Empowering Approaches for Creative, Intentional, and Informed Use of Robots

a thesis submitted by
Isaac Sheidlower

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

In

Computer Science

Tufts University

May 2025

Adviser: Prof. Elaine Schaertl Short

ABSTRACT

All people, regardless of technical expertise, should be able to utilize a home robot to its full potential so that they can use it for the purposes they want to improve their lives. To do this, people need to have control over their robot, the algorithms that dictate its behavior, and how they choose to interact with it. People who have this control are *empowered* to use the robot for general purposes. Furthermore, the user should be able to leverage all functions of the robot to accomplish tasks in the they want: by leveraging robot autonomy through Reinforcement Learning (RL) for example, or by actively collaborating in the task completion process. In this dissertation, we introduce four "empowerment strategies" for empowering non-experts users of robots: to facilitate innovative collaborations with a robot; to have personalizable robot autonomy; to have intuitive teaching tools; and to provide useful performance information.

An important aspect of user-empowerment is to use a robot in novel ways. We present a strategy for facilitating emergent uses of pre-trained robot policies for novel tasks by ensuring a robot's behavior is predictable as a person partially controls its actions. This predictability and interpretability leads to better task outcomes and user-experience outcomes. We then present an algorithm which allows a user to customize the style of a robot's behavior in real time as it executes a given task. This allowed the robot to complete the person's requested task and do so in the way want. However, a robot does not always know how to do a task. To partially address this, we introduce an algorithm that allows people to effectively teach a robot complex tasks with good/bad binary feedback. Finally, we empower users of general purpose robots more broadly by ensuring they have access to information that is useful for understanding the robot's capabilities so they can use their robot confidently for any task. The primary contribution of this dissertation is that we enable non-expert users of robots to leverage autonomous robot behavior to accomplish novel and creative tasks, while ensuring a positive user experience and robot transparency.

ACKNOWLEDGMENTS

I would like to thank my advisor Elaine Schaertl Short. Elaine is an awesome person and advisor. I feel very lucky to have been given the chance to work with her. Elaine guided me through this journey and predicted most of the steps along the way. I am very grateful.

Thank you to my committee Jivko Sinapov, Matthias Scheutz, Chris Rogers, and Maya Cakmak who guided me through the process of graduating and how to collect my various works into a single story. Especially being part of a new lab, my committee helped me become a more complete researcher.

I also would like to thank the members of the AABL lab. Kat Allen who was a constant reminder to keep pushing forward and take breaks when appropriate. Mavis Murdock who was always happy to talk through the challenges of getting a PhD and lifted my spirits. Hayley Owens who kept me grounded by reminding me there is more to life than what goes on in the lab and academia. Special thanks to James Stayley, Hang Yu, and Jindan Huang. All three of us are the first PhD students of AABL lab and consequently have been through quite a lot together. Hang was there to help get my very first research project off the ground and has since become a dear friend and collaborator. James taught me many things both in and outside of the lab that I will never forget (he is very wise). And an extra special thanks to Jindan without whom this dissertation would not have been possible.

Thank you also to Reuben Aronson who was a mentor to throughout much of my time at Tufts and who challenged me and my ideas (for the better of course). Thank you for all your help and last minute zoom meetings with your cat.

Thank you to my friends and other collaborators. Allison Moore, a great person that I feel lucky to know. Emma Bethel, whose skills are undeniable. Yumiko Mitsui for easing the challenge of starting a human robot interaction PhD during a pandemic. Douglas Lilly, who made up for my lack of engineering skills more than once. Andre Cleaver for making my smile every time we met. Thank you to Bingyu Wu, Qicong Chen, Teo Patrosio, Alina Shah, Yash Shukla, Shivam Goel, and Chris Thierauf.

Of course thank you to my family: David Sheidlower, Elaine Sheidlower, Nathaniel Sheidlower, Danielle Shalom. Special thanks to my dad David, who is the best dad I could have asked for.

Lastly, thank you to several restaurants whose food sustained me throughout the PhD: Davis Square Donuts and Bagels, True Grounds, Waikiki, Dakzen, Chicken and Co, Szechuan Dumpling,

Lao Hu Tong, Hanto Resturant, Masala, Gong Cha, Ciao Somerville, Sugar & Spice, Dragon Pizza, Semolina Kitchen and Bar, Mediterranean Grill, Tsuramen, and Yego Cafe.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

1.1 Overview

People should have control over their own robot, especially in their homes. Users of robots not only know their own needs and wants better than anyone else but, if given the correct tools, can use the robot to address these needs. This dissertation showcases what these tools may be and how to design them given a *capable robot* and a *capable user*. This dissertation does this by leveraging technical frameworks, primarily Reinforcement Learning (RL), to design user-centered algorithms and approaches that provide novice users with the impetus of control to perform novel tasks. Then, through evaluative techniques from the field of human-robot interaction (HRI), this dissertation evaluates these tools in context with people.

People who understand and are in control of their robot can use it for the purposes they want and in emergent, novel ways the robot may not have initially been designed for. Such users can be thought of as *empowered* as they can influence the robot's behavior for general purposes. Importantly, the robot may already be capable and "autonomous" in the sense that it knows how to perform the task the user may want, but an empowered user can not only specify tasks, but also how that task is to be completed and how collaborative the process is. In particular, this dissertation will focus on novel robot manipulation tasks that afford the user with precise control over how the robot affects their environment. These types of tasks are useful in and of itself, as a person may want to use the robot for a very specific purpose, but such tasks can also composed and combined to perform multi-step tasks. Thus, ensuring people can control a robot to perform precise manipulation tasks, is a necessary ingredient for empowering them to use the robot for the purposes they may want.

This dissertation focuses on how to empower users through different strategies, or ways a user can interact with or control their robot, each enabling a novel use of an already autonomous robot. To scope each of these empowerment strategies and how they are developed, this dissertation assumes a realistic setting where the robot has three functional ways with which to interact with the user.

First, it has a library of task policies, a "policy" being what determines the robot's behavior for a given task and are typically trained through RL or machine learning. Users can invoke these policies to have the robot subsequently attempt to perform that task autonomously. The robot will also have a means of teleoperation such that the user can directly control the robot. Finally, there will be software and hardware mechanisms which allow people to teach the robot new tasks. This dissertation builds upon these functions to design robot software that empower users through algorithms and by providing accessible information about the robot's capabilities.

Given these functions, this dissertation proposes four essential empowerment strategies that lead to tools robot should be equipped with and designed to promote. The four strategies are the following: the robot facilitates innovative uses of itself; its behavior is broadly customizable; it is able to be taught arbitrary tasks in a user-friendly way; and it is able to inform people about its capabilities. For each strategy, we design algorithms and studies that test their effect on human-robot interaction. In all cases, the strategy enhances the human side of HRI, improving user experience; the robot side, improving the robot's ability to successfully complete a task; or both. Although this list may not be exhaustive, designing around these strategies and their broader approach bridge gaps between human-centered HRI and robot-centered learning to empower people. After studying each strategy in depth, we also examine and discuss the future of general-purpose robot policies deployed in the home and how to design them in a similarly user-centered way. Overall, in this dissertation, **we enable non-expert users of robots to leverage autonomous robot behavior to accomplish novel and creative tasks**, while ensuring a positive user experience and robot transparency. For the remainder of the section we summarize the chapters that make up this dissertation.

1.1.1 Facilitating Innovation

Robots that can perform various assistive manipulation-based tasks need to be designed to do so. This means that their hardware and software capabilities must be robust to different environments and objects. Because of this, the robot has the potential to perform a very wide range of tasks. Some

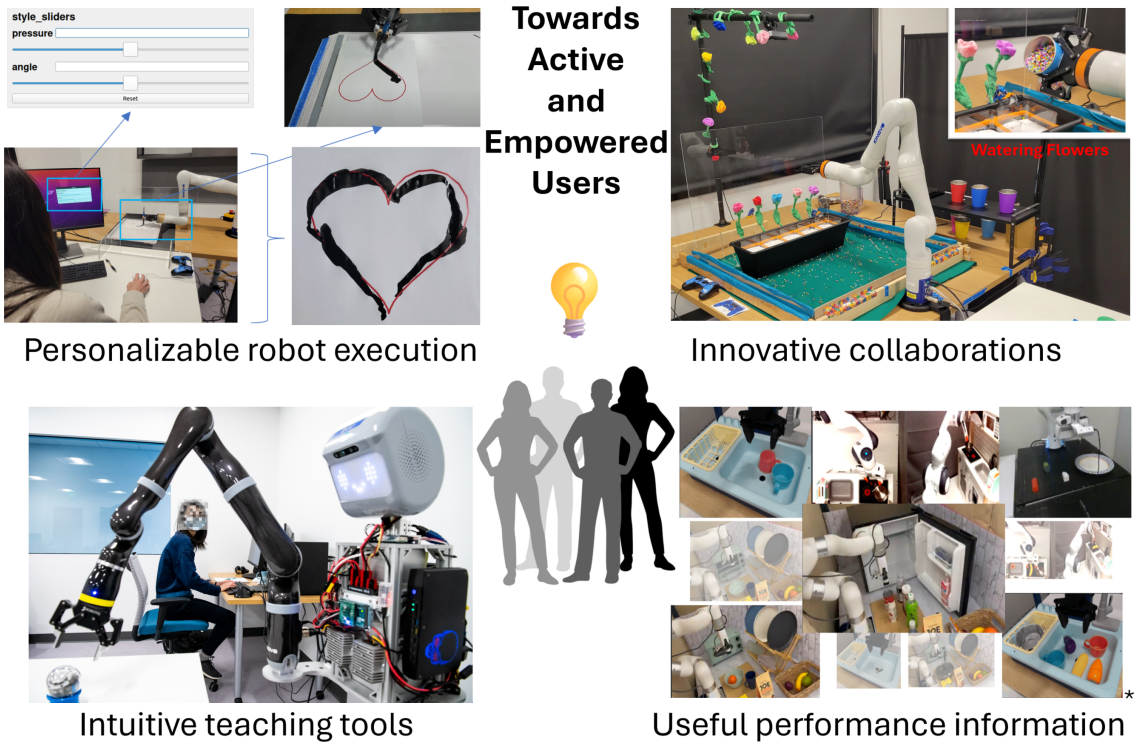


Figure 1: This dissertation focuses on granting non-expert users expert-like control over home robots. This entails designing human-centered algorithms, evaluating those approaches with real users, and studying user perceptions of state-of-the-art robot learning approaches.

of these tasks are relatively likely to requested by many different users, such as asking the robot to fold one's laundry or to fetch an item across the house. In these cases, robot designers will likely ensure that the robot knows how to perform these tasks in advanced prior to the robots deployment in a home. However, because of the general purpose nature of the robot, people will want to use it for tasks that the designers did not foresee. In these cases, a home robot should allow people to use it in innovative ways. Such allowance may both only lead to a more positive user-experience, being able to use the robot for tasks they want, and expand the robots capabilities in the future if it can learn from the new experience.

While people may want to teach an entirely new task, they may also want to use and apply its existing autonomous capabilities as well. Given a person is familiar with a certain task their robot can perform, they should be empowered to use that behavior in novel, unexpected, and fulfilling ways. For instance, a person that uses their robot to assist them in brushing their hair ought to,

with relatively few modifications or interventions, be able to use that robot to brush their cat or dog. Similarly, a user should be able to leverage a robot policy that can carry cups of water from place to place to water plants. In this dissertation, we investigate whether unmodified RL or learned policies can facilitate this kind of interaction. We find that this is not the case. Robots are trained using RL to complete a specific task are typically penalized for behaviors that deviate from that task. Thus, if a user attempts to use that policy for means other than the task it was trained for, the robot may act in unexpected or sporadic ways, or fight the user's attempts altogether.

We propose and study how to facilitate, rather than inhibit, innovative user interventions and innovative uses of pre-designed behaviors. To do this, we first formulate an exemplary type of intervention with a pre-trained policy, that of *partitioned control* (PC). In PC, which is a type of shared control more generally, users take teleoperation control over some parts of the robot's actions while the robot fully autonomously controls the other actions. Cruise control is perhaps the most prevalent example of PC: the user controls the steering while the vehicle, the robot, maintains the speed. PC is a way for users to use an existing robot policy in new ways and to reach new goals by combining their own teleoperation control with the robots internal control.

PC, however, is only as successful as the user's teleoperation capabilities but, more importantly, the robot's autonomous behavior. If the robot's behavior does not align with the user's expectations, then the system becomes both hard to predict and control, especially for an on-the-fly improvisational use. Inspired by the insight that during PC the robot should act as the user expects relative to behavior the user has seen before, we introduced the Imaginary Out-of-Distrubution (IODA) algorithm to address this problem.

During PC when the robot is potentially going to act unexpectedly, the IODA algorithm projects its current state back to a different state, such that it acts in a more predictable fashion. Specifically, it imagines that it is continuing to execute its task fully-autonomously without any user intervention. To test the effectiveness of PC for allowing users to leverage pre-existing policies to accomplish new tasks, we conducted a user study with non-experts. In this study, people used IODA to water flowers

by rotating the wrist of a robot arm as it carried a cup of water over a flower bed. The underlying policy was trained to not spill any liquid. Thus, the task of watering flowers, intentionally spilling, was a novel and unintended use of the original policy. When comparing both the task performance and qualitative user experience of IODA to an unaltered RL policy and an intuitive heuristic baseline (that of the robot being trained to stop moving when out-of-distribution), IODA outperformed both approaches. These results show the benefit of designing open-ended tools for user empowerment in terms of both user satisfaction and task success.

1.1.2 Customizing Behaviors

As discussed, the ability of a user to combine an existing policy with their own control can lead to high user satisfaction and task success. This type of collaboration requires the user to execute low-level teleoperation, albeit, while not necessarily difficult, people may want to interact with and shape robot behaviors at a higher level. *One policy does not fit all.* In other words, when a user requests a robot to perform some task, how that task is executed may not suit a user's preferences or needs. In contrast, an RL-based policy, or a pre-trained robot behavior model, may be able to complete a task, but it may only do so in one way; that way may not be aligned with a user's preferences. And, although the ability to specify a preference in the task specification, for instance, may help to alleviate this issue, it requires much trial-and-error to properly specify and does not allow for real-time modifications. In this dissertation, we study how to give users real-time control over a robot policy without the need for further teaching. This allows users to customize the robot's behavior through both "setting and forgetting" the policy's behavioral style, or adjusting it in real-time, allowing user expression and creativity in the task execution.

Although much prior work in preference learning, shared control, and RL has had an emphasis on task completion, a framework that emphasizes customization without sacrificing task progress was necessary to further empower users. We introduce this framework as online behavior modification. Online behavior modification is a framework for setting up or training a robot that, if followed,

empowers users with greater control over the robot to express their preferences without hindering or interrupting task performance. Online behavior modification also ensures that the control given to the user is interpretable, meaning the user can easily predict the outcome of their input to the robot. The importance of interpretability as a means of empowering users is a concept emphasized throughout this dissertation. This dissertation demonstrates the benefits of online behavior modification with a corresponding algorithm that is deployed in a user study.

Consider what an algorithm for online behavior modification should do: it should learn to complete the task, learn how to vary its behavior when completing that task, and ensure that those variations can be controlled directly. This dissertation introduces an initial algorithm for doing these things in an RL setting, Adjustable Control of Reinforcement learning Dynamics, or ACORD. ACORD learns a policy which both completes a task and can vary its behavior through pre-specified behavior features (such as how fast the robot moves, for instance). When we deployed ACORD in a user study in which people could adjust the style of how a robot paints certain shapes, we found that ACORD was generally well received by non-experts and conferred high levels of a sense of control while having a minimally adverse impact on the underlying task performance. Robotic systems, if explicitly designed in a human-centered way to give people more control, can do so without degrading the effectiveness of the robot or the interaction as a consequence of poor robot performance.

1.1.3 Novice-Friendly Teaching

Next, we consider the case where the robot *may not* know how to do a task. In these cases, it is necessary that people can teach the robot to do that new task, regardless of the difficulty of the task, and to be able to do so with relative ease. In general, for experts to teach robots new tasks, they can do so at any level abstraction and employ many different techniques. This has generally meant that experts have access to tools that allow them to teach robot tasks that are beyond the reach of novice users. This dissertation works to bridge the gap between what experts can teach to robots and what non-experts can teach robots through novice-friendly teaching tools. In this

context, "non-expert" is not meant to disparage people's ability to learn or adopt new tools; rather, we are focusing on teaching tools that anyone can use regardless of if they have specialized robot knowledge.

The primary paradigm which describes humans teaching robots is human-in-the-loop robot learning (HIRL). The focus of HIRL is generally the scenario of one person teaching one or more robots in real-time. HIRL consists of many different types of learning from instructions, including robot learning from demonstrations (LfD), learning from preference rankings (preference learning), and learning from evaluative numeric feedback. While each of these forms of teaching has had much success in teaching robots, in this dissertation we focus on evaluative feedback, specifically, binary feedback. Binary feedback allows a person to teach a robot by simply indicating whether the robot's behavior was good or bad. With enough such interactions, the robot can learn to complete a task according to the user's feedback. However, previous approaches to teaching with binary feedback alone were limited to relatively simple tasks. In fact, there was a large gap between the relatively complicated control tasks robots could learn autonomously through a reward function designed by an expert, and the tasks a non-expert could teach a robot using easy-to-provide binary feedback.

In this dissertation, we present work that sought to bridge the gap between what robots where able to learn with expert designed reward functions, and what novices could teach using the much more straightforward means of binary feedback. This dissertation introduces the Continuous Action-space Interactive Reinforcement learning algorithm, or CAIR. CAIR learns simultaneously from binary feedback and environmental reward. In particular, it relies on environmental reward to stabilize learning when binary feedback struggles to be precise enough to specify certain behaviors. Through simulation experiments, we show that CAIR not only outperformed the state-of-the-art in deep RL at the time but did so to the greatest extent in sparse reward tasks/environments. The benefit of sparse reward environments is that the reward function is relatively straightforward for even a non-expert user to specify: all that is needed is to provide the condition for when the task is successfully completed. We further evaluated CAIR with real teachers in an online study and

demonstrated CAIR outperformed baseline approaches. CAIR demonstrates the ability to easily teach a robot to perform autonomous tasks in accessible ways and expands people's access and control over their robot when they want the robot.

1.1.4 Informing Users about Robot Foundation Models

Ensuring people can use their robot effectively is both about single task interactions, but also about the robot's general capabilities as well. Robot Foundation Models (RFMs) are generalist policies that, ideally, can attempt and complete an arbitrary range of tasks. RFMs typically work by having a user specify a task, through natural language for example, then the robot will autonomously do the task. RFMs, however, generally have the risk of failing at the requested task, especially when that task is a novel request that was not present in the RFM's training data. Thus, for people to make use of this very powerful and promising technology, especially in their home where failure has potentially high risks, users must be *informed* about the RFM's performance and the risks associated with using it for various tasks. Although informing people of a robot's capabilities is important regardless of the underlying algorithm, RFMs have a lot of potential to, due to their general capabilities, be able to at least "do okay" on many tasks. As we previously discussed with IODA for example, being able to execute even parts of a task may have utility and can empower people to know when and how they can deploy their robot effectively.

What sorts of information are necessary to provide to people to both inform them about an RFM's performance and ensure that they can use the RFM confidently, without regrets? This dissertation begins to address this question. We first looked to the RFM literature to identify common metrics used in RFM evaluation to report its performance. From this literature review, task success rate (TSR) was by far the most common. TSR describes the ratio of the number of times the robot succeeded at a given task divided by its total number of attempts. The second most prevalent type of information is discussion of failure cases. Failure cases describe a failure(s) that occurred during the robot's deployment. This dissertation investigates how non-experts interpret TSR and failure

cases when deciding when and how to use a robot.

We conducted a user study where people were shown TSR and failure cases from real RFM research evaluations. In this study, people were presented with these different information types for both a task they requested and a similar task (to help gauge the overall robot performance) and asked a series of questions about their perception of the robot and the information. There were two key findings from this study: we verify that non-experts understand and are informed by TSR, a jargony metric otherwise aimed at experts; we demonstrate that failure cases, though very underrepresented in research literature, are also critical to and valued by users when they are being informed about an RFM's performance; highlighting the need of reporting failure cases for every task to become standardized. These results show that novice users are capable: being able to make use of otherwise technical information without much instruction. Furthermore, this dissertation claims that understanding user needs in advance is important for creating easy-to-use systems as people's needs may differ from what experts design or what information they make available.

1.1.5 User-empowerment with Robot Foundation Models, Design and Algorithm Considerations

This chapter considers more broadly the future of general purpose robots equipped with RFMs and how these systems can be designed to promote the four empowerment strategies discussed throughout the dissertation. We primarily focus on the algorithmic structure of RFMs and how they can be designed to be more controllable and predictable. We present preliminary results for a novel algorithm, Diffusion for Policy Parameters (DPP), which demonstrates how to design an RFM that can generate policies detached from the RFM itself. This has the benefit of the user being able to become familiar with and to adjust that policy without concern of affecting the RFM's performance or behavior on other tasks. We conclude with a discussion of the implications of approaches like DPP as it pertains to empowering users and, more broadly, how state-of-the-art foundation models can be designed and developed in a user-centered way.

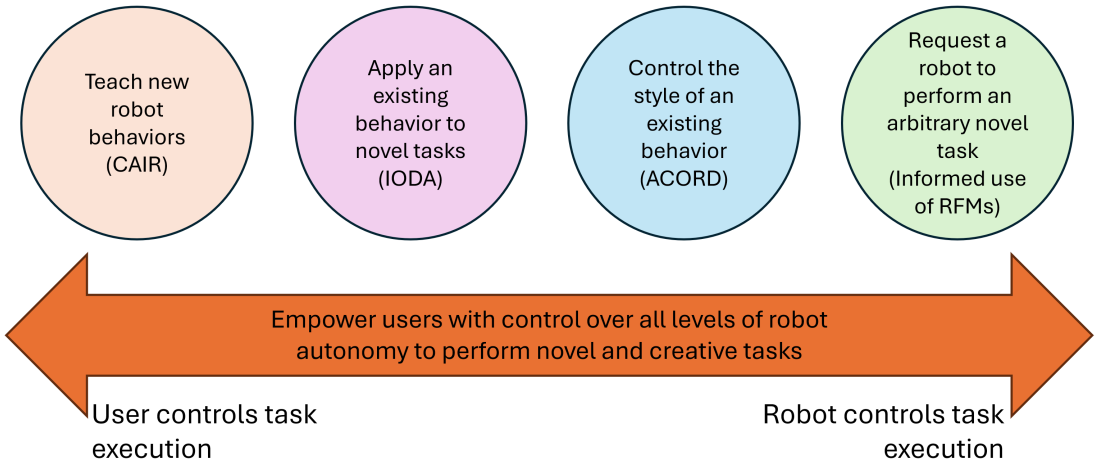


Figure 2: In this dissertation, we present "empowerment strategies" that empower users across different levels of user participation in task completion and robot control.

1.2 Contributions

In this dissertation, **we enable non-expert users of robots to leverage autonomous robot behavior to accomplish novel and creative tasks**, while ensuring a positive user experience and robot transparency. In a setting in which the user can control their robot on a range of manual control to specifying tasks for fully autonomous execution, we contribute to HRI, RL, and RFM research by demonstrating how systems can be designed in such a way to grant users high levels of control over their robot. These strategies facilitate user creativity and expression; enable innovative uses of pre-defined robot behaviors and policies to accomplish novel tasks; help bridge the gap between tasks experts can teach a robot with tasks novices can teach robots with simple feedback; and inform users so they can more confidently use their robot and gain expertise. This work functions as a practical guide for doing user-centered algorithmic HRI research that both enhances a robot's capabilities and enhances the user's control of that robot.

The following are the primary contributions of the dissertation.

- Two HRI theoretic contributions about how to give user's control over a pre-trained autonomous robot in creative or novel task settings.

- Three novice-friendly HRI algorithms that grant a user different types of control over a robot and afford both successful task outcomes and positive user experience.
- An evaluation of how people perceive, understand, and use robot foundation model performance information.

The following are secondary contributions:

- Recommendations for both robot foundation model and human-robot interaction researchers for human-centered design of robot foundation models and their evaluation.
- A novel algorithm for generating task-specific policies, which is evaluated and discussed in the context of user-centered robot foundation models.

Portions of this dissertation have been published in conference proceedings [236, 239, 237, 240, 238, 235]. Full citations are provided in the corresponding chapters at the end of the introduction to the chapter.

2 Related Works

This chapter reviews the literature that is broadly relevant to and is referenced throughout this dissertation. Each subsequent chapter has its own related works section that goes into more detail about the preliminaries related to each work. Each of the works discussed is related to user-empowerment in HRI. In this dissertation, user-empowerment is synonymous with giving people the means to control their robot, whether through direct, low-level control, or by eliciting a desired autonomous behavior.

2.1 User-empowerment and User-centered Design in Algorithmic Human-Robot Interaction

Much of this work is influenced and inspired by user-centered design, or the practice of involving users in the technology development process to ensure their needs are met once the technology is deployed [165, 2, 8]. User-centered design has also been successfully used for HRI applications. [202, 108, 286, 169] Although this dissertation does not always use user-centered design directly, it introduces strategies that seek to empower users with this sort of broad control such that the user can dictate the entire interaction. That is to say, while users may not be directly involved in the design process of all the approaches in this work, contrary to the fundamental principle of user-centered design, the goal of giving users more control is the same. As will be discussed throughout, user-centered design is also compatible with the user-empowerment based strategies from this dissertation.

In the field of HRI, much work is aimed at designing algorithms that enable a person and a robot to successfully collaborate on a given task. One approach to this is to design an algorithm for a specific type of task, then have the user interact with the robot where the algorithm is acting in the background and responding to what the user does. This type of approach is often found in Shared Autonomy (SA) applications [209], where the user's control of the robot is being augmented by a

black-box algorithm; or learning from demonstration (LfD) [208], where a user provides demonstrations and then an algorithm extrapolates from those demonstrations how to successfully complete a task without further user involvement. There are also extensive examples of this sort of approach to social, non-task-centric, human-robot interaction, but those are outside of the scope of this work.

Other approaches design algorithms that both allows people to interact with it largely as a black box, but also has components that a user can, in a similarly intuitive and or easy way, directly change. An example in the SA paradigm can be found in [93]. In this work, users have control over an underlying SA parameter that dictates how their control is traded-off with the robot's autonomy. Having more control was generally preferred to having less. Similarly, [224], incorporates user feedback into LfD that leads to the algorithm MIND MELD being better able to learn from their demonstrations afterwards. In this way, the user participates in shaping the underlying algorithm to ultimately lead to successful outcomes. Much of the work in this dissertation follows this approach to human-centered algorithm design.

2.2 Reinforcement Learning

For robots to assist and collaborate alongside people, they need some degree of autonomy. Autonomous robots also allow people to leave a robot to perform a task whilst they do other things. Reinforcement Learning (RL) [254, 125], is a paradigm that facilitates robot's learning to perform tasks autonomously and has been widely successful in robotics and AI alike. In RL, an robot acts within a Markov Decision Process (MDP). An MDP is modeled as a tuple (S, A, T, R, γ). Where this tuple is made up of robot observation states S, robot actions A, a transition function $T(S, A) \rightarrow S$, and reward function $R(S, A, S) \rightarrow S$. A policy π maps states of the environment to actions of the robot; and a discount factor γ. The objective of a RL agent is, in general, to maximize the sum of discounted future returns: $\sum_{t=0}^{n} \gamma^t R(s_t, a_t, s_{t+1})$. The policy $\pi*$ is the optimal policy which maximizes that objective. It is also helpful to define Q-functions and Value

functions. Q-functions are estimates of the "goodness" of state-action pairs under a given policy, expressed as $Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_t = s, a_t = a\right]$. Similarly, the value function is defined as $V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_t = s\right]$. Through careful design of reward functions, making sure to avoid poorly specified reward functions [34] or reward hacking [246], experts can elicit agents to autonomously perform tasks.

Much of the applicability of RL to complex robotic control tasks has come from Deep RL [17], or reinforcement learning done with deep learning and neural networks [144]. The first major success in Deep RL came via the Deep Q-Network (DQN) algorithm [175], an neural network based extension of the original Q-learning algorithm [280]. Since DQN, many more efficient Deep RL algorithms have emerged. Algorithms such as Proximal Policy Optimization Algorithms (PPO) [226], Twin Delayed Deep Deterministic policy gradients (TD3) [84], and Soft Actor-Critic (SAC) [99], all maximize the same objective but do so in slightly different ways and are significantly more efficient than the original DQN. All of these approaches, however, learn by exploring and interacting with the environment. PPO is an "online learning" algorithm in which the learning agent updates is estimation of state-action values as it takes actions and observes the next state. Whereas SAC and DQN for instance are "offline learning" algorithms that maintain a replay buffer of previous interactions which are sampled from to estimate the Q-function.

