Scaling Human Supervision for Robot Manipulation

Michael Murray

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Maya Cakmak, Chair

Joshua Smith

Abhishek Gupta

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Scaling Human Supervision for

Robot Manipulation

Michael Murray

Chair of the Supervisory Committee:
Maya Cakmak
Computer Science & Engineering

Robots are increasingly deployed in unstructured, human-centric environments—such as homes, warehouses, and hospitals—where they must adapt to diverse tasks, novel objects, and evolving user preferences. While pretraining on large-scale datasets or in simulation provides a useful foundation for general-purpose manipulation, the domain gap and scarcity of task-relevant real-world data remain major obstacles to robust deployment. Learning from real-world experience is critical for improving generalization and reliability, but the real world provides no automatic supervision. Human supervision—through demonstrations or interventions—remains the most effective and grounded signal for guiding robot learning, yet it is difficult and expensive to scale.

This dissertation explores two complementary approaches to address this challenge. First, it investigates methods for distilling human supervision into reward models that enable reinforcement learning beyond the original data. These learned rewards allow robots to refine their behavior autonomously, increasing sample efficiency while reducing dependence on constant human input. Second, it explores how vision and language foundation models pretrained on internet data can simplify and enhance human supervision. By leveraging these models to extract task-relevant

structure from multimodal demonstrations, robots can acquire skills from a single example and generalize to new objects, tasks, and environments.

These approaches are validated across a range of real-world robotic manipulation tasks. By making human supervision both scalable and intuitive, this work aims to enable robots that require less supervision, learn more efficiently, and succeed more reliably in the open-ended, human-centric environments of the real world.

Contents

Acknowledgments

This thesis reflects the culmination of work done during my time as a PhD student in the Human Centered Robotics Lab at the University of Washington. I feel incredibly fortunate to have spent these years surrounded by people who challenged, inspired, and supported me through every high and low.

First and foremost, I am deeply grateful to my advisor, Maya Cakmak, for her steady guidance and encouragement throughout my PhD. Her thoughtful mentorship helped me grow not only as a researcher but also as a more deliberate and human-centered engineer. I also want to thank my cohort in the Human Centered Robotics Lab—Vinitha Ranganeni, Nick Walker, Amal Nanavati, and Leah Perlmutter. Each of you brought unique perspectives to our group, and I learned so much about robotics, research, and collaboration from your example.

Thanks to Joshua Smith and all the contributors to the UW + Amazon Science Hub. Thanks to Abhishek Gupta and everyone in the WEIRD Lab for their generosity with time, ideas, and feedback over the years. Thanks to Sylvia Dai and Mike Wolf from Amazon, for their encouragement and support throughout the years.

To my family—my parents, Cindy and Tim, and my siblings Shannon, Amanda, Carly, and JP—thank you for always being in my corner.

And finally, to Brooke: thank you for doing this whole thing with me. You carried me through the hardest moments and made the best ones even better. I couldn't have done this without you.

Chapter 1

Introduction

1.1 Introduction

Robots are steadily transitioning from the highly structured domains of factory floors to the unstructured and unpredictable spaces of everyday human life—homes, warehouses, hospitals, and beyond. In these environments, robots must adapt to diverse objects, tasks, and user preferences, often in real time. Yet despite recent advances in machine learning, enabling robust general-purpose manipulation in the real world remains an open challenge.

A key obstacle is the scarcity and cost of real-world data. Pretraining in simulation or on large-scale internet datasets provides useful priors, but these methods struggle to bridge the domain gap: simulated sensors and dynamics are imperfect proxies for reality, and open-source datasets rarely capture the nuanced requirements of specific manipulation tasks. As a result, robots often fail to generalize when deployed outside controlled environments.

Human supervision—whether through demonstrations, interventions, or feedback—remains the most reliable source of task-relevant information. It offers a grounded way to correct behavior, inject domain knowledge, and encode task goals in diverse settings. However, human supervision

is expensive, time-consuming, and difficult to scale. Teaching every robot every skill by hand is untenable.

This thesis explores two complementary strategies for scaling and simplifying human supervision in robot manipulation:

1. **Distilling supervision into reward models.** Inspired by reinforcement learning from human feedback (RLHF), this work shows how demonstrations and interventions can be transformed into reward functions that generalize beyond the original data. These learned rewards enable robots to refine their behavior autonomously, allowing each moment of human guidance to influence many future actions.

2. **Leveraging foundation models to enhance supervision.** Recent advances in large-scale vision and language models have unlocked powerful new priors that capture semantics, affordances, and commonsense reasoning. This thesis demonstrates how these pretrained models can extract task-relevant structure from multimodal demonstrations—combining visual cues with spoken narration—and enable skill acquisition from just a single example.

These contributions converge on a central thesis: robots can achieve general-purpose manipulation not through exhaustive supervision, but by learning to extract maximum insight from minimal human guidance. By distilling human guidance into generalizable reward models and leveraging foundation model priors to interpret demonstrations, robots can acquire robust manipulation skills from remarkably sparse supervision.

1.1.1 Thesis Overview

The rest of this dissertation is organized as follows:

- **Chapter 2** reviews key approaches and challenges in scaling human supervision for robot manipulation learning, examining different forms of supervision, learning frameworks, and the role of foundation models in robotics.

- **Chapter 3** presents Interactive Visual Failure Prediction, a method for distilling human interventions into scalable reward functions through interactive perception. We demonstrate how this approach can amplify supervision beyond the original data and guide reinforcement learning in real-world grasping tasks.

- **Chapter 4** introduces Diffusion-PbD, showing how features from pre-trained diffusion models can enable one-shot learning from visual demonstrations. We demonstrate successful transfer of manipulation skills across different viewpoints, objects, and environments using only foundation model priors.

- **Chapter 5** explores ShowTell, a neuro-symbolic framework that combines visual demonstrations with spoken narration to synthesize robot manipulation programs. We show how multimodal demonstrations enable learning of complex behaviors involving conditions, iteration, and logical reasoning from a single example.

- **Chapter 6** concludes with a discussion of broader implications, limitations, and future directions for scaling intuitive human supervision in robotics.

By making supervision both scalable and intuitive, this thesis aims to take a step toward robots that can learn more autonomously, generalize more effectively, and collaborate with humans more naturally in the open-ended environments of the real world.

Chapter 2

Background

This chapter reviews key approaches and challenges in scaling human supervision for robot manipulation learning. We begin by examining different forms of human supervision and their relative strengths. We then discuss learning frameworks that incorporate this supervision, with particular attention to scaling challenges. Finally, we explore recent work leveraging foundation models as priors in robotics, highlighting opportunities for making human supervision more efficient. This background establishes the foundation for the three complementary approaches presented in this thesis: Interactive Visual Failure Prediction (Chapter 3), Diffusion-PbD (Chapter 4), and ShowTell (Chapter 5).

2.0.1 Forms of Human Supervision in Robot Learning

Human supervision in robotics can be characterized both by the *interaction mechanism* (how humans provide feedback) and the *feedback format* (what information is conveyed). Understanding these dimensions and their trade-offs is crucial for developing more scalable approaches.

Feedback Formats

The format of human feedback exists on a spectrum from rich demonstrations to simple binary signals:

Demonstrated Trajectories. The most information-rich format, where humans provide complete state-action sequences. While these contain detailed behavioral information, they require significant time and expertise to collect. Early works such as Schaal [107] and Billard et al. [16] establish how robots can acquire manipulation skills from human demonstrations. Multiple broad surveys on robot learning from demonstration have been conducted [9, 101] categorizing different approaches, including direct imitation, motion primitives, and reward-based learning from trajectories. More recently, Mandlekar et al. [86] systematically analyze what matters in learning from offline human demonstrations, highlighting the importance of dataset composition, task diversity, and model architectures in large-scale robot manipulation. As we demonstrate in Chapter 4, the limitations of traditional trajectory-based learning can be addressed by leveraging foundation model features to enable generalization from single demonstrations.

Binary Feedback. Binary feedback provides a simple and unambiguous supervision signal, indicating only whether a behavior succeeded or failed. It is easy to provide and requires minimal human effort, making it useful in settings where rapid feedback is needed. While limited in expressivity, it has been effectively used in RL and reward shaping for policy improvement [124, 66]. However, its inability to convey failure causes makes generalization difficult.

Natural Language Feedback. A flexible and intuitive way for humans to specify high-level task objectives and fine-grained corrections without requiring explicit demonstrations. There are many recent approaches to grounding language instructions into robot policies [69, 111, 3, 63, 18]

and rewards [147, 80]. Similarly, language has been used for corrective feedback to refine robotic behavior [109, 60]. A recent survey [54] categorizes language use in robot learning, including commanding robots, inter-robot communication, and internal reasoning. The study highlights the role of large language models (LLMs) in interpreting human feedback and integrating language with multimodal inputs such as vision and proprioception. While language-based feedback reduces the need for demonstrations, challenges remain in handling ambiguity, grounding verbal corrections into structured actions, and integrating language with low-level control signals. The ShowTell framework presented in Chapter 5 addresses these challenges by combining visual demonstrations with spoken narration to enable structured program synthesis.

Preference Comparisons. Preference-based reward learning leverages binary feedback, where a human annotator selects which of two behaviors is preferred. This approach, pioneered by [27], has been shown to be more reliable than scalar ratings, as humans often find relative comparisons easier than absolute numerical judgments. Instead of requiring annotators to quantify success explicitly, preference comparisons allow robots to infer a reward function from pairwise ranking data, making it a more intuitive and scalable feedback mechanism. Further work by [96] explores how preference comparisons can be efficiently collected and integrated into reward learning. By combining demonstrations and pairwise preferences, their method improves sample efficiency while reducing the annotation burden. However, preference learning can still suffer from label inconsistencies, especially when preference annotations are ambiguous or context-dependent. Addressing these challenges requires robust human response models that account for preference inconsistencies and biases.

Scalar Feedback. Scalar feedback, such as numerical ratings or continuous scores, provides richer supervision than binary signals, enabling more fine-grained learning [133]. However,

human-provided scalar rewards can be inconsistently calibrated across annotators [143] and impose a higher cognitive burden. Simplified alternatives like Likert scales reduce complexity but at the cost of expressiveness [65, 82, 10]. These challenges highlight the trade-off between expressivity and reliability in human feedback mechanisms. While scalar feedback is more informative than binary comparisons, its effectiveness depends on careful design considerations to mitigate bias, inconsistency, and cognitive load on human evaluators.

Interaction Mechanisms

Different feedback formats can be conveyed through various interaction mechanisms:

Direct Teleoperation. Humans directly control the robot, typically providing demonstrated trajectories. This remains one of the most effective methods for acquiring high-quality demonstrations, as seen in studies on human-in-the-loop RL [67, 79] and teleoperation-based imitation learning [56, 18]. However, teleoperation requires significant expertise and constant human attention, making it difficult to scale.

Kinesthetic Teaching. Physical guidance of the robot, commonly used for collecting demonstrations in manipulation tasks [9]. This provides intuitive interaction and naturally accounts for robot dynamics, but requires the demonstrator to be physically present and may not be scalable to all robots or remote deployments.

Remote Interfaces. Remote interfaces allow humans to supervise and guide robots without being physically present, enabling scalable supervision across multiple robots. These interfaces vary from high-fidelity VR/AR-based teleoperation systems [62, 31] to web-based dashboards [100, 88]. While remote supervision reduces physical constraints, it introduces challenges such as latency, reduced situational awareness, and control mismatches. Advances in multimodal feedback,

such as haptic or visuotactile signals [13], aim to mitigate these issues by providing richer feedback beyond visual inputs.

Offline Annotation. Humans provide feedback on previously collected robot trajectories without real-time interaction. Several works have demonstrated how offline preference labels or scalar feedback can be efficiently collected and used to learn reward functions [27, 106, 21]. This mechanism is particularly attractive for scaling, as it allows asynchronous feedback collection from multiple annotators and can leverage crowdsourcing platforms [90]. However, it requires careful consideration of how to present robot behaviors effectively to human annotators.

Passive Demonstration. Passive demonstration is an emerging paradigm where robots learn by observing human behavior without direct control signals. This includes learning from video demonstrations [35, 49, 11, 140, 89] and leveraging large-scale human activity datasets [81, 91, 144]. While passive learning offers significant scalability advantages by removing the need for explicit teleoperation, it introduces challenges in viewpoint matching, action correspondence, and intent inference. Addressing these challenges often requires learning robust representations for mapping human actions to robot motions, as explored in Chapter 4.

2.0.2 Learning Frameworks for Incorporating Human Supervision

Various learning frameworks have been developed to integrate human supervision into robotic learning. These frameworks differ in how they utilize human-provided signals, how they scale to larger datasets, and how they balance human effort with autonomous learning. We categorize these frameworks into three primary approaches: imitation learning, reward learning and reinforcement learning, and hybrid approaches that combine demonstrations, feedback, and interventions.

Imitation Learning

Imitation learning (IL) is one of the most direct ways to integrate human supervision by mapping observations to actions using human-provided demonstrations. By directly learning from expert trajectories, IL bypasses the need for explicit reward functions, making it an appealing method for robotic learning. The simplest form of IL is Behavior Cloning (BC), where a policy is trained via supervised learning on a dataset of demonstrations. However, traditional BC suffers from distributional shift—small errors compound when the agent encounters states not present in the training data, leading to cascading failures.

One approach to improving IL scalability is imitation learning with weak supervision, where the agent learns from suboptimal or incomplete demonstrations. Works like GAIL (Generative Adversarial Imitation Learning) [46] frame IL as an adversarial learning problem, allowing robots to learn robust policies even from noisy or limited expert data. Similarly, BC-Z [56] demonstrates how large-scale video datasets can be leveraged for scalable imitation learning.

Modern approaches have introduced more expressive models for imitation learning from demonstrations. Diffusion Policies [26] propose using diffusion models for trajectory generation, capturing multi-modal behavior and improving over traditional imitation learning methods. Similarly, the Action Chunking Transformer (ACT) [152] leverages transformers to process and learn temporally extended action sequences, enabling robots to imitate long-horizon human demonstrations more efficiently. Building on these advances, Chapter 4 demonstrates how pretrained diffusion model features can enable one-shot learning from visual demonstrations, significantly reducing the data requirements of traditional imitation learning.

While IL is effective for robotic manipulation tasks, its reliance on high-quality demonstration data remains a major bottleneck for scaling to diverse real-world settings. This limitation has motivated hybrid approaches that incorporate reinforcement learning (RL) and reward learning to

improve policy learning beyond pure imitation.

Reward Learning and Reinforcement Learning

Reinforcement learning (RL) provides a powerful framework for robots to learn from experience, but its success hinges on having a well-defined reward function. Since manually specifying reward functions is difficult and often leads to unintended behaviors, a major research focus has been on learning reward functions from human supervision.

One of the earliest approaches in this direction is inverse reinforcement learning (IRL) [92, 1], which learns a reward function by assuming that expert demonstrations are optimal and inferring the underlying objective that explains them. While IRL enables robots to generalize beyond direct imitation, it is often ill-posed, as many different reward functions can explain the same demonstrations, and requires careful modeling of expert intent.

To reduce the need for full demonstrations, more recent methods focus on learning reward functions from human preference feedback. Instead of requiring explicit numerical rewards, robots learn by comparing different trajectories and optimizing behaviors that humans prefer [106]. This preference-based RL, inspired by RLHF, allows robots to learn task objectives without requiring extensive demonstrations. However, these methods are typically sample inefficient, as preference labels provide a weak learning signal, requiring a large number of comparisons before a robust reward function emerges [23].

An alternative strategy is to incorporate structured priors about human intent, rather than relying solely on reward learning from raw feedback. Hiranaka et al. [44] explore this idea by proposing a framework that learns primitive skill-based representations from human evaluative signals. Instead of learning a task-specific reward function from scratch, this approach enables robots to generalize across variations of a task by leveraging structured skill representations.

