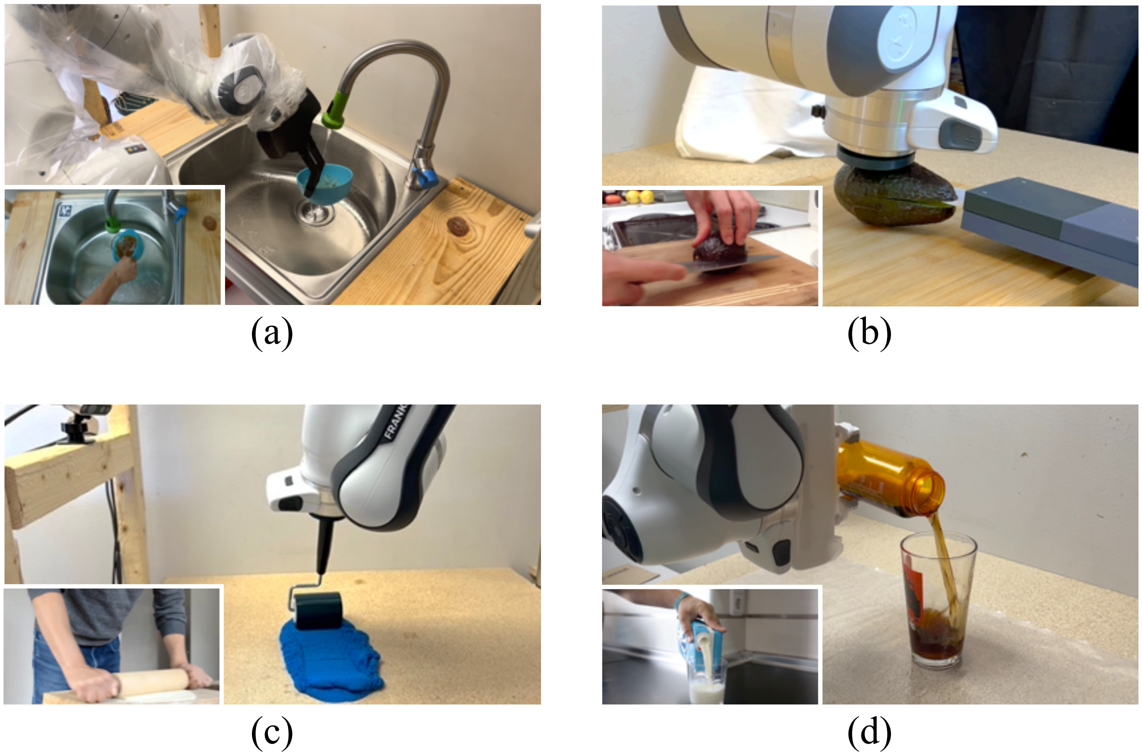
The first significant challenge is the difficulty in acquiring and annotating scalable, reusable robot data. Methods like behavior cloning and diffusion often struggle to find consistent visual elements across different demonstrations, and the scarcity of alignable demonstrations for specific tasks further complicates this issue. The variability in how individuals perform tasks poses a significant hurdle for methods not requiring explicit reward shaping [5, 37], as it hinders the convergence of these methods amidst such diversity. While self-collected data and residual learning approaches [102] offer some solutions, they demand considerable system-specific engineering, which may not be transferable to other tasks. Hence, an ideal approach would be to enable robots to learn effectively from single or a few demonstrations, reducing dependence on large, varied datasets and simplifying the learning process.

Secondly, it is hard to learn without demonstrations or directly within simulation. When training in simulation, environmental collisions and dynamics can vary significantly from real-world settings, making the sim2real transfer nearly impossible without highly accurate physics simulations and detailed models of objects and environments. Moreover, reinforcement learning in this context requires the laborious definition of rewards, which can be a painstaking and intricate process. Training a robot in a new, real-world environment often requires extensive retraining and potential adjustments in the reward function parameters. This lack of adaptability and the need for prolonged training periods in new settings pose significant hurdles to the development of versatile and efficient robotic systems that can seamlessly transition from simulated to real environments.

Thirdly, manual definition of primitives is laborious and not generalizable. Typically, each task demands a unique set of primitives, which leads to a lack of generalizability and adaptability in different environments. While there have been efforts to learn primitives based on contact relationships [92], these approaches primarily focus on the making of contact between objects, not applying to long-horizon tasks that involve more complex and varied interactions over extended periods. This limitation restricts the scope of tasks that robots can learn and execute, particularly in dynamic and unstructured environments where the requirements can change significantly over the course of a task.

The aim of this research is to enable robots to learn manipulation tasks from a single human demonstration, with predefined primitives that are generalizable across various objects and environments. The initial focus, in Chapter 2, is on learning object poses from demonstrations collected in controlled environments, with manual dishwashing as a representative example. Then in Chapter 3 we extend the work to learn physics from less controlled environments such as YouTube videos.

In Chapter 2, we developed a system that can identify, learn, and execute manipulation tasks from a single RGBD human demonstration. This involves processing video demonstrations to identify task-relevant key poses and frames using Grounded Segment Anything [35]. To bridge the gap between human demonstrations and robot execution through generalizable manipulation, we address challenges for robots in replicating human actions, such as collision avoidance, different kinematics, and joint range limitations, to enable more nuanced and precise robotic tasks. To validate the effectiveness of our approach, we conducted experiments focusing on the complex manipulation task of manual dishwashing. Initially demonstrated in a lab kitchen, the method was tested under varied conditions in a standard home kitchen, differing in geometry, appearance, and physics from the learning environment. Employing a Franka Research 3 Robot, we successfully replicated dishwashing, as shown in Figure 1.1(a), demonstrating our method's adaptability and potential in real-world applications. This evaluation results underscore our approach's robustness, emphasiz-



(a) (b) (c) (d)

Figure 1.1: Robot Performing Experiment Tasks with Human Demonstrations in corners. (a) Washing dishes, (b) cutting an avocado, (c) rolling dough, (d) pouring liquid.

ing its ability to efficiently rely on contact information and learn from a single human demonstration, with generalizablity to similar objects and environments.

In Chapter 3, we broaden the scope of learning to more generalized data sources, particularly focusing on videos from unstructured environments such as YouTube videos. We translated YouTube videos into physical constraints and goals (represented as scene graphs) for simulations, aiding in the learning and optimization of robot manipulation tasks in diverse environments. We also demonstrated the transferability of tasks learned in simulation to real-world scenarios with experiments on the Franka and xArm robots, encompassing tasks like slicing fruits, pouring, and dough manipulation, shown in Figure 1.1(b)–(d). Each task, learned from a single YouTube video, represents a significant advancement in robot learning from human video demonstrations, paving the way for future scalability in this domain.

# Chapter 2

# Robot Manual Dishwashing: Learning from RGBD Human Demonstration Videos

## 2.1 Introduction

Recent advancements in robot learning have significantly expanded the range of tasks that robots can perform. Despite numerous breakthroughs in the field, robots have yet to match human proficiency in performing simple household tasks like cooking, dishwashing, and laundry. These tasks remain challenging, primarily due to limitations in computer vision and robot learning capabilities. This work extends these advancements to encompass multi-step tasks with enhanced generalizability. Our research seeks to address this challenge, focusing initially on one of the most fundamental kitchen tasks: manual-style dishwashing.

The need for robots to wash dishes by hands, despite the existence of dishwashing machines, stems from several practical considerations. First, many items, such as bamboo cutting boards and fine china, are unsuitable for machine washing and require hand washing. Additionally, pre-rinsing dishes before placing them in a dishwasher is a common practice, underscoring the need for robotic assistance in these tasks. Moreover, dishwashing represents a challenging multi-step manipulation task for

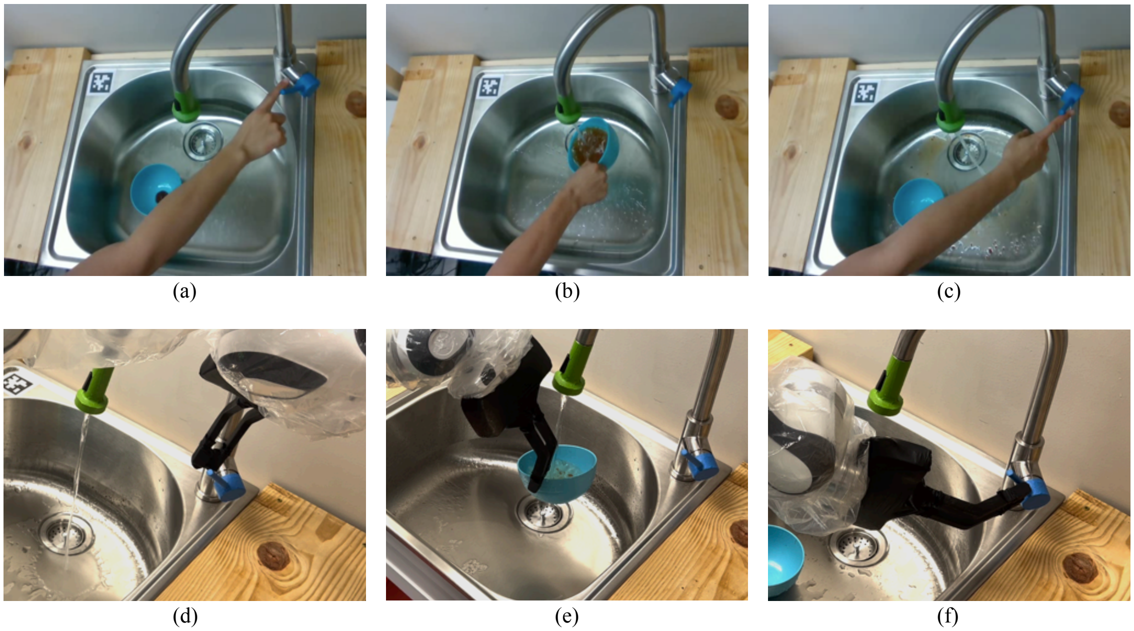


Figure 2.1: Comparative Overview of the Manual Dishwashing Task. The top row (a-c) depicts human demonstrations and the bottom row (d-f) shows robot executions. Key actions include (a) and (d) interacting with the faucet to turn on the water, (b) and (e) holding a bowl under the running water, and (c) and (f) turning off the water. This side-by-side comparison highlights the task's sequential nature and the robot's ability to mirror human actions in a real-world setting.

research, involving the handling of liquids and objects with high variance in shapes and sizes.

Our approach to robot learning from human demonstrations revolves around a generalizable method for manipulation tasks. The core concept is to enable robots to autonomously interpret human demonstrations by segmenting the task into modular, learnable primitives that can be executed by robots. This process demands an advanced vision system capable of identifying objects in videos, video temporal segmentation, and understanding of 3D spaces. Additionally, it necessitates task understanding alongside precise robot motion planning and control, particularly critical in contact-rich environments.