RL agents generally learn by exploring MDP environments for a sufficient amount of time until they learn to exploit it and maximize reward. Because of this, RL applications to robotics were, for a while, relatively limited as it was costly to deploy robots for a long period of time. The advent of Deep RL, however, meant that, if given sufficient resources, such as many robotic arms deployed at once [148], robots could begin to learn complicated and useful manipulation tasks. Other techniques were also developed to speed along the RL learning process for robots, such as Hindsight Experience Replay (HER) [12], which learned from a robots failed attempts at a task, or learning in a simulation as opposed to in the real-world [303, 103, 219, 191, 170]. Despite these advances, traditional RL requires difficult-to-specify reward functions, costly simulators, or a lot of time and resources. A

powerful way to overcome many of these issues, is to incorporate more information than is just available in the MDP through human feedback and teaching.

2.2.1 Reinforcement Learning as a Task Learning Paradigm

In this dissertation, we primarily use RL as a means of learning tasks and to facilitate various human-robot interactions. RL defines tasks as a MDP, requiring the definition of an agent's states, actions, and reward function. Because of the open-ended nature of reward function design and the ability of RL algorithms to learn to maximize the reward, RL has seen much success in applications ranging from robotics to playing games [174, 245, 288]. RL, however, is not the only paradigm for learning and defining tasks. And, while this dissertation focuses on RL, the empowerment strategies can be carried out with the application other paradigms as well. We highlight some other paradigms here as they have been shown to be affective when applied to task learning and HRI problems.

As highlighted in [91], the process of task learning, specifically interactive task learning, can take multiple forms. The tasks themselves can range from high-level, multi-step, tasks (such as assembling a piece of furniture), to lower-level manipulation tasks (such as pouring a cup of water). If a person can teach a robot these tasks in a natural manner, as they may teach another person, then that process likely involves language, demonstrations, and implicit feedback, all as forms of communication [223]. Along with RL, other paradigms have been used to try and facilitate this natural teaching interaction. Linear temporal logic (LTL) [104, 133, 232] is a popular approach to specify tasks to a robot due to its interpretability, preciseness, and expressivity. Furthermore, despite LTL-based task definitions occasionally being difficult even for experts to specify, the use of LLMs have shown promise for enabling non-expert users to specify LTL-based tasks through natural language [156, 195, 54]. Tasks can also be specified as planning problems for the robot where the task consists of a series of steps the robot must complete and do so in a specific order [4]. These planning problems are often expressed programatically, and similarly to LTL, have recently begun to be addressed with the use of LLMs to both find good plans for the robot and to make the

specification of those plans easier for the user [273, 149, 252].

There are also task-learning paradigms that often use a MDP representation other than RL worth highlighting and are relevant to the empowerment strategies presented in this dissertation. Transfer learning focuses on how to use knowledge and policies for given task and apply them to new, unseen tasks [305, 157, 118, 242]. Transfer learning techniques have also been applied to improve user-experience in human-robot interactions [45, 56]. Similar to transfer learning, curriculum learning addresses the problem of learning complex tasks by breaking down the complex task into a series of simpler tasks for the agent to learn [27, 274, 243]. Curriculum learning is a promising approach for empowering users to teach robots complex and long-horizon robot manipulation tasks.

2.3 Human-in-the-loop Robot Learning

Human-in-the-loop robot learning, or human-in-the-loop interactive robot learning (HIRL), refers to a broad category of techniques for allowing people to teach robots. This includes approaches for the robot learning from human demonstrations of the task (LfD) [14, 208], learning from people's preference ranking of different observed behaviors, or preference learning [50, 285], learning from evaluative numeric feedback [19], and learning from instructions [81, 186, 97]. Each of these approaches have the benefit of both quicker robot learning, and the teacher being able to express what their desired task may be without necessarily being an RL expert. While each of techniques are important and have their benefits, in this dissertation we will primarily focus on teaching via evaluative feedback as, while it is generally the most intuitive to provide, where the gap between what non-experts can teach with it and what experts can teach via other methods has been the largest.

2.3.1 Interactive Reinforcement Learning (IntRL)

Throughout this dissertation, we will refer to teaching via providing numeric feedback to a learning agent as Interactive Reinforcement Learning or IntRL. This type of teaching does not necessarily need to act an on RL algorithm, but it is very often the case that the learning robot is

acting an MDP and thus we assume this name. The primary benefit of IntRL is that is relatively intuitive for people: "when the robot is doing good, I should provide good feedback or a higher score; when the robot is doing bad, I provide bad feedback or a relatively low score. Just like training a dog." This type of feedback, however, is relatively abstract. Furthermore, it is known that people vary in terms of how they judge the goodness of behaviors, when they provide feedback, and how they interpret the robot's response to their feedback [111, 112].

Given the feedback the person is giving is relatively abstract, much of the success of an IntRL approach is how that algorithm interprets the persons feedback. The TAMER framework [138], one of the first IntRL algorithms treats a person's feedback as a reward signal. Other approaches treat both the human feedback and the environmental reward as equally valid sources of information for creating policies, such as in Policy Shaping (PS)[94], or as policy "advise" such as COACH [161]. The TAMER and COACH frameworks have also been extended to continuous state-space settings such that it could learn from images, for example [278, 13, 18]. The TAMER framework was also extended to work with continuous action-spaces by incorporating an actor-critic style approach, was still not applied to high dimensional robot tasks [267, 46, 173, 135] Work by Faulkner et al. has more closely considered the experience from teacher's perspective. For example in [75] and [131], they consider how to algorithmically cope with a teacher who may not always be paying attention or needs to multitask whilst teaching. Still other work has focused entirely on understanding and improving the experience from the teacher's perspective [111, 51] or understanding their feedback regardless of the underlying IntRL algorithm [295, 296].

2.3.2 Learning from Demonstration (LfD)

Learning from Demonstration, or LfD, refers to a type of teaching where a person will provide demonstrations of how to perform a task to a robot, and the robot will then use those demonstrations to both learn to replicate the user's demonstrations as well as learn to a policy that can be used for the same task but under different circumstances or in different environments. LfD is a particularly

effective approach for teaching robot's new tasks as the demonstrations themselves can provide information both about how the robot should move and about the robot's environment. Neural-networks, for example have allowed LfD approaches to become more robust to changes in the exact task set-up (e.g. if the demonstration involved a person getting a cup from the middle shelf, the robot may learn to generalize to be able to get the cup from the top or bottom shelf as well). In particular, algorithms such as neural-network based behavior cloning (BC) [262, 77, 221], which learn to match the output actions of a neural-network based policy to those actions provided the teacher, or Inverse RL methods [16, 83, 184], which learn a reward function that explains the teacher's demonstrations and then executes RL to learn a policy from that reward function. Along with the LfD algorithm, the mechanism for providing demonstrations is also important to ensure that the user can comfortably demonstrate the task to the robot.

There are many different ways people can provide demonstrations to a robot for LfD. Kinesthetic teaching, for instance, involves the user moving the robot by hand to perform the task, and often the user will specify key-frames, or points the robot should pass through when it autonomously attempts the task [6]. Another common approach is for people to will demonstrations to the robot via teleoperation, often through a joy-stick based controller [244, 248, 63]. Recent work has also attempted to learn directly from videos of people performing tasks, particularly because of the large amounts of data for that type of learning (ideally any video on the internet of someone performing a task, could be used as a demonstration) [25, 292, 234, 205]. Another reason to attempt to learn from videos, is that demonstrations are relatively "expensive" for the user to provide. In contrast to IntRL, for example, the teacher themselves needs to be able to perform the task, which may be difficult depending on the task and the interface the user has access to. To mitigate the problem that providing demonstrations may be difficult, researchers have developed interfaces that make providing demonstrations easier, such as a light-weight replica of a robot's gripper that can collect data [71, 257], developing algorithms that can efficiently learn from only a few demonstrations [48, 69], or creating large datasets of demonstrations [55, 164].

2.4 Learning User Preferences

While IntRL is generally a paradigm for teaching a robot a task that is either completed successfully or not, other work has focused on learning people's preferences about how that task should completed. Notably, work by Bobu et al. has presented algorithms and techniques that allow people to teach robots their preferences of "behavior features" [33, 32]. Behavior features are stylistic components of how a task is executed (for example, how close the robot carries liquid to an expensive laptop or how much force the robot applies when pushing a button). In these works, the robot generally knows in advance how to perform the task in terms of completion, but needs to be taught the person's preferences. Other work has also focused on interactive reward shaping [29, 30, 207, 180], defining safety constraints [281, 9, 159], or using shielding, or dictating what actions a robot can or cannot take [36], to align robot behavior with user preferences. Very few works have focused on real-time customization, where a user's preferences may change on the fly, which we partially address in this dissertation.

2.5 Shared Control

People will have access to a means of directly controlling robots through teleoperation [199, 129, 130, 53, 271]. However, direct teleoperation, especially without specially designed interfaces, is generally challenging and potentially cumbersome [145]. To alleviate some of these difficulties, shared control (SC) paradigms combine a person's control with that of the robots. Work in SC often blends user and robot control without inferring what the user's intended goal may be or significantly augmenting the users' input [166, 206, 268]. Shared autonomy (SA) is an extension of SC which grants a user partial direct control over a robot that can already complete the user's intended task autonomously. However, the person's intended goal is assumed not to be fully known in advance. Thus, a primary goal of SA algorithms is for a robot to infer a user's intended goal or skill based on some input [200, 93, 120, 228, 178]. Using RL in the SC loop [79, 209], or using latent spaces to interpret user control [121, 293], have also been deployed to improve SA performance and user

experience. More recently, there has also been work that examines how much control users have over the entire SA system, as opposed to just examining individual tasks [15].

2.6 Robot Foundation Models (RFMS)

2.6.1 What are RFMS?

Robot foundation models (RFMs), also known as generalist policies, large behavior models, or vision language action models (VLAs), are gaining popularity as a way to distill data from a wide range of tasks into a single general model. RFMs, such as RT-1 [38] and RT-2 [39], are typically an end-to-end policy which take as inputs a language instruction or goal image and a camera(s) view(s) as an observation, and output a sequence of actions typically in end effector space. These models require a large amount of data to train a transformer architecture-based policy so that they can learn to perform many different tasks in many different environments. Octo [89], for example, is an RFM trained with data across many different robot environments collected from many different institutions; specifically, with the Open X-Embodiment dataset which was also used to train variants of RT-1 and RT-2 [187]. Like LLM model scaling [302], more diverse data and larger models tend to improve an RFMs performance and generalization capabilities. For example, Open-VLA [134], a 7 billion parameter model trained on diverse data, generally achieves higher success rates on out-of-distribution/unseen tasks than Octo (a significantly smaller model) and RT-1/2-X (models trained with less data diversity). It should also be noted that the success of RFMs has coincided with the advancement of data-efficient and lower cost imitation learning methods [48, 301, 257, 231] and smaller-scale but effective multi-task learning techniques [241, 275, 210, 181, 102]. Ultimately, these models are still in their infancy but are rapidly evolving as the underlying techniques to train them become more efficient and affordable.

2.6.2 Evaluating RFMs

For non-experts to best make use of RFMs, it is important they understand their capabilities. Current research in RFMs typically present results through *task success rate* (TSR) across a variety of different task environments. These tasks typically are short-horizon pick-and-place tasks done using robot arms such as the Franka Emika Panda [80] or WidowX [284]. Although many of the RFMs are evaluated with real robots, recently the development of high-fidelity simulations has allowed simulation evaluations as well [154, 117, 298]. Whether on a real robot or in simulation, TSR is by far the most common performance metric for evaluating RFMs. TSR can be useful to the research community for allowing one to directly compare different algorithms and RFMs on the same tasks. However, TSR is not the only means of evaluating these models. For example, some works also discuss *failure cases* across a subset of different tasks. Despite how impressive these RFMs are, they are still very prone to failing; though less commonly reported, how they fail can be useful information for researchers [158, 68] and potentially novice users alike. Work by Wang et al. [276, 277], proposes benchmarks for evaluating RFMs in simulation by randomizing aspects of the environment such as camera angles and lighting conditions. Furthermore, they break down tasks into sub-tasks which they also present success rate for (e.g. a pick-and-place task requires the robot to move to the correct object and successfully grasp it before moving it to a different location). Estimating TSR and if/how a robot will fail is also an ongoing area of research [92, 65].

3 Facilitating Innovation

3.1 Introduction

In this chapter, we examine how to facilitate innovative uses of pre-trained policies. This is a key ingredient in empowering users as it allows them to easily use a robot for novel applications without requiring a significant effort teaching or correcting robot behavior. Given that people will have access to a wide variety of robot behaviors that they frequently deploy and interact with in their home, it is likely that people will want to use those behaviors for unexpected purposes. In this chapter, we assume that these behaviors are executed through RL policies. RL policies learn to perform tasks via reward functions that specify what the robot should and should not do when attempting this task. This chapter investigates how a user may collaborate with an RL policy through shared control to perform novel tasks that differ from the task defined by the reward function. We formulate the problem of facilitating shared control collaboration with an RL policy as one of aligning robot behavior with user expectations. To do this, we first formalize a type of shared control, partitioned control (PC), demonstrate limitations of naively applying PC to RL, and show that explicitly aligning robot behavior during PC leads to better task-based and user-experience outcomes.

To investigate how a RL policy would react to PC, we conducted simulation experiments. In these experiments, a simulated user would try to use PC to reach sub-goals in a Euclidean plane where the robot would control the y-axis actions and the user would control the x-axis actions. Rather than continuing to act optimally in terms of reaching the goal on the y-axis, as one would expect, the PC brought the RL trained policy *out of distribution* (OOD). Since the underlying policy was trained with a neural network, being OOD meant the policy acted almost randomly, meaning no matter what the user tried, they could not effectively collaborate with this robot in this task. This issue, present even in a very simple task, highlights the need for approaches that explicitly account for user intervention and creativity. Inspired by the insight that during PC the robot should act

as the user expects relative to behavior the person has seen before, we introduced the Imaginary Out-of-Distribution (IODA) algorithm to address this problem.

We deployed IODA in a user study in which people were tasked with using a PC to accomplish a novel task, in this case, watering flowers. The results demonstrate that, compared to naïve but intuitive baselines for PC, IODA led to both the best task performance and was the most preferred by users. Furthermore, because IODA more closely aligned with user expectations, and because of the smoothness of execution during PC that IODA afforded, there was significantly less time spent on both doing the task and learning how the underlying PC algorithm worked. We also find across conditions that aligning with user expectations and reducing people's surprise during PC both correlate with task performance. Though this secondary finding is only for this single task, we posit that this is generally true for PC. These results demonstrate the benefit of designing robot algorithms that empower people by allowing for open-ended uses of the robot.

The majority of the work in this chapter was published at IEEE RO-MAN 2022 as Sheidlower, Isaac, Emma Bethel, Douglas Lilly, Reuben M. Aronson, and Elaine Schaertl Short. "Imagining In-Distribution States: How Predictable Robot Behavior Can Enable User Control Over Learned Policies." In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, 1308–15, 2024. https://doi.org/10.1109/RO-MAN60168.2024.10731233 [236] *.

3.2 Related Works

Legible robot motion refers to robot actions that are straightforward for a human to anticipate and comprehend. A common way to generate legible motion in goal-based robotic tasks is to model the user as having an internal cost function that is minimized when the robot's motion saliently moves towards a given goal [67, 74]. An alternative approach is to learn from humans through demonstrations or feedback [42, 28]. Importantly, legibility and predictability are in the

*The second author helped with making visualizations for the figures. The third author helped make the sensor-equiped flower bed used in the study. The second to last author consulted on the experiment design. The last author supervised the work and consulted on the experimental design

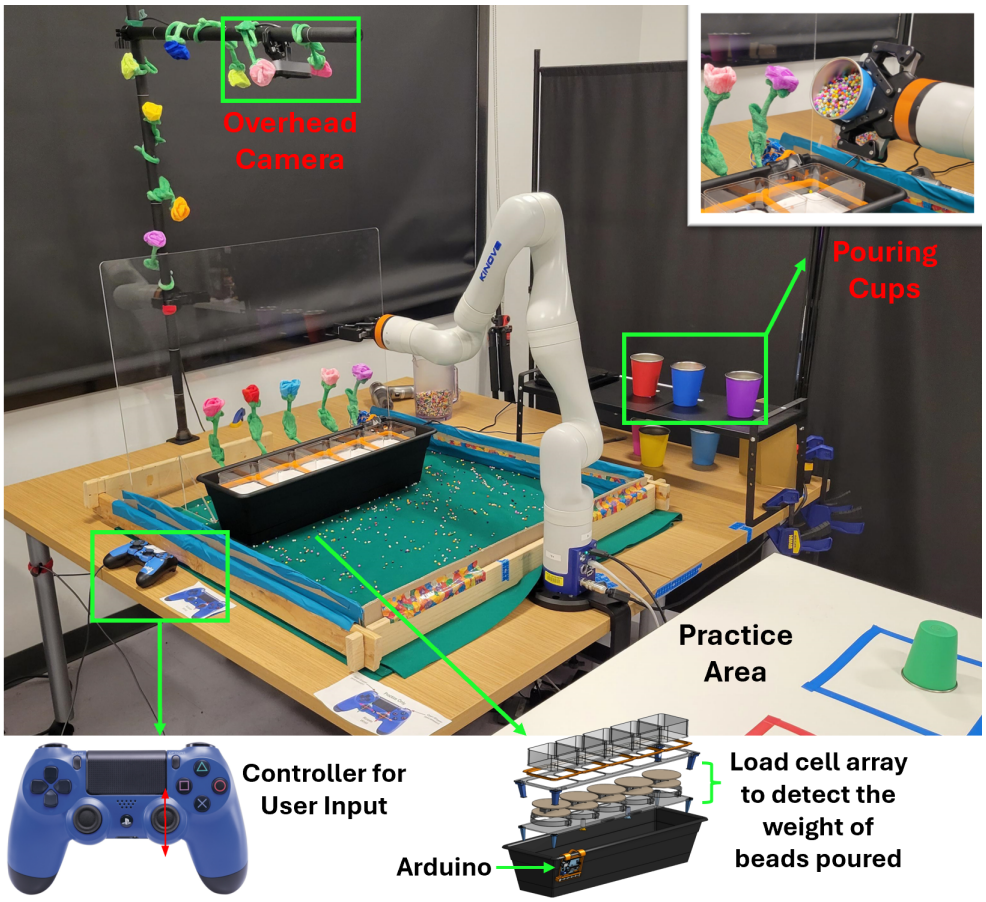


Figure 3: A depiction of the "flower watering" task setup used to study Partitioned Control and IODA with novice-users.

context of the robot *completing the task* and are often in real-time as opposed to pre-hoc or post-hoc explanations [61, 58, 217, 194]. Predictability is also an important part of our work. We operate under the assumption that a robot's behavior is predictable if the user has previously encountered similar behavior. This assumption is somewhat analogous to robot-centric concepts of out-of-distribution (OOD) states and behavior.

OOD detection is useful in many robotic and machine learning tasks [291, 253]. Identifying scenarios that are OOD relative to a robot's training data or past experiences can provide implications about the robot's environment or its performance, such as when an RL agent may behave sporadically or unexpectedly when in a state it has never been in before [143]. It has also been used to identify when a robot may require feedback from a person to help complete a novel task [62]. In

RL. OOD has recently been studied to infer when an agent is acting in a new MDP [101]. We use similar techniques to detect when the robot is in a state that it would not otherwise act in. This can happen when a user is partially controlling a robot to perform a new task.

3.3 Problem Setting and Partitioned Control

We describe a problem setting in which a user is accustomed to the autonomous execution of a task and wishes to partially control a robot during that execution to accomplish another task. The user creates a plan to partially control the robot based on how they expect the robot to behave to accomplish a novel task. Thus, it is critical that the robot behaves in a way that is predictable to the user, no matter the robot's current state. Given the robot behaves in a user-predictable way, as opposed to sporadically or in an unfamiliar way, a user can perform various novel tasks with little surprise and relative ease.

In this setting, the user has seen the robot complete its task many times. We refer to this as a history of task "rollouts." Based on this, we assume the following: when the robot is in a state the user has never seen before, they expect that the robot will act the same as it would do in the "closest" state to its current state. The "closest" is both problem and user-specific; however, the intuition is that the robot will behave similarly in similar circumstances and that in novel circumstances, a user will project onto what they have seen before. This assumption temporarily constrains the problem space. However, it is a reasonable assumption in many robotic tasks. Thus, the problem can be defined as: for any given state unseen to the user, the robot should find a state that the user has seen before and act as if it were in that state.

We will now define the original task the robot can complete autonomously, and how this task is used to build up a user's expectation of the robot's behavior. Let task *orig* be defined as an MDP with states $S \subset \mathbb{R}^n$, actions A, reward function $r : S \times A \to \mathbb{R}$, and transition function $T : S \times A \to S$. There is a robot that has learned an optimal policy for the task denoted π^*. Let D be defined as a history of rollouts, or trajectories made up of a sequence of states, under π^*_{orig} that

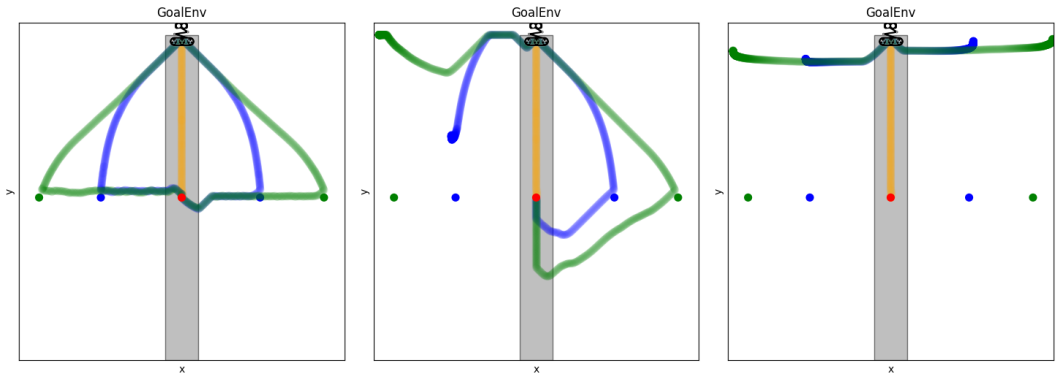
the user has seen. Then, let the user's expectation of the robot's behavior given D be $W_D : S \rightarrow S$. Here, W is a function that maps from the robot's current state to the user's anticipation of what the next state will be.

To this setting, we introduce Partitioned Control (PC), where the user teleoperates one or more parts of the robot's actions. We separate the action space into two separate sets $A_U, A_R \subset A$; A_U denotes the actions that the user can take and A_R denotes the actions that the robot can take. In PC, the user and robot action spaces are *disjoint*; that is, $A_U \cap A_R = \{\mathbf{0}\}$. In other words, the user and the robot control different parts/different axes of the action space (this is in contrast to many SC approaches that blend the user's actions and the robot's actions [209, 120]. For example, if the robot is acting in Cartesian space, the user may take control over x-axis actions, or take control over the rotation of a specific joint. We denote the user's expectation of how the robot will act with their partial control signal as $W_{D,U} : S \rightarrow S$, where U is the user control. This expectation function can be read intuitively as "given the trajectories the user has seen before, as well as their current control signal/action during PC, the user expects the entire robot system to transition from its current state into a particular next state." We assume that this expectation is not a function of the robot's current action, or rather, that the user's perception of the robot's actions are through its state transitions and not by observing the numeric value of each robot action. For brevity, hereafter we refer to this only as W.

To make the robot's behavior more predictable for the user, we want to adjust the behavior of the robot policy when it is outside the user's observation set D. The goal is to identify when the robot is in a novel state s where there exists a state $s' \in D$ that leads to more predictable behavior. Formally, identify when $\exists s' \in D$ s.t.:

$$d(W(s), T(s, u \circ \pi^*(s))) \geq d(W(s), T(s, u \circ \pi^*(s'))) \tag{1}$$

where d is a task-dependent distance metric between states, and $u \circ \pi^*(s)$ denotes the disjoint



(a) IODA

(b) Out-of-distribution, no imagined state

(c) Must return to workspace, no imagined state

Figure 4: In a 2D goal navigation task, a simulated user is trying to leverage an optimal policy to reach sub-goals by controlling the x-axis of the robot whilst the policy controls the y-axis. These sub-goals are outside the robot's original workspace (highlighted in gray). Each line represents the robot's trajectory to a certain sub-goal and are color-coated. For instance, each green line on the left hand side of the workspace represents the robot's trajectory trying to reach the left green-dot sub-goal then proceeding to the red-goal in the workspace. Our algorithm IODA allows the user to seamlessly reach the sub-goals.

combination of the autonomous action of the robot and the user's teleoperation. When such a s' is identified, the robot should act as if it were in s'. Specifically, we want to select a new proxy state s' such that the user's *predicted* state $W(s)$ is closer to the actual resultant state when simulating the policy in s', $T(s, u \circ \pi^*(s'))$ than to the resultant state of running the policy directly $T(s, u \circ \pi^*(s))$. Lastly, in this setting, the true W and the nature of the new task that the user wishes to accomplish are unknown. However, formalizing W as such can be useful for modeling and/or simulating, creating a learning objective, or creating metrics to measure the success of algorithms applied to this problem. See the limitations section of this chapter for a discussion of alternative ways of formulating the problem of aligning the robot's behavior with user expectations.

3.4 Imaginary Out-of-Distribution Actions (IODA)

In this section, we present Imaginary Out-of-Distribution Actions (IODA), to facilitate a user to accomplish new tasks given a policy and a means of teleoperating an axis of robot behavior (the problem setting described in the previous section). Our key insight is that when the robot

Algorithm 1: Imaginary Out-of-Distribution Actions (IODA)

Initialize: Rollout history D

Initialize: OOD-state detector

Initialize: State s

while not done **do**

  if s is OOD **then**

    $s \rightarrow \arg\min_{s' \in D} d(s, s')$

  end if

  $a \rightarrow \pi^*(s)$

  $u \rightarrow$ user's control signal

  $s \rightarrow T(s, u \circ a)$

end while

is acting in a region that greatly differs from what the user has seen before, the policy should act with imagined states that are as similar to the real state as possible while being "in-distribution" of what the user is familiar with and anticipates. Unless otherwise specified, we refer to "in/out of distribution" states with respect to D.

The complete IODA algorithm is presented in Algorithm 1. Here, we require that an OOD detector be trained on D. This is then used to detect "novel" states. While this technique is not new from a robot-centered perspective, it is also a human-modeling choice that draws an analogy between when a state is OOD and when a human may be projecting to a state they have seen in the past. Thus, it is also being used to determine when to search for a state that the robot policy should "imagine" it is in.

3.5 Simulation Example

In this section, we demonstrate IODA in a 2d navigation task. In the original task (see Figure 4), the robot learned to navigate to any specified goal point from within its workspace (highlighted in gray in Figure 4). A user then seeks to leverage the behavior they have observed to control the x-axis actions to first guide the robot to an intermediate sub-goal outside of the workspace (shown in Figure 4 as blue or green dots) and then to the primary goal: this represents a novel task not represented by the robot's policy.

We train two RL agents to optimally solve two slightly different versions of the navigation task using the off-policy RL algorithm SAC [100]. We use SAC as it has been shown to be relatively robust out of distribution [143]. In one version of the task, Figure 4, b, the RL algorithm was restricted to the gray workspace by being penalized for leaving it. In the other version, Figure 4, c, the agent is penalized if it is out of the workspace, and further penalized by moving in the y-axis whilst it is out of the workspace. This encourages the agent to return to its workspace as quickly as possible before continuing the task. In both cases, the agent, when out of its workspace, may behave in a way unpredictable to a user. As a user has only seen optimal rollouts, they may not be familiar with what happens when the robot "fails" or is OOD and will likely expect that the robot would continue toward the primary goal along the y-axis.