Beyond sample efficiency, a core issue in reward learning is exploration, as robots must

efficiently discover good behaviors without blindly searching through all possible actions. Torne et al. [126] introduce a goal-conditioned exploration strategy, where humans provide breadcrumbs to guide the agent toward rewarding behaviors. This structured guidance reduces the need for exhaustive trial-and-error by helping the robot prioritize promising trajectories.

Taken together, these works illustrate the strengths and limitations of learning from human preference feedback and evaluative signals. While IRL, RLHF, and reward sketching provide useful supervisory signals, they often remain too sparse or indirect to fully specify a task's reward structure. This motivates alternative approaches, particularly those that learn rewards from human interventions, where humans provide real-time corrective actions rather than passive comparisons or scores. The next section explores these hybrid approaches, which integrate demonstrations, feedback, and interventions into a unified framework for more scalable and intuitive robot learning.

Hybrid Approaches: Combining Demonstrations, Feedback, and Interventions

While imitation learning and reinforcement learning have both been widely used for robotic learning, each has limitations. IL suffers from distributional shift when deployed outside its training data, while RL requires extensive exploration and well-designed reward functions. To address these issues, researchers have explored hybrid approaches that integrate both paradigms, leveraging demonstrations, reward learning, and human interventions to improve sample efficiency and generalization.

Combining Reinforcement Learning and Imitation Learning A key research direction has been to combine IL and RL so that demonstrations provide an efficient initialization, while RL enables policy improvement beyond expert data. One early approach is residual RL [58, 8], where a policy learns a residual correction on top of an imitation-learned policy. This allows RL to fine-tune a behavior-cloned policy while preserving the benefits of initial expert demonstrations. Similarly,

deep Q-learning from demonstrations (DQfD) [43] pretrains a Q-function using demonstrations before continuing RL fine-tuning, reducing sample complexity by leveraging human-labeled transitions. A more recent development is reinforcement learning from prior data (RLPD) [12], which optimizes RL policies using offline data from demonstrations and prior robot experience. Unlike earlier approaches, RLPD balances behavior cloning with RL objectives to avoid catastrophic forgetting while still allowing policies to improve beyond imitation. This approach has shown strong performance in domains where offline data provides a useful initialization for RL without requiring an explicit reward function.

Incorporating Human Interventions While IL and RL integration improves sample efficiency, human interventions offer a way to further accelerate learning by correcting robot mistakes in real time. One of the most notable early approaches in this area is DAgger (Dataset Aggregation) [105], which iteratively collects corrections from a human expert. DAgger significantly improves generalization by allowing the robot to learn from states it encounters during execution rather than just those observed in demonstrations. A more recent approach is to learn from the interventions using weighted behavior cloning [72]. Beyond behavior cloning, some methods integrate interventions directly into RL. For example, Reinforcement Learning with Intervention Feedback (RLIF) [78], treats human corrections as implicit indicators of failure states within an RL framework. Human-in-the-Loop Sample-Efficient Reinforcement Learning (HIL-SERL) extends RLPD by simply integrating interventions into the RL replay buffer [79]. HIL-SERL builds on the idea that human corrections provide high-value training data, incorporating intervention-labeled transitions into the experience replay mechanism to improve sample efficiency. This approach enables robots to continuously refine their policies using both past experiences and human supervision.

2.0.3 Leveraging Foundation Models as Priors

Foundation models offer a powerful way to reduce the need for domain-specific supervision in robotics. By providing strong perceptual and conceptual priors, these models enable robots to better interpret human feedback, ground language instructions, and generalize across diverse scenarios.

Pretrained Language Models for Robotics. Language models pretrained on massive corpora (e.g., GPT-4, LLaMA) provide rich priors for task specification and action reasoning. Many recent works use language models to map from high-level language goals to sequences of robot actions [69, 111, 3, 63, 18] and reward functions [147, 80]. While language models are often used zero-shot, some recent works have explored how to fine-tune LLMs specifically for robot instruction following [137, 70].

Pretrained Multimodal Models for Robotics. Vision-language models (VLMs) such as CLIP [97] and GPT4-V [142] have been widely used for robot manipulation. These models can be used zero-shot to ground concepts in visual observations [131, 129] or they can be integrated into larger model training pipelines as backbones [110, 20]. These models provide strong pre-trained visual priors, leveraging web-scale knowledge to help robots generalize and learn more efficiently.

Vision-Language-Action Models (VLAs). Recent advances have explored jointly training vision-language-action models to unify perception, reasoning, and control. PaLM-E [30] extends a large language model (PaLM) with ViT-based visual inputs, allowing end-to-end language-conditioned robotic control. OpenVLA [52] and Octo [57] further integrate pretrained vision encoders (ViT) and LLMs into robotic decision-making, demonstrating strong zero-shot and few-shot generalization across unseen tasks.

Addressed Challenges

The integration of human supervision frameworks and foundation models provides a promising path toward efficient robot learning in the real world. However, several key challenges have limited the practical deployment of these methods. This thesis addresses four critical limitations through novel technical contributions that span reward learning, few-shot generalization, and multimodal understanding:

- **Transforming Human Feedback into Scalable Rewards:** A fundamental challenge in real-world robotics is converting sparse human feedback into reward functions that can guide autonomous learning. While human interventions provide valuable corrective information, existing approaches largely treat them as additional training samples rather than distilling them into generalizable reward models. Additionally, human supervisors often lack access to complete world state information, leading to inconsistent labels. Chapter 3 addresses these challenges through Interactive Visual Failure Prediction, which transforms human intervention feedback into structured reward functions while actively gathering additional sensory information to improve reward quality. This approach enables robots to learn more efficiently from human supervision by both distilling interventions into scalable rewards and addressing partial observability in real-world manipulation tasks.

- **Few-Shot Generalization:** While humans can learn a skill from a single demonstration and generalize to new objects and scenarios, most robotic learning approaches still require extensive datasets or fail to generalize effectively. Current foundation model-based methods struggle to capture the compositional and transferable nature of human skill acquisition. Chapter 4 demonstrates how pretrained diffusion model features can enable true one-shot learning from visual demonstrations, allowing robots to transfer manipulation skills across different viewpoints, objects, and environments using only foundation model priors without

any robot-specific training data.

- **Multimodal Demonstration Understanding:** Most existing vision-language models in robotics use language for simple instruction following or task descriptions, but few leverage it for understanding complex multimodal demonstrations. Language can provide a structured interpretation of demonstrations, encoding logical dependencies, conditions, and iterative behaviors that are common in real-world tasks but rarely represented in current approaches. Chapter 5 shows how language can be combined with visual demonstrations to enable learning of complex behaviors involving conditions, iteration, and logical reasoning from a single example.

- **Beyond Standard Vision-Language Representations:** Current representations in vision-language models for robotics remain largely monolithic embeddings that struggle with higher-level reasoning. There is an opportunity to explore neuro-symbolic representations, which can fuse foundation models with structured reasoning frameworks to improve both generalization and hierarchical task understanding in robotic manipulation. The ShowTell framework presented in Chapter 5 addresses this limitation through a neuro-symbolic approach that combines visual demonstrations with spoken narration to synthesize structured robot manipulation programs, enabling more sophisticated reasoning about task structure and dependencies.

2.0.4 Chapter Summary

The approaches reviewed in this chapter highlight both the promise and limitations of current methods for scaling human supervision in robotics. While significant progress has been made in imitation learning, reward learning, and foundation model integration, key challenges remain in making supervision both scalable and intuitive. The following chapters present three com-

plementary approaches that address these challenges: Interactive Visual Failure Prediction for learning interactive reward functions under partial observability (Chapter 3), Diffusion-PbD for leveraging foundation models to enable one-shot learning (Chapter 4), and ShowTell for combining multimodal demonstrations with structured reasoning (Chapter 5). Together, these contributions demonstrate how intelligent architectures can transform human demonstrations and interventions into generalizable manipulation capabilities.

Chapter 3

Learning Rewards with Interactive Perception

The ability to grasp diverse objects from cluttered environments is central to many robotic applications: from picking items off warehouse shelves to unloading groceries at home. Robots that can reliably grasp objects can automate tasks such as object picking, sorting, and packing. However, developing robust grasping behavior is not trivial, especially in unstructured environments with clutter and large amounts of object diversity. For example, modern warehouses process millions of unique objects from rigid to highly deformable with various shapes and sizes. These objects are often densely packed into highly cluttered containers. The diverse and complex dynamics of such environments make simulating or directly modeling the objects challenging.

Learning from real-world experience is a promising approach that circumvents the challenges of simulation, but typically requires extensive human supervision both in terms of providing labels and in terms of resetting up scenes for autonomous data collection. Additionally, executing picks in the real world is time-consuming, can induce costly failures or object damage, and often requires extensive human intervention. During training, this significantly increases the burden of

data collection and limits the scale at which data can be collected. During execution, failures can be irreversible or require difficult recovery, which can disrupt operational efficiency and limit the viability of robot deployments.

Ideally, failures would be detected early in the picking process, without requiring full execution to determine if a pick will succeed. This would enable us to avoid costly failed picks before they happen. Such capability could also be used to autonomously reward picks without disturbing the scene, providing supervision to continuously shape and improve picking behavior as the robot performs picks in the real world, while minimizing human intervention. We observe that picking can be divided into two sub-tasks: grasping and extraction. Grasping success is critical and highly informative of pick success, while irreversible failures typically happen during extraction. This presents an opportunity to avoid costly failures by predicting success before extraction.

However, it's often difficult to determine whether a grasp is successful and stable from passive visual observation alone due to partial observability, an issue that is compounded by the low visibility and high occurrence of occlusions in densely packed bins. Tactile feedback can help, but is insufficient due to being unable to detect certain modes of failure that are common in cluttered scenes, such as multi-picks. To address this challenge, we draw on ideas from *interactive perception*, a broad class of techniques in which the environment is manipulated to create rich sensory signals that would not be present through passive perception alone [19]. By using interaction to probe for information about the stability of a grasp, we can visually detect failures that are not perceivable by passive vision or tactile feedback. In doing so, we can both improve the training throughput since interventions can be minimized and the detected failures can be used to finetune grasp strategies, and also improve success rates since unstable grasps can be pre-empted and avoided.

While detecting suboptimal grasps using probes is useful for both training throughput and evaluation success rate, it can be challenging to actually perform this detection autonomously. On the other hand, humans possess a remarkable ability to visually judge grasp quality and

refine their judgement through visual feedback [85] while only partially executing grasps. We are interested in exploiting this ability by directly leveraging human feedback for learning to perform and evaluate robotic grasping behavior, both in terms of the actual grasping behavior and in terms of preemptive evaluation of unsuccessful grasps. We propose a framework in which a human first demonstrates a potential grasp by teleoperating a robot, then observes the robot using probing motions to reveal information about the object configurations in the cluttered scene and test the stability of the grasp. We find that by observing the robot verify their grasp through interaction, humans are able to quickly and accurately classify grasp success.

This enables us to train an interactive visual grasp classifier capable of evaluating grasps in clutter without executing full picks, a capability we refer to as *Interactive Visual Failure Prediction (IVFP)*. Such a capability can be used to autonomously verify grasps during execution to avoid costly downstream failures, which directly improves success rates. IVFP can also be used to autonomously reward grasps as the robot performs picks in the real world, enabling real world learning to improve grasp success with minimal human intervention. Moreover, during evaluation at test time, expensive failures can be preempted by first performing interactive probing and IVFP, and avoiding risky and unsuccessful grasps. We evaluate our approach in a real-world robot deployment using a Stretch RE1 in an industrial warehouse setting. Our experiments show that IVFP can immediately improve picking success by performing introspective online verification. Moreover, we show that IVFP used as a reward function can help improve grasping policies to outperform policies learned through imitation alone. Finally, we show that data collection with IVFP requires significantly less human intervention than typical data collection pipelines wherein picks are fully executed. This suggests that interactive probing can provide significant gains both in terms of training throughput and in terms of overall system success rate in cluttered warehouse settings.

3.1 Problem Statement

We consider a picking task initiated when a robot arrives at a scene of diverse objects densely packed into a cluttered bin. The robot must grasp and extract a given target object without grasping other objects in the bin. The picking task is performed by a manipulation robot with Cartesian motion and a parallel-jaw gripper.

Each *grasp* is defined as a set of variables determining actions of the robot: a 3D point (x, y, z) indicating the grasp point and a pre-grasp gripper width w.

Let $\mathcal{G}$ be the set of all possible grasps, and $\mathcal{S}$ the set of scene states. At each timestep t, the current state $s_t \in \mathcal{S}$ is defined by the bin layout, the poses and states of all objects in the bin, and the pose and state of the robot. The robot does not have access to the state s_t, but only to an observation o_t. An observation $o_t = (I_t, M_t)$ includes an RGB-D camera image I_t and the target object mask M_t. Given the observation o_t, the robot's goal is to generate a grasp action $a_t \in \mathcal{G}$. Once a grasp is generated and executed, the robot performs a fixed extraction motion. The task is considered successful if the target object masked by M_t is extracted from the bin with all other objects remaining in the bin. Once the entire pick has been executed, whether successful or not, the process starts over on the next scene, which may be a slightly modified or entirely new scene.

3.2 Interactive Visual Failure Prediction

We are interested in developing IVFP capability for two important applications. First, we want to verify potential grasps in order to avoid costly failures downstream. Second, because learning methods are limited by the cost of collecting human supervision, we are interested in using IVFP to autonomously reward grasps and improve them through experience. IVFP provides multiple advantages for these purposes. The interaction produces visual feedback that is highly informative

of pick success, supporting accurate grasp classification, and the probe allows us to classify grasps without executing a full pick, enabling execution and training operations with minimal human intervention.

To capture these advantages, the probe design should prioritize (1) reversibility, so as to not disturb the scene, and (2) information gain, to enable accurate classification. We design our probes as a partial execution of the extraction step, where the item is lifted and pulled, but not removed from the bin. In this way, we can gain information about the grasp's impact on extraction before irreversible failures can occur. We also note that by designing the probe as a partial execution of the extraction step, we can simply continue with extraction in the case of success, further facilitating efficient data collection. Since it is challenging to heuristically determine grasp success from probes, we use human supervision to extract the rich information provided by the probe. We note that humans have the ability to both demonstrate potential grasps and perceive when a grasp will fail from interaction, and we utilize human operators for both types of supervision.

We illustrate our framework for learning with IVFP in Figure **??**. Our approach consists of a learned grasping task policy π_θ, a learned grasp classifier C_ϕ, an interactive perception policy π_{IP}, and two stages of learning. In the first stage, a human *demonstrates* a grasp, observes the interactive perception policy π_{IP} probing their grasp, then subsequently *labels* their grasp based on the observations produced by the probe. The demonstrations are used to train an initial grasping task policy π_θ and a grasp classifier C_ϕ is trained from the labels.

In the second stage, we use the components learned from human supervision as building blocks for learning from experience. Now, the robot autonomously generates potential grasps using the latest task policy π_θ. The learned classifier C_ϕ is used both for avoiding failures and for autonomously determining task reward. Using the reward determined by C_ϕ, the policy π_θ is updated offline periodically to maximize predicted reward.