A pivotal aspect of object manipulation in robotics is the understanding of contacts, centering on hand-object and object-object relationships. Based on this principle, we have developed methods to segment manipulation tasks by their contact states: initiating contact, move while maintaining contact, and breaking contact. For instance, as illustrated in Figure 2.1, these contact phases are matched in both the human demonstrations and robot executions of a manual dishwashing task. The robot is trained to estimate and replicate contact states observed in the demonstrations, thereby replicating some intentions (the contacts) of the human actions.

Our system integrates three modules: vision, learning, and manipulation. The vision module initiates the process by analyzing videos of human demonstrations, constructing point clouds, and estimating object poses. This visual information is then fed into the learning module, where action templates are employed to recognize and generalize the human intentions behind the demonstrated actions. The final step involves the manipulation module, which maps the templates into robot programs, enabling robots to replicate the observed tasks in varied settings and with different objects. Essentially, our system is designed to interpret, learn, and execute tasks in diverse scenarios based on a single human demonstration.

The main contributions are:

1. Developed a system that can analyze, learn, and execute tasks from a single human demonstration. This involves processing video demonstrations to identify task-relevant key poses and frames using Ground Segment Anything [35].
2. By estimating and replicating human intent, we enable robots to perform a demonstrated task differently from the human demonstration when the robot cannot exactly copy human motion due to differences in kinematics, limb shapes, joint ranges, and strength.
3. Implementation and evaluation of these methodologies on real-world robots, providing practical evidence of their effectiveness and applicability.

Our method significantly enhances an agent's ability to learn from a single human video demonstration. Initially demonstrated in a kitchen mockup in the lab, the method was tested under varied conditions with a Franka Research 3 Robot in a standard home kitchen, differing in geometry, appearance, and physics from the training environment.

## 2.2 Related Work

### 2.2.1 Learning from Demonstration

Extensive research has been done on Learning from Demonstration (LfD) [8, 16, 75], , where a robot acquires manipulation behaviors given demonstrations of experts performing the task. Many works seek to learn a policy. A popular algorithm in this field is behavior cloning [74], which optimizes a policy to generate actions that match the demonstration data. While conceptually simple, these systems often require significant neural network engineering (e.g. transformer architectures [21, 27], multi-modal prediction heads [65, 77], etc.) and/or human-in-the-loop data collection algorithms [42, 74] to work in practice. Human supervision typically involves human supervision through methods like teleoperation (using a joystick or VR interface) [10, 103, 105] or kinesthetic teaching [18, 57], where a user physically guides the robot arm. However, collecting demonstrations with these approaches can be laborious and time-consuming.

Recent developments have explored alternative methods for providing human demonstrations, such as retargeting hand-pose estimation to robot end effector [11, 73, 81] and training policies directly from first and third-person human demonstrations [14, 82, 89]. These approaches aim to replicate human actions in robots, ignoring human-robot differences in tasks beyond robot capability with human motion. Another route emphasizes objects in demonstrations [92, 106, 107], but these approaches do not apply to complex multi-step tasks that involve contact-rich interactions over extended periods. Addressing forementioned limitations, our system centers on the contact dynamics between human hand and objects, as well as between objects themselves. Our work enhances generalizability and data efficiency, empowering robots to learn multi-step manipulation tasks effectively from a single demonstration.

### 2.2.2 Object Pose Estimation

We rely on the recovery of object poses to provide robots with an initial understanding of the scene. We broadly categorized methodologies into four types: Point Pair Features based methods [30, 69, 86], Template Matching [29, 38], Learning-based approaches [39, 83, 88, 93], and methods utilizing 3D Local Features [36]. Other works seek to enhance pose estimation accuracy using multiple views [19, 55, 80]. However, these all require additional training for adaptation to unseen data and suffer from limited generalizability to objects and scenarios deviating from their training data.

Recent advancements propose using Transformers [6, 41, 104, 108] and Neural Radiance Fields (NeRF) [58, 100] for pose estimation enhancement. Ongoing developments regularly reported in the Benchmark for 6D Object Pose Estimation [84]. Yet, applying these methods to customized environments remains challenging because they are specifically designed for standard datasets and benchmarks, and are not tested for generalization to data that is similar to, but different from, the training environment.

Addressing these limitations, we instead combine identifying object pixels through 2D image segmentation with the fundamental template matching technique, Iterative Closest Point (ICP) [22], for determining object pose. We employ Grounded Segment Anything [35] for object detection and segmentation from images with text prompts. This approach offers generalizability, functioning effectively in diverse scenarios without additional training. Moreover, this part of our system is modular, allowing for replacement as more advanced technique emerge, thus ensuring adaptability and future-proofing.

### 2.2.3 Temporal Video Segmentation

Temporal video segmentation is essential in our work for segmenting videos into meaningful robot actions. There are many datasets in this domain, some incorporating spatio-temporal annotations and object relations [43], which mainly focus on bounding boxes. Seminal video datasets [70, 91, 97] offer pixel labels over time, yet these are typically short-term and lack fine-grained action labels. The Epic-Kitchen dataset [26], collected using an ego-centric camera, is particularly relevant to our research. It has been the basis for numerous video segmentation tasks, including Hand Object Segmentation [25, 78], which aims to identify contact relationships between hands and objects in images. This dataset also supports experiments in tasks including action recognition [17, 71, 87, 99] and action detection [31, 59]. However, these methods generally do not surpass 60 % in accuracy.

To tackle this challenge, we opted for an approach using videos with depth information. By leveraging accurate image segmentation, we reconstruct object point clouds and calculate contact relationships to segment actions. We identify contacts by thresholding the minimum distance between objects. This approach eliminates the dependency on a particular dataset and provides accurate segmentation points for robots to understand and mimic human actions.

## 2.3 Problem Formulation

The problem is a human demonstrating a manipulation task to the robot system once, which the system then learns to execute. The input is a video of the human demonstration, and the system should recognize and execute corresponding robot primitives. These primitives need to be adaptable to various objects and environments, differing from the learning scenario, and executable by different robot arms when provided with task-specific parameters. Therefore, the goal is to develop a function $g$ that maps each frame $f_i$ in a video $V$ to a robot primitive $p$ in the set of primitives $P$. Formally,

$$g : F \times O \to P,$$

where $F$ represents the set of frames in the video, and $O$ denotes the set of task-relevant objects.

The set $P$ could be pre-defined and needs to span the range of $A$, all possible actions observable in human demonstrations. A fundamental feature of these primitives is their clear demarcation, designed to avoid overlap, as the robot executes one primitive at one time. In this case, each action $a$ maps to a unique primitive $p$ through a function $\phi$, leading to the following definition of $P$:

$$P = \{p \mid p = \phi(a), \forall a \in A\}.$$

We make three assumptions. (1) All task-relevant objects and environment are visible in at least one frame of the video. (2) The objects can be treated as rigid, even if not rigid in nature. (3) The scenario involves only one instance of each object type, eliminating the complexity of object tracking.

## 2.4 Methods

The system has vision (Section 2.4.1), learning (Section 2.4.2), and manipulation (Section 2.4.3) modules to achieve its adaptive capabilities. The vision module processes video from human demonstrations, forming point clouds and detecting object poses. The learning module then uses this information, coupled with predefined primitives, to understand and generalize the intent behind the demonstrated actions. The manipulation module takes robot policy and controls the robot, allowing it to replicate the observed task in varied environments and with similar objects. In essence, from just a single human demonstration, our system identifies, learns, and performs manipulation tasks across different objects and environments.

### 2.4.1 Vision Module

The vision module serves as a bridge between raw visual input and actionable data for other modules of the system. It processes RGBD images captured from *(a) Multi-Camera System* into a *(b) Point Cloud Registration*. With *(c) Object Segmentation*, the vision module tags each point of the point cloud to a corresponding object or environment, and subsequently transforms the points into meshes, as described in *(d) Model Construction*. These meshes serve as models in *(e) Object Pose Estimation* and collision detection in the manipulation module.

#### (a) Multi-Camera System

We employed eight RealSense D435 depth cameras, chosen for their precision and compatibility with our requirements. Calibration is the first step to enable accurate depth perception and spatial understanding. To achieve this, we utilized Multical [12], paired with a customized April Tag [67] board, as shown in Figure 2.3(b) to perform the calibration process. Throughout the human demonstration phase, all eight cameras capture video at a resolution of 640 × 480 pixels and a rate of 30 fps.

It's important to note that while our current setup provides satisfactory results, there remains potential for improvements. The accuracy of the camera's depth perception, coupled with the calibration process, offers avenues for further refinement, allowing even more precise and consistent data capture in future iterations.

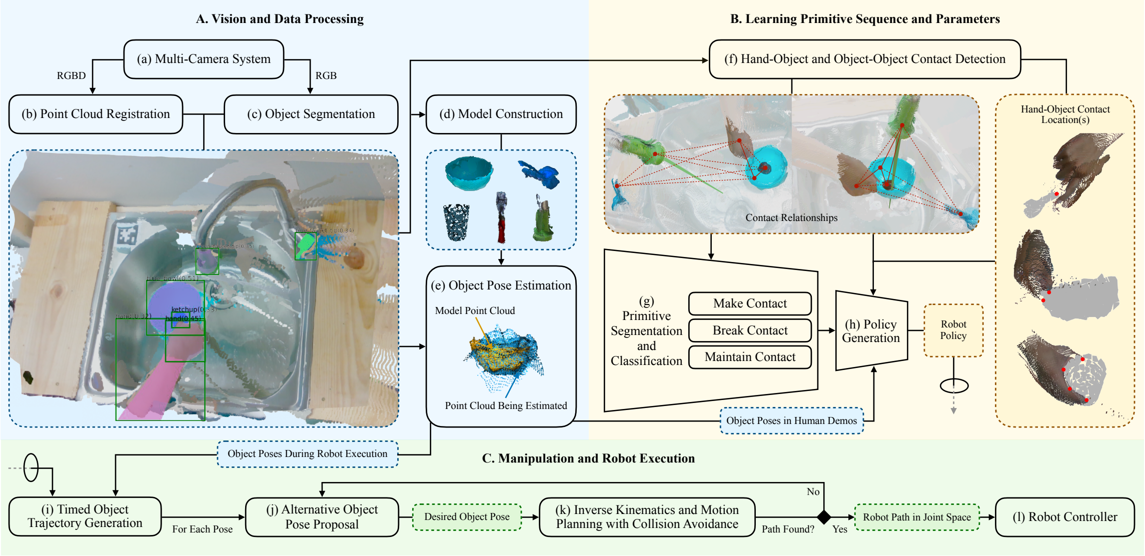

A. Vision and Data Processing
(a) Multi-Camera System
RGBD
RGB
(b) Point Cloud Registration
(c) Object Segmentation
(d) Model Construction
(e) Object Pose Estimation
Model Point Cloud
Point Cloud Being Estimated
B. Learning Primitive Sequence and Parameters
(f) Hand-Object and Object-Object Contact Detection
Contact Relationships
Hand-Object Contact Location(s)
(g) Primitive Segmentation and Classification
Make Contact
Break Contact
Maintain Contact
(h) Policy Generation
Robot Policy
Object Poses in Human Demos
Object Poses During Robot Execution
C. Manipulation and Robot Execution
(i) Timed Object Trajectory Generation
For Each Pose
(j) Alternative Object Pose Proposal
Desired Object Pose
(k) Inverse Kinematics and Motion Planning with Collision Avoidance
Path Found?
No
Yes
Robot Path in Joint Space
(l) Robot Controller