In these environments, we collected 1000 rollouts of the optimal policy and trained Deep SVDD OOD detectors [213] on the states of those rollouts. We choose d to be the $L1$ distance between two states. Finally, we substitute human user control for an optimal x-position controller given the current x-position and sub-goal location. As can be seen in Figure 4 IODA is the only condition in which the simulated user can reach all sub-goals and then easily proceed to the primary goal. In Figure 4, b, the agent acted relatively sporadically when brought out-of-distribution, and could only reach both goals half the time. In Figure 4, c, since the agent was trained not to move in the y-axis when outside of its workspace, the agent's behavior inhibited the simulated user from reaching the sub-goals. Furthermore, D did not contain any indication that the robot would stop.

3.6 Methodology

To study how users can leverage PC to accomplish new tasks as well as the efficacy of the IODA algorithm, we conducted an in-person user study, where people use PC with various underlying algorithms to accomplish a novel task. We hypothesize that users can leverage their expectations of robot behavior along with PC to accomplish this task. The IODA algorithm was designed to facilitate this. Thus we seek to validate users can use PC in this way and that when the robot's behavior more closely aligns with user expectations, the user can more readily complete the task.

Plant Watering Task To study PC and IODA, we choose to replicate the scenario discussed in Section I. In this task, there is an RL robot policy that transports cups of liquid from one place to another, for handover, table setting, etc. The user then posits that they can use this task to water their flowers if they can rotate the robot's wrist as the robot carries the liquid to pour it over the flower bed. The fully autonomous component is the robot traveling along one side of the flower bed to the other, while users are prompted to pour out liquid to water the flowers by rotating the robot's wrist. This task is intuitive and entails PC over a single action space dimension and is thus suited for a study where participants are still relatively novice at teleoperation.

The base policy for this task was trained using RL via SAC [100]. The reward function used penalizes the robot per time step while the cup is not at the goal or if the robot spills liquid (by overly rotating the wrist or moving too fast). There is a large positive reward for reaching the desired goal position. Based on the setup for this task, rollouts of the optimal policy would not include the robot spilling or largely rotating its wrist past a threshold. However, this is precisely what users will need to do to perform the pouring task. While users have seen rollouts of the optimal policy they have not seen the robot train nor know what happens when the robot enters the OOD state of pouring out liquid.

Experimental setup The setup consisted of a Kinova Gen3 7-dof arm located between two tables (Figure 3). A table was used for the participant to practice controlling the robot through teleoperation; the other was used to demonstrate the robot cup-carrying policy and for the watering

task. The pouring material used were small beads meant to replicate pouring a fluid. The flower bed was equipped with 5 different containers, each of which had scales below them to measure the amount of beads poured into each container. Each container represents an individual flower.

3.6.1 Conditions

For all conditions, the policy for the robot's autonomous behavior is constant. The robot will attempt to transport a cup full of beads from one end of the flower bed to the other. While the robot is doing this, the user will have roll axis control over the robot's wrist. Each condition is an approach one may take for PC.

Unaltered Base Policy In this condition, there is no alteration to the base liquid carrying policy during the user's PC. As the setting and PC are novel, this intuitive baseline is important to serve for both studying a user's experience during PC and how an underlying policy may perform in these scenarios. However, because this policy is unaltered, it may suffer from the problems associated with out-of-distribution states examined earlier. That is, the robot may act or move sporadically along its path if the user's control causes the robot to start spilling the beads (which they are intentionally trying to do). We expect that this will result in both lower task success and that the robot's behavior will not align with the user's expectations based on what they have seen prior. We will refer to this as the *RL* condition.

Base Policy with Enhanced Failure Recovery ("STOP") In this condition, the base policy also has an explicit failure recovery component. Although the unaltered policy may still try to recover from spilling liquid, there is no explicit instruction for what the robot should do while spilling. For example, the robot should stop moving along its path to minimize the spread of the spill. In this condition, however, there is an additional safety constraint that while the robot is spilling, it will stop moving along its path until it is no longer spelling liquid. This behavior is likely desirable for a "carry liquid policy," but it also may or may not be expected by a user who has only seen successful policy examples. We expect that, especially for users who do not expect this

stopping behavior, many of the beads will accidentally be poured into one or two flowers as opposed to an even spread. This is because, if the stopping behavior is unexpected, a user may need time to react to the robot stopping moving along its path once the beads start pouring out. We will refer to this as the *STOP* condition.

IODA In this condition, we apply the IODA algorithm while the user is engaged in PC. An OOD detector was trained on optimal policy rollouts before the study. We used the L1 distance as the distance function used in the algorithm. Because the IODA algorithm will result in the robot roughly following its original path regardless of the presence of PC, we expect that as users rotate the robot's wrist, beads will be evenly poured into the flower basin. Furthermore, this is the behavior we hypothesize that users will expect.

3.6.2 Experimental procedure

After participants read and signed an informed consent form, they practiced teleoperating the robot for up to three minutes. For practice, users were given XYZ control as well as roll/wrist control as the robot grasped an empty cup and were encouraged to get comfortable controlling the robot. The speed of roll rotation matched the speed during each condition. The purpose of this practice task was to ensure that all users had a similar minimum level of familiarity with controlling the robot before moving on to the pouring task.

After the practice session, we explained that the robot had an autonomous policy to carry cups of liquid. Users then watched the robot carry a cup of beads to three different locations. We will refer to this as the familiarization phase. After familiarizing themselves with the robot's behavior, they were then instructed that they would take control of the robot's wrist as it carried a cup of beads from one end of a flower bed to another. Their task was to try to water the five flowers as evenly as possible while using the most beads possible. Participants would then complete the pouring task in one of the 3 conditions (the choice of which was fully counterbalanced). The task ended after one minute or until the robot reached its goal pose, irrespective of how many beads they had already

successfully poured. Users completed a post-condition survey after their experience. We repeated this in each of the two remaining conditions. Finally, users were thanked and given compensation.

Outcome Measures The post-condition survey included questions from the UTAUT [266] survey. We adjusted the scale of all questions to a 5-item Likert-scale. We also asked two other Likert-scale questions: *How much did the robot's behavior align with your expectations?* and *I was surprised by the robot's behavior*, and two free-response questions: *Did the robot behave as you expected? If not, please explain how.* and *How much do you feel the robot's ability to complete the task depended on your input?.*

For a quantitative performance metric, we define "pour error." We measure the total deficit of the bins relative to an optimal pour of $w = 68$g each. We measure the deficits and not the overfills since measuring overfill would count this error twice. This pour error ϕ is equivalent to measuring the total amount of beads lost in the process, combining bin overflow, beads that did not land in bins, and beads remaining in the cup. Thus,

$$\phi = \sum_{i=1}^{5} \max(w - b_i, 0), \tag{2}$$

where b_i represents the measured weight of beads in bin i.

Hypotheses Based on what we know about how the robot will act under PC in each of the three conditions, we propose the following hypotheses. **H1**: IODA will most meet user expectations, followed by STOP and then RL; **H2**: IODA will lead to overall the best task performance, followed by STOP and then RL; **H3**: In PC, there will be a strong positive correlation between meeting a user's expectation and task performance.

3.6.3 Results

Participants We recruited 18 participants from the university and the surrounding area with a variety of different backgrounds. All participants were 18 years or older. Of these participants, 10 were female, 6 were male, 1 was nonbinary, and 1 was genderqueer. 13 participants were in the 18-24

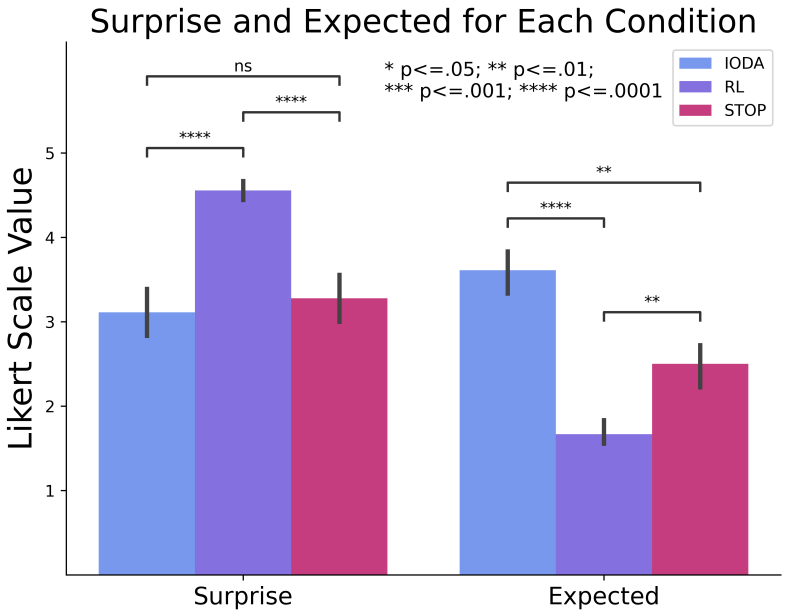


Figure 5: User reported expectation alignment and degree of surprise for each condition.

age range, 4 in the 25-35 age range, and 1 in the 35-44 age range. Of these participants, 2 were self-reported robot experts (i.e., attend robotics conferences regularly), while all other participants reported having interacted with a robot in the past (i.e., a Roomba). The study lasted approximately 30 minutes and participants were compensated $10. The study procedure was approved by the Tufts University IRB.

Analysis To analyze the data we use both Bayesian statistics and p-values. All tests were done using independent samples t-tests and the Bayesian tests were done with a Cauchy prior distribution with $r = 1/\sqrt{2}$.

User Expectations Before watering the flowers in any of the three conditions, as mentioned, users had both practice time and were able to watch the robot carry cups of beads to familiarize themselves with its movement. We expect these two initial phases, as well as any ordering effect, influenced how a user reported both how much the robot's behavior met their expectations and how surprised they were by the interaction. The results of the post-condition Likert-scale questions can

be found in Figure 5. As we can see, IODA led to robot behavior that both best aligned with people's expectations and induced the least amount of surprise. Specifically, IODA met user expectations to a greater extent over RL ($p \approx 0.0$, $\mathbf{BF} > 10000$), and to a slightly greater extent over STOP ($p \approx 0.0064$, $\mathbf{BF} = 7.05$). Comparing STOP to RL, we find STOP more closely meets user expectations ($p \approx 0.0099$, $\mathbf{BF} = 5.035$). A large part of why the RL condition least met user expectations is because the sporadic behavior caused by the RL policy being out-of-distribution when the robot's wrist was rotated was that it would begin to move away from the participant as opposed to towards and away from the center of the flower bed (Figure 7). These results support **H1**.

Task Performance The primary task metric we analyze is *pour error*. The results are shown in Figure 6. We find IODA led to much better task performance than RL ($p \approx 0.0$, $\mathbf{BF} > 10000$) and slightly better performance than STOP ($p \approx 0.020$, $\mathbf{BF} = 2.974$). Notably, in the STOP condition, many users reported "figuring it out" after some trial and error. This is partially captured in the time-on-task chart in Figure 6, although even an expert in the STOP condition would still take longer than in the IODA condition due to the nature of the stopping behavior. That being said, we do find IODA led to slightly better performance with significantly less time-on-task. Similarly, 5 of 18 participants did figure out that in the RL condition, they could wait for the cup to be almost at the end of the flower bed and then begin rotating the robot's wrist so that it would move back across the flower bed while pouring out the beads. However, this took most of the participants almost the entire 60-second trial time to realize. These results support **H2**.

The Importance of Meeting User Expectations in Partitioned Control We hypothesize that during PC, robot behavior that meets a user's expectations will correlate to higher task success. Although various collaborative shared control paradigms are designed to work despite a user's expectations or work under the assumption that a user and robot share a world model, as in classic SA [120], in PC, aligning user expectations and robot behavior is critical for task and user performance. We analyze the relationship between the user's reported expectation alignment and the user's reported surprise. The results are displayed in Figure 8. We find that there is a strong correlation between high

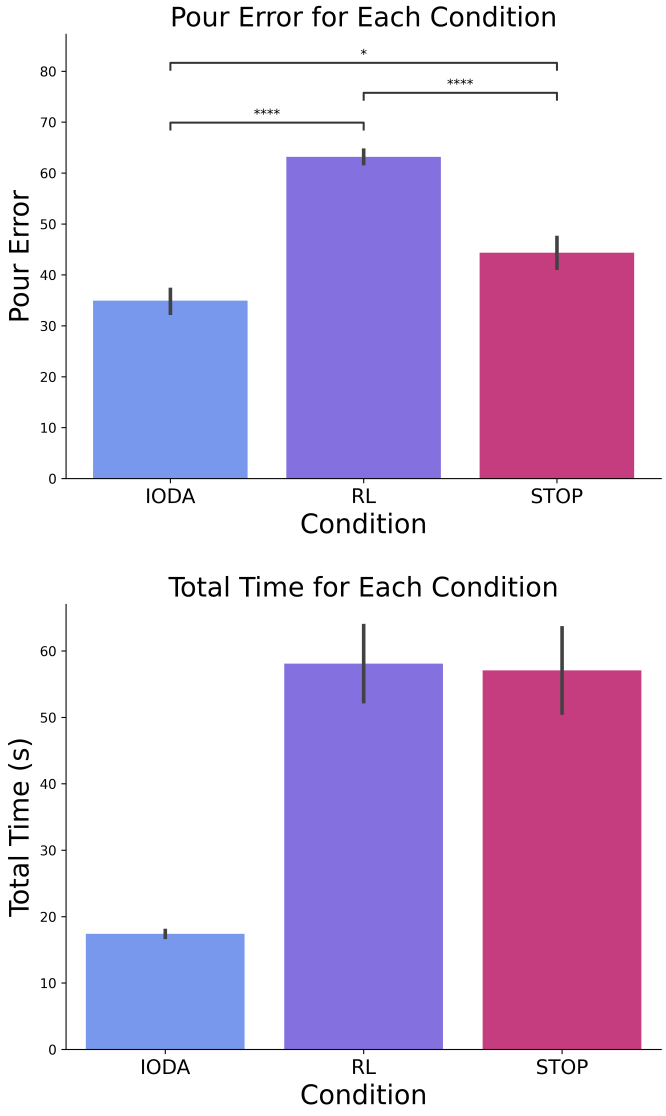


Figure 6: *Top*: IODA performed the best in the watering task with the least error. *Bottom*: Mean and standard-deviation for time-on-task for each condition

task performance (low pour error) and meeting user's expectations, as well as a strong correlation between low levels of surprise and high task performance. These results support **H3**.

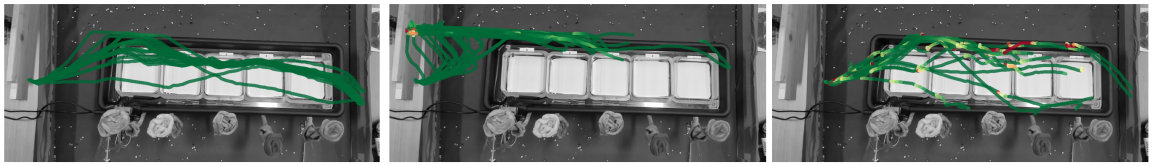


Figure 7: Trajectories of the cup for all 18 participants. The redder the line indicates how long the cup was stopped at that point. The reddest point indicates that the cup is stopped for at least 7.5 seconds

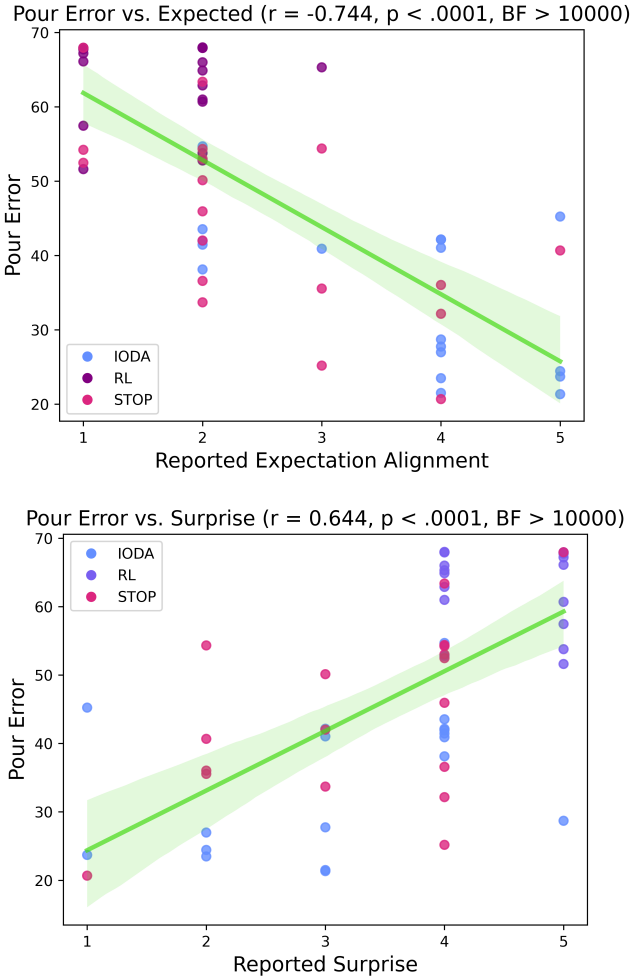


Figure 8: *Top*: Meeting user's expectations is strongly correlated with task performance in PC. *Bottom*: The same is true of reducing surprise and performance.

3.7 Limitations and Future Work

While IODA was shown to provide a positive user experience as well as enable high task performance in our user study, there are several limitations with the PC problem setting described and the IODA algorithm. In the PC problem setting presented in this chapter, we proposed an objective to identify when there exists a state s', a state that if passed to the policy would lead to behavior more aligned with the user's expectations than the real state s, that is in D. How often this is the case however, is largely dependent on how many trajectories are present in D and how diverse that set is. For example, if the user is attempting to perform PC with an original task that they have only seen once, $|D| = 1$, then there may not exist any state in D such that an s', is identified. In those cases, even if an algorithm like IODA is functioning properly, the interaction may not be changed since there is never a reason to act according to any other state than the actual state. An alternative to this approach, could be to loosen the assumption that s' is necessarily in D. Rather, if s' does not have to come from trajectories that the user has seen before, then s' could be *generated*, for instance, by a different algorithm. While such a generative approach would also have to make assumptions about the unobservable expectation function W, it is nonetheless more general as it could also include states present in D. Relaxing this assumption leads to a more generalizable problem setting.

The other primary limitation of the problem setting is the use of a distance function to compare state similarity. In robotic tasks where the task itself can be well represented with relatively low-dimensional geometric state-spaces, such as Cartesian space or end-effector space, a distance function to compare similar states can be useful. Beyond the navigation and pouring tasks presented in this chapter, states in other PC task, such as a user controlling the compass-direction a robot arm is pointing and the height of the robot arm to pass out hors d'oeuvres to people, as the other joints of the robot autonomously ensure the tray is balanced, can be related by distance metrics such as the L1 or L2 distance. However, there are also many tasks where to measure the distance between different states in a meaningful way requires a more complicated and task-specific distance function.

For example, if a user wished to control a single arm of a bimanual humanoid robot using PC, a geometric distance metric over the entire state, may not appropriately capture state similarity with respect to the user's expectations. This is both because of the various nonlinearities between different components of the state space and because the user may only care about how the other parts of the robot arm is moving, and not necessarily how the robot is standing, for example. Thus, the choice of distance function for the problem setting is task-dependent and may be difficult to choose in advance of the interaction. Recently, there has been much work in robot learning that attempts to capture the similarity between certain states and tasks more generally for transfer learning applications [60, 153, 294, 146]. Works such as these may be useful for providing other distance functions as well as automating the choice of distance functions for algorithms like IODA.

IODA is an algorithm which rather directly addresses the PC problem setting. Because of this, most of its limitations are shared with the problem setting itself. The primary limitation unique to IODA has to do with the introduction of the notion of the agent being out-of-distribution. IODA uses an OOD detector to determine when the algorithm should search for an s' in D. This has the benefit of saving the computation time of always searching through the a robot's history every state. However, what is OOD as defined by a history of successful trajectories, may not always align well with when the agent should be imaging a different state. For instance, a state may be in-distribution from the robot's perspective, but the behavior of the policy in that given state may nonetheless not align well with the user's expectations and thus a different state should be imagined. A potential way of mitigate this issue is to establish "common ground" between the user and the robot about what is and is not OOD. This can be done through a collaborative labeling of OOD trajectories between the user and the robot. If the person and the robot both agree on what states or behaviors are OOD, then the robot can better know when to imagine different states or how to better align its behavior with the user's expectations.

3.8 Discussion

A naïve learned robot policy may not be suitable for flexible interactions with real users, especially when they have the propensity to use the policy in unexpected ways. Such a propensity is not exclusive to human-robot collaboration: people will use a shovel as a crowbar, a crowbar as a hammer, a hammer as a hook, and a hook as a shovel. We investigated Partitioned Control (PC), in which a user controls some dimensions of the behavior of an RL-trained robot and can use that control to drive it into states that are not reflected in training. We present an approach, Imagined Out of Distribution Actions (IODA) that enables such a partially-controlled system to behave in alignment with user expectations. We demonstrated that a standard RL-trained agent will behave erratically under PC, while IODA results in more expected robot behavior. Furthermore, we show that in a realistic PC setting, when a robot's behavior is more aligned with a user's expectations, the user can more effectively perform the novel task they are trying to achieve.

There are, however, aspects of this which warrant further investigation. One is to study how users build up their expectations before and during PC. Here, we assumed that user expectations are based on teleoperation experiences and viewing prior rollouts of a given policy. However, there may be other important factors. A second aspect is the use of distance functions over the state to quantify user expectation alignment. We assumed that a distance function can be used as a proxy for what a user considers "similar" states, and we used an OOD detector to approximate when the robot is in a state that a user is unfamiliar with. While this approach was effective in our user study, there is more to learn about the properties and assumptions of IODA and the use of imagination to better meet user expectations. IODA may also run into latency issues if the calculations of the distance between states are not relatively fast.

Our study addressed a "one-shot" interaction where users performed the flower watering task with each condition once. This is an important setting because in many real-life scenarios, it is ideal for a task to work on the first go. Enabling "one-shot" interactions improves user satisfaction and generates successful demonstrations that *could be used to learn the new task in the future*, through

an LfD algorithm for example. That said, most users, across all three conditions and regardless of whether the robot met their expectations, wanted to interact with the robot again. This is not only because they enjoyed the task, but also because they thought they could better perform the task knowing what to expect. Thus, user expectations change as a result of PC interactions, and future work is needed to address how PC and IODA change these expectations over time.

4 Customizing Behaviors

4.1 Introduction

In this chapter, we introduce a novel problem formulation and algorithm for creating customizable and personalizable robots. Robot's that are trained through RL, typically only learn how to perform a task in a single way. In the case of IODA, we demonstrated how that predictability could be leveraged by a person to perform new tasks. Here, however, we focus on giving users the impetus to control the style of *how* the robot completes that task. This is important as the way the robot initially learned to do the task, may not suit the user's needs or preferences, it could even make them uncomfortable in a worse case. Furthermore, users should not be burdened with having to entirely re-teach the task or their preferences to the robot because of this mismatch. A further benefit of giving users control over the robot's behavior in real-time is that they can use that behavior for creative tasks that require on-the-fly behavior adjustments: such as painting in this work, but could also apply to dance, sculpture, etc. This chapter introduces *online behavior modification*, a problem formulation to enable such real-time control and adjustment to further equip novice users with greater degrees of robot control.

Online behavior modification is a novel problem formulation for training RL agents that ensures the final learned policy of the robot is both capable at the task and has behavior with can be easily adjusted by a non-expert user. Online behavior modification has three key components or requirements: the robot must always be making progress on the task, this ensures that as the user customizes how the robot completes the task it does not adversely impact task completion itself; next, there must be behavior features, or stylistic aspects of the robot's behavior that the person can adjust in real-time; lastly, the adjustment of those behavior features must be accessible to the user, in this case accessible both in terms of the interface as well as ensuring the adjustment of those features is interpretable and predictable. These components support both user empowerment and robot task performance.

To deploy online behavior modification and study peoples experience with it, with developed the Adjustable Control of Reinforcement learning Dynamics (ACORD) algorithm. ACORD combines RL-based task learning with behavior-diversity inspired skill learning. This allows ACORD to simultaneously learn an RL task while also learning how to change its style of behavior. A person can then change the input parameters of ACORD, and consequently the robot policy will attempt to match that specified style. We evaluated ACORD both in simulation to test if it meets the requirements set by online behavior modification and in a user study. For the user study, we designed a painting task wherein the robot must trace over thin lines that outline a shape whilst the participant can vary the robot painting style as the robot paints. This task allowed us to both quantitatively measure task performance through calculating how well the outline was traced; and to measure users qualitative experience expressing their own style and creativity. In the study, we compared ACORD to Shared Autonomy (SA), during SA users have direct low-level control of parts of the robot behavior, and the option of being able to pick among different styles in advance, each trained with RL. We find that ACORD provides users with a strong sense of control over robot behavior, allowing them to express their style and preferences in real-time, similar to SA, while maintaining task performance more similar to RL.

The majority of the work in this chapter was published at HRI 2024 as Sheidlower, Isaac, Mavis Murdock, Emma Bethel, Reuben M. Aronson, and Elaine Schaertl Short. "Online Behavior Modification for Expressive User Control of RL-Trained Robots." In Proceedings of the *2024 ACM/IEEE International Conference on Human-Robot Interaction*, 639–48. HRI '24. New York, NY, USA: Association for Computing Machinery, 2024. `https://doi.org/10.1145/3610977.3634947` [238] [†].

[†]The second author helped organizing the photos of participant paintings and the appendix. The third author helped with visualizations for figures. The second to last author consulted on the experiment design and assisted in programming the shared autonomy condition used in the study. The last author supervised the work and consulted on the experimental design.

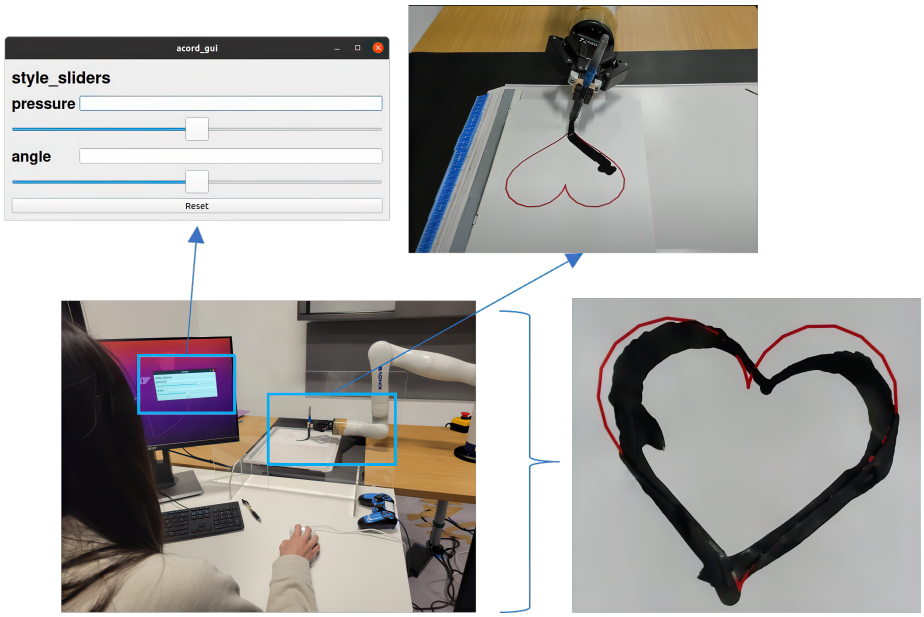


Figure 9: A participant using ACORD to adjust the style of a painting as the robot traces a heart autonomously.

4.2 Learning Policies for Online Behavior Modification in RL Settings

To enable human-centered control over how a robot complete its task, we propose three key properties for *online behavior modification*. First, the robot must always **autonomously make "task progress"** and ensure the task does not fail. In this context, "progress" may mean "expected completion in finite time" or "always getting closer to a goal"; formalization depends on the task. Second, there must be a non-empty set of **behavior features**, each of which has an associated *behavior oversight parameter*, k, that control the robot along the behavior feature axis. In other words, the policy must be explicitly parameterized with one or more observable variables that dictate an aspect of the robot's behavior. Finally, for each behavior feature that has a certain k associated with it, the adjustment of that k must be **interpretable to a user** and there must be an **accessible interface** that facilitates a user to freely adjust each k as the robot completes its task. These properties describe an interaction that ensures the user can have a robot that both meets their needs and can be personalized without having to teach the robot the task or their preferences.

In this section, we present Adjustable Control Of RL Dynamics (ACORD), a proof-of-concept

algorithm for learning a policy for online behavior modification in continuous state and action space robotics tasks. ACORD is a behavior-diversity–inspired algorithm which explicitly gives users control over a robot's behavior. We describe how to adapt a standard RL setting to facilitate ACORD and demonstrate it in a simulation environment.