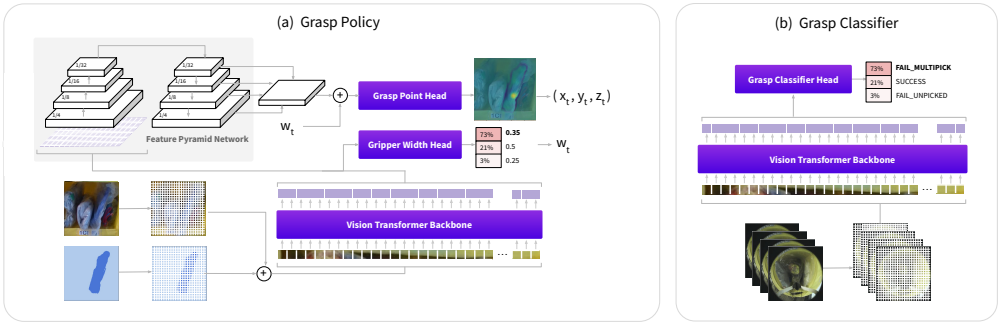

Figure 3.1: (a) Architectures of the neural networks used to model the grasp policy π_θ. The policy takes as input the current RGB image I_t and target object mask M_t. The output includes a 3D grasp position (x_t, y_t, z_t) and pre-grasp gripper width w_t. (b) Architecture of the neural network used to model the grasp classifier C_ϕ. The input to the classifier is a video of the interactive perception policy π_{IP} testing a grasp. The output is a grasp class prediction c_t.

3.2.1 Modeling the Grasp Policy

At each timestep t, the input to the grasp policy is the current observation $o_t = (I_t, M_t)$ and the output is a grasp action $a_t = (w_t, x_t, y_t, z_t)$. The grasp policy is responsible for choosing grasps that are most likely to succeed based on the current observation. Note that the grasp policy performs the initial grasp, while interactive probing and grasp success classification are done with a separate partial execution strategy outlined in Section 3.2.1.

We separate the policy into two action-value modules (Q-functions) that correspond to grasp success: The gripper width module Q_{width} chooses a pre-grasp gripper width, and conditioned on the chosen gripper width, the grasp point module Q_{grasp} decides where to grasp. Both modules are implemented as neural networks and their architectures are illustrated in Figure 3.1a. Note that rather than directly outputting grasp positions and widths, we represent these with implicit functions as noted in prior work [84].

For both modules, the raw observation o_t is first embedded into a pre-trained feature representation space by a vision transformer backbone. This backbone serves as a function that takes the raw observations o_t as input and outputs patch embeddings F_t. The gripper width module

Q_{width} first applies a global average pooling layer to the patch embeddings F_t followed by a linear classifier. This network models an action-value function $Q_{width}(w_t|F_t)$ that correlates with grasp success which we sample from to obtain the pre-grasp gripper width w_t:

$$w_t = \underset{w}{\text{argmax}} Q_{\text{width}}\left(w \mid F_t\right)$$

The grasp point module Q_{grasp} models a spatial action-value function [135, 149, 110] taking input $\gamma_t = (F_t, w_t)$ and outputting a dense pixelwise prediction $Q_{grasp} \in \mathbb{R}^{H \times W}$ of action-values which are used to select a grasp point:

$$(u_t, v_t) = \underset{(u,v)}{\text{argmax}} Q_{\text{grasp}}\left((u, v) \mid \boldsymbol{\gamma}_t\right)$$

To execute the grasp, we map the selected point (u_t, v_t) from the camera image frame to a 3D grasp location (x_t, y_t, z_t) using the depth channel of the image and the known camera calibration. We base our network Q_{grasp} on the Upernet [136] architecture for its high efficiency on spatial tasks. The visual feature embeddings F_t are fed through a Feature Pyramid Network [71] and the outputs are fused. We project the pre-grasp gripper width w_t to match the dimensions of the fused feature map, concatenate them together, then finally apply a convolutional layer to produce a dense pixelwise prediction.

Both networks Q_{width} and Q_{grasp} are initially trained in a supervised maximum likelihood manner to predict grasp actions that imitate the human demonstrations. The networks are trained separately with Q_{width} using standard cross entropy loss and Q_{grasp} using a modified version of the cross entropy loss that incorporates a Gaussian penalty to encourage the model to make predictions that are close to the target point without requiring exact matches.

3.2.2 Interactive Perception and Grasp Success Classification

After performing a grasp according to π_θ, we want to predict if the grasp will succeed in order to avoid costly failures during execution and efficiently reward grasps during training. But it is difficult to determine grasp success from passive observation alone, so to better inform grasp classification, the robot verifies the grasp using the interactive perception policy π_{IP}. For this work, we used a fixed interactive perception policy that performs a cyclic lift-and-pull probing motion to test the grasp. The motion is designed to be a reversible partial execution so as to not perturb the scene, while being able to be executed directly if the grasp is predicted to be successful. This motion produces a set of visual observations I_t^{IP}.

Based on these observations, the grasp classifier C_ϕ is responsible for determining whether a continuation of this particular grasp would be successful or not. The classifier is implemented as a neural network that takes I_t^{IP} as input and outputs a grasp class prediction $c_t \in \{\text{SUCCESS}, \text{FAIL}\}$. An illustration of the network architecture can be found in Figure 3.1b. The network begins with a vision transformer backbone which is pre-trained using a masked auto-encoding scheme [125] on *Something-Something v2* [38], a large-scale dataset with 220,847 videos of humans manipulating objects. The backbone is used to obtain patch embeddings F_t^{IP} followed by a global average pooling layer and finally a linear classifier. This network models the distribution $P(c_t|I_t^{IP})$ from which we sample c_t. The network C_ϕ is trained in a supervised manner using standard cross entropy loss.

We combine the interactive perception policy π_{IP} and the learned classifier C_ϕ to achieve IVFP capability. This capability allows us to both autonomously determine rewards for learning from experience and autonomously verify grasps during execution. In the following sections we describe each of these applications in detail.

3.2.3 Using IVFP for Autonomous Reinforcement Learning

By imitating human demonstrated grasps, we can bootstrap our initial grasping task policy π_θ. As the performance of this policy is limited by the cost of human supervision, we want to further improve the policy by learning from experience. For this purpose, the IVFP capability achieved through the combination of π_{IP} and C_ϕ serves as an interactive reward function (IRF)[50]. After each grasp action a_t, the policy π_{IP} is executed to produce I_t^{IP} which is used by C_ϕ to predict the grasp classification c_t. This classification is used to directly determine the reward $\mathcal{R}_t$:

$$\mathcal{R}_t = \begin{cases} 1, & \text{if } c_t = \text{SUCCESS} \\ -1, & \text{if } c_t = \text{FAIL} \end{cases}$$

After accumulating a dataset of action-reward pairs, we fine-tune the grasping task policy π_θ using an off-policy variant of the REINFORCE algorithm [122] in a contextual bandit setting. Specifically, we update the policy to maximize the expected reward using the policy loss:

$$\mathcal{L} = -\mathbb{E}[\mathcal{R}_t \cdot \nabla_\theta \log(\pi_\theta(a_t|s_t))]$$

3.2.4 Using IVFP for Verification in the Loop

In addition to autonomously determining grasp rewards, we want to utilize the IVFP capability to verify grasps and avoid costly failures. To this end, at test-time we sample multiple grasp parameters from our action-value networks Q_{width} and Q_{grasp} so that we can iteratively attempt alternative grasps in the case of failure. First we sample multiple gripper widths from Q_{width} and each candidate gripper width is input into Q_{grasp} along with the patch embeddings F_t. This results in a set of action-value maps, one for each candidate gripper width. We then sample multiple

Figure 3.2: A subset of the objects used for evaluation. Our item set includes 42 unique objects with a variety of object sizes, shapes, and physical properties. The objects can be rigid or highly deformable.

(w, u, v) combinations across all of the action-value maps weighted by predicted grasp success. We first execute the grasp parameters that are most likely to succeed, verify that grasp using π_{IP}, and evaluate the grasp using C_ϕ. When a failed grasp is detected, we move on to the grasp parameters that are the next most likely to succeed. This process repeats until we have either detected a successful grasp or exhausted our set of candidate grasp parameters.

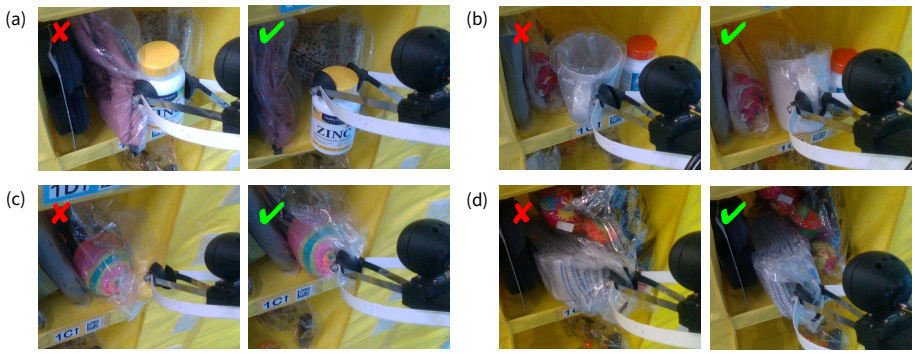

Figure 3.3: Qualitative examples of IVFP utilized for Verification in-the-Loop (VitL). Failed grasps (left) are identified by IVFP and iterated upon to produce successful grasps (right). In (a) and (d) the initial grasp configuration resulted in a multi-pick failure. In (c) the initial grasp configuration resulted in collision with the bin and a subsequent missed-pick. In (b) the initial grasp configuration resulted in a missed-pick.

3.3 Experiments

3.3.1 Hardware

To evaluate our approach, we conduct a series of experiments on a Stretch RE1 robot [61]. The robot's mobile base, arm lift, and telescoping arm are moved in conjunction to reach a 3D target grasp point. An Intel RealSense D435i RGB-D camera is mounted to the frame and a 185 degree FOV fisheye camera is mounted to the wrist, providing observations for the grasp policy and the grasp classifier respectively. We deploy the robot in front of a picking workcell, like those found in industrial fulfillment warehouses, with a shelving unit housing densely packed bins.

3.3.2 Item Set

Our item set consists of 42 unique objects with various shapes, sizes, and physical properties, including deformable and bagged objects. 32 of the objects are used during training and 10 are held out for unseen object evaluation. A subset of the objects can be seen in Figure 3.2.

3.3.3 Data Collection

Our dataset consists of 2,143 human teleoperated picks. To teleoperate the robot, participants use a custom web-based interface designed specifically for this task. First, a camera image of the target bin is displayed and the participant is prompted to select a grasp point by clicking on the image. We map the selected (u, v) position from the camera image to a 3D grasp location using the depth image and known camera calibration. Next, the robot moves its end effector to a pre-grasp pose relative to the selected grasp point and the user is prompted to select a pre-grasp gripper width using a slider. Finally, the robot executes the grasp followed by our fixed interactive perception motion policy, effectively testing the participant's chosen grasp parameters. After watching the images produced by the interactive perception motion, the participant chooses to classify the grasp as a success (in which case the robot executes a fixed extraction policy) or as a failure (in which case the robot resets and a new grasp point is chosen). In total we had 12 participants provide demonstrations including one of the researchers and 11 colleagues recruited from our department.

3.3.4 Training Details

In the demonstrations, successful picks are more common than failed picks, resulting in an imbalanced dataset. For classification we undersample successes to create a more balanced dataset consisting of 975 successes and 961 failures.

3.3.5 Baselines and Experiments

Centroid: A heuristic baseline always grasping from the center of the masked object.

Random: A random baseline sampling points uniformly from within the masked pixels.

Imitation Learning (IL): A learned baseline using the initial grasping policy produced by

imitating the behavior of the human demonstrations.

Verification-in-the-Loop (IL+VitL): In this method, IVFP is used to verify grasps and retry failed grasps until success or no candidates grasps remain.

Reinforcement Learning (RL): In this method, the predictions from IVFP are used to fine-tune the grasp policy using reinforcement learning. We report results after training for 20 iterations and 50 iterations. In each iteration, we collect a batch of 64 grasps.

3.3.6 Evaluation Metrics

To quantify these approaches we report on the following two evaluation metrics:

Success Rate (SR) is the percentage of picks for which the target object was extracted successfully.

Units Per Hour (UPH) indicates how many target objects could be picked per hour, quantifying the speed at which the robot is picking.

3.3.7 Additional Experiments

To study the effect that interaction has on performance, we perform an ablation study where we compare accuracy of a model with access to the observations produced from interaction against a model with access to only passive observations. To evaluate the data throughput benefits and tradeoffs of our approach, we compare a 1 hour data collection with IVFP to a 1 hour data collection using a more typical collection pipeline wherein the robot fully executes each pick. We report on three metrics: picks collected per hour, human interventions per hour, and collected label accuracy.

Method	Seen objects		Unseen objects	
	SR	UPH	SR	UPH
Centroid	44.83%	36.08	-	-
Random	29.61%	25.2	-	-
IL	56.76%	45.92	49.16%	40.18
IL+ViTL	67.33%	43.76	57.51%	37.05
RL @ 20	69.16%	56.58	61.66%	48.8
RL @ 50	73.33%	58.4	62.51%	49.6

Table 3.1: Evaluation results of various grasping methods across two metrics: Success Rate (SR) and Units Per Hour (UPH).

Method	Picks/Hr	Interventions/Hr	Label Acc.
Full picks	82	24	100%
IVFP	158	6	96%

Table 3.2: Comparison of data throughput tradeoffs between data collection with full picks and with IVFP.

3.4 Results

All methods described in Section 3.3 were evaluated on both seen and unseen object sets. For each method, we evaluate over 10 trials each consisting of 12 picks.

In Table 3.1, we can see that using IVFP for verification in the loop results in significant performance gains. Qualitative examples can be seen in Figure 3.3. This method can be applied immediately as it requires no additional training. However, it comes at the cost of operation speed as verifying every grasp results in a decrease in UPH. Results of RL from 10 to 50 iterations show that we can improve performance by using IVFP to learn from experience.

The results of our data throughput experiment are summarized in Table 3.2, emphasizing that our approach can significantly reduce the burden of data collection with a minimal impact on collected label accuracy. In Table 3.3, the results of our ablation study on the effect of interaction show that interaction is crucial for classification and illustrate the tradeoff between interaction time and classifier accuracy.

Perception	Seen objects			Unseen objects		
	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.
Passive	0.5	0.56	46%	0.43	0.41	41%
Interactive (0.5s)	0.69	0.72	72%	0.61	0.62	64%
Interactive (1.0s)	0.79	0.81	81%	0.72	0.76	75%
Interactive (1.5s)	0.84	0.86	87%	0.81	0.83	83%
Interactive (2.0s)	0.94	0.93	94%	0.92	0.9	90%

Table 3.3: Performance of the learned classifier when interaction is used as compared to when passive perception is used.

3.5 Discussion

In this work, we presented an approach to grasping in clutter using Interactive Visual Failure Prediction (IVFP). In our approach, the robot interacts with the environment by performing a cyclic interactive probe designed to inform grasp success. We combine the interactive behavior with a visual classifier learned from human feedback to achieve IVFP. We perform experiments in the context of a real-world robot deployment showing that this approach both improves grasping performance and reduces the burden of data collection. While effective in our domain, our approach utilizes a fixed interaction policy which won't necessarily generalize to other domains. To address this limitation, exploring methods of learning interaction policies as in [50] is an exciting direction. Additionally, our task is performed in a relatively constrained contextual bandits setting and future work should explore how to apply IVFP on longer horizon problems with richer action spaces.

Chapter 4

One Shot Programming by Demonstration via Diffusion Features

General-purpose robots have the promise to automate tasks in many human-centric environments such as homes and workplaces. However, programming robots to robustly perform behaviors with every possible object in every possible environment is extremely challenging. Programming by Demonstration (PbD) is a popular approach that enables end-users to program new robot capabilities by simply demonstrating the desired behavior [17]. For robots deployed in human-centric environments, demonstration provides an intuitive way for end-users to teach robots new skills without having technical training or expertise in robotics. But this approach typically requires a large-scale and diverse set of demonstrations in order for the programmed capabilities to generalize to new environments and objects, which is not feasible for an end-user to provide. Ideally, an end-user could program robot capabilities by providing just a single demonstration of the desired behavior and those capabilities would generalize to new scenarios. For example, after demonstrating how to put a mug into a coffee machine, the robot should be able to repeat this task with other mugs even if they are visually distinct. Additionally, if the coffee machine and

mugs are re-arranged or moved to an entirely different location the robot should still be able to perform the demonstrated task.