Figure 2.2: System Overview. In the Module A. Vision and Data Processing, we start with (a) eight pre-calibrated RGBD cameras capturing human video demonstrations, whose outputs collectively generate one (b) point cloud. Each color frame undergoes processing by (c) Grounded Segment Anything. Then We use the segmented point clouds to (d) construct object models, serving as templates for (e) object pose estimation using ICP. In the Module B. Learning Primitive Sequence and Parameters, a (f) contact detector analyzes the segmented point clouds to determine contact relationships and locations. This information helps to (g) classify primitives and (h) generate policy parameters for robot execution. The object poses derived earlier are also necessary in formulating effective robot policies. The Module C. Manipulation and Robot Execution execute primitives on a robot by providing execution parameters in the testing environment.

### (b) Point Cloud Registration

Harnessing the depth and color information from the recordings, coupled with calibrated intrinsics and extrinsics, enables the construction of point clouds. These point clouds amalgamate data from all eight cameras, providing a comprehensive and cohesive spatial representation. Given the issues that arise when synthesizing data from multiple RGBD images, we chose point clouds as the medium for representing this fused information because they are convenient for later processing, such as mesh reconstruction and distance calculation. The voxel grid could be an alternative method worth exploring further. Additionally, to establish a consistent reference

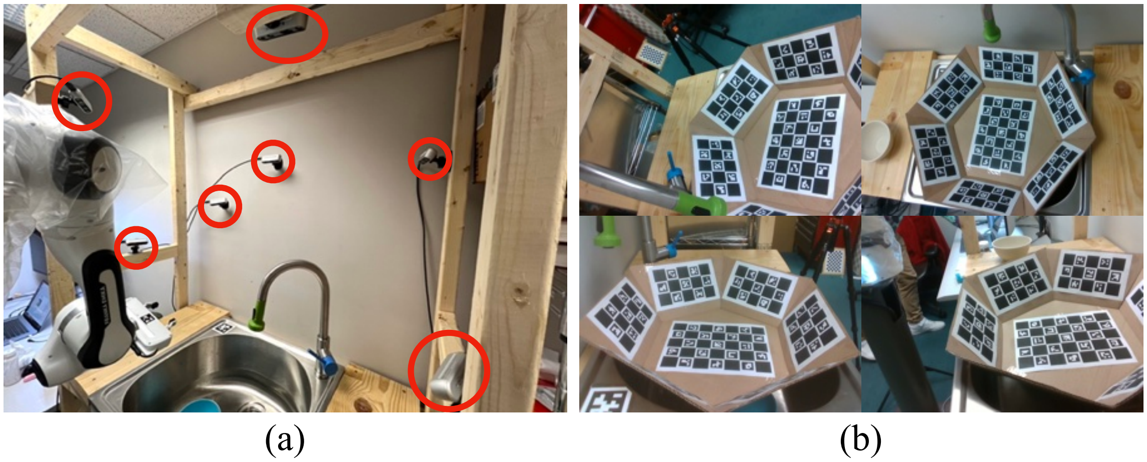


Figure 2.3: Camera Setup and Calibration. Eight RealSense D435 depth cameras, seven of them shown in (a), strategically positioned around the sink area, capturing different angles and perspectives for comprehensive depth perception. Calibration board is shown in (b), used for determining the camera intrinsics and extrinsics.

framework, the origin of the point cloud is set at a specific fixed camera, ensuring standardization and ease of interpretation across different data sets.

### (c) Object Segmentation

We conduct image segmentation on captured color images, providing a mask for each object of interest. To achieve this segmentation, we use an off-the-shelf method called Grounded Segment Anything [35], a combination of Grounding DINO [62] and Segment Anything [49]. This method detects objects using text prompts and produces detailed object masks. Notably, during the recording of human demonstrations, we define the objects of interest through text prompts for detection.

### (d) Model Construction

To capture the environment's point cloud, we focus on the initial frames of the demonstration when the scene is devoid of any objects. We refine the point clouds of both the environment and objects by removing outliners that are further away from their neighbors in average. These refined point clouds serve as models for pose estimation.

Next, we transform the point clouds into meshes through the Poisson surface reconstruction method [46]. This method retains points in point clouds as the vertices of a resulting triangle mesh. We then decompose the reconstructed surfaces into convex components, using V-HACD technique [85]. These meshes, less noisy and more computational efficient than original point clouds, serve as models for collision detection, enabling the safe and efficient operation of the system in real-world scenarios.

#### (e) Object Pose Estimation

Estimating the pose of an object is a part of understanding its spatial context. For each frame in recorded video, we compute relative poses between the segmented point cloud and the model point cloud of the object, using Iterative Closest Point (ICP) [22]. ICP, as it iteratively refines the alignment of two point clouds, is effective in this context, where capture point clouds represent similar objects under similar conditions. By minimizing the distance between corresponding points in these point clouds, ICP can provide accurate object pose estimation.

During robot execution, the vision module detects the presence of objects and discerns their poses. When there is a need to determine an object's pose, all cameras synchronously capture RGBD images, and we determine the pose using the same procedure as in processing demonstration videos. The manipulation module uses this pose as a bridge to align the execution reality with human demonstrations.

### 2.4.2 Learning Module

The primary goal of the learning module is to convert human demonstrations into robot primitives, emphasizing the understanding and recovery of contacts in manipulation tasks. In this process, we calculate distances between object point clouds for *(f) Hand-Object and Object-Object Contact Detection*, and we use changes in contact relationships to guide *(g) Primitive Segmentation and Classification*. By combining object poses from human demonstrations with contact information, *(h) Policy Generation* formulates a robot policy for execution in the manipulation module.

### (f) Hand-Object and Object-Object Contact Detection:

We assess the distances between two point clouds to determine their contact relationship. To minimize noisy fluctuations in contact change, especially when the distance is close to the threshold, we use two distinct thresholds in a sequential test. The threshold for breaking contact is higher than that for making contact. Additionally, we conduct this test in reverse order. If the contact results vary when evaluated forward and in reverse, we choose the outcome that leads to the fewest contact changes.

Moreover, we record the contact location(s) between the hand and the object it holds, which could be useful in generating a robot policy for making contact with the object. In the reference frame of the object's model point cloud, we cluster points that are close to the hand, and the center of each cluster represents a contact location.

### (g) Primitive Segmentation and Classification

We compartmentalized human actions into three generalizable robot primitives, each rooted in the nature of hand-object contact relationships:

- **Make Contact**: This represents the initial point of interaction between the hand and the object. For instance, it correlates to the initial touching of an object to grasp it or push it.
- **Break Contact**: This denotes the cessation of hand-object interaction. Typically, this corresponds to letting go of an object.
- **Maintain Contact**: This primitive encompasses actions where the hand maintains contact with an object over a prolonged period, manipulating it in various ways. An example is rinsing a bowl while continuously holding it.

We posit that we can distill most human manipulation actions into these three foundational categories, under the assumption that all objects are rigid bodies. Figure 2.6 presents an example of breaking down the task of washing a bowl.

The boundaries between primitives can sometimes be blurred. For example, the action of switching a faucet on or off may involve making and breaking contact with faucet within a few frames. However, we generate robot policy relying on contact changes, not on these blurred boundaries.

It's worth noting that while our demonstrations do feature deformable objects like water and ketchup, we've opted to bypass any consideration of changes in their shape for the scope of this work. Instead, our focus is their presence or absence. In Chapter 3, we explore the manipulaiton of soft or deformable objects. In that exploration, we're assessing the manipulation of such objects based on attribute changes such as shapes and friction.

#### (h) Policy Generation

The robot policy for a task comprises a sequence of robot primitives along with their learned parameters. The sequences of primitive mirror those in human demonstrations, and each kind of primitive has different learned parameters.

Making contact requires hand-object contact location(s), with the goal of having the robot make contact with the object similarly to human demo. If a human makes contact with an object at more locations than the two fingers of a robot can accommodate, as illustrated by holding a cup in Figure 2.2 (f), we manually define applicable contact locations for the object.

Breaking contact requires the object's final pose at the moment it breaks contact with the hand. We can override this with a manually defined object pose in cases where we do not want the robot to place the object at a location similar to that in the human demonstration.

While the hand maintains contact with an object, changes in object-object contact relationships and the object's state, such as its appearance or disappearance, are crucial. The learned parameters include object poses and contact relationships captured in the demonstration whenever these changes occur. The aim is to have the robot replicate the relative object pose with the same contact relationships.

### 2.4.3 Manipulation Module

The manipulation module executes robot primitives by providing execution parameters in the testing environment. It generates a timed object trajectory and convert the trajectory into desired joint angle commands. Getting a robot to make the object move along the desired object trajectory raises challenges, including different kinematics, collisions, and many constraints imposed by the robot's configuration.

### (i) Timed Object Trajectory Generation

The robot primitives operate by taking inputs derived from demonstrations as well as real-time object and environment statuses and churning out a trajectory detailing key poses that guide how the object should be manipulated. This streamlines the transition from human demonstration to robot execution, enabling robots to replicate human's actions.

We represent this translation through a "timed object pose trajectory", a sequence of object poses each tagged with a timestamp. This trajectory is object-centric, with poses being relative, focusing on the spatial and temporal evolution of the object's pose over the course of a demonstration. The trajectory enables the generalization and replication of actions observed in demonstrations across various scenarios, regardless of specific robots or environmental conditions.

### (j) Alternative Object Pose Proposal

Humans and robots are different in their anatomical and mechanical configurations. This divergence means that a robot cannot merely copy human demonstrations exactly. For instance, a motion or pose of a human wrist might be inaccessible or impractical for a robot's end-effector due to its structural constraints. Our system adjusts target object poses when confronted with configurations the robot cannot feasibly achieve.