4.2.1 ACORD for Continuous Control RL-tasks

We assume a task modeled as a Markov decision process (MDP) with states S, actions A, transition function $T(s, a) \rightarrow s'$, and discount factor γ. To define task failure, we assume some environmental reward function R_{env}. To this system, we introduce *behavior oversight parameters*. Assume that $S = \mathbb{R}^n$ and define the space of behavior oversight parameters as $K = [0, 1]^m, 1 \leq m \leq n$. Consider the coordinate representation of $s = \langle s_1, \cdots, s_i, \cdots, s_n \rangle$ and $k = \langle k_1, \cdots, k_j, \cdots, k_m \rangle$. Each coordinate of k, k_j, controls a coordinate of s, noted s_i. The set of all s_i that have a k_j mapping to them define a set of *behavior goals* for the robot, and the corresponding i-axes are *behavior feature* axes. Any s_i with no corresponding k_j is a free variable whose value is not explicitly constrained by a setting of k. For generality, we assume the range of behavior goals is unknown prior to learning (e.g., the maximum and minimum speeds the robot can move while completing its task are unknown). After learning, a user can directly adjust the values of k, thus changing the robot's behavior goal on the axis s_i, and consequently changing its behavior along that axis within a range that is learned by the algorithm, subject to "non-failure" condition above. This representation could be trivially extended to having k_j control multiple coordinates.

Learning a policy for ACORD entails finding a policy parameterized by k, π_k, which both makes progress in the task and enforces the behavior goals. To ensure that the learned mapping from each k_j to each s_i is interpretable by a user, we propose the soft constraint that the robot should learn a monotonic mapping from k_j to s_i and that the mapping range is as large possible without preventing the robot from completing its task.

4.2.2 ACORD Algorithm

ACORD makes use of three components: a discriminator that learns a continuous mapping from $s_i \rightarrow k_j$ to generate a diversity-inspired reward; an environment reward to define failure states and a task progress heuristic $h(s, a)$ to ensure task performance; and a domain randomization component that ensures that the agent learns and is robust to various different settings of k such that k may be adjusted in real time.

ACORD Discriminator We train a set of discriminators W_j to predict k_j given s_i, denoted: $W_j(s_i) \in [0, 1]$. We parameterize the discriminator as a neural network and train it via the novel loss function:

$$L(W_j(s_i), k_j) \quad = \quad MSE(W_j(s_i), k_j) \quad + \quad \frac{1}{|\max(W_{j,s_i \sim D}(s_i)) - \min(W_{j,s_i \sim D}(s_i))| + \varepsilon} \tag{3}$$

where $W_{j,s_i \sim D}$ refers to the discriminator output of a batch sampled from a replay buffer D_W, and ε is a small number to avoid division by zero. This loss function enforces high prediction accuracy (via MSE) and that the predictions cover as wide a range over the behavior oversight parameter as possible. The latter property is explicitly enforced by the denominator, leading to a faster convergence to the range of behavior covered by each k_j, resulting in more stable task behavior (see supplementary material for ablation study).

RL Task Description and Agent We define the state space of the RL agent to be $S \bigcup K$. This makes k observable to the agent so that when a user adjusts a specific value in k, the agent knows to adjust its behavior. We will denote any state from the environment without k as s_env, and use still use s to refer to the state with k appended to it. We design a reward function such that the agent avoids failures, makes progress, and learns to enforce behavior goals:

$$R(s,a) = \begin{cases} R_{\text{env}}(s_{env}) & \text{if } s_{env} \in F^* \\ -c & \text{if } h(s_{env}, a) \leq 0 \\ \frac{1}{m}\sum_{i=1}^{m}(-\log|W_i(s_i) - k_i|) & \text{else} \end{cases} \tag{4}$$

where R_{Env} denotes the reward from the environment, F^* is the set of failure states which lead to a large negative reward, $h(s_{env}, a)$ denotes a heuristic for measuring task progress, and c is a positive constant that punishes the agent if it fails to make task progress. Lastly is the reward generated by the discriminator which ensures that, for a given k_i, the agent is acting in the part of the state space where the discriminator can easily predict the k_i value. Since $|W_i(s_i) - k_i| \in [0, 1]$, this reward is always positive and the other conditions are always negative. This allows the reward function to be adapted and scaled to different environments with relative ease. Each of these terms may be scaled by a constant. We maximize this reward via the off-policy RL algorithm SAC [99].

Domain Randomization Over K We employ domain randomization [261, 179] for the setting of k during training. Every n time steps, we sample $k_i \sim \text{Uniform}(0, 1)\forall k_i \in k$. The choice of n can be difficult as when a given k_i changes, it may take several steps for the robot to adjust its behavior accordingly. If n is too small, the algorithm cannot learn to enforce the value of k over time, and if n is too large, it cannot learn to react efficiently to a user changing k real time. Empirically, we find in the tasks in this paper that a reasonable choice for n is about half the length of an episode; we expect that this would be the case for many tasks.

4.2.3 On Using a Heuristic Progress Function

Online behavior modification as an interaction emphasizes that the robot can autonomously complete the task by constantly making progress in that task. There are several ways to formalize this constraint, and online behavior modification does not necessarily require a particular one. For example, in this work we define a task progress measure $h(s_{env}, a)$ and require that π_k prioritize trajectories that make $h(s_{env}, a)$ non-negative; this approach is appropriate for many robotics problems

Algorithm 2: ACORD

Initialize off-policy RL Learner Ψ

Initialize Discriminator(s) W

for *environment step* t **do**

  if *nth step* **then**

    $k \sim \text{Uniform}(0,1)^m$

  $s_t \sim s_{t,\text{env}}$ concatenate k

  $a_t \sim \pi_\Psi(a_t|s_t)$

  $s_{t+1,\text{env}} \sim T(s_{t+1}|s_t, a_t)$

  $s_{t+1} = s_{t+1,\text{env}}$ concatenate k

  $r_t \sim R(s, a)$ [see Eq. 2]

  $D_\Psi \leftarrow D_\Psi \bigcup (s_t, a_t, s_{t+1}, r_t)$

  $D_W \leftarrow D_W \bigcup (s_t, a_t, s_{t+1}, r_t)$

  if *zth step* **then**

    Update Ψ via gradient descent

  if *vth step* **then**

    Update all W via loss in Eq. 1

where there is a physical destination for the robot's motion (e.g., [167]). Another natural approach might be to use the environmental reward function $R_{\text{env}}(s_{env}, a)$ to measure task progress or require that the trajectories following π_k eventually reach a terminal success state. The exact specification will depend on the task and the formulation of the learning problem.

A heuristic progress function h can ensure the robot always completes the task despite a user changing how it does so. This aligns with our goal of giving users the most control possible over a robot's behavior while still accomplishing the task. This is in contrast to prior approaches that optimally solve for a trade-off between environmental reward and diversity, as in Quality-Diversity-based approaches [24, 25], or use a hyperparameter to dictate how each of the two objectives are weighted [26].

4.2.4 ACORD in Simulation

We train ACORD in simulation to show that the learned policy has the desired properties: it aligns pre-specified behavior features to the values specified by k; it has an interpretable behavior range over ks; and it completes the task and avoids failures robustly in variations in k. In a bipedal walker task [37], we specify two behavior oversight parameters: k_1 to control the speed of the robot along its x-axis and k_2 to control the angle of its hull. Failure cases are specified as crashing (-100 reward from the environment). We measure task progress by setting $h(s, a) = v_x$, the velocity of the robot along the x axis. Then, Equation 4 penalizes the system for moving backwards in x. We trained the agent to convergence prior to evaluation ($\sim$2 million steps; for a discussion of algorithm efficiency see Section 6). Figure 10, left, shows the resulting behavior by varying both ks. By changing k_j, there is a predictable change in behavior along the specified feature axis. Figure 10, right, shows the range over the robot's speed for various settings of k_1 given across different values of k_2. This demonstrates that ACORD can be robust to multiple settings of k_1 given k_2: varying the hull angle does not fully constrain the agent's ability to vary its speed. Of course, if two features are directly in conflict with each other, such as a k_i mapped to going backwards and a k_j mapped

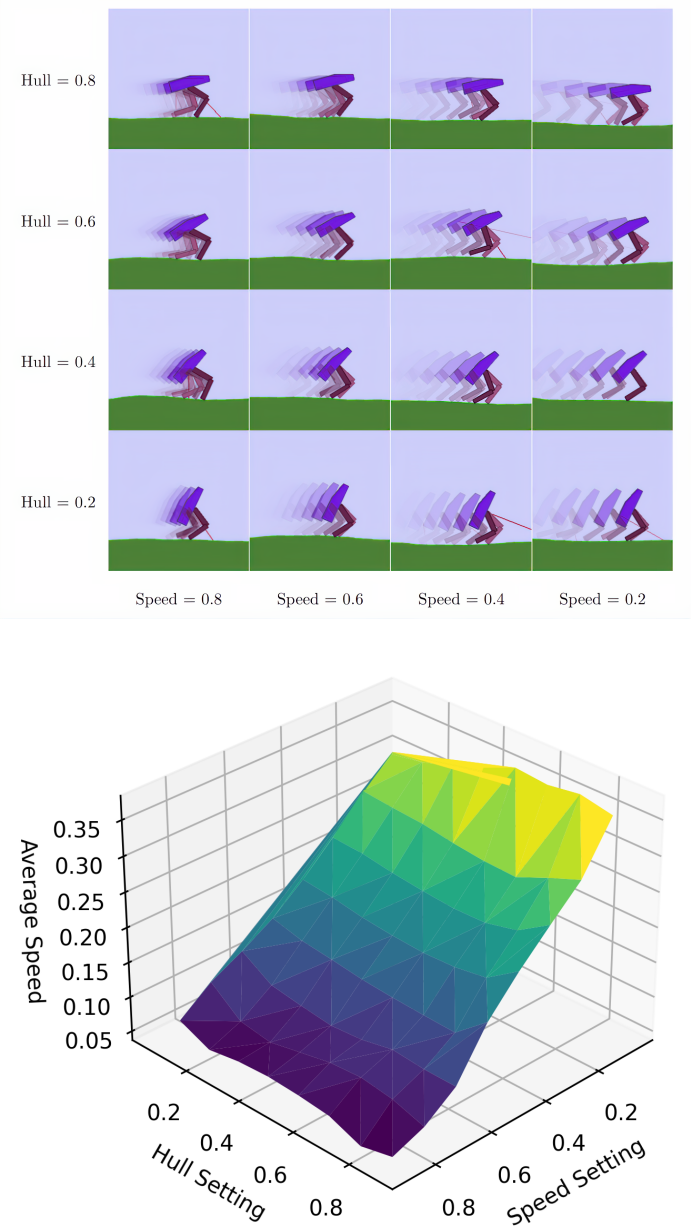


Figure 10: **Left**: The walking agent varies its behavior in a predictable and interpretable way given changes of k. The ghost traces from the previous six video frames show the agent's change in speed. **Right**: The resulting manifold learned by ACORD in the walker environment. The speed is robust to different hull angles.

to going forwards, the behavior of the robot may not be as expected. Lastly, over multiple runs, the agent avoids crashing $\sim 94\%$ of the time with variations in many settings of K.

We also evaluated the agent's task performance for different settings of each behavior oversight

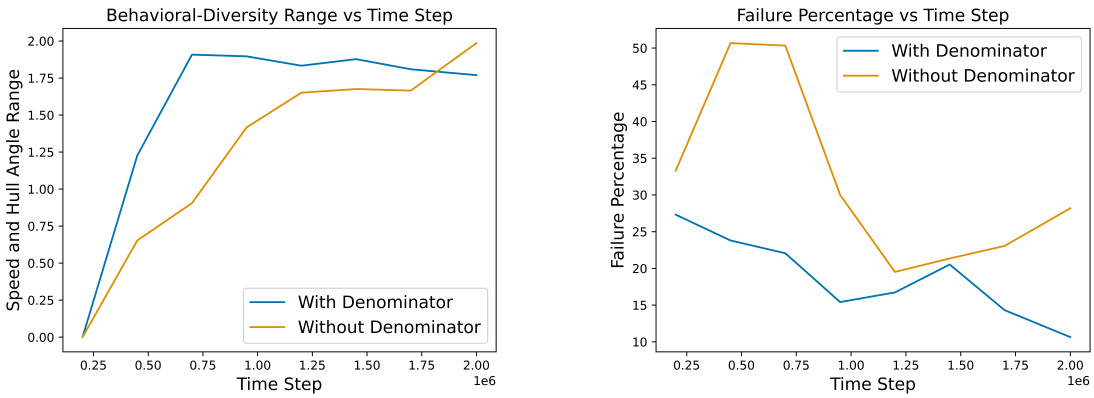


Figure 11: ACORD ablation study.

parameter k, dictated by a 10×10 grid ($k_1 = \text{hull} \times k_2 = \text{speed}$). The agent trained with just k_1 (speed) varying was evaluated for $k_1 \in \{0, \frac{1}{n-1}, \cdots, 1\}$ with $n = 10$ different settings of k_1. This agent avoids crashing 85% of the time ($\sigma = 25\%$) and the agent trained with one k avoids crashing 87% of the time (σ =22%). Furthermore, the majority of failures took place on boundary settings of k ($k \sim 0$ or $k \sim 1$); if we exclude those endpoints, the agent trained with one k avoids crashing 93% of the time ($\sigma = 15\%$), and the agent trained on two ks, avoids crashing 94% of the time ($\sigma = 9\%$) . The propensity of the agent to avoid crashing also has to do with the different chosen reward scales. There is a trade-off between the penalty of failure and the time it takes to learn a large behavior manifold.

We note that the min and max average speed across all ks is between $\sim$0.02 (roughly standing still) and $\sim$0.45 respectively. While the max speed a typical RL agent would run is $\sim$0.7, the ACORD agent is relatively robust to changes in k, thus being able to dynamically change its speed.

Ablation Study We performed an ablation study to study the effect of the denominator in Equation 1. The denominator explicitly enforces that the predictions of the discriminators W for any given replay buffer sample cover a large range of k. The loss implicitly encodes this property without the denominator and will learn to predict a wide range of k naturally through optimization. We have found, however, that this explicit term encourages this behavior earlier on and leads to a more stable agent performance as it pertains to making task progress. This is likely because the

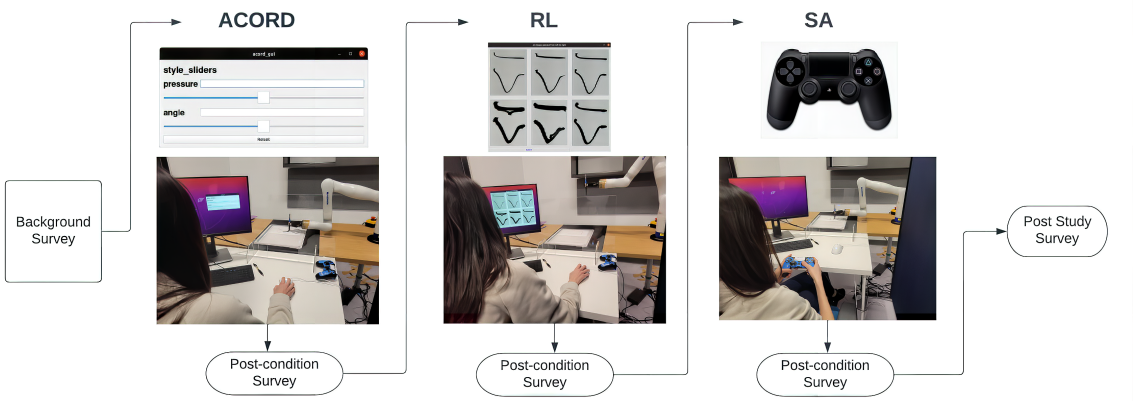


Figure 12: Overview of the study procedure. Participants interacted with each of the three conditions (order was counterbalanced), completing a survey after each condition.

faster convergence to predict a given range of k leads to a more stationary reward function earlier in training.

We tested this hypothesis in an ablation study in the bipedal walker environment with the same two k representations. Every 250,000 time steps, we evaluate the diversity of behavior and task performance for 10 episodes of both an agent trained with the denominator term and one trained without it. The results can be seen in Figure 11. The left graph shows the sum of the average range of the speeds and hull angles (both ranges were normalized to be between 0 and 1) over time. The right graph shows the percent of critical failures over time. As we can see, the agent with the denominator converges to a wide range of behaviors relatively quicker and does so with significantly fewer critical failures.

4.3 User Study

To study ACORD and online behavior modification with real users, we designed a robot painting environment wherein users can adjust a robot's painting style as or before it traces a drawing. This domain is an inherently creative activity in which a person has styles and preferences that they wish to express. Online behavior modification captures the idea that task completion itself is not always the only desirable metric of a human-robot interaction: having control over *how* the task is

completed can also be an important factor, as is the case with painting and other artistic tasks.

Robot Painting Task The painting task involved the robot tracing a previously generated shape. We specify each shape as an ordered list of waypoints in the x-y plane, $(p_0, \cdots, p_r)$. We formulate the task as an MDP where the state s_{env} is a vector containing the robot's end-effector position, orientation, and velocity; the position and orientation of a brush the robot is gripping; and the next waypoint that the robot should reach. Actions are relative Cartesian x-y velocities. Reward is given as $R(s_{env}, a) = -|p_{\text{brush}} - p_i|$, the negative distance between the current pose of the brush p_{brush} and the next waypoint p_i. Episodes terminate when the robot has reached every waypoint that makes up the shape or with failure when the arm leaves the workspace or is in collision.

Experimental Setup The setup (Figure 12) consisted of a Kinova Gen3 robot arm on a table with the participant sitting next to it. Depending on the condition, users had access to a different interface to interact with the robot. On the table was paper with a shape outlined in red on which the robot would paint. The participants were told which shape they would paint: heart or house (Figure 13). These shapes contain various motions and strokes and provide scope for participants to paint in their own style.

Painting Styles We define two different axes for the robot to vary its painting style. One is by adjusting the height of the brush or end-effector, thus affecting the pressure that the brush applies to the canvas. This can result in thinner or wider strokes. The other way is by rotating the robot's wrist or brush. This adjusts the angle of the brush, resulting in more varied strokes.

4.3.1 Conditions

We assume for all conditions that the robot knows how to perform the task optimally according to the MDP formulation. We fix the painting policy across each baseline to ensure the same amount of time is spent on each painting and that the style adjustment was the primary difference between conditions. We compare ACORD to two alternatives to vary the style of robot behavior: RL and SA.

Choosing Among a Discrete Set of Style-Varying RL Policies This condition gave the robot the most autonomy. Participants selected one of six styles based on an example image before the robot drew the shape. Each style represented a fixed value for the pitch and height of the end-effector. The robot then painted the shape autonomously according to that selected style. This type of control, in which a user chooses between a set of RL policies, is appropriate for tasks where RL control is necessary and/or available and "styles" are well defined, such as choosing a "risky" or "risk-averse" obstacle avoidance strategy. In other cases, these pre-defined policies may have been learned via human feedback, but their execution during this single task is fixed.

Shared Autonomy (SA) This condition gave the participants the most direct control. Users were given assisted velocity control over the height and pitch axes of the robot end-effector through a controller. The input was augmented with a SA assistance strategy following [120, 119], with $\alpha = 0.5$ to allow the user's commands to directly influence the robot position [183]. The SA assistance infers online which of the six styles defined in the previous condition the user is intending to achieve.

While similar to the standard goal-based SA paradigm, we note two key differences. First, the system continuously moved along the x-y plane via the optimal policy while the user controlled the style axes. Second, rather than considering goal states to be terminal, the user continued to control the style axes for the whole trajectory and could move from one goal then to another. This approach allows for the closest comparison between ACORD and SA, but this multi-goal formulation of SA is a direction for future research in itself.

Adjustable Control of Reinforcement learning Dynamics We trained and deployed an ACORD agent using sim-to-real via the Gazebo simulation environment [139]. Failure was defined as leaving a set workspace. We defined $h(s_{env}, a) = a \cdot (p_i - p_{\text{brush}})$, the component of the action in the direction towards the current waypoint p_i. Penalizing $h(s_{env}, a) \leq 0$, as in Eqn. 4, penalizes actions that move away from p_i.

Two ks were learned to allow for *continuous control* over the painting style: one for the height, k_1, and angle, k_2, measured at the *brush tip* rather than at the robot's end-effector. This means

Figure 13: Participant paintings. Users were able to produce a wide range of different styles for the pre-specified shapes, including the emergent "polka dot" style in SA (4th column from left) and widening or narrowing "strokes" using ACORD (rightmost column, top and center).

when a user moves the slider to adjust the brush's rotation, through k_1, ACORD maintains contact with the paper since k_2 stays the same. The users had access to a GUI with two sliders to control both ks. Users adjusted the sliders, affecting the robot's behavior and painting style in real time.

4.3.2 Experimental Procedure

Recruitment We recruited a total of 24 participants from the university and the surrounding area with a variety of different backgrounds. All participants were 18 years or older. Of those participants 15 were female and 9 were male. 13 participants were in the age range of 18-24, 9 in the range of 25-35, 1 in the range of 35-44 and 1 in the range of 55-64. Participants reported their level of programming expertise from 0 (none) to 10 (expert). The mean level of programming experience was 2.9 with a standard deviation of 2.3. Furthermore, 11 participants reported having experience interacting with robots, and 3 of those 11 had significant expertise (attending robotics conferences and events regularly). The study lasted approximately 45 minutes and participants were compensated $15. Of the 24 participants, the data from one participant was excluded due to non-participation (ignoring the robot's behavior and providing only uniform feedback on all surveys).

This left data from $n = 23$ participants for analysis. The study procedure was approved by the Tufts University IRB.

Procedure Participants provided informed consent then took a background survey. The experimenter then explained the task and control in the conditions, including allowing participants to practice with SA and ACORD. In each condition, participants painted the house shape and then the heart shape, then filled out a survey about that condition. Conditions were fully counterbalanced within subjects. Finally, participants completed a post-study survey, were thanked, and given compensation.

Outcome Measures The post-condition survey included NASA TLX [260] and UTAUT [266] surveys. We adjusted the scale of all questions to a 5-item Likert-scale. We also asked two other Likert-scale questions: *I had control over the robot's behavior* and *I could express myself through the robot*, and an open response question: *How much do you feel the robot's ability to complete the task depended on your input?* The post-study survey had participants rank each condition based on their preference, the ability to express themselves, the perceived reliability of how well the robot traced the shape, and which mechanism (e.g. controller or sliders) they preferred. In addition, it asked two open response questions: a request for general comments and the question *how could the interactions be improved?*

We evaluated two quantitative metrics for how reliably the shape was traced. For each painting, we calculated the *coverage*, or percentage of the red line that remained visible in the image after the task was complete. We also calculated the *consistency*, or the coverage of the red line after applying translations and rotations of the painting to best align with the shape of the red line.

Hypotheses We expect that ACORD will give users control over the robot's behavior while still effectively completing the task, as users have more direct control than RL but less than that of SA. Thus, we expect that ACORD will be the most preferred approach and that it will give users feelings of slightly less control as SA while having similar performance to RL. This results in three hypotheses:

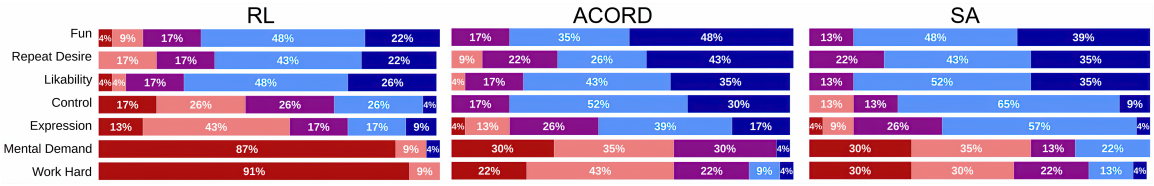


Figure 14: Responses to post-condition 5-point Likert scale questions. The darkest blue represents "strongly agree" or, in the case of Mental Demand, "very high." The darkest red represents "strongly disagree" or, in the case of Mental Demand "very low."

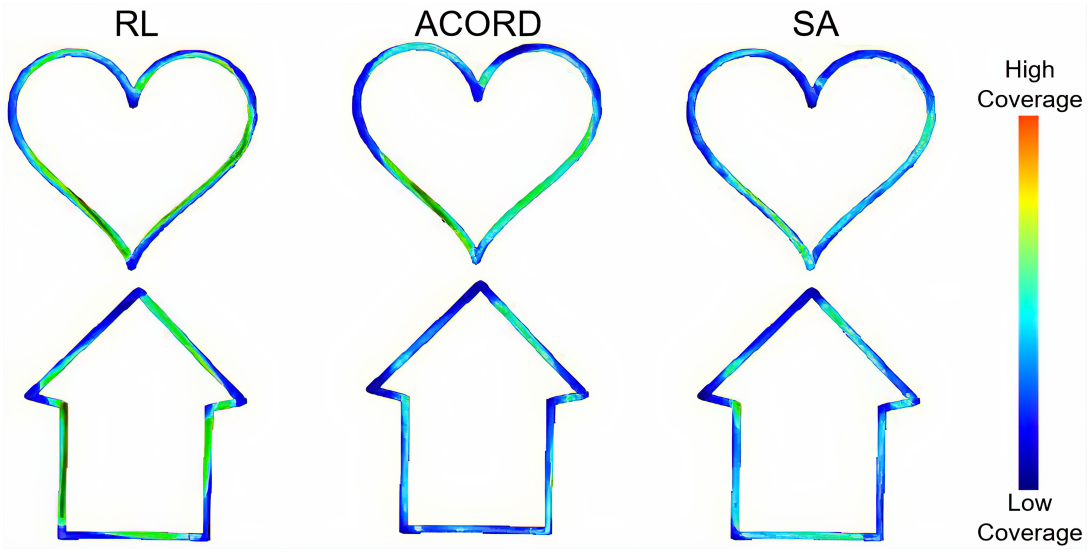


Figure 15: Heatmaps depicting the *consistency* of each approach sorted left to right from most consistent overall to least consistent. The heatmap consists of the participant's paintings layered on top each other after being shifted for maximal coverage. Areas of high coverage depict areas where many participants painted over, and vice versa for areas of low coverage.

H1: Users will prefer to interact with ACORD over SA and RL.

H2: Users of ACORD will feel more in control of the robot than in RL but less than in SA.

H3: RL will be objectively and subjectively the most reliable, ACORD the second most and SA the least.

4.4 Results

To analyze the data, we use Bayesian statistics following the interpretation scheme presented in [66]: a Bayes Factor (BF) between 3 and 10 we interpret as "moderate evidence" for the alternative hypothesis, between 10 and 30 as "strong evidence," and 30 or above as "very strong evidence."

To evaluate the post-study survey data, we encoded responses as pairwise comparisons between two of the three conditions. For each comparison, the rank was encoded as 1 if the "left" condition was preferred, -1 if the "right" condition was preferred, and 0 if the participant ranked the two conditions equally. To analyze this data, we used a Bayesian Wilcoxon Signed Ranked test with a Cauchy prior distribution with $r = 1/\sqrt{2}$. To analyze the Likert scale data , we used a Bayesian Repeated Measures ANOVA. We used a Bayesian Paired Samples T-Test to analyze the coverage and consistency metrics.

User preferences We find strong evidence that ACORD is preferred over RL (**BF**=17.16) and anecdotal evidence that people prefer SA over RL (**BF**=2.11). There is strong evidence that people found ACORD more fun than RL (**BF**=79.87) and moderate evidence people found SA more fun than RL (**BF**=5.03). These results provide support for ACORD being preferred over RL while being no less preferred than SA. We also find a trend towards ACORD being preferred to a greater extent over RL than SA. Finally, we found that users rated RL as much less mentally demanding than SA and ACORD (**BF**=112.87 and **BF**=45.92 respectively), and much less hard work (**BF**>10000 and **BF**>10000), although the previous results suggest this was not a significant factor in user preferences. These findings partially support **H1** and directly support that ACORD provides at least as much benefit to user experience as SA.

User Control and Expression In the post study-survey we find strong evidence that people find ACORD and SA more expressive than RL (**BF**=18.40 and **BF**=13.65) and similarly for the post-condition survey measure of expressiveness (**BF**=23.38 and **BF**=40.31). Users also found a greater sense of control with ACORD and SA (**BF**=6318.61 and **BF**=40.31). There is anecdotal evidence that users reported more control in ACORD than SA (**BF**=2) and differences between the two were often commented on in open-ended responses. These results support the first part of **H2**, that users felt more in control in ACORD than in RL, however our results suggest that some users may have felt an even *greater* sense of control in ACORD than in SA.