Humans possess a remarkable ability to learn tasks from a single demonstration and to apply the learned behaviors to new situations [6, 14, 48, 99]. This is achieved in part by drawing on prior conceptual knowledge to infer the underlying structure of the task being demonstrated rather than directly mimicking the demonstrator's low-level actions [48]. For example, to learn new manipulation skills we primarily pay attention to interactions between the end-effector and objects rather than the relative motions within and between joints. By extracting the high level structure of the task rather than the low level actions, we are able to more easily transfer the task to new scenarios by identifying corresponding structure in new scenes.

Inspired by these insights, we propose a novel approach to PbD that enables programming generalizable robot manipulation skills from a single observed demonstration, illustrated in Figure **??**. Our approach draws on the prior conceptual knowledge encoded by pre-trained web-scale foundation models to both extract the salient structure from an observed demonstration and to identify the corresponding structure in new scenes. In particular, we utilize features from pre-trained diffusion models. While diffusion models are primarily models for image synthesis, they have been shown to implicitly encode rich information about the structure of the scene, objects, and object parts within an image. We show that within the context of a PbD framework, such capability provides an elegant mechanism to generalize observed demonstrations to new scenarios. We study the performance of our method across 14 tasks on a real robotic manipulator and find that our approach is surprisingly effective at a wide range of manipulation skills, while utilizing only off-the-shelf models with no additional fine-tuning required. We thoroughly analyze the generalization capabilities of our approach and study the contribution of diffusion features as compared to popular alternatives.

4.1 Related Work

4.1.1 Programming by Demonstration

Programming by Demonstration, also referred to as Learning from Demonstration or Imitation Learning, has been the subject of four decades of robotics research [17, 9]. Approaches are often categorized based on the method of providing demonstrations and in contrast to methods that require moving the robot (e.g. through teleoperation [112, 155, 154], kinesthetic teaching [75, 42, 5], or spoken commands [132, 37, 118]), in this work we focus on programming by passive observation, where the robot is programmed by observing a human perform the desired behavior [74, 49, 11, 138, 116]. This is particularly easy and intuitive for the user, requiring almost no training to perform. However, learning generalizable skills from passive observation is especially challenging and approaches typically either heavily restrict the domain or require a large and diverse set of demonstrations in order to scale to scenarios outside of the demonstrated examples. Some works take a user-guided approach to the generalization problem, where the user provides additional information to adapt the learned skills to new scenes [22, 36, 7, 32]. Another approach is to extract a reward function from the provided demonstration which can then be used to fine-tune the skill in novel scenes [153, 11, 116], but this requires additional training time to fine-tune the robot policy in new scenes. In this work, we propose to leverage large-scale visual foundation models off-the-shelf, with no additional fine-tuning, to extract robot manipulation skills from a single observed human demonstration and to apply those skills to new objects, viewpoints, and scenes.

4.1.2 Diffusion Models for Robotics

Diffusion models [115] have made great breakthroughs in generative tasks such as image and video synthesis [47, 29, 103, 45, 33]. Within robotics, diffusion models have been trained to generate actions for manipulation [26, 87, 94], navigation [117, 139], and human-robot collaboration [145, 93]. Additionally, pre-trained image diffusion models have been utilized to generate images used for robot training [146, 24, 127, 15] and planning [59, 150]. While diffusion models are primarily used for generative tasks, recent works show evidence that they implicitly encode rich information about the structure of objects and scenes in images [123, 77, 151]. Based on this insight, we propose to leverage features extracted from pre-trained generative image diffusion models within a PbD framework in order to find correspondences between structures observed in demonstration scenes and those observed in novel scenes.

4.2 Diffusion PbD

We present Diffusion-PbD, a robot PbD framework for synthesizing generalizable robot manipulation programs using only a single passively observed human demonstration. Our approach utilizes pre-trained visual foundation models to both extract salient structure from the observed demonstration and to find corresponding structure in novel scenes. Specifically, we use pre-trained models with strong hand-object priors to extract waypoints relative to observation-centric reference points, then we utilize pre-trained diffusion features to find corresponding reference points in novel settings. In the following sub-sections we first formalize the problem setting, then we provide a high-level overview of the approach, and finally we describe each phase of the approach.

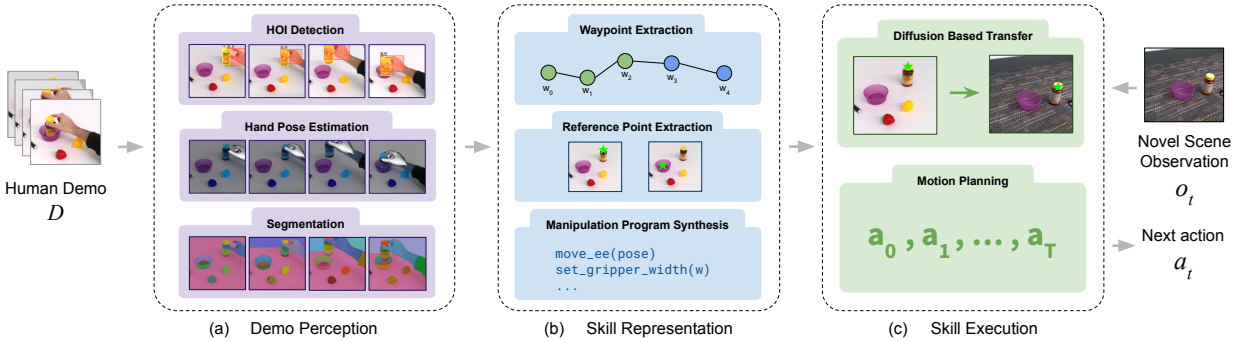

Figure 4.1: Diffusion-PbD composes a mixture of pre-trained web-scale foundation models to both extract salient structure from demonstration videos and to transfer that structure to new scenes. Diffusion-PbD is composed of three main phases: (1) human and object detection, (2) waypoint extraction, (3) skill execution. In the first phase, we pre-process the demonstration frames by detecting human hands and their interactions with objects in the scene. Next, we map these detections to waypoints and robot gripper configurations. We anchor the waypoints relative to observation-centric reference points. This representation allow us to map the skill to new scenes by finding corresponding reference points in the new observations.

4.2.1 Problem Formulation

In this work, we consider PbD for robotic manipulation tasks. Let $\mathcal{A}$ be the set of robot actions, and $\mathcal{S}$ the set of world states. We assume access to a human demonstration $D = \langle d_0, d_1, \ldots, d_{T_D} \rangle$ where each demonstration frame d_t is an RGB-D image at time t. Given a demonstration D and an initial state $s \in \mathcal{S}$, the goal is to generate an execution $\xi = \langle a_0, a_1, \ldots, a_{T_\xi} \rangle$, where $a_t \in \mathcal{A}$ is an action taken by the robot at time t. The initial state s is defined by the environment layout, the poses and states of all objects, and the pose and state of the robot. The robot does not directly have access to the initial state s, but only to an initial observation o. The initial observation $o = (I, K, D)$ includes an RGB-D camera image I, the robot's proprioceptive state K, and the demonstration D. The task is considered successful if the goal-conditions corresponding to the demonstration D are true at the end of execution.

4.2.2 Overview

Our PbD method composes a mixture of pre-trained web-scale foundation models to both extract salient structure from demonstration videos and to transfer that structure to new scenes. Our method has three main phases (see Figure 4.1): (1) demo perception, (2) skill representation, (3) skill execution. In the first phase, we pre-process the demonstration frames by detecting human hands and their interactions with objects in the scene. Next, we map those detections to waypoints and robot gripper configurations. We represent waypoints relative to reference points in the observation, which allows us to map the skill to new scenes by finding corresponding reference points in the new observations.

4.2.3 Demo Perception

For robot manipulation tasks, timesteps when the end-effector interacts with objects in the environment are particularly important. We process the demonstration D to extract information about the hands in the scene and the objects that they contact using 100DOH [108], a hand-object interaction model that has been pre-trained on 100K images extracted from a large-scale (131+ days) video dataset of humans interacting with objects. We use 100DOH to extract, for each demonstration frame d_t, a hand bounding box b_t^h and a boolean contact variable c_t indicating whether the hand is in contact with an object or not. For every frame where the hand is in contact with an object we additionally extract an object bounding box b_t^o. While bounding boxes give us the rough position of hands and objects in the scene, we look to obtain fine grained masks using Segment Anything Model (SAM) [64]. For each hand bounding box b_t^h and object bounding box b_t^o we prompt SAM to produce a hand mask m_t^h and object mask m_t^o. 3D perception of the scene is crucial for manipulation tasks, so we additionally produce a point cloud C_t for each demonstration frame d_t using the RGB-D image and camera intrinsics. To properly imitate the

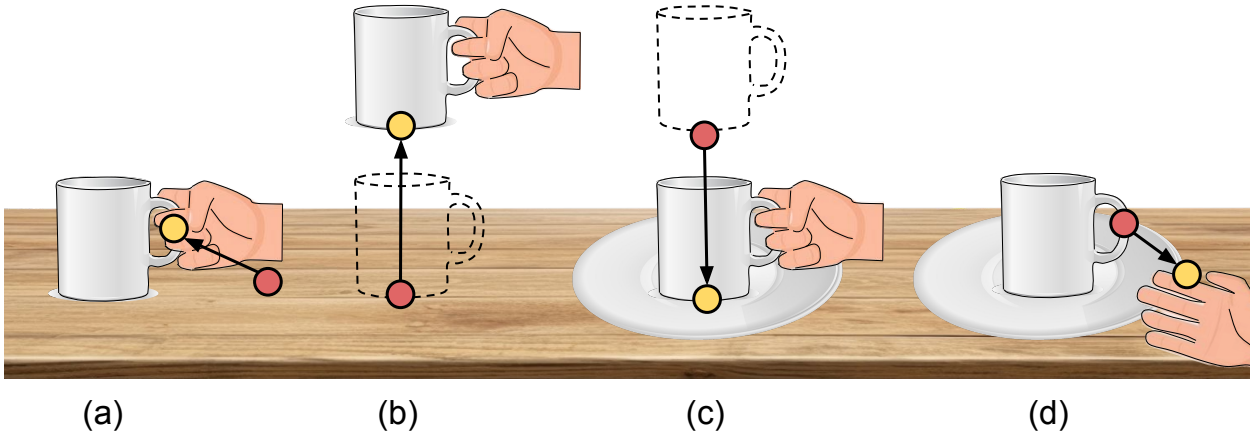

Figure 4.2: An illustration of the conditions used to identify key contact frames for an example where a cup is lifted by the handle and moved onto a plate. Key frames are extracted when (a) contact is made between the hand and the target object, (b) contact is broken between the target object and the environment (c) contact is made between the target object and the environment or (d) contact is broken between the hand and the target object.

grasp and interaction on a robot, the pose of the hand is important as well. We employ Mediapipe [76] for this purpose, and detect the human hand pose p_t represented as 21 landmarks following the topology in [113]. The two landmarks on the thumb and another two on the index finger are used to represent a parallel jaw gripper. We define r_t as the rotation of this gripper model and g_t as the distance between fingers.

4.2.4 Skill Representation

Inferring the hand-object interactions from the demonstration is useful, but ultimately we want to extract waypoints that can be executed by the robot. To accomplish this, we first identify and extract contiguous *contact sequences*, or clusters of timestamps where c_t is true. Because hand-object interaction detection can be noisy, we filter any sequences that span less than three timestamps, leaving only those that indicate sustained contact. We extract this set of contact sequences $\Sigma = \langle \sigma_0, \sigma_1, \ldots, \sigma_{L_\Sigma} \rangle$ where each contact sequence is initially represented as a start

timestamp and end timestamp $\sigma = (t_{start}, t_{end})$. We additionally compute a pre-contact timestamp t_{pre} by backing up from the start of contact t_{start} until the hand mask m_t^h no longer overlaps with the object mask m_t^o to obtain timestamp t_{pre}. For each contact sequence in Σ, we extract a set of waypoints where each waypoint $w_i = (P_i, g_i, t_i)$ is made up of a 6-DOF pose P_i, gripper width g_i, and timestamp t_i. We first define a waypoint at the start of interaction, w_{start} using the hand pose landmarks at the start of contact $p_{t_{start}}$. The two pose landmarks on the thumb and another two on the index finger are used to represent a parallel jaw gripper. These points are lifted into 3D using the depth map and averaged to obtain our contact point, which is combined with the gripper rotation $r_{t_{start}}$ to obtain our contact pose P_{start} and waypoint $w_{start} = (P_{start}, g_{t_{start}})$. We additionally compute a pre-contact waypoint w_{pre}, the points from the thumb and index finger landmarks at t_{pre} are again lifted into 3D, averaged, and combined with $r_{t_{pre}}$ to produce pose P_{pre} and waypoint $w_{pre} = (P_{pre}, g_{t_{pre}})$. As illustrated in Figure 4.2, we identify additional waypoints centered around timesteps where contact is made or broken between the target object and the environment. Finally, we define a waypoint at the end of interaction w_{end} by repeating this process with the pose landmarks at the end of contact $p_{t_{end}}$. At the end of this process each contact sequence σ is represented as a set of waypoints $\sigma = \langle w_0, w_1, \ldots, w_{L_\sigma} \rangle$.

The set of contact sequences Σ contains waypoints to reproduce skills in the current scene, but we desire to reproduce skills in novel scenes, including those with novel viewpoints, object configurations, and objects. In this work, we aim to leverage the features from pre-trained image diffusion models for the purpose of re-identifying key waypoints in new scenes. To that end, we extract waypoints relative to *observation-centric* reference points. To obtain reference points, we look for key frames where contact is made between the hand and the target object or between the target object and the environment. To obtain a reference point for a key frame where contact is made between the hand and target object we extract a 3D point from the pose at the start of contact P_{start}. We project the 3D point onto the image at timestep t_{pre} to obtain a 2D reference

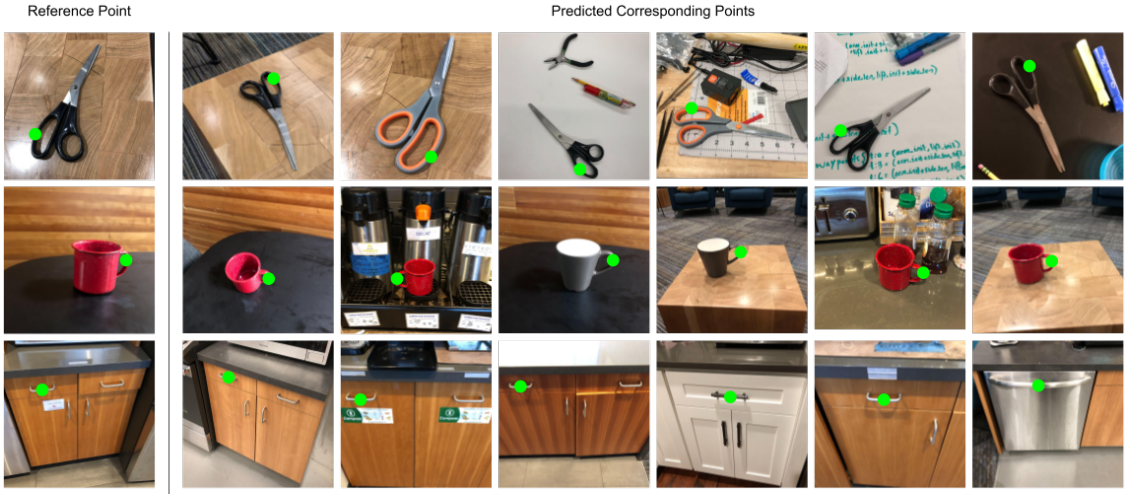

Figure 4.3: In Diffusion-PbD, features from a pre-trained Stable Diffusion [104] image model are utilized to transfer demonstrated contact points to new scenes. The examples in this figure show the effectiveness of this method at finding corresponding points in novel viewpoints, objects, and scenes. The reference points on the left are extracted from human demonstrations, and the corresponding points on the right are predicted through the use of diffusion features.

point in image space. To obtain a reference point for a key frame where contact is made between the target object and the environment we average the points in contact to obtain a 3D point and project the resulting point on the the image at timestep t_{pre} to obtain a 2D reference point. After identifying reference points, we recompute the pose in all waypoints using relative translation from the nearest preceding reference point.