For a pose that has no solution from IK or motion planner, our system proposes a new alternative desired object pose that are both logical and maintain the functional intent of the original pose. Central to this strategy is the consideration of an object's inherent symmetries and an emphasis on preserving the relative spatial relationships among interacting objects. Take an example of holding a bowl. While the position of the bowl is invariant, the orientation of the bowl can be flexible about its symmetric axis. In this case, the program can suggest rotating an object about its symmetric axis, while remaining contact relationships between objects unchanged. Such an adjustment not only retains the functional intent but also provides the robot with an expanded range of configurations to explore, especially in complex environments like around a sink.

### (k) Inverse Kinematics and Motion Planning with Collision Avoidance

Robot movement needs the amalgamation of precise inverse kinematics and adept motion planning, particularly when the imperative is to navigate in cluttered or unpredictable environments. Many inverse kinematics algorithms lack integrated collision avoidance capabilities. To this end, we've adopted a two-pronged approach: employing the TRAC-IK solver [13] for inverse kinematics, complemented by the RRT-Connect [51] for motion planning.

The IK solver produces a set of potential solutions for a desired robot end-effector pose. Solutions causing robot collision are discarded. The remaining solutions are designated as the preferred robot joint angle set. Following this, the RRT motion planner, informed by the robot's current and desired joint angles, generates a collision-free path. This path's viability is ascertained using a collision detection function in PyBullet [24], which, given the robot's joint angles and the environmental mesh data, determines whether the robot is in collision.

### (l) Robot Controller

For controlling the Franka robot, a NUC equipped with a real-time kernel is employed to interface with the robot's control box, enabling high-frequency control important for precise task execution. Command signals are transmitted from an Ubuntu computer to the NUC via Ethernet. The controller implementation and pipeline is provided by Polymetis [61].

The primary basic control employed is Proportional-Derivative (PD) joint position control. In the PyBullet simulation phase, a robot joint path is calculated based on desired object poses. This path is then smoothed and sampled at a high frequency of 1000 Hz using the method described in [54]. This helps the robot's capability to adhere to the command path, although it is found less critical in actual testing scenarios. However, one issue encountered is the ”path deviation” error, which occurs randomly at specific joint angles. The cause of this error remains elusive. Our current workaround involves restarting the robot controller and resuming motion from the point of interruption whenever this error triggers a stop.

## 2.5 Experiments and Results

We present a series of experiments and evaluations to test the robot system's effectiveness. Initial experiments are conducted in a laboratory kitchen mockup, where the foundational aspects of the system, including key pose identification and collision avoidance, are tested. The experiments are then extended to a standard home kitchen, offering a more challenging and varied environment to test the adaptability of the system in real-world scenarios.

### 2.5.1 Environment and Robot Setup

To prepare environments and the robot for experiments, we reconstructed kitchen settings in the PyBullet simulator, created a waterproof skin for the Franka Research 3 robot, and customized the robot's gripper to adapt to the specific requirements of the dishwashing task which included reaching into the sink, which had limited space. Additionally, the experiments also required a control system implemented for precise task execution, enabling the adaptability of our setup to various robot platforms.

#### Learning and Testing Environment

Human demonstrations were initially recorded in a lab kitchen mockup, shown in Figure 2.4(a). This setting facilitated focused and safe testing, particularly in terms of managing water usage during robot experiments. Subsequently, the system was moved to an actual home kitchen, also illustrated in Figure 2.4(b). In this new environment, key elements such as the sink, faucet, and camera positions differed from the lab. Despite these changes, the learned primitives demonstrated adaptability to different environments, thereby validating the potential for generalization inherent in our system.

Both the lab and home kitchen environments were modeled in the PyBullet simulator. The process involved creating detailed meshes from the point clouds captured by the multi-camera system, as described in Section 2.4.1. The resulting simulation, which accurately mirrors the real-world lab kitchen, is shown in Figure 2.4(c).

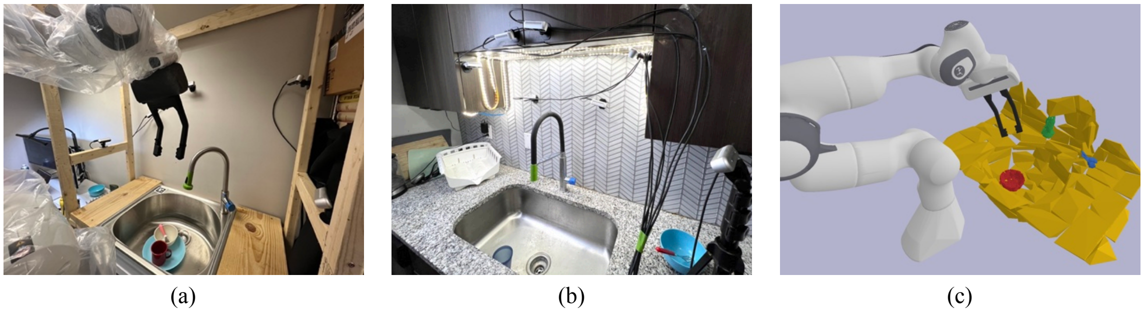


Figure 2.4: Learning and testing environment. Human Demonstrations are recorded in (a) the lab kitchen, and the robot can perform the task at a previously unseen (b) home kitchen. The (c) reconstructed lab kitchen in simulation is used for motion planning.

### Robot

For experiments, we utilized the Franka Research 3 robot arm equipped with the Franka Hand as the end-effector, augmented with a custom 3D printed finger. The original fingers of the Franka Hand were too short, and the hand and wrist dimensions were too large for dishwashing, as depicted in Figure 2.5(a). In the original configuration, when holding objects like bowls or cups under the faucet, the robot inadvertently obstructed the water flow. Therefore, we engineered a longer finger with a 135-degree angle, shown in Figure 2.5(b), specifically designed to position the robot's wrist away from the manipulated object, allowing unblocked water flow while washing dishes.

In the experiment, the presence of real water requires the waterproofing of the robot, given the inherent sensitivity of electronic components to moisture. Figure 2.5 (c) illustrates the robot arm was encased in poly tubing, while the gripper was sheathed in a rubber glove. This design is practical and cost-effective, as selected materials not only maintain the robot's operational flexibility and dexterity but also offers ease of replacement and maintenance, ensuring the robot's longevity in wet environments.

## 2.5.2 Results

Experiments were designed to test the adaptability of our system across a variety of scenarios, including interactions with both seen and unseen objects, their placement

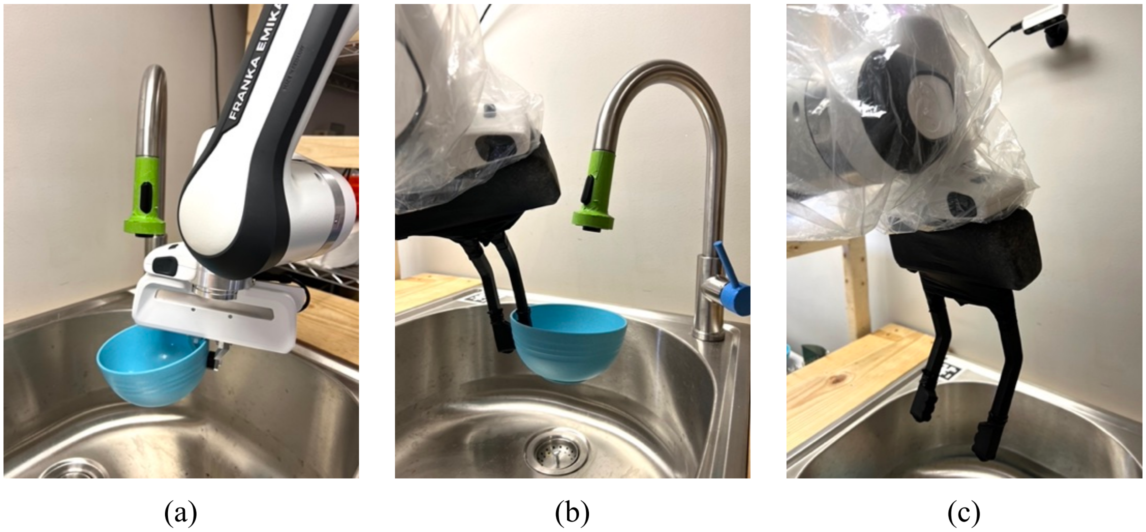


Figure 2.5: Robot Finger and Waterproof Design. The (a) original Franka Hand fingers are so short that the robot blocks the water. We solved this by designing (b) longer fingers. (c) The robot is waterproofed with poly tubing on the robot arm and glove on gripper.

locations, and different environments. The results, detailed in Figure 2.6 and Table 2.1, illustrate how the system performed in different settings. The seen environment is the lab kitchen, used for human demonstrations, while the unseen environment was the home kitchen. The range of objects tested included bowls, cups, and forks. In the demonstrations, a blue bowl with ketchup was used, and this knowledge sucessfully generalized to similar objects, such as a red bowl with mustard and a plastic white bowl. Cups were varied in their design; a blue cup without a handle was shown to the system, whereas a red cup with a handle is the unseen variation. Similarly, the system's ability to generalize from a fork to a differently shaped spoon was assessed.

Initially, the experiments were conducted in a laboratory setting, where the objects were placed in the same locations as demonstrated by a human. In this controlled environment, the system achieved an approximate success rate at least 80 % for each subtask. However, the errors in individual subtasks accumulated over time. Consequently, when the robot attempted to complete the overall task which had the five components listed, the overall success rate decreased to around 40 %.

When the experiment's conditions were altered by moving the objects to different locations, the system's performance declined, with the success rate of each

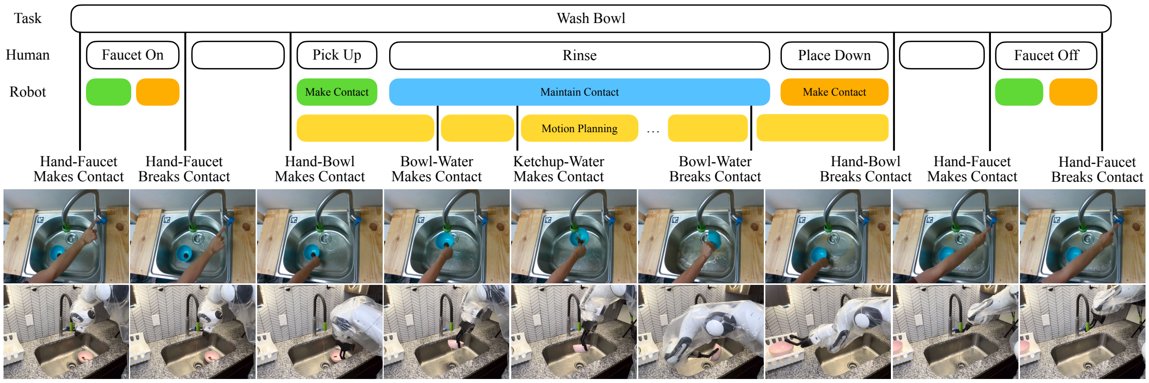


Figure 2.6: Overview of the Task of Washing A Bowl. We break down human demonstrations into robot primitives (top row), conduct experiments in the lab kitchen mockup (middle row), and test the system in a home kitchen (bottom row).