Quantitative Painting Analysis We find on average, across both shapes, ACORD and SA

had better coverage than RL (**BF**>10000 and **BF**=1095.2), likely due to the persistent offset in the RL condition caused by *bristle drag* of the brush. We account for misalignment by computing the maximum coverage found over small translations and rotations of the template, which we refer to as consistency. As expected, RL has better consistency than SA and ACORD in both shapes and, in general, the normalized sum across both shapes (**BF**>10000). While SA has higher consistency in the house shape (**BF**=1884.64), ACORD has much higher consistency in the heart shape and a higher consistency overall (**BF**>10000 and **BF**=11.67). A visualization of the consistency results can be found in Figure 15. According to our two reliability metrics, **H3** is supported by the consistency metric and not by the coverage metric. The coverage findings, however, showcased how a human in the loop can use the flexibility of added control to compensate for execution-time limitations in pre-trained RL models.

Qualitative Results Figure 13 shows paintings from each condition that are representative of the different painting styles found and the *emergent behaviors* that users demonstrated. With ACORD, we see the emergent behavior of brush strokes, where users moved both sliders quickly to make a specific stroke. In SA, some users made polka dots by bringing the brush up as much as they could, releasing the joystick, then letting the assistance bring the brush back to the paper. This was a surprising use of SA and goes against the task description of tracing the shape, yet gave users who figured this out a new way of expressing themselves and highlights that users had a desire for control and creativity in the task. While both ACORD and SA enabled this control, many users emphasized "consistency" and "ease of use" when describing ACORD; in contrast, users described SA as "mentally demanding" or "too sensitive." Some users did not enjoy that ACORD required "shifting their eyes" from the screen to the robot, although of course this is an issue with the interface and not with ACORD itself. RL was criticized for not being able to adjust the style in real time; however, multiple users said it would be ideal for a "mass production" setting.

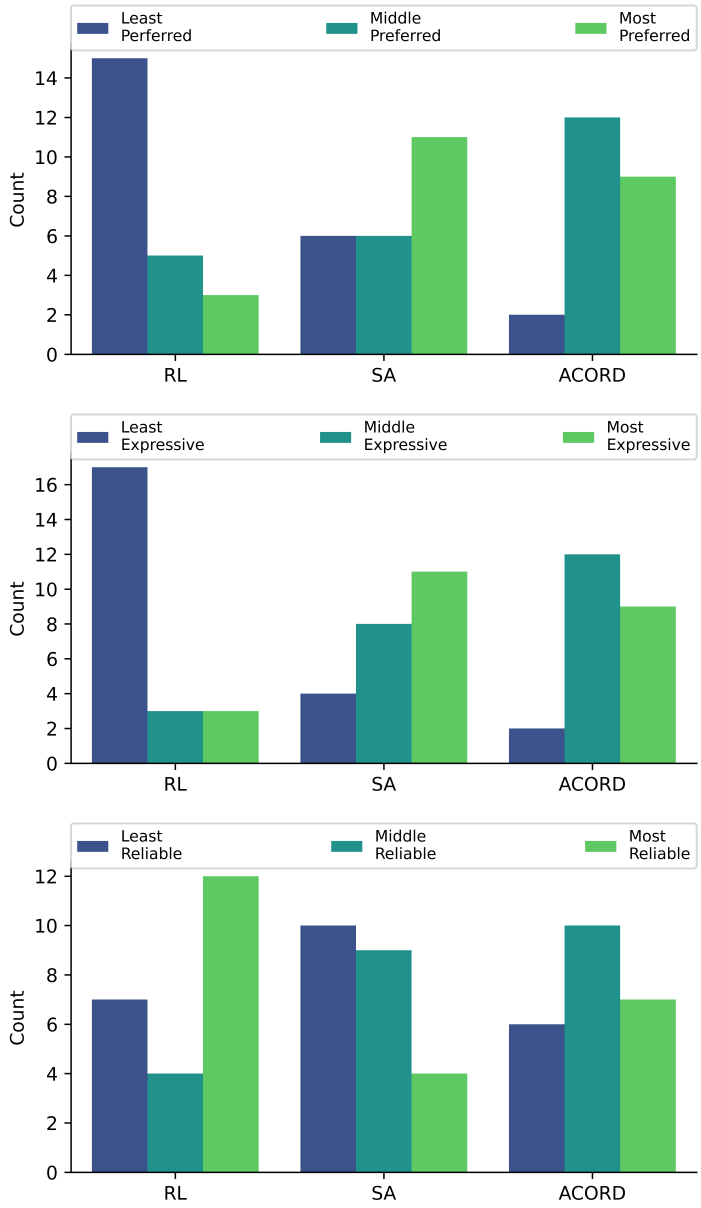


Figure 16: Results of the post study surveys. Users ranked each condition based one their preference (top), perceived expressive potential (mid), and perceived reliability (bottom).

4.5 Discussion

Online behavior modification describes an interaction in which a user has control over how an otherwise autonomous robot completes a task. While prior work has offered various algorithmic avenues to fulfill this type of user control, such as GCRL or Skill Learning, they have been formulated in robot-centered ways and lack validation in terms of usability and acceptance by actual users. In

contrast, online behavior modification is a user-centric formulation that can leverage the benefits of these approaches to empower users in ways that can be systematically tested and compared.

Online behavior modification occupies a novel place within approaches to combine autonomous execution with human input. Our user study compared the ACORD algorithm to both a library of autonomous RL policies and a version of SA modified for a multi-goal setting where different styles represent different goals. We validate that ACORD can be used to adjust the style of a robots behavior and is perceived favorably by users. Our study shows that ACORD provides high levels of perceived control and expressiveness, as SA does, while being easier to use. There are also key technical and theoretical differences between online behavior modification and SA. In the context of SA, the task-level goal is unknown, and the robot, through an interpretation of the user's control signal, is attempting to infer the goal of the task. In contrast, in online behavior modification, the task-level goal is known, and the purpose is to maximize the user's control over how the robot autonomously completes that task. SA also requires the user to operate directly in the robot's action-space defined for the task, while algorithms such as ACORD build a separate new space for user input. In a larger system, online behavior modification algorithms like ACORD could work *with* SA, for example by using an SA system to infer *where* the user wants to go, and ACORD to give the user control over *how* the robot gets there. This opens up various directions for future research, both studying and comparing different algorithms for online behavior modification, as well as how online behavior modification may fit into or be combined with other paradigms.

Limitations An assumption in this work is that the designers of the system *know which axes of behavior people care about for the task*. This could be resolved by working with users to understand which behavior features they wish to adjust. Future work might also develop a general understanding of the types of features that users most want to adjust for a given task or types of tasks. Another limitation of the study is that we only considered m=2 behavior parameters to adjust. [190] have shown that the diversity-based methods ACORD is partially based on can learn effectively with up to 25 discrete latent variables. However, a large number of latent variables may impede the usability

and interpretability of the system. Thus, more work is needed to understand how users interact with more numerous and abstract features. Similarly, the features used in this work for behavior oversight parameters where "static goals" in the sense that they were behavioral goals, which once reached, the robot would not change its behavior over time. "Dynamic" behavior features would be ones that, given a specific user setting, the robot's behavior could change over time. A feature for a robot's acceleration, as opposed to a target speed, would be an example of a dynamic feature. For ACORD to learn dynamic features, additional augmentations would need to be made to the robot's state to include temporal information such as when a certain behavior oversight parameter was last adjusted.

While ACORD was sufficiently efficient to be deployed on a real robot and be used by real users, the algorithm is relatively sample-inefficient (about 3 hours of fine-tuning after training in simulation). Future work could improve ACORD's efficiency by leveraging other techniques, such as hindsight and Constrained MDPs [12, 11]. Lastly, although online behavior modification entails the robot avoid task failures, that specification may not be sufficient for safety-critical scenarios unless, potentially, combined with safe RL methods [9, 87, 168].

5 Novice-Friendly Teaching

5.1 Introduction

In this chapter, we consider the case where a home robot does not know how to perform a task the user wants it to perform. Up to this point, we have considered a robot that has a policy which can be leveraged by the user in different ways: either through an innovative use that was not previously designed for, as in IODA, or through interfaces that were explicitly designed to give users more control, as in ACORD. Inevitably, however, there will be tasks that the robot does not know how to do, or at the very least, cannot be easily specified by a user through natural language. In those cases, users ought to be empowered to teach the robot those new tasks via their own means, and without the necessity of being a robotics expert. This type of novice user empowerment, ensuring they have tools to teach the robot how and when they want, is also a collaboration between a user and a robot policy, the policy in this case, however, changing as a result of the interaction.

We sought to bridge the gap between the robot behaviors that experts could instill in robots and what users could teach, so we focused on the teaching modality where the gap was largest. A person teaching a robot with binary feedback (e.g. good or bad teaching signals), also known as Interactive Reinforcement Learning (IntRL), has in general been limited to relatively rudimentary robot tasks and has lagged behind both what was learnable from a robot autonomously and behind other teaching paradigms such as learning from demonstration (LfD) and preference learning. While this is intuitive, as binary feedback as a teaching signal provides relatively sparse information content, it is nonetheless important that people who want or need to teach home robots complicated manipulation with binary feedback are not limited in doing so solely because of the feedback modality the choose.

To narrow the gap between what could be taught with IntRL alone and what robots could be taught by experts, in this chapter, we introduce and evaluate an IntRL algorithm Continuous Action-space Interactive Reinforcement learning, or CAIR. CAIR is the first IntRL algorithm that could compete with state-of-the-art Deep RL algorithms in high-dimensional continuous control

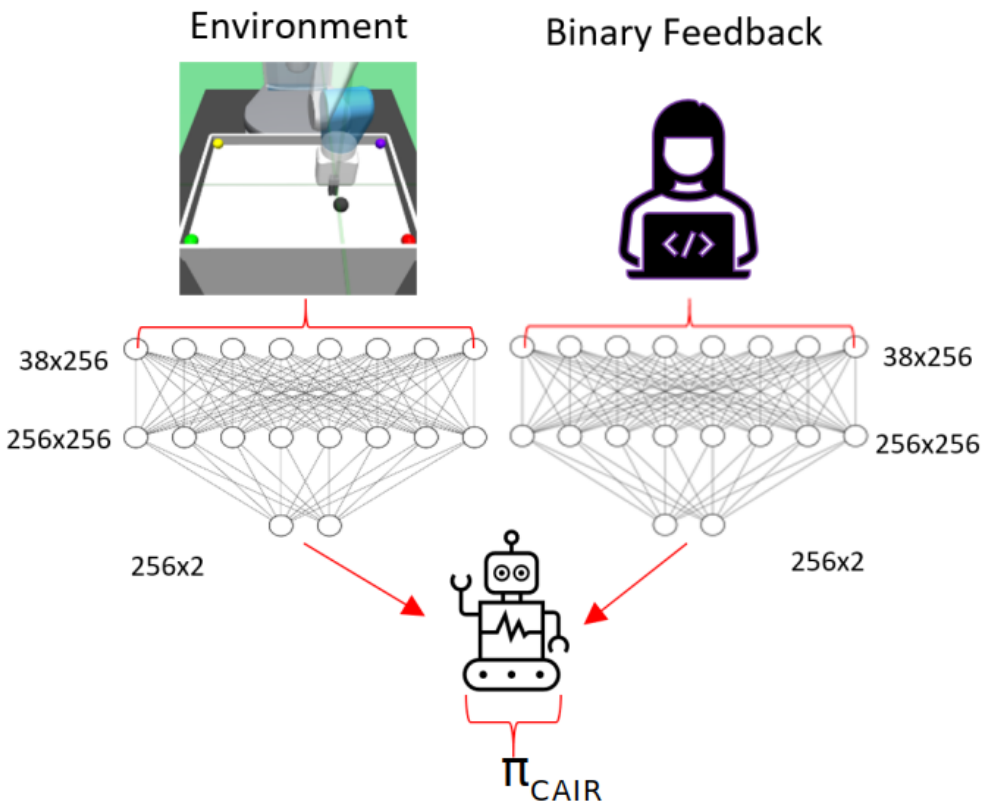


Figure 17: CAIR architecture. The robot uses a combined policy that incorporates both environmental reward and human feedback.

tasks. We evaluated CAIR in simulation with heuristic-based simulated teachers as well as in an online-study with real people. Our results show that CAIR enables people to teach robots tasks with continuous action-spaces as well as or better than both previous IntRL approaches and RL algorithms with multi-robot parallelization. Importantly, CAIR performs particularly well in sparse-reward environments. Since CAIR also uses an environmental reward function, the benefit of performing well in sparse-reward environments is that they are relatively easy for nonexperts to specify. Overall, in this chapter, we introduce an algorithm to further the breadth of tasks people can teach robots through IntRL, further empowering users of robots in cases where their robot needs to be taught something new.

The majority of the work in this chapter was published at IROS 2022 as Sheidlower, Isaac, Allison Moore, and Elaine Short. "Keeping Humans in the Loop: Teaching via Feedback in Continuous Action Space Environments." In *2022 IEEE/RSJ International Conference on Intelligent Robots*

and Systems (IROS), 863–70, 2022. https://doi.org/10.1109/IROS47612.2022.9982282 [237]. Portions were also published at AAMAS 2022 as Sheidlower, Isaac, Elaine Schaertl Short, and Allison Moore. "Environment Guided Interactive Reinforcement Learning: Learning from Binary Feedback in High-Dimensional Robot Task Environments." In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 1726–28. AAMAS '22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2022 [239][‡].

5.2 Continuous Action-space Interactive Reinforcement learning (CAIR) Algorithm

Approach	CAIR	DQN-TAMER	Deep TAMER	Deep COACH
Input dimensions	38	100	100	100
Output dimensions	4	6	4	3
Pre-training?	no	no	yes	yes
Continuous actions?	✓	x	x	x
Continuous state?	✓	x	✓	✓
Validated in simulation	✓	✓	✓	✓
Validated with users	✓	x	✓	x
Validation study size	54	-	18	-

Table 1: Prior IntRL approaches compared to CAIR.

5.2.1 Preliminaries

We model the environment as a Markov decision process (MDP) with states S, actions A, a transition function $T : (S, A) \to S$, and reward function R. A policy π maps states of the environment to actions of the robot; the optimal policy π^* is the one which maximizes the reward, subject to a discount factor γ.

IntRL algorithm which combines the policy generated from Q-learning [279] with a policy generated from binary human feedback. With a teacher whose feedback agrees with environmental

[‡]The second author assisted in testing and provided feedback for the study interface used in the online study. The last author supervised the work and consulted on the experimental design.

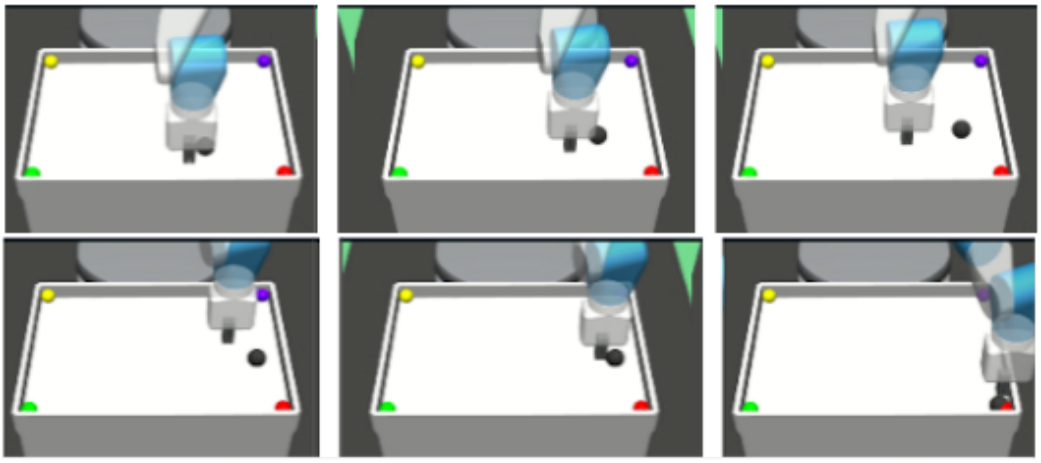


Figure 18: (Viewed left to right) A CAIR push robot shows robust pushing behavior after ∼25 minutes of real-time training.

rewards, PS can significantly speed up early learning by keeping track of previously received feedback for each state-action pair in an MDP [94]. A hyper-parameter C defines the agent's confidence that the human feedback is optimal. The probability that an action, a, in state, s, is taken by the agent is:

$$P(s,a) = \frac{P_Q(s,a)P_F(s,a,C)}{\sum_{a'}^{A} P_Q(s,a')P_F(s,a',C)} \tag{5}$$

where P_F denotes the probability of an action based on human feedback, P_Q denotes the probability of taking an action from the Q-table, and A is the size of the action-space. CAIR borrows two key properties of PS. First, CAIR limits a teacher's overall influence on the agent's policy via confidence parameter κ and may be thought of as the maximum amount of trust the agent puts into the teacher's policy at any given time (this can be thought of as similar to the parameter C in PS). Second, the reward from the environment is *unchanged* by the value of the teacher's feedback. This means that the agent has two internally independent sources of information and can learn from the environment even without a teacher present.

CAIR uses a Soft Actor Critic (SAC) architecture [99] for both a teacher network and an environmental network, where their rewards are made up of binary feedback and the environmental reward respectively. SAC is an off-policy model-free actor-critic based RL algorithm. It works well

in high dimensional state and action space tasks by using a maximum entropy approach.

SAC takes advantage of three networks: a value network which evaluates the current policy; an actor network that improves the current policy and is used to sample actions; and a critic network which minimizes the Bellman Residual:

$$J_Q(\theta) = \mathbf{E}_{(s_t,a_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(s_t,a_t) - \hat{Q}(s_t,a_t)\right)^2\right] \tag{6}$$

θ is a parameterization term, D is the robot's replay buffer, Q_θ is a soft Q-function and $\hat{Q}_\theta$ is the reward at (a_t, s_t) plus the discounted expected value of the value function at the next state. The action network's policy comes from optimizing the objective function:

$$J_\pi(\theta) = \mathbf{E}_{s_t\sim D,\epsilon_t\sim\mathcal{N}}\left[log\pi_\theta(f_\theta(\epsilon_t;s_t)|s_t) - Q_\theta(s_t, f_\theta(\epsilon_t,s_t))\right] \tag{7}$$

f_θ introduces Gaussian noise into the network and π_θ is the robot's current policy. SAC uses a replay buffer that it samples from at a fixed rate (e.g. once every other time step). The output of the policy network π is a mean, μ, and standard deviation, σ, for each given dimension of action space.

5.2.2 CAIR: Continuous Action-space Interactive Reinforcement learning

The CAIR algorithm is specified in Algorithms 1, 2, and 3. CAIR uses two neural networks: a teacher network, $Teach$, and an environment network, Env. $Teach$ learns via a binary $(+1, -1)$ reward from a human teacher or heuristic model. Env is trained using any stochastic *off-policy* RL algorithm and learns from the environmental reward. Both $Teach$ and Env output a policy π_{Teach} and π_{Env} respectively for any given state, both of which are defined by sampling from a Normal distribution with mean μ and standard deviation σ. To combine π_{Teach} and π_{Env}, we leverage two observations. First, in the types of environment we are interested in, a robot can learn a high level strategy from a teacher more quickly than the precise control necessary to maximize

an environmental reward. Second, if π_{Teach} and π_{Env} are similar, then the robot should rely more on π_{Env} as it is in an area of the state space where low-level control that attempts to maximize environmental reward *also* satisfies the teacher's shaped strategy. CAIR will give more weight to a teacher's policy the further it diverges from the environmental policy. We chose this assuming that the teacher's policy is generally good and that the environment should be relied upon when the policies are similar. We achieve this effect via a weighted average defined by the KL-Divergence between the two policies and then select an action by sampling from the normal distribution defined with:

$$\begin{cases} \mu_{CAIR} = \Delta_\pi * \mu_{teach} + (1 - \Delta_\pi) * \mu_{env} \\ \sigma_{CAIR} = \sigma_{env} \end{cases} \tag{8}$$

and

$$\Delta_\pi = min(max(tanh(KL(\langle \mu_{teach}, \sigma_{env} \rangle, \langle \mu_{env}, \sigma_{env} \rangle)), 0), \kappa) \tag{9}$$

Together, these equations form the function **CAIRAction** used in Algorithm 1. Teachers may be relatively noisy, so we set $\sigma_{teach} = \sigma_{env}$ for numerical stability. We apply the *tanh* function then then an element-wise max function to map the KL-Divergence to a value between 0 and 1. This allows us to limit the influence of the teacher network on the overall policy via a cap term κ, which is between 0 and 1. We use κ=.9 for our heuristic experiments κ = .8, .9 for our human subject study. A κ value of between 0.7-0.9 is likely to work for most teachers. If κ is too large, then the CAIR robot is more likely to get stuck in local minima according to the environmental reward by relying too heavily on π_{Teach}; if κ is too small, potential for early gains in learning my be reduced.

To work with delayed human feedback, CAIR makes use of a *window* based credit assignment paradigm. If an action occurred within a preceding window of the feedback being received, then that action, state, feedback tuple is stored in $Teach$'s replay memory. Window duration may vary depending on the environment. Deep TAMER used a window of 0.2-4.0 seconds. We found 0.1-1.0 to be an effective window when using dense feedback with CAIR (Section V). We do not simulate

Algorithm 3: CAIR

Initialize TeachNet, TeachReplay

Initialize EnvNet, EnvReplay

Initialize ActionQueue, κ

for *Step* t **do**

  $\mu_{teach}, \sigma_{teach} \leftarrow TeachNet.\pi(s_t)$

  $\mu_{env}, \sigma_{env} \leftarrow EnvNet.\pi(s_t)$

  $a_t \leftarrow$ CAIRAction$(\mu_{teach}, \sigma_{teach}, \mu_{env}, \sigma_{env}, \kappa)$

  time_start = ClockTime()

  $s_{t+1} \leftarrow$ env.DoAction(a_t)

  time_end = ClockTime()

  ActionQueue $\leftarrow (s_t, a_t, s_{t+1}, r_t, time\_start, time\_end)$

  do CAIR-Env Update

  do CAIR-Teach Update

feedback delay if a heuristic (automated) teacher is used, and feedback may be given to the last robot action. We keep track of a robot's past actions, states, and the time stamps of those actions via a FIFO *Action Queue*. The Action Queue's max-size may be specified such that only state-action pairs that fall into the credit assignment window if feedback is received are stored.

5.3 Validation in Simulation

5.3.1 Environments

CAIR was deployed in two simulated environments (Figure 19), including a custom version of OpenAI gym's push environment and a standard RL testing environment, BipedalWalker (BW). The custom environment, Robot Push Multi (RPM), is a multi-goal environment wherein the task of the robot is to push the ball, which spawns in random positions near the center of the table, to

Algorithm 4: CAIR-Env Update

(Off-Policy SAC)

```
1 EnvReplay ← (s_t, a_t, s_{t+1}, r_t)
2 EnvNet.SAC_Gradient_Updates()
```

Algorithm 5: CAIR-Teach Update

```
1 if feedback then
2     time_feedback = ClockTime()
3     for action in ActionQueue do
4         if w(time_start, time_end,
5         time_feedback) ≠ 0 then
6             TeachReplay ← (s_t, a_t, s_{t+1}, feedback)
7 TeachNet.SAC_Gradient_Updates()
```

any one of the four goals in the corners. This is a *sparse* reward environment: the robot receives an environmental reward of -1 each time step the ball is not at one of the goals and a reward of 0 when the ball reaches the goal and while the ball remains at one of the goals. RPM was designed as an environment with clear, easy to understand goals and strategies that a human teacher may identify. For instance, a human teacher may prefer one color goal over another or simply the goal the ball is closest to. The observation space of the environment is a vector of size 38 made up of joint positions, the relative position of the end-effector to the ball, and the distance of the ball to each goal. The action space is two-dimensional, and represents the desired end effector position, relative to its current position, in the X-Y plane.

BW, is a *dense* reward environment. The reward given to the robot is based on distance traveled, whether the robot has crashed, and a slight negative reward for applying torque to its joints. The state-space is a 23-dimensional vector made up of joint angles and velocities, the hull velocity, whether or not a leg is touching the ground, and a 10 lidar reading range finder attached to the hull.

The action-space is a 4-dimensional vector consisting of the torque applied to each of the robot's hip and knee joints. "Solving" the environment consists of producing a walker that can average a reward of 300 (scores above 0 generally means the robot is constantly moving forward with no crashes). Both environments were chosen due to their complexity and because there are multiple viable strategies a robot may learn.

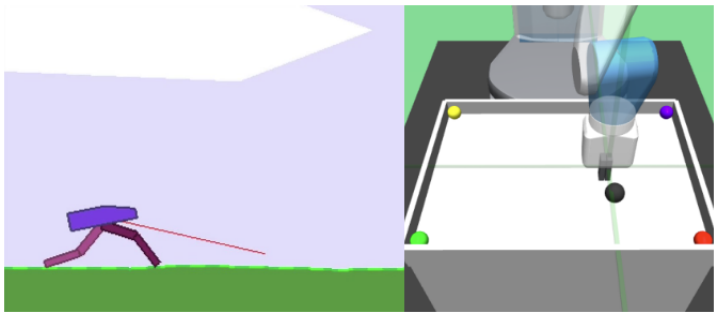


Figure 19: Left: BipedalWalker-v3, Right: Robot Push Multi

Heuristic Oracles For the simulation evaluations, we used *heuristic teachers*. The teachers enact different strategies via feedback that would be consistent across all good policies and reflects *easy to understand* properties of the environment. Compared to using the output of a pre-trained model, these heuristic teachers are a better test of CAIR's ability to represent the teacher model separately from the environment model, and may be a better model of human teaching (a human teacher's task model is unlikely to exactly match that of the RL agent).

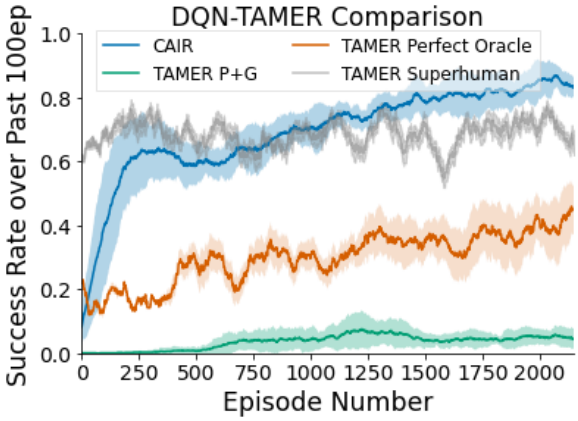
In RPM, we primarily considered heuristics based on the end effector's contact with the ball (as in Vulin et al. [269]). The heuristics tested for RPM were: 1. *push*: positive feedback if the robot has touched the ball or the ball's velocity is >0, negative otherwise; 2. *push+goal*: *push*, with the addition of positive feedback when the ball is at a goal; 3. *goto*: *push*, with the addition of positive feedback if the end effector moves at all closer to the ball. For BW, we observe that any decent policy would entail the walker moving forward and having at least one leg off of the ground most of the time, while recognizing downward hull velocity may precede a crash (negative y-velocity is bad), and would punish crashing. The heuristics tested for RPM were: 1. *general*: positive feedback if

the hull's x-velocity > 0, y-velocity ≥ 0, at least one leg is off the ground, and the robot has not crashed, negative otherwise; 2. *seeable*: *general* but the walker must be moving at roughly a human seeable forward velocity (x-velocity ≥ 0.1 and y-velocity $\geq$ -.1); 3. *forward*, positive feedback if the hull's velocity is in the forward direction, negative otherwise.