4.2.5 Skill Execution

To apply skills to new scenes we first map our reference points to the novel observations using the popular Stable Diffusion (SD) [104] image foundation model. SD has been pre-trained on billions of images and the intermediate-layer features of the model have been shown to implicitly encode rich information about the structure of objects and scenes in an image. In this work, we propose to utilize these features within a PbD framework for robust reference point generalization to unseen

viewpoints, objects, and scenes as illustrated in Figure 4.3. For each contact sequence in Σ, we use SD to extract the diffusion features of our reference demonstration frame $d_{t_{pre}}$ and the first observation image in the new scene I. The features are generated by adding noise to the images, feeding the images through the network of SD, and extracting the intermediate layer activations. For more details we refer the reader to [123]. Through this process we obtain two diffusion feature maps F_{ref} and F_{target}. For every waypoint w_i in σ, we compare the cosine similarity of the two features maps and identify the point in F_{target} that is most similar to the reference point in F_{ref}. This point is then lifted into 3D using the depth map from I to produce a 3D point in the new scene $\hat{P}_i$ and new waypoint $\hat{w}_i = (\hat{P}_i, g_i)$. Ultimately we obtain a set of waypoints for the new scene $\hat{\sigma} = (\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_{L_\sigma})$. We convert each contact sequence to a manipulation program for execution on the robot wherein the end-effector motion and gripper state are commanded according to the waypoints in $\hat{\sigma}$. To generate the motion between waypoints, we use a collision-free motion planner to generate a trajectory of robot actions for reaching the next desired waypoint goal. Specifically, we use the GPU accelerated motion generation library cuRobo [120]. After successfully reaching every waypoint goal, this process is repeated for every contact sequence in Σ.

4.3 Experiments

To evaluate our approach, we conduct a series of real world experiments across 5 indoor environments as illustrated in Figure 5.3. In our experiments we seek to answer the following research questions: 1) Is Diffusion-PbD practical for a wide range of robot manipulation tasks? 2) How effective is Diffusion-PbD at applying demonstrated manipulation tasks to new viewpoints, objects, and scenes? 3) To what extent do diffusion features contribute to the effectiveness?

4.3.1 Hardware and Environments

We use the Stretch RE2 robot [61] for our experiments. The robot's mobile base, arm lift, and telescoping arm are moved in conjunction to reach 6-DOF target waypoints. The robot's end effector is a parallel-jaw gripper with rubber fingertips. An Intel RealSense D435i RGB-D camera is mounted to the frame which is used both to record demonstrations and to provide observations during execution. One of the authors initialized scenes and categorized tasks as success or failed based on the criteria in Section IV-B.

4.3.2 Evaluation Tasks

We evaluate our approach using 14 different real world manipulation tasks. We design our evaluation tasks to cover a wide range of contact-rich manipulation behaviors involving prehensile and non-prehensile motions. The tasks range from rearranging objects, to multi-step extraction from cluttered scenes, to tool use, to manipulation of deformable and articulated objects. Below, we describe each of the tasks and how success is defined for each task.

Pick-and-place

In this task, the robot picks up a bottle by its top and places it into a bowl. The task is successful if the robot grasps from the top of the bottle and the bottle is contained inside of the bowl at the end of execution.

Bookshelf extraction

In this task, the robot is required to do both non-prehensile and prehensile motions to successfully extract a slender object from a bookshelf. The target object is densely packed into the shelf, so the robot must first tip the object with a pushing motion from the top before grasping and extracting

the object. The task is successful when the robot extracts the target object without displacing any other objects from the shelf.

Occluded pick

For this task, a target object is occluded by another object in the initial scene. The robot must first push the occluding object out of the way using non-prehensile motion and then extract the target object. The task is successful if the object is extracted by the robot.

Occluded place

For this task, the robot must use non-prehensile motion to push an object out of the way to make room for the target object on a surface. The target object is then picked and placed onto the surface. The task is successful when the target object rests on the target surface.

Open drawer

In this task, the robot is required to open a drawer. This requires a precise grasp of the drawer handle and careful imitation of the demonstrated trajectory to open the drawer. The task is successful if the drawer is open at the end of execution.

Close drawer

In this task, the robot is required to close a drawer. The task is successful if the drawer is closed at the end of execution.

Stack blocks

This task demonstrates a manipulation program with a multi-step horizon. The robot must stack a set of three colored blocks in the same order as the demonstration. The task is successful when

the blocks are stacked in a stable column following the order given by the demonstration.

Unstack blocks

Another multi-step horizon task, the robot must unstack a set of three colored blocks. The task is successful when none of the blocks are stacked.

Clear table into drawer

Our longest horizon task where the robot must first open a drawer by the handle, then pick objects one by one off of a counter and place them into the drawer, and finally close the drawer. The task is successful when the drawer is closed with all items from the counter top contained inside.

Unplug charger

In this task the robot must grasp and pull a laptop charger to remove it from a power outlet socket. The task is successful when the charger is removed from the socket.

Assemble bento

In this task the robot must pick and place food items into a bento box, putting the items into the same sections of the box as the demonstrator.

Push chair

In this task the robot must perform a non-prehensile motion to push a chair into a table.

Clean whiteboard

This task demonstrates a manipulation program with tool use. The robot must first grasp a cloth, then follow the demonstrated trajectory to clean a marking off of a whiteboard using the cloth.

Figure 4.4: We evaluate Diffusion-PbD using a Stretch RE2 robot to perform 14 real world manipulation tasks across 5 visually distinct environments. We show that this approach is effective for single-shot imitation of a wide range of manipulation tasks and generalizes to novel viewpoints, objects, and scenes.

The task is successful when the whiteboard is cleaned.

Fold towel

This task demonstrates a manipulation program with deformable objects. The robot must first grasp the corner of a towel, then follow the demonstrated trajectory to fold the towel. The task is successful when the towel is folded.

4.4 Results

To demonstrate the ability of Diffusion-PbD to synthesize a wide variety of robot manipulation skills, we perform experiments on a set of 14 tasks, ranging from pick-and-place, to tool use, to manipulation of deformable and articulated objects. The results are summarized in Table 4.1. For each task, we report results averaged across 15 trials, with a new viewpoint and human demonstration used for each trial. Diffusion-PbD is able to complete all 14 tasks with an average success rate of 81.3%.

Generalization to Unseen Scenarios: To understand the ability of Diffusion-PbD to apply

demonstrated manipulation tasks to new viewpoints, objects, and scenes we perform a deeper analysis with a representative subset of the manipulation tasks: pick-and-place, open drawer, fold towel, and clear counter into drawer. For each task, we perform additional trials across three novel scenarios. First, we perform 15 trials from unseen viewpoints. Then we perform 15 trials with unseen objects: for the pick-and-place task bottles distinct in appearance and size are used, for the drawer task we use a visually distinct drawer, and for the towel folding task we use towels of varying colors and sizes. Finally, we perform 15 trials from an entirely unseen environment. For each evaluation scenario, we report the average across the 15 trials. The results summarized in Table 4.2 show the strong generalization ability of Diffusion-PbD.

Contribution of Diffusion Features: To evaluate the major design decision of using features from SD within our framework, we conduct additional experiments using two other pre-trained feature spaces commonly used for correspondence matching in similar robotic applications: CLIP [97], and DINOv2 [95]. The results in Table 4.1 and Table 4.2 highlight the importance of this design decision as features from SD enable a higher success rate on a range of manipulation tasks and across a range of previously unseen scenarios. Figure 4.5 qualitatively illustrates this advantage, with examples of the strong correspondence matching enabled by the SD features.

Failure Analysis: While the results show that this approach can be practical for all 14 benchmarked manipulation tasks, there is still room for improvement. To better understand the failure cases, we analyze the failures in Figure 4.6, finding that the most common source of failure happens during demo perception due to inaccuracies in hand-object detection models, highlighting detection improvements as an important area for future work.

Task	Success Rate		
	CLIP	DINOv2	SD
Pick-and-place	0.86	0.86	**0.93**
Bookshelf pick	0.40	0.53	**0.67**
Occluded pick	0.40	0.60	**0.80**
Occluded place	0.33	0.80	**0.87**
Open drawer	0.53	0.73	**0.80**
Close drawer	0.40	**0.73**	**0.73**
Clear counter into drawer	0.33	0.66	**0.73**
Stack blocks	0.53	**0.80**	**0.80**
Unstack blocks	0.40	0.80	**0.87**
Unplug charger	0.66	**0.93**	**0.93**
Assemble bento	0.66	0.66	**0.80**
Push chair	0.86	0.80	**0.93**
Clean whiteboard	0.66	**0.73**	**0.73**
Fold towel	0.60	0.73	**0.80**

Table 4.1: We present a set of evaluations on 14 real world tasks. For each task the robot must imitate a human demonstration.

4.5 Discussion

We propose Diffusion-PbD, a novel method for robot PbD that can synthesize generalizable robot manipulation programs from observing a single human demonstration. Our method utilizes pre-trained image foundation models off-the-shelf to both extract salient structure from human demonstrations and to transfer that structure to novel scenes. We perform an evaluation on a Stretch RE2 robot and demonstrate the ability of our approach to synthesize robot manipulation programs for a wide-range of different manipulation tasks. Our analysis shows that Diffusion-PbD is effective at generalizing demonstrated skills to unseen viewpoints, objects, and scenes, and highlights the utility of diffusion features for robot PbD.

Despite the promising results, Diffusion-PbD has multiple important limitations. First, our approach relies on sampling a set of waypoints from the provided demonstration. While this representation allows our approach to imitate a wide variety of manipulation tasks, some tasks

Task	Eval Scenario	Success Rate		
		CLIP	DINOv2	SD
Pick-and-place	Canonical	0.86	0.86	**0.93**
	Unseen Viewpoint	0.66	0.73	**0.80**
	Unseen Objects	0.33	**0.73**	**0.73**
	Unseen Scene	0.66	**0.86**	**0.86**
Open drawer	Canonical	0.60	0.73	**0.80**
	Unseen Viewpoint	0.40	0.66	**0.80**
	Unseen Objects	0.53	0.66	**0.73**
	Unseen Scene	0.40	0.73	**0.80**
Fold towel	Canonical	0.60	0.73	**0.80**
	Unseen Viewpoint	0.53	0.53	**0.73**
	Unseen Objects	0.33	0.66	**0.80**
	Unseen Scene	0.33	**0.73**	**0.73**
Clear counter	Canonical	0.33	0.66	**0.73**
	Unseen Viewpoint	0.26	0.66	**0.73**
	Unseen Objects	0.26	0.53	**0.66**
	Unseen Scene	0.13	0.53	**0.66**

Table 4.2: To study the robustness of Diffusion-PbD, we evaluate a representative subset of tasks on unseen viewpoints, unseen objects, and unseen scenes.

may need a more densely sampled set of waypoints or alternative trajectory representations for finer grained manipulation which is an exciting direction for future work. Our approach uses open-loop execution of actions and could be extended to use dynamic motion generation in order to handle dynamic disturbances or changes to the environment during execution. Additionally, our approach assumes that viable reference points are visible in new scenes, and future work should explore strategies for handling missing or occluded objects.

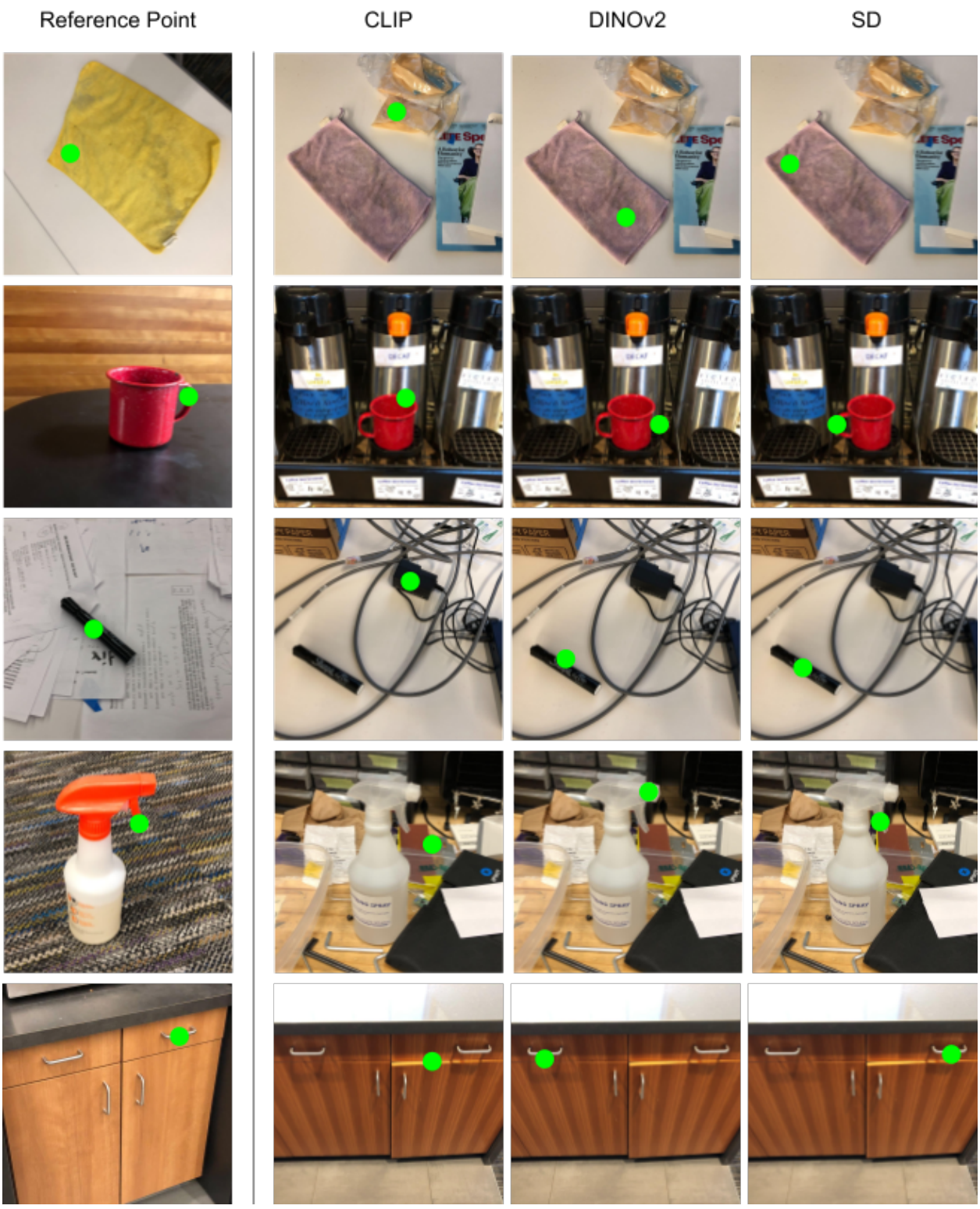

Figure 4.5: Qualitative comparison of point correspondences using features from Stable Diffusion [104], DINOv2 [95], and CLIP [97] for scenes with various visual distinctions from the reference and various amounts of clutter.