Table 2.1: Success rate of dishwashing task tested on **S**een and **U**nseen **Obj**ects, **Loc**ations, and **Env**ironments.

| Obj | Loc | Env | Faucet On/Off | | Pick | Place | Rinse | Whole Sequence |
|---|---|---|---|---|---|---|---|---|
| S | S | S | 0.9 | 0.8 | 0.8 | 1.0 | 0.8 | 0.4 |
| U | S | S | | | 0.8 | 1.0 | 0.4 | 0.2 |
| S | U | S | 0.6 | 0.5 | 0.6 | 0.8 | 0.8 | 0.2 |
| U | U | S | | | 0.4 | 0.8 | 0.6 | 0.2 |
| S | U | U | 0.5 | 0.5 | 0.8 | 0.8 | 0.8 | 0.4 |
| U | U | U | | | 0.6 | 1.0 | 0.6 | 0.2 |

subtask dropping to about 50-80 %. This reduction in effectiveness reduced the robot's ability to successfully complete a sequence of five tasks. The experiment was further extended to a home kitchen environment, posing additional challenges for the system, particularly for the vision module. In this setting, the system struggled with accurately determining the pose of the faucet lever, which was crucial for the task of turning the faucet on and off. Despite the challenges, the system demonstrated some generalizability, achieving a success rate of 20 % across all tasks.

The analysis of failure cases in the experiments highlights several key areas of concern. Pose estimation inaccuracies were the most significant issue, accounting for 50 % of the failures, indicating challenges for the vision system's ability to precisely locate and orient objects. Motion planning issues, where the robot couldn't find a feasible path to complete a task, contributed to 26 % of the failures. In 10 % of cases, the robot failed to rinse off the ketchup adequately. The robot dropping the bowl when it became too heavy with water accounted for 8 % of failures, suggesting a need for better grasping and handling strategies. The faucet being only half-opened resulted in 3 % of failures and other miscellaneous issues comprised the remaining 3 %, indicating a range of smaller, yet significant, challenges that need addressing to improve the overall system's reliability and efficiency.

## 2.6 Limitations and Future Works

This section outlines the challenges encountered in the areas of vision, learning, and manipulation within the current system. These challenges highlight key areas for improvement and guide the direction for future work to enhance the system's overall performance and adaptability.

### 2.6.1 Vision

The vision module of our system, while functional, faces several limitations, predominantly due to the inherent challenges in point cloud generation and object pose estimation. Notably, the point clouds derived from the RealSense depth cameras exhibit significant noise. This is primarily because these cameras rely on stereo vision features to estimate depth, which can be inaccurate, especially around the edges of objects. Calibration and global registration of point clouds also present areas for improvement. These inaccuracies tend to accumulate and are particularly evident when dealing with small objects, such as the tip of a faucet lever, which has a diameter of less than 1 cm. Incorrect pose estimation can lead to the robot missing the intended object, resulting in operational failures like the inability to turn on a faucet.

A promising future direction is the adoption of more advanced depth cameras, such as the Azure Kinect, which employs Amplitude Modulated Continuous Wave (AMCW) Time-of-Flight (ToF) technology. However, a challenge with using multiple such cameras is the potential interference due to the emitted light. An alternative approach might involve using multiple RGB cameras, although this would necessitate highly accurate calibration of camera intrinsics and extrinsics. Advances in Neural

Radiance Fields (NeRF) offer the potential to reconstruct point clouds from RGB cameras without needing explicit calibration.

The current inaccuracies in the vision module also adversely affect the calculation of contact relationships, leading to over or under-segmentation of primitives. This can have catastrophic consequences for the system's operation. Currently, we manually correct the sequence of primitives, but a more automated and accurate method would be necessary for scaling up the system. Additionally, the current system lacks tracking capabilities; it operates under the assumption that all task-relevant objects are visible in at least one frame and that there is only one instance of each object type.

For future improvements, one avenue could be integrating the vision system with Large Language Models (LLMs) to enhance the identification of task-relevant objects and their sequence. Another promising approach is to develop Diffusion policies that bypass state estimation, combining task and motion planning with visual feedback for a more integrated and robust system. Such advancements could significantly enhance the precision and scalability of our vision module, contributing to more efficient and reliable robotic operations.

### 2.6.2 Learning Generalizability

The generalizability of our current learning setup is primarily oriented towards replicating the same sequences of primitives as observed in human demonstrations. While the modular nature of these primitives theoretically allows for adaptation to different sequences, this flexibility has not been fully explored or utilized. This limitation becomes particularly apparent in complex scenarios where the environment or sequence of primitives might change unexpectedly, such as the sudden appearance or disappearance of objects. Under such circumstances, the learned primitives may not adequately address the challenge, for example, when the target object is obstructed by unforeseen objects.

Regarding dynamic changes in task execution, several potential solutions emerge. 1) One approach could involve integrating new human demonstrations to directly instruct the robot in handling novel scenarios. 2) Reinforcement learning within simulations could offer a means to teach the robot to adapt to a broader range of situations. 3) Expanding the dataset to include more varied demonstrations, each presenting different solutions to the same problem, could also enhance the system's adaptability. 4) Applying learning from analogy or transfer learning might enable the system to draw parallels from different tasks and apply this knowledge to new, unencountered scenarios.

One intuitive method, for example, is to sample using k-nearest neighbor to generate each key pose within this trajectory. Specifically, it samples a key pose that is proximate in characteristic to the corresponding key poses observed in the demonstrations. This methodology bears resemblance to the construction of token matrices in linguistic research. In such matrices, the prediction of a subsequent word relies on its historical likelihood given the context of preceding words.

Another significant limitation lies in our approach to soft objects, which are currently treated as rigid. This treatment can lead to difficulties in segmentation and action identification, especially for actions involving soft object manipulation. Addressing this challenge requires much more accurate simulation and a deeper understanding of the interaction dynamics with soft materials. One area of potential improvement, as discussed in Chapter 3 of this thesis, involves exploring new methods and sensory inputs to better grasp the mechanics of soft objects. This advancement would necessitate a more nuanced approach to force and impedance estimation, going beyond the capabilities of pure vision-based systems.

### 2.6.3 Grasping and Motion Planning

The concept of making contact in robotic object manipulation, which is commonly referred as robot grasping problem, is a vast and highly active area of research. In our work, the 'make contact' primitive is somewhat limited, focusing primarily on one-contact pushing and two-contact grasping. However, human hands demonstrate far greater dexterity, capable of producing numerous contact points and modes, especially in soft contacts. Our current modes are unable to encompass all these variances, leading to situations where specific actions, such as picking up a plate using the sink's wall, must be manually defined. These non-prehensile strategies present a future research direction, leading to more complex and varied grasping techniques.

Hardware limitations of the robot also pose challenges. For instance, the current gripper may struggle to hold a bowl full of water due to its weight, leading to assumptions like the absence of drift between the end effector and the object. The robot's speed further restricts the range of interactions with objects and water. While hardware improvements are one solution, advancing the robot's intelligence is equally crucial. Techniques like residual learning could address issues like drift during object holding, enhancing the robot's capability to handle complex dynamics changes using current trajectories as references.

Motion planning presents another area of limitation, particularly in terms of inefficiency and limited success rates. Our current setup can generalize to some extent, adapting to scenarios where human demonstrations might not be directly replicable by the robot. This is achieved by altering the desired object pose without changing the contact relationship requirements. More intelligent methods could be developed to bridge the gap between human capabilities and robot configuration limits.

Moreover, humans possess a remarkable ability to adapt to new environments and tasks, developing a kind of 'muscle memory' that allows for efficient task execution without extensive conscious thought. Exploring the potential of implementing a similar strategy in robotics could lead to faster motion planning and more efficient task execution. Such an approach would enable robots to perform complex tasks more autonomously, such as driving and typing, resembling human-like adaptability and learning efficiency.

To briefly conclude this chapter, we have explored the complexities of robotic learning from human demonstrations, focusing on the nuances of long-horizon tasks like dishwashing. Our primary contribution is identifying weak points in current approaches. Another contribution is developing a system that segments these tasks into modular primitives, based on rigid body object pose, has shown promising results in both lab and home environments for a robot to understand and replicate human actions. While our methods showed promising adaptability and generalizability, they also highlighted certain limitations, particularly in vision accuracy and efficient task learning and execution. Addressing these challenges, our future work aims to refine the system's adaptability and dexterity, exploring more advanced computer vision and more intelligent robot learning techniques. In the next chapter, we extend our work beyond relying solely on rigid body object pose to exploring the learning of force and impedance from human demonstrations using simulation, pushing the boundaries of what is achievable in the field of robotic manipulation and learning from human demonstrations.

# Chapter 3

# Beyond Object Pose: Learning Impedance by Watching YouTube and Trying in Simulation

## 3.1 Introduction

The previous chapter demonstrates the capability to learn from human demonstrations in a controlled lab setting using depth cameras to create segmented point clouds. This chapter aims to broaden the scope of learning to more generalized data sources, particularly focusing on videos from unstructured environments like YouTube. The potential of such platforms for scaling robot learning is substantial.

Extracting robot programs from YouTube videos, however, presents unique challenges. The visual characteristics of these videos often significantly differ from target test environments, impacting the transferability of visually-based programs such as Behavior Cloning [74] or Diffusion Policies [23]. Moreover, to extend learning beyond rigid body object pose to encompass a broader range of deformable objects, liquids, and granular materials requires strategies to infer physics from videos, determining the dominant forces for different objects. Understanding forces and impedance at contacts is key to success for dynamic tasks as well as many quasi-static tasks..