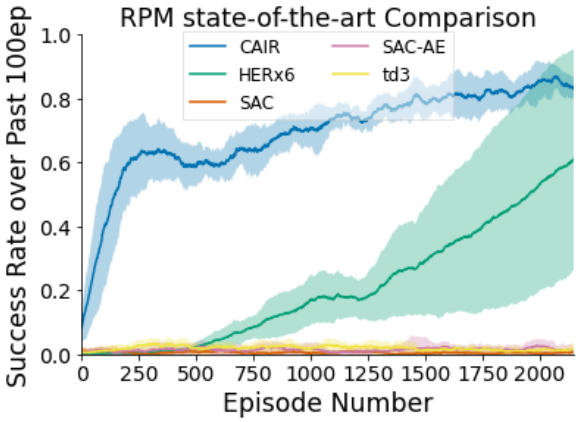
In RPM, *push* and *push+goal* performed much better than *goto*. *Push* and *push+goal* achieve between approximately a 50-75% success rate within the first 250 episodes ($\sim$20-25 minutes of training). *Goto* similarly lead to a large boost in early learning in comparison to traditional RL algorithms. In BW, *seeable*, the heuristic most similar to feedback a human teacher could provide, has both the greatest early learning gains and highest peak performance. *Seeable* achieves a positive reward, within the first 150 episodes ($\sim$45 minutes to 1.5 hours of training). *Forward*, the heuristic most similar to the environmental reward, still improves performance over a SAC robot by itself (Section 4), but is worse than both general and *seeable*. Based on these results, we used *push* and *seeable* for the tests in the following sections.

5.3.2 Results

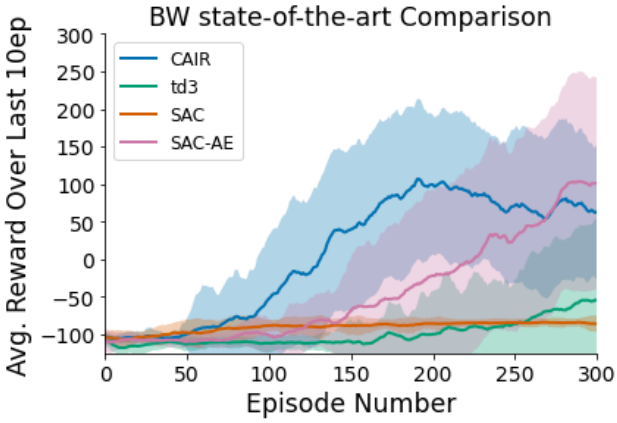
We compared CAIR to multiple state-of-the-art RL algorithms. For RPM, CAIR was compared to DDPG+HER[12], td3[84], and SAC[99]. Since DDPG+HER relies on occasionally replacing the goal state with the robot's final state in its replay memory and there are multiple goals in the RPM environment, we randomly selected which of the goals would be replaced by the robot's final state to avoid bias amongst the four goals. We compared with the IntRL algorithm DQN-TAMER [13]. To do this we discretized the action space into a MultiDiscrete action space [128] (8 directional movement+do nothing), and implemented three versions of TAMER, each with a different source of feedback: 1. *TAMER Perfect Oracle*: trained using a fully-trained model, where every five timesteps, positive feedback is given if the agent's action matches the fully trained model's action, and negative feedback is given otherwise; 2. *TAMER Superhuman*: the same as TAMER Perfect Oracle, but with feedback given every timestep; *TAMER P+G*: Trained using the same *push+goal* heuristic used for



(a)



(b)



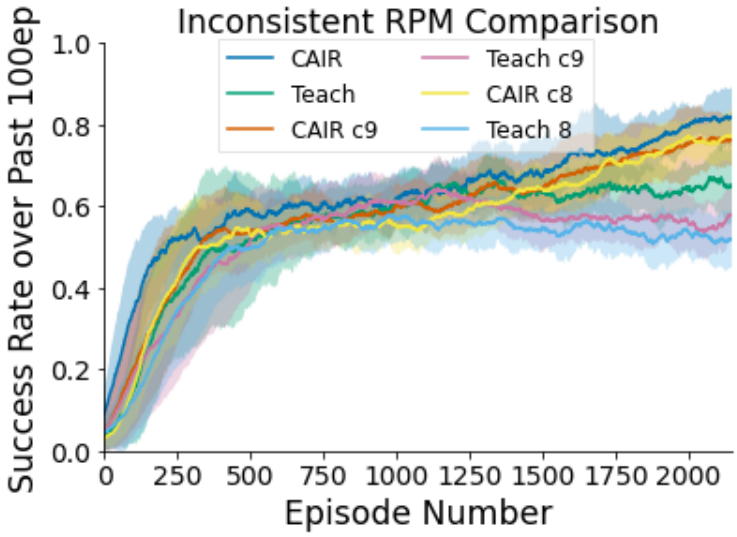
(c)

Figure 20: Comparison between CAIR, Deep TAMER, and state-of-the-art RL algorithms. CAIR. Note that the "Perfect" and "Superhuman" oracles are similar to how TAMER has previously been evaluated, although this type of feedback is known not to be representative of how humans

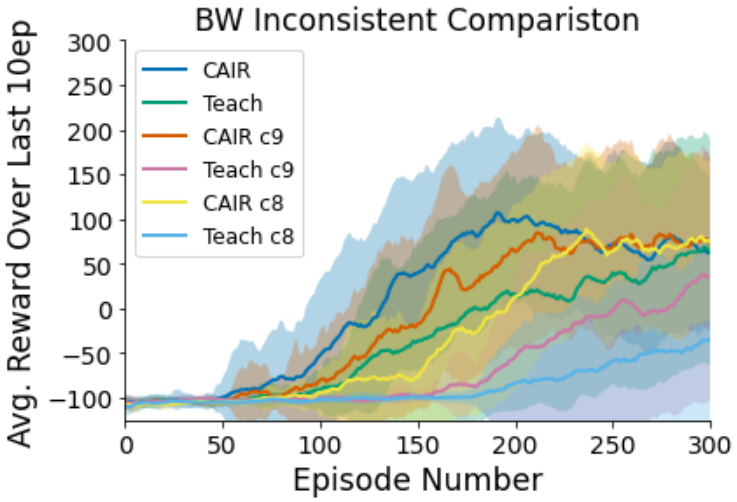
can/do give feedback [19]. For BW, CAIR was compared to SAC[99], SAC-AE[100], and td3[84]. We ran all algorithms for 10 runs of 2250 episodes in RPM and 30 runs of 300 episodes in BW.

Learning Results In both environments, CAIR demonstrates it greatly increases in learning speed early on in training (Figure 20). For the RPM environment, CAIR reaches an ∼60% task success rate in the first 250 episodes (∼ 25 minutes). SAC, the same algorithm as the environment network in the CAIR robot, could not achieve better than random performance. Similarly for SAC-AE and td3. DDPG+HER achieves notable performance with *parallelization* of 6 robots learning in parallel (HERx6). The CAIR robot in BW achieves a decent policy (a positive reward) very quickly in comparison to traditional RL algorithms and is the best performing from episodes 50-250 (∼1-3 hours of real time training). Towards the end of training, SAC-AE eventually outperforms CAIR, but with a significant amount of variance up to that point. CAIR also slightly drops off in performance, likely because the seeable heuristic, since it is not adaptive, will eventually give almost entirely positive feedback. An adaptive teacher, such as a human or "push" in RPM, may avoid this slight drop in performance.

Ablation Study To study both CAIRs robustness to noisy teachers as well as how including the CAIR-Env network stabilizes learning, we conducted an ablation study. In this study we compare CAIR to CAIR-Teach only, or CAIR but only learning from binary feedback. Furthermore, we use the same heuristic oracles as previously discussed, but vary how often optimal feedback according to that heuristic is provided. We refer to this variation as the consistency of the oracle. A consistency of .8 for example means 80% of the time the oracle will provide feedback according to its well-performing heuristic, while the other 20% of the time it will provide the feedback the opposite of the heuristic. The lower the consistency, the worse the teacher is. As shown in Figure 21, the more inconsistent the teacher becomes, the better CAIR performs compared to CAIR-Teach alone. These results support the notion that CAIR-Env leads to better learning outcomes, especially when inconsistent or sub-optimal feedback is provided.



(a)



(b)

Figure 21: Comparing CAIR to a Teach network with no environment component. c refers to how consistent the heuristic is at providing feedback (the absense of a c means 100% consistency).

5.4 Human Subjects Validation

We conducted a human teaching study through Amazon Mechanical Turk (MTurk). The study was approved by the University Institutional Review Board (IRB). Participants were tasked to teach an inverted pendulum robot to balance through a web interface made using Flask [95] connected to a simulated robot environment implemented using Brax [82]. We chose this environment for simulation stability and its easy-to-understand nature.

5.5 Learning Environment

The pendulum environment consists of 5 continuous state-space dimensions and 1 continuous action-space for controlling the pendulum's torque. This is a *sparse* reward environment: the pendulum receives +1 reward for each time step it has not fallen. Each episode lasted 60 seconds or until the robot fell. After the robot fell, it reset to its upright position and waited 1 second before acting again to give participants time to react to the reset. The simulation was also slowed slightly to try and accommodate for widest range of participants ($\sim$4-5 actions per second). Balancing for 60 seconds yields a cumulative reward of $\geq$190.

5.5.1 Procedure

After selecting the HIT on MTurk, participants were linked to a website where they provided informed consent and completed the study. Participants were instructed to teach PendulumBot to balance via "good" and "bad" feedback buttons. They were presented with a short video illustrating the teaching process. No teaching strategies were given other than indicating to the participant that "Good" feedback should be provided when PendulumBot is balancing or trying to balance near its vertical center and "Bad" feedback should be provided when the pendulum appears to be falling on its side. Given the computationally demanding nature of the learning and simulation, variability in peoples internet speeds, and website traffic, a 10-15 minute interaction lasted 1500-3000 time steps. Participants could end the study at any time without consequence leading to a shorter interaction, although as described below, participants who gave feedback for few timesteps were excluded from the analysis.

Participants were randomly assigned to one of three conditions: CAIR κ=.8 (CAIR k8), CAIR κ=.9, and DQN-TAMER. For DQN-TAMER the environment was discretized to have 2 discrete actions (apply + or - a fixed small amount torque). We ensured the pendulum was able to balance given this discretization. Participants in the CAIR conditions used a novel *toggle* feedback approach designed for use with robots that take fast actions. Toggle feedback means the when the "good"

button is pressed, the robot will receive positive feedback until the "bad" button is pressed and vice versa, dramatically increasing the effective density of feedback with no additional effort.

5.6 Results

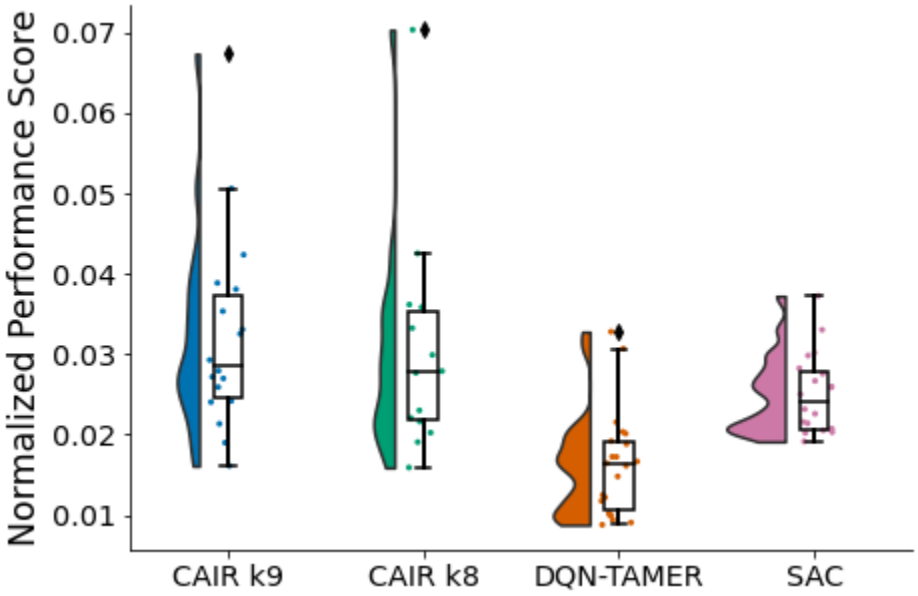


Figure 22: Performance metric distribution across conditions.

95 participants were recruited, limited to MTurk workers with a HIT rating $\geq 99\%$ and at least 50 completed HITs, and compensated \$3 for a 15 minute teaching session. We applied two exclusion criteria to the data: first, we excluded the use of participants data who taught the robot for less than 1500 environment steps due to slow internet speeds or server errors (36 participants). Second, we excluded participants who who averaged fewer than two clicks of a button per episode, since such participants likely either misunderstood or were not engaging with the task (5 participants). The data from the remaining 54 participants was distributed as follows: 14 participants in the CAIR k8 condition, 18 in the CAIR k9 condition, and 22 in the DQN-TAMER condition. We also trained a SAC agent over 20 runs (with no human feedback).

Because the amount of time spent by teachers varied and because longer interactions would be expected to have higher performance due to the increased time for learning, we use a normalized

performance metric for analysis. This metric is defined as the max score a participant achieved across all episodes divided by the total number of time steps they spent teaching the agent. In other words, we normalize how good of a policy was learned by the amount of time the teacher-agent pair had to achieve that performance. Based on this metric, both CAIR algorithms outperform both DQN-TAMER and SAC, with the following performance: CAIR k9: **0.0322±0.0119**; CAIR k8: **0.0304±0.0133**; DQN-TAMER: **0.0162±0.0063**; SAC: **0.025±0.0048**.

We compare the human subject and a (2500 time step) SAC agent via a one-way ANOVA and a follow up Tukey-Kramer test. The performance differed significantly across conditions ($F(70, 3) = 11.197, p \approx 0$). There also significant differences between both CAIR k9 and CAIR k8 with DQN-TAMER: $t(39) = 0.016, p \approx 0$ and $t(35) = 0.014, p \approx 0$ respectively, and a marginally significant difference between the performance of CAIR k9 and SAC ($t(37) = 0.007, p = 0.1$). This analysis demonstrates both the consequences of discretizing a state space and the significant limitations of the prior state-of-the-art of IntRL in fast environments.

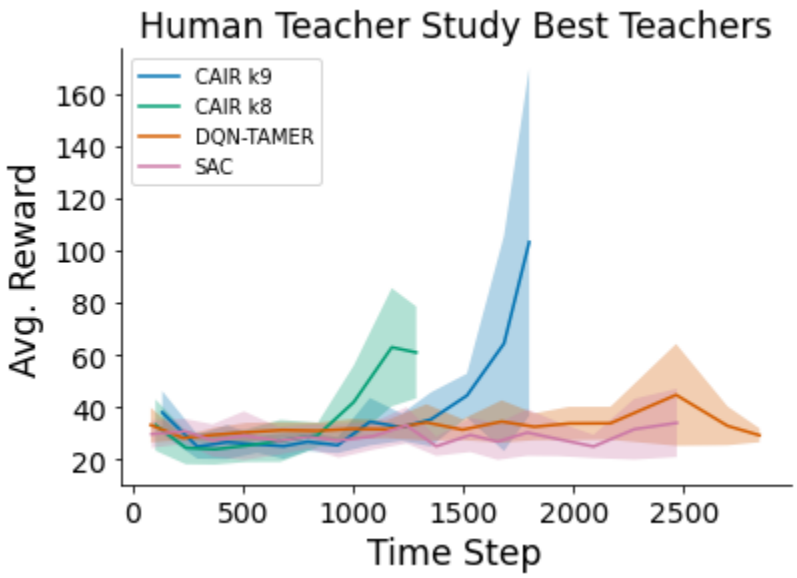


Figure 23: Comparison of the best performers of each condition.

Figure 22 shows the distribution of performance across the conditions. The superior performance of CAIR over DQN-TAMER is clearly seen, with even the lowest-performing teachers with CAIR

having performance close to the median performance with DQN-TAMER. There is more overlap in performance between CAIR and SAC, with SAC having a much lower variance in the scores, but both CAIR algorithms having higher average performance. We also highlight that one CAIR k9 participant was able to fully balance the pendulum within the 15 minute interaction, receiving rewards ≥ 190, something that no run of SAC or DQN-TAMER was able to achieve. Figure 23 illustrates how the algorithms learn over time: it shows the average reward obtained by the three best teachers in each condition over time. To account for the varied interaction length, we averaged over every fifth episode and bounded the data to the teacher with the shortest interaction. Although we focus on best-case performance in this figure, it illustrates the potential of CAIR to leverage good (but still non-expert) teachers and obtain significantly improved *learning* performance over non interactive RL methods.

5.7 Discussion

The primary contribution of this work was introducing a novel binary feedback Interactive RL algorithm that can learn in complicated environments with continuous action spaces. The key innovation of our approach is the incorporation of the environmental reward into our algorithm as a source of stability even when a teacher is noisy. We also introduced various heuristic teachers that reflect human perceptible properties of the robot acting in the environment. Such heuristics capture high-level human intuitions about the task, and can be quickly specified apriori without tuning a reward function. Additionally, we demonstrated the effectiveness of CAIR with human teachers with an online study with MTurk crowdworkers. This is to our knowledge the first evaluation of an IntRL algorithm with unsupervised crowdworkers, a population of users who are highly likely to provide training data to real-world robots as crowdsourcing continues to be a popular source for machine learning training.

A key area of future work is to validate the performance of CAIR on a physical robot. Since SAC

has been shown to work well on real robots [100], it is very likely CAIR will work as well. Furthermore, the early gains CAIR provides would be even more profound on robots deployed in the wild, where failures and the time to a good policy have a higher cost than in simulation. Because CAIR learns from the environment and a teacher independently, there is potential to augment environmental learning with other techniques such as parallelization and sim-to-real, while simultaneously learning from a teacher.

Though the resulting policy of combining the Env network and the Teach network yields a better one than either would when training individually, this does not imply that the Env policy and the Teach policy when taken from a CAIR robot will perform as well as the combined policy. This is true despite having samples generated from the good CAIR policy stored in their respective replay buffers. We hypothesize the main reason for this is *bootstrapping* or *extrapolation error*, which results from an RL robot's replay buffer containing actions that where not generated directly from that robot's policy [85, 142]. Future work is needed to address this problem, for example by extending CAIR into a multi-robot system wherein Env can be trained largely on-policy whilst a teacher critiques the robot.

6 Informing Users

6.1 Introduction

In this chapter, we consider a person interacting with a home robot more broadly, rather than an interaction centered around a single task. Specifically, we seek to empower users across many different tasks, given their home robot has general enough capabilities to attempt and accomplish most tasks. The promise of such a generalist robot is being realized through the development of robot foundation models (RFMs). In this chapter, we argue and present results that support that an effective way of empowering users of a generalist robot, is for them to be informed about that robot's capabilities such that they know how and when to use it's autonomy; or, when a task would be more successful or fulfilling if done via or in conjunction with the previous methods discussed in this dissertation.

In general, RFMs are highly capable and easy-to-use for experts. Novice users should also be able to decide how and when to use them and do so in a confident way. To do this, they need to be informed about the RFMs capabilities and performance for both previously evaluated and unseen novel tasks. An informed user can make better decisions about the degree of supervision the robot needs when executing a task, e.g., if it is necessary to be extra cautious in case of failures or if the user can confidently leave the robot alone to perform the task. An informed user may also be less surprised about the behavior of the robot, will know what tasks it can and cannot do, and will know when the robot may need to be taught or learn more before attempting a task. In this chapter, we investigate how users interpret commonly RFM performance information as well as what kinds of information are important to user understanding and confidence about RFMs.

6.1.1 RFM Preliminaries

A primary goal of RFMs is to allow the user to simply query the robot to do a certain task, and allow the robot to do so autonomously. The development of these RFMs has been enabled by large

transformer-based models [265] such as large language models (LLMs) [302] and vision-language models (VLMs) [299]. LLMs and VLMs have already seen numerous consumer deployments, such as Chat-GPT [188], in which end-users freely interact with and query the model for various purposes. Part of this success is due to the relatively low-cost to the user to experiment with and easily change the output of the model through techniques such as prompt engineering: when LLMs or VLMs fail to produce a desired output, we trivially try again. However, unlike LLMs and VLMs, RFMs have a higher cost to failing or not meeting a user's request. This is because robots interact with the physical world, with objects that can break, and with actions that cost time for the robot to safely move around. Thus, when there is a chance that the robot fails, especially on new or unseen tasks, users need to be made aware of the risks associated with the model and its potential failures. We want users to be able to make *informed decisions* about when to simply ask the robot to do a task and walk away, when to closely supervise the robot, or when they would rather teach the robot more before letting it attempt a task. To make informed decisions, users need access to relevant and useful information about the robot and the RFM.

Performance information about RFMs as presented in current RFM research papers is typically based on the task success rate for different tasks or categories of tasks. Task success is usually presented as a binary signal as to whether the robot's actions lead to some success criteria, such as grasping and lifting up a cup for some amount of time. This information is very useful for other researchers and experts in the field as a means of benchmarking and comparing the performance of RFMs on the same task environment. Task success rate is often shown for both tasks seen in training and previously unseen tasks both with varying degrees of difficulty. Some papers also present a discussion of select failure cases and a few hand-selected videos to try and better represent how the model performs on real robot tasks. Although this information is useful to and well understood by experts, it is not yet known whether this information is equally easily interpreted by non-expert users of RFMs. Because the cost of failure can be relatively high, it is crucial that RFMs be deployed with means of communicating their known performance for known tasks and expected performance

on unseen tasks to novice and expert users alike. Furthermore, there is a critical need to understand what types of supplementary information should be made readily available for users. This can inform both RFM data collections and data releases, as well as company deployments of these technologies. In this chapter, we present a study to advance our understanding of the key question: *what types of information need to be provided to users when they request an RFM to perform a task?*

To investigate how users interpret and use RFM performance information, we conducted an online user study and in-person follow-up. Our study focuses on task success rates and failure cases, divided into two categories: *estimates* of the robot's performance on a user-requested task and *real data* from similar tasks previously evaluated. We make this distinction because it is crucial to the unique scenario RFMs afford: where a user is requesting a novel or previously not attempted task. In this case, there are no previously collected evaluation data about the task the user is requesting, nor does the user have prior experience watching the robot perform that task. Thus, estimates provide information about the requested task, while real data on similar tasks provide empirical evidence about the RFM's performance. In our online study (n=112), participants assessed robot performance using real evaluation data and tasks from previously published works. We found that users prefer to have more information than less when assessing RFM performance, that novice users can reason over and strategically use information commonly reported in RFM research when making decisions about how to use a robot, and that there are information types that users want that are not currently well represented in the research literature. We then corroborated these results in an in-person study, n=14, with an embodied interaction with an autonomous robot.

This chapter demonstrates three key findings: 1. We verify that non-experts understand and are informed by TSR, a jargony metric otherwise aimed at experts. This is shown with correlations of TSR with user trust and confidence with the expected linear positive relationship. 2. We demonstrate that failure cases, though very underrepresented in research literature, are also critical to and valued by users when they are being informed about an RFM's performance; highlighting the need of reporting failure cases for every task to become standardized. This is shown through the responses

users provided when asked about the importance of failure cases. 3. We highlight the necessity of providing users both with real evaluation data of similar tasks to the ones users request, suggesting the need for RFMs to be deployed with access to a large history of evaluations, and provide a means for the RFM to accurately estimate its performance on a task it has not yet been evaluated for. In other words, RFMs must be developed such that their behavior on unseen tasks is predictable or comes with performance guarantees.

The majority of this work is to be submitted for review at NeurIPS 2025 as: Sheidlower, Isaac, Jindan Huang, James Staley, Bingyu Wu, Qicong Chen, Reuben M Aronson, Elaine Short. "How Users Understand Robot Foundation Models Through Performance Information."[§]

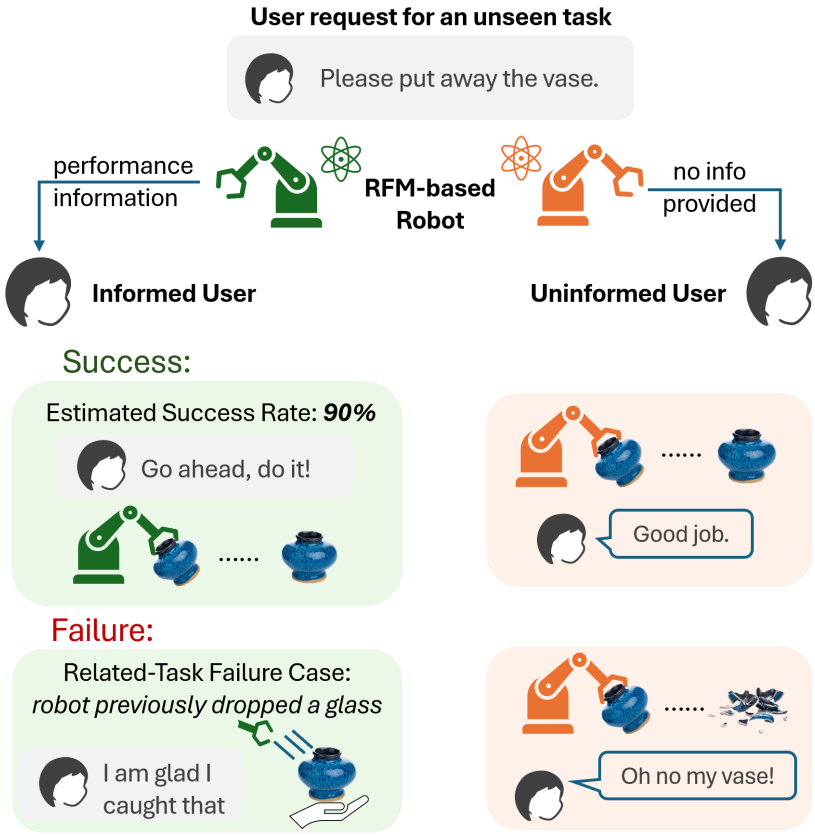


Figure 24: Informed users of robot foundation models will both know what to expect from task-execution and make better decisions about when and how to use a RFM-based robot.

[§]The second author assisted in programming the study website and consulted on its design. The third author implemented the Diffusion Policy algorithm used in the public-space study. The fourth and fifth authors helped organize data from the literature review on RFM evaluations. The second to last author consulted on the study design. The last author supervised the work and consulted on the experimental design.

6.2 Related Work

In the field of human-robot interaction (HRI), it is well studied and understood that users who are informed about a robot and its capabilities improve their ability to collaborate with, teach, and appropriately trust a robot [250, 196, 22, 255, 115, 176]. With RFMs in particular, given their ability to generalize across a wide range of tasks, understanding the model is crucial so that one knows both how and when they can use the RFM for a novel task and so that they are not surprised when the robot fails. Failure can significantly impact a user's trust in a robot and how it impacts that person can vary depending on both the nature of the task and the user [64, 218, 107]. Furthermore, specific types of robot failures and their severity can alter how a user interacts with a robot, by influencing their chosen method of teaching it for example [112]. Thus, it is necessary to understand both how novice users interpret metrics like TSR when making judgments about a robot performing a task but also how useful failure information can be to users in the case of querying an RFM. By understanding this, we can deploy RFMs with additional information and functions that make them more usable and accessible.

6.3 Methodology, Studying and Analyzing User Experience with Different Information Types

The primary goal of this work is to study and better understand both how users interpret information about a RFMs performance and what sorts of information might they want when making decisions about how to use a RFM-based robot. We are also interested in how users interpret the types of performance information typically reported in research as to investigate whether there is a gap between the information reported and user wants and needs. To study these research questions, we designed an online user study in which participants would be presented with real evaluation data from the research literature, evaluate the robot's capabilities, and reflect and brainstorm about what types of information would be useful when using an RFM-based robot. We followed-up this study with an in-person study to corroborate the results.

6.3.1 Types of information

Here we describe the four primary types of information we studied. These information types are drawn from the commonly reported "task success rate" and "failure case" information types found in RFM evaluations.

Estimated Task Success Rate (ETSR) This type of information refers to a robot's internal estimate of how likely it is to perform a requested task successfully. Such functionality is likely crucial for robots to be deployed safely. Estimating a robot's ability to perform a given task is an ongoing area of research.

Estimated Failure Case (EFC) This type of information refers to a robot's internal estimate of how it may fail when performing a task. Similarly to ETSR, this type of functionality has both safety and usability implications. In this work, we focus on EFC as a verbal description in natural language, as opposed to a failure example shown in a simulator for instance. Furthermore, we assume and treat this estimate as the most likely failure case. We note that this may not be the only useful failure case; a "worst case" failure scenario, for example, may also be important to users. We choose to present EFC to participants in the framing as the "most likely failure case," as such cases are often discussed in research.

Related Task - Task Success Rate (RT-TSR) This type of information refers to previously collected task data from an RFM and its resulting TSR. Specifically, this is TSR data from a task that is "similar" to the user's requested task. Similarity can mean different things and be measured in different ways (see Section III, B for how we measure it during this study). Unlike ETSR, RT-TSR is derived from real roll-outs of the RFM and users may value it differently than a robot's internal estimate. This type of data can also be derived from RFM deployments on different robots and environments. Although the performance of an RFM on one task may not always be indicative of how it performs on another, RT-TSR provides a grounded data point for understanding an RFMs capabilities.