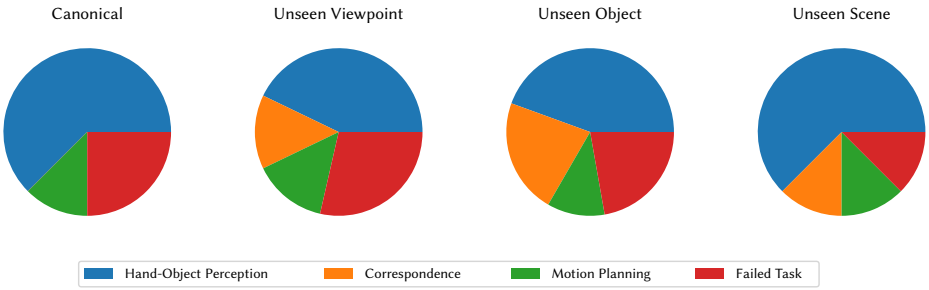

Figure 4.6: The distribution of failures for pick-and-place, open drawer, and fold towel tasks across unseen viewpoints, unseen objects, and unseen scenes. Executions can fail due to errors in hand-object perception, errors in correspondence matching, failure to motion plan, or failure to meet the task requirements.

Chapter 5

Teaching Robots with Show and Tell

General purpose robots have the potential to enhance productivity and reliability in human-centric, task-oriented settings such as kitchens, warehouses, and offices, but one of the key challenges to achieving this potential is that each environment, user, and task combination demands tailored behavior from the robot. Programming by Demonstration (PbD) is a popular approach to this challenge, enabling end-users to program robots for personalized tasks and environments by providing demonstrations of the desired behavior. However, existing PbD techniques typically require a large number of demonstrations or are unable to extract high-level task information related to control flow and action parameterization from the demonstrations. Natural language is another popular method for personalizing robot behavior, but low-level details can be cumbersome and error prone to communicate with language alone, and most works assume a fixed set of low-level primitives are available to be composed by the language instructions, limiting the scale and complexity of tasks that can be programmed.

Studies in human observational learning show that humans use both language and demonstration when teaching each other new tasks, with both communication modalities playing important roles [6, 119]. Language allows us to transmit abstract information, while demonstrations instan-

tiate that information in concrete examples [28]. Inspired by this insight, we envision a more natural and intuitive PbD system, where end-users can program robots for personalized tasks and environments by flexibly using both language and visual demonstrations.

In this work, we propose SHOWTELL – a system that enables end-users to teach robots new tasks the same way they would teach another person, by visually demonstrating and verbally describing what they are doing as they demonstrate. SHOWTELL is modular, composing a set of pre-trained large language models (LLMs) and vision-language models (VLMs) to synthesize robot policies that can jointly reason about the language and visual components of the demonstrations and execute the demonstrated behavior in novel scenes. SHOWTELL is designed to be generalizable across a wide range of tasks and environments, and to be intuitive and natural for end-users to use, while requiring only a single demonstration with no additional training or fine-tuning.

While both language and demonstration are traditionally popular interfaces for task planning, there has been comparatively little attention paid to combining the two. Most existing works attempt to train end-to-end models, an approach that is difficult to scale because it requires enough training examples to implicitly learn to understand all demonstrations and perform all tasks within the forward pass of a neural network. In contrast, our approach requires no additional training, utilizing the vast knowledge encoded in off-the-shelf foundation models. Our approach is neuro-symbolic in that it aims to combine the strengths of neural networks, which excel at learning from data, with symbolic reasoning, which excels at manipulating symbols and logical rules, to create a more robust robot PbD system that is modular, allowing us to scale to new tasks by composing existing modules in novel ways, and also also interpretable, allowing us to understand the reasoning behind the generated policies and to easily modify or extend them.

Through real-world robot experiments, we validate our approach, showing that SHOWTELL is able to synthesize robot policies from language and visual demonstrations including tasks that require high level logic such as conditions, iteration, and segmentation. We show that our

approach out-performs a state-of-the-art baseline on a variety of tasks, and that it is able to generalize to new objects and environments, while requiring only a single demonstration with no additional training or fine-tuning. We believe that our approach has the potential to significantly improve the usability and performance of robot PbD systems, and to enable robots to learn from demonstrations that are more natural and intuitive for end-users.

5.1 Related Work

Programming by Demonstration. Programming by Demonstration, also referred to as Learning from Demonstration or Imitation Learning, has been the subject of four decades of robotics research [17, 9]. Approaches are often categorized based on the method of providing demonstrations and in contrast to methods that require moving the robot (e.g. through teleoperation [112, 155, 154] or kinesthetic teaching [75, 42, 5]), or the use of specialized demonstration hardware [83], in this work we focus on programming by passive observation, where the robot is programmed by observing a human perform the desired behavior [74, 49, 11, 138, 116, 55]. Most existing works attempt to train end-to-end models, an approach that is difficult to scale because it requires enough training examples to implicitly learn to understand all demonstrations and perform all tasks within the forward pass of a neural network. Instead, we propose to leverage the prior knowledge encoded in pre-trained foundation models to synthesize programs that can both reason about demonstrations and execute the demonstrated behavior on a robot with novel scenes and objects.

LLMs for Task and Motion Planning. With the advent of large-scale pre-trained language models, there has been growing interest in using these models for robotics tasks. A large body of work has focused on planning and reasoning from text-based natural language instructions [4, 53, 102, 134, 73]. These works typically output their plans as a sequence of robot actions, but recent approaches show the benefits of using LLMs to synthesize code with logical constructs that

can be executed by a robot [69, 114, 51]. While much progress has been made in synthesizing code policies from text-based instructions, there has been comparatively little work on synthesizing code policies from visual demonstrations. **(author?)** [130] assume that a demonstration has been converted to a textual description, and focus on generating policies from the text. In contrast, we focus on generating policies directly from visual demonstrations. Most similar to our work, **(author?)** [128] propose to use a VLM to summarize visual demonstrations and generate policies from the resulting summaries. In contrast, we propose to synthesize programs that can reason about the visual demonstration, which allows us to handle more complex demonstrations, better align the language with the visual components of the demonstration, and more easily interpret the reasoning behind the generated policies.

LLMs for Visual Reasoning. Visual reasoning approaches have typically used end-to-end trained models, but recent works have shown that pre-trained LLMs can be leveraged to accomplish state-of-the-art for many visual reasoning tasks. Early iterations represent the visual information from images as text via captions, objects, and attributes and feed this textual representation to LLMs along with task instructions and in-context examples [141]. **(author?)** [148] propose a modular approach wherein the LLM leverages other pre-trained models such as vision-language models (VLMs) and audio-language models through program generation. More recent works have scaled this idea to additional tasks including knowledge tagging, image editing, and causal/temporal reasoning over videos [40, 121]. In this work we seek to leverage LLMs for visual reasoning in the context of robot PbD by using the LLM to generate modular programs that jointly reason over video demonstrations and spoken language instructions to understand the task being demonstrated and ground that understanding to robot actions in new scene observations.

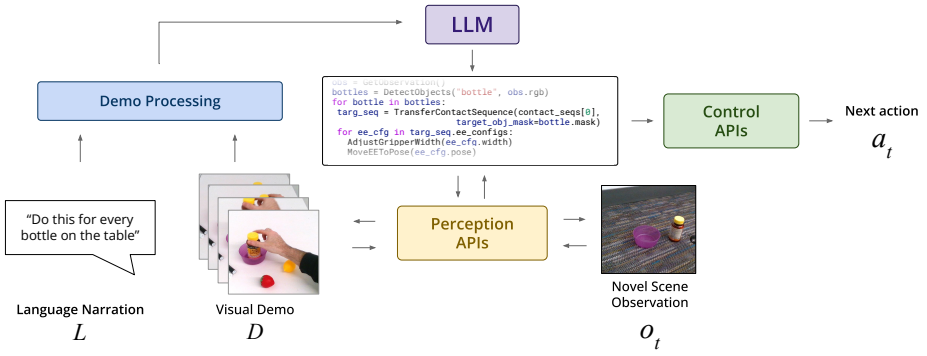

Figure 5.1: An overview of the SHOWTELL framework. First, the visual and spoken components of the demo are processed and fed into an LLM. The LLM synthesizes a modular program that can jointly reason about the provided demonstration and novel observations to transfer the demonstrated skill to new scenes.

5.2 Method

We present SHOWTELL, a neuro-symbolic robot PbD framework for synthesizing modular, generalizable, and interpretable robot manipulation programs from visual demonstrations and natural language. Our approach, illustrated in Figure 5.1, requires only a single demonstration with no additional training or fine-tuning, as it utilizes a combination of pre-trained foundation models and hand-engineered components to reason about observed demonstrations and to transfer learned skills to new scenes. In the following sub-sections we first formalize the problem setting, then we provide a high-level overview of the approach, and finally we describe each phase of the approach.

5.2.1 Problem Formulation

In this work, we consider multi-modal PbD for robotic manipulation tasks. Let $\mathcal{A}$ be the set of robot actions, and $\mathcal{S}$ the set of world states. We assume access to a human demonstration consisting of visual demonstration component $D = \langle d_0, d_1, \ldots, d_{T_D} \rangle$, where each demonstration frame d_t is an RGB-D image at time t, and a spoken language component L. Given the demonstration D, language L, and an initial state $s_0 \in \mathcal{S}$, the goal is to generate an execution $\xi = \langle s_0, a_0, s_1, a_1, \ldots, s_{T_\xi}, a_{T_\xi} \rangle$,

where $a_t \in \mathcal{A}$ is an action taken by the robot at time t, $s_t \in \mathcal{S}$ is the state before taking a_t, and $s_{t+1} = \mathcal{T}(s_t, a_t)$ under environment dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. The state s_t is defined by the environment layout, the poses and states of all objects, and the pose and state of the robot. The robot does not directly have access to the state s_t, but only to an observation o_t. An observation $o_t = (I_t, K_t)$ includes an RGB-D camera image I_t and the robot's proprioceptive state K_t. The task is considered successful if the goal-conditions corresponding to the demonstration D and spoken language L are true at the final state s_{T_ξ}.

5.2.2 Overview

Our method utilizes a mixture of pre-trained foundation models and hand-engineered modules to both understand the provided demonstration and to transfer that understanding to new scenes. We first pre-process the demonstration to make relevant information readily accessible for program synthesis. Next, an LLM is used to compose a set of modules into a program. The modules available to the LLM include tools for visual and spatio-temporal reasoning about the video demonstration frames, tools for aligning the spoken language of the demonstration with the visual demonstration, tools for transferring the demonstrated skill to the current environment, and tools for controlling the robot.

5.2.3 Pre-processing

The first step in our approach is to pre-process the demonstration to make relevant information readily accessible to the LLM. The spoken component of the demonstration, L, is transcribed using the Whisper speech-to-text model [98]. The transcription is saved and uttered words are indexed by their timestamp relative to the start of the demonstration. The visual component of the demonstration is processed to detect human hands and their interactions with objects in the

scene. We utilize the Mediapipe [76] hand landmark detection model, and detect the human hand pose represented as 21 landmarks following the topology in [113]. The two landmarks on the thumb and another two on the index finger are used to represent a parallel jaw robot gripper.

For robot manipulation tasks, timesteps in which the end-effector interacts with objects in the environment are particularly important, so we seek to identify and extract contiguous *contact sequences*, or clusters of timestamps where the hand is in contact with an object, from the demonstration D. We extract information about the hands in the scene and the objects that they contact using 100DOH [108], a hand-object interaction model that has been pre-trained on 100K images extracted from a large-scale video dataset of humans interacting with objects. We use 100DOH to extract, for each demonstration frame, a hand bounding box, in-contact object bounding box, and a boolean contact variable indicating whether the hand is in contact with an object or not. We obtain fine grained masks for both hands and objects using Segment Anything Model (SAM) [64] with bounding boxes provided as prompts. 3D perception of the scene is crucial for manipulation tasks, so we additionally produce a point cloud for each demonstration frame using the RGB-D image and camera intrinsics.

5.2.4 Program Synthesis

ShowTell uses a powerful code-generating language model (GPT-4 [2]) to synthesize programs that can both reason about a demonstration and execute the demonstrated behavior in a new scene. The LLM is provided with a prompt that consists of import statements and API documentation that specifies the avail-

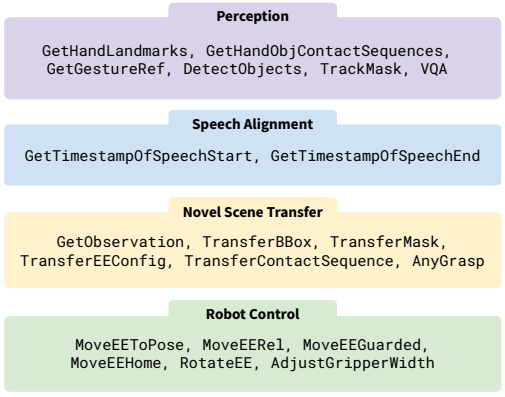

Figure 5.2: Taxonomy of modules available to the synthesized neuro-symbolic programs.

able functions, a brief summary of contact se-
quences extracted during the pre-processing
phase, and the full transcribed narration. The program is run with the Python interpreter, allowing the use of control flow tools like for loops, conditionals like if/else, built in functions like sort, and built in modules such as datetime or math. All of the information extracted during the pre-processing phase is made available in addition to a suite of modules. The additional modules made available to the program include a mixture of pre-trained foundation models and hand-engineered modules, including modules for perception, speech alignment, transferring to novel scenes, and controlling the robot. In the following sections we describe each of the custom modules and their APIs and a taxonomy of the modules is provided in Figure 5.2.

Perception

The perception modules made available to the neuro-symbolic program enable extracting information about both the visual demonstration, D, and the new scene observations, $\langle o_0, o_1, \ldots, o_{T_\xi} \rangle$. The perception modules include APIs for accessing visual information made available in the pre-processing step via `GetHandObjectContactSequences`. The perception modules also include APIs for free-form visual reasoning, `DetectObjects`, `TrackMask`, and `VQA`. The `DetectObjects` module uses ViLD [39], a pre-trained open-world object masking VLM. Detected masks can be tracked to new frames and scenes with the `TrackMask` module, which uses the XMem mask tracking model under the hood [25]. The `VQA` module is a pre-trained visual question answering VLM, BLIP-v2 [68], that can be used to answer questions about the visual scene.

Language Alignment

To understand a multi-modal demonstration, the robot must align the spoken language of the demonstration with the visual demonstration. For this purpose, we provide the LLM with two

modules, `GetTimestampOfSpeechStart` and `GetTimestampOfSpeechEnd`, which are used to extract the start and end timestamps of spoken words from the language component of the demonstration, L, and enable the program to align the language with timestamps corresponding to frames in the visual demonstration, D.

Novel Scene Transfer

Ultimately, the goal is to transfer the policy generated from the demonstration to new environments. To do this, the robot must match the visual demonstration D to the current environment observed in o_t. The `TransferBBox`, `TransferMask`, `TransferEEConfig`, `TransferContactSequence` modules are used to transfer detections to new scene observations. These modules find corresponding reference points by leveraging features from Stable Diffusion [103], a pre-trained image diffusion model which have been shown to implicitly encode rich information about the structure of objects within an image, and have been shown to be highly effective for finding corresponding points for visual reasoning tasks [123, 77, 41]. Using these features, we can match similar points on within-category objects in addition to exact points. For example, after demonstrating how to pick up a mug by its handle, the robot should be able to repeat this skill for visually distinct mugs and mugs of different sizes.

In order to facilitate grasping of objects in the new scene, we provide the `AnyGrasp` module, which is used to find the best grasp configuration for a particular object mask in the scene. This module uses a pre-trained grasp prediction model [34] to predict the best grasp configuration for a particular object based on its point cloud, and can be used to find grasp configurations for objects that were not present in the demonstration.