To address these challenges, we focus on object contacts and physical constraints,

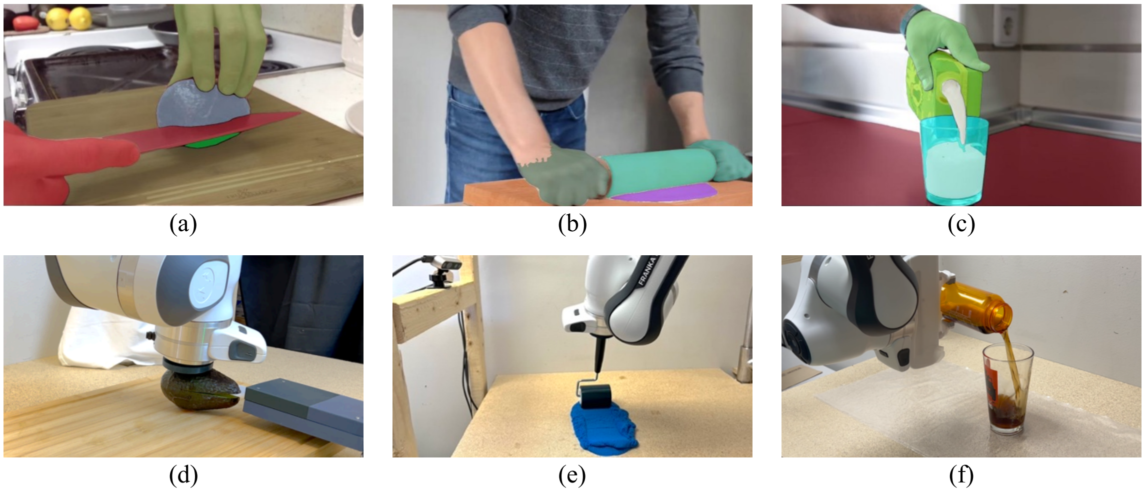


Figure 3.1: Three Tasks Learned from YouTube. (a) and (d) Cutting an avocado. (b) and (e) Rolling dough. (c) and (f) Pouring.

such as establishing and breaking contacts, rather than relying solely on direct visual appearances. With these object states and contact constraints, we construct scene graphs, a popular model in computer vision for describing attributes and relations among objects. These graphs, combined with advanced simulation environments, capable of modeling non-rigid objects and providing force information, can help perceive the invisible physics in video content.

The key contributions of this chapter include:

1. Enabling the use of single YouTube video as a source for robot learning manipulation of deformable objects, liquids, and granular materials.
2. Abstracing unseen videos into physical constraints and goals (represented as scene graphs) for simulations, aiding in the learning and optimization of robotic tasks in diverse environments.
3. Demonstrating the transferability of tasks learned in simulation to real-world scenarios using actual robots.

The effectiveness of this methodology is validated with experiments on the Franka Research 3 Robot and uFactory xArm, encompassing tasks, illustrated in Figure 3.1, like slicing fruit, pouring, and dough manipulation. Each task, learned from a single YouTube video, represents a significant advancement in robot learning from human video demonstrations, paving the way for future scalability in this domain.

## 3.2 Related Work

### 3.2.1 Scene Graphs

Scene graphs have become a popular representation in computer graphics for describing, manipulating, and rendering complex scenes [90]. In computer vision, scene graphs have primarily been used to abstract the content of 2D images. Visual Genome [50] proposed using natural language captions to generate scene graphs to model attributes and relations among objects, which have been used for many tasks such as image captioning [7], retrieval [45], action recognition [63] and visual question-answering [34].

Armeni et al. [9] proposed a semi-automatic algorithm to construct 3D scene graphs for static rooms. However, their method is limited to static scenes and does not include dynamic objects and their relationships over time. Similarly, Kim et al. [47] proposed a 3D scene graph model for robotics that only includes static objects. These methods rely heavily on SLAM to understand 3D geometry, which is proving difficult to apply to YouTube videos that mainly focus on moving objects. Our work extends the usage of scene graphs to typical instructional YouTube videos, enabling us to reason about dynamic objects and their relationships in 3D over time.

### 3.2.2 Inverse Reinforcement Learning

Similar to Learning from Demonstration (LfD), Inverse Reinforcement Learning (IRL) [5, 33, 37, 66] aims to recover the underlying reward function responsible for a teacher's observed behavior. By understanding and imitating the teacher's decision-making process, an RL agent can learn from demonstrations without the need for manually designing reward functions.

Recent research has focused on enabling RL agents to learn directly from raw sensory data [20, 32, 44, 56, 76, 95], such as videos, which have become a valuable resource for demonstrations. For example, XIRL [101] learns a reward function from video demonstrations through temporal cycle-consistency and trains an RL agent to maximize the learned rewards. Another example is GraphIRL [53], which improves visual representation by employing a graph-structured abstraction. Although these works demonstrate promising results and exhibit potential for cross-embodiment transferability, they often rely on a substantial number of in-domain expert demonstrations captured in similar environments, which significantly restricts the applicability of these methods.

## 3.3 Methods

In this section, we first outline the process of constructing a scene graph from YouTube videos using off-the-shelf tools (Section 3.3.1). Following that, we delve into the process of constructing a digital twin for trajectory optimization within simulation environments to ground the physical constraints (Section 3.3.2). Lastly, we explain the method for transferring the policy to the real world (Section 3.3.3).

### 3.3.1 Video to Scene Graphs

A human demonstration video on YouTube is transformed to scene graphs which detail attributes and relationships among objects and later serve as constraints for simulation and optimization of robot programs.

To construct scene graphs from videos, our approach consists of three key steps, as outlined in Figure 3.2. First, we employ computer vision techniques to extract depth information, perform instance segmentation, and estimate optical flow, which enables us to reconstruct a semantic point cloud for each frame. Next, we decouple camera motion from object motion and project all frames into the coordinates of the first frame. Finally, we retain only objects of interest and calculate their attributes and relationships with other objects, facilitating the construction of the scene graph. We now introduce each module in detail.

#### Semantic Point Cloud Reconstruction

We first estimate the monocular depth. We leverage ZoeDepth [15], an off-the-shelf solution that has been trained on 12 datasets using relative depth and further fine-tuned on two datasets using metric depth. ZoeDepth utilizes a lightweight head with a novel metric bins module, allowing for domain-specific adjustments. During inference, an input image is automatically routed to the appropriate head based on

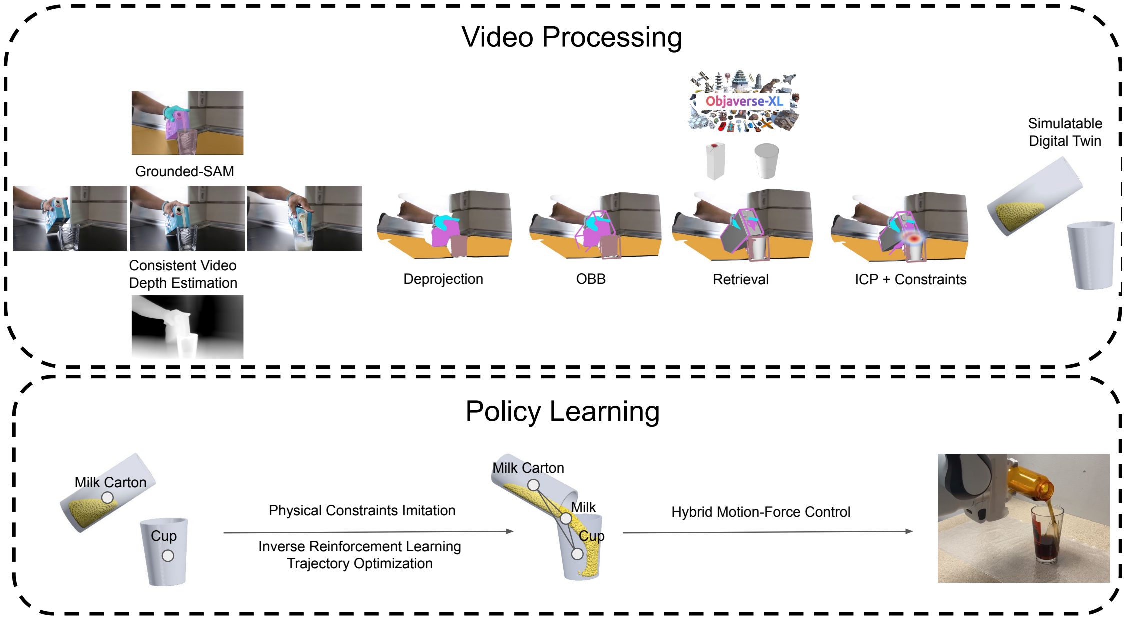


Figure 3.2: Pipeline overview for learning from YouTube videos. (a) Video Processing: Scene graphs are generated by instance segmentation, consistent video depth estimation, and object relationship calculation. (b) Policy Learning: Constructing a digital twin for trajectory optimization and performing sim to real transfer with hybrid motion-force control.

a latent classifier. By leveraging ZoeDepth, we can obtain depth information with metric scale for each RGB frame.

Next, we utilize Grounding DINO [62] for detection and Segment Anything (SAM) [49] for segmentation. Grounding DINO takes an RGB image and a predefined category of interest, producing bounding boxes for each object. These bounding boxes are then used as prompts for SAM to generate masks for each object. Both models exhibit strong zero-shot generalizability to unseen environments, providing accurate 2D positions and semantic information for individual objects.

We also employ GMFlow [96] for optical flow estimation, which is treated as a global matching problem. GMFlow extracts image features using a Convolutional Neural Network (CNN), enhances them with a Transformer, and calculates pairwise feature similarities. We then compute the optical flow in a softmax matching layer. With GMFlow, we obtain crucial motion information between consecutive frames, which aids in tracking objects with the same identity across a video.

Overall, we generate point clouds from depth estimation, mapping the instance segmentation onto these point clouds, and track objects with optical flow. This process enables us to construct a semantic point cloud that encompasses both spatial and semantic details.

### Consistent Video Depth Estimation

We employ Consistent Video Depth Estimation (CVDE) [64] to produce temporally coherent and geometrically consistent depth maps throughout the entire video. Using a monocular video as input, CVDE selects a pair of frames, potentially distant, and utilizes a pre-trained single-image depth estimation model [15] to generate initial depth maps. Establishing correspondences through optical flow with forward-backward consistency checks, CVDE then leverages these correspondences and camera poses to extract 3D geometric constraints. These constraints are decomposed into two losses, namely spatial loss and disparity loss, which are utilized to fine-tune the weight of the depth estimation network via standard backpropagation. During test-time training, this process compels the network to minimize geometric inconsistency errors across multiple frames specific to the video. Following the fine-tuning stage, the final depth estimation results for the video are computed using the fine-tuned model.

### Graph Generation

To generate the scene graph, our first step is to filter out irrelevant objects. We specifically identify objects that interact with the hand as objects of interest. This concept of interaction can be hierarchically propagated. For example, objects directly interacting with the hand are classified as the first level of interaction, and objects interacting with first-level objects are classified as the second level of objects, and so forth. In our current implementation we preserve objects with up to three levels of interaction, prioritizing the most pertinent ones.