Related Task - Failure Case (RT-FC) This type of information refers to real failure cases

Model	Robots Used	Tasks
OpenVLA [134] *	WidowX, Franka Emika Panda	Remove battery from sink; Move salt shaker onto plate; Move eggplant from sink to pot; Move carrot from sink to plate; Stack cups
Baku [102]	Ufactory xArm	Remove can from refrigerator; Close oven door; Lift lid off pan; Lift orange out of bowl; Put coke can in basket; Wipe cutting board; Move tea bottle to refrigerator
Multimodal Diffusion Transformer (MDT) [210]¶	Franka Emika Panda	Move banana from stove to sink; Move banana from sink to stove; Move pot from sink to stove; Push toaster leaver down

Table 2: List of models, robots, and tasks shown to users during the study.

that happened on similar robot tasks. Similar to EFC, we focus on the case of a verbal description of the most likely failure for that task. RT-FC can be useful for gauging how an RFM may be prone fail. For example, in a pick-and-place task, an RFM failing to identify the right object to pick may indicate something different than an RFM failing to grasp the requested object.

6.3.2 Task Data Collection and Coding

To study these four types of information, we collected real RFM data by survey and collaboration with authors of various RFM and multi-task robot learning works. We collected task data that contained: task success rate as collected from an RFM evaluation, including the number of total trials; a video of the robot successfully performing a task; and a video of the robot failing a task if the task success rate was below 100%. We surveyed various state-of-the-art RFM and multi-task robot learning publications and reached out to the authors asked if the required data could be provided if it was not already publicly available. This process resulted in 16 tasks from three evaluations, as shown in Table 1. All of the participant-facing material used in the study is from real robot results and deployment videos in published works. The data used for ETSR and EFC was also this real data, but was framed as estimates to participants.

* We thank the authors of this work for generously providing us with the evaluation data and videos upon our request.

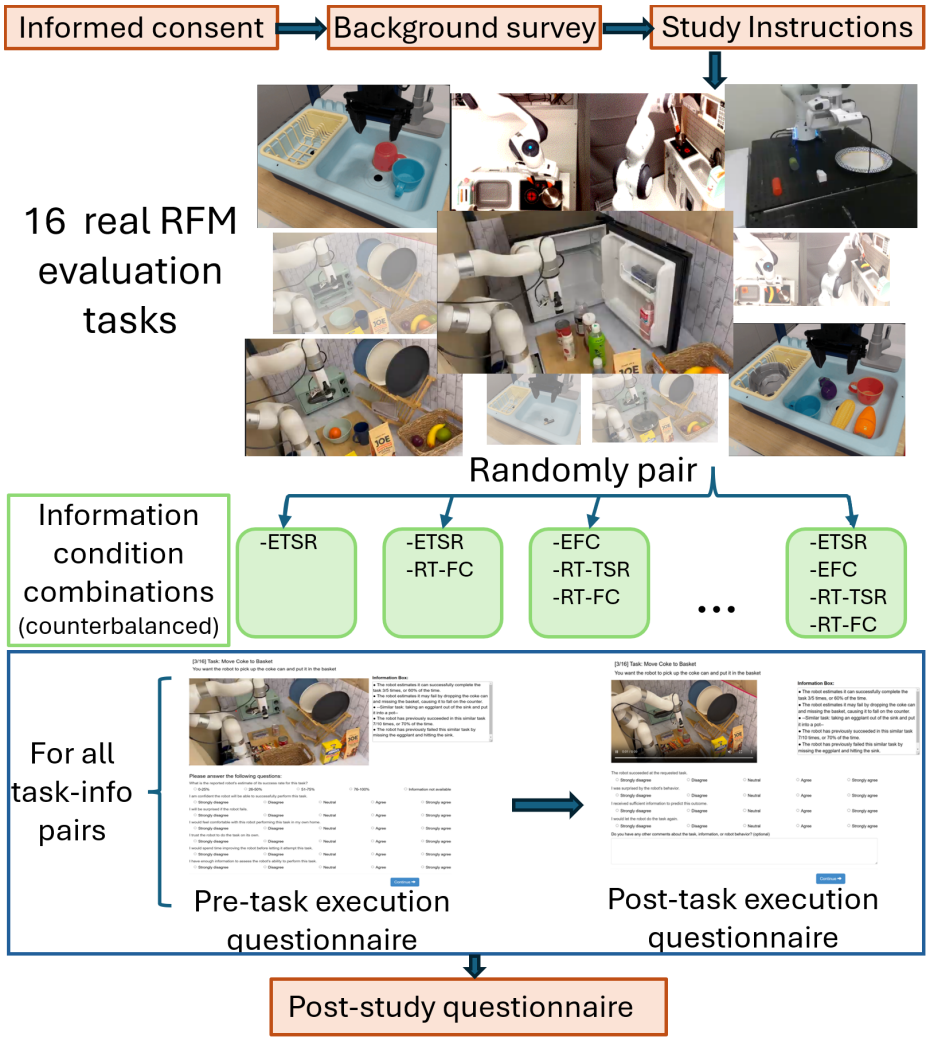


Figure 25: Overview of the study procedure. Users saw a successful or failed trajectory based on a probabilistic sample from the real evaluation success rate for that task.

The study required pairs of similar tasks to present RT-TSR and RT-FC. To get these pairs, we used qualitative data coding. While we tested several methods for encoding task similarity, such as the distance of the task descriptions in language embedding space, and querying LLMs after providing a list of all tasks, qualitative coding ended up being the most consistent. We had three members of our research lab, each of whom was not aware of the content of the study, independently label the most similar task for each of the 16 tasks. They were asked to label the two most similar tasks; where similarity is based on the "robot skills" required to complete each task given a task description and the failure and success videos. When there was a disagreement in the coding as to

Age	Gender	Robot Experience
18-24: 16, *25-34*: 40, *35-44*: 24, *45-54*: 13, *55-64*: 7, *65+*: 1	Female: 54, Male: 46, Non-binary: 1	None: 36, Slight: 38, Moderate: 21, Significant: 6

Table 3: Demographic information from online study

what the most similar task was, it was resolved through discussion. In addition to the similarity coding, we also wrote verbal descriptions of the failure cases in the videos that we had independently checked by two researchers for accuracy and objectivity. In the failure descriptions, we intentionally avoided any speculation as to what may have caused the error or failure, but rather attempted to just describe the physical interaction which took place. All of the failure descriptions and task similarity coding results can be found in the appendix.

6.3.3 Study Design

The within-subjects study had users report on their perception of a robot's capability to perform a task given different amounts and types of information (see Figure 25). We generated 16 information-task pairs for each participant to experience. The types of information were presented in all 16 permutations, including the case of no information, and paired randomly with the 16 tasks. As RT-TSR and RT-FC could contain information about a task the user is yet to see, some participants saw information about a task before being presented with that task. To mitigate this, as well as any other potential ordering effects, we used Latin square methodology to counterbalance the order each participant saw the information permutations. At the beginning of the study, participants also experienced one example task and questionnaire, which displayed all four types of information.

6.3.4 Procedure

After participants provided informed consent, they were presented with an instruction page that explained the study. Users were told that they would be making judgments about "whether or not you want a robot to perform a task in your home." Each type of information was explained and they were shown an example task. The example task was for the robot to pour corn from a red

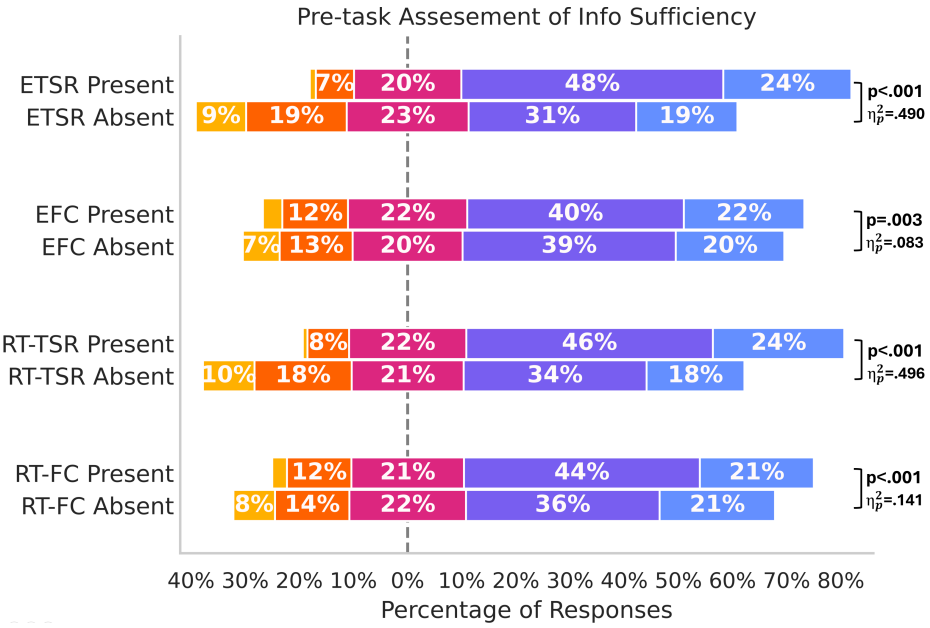


Figure 26: Responses to the pre-task Likert question of information sufficiency under different conditions.

Legend: ■ Strongly Disagree/1, ■ Disagree/2, ■ Neutral/3, ■ Agree/4, ■ Strongly Agree/5

bowl to a pot. For RT-TSR and RT-FC the task was analogized to pouring milk into a bowl. After the instructions, users filled out a background survey that asked for general background information (e.g. age and occupation), their level of robot experience, and to fill out the Personal Level Positive Attitude and Personal Level Negative Attitude sub-scales of the General Attitudes Towards Robots Scale (GAToRS) survey [140].

Participants viewed 16 different robot tasks. For each task, they were presented with a task-request description, such as "you want the robot to move the can of soup to the refrigerator," a still image of the robot about to perform the task, and an information box containing the available supplementary information for the task. Under this, users were asked a series of 5-point Likert questions about their perceptions of the robot, their predictions of its behavior, and if they felt the information provided was sufficient to asses the robot's capabilities in the task. Participants were also given a manipulation check question to report the ETSR from the information box, if present. Following these questions, users went on to a page that had a video of the robot performing the task. Whether a success or failure video was shown was determined by sampling the actual task success

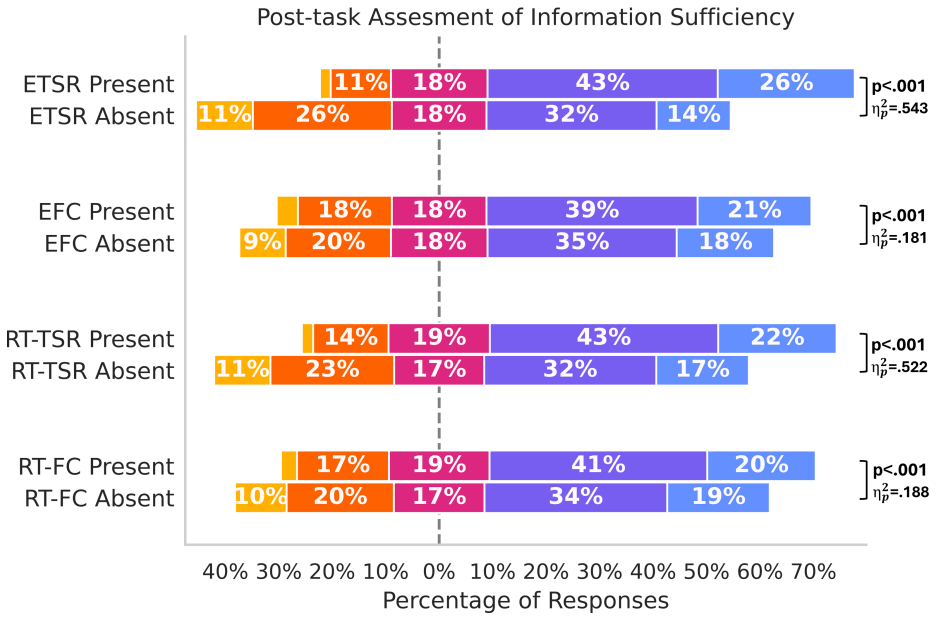


Figure 27: Responses to the post-task Likert question of information sufficiency under different conditions.

rate from the collected data as to realistically reflect user interactions with the model. Under the video, they were asked questions about their level of surprise and asked to reflect on the sufficiency of the information they had been provided. They were also asked for their judgment as to whether the robot was successful in completing the requested task. The videos were altered by being slowed towards the end so users had more time to visualize the consequences of the robot's actions; they could also rewind the video at any time. Before engaging in the 16 tasks, users practiced responding to the questions for a practice task. The task was to put a tennis ball into a tube [282], and the data from this practice was discarded. Lastly, after all 16 tasks they completed a post-study questionnaire.

The full list of questions used in the study can be found in the appendix. The questions address how the different information types may inform a users perceptions of a robot's capabilities and their own willingness to use the robot. They were also developed following recommendations from [225] for developing Likert questionnaires for HRI. A 5-point scale was intentionally chosen due to the length and repetitiveness of the study in an attempt to mitigate user fatigue. Participants were compensated $9.07 for the approximately 34 minute study. This study's procedure was approved by

the University's Institutional Review Board (IRB).

6.3.5 Results

We recruited 112 participants from the online research platform Prolific [204]. Of the 112 participants, 11 were excluded from data analysis either for not completing the study or for consistently failing manipulation checks. Common demographic information of non-excluded participants can be found in Table 2. To perform statistical analysis, we used parametric methods, including repeated measures ANOVA where the presence or absence of an information type is treated as a repeated measure (resulting in 16 levels), and non-parametric methods including the Wilcoxon singed-rank test. We used both Frequentist and Bayesian statistics [66] where applicable.

Quantitative

To investigate the impact of each type of information on user's perceptions and responses, we developed three hypotheses to test.

- **H1:** Users' perception of having sufficient information to assess the robot will improve with each additional type of information, both before and after the robot attempts the task.
- **H2:** Users will report greater comfort with the robot performing the task as more information is available.
- **H3:** Users will find all types of information useful, with ETSR being considered the most useful and equality among others.

To analyze H1, we used responses from two Likert questions that attempted to measure the perceived sufficiency of each information type when being used to asses robot performance, where higher values indicate greater sufficiency. The first question was asked before watching the robot attempt the task: "I have enough information to assess the robot's ability to perform this task." The second question was asked after having seen the robot either succeed or fail at the task: "I received sufficient information to predict this outcome." The distribution of responses along with p-values and effect

sizes can be found in Figures 26 and 27. With an RM-ANOVA, we find that the presence of each type of information significantly increases the users' reported perceived information sufficiency. We find noticeably larger effect sizes in the presence or absence of success rates than in failure cases. These findings support **H1**.

For **H2**, we consider two pre-execution Likert questions: "I would feel comfortable with this robot performing this task in my own home" and "I trust the robot to do the task on its own." Both of these questions relate to the users willingness to let the robot attempt the task. The former question asks about a user's comfort with the robot performing the task in their own home, where, presumably, there is a higher perceived cost of failure or impact to the environment. The latter question asks about the user's trust with regard to degree of supervision. The amount of information present was a weak predictor of comfort for both the "home" and "trust" questions ($R^2 = .008$, $F = 13.107$, $p < .001$, $BF = 35.86$ and $R^2 = .011$, $F = 17.439$, $p < .001$, $BF = 309.22$ respectively). An RM-ANOVA of ETSR and RT-TSR shows that if the robot has both a high ETSR and a high RT-TSR, user comfort is increased ($p < .001$ for all cases). In contrast, neither EFC nor RT-FC increased reported user comfort for either question. Upon closer examination, we also intuitively find the higher success rate, the higher reported comfort for both the "home" and "trust" questions ($R^2 = .26$, $F = 32.460$, $p < .001$, $BF > 1000$ and $R^2 = .26$, $F = 32.434$, $p < .001$, $BF > 1000$ respectively). These results partially support **H2**, with success rate leading to greater reported comfort.

Subjective post-study user assessments of the utility of each information type are given in Figure 28. For each information type, a large majority agreed or strongly agreed that the information type was useful (ETSR: 79%, EFC: 63%, RT-TSR: 84%, RT-FC: 71%). A Wilcoxon signed-rank test showed that ETSR was reported as more useful than EFC ($p < .001, BF = 331.52$) and that RT-TSR was reported as more useful than RT-FC ($p = .004, BF = 12.46$). These results support the first half of **H3**, that users find all types of information useful when assessing robot performance. However, they do not support the latter part, as ETSR was not significantly more useful than all

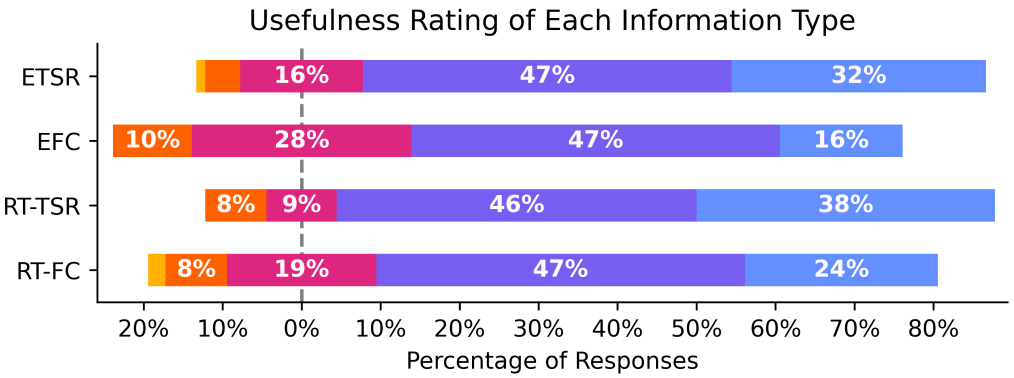


Figure 28: Users from the online study generally reported each type of information as useful when making decisions about when to use a robot.

other types.

Qualitative

To better understand the nuances of how users interpreted the different information types as well as what other information types they may want, we asked two open-response questions in the post-study questionnaire. The first question asked users about their strategy using the information throughout the study: "How did you use each type of information to make your decisions? What made certain information more useful than other information?" The second question asked users to brainstorm what other information types they may find useful. "What other types of information may be useful to you to decide when a robot can or cannot reliably perform a task?" The purpose of this question was to investigate both what users want as well as to identify potential gaps in user-wants and what is readily available or reported in RFM research.

To analyze the responses to these questions, we employed thematic inductive coding [215, 264]. For each question, a researcher came up with themes, codes within those themes, and a codebook with descriptions for each code. Two researchers then independently used that codebook to label each response with up to five codes with spreadsheet annotation. Any disagreement about the codes or about codebook revisions was resolved through discussion. The final codebooks and counts for each code can be found in the appendix.

Information Usage From the responses to the information usage question, three themes emerged:

preference, trust, and strategy. Within those themes we developed 14 codes in total. We found a wide variety in user preferences for types of information, with a divide between those who preferred real data and those who preferred estimates, although many participants relied on both types. 31 participant responses were tagged with the "preferred real data" code and 30 were tagged with the "preferred estimates" code. Participants were also divided on whether they trusted estimates (18 participants explicitly reported to "trust estimates" while 11 participants were "skeptical of estimates"). Those who were skeptical of estimates typically also indicated that they preferred real data to make their decisions. **P. 45**, who preferred real data, said it provides "tangible evidence of its reliability." Another, **P. 80**, said real-data is "crucial because it provides real-world evidence of the robot's capabilities and potential failures." In contrast, users who preferred estimates both mentioned trusting them and appreciated that "it was a direct factor on whether or not the robot would accomplish the task." While some users relied only on estimates or real-data, many had strategies for using both.

Users who tried to make use of all the available information often had specific strategies. 20 participants were tagged with the "ordered preference" code: they explicitly mentioned a ranking or process for using some information to support another. **P. 83** said "I used success estimates to gauge the robot's reliability and past performance to assess trustworthiness." Similarly, 5 of these participants mentioned an ETSR threshold for deciding how to use information or trust the robot. For example, **P. 8** mentioned an ETSR "below 70% felt iffy [unreliable]" and RT-TSR " helped me with my confidence in the absence of a success rate for the main task itself." Many of the participants (19) mentioned their ability to use the real-data in context depended on how similar the information was. **P. 48** for instance explained their usage of real-data "would depend on how similar it was to the actual task; if it was not that similar then I would disregard the information and just use my gut." Overall, the responses to how information was used highlights the diversity among user strategies and preferences. They also highlight the importance of being able to quantify task similarity in the context of RFMs as an indicator of potential task performance on a new task.

Participant Suggestions for Additional Information Types Users were asked what other information would be useful when determining the reliability of a robot to perform a task. From their responses, we derived seven themes: Robustness, Robot Capability, Learning, Failure, Task, and Miscellaneous. Within those themes we developed 18 codes in total. The most common request was that of wanting more of the information from the information types already available, such as more estimated failure cases or more information about related tasks (38 participants). This result is consistent with the quantitative analysis of the presence of each information type being useful. However, users also expressed diverse and rich desires about what other information they may want.

"Robot capability" was the second most common type of information requested by participants. Participants expressed a desire to know about the robot's physical specifications, such as the "robot's strength and dexterity" and/or its sensing capabilities, such as "color identification and motion detection." Similarly, 8 participants mentioned that they wanted to know about "speed" or how fast the robot could perform the task. Despite EFC and RT-FC being rated as less useful than their TSR counter part, many users identified other aspects of failure as important (the code "failure rate and case" was tagged 23 times, the code "failure degree" tagged 7, and the code "failure recovery" tagged 5 times). "Environment factors" and comments about robustness were also frequently mentioned. **P. 43** provided a detailed response that captured well the "environment factors," "robustness to environment," and "robustness to task" codes:

"It would be helpful to know more about the robot's past experiences with similar tasks in different environments or settings, as well as its ability to adapt to new challenges. Also, understanding how the robot handles unexpected obstacles or changes in the task would give a better idea of its reliability."

Details concerning the robot's algorithm and performance were also present, albeit less focused and with less consensus than topics such as robot capability. Intuitively, users either wanted real success rate or failure cases on the requested task (13 participants), or wanted to watch the robot repeatedly attempt the task (8 participants). 31 participants wanted to know more about the nature of the task itself or "thought about how easy or difficult it would be for the robot to perform the

task." Some participants wanted information about the robot's ability to learn. **P. 65** for example said "understanding how the robot learns from past mistakes would help me trust it more."

6.3.6 Follow-up: Verifying and Exploring User Experience Offline

In this follow-up study, we sought to explore users' information preferences and needs when they are exposed to a real robot acting with an autonomous policy. Watching a robot perform a task in real-time offers the benefit of seeing how the robot's failures or successes affect the surrounding environment, and may change a users' assessment of the robot from embodiment alone [270, 229]. This follow-up study seeks to identify the effect of embodiment: checking for differences in the types of information users perceived as valuable in-person compared to when evaluating videos of a robot, and exploring what new types of information they may want based on their experience with a physical robot.

Study Design and Procedure

We conducted a public space study in which we deployed a Kinova Gen3 Lite robot [136] in the lobby of a large University building. To maintain similarity with the tasks from the online study and the task that users would be watching in this study, we designed a kitchen chore-inspired task in which the robot had to place cans of soup on a shelf (see Figure 29). To enable the robot to complete the task and demonstrate to users a fully autonomous system, we trained the robot to perform the task with Diffusion Policy (DP) [48] using an implementation from LeRobot [43] and further adapted to ROS [249]. While DP is not a RFM in itself, it is nevertheless a start-of-the-art imitation learning algorithm often used as a baseline for RFMs and was feasible to train and run inference locally (notably, current work is trending towards making RFMs more affordable to use [282, 26]). The policy was able to complete the task most of the time but would also fail regularly, typically by failing to grasp the can (hovering above it or pushing its gripper into it from above), pushing the can into the shelf itself instead of placing it atop the shelf, or dropping the can onto the

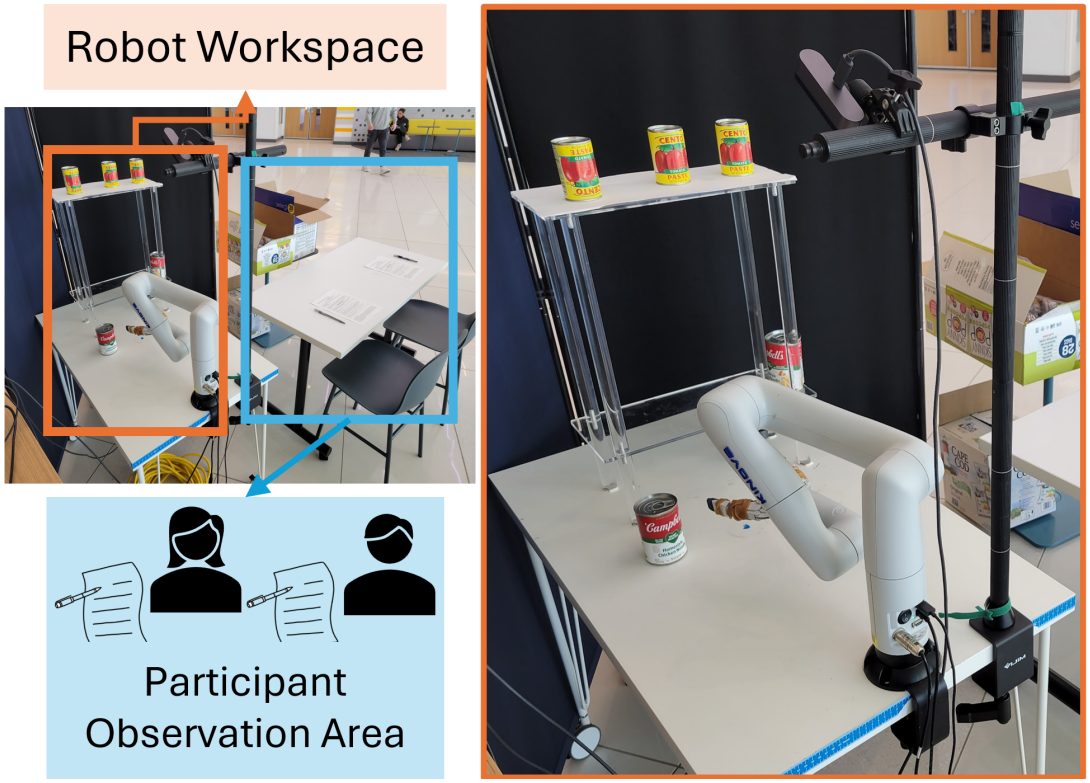


Figure 29: In this public space study, deployed in a University building lobby, participants watched as the robot autonomously and repeatedly attempted to put away the can on the shelf.

table due to a poor grasp. We deployed the setup in the lobby of the building and had the robot repeatedly and continuously attempt the task. A researcher manually reset the environment after each run. In this way, the robot served as an example to help users in their response to questions.

Participants were recruited from passersby, either by posters near the study location or by word of mouth. If users provided their consent to participate, they were given a one page questionnaire which described the task that the robot was doing, and provided an information box (similar to study 1) that explained each type of information (ETSR, EFC, RT-TSR, and RT-FC), and provided examples of each type for the robot's can-clean-up task. In the instructions and task description, users were asked to think about how the provided information may be helpful or not when deciding whether a robot is capable enough to perform tasks in a home.

As in the online study, they were asked a Likert question for indicating how useful each type of information is for determining whether one would want to use the robot. They where also asked

two open ended questions: "What other types of information may be useful to you to decide when a robot can or cannot reliably perform a task?" (also in the online study); and "How would you decide whether or not the robot was good enough at a task to want [to use] one?" After completing the questionnaire, they were thanked and offered a variety of snacks as compensation. The study took around five to ten minutes to complete. This study's procedure was approved by the University's Institutional Review Board (IRB).

Results

In total, 14 people participated in the study. The Likert response data can be found in Figure 30. Like in the online study, a large majority of users reported all types of information as being useful. Unlike in the online study, we did not find any significant differences between information types. To analyze the responses to the open-ended question about other information types, we used the same codebook developed from the online study.