Robot Control

To control the robot we provide a set of modules to specify end-effector goal poses and configurations. The modules `MoveEEToPose` is used to specify a goal EE pose in 3D space, `MoveEERel` is used to specify a goal relative to the current EE pose, and `MoveEEGuarded` performs a guarded movement along a given vector until contact occurs. The module `RotateEE` is used to specify a goal EE rotation. And finally, `AdjustGripperWidth` is used to specify a desired gripper width to open and close the gripper. These modules are used to specify the robot's end-effector goal poses and configurations, which are used to generate the robot's actions during program execution as described in Section 5.2.5.

5.2.5 Skill Execution

After program synthesis, the resulting program is run with a Python interpreter and its execution is a simple Python call. During execution, correspondence matching modules are evaluated to transfer reference points identified in the demonstration to the new scene observation, and all waypoints are redefined relative to the reference points in the new scene. To interpolate robot motion between end-effector goal poses, we use a collision-free motion planner to generate a trajectory of robot actions for reaching the next desired waypoint goal. Specifically, we use the GPU accelerated motion generation library cuRobo [120].

5.3 Experimental Setup

Hardware and Environments: To evaluate our approach, we conduct a series of real world experiments with a Stretch RE2 robot [61] across 5 indoor environments as illustrated in Figure 5.3. The robot's mobile base, arm lift, and telescoping arm are moved in conjunction to reach

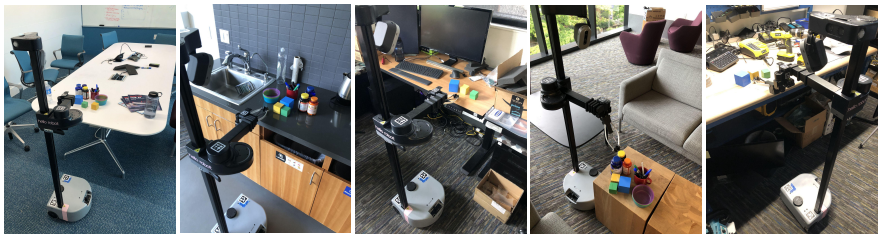

Figure 5.3: We evaluate ShowTell using a Stretch RE2 robot to perform 16 real world manipulation tasks across 5 visually distinct environments including a conference room, a kitchen, a classroom, an office lounge, and a cluttered workbench. We show that this approach is effective for teaching manipulation tasks requiring high level logic such as conditions, iteration, and segmentation.

6-DOF target waypoints. The robot's end effector is a parallel-jaw gripper with rubber fingertips. An Intel RealSense D435i RGB-D camera is mounted to the frame which is used both to record demonstrations and to provide observations during execution.

Baselines and Experiments: We evaluate our approach against GPT4-V-Robot [128], a state-of-the-art method for robot task planning from language and visual demonstrations using LLMs. In this baseline approach, demonstration frames are directly fed to a VLM (GPT4-V [2]), which generates a summary that is used by the LLM to generate a policy. To study the contribution of the visual and spoken components of the demonstration we perform a series of ablation experiments. First we evaluate a language-only baseline, ShowTell-NoVis, with no access to the visual demonstration, similar to the approach proposed by **(author?)** [69]. We also include a vision-only baseline, ShowTell-NoLang, which uses the visual demonstration only to generate policies without access to the language component.

Evaluation Tasks: We evaluate ShowTell using a set of 16 real world manipulation tasks across 5 visually distinct environments. We group evaluation tasks based on the challenge conditions they present. First we evaluate with simple demonstrations that are straightforward to follow, including pick and place, stacking cubes, and opening drawers. Next we evaluate tasks involving control flow including iterative tasks, for example *"move all of the blocks over to this*

container", and tasks with conditionals, for example *"open the drawer only if it is closed"*. Finally, we evaluate tasks that require segmenting the demonstration, for example if the demonstrator makes an error and corrects themselves, the robot is required to segment the demonstration into correct and incorrect segments to generate a policy that correctly imitates only the desired behavior. All of the tasks, including the simple tasks, can require the robot to jointly reason about the visual and spoken components of the demonstration to resolve ambiguities. For example, an instruction like *"pick up this mug and move it over here"* requires aligning the spoken instruction with the visual demonstration to determine which mug to pick up and where to move it.

Metrics: We use two primary metrics to evaluate system performnace: *Success Rate (SR)* and *Goal Condition Recall (GCR)*. The task-relevant goal conditions are the set of state changes that must be satisfied at the end of an episode for the task to be considered successful. *SR* is the fraction of rollouts for which the object positions and state changes completely satisfy the task goal-conditions at the end of the action sequence. *GCR* is the fraction of goal-conditions successfully completed at the end of an episode to those necessary to have finished a task.

5.4 Results

In our experiments we seek to answer the following research questions: 1) Is SHOWTELL practical for teaching a range of robot manipulation skills including those that require high level logic like conditions, iteration, and segmentation? 2) How does SHOWTELL compare to existing state-of-the-art methods for generating robot policies from language and visual demonstrations? 3) Is SHOWTELL able to generalize to unseen environments and within category objects? 4) To what extent do the visual and spoken components of the demonstration contribute to the performance?

To evaluate the performance of SHOWTELL, we perform a total of 50 rollout trials per task: 10 demonstrations are provided (2 demonstrations in each environment) and 5 rollouts are performed

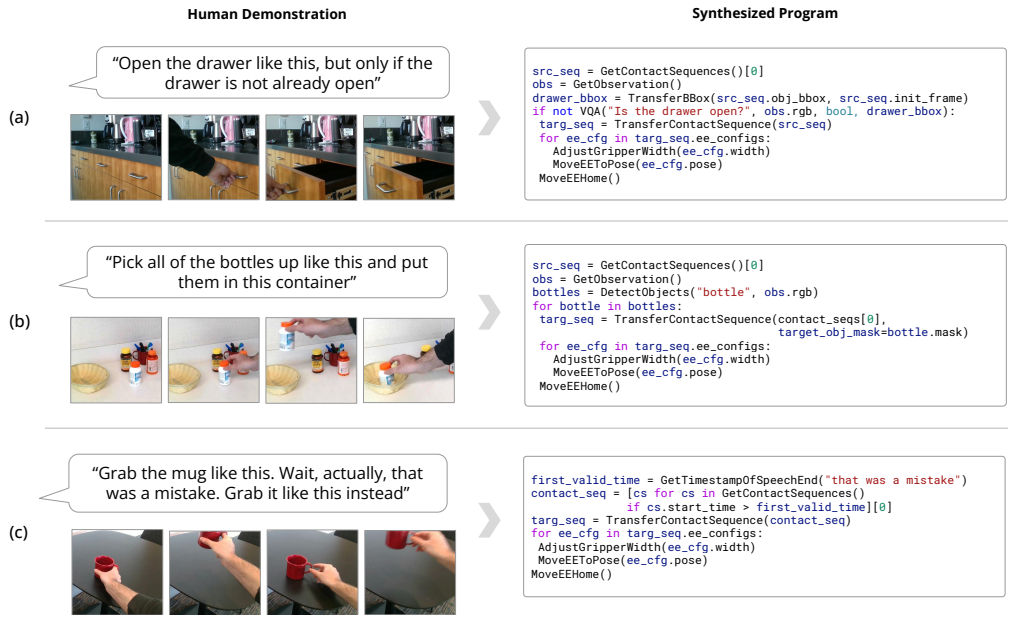

Figure 5.4: Qualitative examples of code synthesized by ShowTell for a set of representative demonstrations. The representative demonstrations show the ability to follow high level logic including (a) conditionals (b) iteration and (c) segmentation.

for each demonstration, resulting in a total of 50 rollouts per task. We group tasks based on challenge conditions as described in section 5.3 and for each task family, we report the average SR and GCR across all rollouts.

The results summarized in Table 5.1 show that ShowTell outperforms existing state-of-the-art methods for generating robot policies from language and visual demonstrations. Even for simple tasks, ShowTell outperforms the baseline methods, as the modular, neuro-symbolic programs are

	ShowTell-NoLang		ShowTell-NoVis		GPT4-V-Robot [128]		ShowTell (ours)	
Demo Type	**GCR**	**SR**	**GCR**	**SR**	**GCR**	**SR**	**GCR**	**SR**
Simple	0.89	0.85	0.86	0.81	0.88	0.85	0.96	0.94
Iterative	0.31	0.12	0.61	0.59	0.43	0.22	0.94	0.85
Conditional	0.41	0.39	0.64	0.64	0.41	0.41	0.93	0.93
Segmented	0.12	0.06	0.48	0.42	0.20	0.18	0.93	0.91

Table 5.1: Real-world robot manipulation task performance across different task families.

more easily able to reason about the demonstration and to resolve ambiguities than the monolothic VLM used to summarize demonstrations in the baseline method. This advantage is even more pronounced for the other task families as the purely sequential structure of the baseline method is not well suited to tasks that require high level logic like conditions, iteration, and segmentation. The qualitative examples in Figure 5.4 illustrate the ability of SHOWTELL to synthesize programs from such demonstrations. Comparing the ablation experiments, we see that the visual and spoken components of the demonstration are both important for generating effective policies, as the ablation experiments ShowTell-NoVis and ShowTell-NoLang perform poorly compared to the full SHOWTELL approach, even for simple tasks. This is because the visual and spoken components of the demonstration are often complementary, and help to resolve ambiguities present in the individual components of the demonstration. To better understand the failure cases, we analyze the failures in Section **??** of the appendix and illustrate the distribution of failures in Figure 4.6. We finally evaluate the performance of SHOWTELL across rollouts in the canonical scene (the scene used for demonstration), with unseen objects, and in unseen environments. The results summarized in Table 5.2 indicate that SHOWTELL is able to generalize to unseen environments and within category objects, while requiring only a single demonstration with no additional training or fine-tuning.

Rollout	**SR**
Canonical	0.92
Unseen environment	0.84
Unseen objects	0.88

Table 5.2: Performance across rollouts: in the canonical scene, with unseen objects, and in unseen environment.

5.5 Limitations

Our approach has several limitations. Most crucially, the approach is constrained by the limitations of the pre-trained models it uses, although we note that the modularity of the framework allows

for easy integration of new models as they become available. As parallel fields progress and new models are developed, we expect that the performance of ShowTell will improve and the variety of applicable skills will expand. Additionally, fine-tuning of pre-trained models to mitigate this limitation and reduce the need for explicit pre-processing is an exciting direction for future work. Our approach is also limited by the quality of the demonstrations provided, assumes demonstrated grasps can be mapped to a robot gripper, and may struggle with demonstrations that are incomplete or difficult to perceive due to occlusion or poor quality. Future work could leverage the interpretability of the approach to mitigate this limitation through interactive program repair. The inclusion of both visual and language inputs may introduce ambiguities that could complicate the approach. Future works could incorporate interactive dialog with the user to disambiguate inputs that are ambiguous or unclear. Finally, the approach uses closed loop execution to reach each goal waypoint and may struggle with dynamic disturbances or changes in the environment during execution.

5.6 Discussion

In this work, we present ShowTell, a neuro-symbolic robot PbD framework for synthesizing modular, generalizable, and interpretable robot manipulation programs from visual demonstrations and natural language. Our approach can teach robot manipulation skills from a single demonstration, without requiring any additional training or fine-tuning, by utilizing a combination of pre-trained foundation models and hand-engineered components to reason about observed demonstrations and to transfer learned skills to new scenes. We evaluate ShowTell on a set of 16 real world manipulation tasks across 5 visually distinct environments and show that this approach is effective for teaching a wide range of manipulation tasks. We show that ShowTell is able to reason about demonstrations that require high level logic like conditions, iteration, and segmentation, and that

it outperforms existing state-of-the-art methods for generating robot policies from language and visual demonstrations.

Chapter 6

Conclusion

Robots are increasingly expected to operate beyond the confines of factory floors, entering homes, hospitals, and workplaces where they must adapt to diverse tasks and user needs. Yet despite remarkable advances in machine learning, enabling robots to learn new manipulation skills efficiently remains a fundamental challenge. The core tension lies in the nature of human supervision itself: while human guidance provides the most reliable source of task-relevant information, traditional approaches to collecting and utilizing this supervision do not scale.

This thesis demonstrates that robots can learn manipulation skills more efficiently by transforming how we leverage human input. Rather than treating supervision as raw data to be collected in large quantities, we show how intelligent architectures can extract maximum insight from minimal human guidance. Through three complementary contributions, we address fundamental limitations in current approaches to human-robot learning.

Interactive Visual Failure Prediction (Chapter 3) shows that human supervision can be distilled into scalable interactive reward functions that operate in cluttered, unstructured scenes. By actively gathering additional sensory information to address partial observability, our approach achieved 73% success rates while requiring 4x fewer human interventions than traditional methods.

This work demonstrates that sparse human corrections can guide extensive autonomous practice when properly transformed into learning signals.

Diffusion-PbD (Chapter 4) leverages pretrained diffusion model features to enable one-shot learning from visual demonstrations. Without any robot-specific training data, our approach successfully transferred manipulation skills across 14 diverse tasks with 81% average success rates, demonstrating robust generalization to new viewpoints, objects, and environments. This work shows how foundation model priors can capture the compositional nature of human skills, enabling rapid learning from single examples.

ShowTell (Chapter 5) combines visual demonstrations with spoken narration through a neuro-symbolic framework, enabling robots to learn complex behaviors involving conditions, iteration, and logical reasoning from a single multimodal example. By synthesizing structured programs rather than learning reactive policies, our approach captures the hierarchical nature of manipulation tasks and enables systematic generalization.

Together, these contributions establish a new paradigm for human-robot learning. Instead of requiring extensive datasets or constant supervision, robots can learn effectively from sparse but well-leveraged human input. Our approaches transform brief demonstrations and occasional interventions into generalizable manipulation capabilities, moving beyond the traditional bottleneck of data collection toward more intelligent utilization of human guidance.

The empirical results across real-world experiments provide compelling evidence that this paradigm shift is not merely theoretical but practically viable. By combining reward distillation with foundation model priors, we demonstrate that robots can begin to learn more like humans do, from sparse but meaningful examples, with the ability to generalize and adapt to new situations.

6.1 Future Work

The techniques developed in this thesis open several promising research directions that could further advance scalable robot learning.

Adaptive Interactive Perception: While Interactive Visual Failure Prediction demonstrates the value of active information gathering, our interaction policies were relatively simple and task-specific. Future work should explore learning interaction strategies that adapt to new domains automatically. Meta-learning approaches could discover optimal probing behaviors for different manipulation contexts, while reinforcement learning methods could determine when and how to gather additional information most effectively.

Advanced Foundation Model Integration: Our work with diffusion features and vision-language models represents early exploration of foundation models in robotics. As these models continue to evolve, there are opportunities to develop more sophisticated multimodal understanding that integrates proprioceptive feedback, audio cues, and haptic information alongside visual and language inputs. Future systems could create even richer representations of demonstrated skills and enable more nuanced transfer learning.

Long-Horizon Reasoning and Planning: Our current approaches work well for manipulation sequences of moderate length, but extending to longer horizons with complex dependencies remains challenging. Integrating our neuro-symbolic programming approach with more sophisticated planning algorithms could enable robots to learn and execute complex, multi-step procedures that require temporal reasoning and conditional execution.

Safety and Robustness Considerations: As robots learn from human supervision and operate in unstructured environments, ensuring robust and safe behavior becomes critical. Future work should explore how to integrate safety constraints into reward learning frameworks and develop methods to detect and recover from distribution shift during deployment.

Dynamic Human-Robot Collaboration: This thesis focuses on robot learning from human supervision, but there are opportunities to explore more interactive paradigms where humans and robots collaborate fluidly. This could involve shared autonomy systems that seamlessly blend human input with learned behaviors, or interactive debugging frameworks that allow humans to understand and correct robot policies in real-time.