Next, we generate a scene graph representation to capture attribute changes and relationship dynamics among the objects of interest. Given that 4D reconstruction remains an open problem with no comprehensive solution, we propose a retrieval-based approach for estimating 6DOF object poses. Initially, we compute an oriented bounding box (OBB) around each object using the method from [68]. Next, employing

Pointnet++[72], we retrieve the nearest neighbor from a subset of Objaverse-XL[28]. We then resize and orient the retrieved object to fit the OBB of each corresponding object. Due to potential occlusion, the estimated OBBs may not tightly bound the objects. To address this, we further refine the object poses using iterative closest points (ICP) [22], which are used are node attributes. Additionally, we incorporate edge information to indicate whether two objects are in contact with each other. This involves calculating the Chamfer Distance between their respective point clouds. If the minimum distance falls below a predefined threshold, we determine that the two objects are in contact. Additionally, we capture the closest points of each pair of objects in contact. To enhance the representation, we apply Gaussian filters to smooth the contact region, and these smoothed regions are then utilized to optimize physical constraints. Through these processes, we construct a scene graph for a given YouTube video. In this graph, each node corresponds to an object of interest, and each edge denotes the relationship between them.

## 3.3.2 Simulation and Optimization

### MLS-MPM Simulation

To enhance the capabilities of robotic learning from human demonstrations, a specialized simulation environment was developed using Python and Taichi. This environment incorporates the Moving Least Squares Material Point Method (MLS-MPM) [40], as per the approach outlined in [94, 98], which is adept at modeling interactions between both soft and rigid objects, including forces between them. The flexibility and precision of MLS-MPM make it an ideal choice for simulating the complex dynamics observed in real-world tasks.

The MLS-MPM works by discretizing materials into a set of material points (Eulerian method), which carry properties like mass, velocity, and deformation gradients. These points are then mapped onto a background grid (Lagrangian method) where force calculations and updates are performed. After grid operations, the updated information is transferred back to the material points. By integrating the robustness of grid-based calculations with the flexibility of particle-based dynamics, MLS-MPM is not only versatile but also capable of being parallelized on GPUs. This makes it a faster and more efficient option compared to traditional methods like the

Finite Element Method (FEM), thereby enhancing its suitability for a wide range of simulation applications.

While most other simulation using MLS-MPM focus on simulating one specific task, our implemented simulation environment, paired with an extensive object library [28], could be capable of replicating a wide array of tasks, automatically constructed from demonstration videos. By accurately modeling the physical interactions and dynamics of various objects, the simulation bridges the gap between virtual simulations and real-world applications.

### Trajectory Optimization

To ground the physical constraints over the entire video, we initially decompose each task into multiple subtasks based on changes in these constraints, such as the establishment and breaking of contacts. For each subtask, we utilize the physical constraints of the subsequent subtask as optimization goals. We frame this process within a Markov Decision Process (MDP) defined by a set of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, and a deterministic, differentiable transition dynamics $s_{t+1} = p(s_t, a_t)$, where $t$ denotes discrete time, and states are composed of different objects $s_t = \{s_t^i\}_{i=1,\dots,n}$. For any pair of objects $(s_t^i, s_t^j)$, the physical constraints can either exist (in contact) or not (not in contact), and a cost function is denoted as $C(s_t^i, s_t^j)$. For any reference state (6 DoF poses of objects) $\hat{s_t}$, a distance function is denoted as $D(s_t, \hat{s_t})$. The objective is to determine a trajectory that minimizes the total loss $L$. Following [60], we use gradient-based trajectory optimization to solve for an open-loop action sequence:

$$\begin{aligned}&\underset{a_0,\dots,a_{T-1}}{\operatorname{argmin}}\ L(a_0, ..., a_{T-1}) = \underset{a_0,\dots,a_{T-1}}{\operatorname{argmin}}\ \lambda_1 \sum_{i,j} C(s_T^i, s_T^j) + \lambda_2 \sum_{t=1}^{T} D(s_t, \hat{s_t}) + \lambda_3 \sum_{t=1}^{T} E(a_t)\\ &\text{where}\, s_{t+1} = p(s_t, a_t)\end{aligned} \tag{3.1}$$

$C(s_T^i, s_T^j)$ represents the KL divergence [52] in contacting distributions if physical constraints exist between $(s_t^i, s_t^j)$; otherwise, it is 0. The action $a_t$ are the translation velocity and angular velocity of each object, and $E(a_t)$ denotes the energy associated with executing action $a_t$. Additionally, $\lambda_1, \lambda_2, \lambda_3$ are weighting parameters.

Table 3.1: Success Rates of Robot Experiments in the Real World.

| Cutting Avocado | Pouring Liquid | Rolling Dough |
|---|---|---|
| 75% | 80% | 50% |

We solve Equation 3.1 by updating the action sequence using $\nabla a_t L$, $t = 0...T$, with an Adam optimizer [48], initialized with the trajectories from the video. Additionally, we compute the force $f_t$ and torque $\tau_t$ observations at each timestep, facilitating the implementation of a hybrid motion-force controller for real-world applications.

### 3.3.3 Hybrid Motion Control

To transfer the result of optimized simulation to a real world experiment, we we employ a hybrid motion-force controller:

$$p_t^r = k_1 p_t^s + k_2(f_t^r - f_t^s) \tag{3.2}$$

Here, $p_t^r$, $f_t^r$ are positions and forces of real robot, and $p_t^s$, $f_t^s$ are positions and forces from the simulation to use as references in the controller. $k_1$ and $k_2$ are impedance parameters, which are determined during experiments. The desired pose of the object of interest is subsequently translated into the desired robot end-effector pose. Employing inverse kinematics, the robot undergoes PD control in joint space.

## 3.4 Experiments and Results

In this section, we present experiments conducted in both simulated and real-world environments, followed by results and an exploration of ablation studies.

### 3.4.1 Experiment Setup

Our experiments were designed to evaluate the performance of our approach on three tasks: Cutting An Avocado, Pouring Liquid, and Rolling Dough. We selected these tasks, ubiquitous in kitchen settings–an important environment for future robots–to assess the model's generalizability and adaptability. We use the simulation for training,

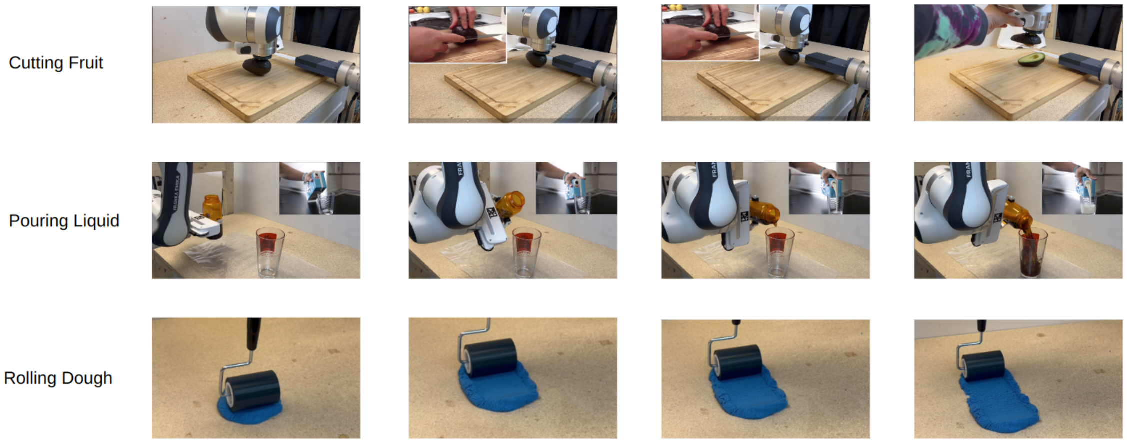


Figure 3.3: Qualitative Results of Robot Experiments. We test on three tasks: Cutting Fruit, Pouring Liquid, and Rolling Dough. The corners of the images present the corresponding frames from the YouTube video demonstrating the human action.

and set up real-world experiment to gauge the model's sim-to-real transferability. Pouring Liquid and Rolling Dough calls for subtle control over the tools and precise handling, testing the dexterity of the learning model. Cutting Fruit, meanwhile, tests the model's ability to learn precise positioning and force application.

### 3.4.2 Results

Table 3.1 displays the success rates of real-world robot experiments. Our approach effectively transferred the policy from YouTube to the robot, achieving a high success rate. Of the three tasks undertaken, manipulating a dough proved most challenging due to the non-uniform nature of the dough, which caused significant disturbances despite controlled force application by the robot during execution.

Figure 3.3 presents the qualitative outcomes of the experiments. Notably, the robot successfully halves an avocado, skillfully navigating around its non-circular seed without attempting to slice through it. This highlights the robot's successful force control during execution.

## 3.5 Limitations and Future Works

This work has a few key limitations. One limitation is that the tasks we tested are simple, allowing a large range of forces and robot impedances. For example, cutting an avocado only requires the seed to be touched by the knife without destroying the avocado, which can be done with a range of forces. However, we think this is characteristic of many tasks, where forces and impedances are constrained to a range but not to exact values. Position control typically has to satisfy much tighter constraints.

Another challenge lies in the current necessity of running optimization processes during task execution, rather than having a pre-defined, optimized program applicable across various scenarios. Furthermore, the system's reliance on the accuracy of vision algorithms poses a significant constraint. Any inaccuracies in scene reconstruction or object identification due to vision errors directly impacts the effectiveness of the simulation and the subsequent robotic action. Addressing these limitations will be crucial in evolving the system into a more precise, efficient, and versatile tool for robot learning.

To conclude this chapter, we extended the learning capabilities of robots to interpret and replicate human demonstrations from unstructured video sources like YouTube. The development and implementation of a simulation environment that supports both rigid and deformable objects and liquids and granular materials have been important in this progress. Using scene graphs from videos as physical constraints for optimization, the system has been able to perceive the invisible physics from demonstration videos with high fidelity, translating them into effective robot policies. By further addressing the current limitations and building on the established foundation, there is potential to significantly enhance the versatility and effectiveness for robots to scale up in learning and performing complex manipulation tasks.

# Chapter 4

# Conclusions

In this thesis, we first explored the complexities of robot learning from human demonstrations, particularly focusing on multi-step tasks such as manual dishwashing. The developed system, which segments task execution into modular primitives based on rigid body object poses and contact relationships, has demonstrated promising results. It has enabled robots to understand and replicate human actions in both laboratory and home environments.

In the second part of the thesis, the learning capabilities of robots were extended to interpret and replicate human demonstrations from unstructured video sources, such as YouTube. The implementation of a simulation environment with optimization was important to infer physics from the videos. By constructing scene graphs from videos and using them as constraints for optimization, the system successfully recovered tasks from demonstration videos with high fidelity, creating robot policies. These achievements are a step towards narrowing the gap between human capabilities and robot execution.

While these methods have some generalization capability, they also revealed certain limitations, particularly in terms of vision accuracy and the effectiveness of task learning and execution. Future work is directed towards refining the system's adaptability and enhancing its dexterity, with a focus on advancing computer vision techniques and intelligent robot learning strategies. By addressing these current limitations and building upon the established groundwork, there is significant potential to further enhance the versatility and effectiveness of robots. This ongoing development lays the foundation for scaling up robots' ability to learn and perform complex manipulation tasks, pushing the boundaries in the field of robot learning from human demonstrations.