6.2 Broader Impact

The vision underlying this work is not to eliminate human input from robot learning, but to make that input so effective that thoughtful human guidance can produce capable, generalizable robot behaviors. As robots transition from structured industrial environments to open-ended human spaces, the need for intuitive and scalable learning will only intensify.

The approaches developed in this thesis point toward a future where programming robots becomes as natural as teaching another person—through demonstration, guidance, and feedback—while maintaining the precision and reliability that robotic systems require. By treating human supervision as a precious resource to be leveraged intelligently rather than simply collected exhaustively, we can build robots that learn more efficiently and generalize more effectively.

With continued advancement in both learning algorithms and foundation models, this vision of intuitive, scalable robot learning is within reach. The foundation laid in this work provides a stepping stone toward robots that can truly adapt to the complexity and variability of human environments while learning from the rich but sparse supervision that humans naturally provide.

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

[4] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[5] Baris Akgun, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 391–398, 2012.

[6] Saleh A Al-Abood, Keith Davids, and Simon J Bennett. Specificity of task constraints and effects of visual demonstrations and verbal instructions in directing learners' search during skill acquisition. *Journal of motor behavior*, 33(3):295–305, 2001.

[7] Sonya Alexandrova, Maya Cakmak, Kaijen Hsiao, and Leila Takayama. Robot programming by demonstration with interactive action visualizations. In *Robotics: science and systems*, pages 1–9, 2014.

[8] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement–residual rl for precise assembly. *arXiv preprint arXiv:2407.16677*, 2024.

[9] 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.

[10] D. Arumugam, S. Krening, M. Dehghani, and M. L. Littman. Deep reinforcement learning from policy-dependent human feedback. In *International Conference on Machine Learning (ICML)*, pages 342–351, 2019.

[11] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.

[12] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

[13] Noah Becker, Erik Gattung, Kay Hansel, Tim Schneider, Yaonan Zhu, Yasuhisa Hasegawa, and Jan Peters. Integrating visuo-tactile sensing with haptic feedback for teleoperated robot manipulation. *arXiv preprint arXiv:2404.19585*, 2024.

[14] Harold Bekkering, Andreas WohlschlaÈger, and Merideth Gattis. Imitation of gestures in children is goal-directed. *The Quarterly Journal of Experimental Psychology Section A*, 53(1):153–164, 2000.

[15] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *arXiv preprint arXiv:2309.01918*, 2023.

[16] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Robot programming by demonstration. *Springer Handbook of Robotics*, pages 1371–1394, 2008.

[17] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Survey: Robot programming by demonstration. Technical report, Springrer, 2008.

[18] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[19] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[20] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Ryan Julian, Nikhil Joshi, Dmitry Kalashnikov, Eric Jang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Igor Mordatch, Ofir Nachum, Brian Ichter, Fei Xia, Ted Xiao, Peng Xu, Ted Xiao, and Atil Iscen. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[21] Serkan Cabi, Relja Arandjelovic, Jost Tobias Springenberg, Marcin Moczulski, Andrew Zisserman, and Martin A. Riedmiller. Scaling data-driven robotics with reward sketching and batch reinforcement learning. In *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

[22] Sylvain Calinon and Aude Billard. Active teaching in robot programming by demonstration. In *RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 702–707. IEEE, 2007.

[23] Stephen Casper, John Thickstun, Jacob Steinhardt, and Ludwig Schmidt. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[24] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[25] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022.

[26] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[27] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4299–4307, 2017.

[28] Gergely Csibra and Rubeena Shamsudheen. Nonverbal generics: Human infants interpret objects as symbols of object kinds. *Annual review of psychology*, 66:689–710, 2015.

[29] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[30] Danny Driess, Fei Xia, Cosimo Ardi, Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Pierre Sermanet, Sergey Levine, Jiajun Wu, Andy Zeng, and Karol Hausman. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.

[31] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. Ar2-d2: Training a robot without a robot. *arXiv preprint arXiv:2306.13818*, 2023.

[32] Sarah Elliott, Russell Toris, and Maya Cakmak. Efficient programming of manipulation tasks by demonstration and adaptation. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1146–1153. IEEE, 2017.

[33] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[34] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

[35] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

[36] Maxwell Forbes, Michael Chung, Maya Cakmak, and Rajesh Rao. Robot programming by demonstration with crowdsourced action fixes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, pages 67–76, 2014.

[37] Maxwell Forbes, Rajesh PN Rao, Luke Zettlemoyer, and Maya Cakmak. Robot programming by demonstration with situated spatial language understanding. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2014–2020. IEEE, 2015.

[38] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[39] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[40] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023.

[41] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *Advances in Neural Information Processing Systems*, 36, 2024.

[42] Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics*, 24(6):1463–1467, 2008.

[43] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] Ayano Hiranaka, Minjune Hwang, Sharon Lee, Chen Wang, Li Fei-Fei, Jiajun Wu, and Ruohan Zhang. Primitive skill-based robot learning from human evaluative feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[45] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[46] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573, 2016.

[47] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[48] Nicola J Hodges, A Mark Williams, Spencer J Hayes, and Gavin Breslin. What is modelled during observational learning? *Journal of sports sciences*, 25(5):531–545, 2007.

[49] Justin Huang, Dieter Fox, and Maya Cakmak. Synthesizing robot manipulation programs from a single observed human demonstration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4585–4592. IEEE, 2019.

[50] Kun Huang, Edward S Hu, and Dinesh Jayaraman. Training robots to evaluate robots: Example-based interactive reward functions for policy learning. In *Conference on Robot Learning*, pages 11–21. PMLR, 2023.

[51] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[52] Wenlong Huang, Fei Xia, Corey Lynch, Pieter Abbeel, Andy Zeng, and Karol Hausman. Openvla: Towards open-world vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2311.04136*, 2023.

[53] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[54] William Hunt, Sarvapali D. Ramchurn, and Mohammad D. Soorati. A survey of language-based communication in robotics. *arXiv preprint arXiv:2406.04086*, 2024.

[55] Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024.

[56] Eric Jang, Alex Irpan, Coline Devin, Yevgen Chebotar, Julian Ibarz, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[57] Xingyu Jiang, Ajay Mandlekar, Wenlong Huang, Fei Xia, Pieter Abbeel, Andy Zeng, and Karol Hausman. Octo: An open-world, real-world robotic manipulation benchmark. *arXiv preprint arXiv:2312.08558*, 2023.

[58] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[59] Ivan Kapelyukh, Vitalis Vosylius, and Edward Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, 2023.

[60] Siddharth Karamcheti, Raj Palleti, Yuchen Cui, Percy Liang, and Dorsa Sadigh. Shared autonomy for robotic manipulation with language corrections. In *ACL Workshop on Learning with Natural Language Supervision*, 2022.

[61] Charles C Kemp, Aaron Edsinger, Henry M Clever, and Blaine Matulevich. The design of stretch: A compact, lightweight mobile manipulator for indoor human environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3150–3157. IEEE, 2022.

[62] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[63] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[64] 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.

[65] W. B. Knox and P. Stone. Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE International Conference on Development and Learning*, pages 292–297. IEEE, 2008.

[66] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16, 2009.

[67] Michael Laskey, Jonathan Lee, Roy Fox, Anca D. Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2017.

[68] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[69] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[70] Jacky Liang, Fei Xia, Wenhao Yu, Andy Zeng, Montserrat Gonzalez Arenas, Maria Attarian, Maria Bauza, Matthew Bennice, Alex Bewley, Adil Dostmohamed, Chuyuan Kelly Fu, Nimrod Gileadi, Marissa Giustina, Keerthana Gopalakrishnan, Leonard Hasenclever, Jan Humplik, Jasmine Hsu, Nikhil Joshi, Ben Jyenis, Chase Kew, Sean Kirmani, Tsang-Wei Edward Lee, Kuang-Huei Lee, Assaf Hurwitz Michaely, Joss Moore, Ken Oslund, Dushyant Rao, Allen Ren, Baruch Tabanpour, Quan Vuong, Ayzaan Wahid, Ted Xiao, Ying Xu, Vincent Zhuang, Peng Xu, Erik Frey, Ken Caluwaerts, Tingnan Zhang, Brian Ichter, Jonathan Tompson, Leila Takayama, Vincent Vanhoucke, Izhak Shafran, Maja Mataric, Dorsa Sadigh, Nicolas Heess, Kanishka Rao, Nik Stewart, Jie Tan, and Carolina Parada. Learning to learn faster from human feedback with language model predictive control. 2024.

[71] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[72] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, page 02783649241273901, 2022.

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

[74] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[75] Tomas Lozano-Perez. Robot programming. *Proceedings of the IEEE*, 71(7):821–841, 1983.

[76] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, et al. Mediapipe: A framework for perceiving and processing reality. In *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, volume 2019, 2019.

[77] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, 2023.

[78] Jianlan Luo, Perry Dong, Yuexiang Zhai, Yi Ma, and Sergey Levine. Rlif: Interactive imitation learning as reinforcement learning. *arXiv preprint arXiv:2311.12996*, 2023.

[79] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024.

[80] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

[81] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[82] James MacGlashan, Mark K. Ho, Robert Loftin, Bei Peng, Girish K. Srinivasan, David L. Roberts, Matthew E. Taylor, and Michael L. Littman. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning (ICML)*, pages 2285–2294, 2017.

[83] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv e-prints*, pages arXiv–2311, 2023.

[84] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. 2017.

[85] Guido Maiello, Marcel Schepko, Lina K Klein, Vivian C Paulun, and Roland W Fleming. Humans can visually judge grasp quality and refine their judgments through visual and haptic feedback. *Frontiers in Neuroscience*, 14:591898, 2021.

[86] Ajay Mandlekar, Yuke Zhu, Roberto Martín-Martín, Li Fei-Fei, and Silvio Savarese. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of Conference on Robot Learning (CoRL)*, 2021.

[87] Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pages 2905–2925. PMLR, 2023.

[88] Michael Murray, Abhishek Gupta, and Maya Cakmak. Learning to grasp in clutter with interactive visual failure prediction. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 18172–18178. IEEE, 2024.

[89] Michael Murray, Entong Su, and Maya Cakmak. Diffusion-pbd: Generalizable robot programming by demonstration with diffusion features. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5168–5175. IEEE, 2024.

[90] Suraj Nair, Eric Mitchell, Kevin Chen, Silvio Savarese, Chelsea Finn, et al. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315. PMLR, 2022.

[91] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Robotics: Science and Systems (RSS)*, 2022.

[92] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 663–670, 2000.

[93] Eley Ng, Ziang Liu, and Monroe Kennedy. Diffusion co-policy for synergistic human-robot collaborative tasks. *IEEE Robotics and Automation Letters*, 2023.

[94] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. `https://octo-models.github.io`, 2023.

[95] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal,

Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[96] Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2019.

[97] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[98] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[99] Richard Ramsey, David M Kaplan, and Emily S Cross. Watch and learn: the cognitive neuroscience of learning from others' actions. *Trends in Neurosciences*, 44(6):478–491, 2021.

[100] Vinitha Ranganeni and Maya Cakmak. Accessible tele-operation interfaces for assistive robots. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pages 91–93, 2024.

[101] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3(1):297–330, 2020.

[102] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[103] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[104] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[105] Stefano Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

[106] Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems (RSS)*, 2017.

[107] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[108] 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.

[109] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*, 2024.

[110] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[111] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[112] Weiyong Si, Ning Wang, and Chenguang Yang. A review on manipulation skill acquisition through teleoperation-based learning from demonstration. *Cognitive Computation and Systems*, 3(1):1–16, 2021.

[113] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[114] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

[115] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[116] Sumedh A Sontakke, Jesse Zhang, Sébastien MR Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: one demonstration is enough to learn robot policies. *arXiv preprint arXiv:2310.07899*, 2023.

[117] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*, 2023.

[118] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[119] Theodore R Sumers, Mark K Ho, Robert D Hawkins, and Thomas L Griffiths. Show or tell? exploring when (and why) teaching with language outperforms demonstration. *Cognition*, 232:105326, 2023.

[120] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023.

[121] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023.

[122] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[123] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[124] Andrea L Thomaz, Guy Hoffman, and Cynthia Breazeal. Reinforcement learning with human teachers: Understanding how people want to teach robots. In *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 352–357. IEEE, 2006.

[125] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[126] Marcel Torne, Max Balsells, Zihan Wang, Samedh Desai, Tao Chen, Pulkit Agrawal, and Abhishek Gupta. Breadcrumbs to the goal: Goal-conditioned exploration from human-in-the-loop feedback. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[127] An Dinh Vuong, Minh Nhat Vu, Hieu Le, Baoru Huang, Binh Huynh, Thieu Vo, Andreas Kugi, and Anh Nguyen. Grasp-anything: Large-scale grasp dataset from foundation models. *arXiv preprint arXiv:2309.09818*, 2023.

[128] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *arXiv preprint arXiv:2311.12015*, 2023.

[129] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *IEEE Robotics and Automation Letters*, 2024.

[130] Yuki Wang, Gonzalo Gonzalez-Pumariega, Yash Sharma, and Sanjiban Choudhury. Demo2code: From summarizing demonstrations to synthesizing code via extended chain-of-thought. *Advances in Neural Information Processing Systems*, 36, 2024.

[131] Zidan Wang, Rui Shen, and Bradly Stadie. Solving robotics problems in zero-shot with vision-language models. *arXiv preprint arXiv:2407.19094*, 2024.

[132] Stefan Wermter, Mark Elshaw, Cornelius Weber, Christo Panchev, and Harry Erwin. Towards integrating learning by demonstration and learning by instruction in a multimodal robot. In *Proceedings of the IROS-2003 Workshop on Robot Learning by Demonstration*, pages 72–79, 2003.

[133] Nils Wilde, Erdem Bıyık, Dorsa Sadigh, and Stephen L Smith. Learning reward functions from scale feedback. *arXiv preprint arXiv:2110.00284*, 2021.

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

[135] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial action maps for mobile manipulation. *arXiv preprint arXiv:2004.09141*, 2020.

[136] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[137] Danfei Xu, Annie Xie, Li Fei-Fei, Silvio Savarese, and Jiajun Wu. Instructrl: Generalizing decision-making with natural language instructions. *arXiv preprint arXiv:2307.09229*, 2023.

[138] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, pages 3536–3555. PMLR, 2023.

[139] Cheng-Fu Yang, Haoyang Xu, Te-Lin Wu, Xiaofeng Gao, Kai-Wei Chang, and Feng Gao. Planning as in-painting: A diffusion-based embodied task planning framework for environments under uncertainty. *arXiv preprint arXiv:2312.01097*, 2023.

[140] Shuo Yang, Wei Zhang, Ran Song, Jiyu Cheng, Hesheng Wang, and Yibin Li. Watch and act: Learning robotic manipulation from visual demonstration. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(7):4404–4416, 2023.

[141] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022.

[142] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

[143] G. N. Yannakakis and J. Hallam. Ranking vs. preference: A comparative study of self-reporting. In *Affective Computing and Intelligent Interaction*, pages 437–446. Springer, 2011.

[144] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[145] Takuma Yoneda, Luzhe Sun, Bradly Stadie, Ge Yang, and Matthew Walter. To the noise and back: Diffusion for shared autonomy. *arXiv preprint arXiv:2302.12244*, 2023.

[146] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[147] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

[148] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

[149] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[150] Guangyao Zhai, Xiaoni Cai, Dianye Huang, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. Sg-bot: Object rearrangement via coarse-to-fine robotic imagination on scene graphs. *arXiv preprint arXiv:2309.12188*, 2023.

[151] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. What does stable diffusion know about the 3d scene? In *arXiv:2310.06836*, 2023.

[152] 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.

[153] Yuxiang Zhou, Yusuf Aytar, and Konstantinos Bousmalis. Manipulator-independent representations for visual imitation. *arXiv preprint arXiv:2103.09016*, 2021.

[154] 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.

[155] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *6th Annual Conference on Robot Learning (CoRL)*, 2022.