# Bibliography

[1] Flippy robot by miso robotics. URL `https://misorobotics.com/flippy/`. 1

[2] Sally the robot. URL `https://www.instagram.com/sallythesaladrobot/`. 1

[3] Hyper robotics, Mar 2023. URL `https://www.hyper-robotics.com/`. 1

[4] Moley robotics, Dec 2023. URL `https://www.moley.com/`. 1

[5] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004. 1, 3.2.2

[6] Arash Amini, Arul Selvam Periyasamy, and Sven Behnke. Yolopose: Transformer-based multi-object 6d pose estimation using keypoint regression. In *International Conference on Intelligent Autonomous Systems*, pages 392–406. Springer, 2022. 2.2.2

[7] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016. 3.2.1

[8] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009. 2.2.1

[9] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019. 3.2.1

[10] Sridhar Pandian Arunachalam, Irmak Güzey, Soumith Chintala, and Lerrel Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023. 2.2.1

[11] Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dex-

terous imitation made easy: A learning-based framework for efficient dexterous manipulation. In *2023 ieee international conference on robotics and automation (icra)*, pages 5954–5961. IEEE, 2023. 2.2.1

[12] Oliver Batchelor. Multi-camera calibration using one or more calibration patterns, May 2023. URL `https://github.com/oliver-batchelor/multical`. 2.4.1

[13] Patrick Beeson and Barrett Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 928–935. IEEE, 2015. 2.4.3

[14] Homanga Bharadhwaj, Abhinav Gupta, and Shubham Tulsiani. Visual affordance prediction for guiding robot exploration. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2.2.1

[15] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 3.3.1, 3.3.1

[16] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Survey: Robot programming by demonstration. *Handbook of robotics*, 59(BOOK_CHAP), 2008. 2.2.1

[17] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 2.2.3

[18] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007. 2.2.1

[19] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021. 2.2.2

[20] Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021. 3.2.2

[21] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021. 2.2.1

[22] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. In *2002 International Conference on Pattern Recognition*, volume 3, pages 545–548. IEEE, 2002. 2.2.2, 2.4.1, 3.3.1

[23] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1, 3.1

[24] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2021. 2.4.3

[25] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022. 2.2.3

[26] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. *Advances in Neural Information Processing Systems*, 35:13745–13758, 2022. 2.2.3

[27] Sudeep Dasari and Abhinav Gupta. Transformers for one-shot imitation learning. In *CoRL 2020*, 2020. 2.2.1

[28] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 3.3.1, 3.3.2

[29] Shivin Devgon, Jeffrey Ichnowski, Ashwin Balakrishna, Harry Zhang, and Ken Goldberg. Orienting novel 3d objects using self-supervised learning of rotation transforms. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1453–1460. IEEE, 2020. 2.2.2

[30] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. 2.2.2

[31] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2.2.3

[32] Arnaud Fickinger, Samuel Cohen, Stuart Russell, and Brandon Amos. Cross-domain imitation learning via optimal transport. *arXiv preprint*

*arXiv:2110.03684*, 2021. 3.2.2

[33] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*. 3.2.2

[34] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 3.2.1

[35] Grounded-SAM Contributors. Grounded-Segment-Anything, April 2023. URL https://github.com/IDEA-Research/Grounded-Segment-Anything. 1, 1, 2.2.2, 2.4.1

[36] Frederik Hagelskjær and Anders Glent Buch. Pointvotenet: Accurate object detection and 6 dof pose estimation in point clouds. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2641–2645. IEEE, 2020. 2.2.2

[37] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016. 1, 3.2.2

[38] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 2.2.2

[39] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective flow aggregation for data-limited 6d object pose estimation. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 2.2.2

[40] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics*, 37(4):150, 2018. 3.3.2

[41] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hotnet: Non-autoregressive transformer for 3d hand-object pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3136–3145, 2020. 2.2.2

[42] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 2.2.1

[43] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages

10236–10247, 2020. 2.2.3

[44] Jun Jin, Laura Petrich, Zichen Zhang, Masood Dehghan, and Martin Jagersand. Visual geometric skill inference by watching human demonstration. In *2020 ieee international conference on robotics and automation (icra)*, pages 8985–8991. IEEE, 2020. 3.2.2

[45] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015. 3.2.1

[46] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006. 2.4.1

[47] Ue-Hwan Kim, Jin-Man Park, Taek-Jin Song, and Jong-Hwan Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE transactions on cybernetics*, 50(12):4921–4933, 2019. 3.2.1

[48] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3.3.2

[49] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2.4.1, 3.3.1

[50] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 3.2.1

[51] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000. 2.4.3

[52] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 3.3.2

[53] Sateesh Kumar, Jonathan Zamora, Nicklas Hansen, Rishabh Jangir, and Xiaolong Wang. Graph inverse reinforcement learning from diverse videos. In *Conference on Robot Learning*, pages 55–66. PMLR, 2023. 3.2.2

[54] Tobias Kunz and Mike Stilman. Time-optimal trajectory generation for path following with bounded acceleration and velocity. *Robotics: Science and Systems VIII*, pages 1–8, 2012. 2.4.3

[55] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020. 2.2.2

[56] Youngwoon Lee, Andrew Szot, Shao-Hua Sun, and Joseph J Lim. Generalizable imitation learning from observation via inferring goal proximity. *Advances in neural information processing systems*, 34:16118–16130, 2021. 3.2.2

[57] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 2.2.1

[58] Fu Li, Shishir Reddy Vutukur, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation supplementary. 2.2.2

[59] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3889–3898, 2019. 2.2.3

[60] Xingyu Lin, Zhiao Huang, Yunzhu Li, Joshua B. Tenenbaum, David Held, and Chuang Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. 2022. 3.3.2

[61] Yixin Lin, Austin S. Wang, Giovanni Sutanto, Akshara Rai, and Franziska Meier. Polymetis. `https://facebookresearch.github.io/fairo/polymetis/`, 2021. 2.4.3

[62] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2.4.1, 3.3.1

[63] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 852–869. Springer, 2016. 3.2.1

[64] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39 (4):71–1, 2020. 3.3.1

[65] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020. 2.2.1

[66] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000. 3.2.2

[67] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561. 2.4.1

[68] Joseph O'Rourke. Finding minimal enclosing boxes. *International journal of computer & information sciences*, 14:183–199, 1985. 3.3.1

[69] Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning*, pages 1783–1792. PMLR, 2023. 2.2.2

[70] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 2.2.3

[71] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2663–2672, 2017. 2.2.3

[72] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3.3.1

[73] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022. 2.2.1

[74] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011. 1, 2.2.1, 3.1

[75] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999. 2.2.1

[76] Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with

interaction. In *Conference on Robot Learning*, pages 339–354. PMLR, 2021. 3.2.2

[77] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning $k$ modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022. 1, 2.2.1

[78] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020. 2.2.3

[79] Lucy Xiaoyang Shi, Joseph J Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. *arXiv preprint arXiv:2207.07560*, 2022. 1

[80] Shreyas Hampali Shivakumar, Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Ho-3d: A multi-user, multi-object dataset for joint 3d hand-object pose estimation. 2019. 2.2.2

[81] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *RSS*, 2022. 2.2.1

[82] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *Robotics and Automation Letters*, 2020. 2.2.1

[83] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022. 2.2.2

[84] Martin Sundermeyer, Tomáš Hodaň, Yann Labbe, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiří Matas. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2784–2793, 2023. 2.2.2

[85] V-HACD Contributors. The V-HACD library decomposes a 3D surface into a set of ”near” convex parts, October 2022. URL `https://github.com/kmammou/v-hacd`. 2.4.1

[86] Joel Vidal, Chyi-Yeu Lin, Xavier Lladó, and Robert Martí. A method for 6d pose estimation of free-form rigid objects using point pair features on range data. *Sensors*, 18(8):2678, 2018. 2.2.2

[87] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural

networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017. 2.2.3

[88] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdrnet: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021. 2.2.2

[89] Jianren Wang, Sudeep Dasari, Mohan Kumar Srirama, Shubham Tulsiani, and Abhinav Gupta. Manipulate by seeing: Creating manipulation controllers from pre-trained representations. *ICCV*, 2023. 1, 2.2.1

[90] Rui Wang and Xuelei Qian. *OpenSceneGraph 3.0: Beginner's guide*. Packt Publishing Ltd, 2010. 3.2.1

[91] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10776–10785, 2021. 2.2.3

[92] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *arXiv preprint arXiv:2201.12716*, 2022. 1, 2.2.1

[93] Yangzheng Wu, Alireza Javaheri, Mohsen Zand, and Michael Greenspan. Keypoint cascade voting for point cloud based 6dof pose estimation. In *2022 International Conference on 3D Vision (3DV)*, pages 176–186. IEEE, 2022. 2.2.2

[94] Zhou Xian, Bo Zhu, Zhenjia Xu, Hsiao-Yu Tung, Antonio Torralba, Katerina Fragkiadaki, and Chuang Gan. Fluidlab: A differentiable environment for benchmarking complex fluid manipulation. In *International Conference on Learning Representations*, 2023. 3.3.2

[95] 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. IEEE, 2021. 3.2.2

[96] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 3.3.1

[97] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark, 2018. 2.2.3

[98] Zhenjia Xu, Zhou Xian, Xingyu Lin, Cheng Chi, Zhiao Huang, Chuang Gan, and Shuran Song. Roboninja: Learning an adaptive cutting policy for multi-material objects. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 3.3.2

[99] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6499–6507, 2018. 2.2.3

[100] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021. 2.2.2

[101] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pages 537–546. PMLR, 2022. 3.2.2

[102] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020. 1

[103] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018. 2.2.1

[104] Zhongqun Zhang, Wei Chen, Linfang Zheng, Aleš Leonardis, and Hyung Jin Chang. Trans6d: Transformer-based 6d object pose estimation and refinement. In *European Conference on Computer Vision*, pages 112–128. Springer, 2022. 2.2.2

[105] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 2.2.1

[106] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. VIOLA: Object-centric imitation learning for vision-based robot manipulation. In *6th Annual Conference on Robot Learning*, 2022. URL `https://openreview.net/forum?id=L8hCfhPbFho`. 2.2.1

[107] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. In *7th Annual Conference on Robot Learning*, 2023. URL `https://openreview.net/forum?id=9SM6l0HyY_`. 2.2.1

[108] Lu Zou, Zhangjin Huang, Naijie Gu, and Guoping Wang. 6d-vit: Category-

level 6d object pose estimation via transformer-based instance representation learning. *IEEE Transactions on Image Processing*, 31:6907–6921, 2022. 2.2.2