In-person users had similar preferences to online users about what robot information they would find useful in evaluating the robot. While only 2 of the 14 participants wanted more information, "robot capability" was the most frequently tagged code. Multiple participants, likely due to being physically co-present with robot, brought up things like space utilization and workspace, which were not brought up in the online study. **P. A12** for example said "At what speed can tasks be performed? (e.g. tx/min) Cost of automation? space utilization, safety, types of products." And **P. A10** similarly wrote "How much installation the robot would require or how much space it takes up. Also how long it takes to perform the task." Descriptions of the failure costs and concerns were also more personal or pronounced. **P. A4** wanted to know if "In the case of failure will it damage anything or anyone," and **P. A5** expressed concern "if it drops it [the soup can] on my ceramic tile floor and doing damage." Responses to the second open response question generally reported wanting a highly capable and safe robot, with 3 participants specifying the robot should be roughly as capable as a human at the task. These results both corroborate many of the findings from the online study, and provide insights into the information needs of users who are physically co-present

with the robot.

6.4 Discussion

In this work, we examined how people interpret and use performance information when making judgments about RFM-based robots. Specifically, we focused on the context when a user requests a novel task that the robot may not be able to guarantee success on. Being aware of when the robot is likely to succeed or fail is crucial for empowering users, ensuring they can safely and effectively use RFM-based robots for various purposes. Furthermore, informing users in this way will become an increasingly important problem as RFMs are deployed and operate around people where the cost of failure is potentially high. Towards better understanding how to inform users, this study resulted in two *key findings*: 1. users are capable of reasoning about commonly reported performance information and have diverse preferences and strategies for interpreting and judging this information; 2. users not only want information about success rates and failure cases, but nuanced and detailed versions of both as well as additional information types altogether.

6.4.1 Implications for RFM Research

Finding (1) is reassuring as it means there is not a large semantic gap between what is reported in research and how users may understand that information. It also encourages further research into how to estimate RFM performance on novel tasks to accurately report ETSR and EFC; and into how task similarity in the context of an RFM ought to be measured so users can easily inquire about relevant real data. Finding (2) indicates a gap between the types information reported in research and the requisite information for a holistic understanding of RFM performance. While there is work towards creating more comprehensive benchmarks, such as [276], new RFM research could also report on metrics users mentioned that are easily collectible during an evaluation, such as the speed of task execution or how often the robot autonomously recovered from a failure. The development of other metrics and information, such as environment robustness, or how the robot's

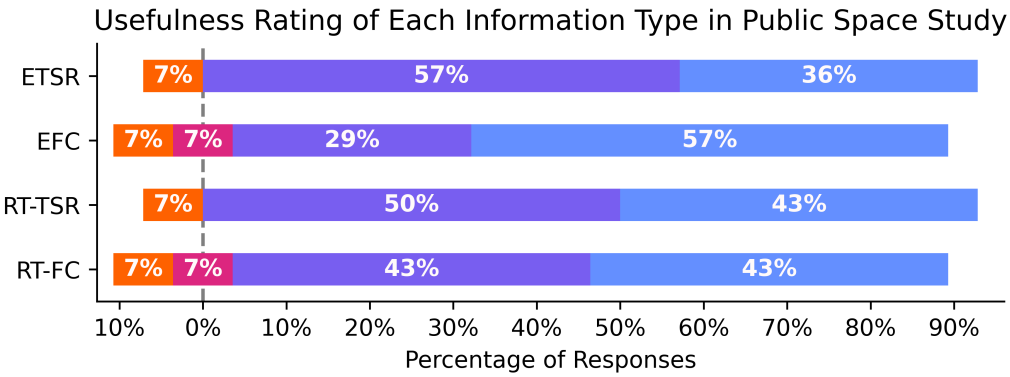


Figure 30: Users from the follow-up study overwhelmingly agreed that each type of information was useful when making decisions about when to use a robot.

capabilities may affect the RFM task execution, are also important but may be more involved or difficult to quantify. These findings suggest promising pathways for the future of RFM evaluation reporting.

6.4.2 Implications for HRI research

This work opens up new research avenues for the HRI community about how users engage with RFMs. RFMs offer a unique promise in which end-users can make open-ended task queries to their robot. This presents a fundamental shift in how people may be interacting with a robot. In particular, successful communication and interaction with an end-user is potentially distinct from robots that have pre-specified tasks, tasks where the human and robot are mutually collaborating, or whose capabilities are assumed to be well-known by the user. Thus, it is crucial that we understand potential gaps in how RFM evaluation information is reported in research and how to best communicate to users to inform them about an RFM's capabilities. This work demonstrated how users interact with information types; future research should explore ways to communicate this information effectively.

Both findings (1) and (2) provide insights into how users interpret RFM information and may be valuable, for example, in developing user interfaces. **P. 7** for instance said that "I felt more confidently when it showed an estimated task success rate higher than 70%, and less when it was

multiple estimations of real task success/failure rates, It was more useful when it didn't mix multiple success/failure rates." This is potentially an issue with the presentation and framing of the information, rather than the content of the information itself. Future work could investigate using LLMs for communicating performance information in a natural and informative way. Since users employed different decision-making strategies, it is likely they formed a mental model of the robot's capabilities based on the performance information. However, this study did not specifically evaluate the accuracy of these mental models. Research in HCI and ML suggests that users' mental models may often be inaccurate or misaligned with the robot's actual performance [197]. A study of how task-type affects the interpretation of different information types is also a valuable direction for future work.

6.4.3 Limitations

There was a trade-off we made between our online study and the in-person follow up. The online study allowed for users to be shown a greater variety of tasks, robots, and real evaluation data. The in-person study allowed for users to be co-present with an autonomous robot, but lacked the task variety and the large number of participants of the online study. The diversity of tasks itself presents a limitation, as nearly all tasks involved a robot arm and were inspired by kitchen chores. This is mainly because robot-arm kitchen settings are a common real-world adjacent evaluation environment for RFMs, yet it nonetheless is a generalizability limitation.

There are also some limitations to the interpretation of the results of the online study as a result of certain design decisions. Because we chose to use real task success rates, and show users a success or failure probabilistically, we did not control for performance or task success. We attempted to mitigate this by having a wide variety of tasks with different success rates and a large number of participants. However, it is likely an important factor in participant decision making that should be studied closer in future work. There was also evidence of a learning-effect: some participants reported using the RT-TSR and RT-FC for information from previous tasks to know the ETSR and

EFC of that task later in the study. Finally, this work did not examine an experienced user of a robot arm interacting with a deployed RFM. While this will need to be addressed in future work as RFMs become more common, this study provides a basis for examining how such users interpret performance data.

7 User-Empowerment with Robot Foundation models, Design and Algorithm Considerations

7.1 Introduction

While the functionality of RFMs can serve to empower users, especially informed users, how they are designed can further impact how successful an interaction or how empowering they are. For example, RFMs are typically large in size and require significant amounts of compute. If the RFM's generalist policy inference speed is relatively slow as a result of that size, it may limit the tasks people can use it for, especially when those tasks require precise timings during a collaboration. Thus, the design of an RFM, in this case ensuring its inference speed is relatively quick and costs relatively low, can either empower or dis-empower a user.

In this chapter, we challenge whether a "generalist policy," or a single large model that takes in observation inputs and outputs robot actions, is the best or only approach to RFM design from an HRI perspective. We introduce alternatives to generalist policies that RFMs can also be *policy generators*, or large models that, when queried, generate a small, task-specific, policy which can then be deployed solely for that task. This approach has the benefit of being modular, such that if one wants to update an RFM in general, it will not affect the behavior of the task-specific policy. This potentially has the benefit of the user more greatly familiarizing themselves with that policy to use it in conjunction with IODA, for example. Similarly, the user is still informed about that policy and how it performs despite changes in the RFM itself. We introduce a proof-of-concept algorithm Diffusion for Policy Parameters (DPP) to show that this is a viable user-centered approach to RFMs.

The majority of the work in this chapter was published at the TAFM Workshop at RLC 2024 as Sheidlower, Isaac S., Reuben M. Aronson, and Elaine Short. "Towards Interpretable Foundation Models of Robot Behavior: A Task Specific Policy Generation Approach," 2024. `https://openreview.net/forum?id=umkqPTUDED` [240][‖] .

[‖]The second author consulted on the framing and writing of the work. The last author supervised the work and

7.2 Preliminaries

In this chapter we present DPP, an algorithm that can be used for RFMs which generate stand-alone task policies. The resulting neural-network from DPP can be thought of as a *hypernetwork*. Hypernetworks in the neural-network literature refer to neural-networks which either generate or modify the weights and parameters of other neural networks [98]. Hypernetworks have been used to augment neural-network performance on various AI-related problems such as classification [192], natural language processing (NLP) tasks [163], and, notably, a variety of computer vision tasks [214, 7, 70]. Recently, hypernetworks have also been applied to RL. Specifically, in Meta-RL tasks, or the task of learning multiple tasks to be able to quickly generalize to a new similar RL environment [23], because of their ability to capture high-level relationships between the different neural-networks that are used to train an RL agent [220, 24, 113]. However, despite the potential shown of hypernetworks to be applied to various areas of AI research, only very recently have researchers began applying them to robotic control tasks [105, 21, 297].

7.3 Potential Challenges with Robot Foundation Models as Generalist Policies

Robots should be able to learn from feedback and have real-time behavior personalization for any given task. If the policy the user is interacting with is a generalist robot policy, two problems may limit a user's ability to do this. The first is that when a user teaches the robot a new task or personalizes the behavior for a certain task, the behavior in separate and unrelated tasks may be affected. This may jeopardize the interpretability and legibility of the system [32]. Another is that updates to the base of the model from the organization which developed the model may have downstream affects on specific tasks/robot behavior that may be unexpected or undesired by a user. This is already the case with consumer-available LLMs such as ChatGPT, however, in the case of robotics, the consistency of the robot's behavior is a crucial component to the user's ability to teach

consulted on the framing and writing.

and interact with the robot. In fact, robots spontaneously acting in unexpected ways around users may cause physical safety concerns beyond those posed by systems operating solely on language. Thus, making sure that a robot's task behavior is changed when and how a user wants is crucial.

7.4 Diffusion for Policy Parameters (DPP)

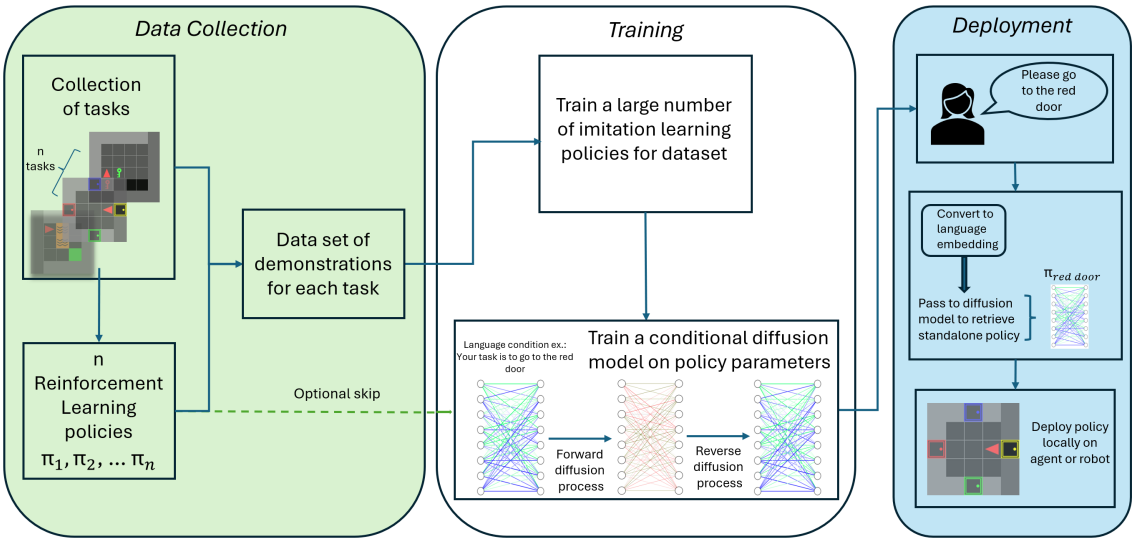


Figure 31: The DPP foundation model of robot behavior design approach

We present Diffusion for Policy Parameters (DPP), a novel approach for learning how to generate standalone policies for individual tasks. DPP alleviates some of the concerns mentioned in the prior section. We then present a proof-of-concept evaluation in a grid-world simulation. This is, to the best of our knowledge, the first generative approach for creating policies in parameter space. While policy search [201, 256, 147, 126] and exploration over policy parameters [78, 177, 259] have been explored, generative AI techniques have not been used directly in parameter space.

The DPP method (Figure 31) learns a conditional diffusion model for generating policies in policy parameter space. The steps for DPP are: collect a dataset of language/goal conditioned tasks and a dataset of demonstrations over those task; train a large set of policies on either the demonstrations or tasks themselves; then train a diffusion model conditioned on the task description and takes the parameters of the policies as input. The result is a model that leads to an interaction similar to a

generalist robot policy: a user asks for or demonstrates a task, and then the robot autonomously executes that task, with the option of further human-in-the-loop fine-tuning if necessary. The key difference being in DPP, a policy independent of the foundation model is generated to execute the task. To study whether DPP is a viable approach for learning to generate policies, we must show it can lead to a model which conditionally generates "good" policies.

7.4.1 Environment and Data Collection

We ran experiments in the Minigrid environment [47] for its suite of language-conditioned tasks on which we can train many agents on in a relatively small amount of time and with limited hardware. All tasks have a similar reward structure: sparse reward with a time-step penalty, resulting in a cumulative reward between 0 and 1. To generate a large number of tasks, we took three language-conditioned tasks and made each goal specification within that task its own task. In particular, we took the environments Fetch, Go to Door, and Go to Object, and for each possible object configuration, made that a task (e.g. Go to Door contains both the "go to red key" and "go to blue box" task specification, and we treat each as a task to train an agent on). We chose the 5x5 versions of each task for computational efficiency and quicker training. We then collected many seeds for each task to ensure random goal positions and obstacles, resulting in 84 unique tasks. While these tasks are significantly simpler than in-the-wild robot tasks, they provide a wide range of separate policies to train on. In the real-robotics case, tasks would range from "make coffee" to "clean the windows" or "water the flowers."

To collect policy data on these tasks, we trained 84 RL agents using PPO [226] to optimality (achieving a mean reward $> .98$). We then trained behavior cloning (BC) agents on trajectories collected from the RL agents until they received a near-optimal average reward of $> .85$. We chose BC as opposed to RL for every agent because it was more timely to train and collect the policies. We trained approximately 1000 agents for each of these tasks, discarding tasks where BC did not achieve high reward given the allotted trajectories. This resulted in BC agents for 64 of the 84 tasks

DPP Model Architecture	
Language Embedding	bge-small-en-v1.5 (size: 384) [289]
Noise Schedule	Cosine, 1000 steps [185]
Noise Type	Gaussian [106]
Model Architecture	Transformer [265], 48 heads, 12 depth, 768 width
Batch Size	128
Input/Output Dimensions	32x82; 32 for MLP policy hidden layer, 82 = 75 (observation size) + 7 (action size)

Table 4: DPP model architecture used in experiments

	Diffusion Sample Policy	Random Policy	Training Parameters (TP) Mean	TP Median	TP Mode	Mixture of Samples (MoS), m=4	(MoS), m=8	(MoS), m=16
Avg. Return	**0.766**±0.16	**0.198**±0.14	**0.189**±0.14	**0.205**±0.12	**0.125**±0.16	**0.816**±0.19	**0.878**±0.15	**0.886**±0.16

Table 5: Results from experiments

and resulted in 74,000 trained policies.

7.4.2 Model Design

Given the dataset of policies, we trained a conditional diffusion model which takes as input a language description of the task and outputs an end-to-end policy network for that task. The model architecture and description can be found in Table 1. The architecture was largely decided on based on trial and error. However, two key decisions were necessary to effectively learn in parameter space. The first was to use an entirely transformer-based architecture, as opposed to, e.g., a U-Net architecture [212, 106]. The other was to use the hybrid loss as proposed in [185]. We also experimented with various loss functions based on evaluation of the generated policies, but they did not lead to high performance.

7.4.3 Evaluation and Results

The evaluation results of the final trained model can be found in Table 2. The evaluation aims to show the model generates meaningful policies in parameter space. For each baseline, we took average performance across all 64 tasks, with 10 runs each on random seeds. "Diffusion Sample Policy" refers to a single sample from the diffusion model conditioned on the task description. We primarily compare to baselines as a means to ensure the model is not learning trivial local minima. If it is not, then we expect a single sampled policy to significantly outperform the baselines. "Random

Policy" refers to an agent that takes a random action in each state. The "Mean," "Median," and "Mode" baselines refer to taking those operations on all of the parameters in the dataset for the specified task. The sample policy significantly outperforms all baselines indicating that the model is learning to generate meaningful and performant polices. A single sample, however, achieves slightly lower returns than the agents in the training set. To achieve a similar performance, we take a simple mixture approach where we sample n policies, and for each observation, take the most common output action. This is referred to as Mixture of Samples (MoS) in Table 2.

7.5 Limitations

Despite promising early results, there are key limitations with the evaluation regarding extrapolating the results to real-world robots. Though diffusion models and transformers have been shown to scale well with large amounts of robot data, we have not shown this scalability with policy parameter space learning. Similarly, we emphasize that the training data needed for DPP is different than for a generalist policies: DPP requires a dataset of trained policies (which could be gathered through simulation or a cross organization effort similar to the Droid dataset [132]), rather than a demonstration dataset. However, for DPP to scale and generalize across tasks, it will likely need both a policy and a demonstration dataset. For this scaling to take place, it will necessitate large amounts of both robot and compute resources. This requirement limits both who can deploy DPP for real robotic tasks, as well as how easily the algorithm can be developed, iterated on, and fine-tuned. Of note, however, this is already true of RFMs in general. We have also only demonstrated results in an environment with a discrete action space. Although some generalist policies, such as [90], have had issues with discrete action spaces, we believe a robot foundation model should be able to handle both discrete and continuous actions. Despite these limitations, DPP presents a promising a direction for the development of user-centered RFMs and similar approaches can be used to further empower end-users with greater control over their robot and its policies.

7.6 Discussion

While generalist robot policies as robot behavior foundation models show clear successes, they do not maintain properties of locality and explainability that would be desired for a deployed system. To limit these concerns, we presented DPP, an alternative which may alleviate some of the outlined concerns. DPP generates smaller, standalone policies for each task; this approach means that those policies are not affected by a user teaching the robot other tasks or by unwanted updates to the general foundation model.

Enabling policies to be stable and therefore more predictable is a key feature for human-usable robots. Human-robot interaction research has consistently shown that robot models need to be not just performant, but also predictable [152]. A predictable robot system not only improves its interpretability, but also allows a user to gain a high degree of familiarity with those policies, and in turn use them to accomplish novel tasks. In this work, we embed this predictability and usability directly into the structure of the model without compromising its flexibility to learn from new data. With foundation models in their infancy, it is an ideal time to explore how these powerful generalized models can be made more usable.

Future work could explore other methods to make foundation models more stable and usable, especially by allowing the user to choose when and how a task policy should be updated. For example, a robot might come deployed with a suite of policies for very common tasks, with the capacity to learn new tasks from the user through human-in-the-loop learning [208, 155]. Another approach is to use sim-to-real RL to train new policies when needed by the user. For example, Eureka [160] uses LLMs and an iterative training procedure to design reward functions for arbitrary tasks. This approach has similar benefits as DPP, but depends on having an accurate simulator and may not be responsive enough for users, as the robot needs to learn tasks from scratch when the user requests it. An advantage, however, is that since it uses a task-specific reward function, it may be more explainable.

We primarily focused on improving interpretability and modularity relative to generalist policies,

but there are also other exciting directions for future research towards usable generalist policies. For example, work is needed on how to explain the behavior of a generalist robot policy. Explainable AI techniques for large models are constantly improving, but more work is needed to understand how techniques for explaining robot behavior apply to generalist robot policies: much prior work in explainable robotics and AI either assumes the robot was trained with RL (see [172] for a taxonomy of explainable RL) or requires semantic knowledge of its past interactions [230]. Some approaches, such as directly generating explanations for non-interpretable policies are easily applicable to generalist robot policies, while others, such as generating intrinsically interpretable policies, may not be.

8 Future Work

Building on this dissertation, there are three key directions for future work: developing a formalization for categorizing different empowerment strategies, building user-interactive RFMs, and proposing government policy considerations to ensure people may use home robots for their needs. A formalization of user empowerment can be used researchers to development new empowerment strategies, identify when a strategy can be applied to a certain problem, and compare strategies in terms of what they afford to users. RFMs deployed in home environments ought to be able to interact with and collaborate with the people in that home, both to better assist those users in their desired tasks, and so that, through collaborative interaction, people can better understand the RFMs capabilities and what it may be useful for. Regulatory policy can ensure people can use robots in novel ways safely, defining both the safety requirements a robot must have and the protections people have when using the robot. In this chapter, we will review and discuss each of these directions in more detail.

8.1 Towards a Unifying Formalization of User Empowerment through Control

Throughout this dissertation, we have presented works that seek to empower users of robots to have greater control over the robot's behavior. We have demonstrated that empowering people in such a way can be an affective approach for both improving user experience in an interaction and facilitating a robot to perform a novel task. For similar empowerment strategies to be widely adopted and applied to many different HRI scenarios, it is helpful to have a framework that concisely describes how a given approach empowers a user with control and to what extent.

Formalizations that characterize a specific task-based problem have been invaluable and common in the field of HRI. For instance, MDPs [283] and POMDPs [247], have been frequently used to characterize different HRI scenarios from RL-based education robots [198, 73, 300] to human-robot

collaborative assembly tasks [88, 211]. Frameworks that formalize or quantify a human-robot interaction more broadly, outside of the scope of a single task, however, are notoriously difficult to define as tasks are hard to model, people even harder, and their interaction partially-observable at best [141]. Despite this difficulty, such formalizations are nonetheless important for understanding and improving different algorithms and interactions. A formalization for user-empowerment through control should ideally enable one to: measure how empowering a certain approach is relative to another (marginal empowerment), define what a "novel task" is relative to the robot's capabilities and as a consequence of user control, and categorize how a certain approach empowers a user to accomplish their desired outcome as a consequence of the interaction of the user's control and a robot's autonomous behavior. In this section propose and discuss what such a formalization for empowerment through control may look like; how each of the works presented in the previous chapters may fit in; and what work is necessary to be done in the future to realize such a formalization.

8.1.1 User Participation and Robot Autonomy for an Empowerment Formalization

For a formalization of user-empowerment, we will consider a task-based setting where a user has some task they wish to accomplish. Within this setting an assumption we make is that users are intentional when interacting with the robot. In other words, they have a specific outcome or goal they wish to accomplish and they are acting in hopes of achieving it. This assumption is agnostic to a specific model of human decision making or actions, such as assuming a person's actions can be modeled as rational but noisy [216], that their preferences are expressed as reward following the Bradely-Terry Model [193, 263, 35], or are not [193, 137]. Though such models can be useful, particularly the more one knows about an individual and a task [44], it is valuable to make a more minimal assumption that they have an end-goal in mind as alignment or satisfaction with a user's goal is easier to verify, by asking the user for example, than the alignment of a user's behavior with a specific model.

As demonstrated throughout this dissertation, people not only have tasks they wish to accomplish

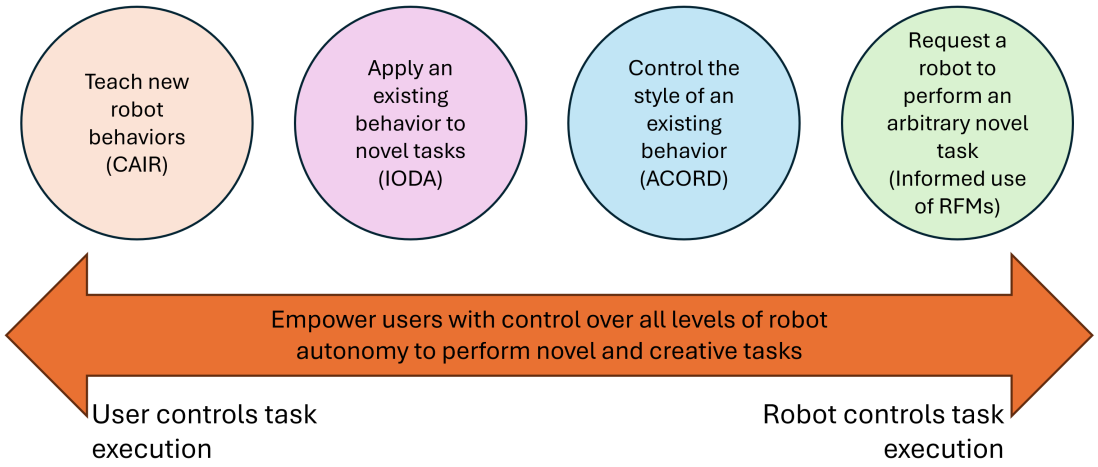


Figure 32: In this dissertation, we present "empowerment strategies" that empower users across different levels of user participation in task completion and robot control.

using a robot, but may want to be more or less involved in the task completion process itself. For example, one may want to query an RFM then walk away as the robot completes the requested task. This is still an example of a person controlling a robot, but they have little to no direct participation in collaborating with the robot to perform the task. On the other hand, a person who wants to directly control parts of the robot, as in IODA, to collaboratively perform the task, naturally is participating more. These cases provide an intuition about different levels, or a range, of user-partition. Such an intuition is important for developing a formalization that captures how empowered a user is over their robot. Similar to human-in-the-loop learning formalizations that quantify human-feedback as information gain to better develop and evaluate HIRL algorithms [59], quantifying the user participation in task completion given a certain algorithm, can be used for measuring empowerment.

Thus, given this setting a formalization of user empowerment through control may involve two "axes" with which to categorize different empowerment strategies: user participation and robot capability or autonomy for a the user's desired task. User participation in the task completion process may involve a person adjusting the parameters of a robot's behavior, manipulating the environment, providing corrective demonstrations, or similar interventions. User participation can potentially be

quantified as the amount of information gain that the robot receives from the user in an information theoretic sense [20], or how much time the user spends interacting with the robot to accomplish a given task. Consider the case of a person teaching a robot through corrective demonstrations, i.e. when the robot makes a mistake, the person will intervene to correct the mistake and the robot will in turn use that correction to update its policy. The number of corrections a user provides or the proportion of correction time to autonomous robot behavior time can be considered a measure of participation. In fact, this is also commonly used a metric for the goodness of algorithms that learn from corrections [124, 49, 272].

The level of robot autonomy should measure how well the robot can complete the desired task on its own. This level of autonomy may also be measured as dependent upon or in-spite of the user's control. In ACORD for example, the user's customization of the robot's behavior does not hinder the robot from completing the task. The robot's autonomy for the given task can potentially be quantified via an evaluation metric for instance, such as TSR, or through a formal reachability analysis [10, 222]. Each of the empowerment strategies in this work can be categorized along these two axes as shown in Figure 32 (this is a repeat of Figure 2 for the reader's convenience). In this case there is a single axis ranging from whether the user or the robot is controlling task execution. This is primarily for visualization purposes. Alternatively, approaches where the user controls task execution can be categorized as having "high" amounts of user participation and "low" amounts of robot autonomy, and vice-versa for approaches where the robot controls task execution. I will explain each example in more detail below.

Teach new robot behaviors (CAIR) The empowerment strategy presented in Chapter 5, is highly dependent on user-participation as the robot does not yet know how to perform the user's requested task. Most HIRL algorithms fit into this categorization, though some may require more or less user participation. For example, the method proposed in [114] includes a pretraining step to learn to generate trajectories for preference learning that reduces the number of labels a teacher needs to provide for the robot to learn the task.

Apply an existing behavior to novel tasks (IODA) The empowerment strategy presented in Chapter 3, is similar to in CAIR in that whether or not the robot completes the user's desired task is dependent upon the user themselves and their input to the robot. A key difference, however, is that the robot already has autonomous function that is the user is leveraging to complete the task. Furthermore, in this case, the user may only need to participate at certain throughout the task execution to achieve their desired goal. Thus, IODA can be thought of as requiring more robot autonomy and potentially less user participation for the desired task to be completed.

Control the style of an existing behavior (ACORD) Unlike IODA and CAIR, ACORD (Chapter 4), is not dependent on the user's participation to complete the task the user specified. However, the user may still be actively involved in how that task is executed. Thus, ACORD can be thought of having high levels of robot autonomy, and low, or very varying, levels of user participation.

Request a robot to perform an arbitrary novel task (Informed use of RFMs) The empowerment strategy presented in Chapter 6 is focused on a situation where the robot completes the task entirely autonomously. The goal of the empowerment strategy to inform the user of the robot's capabilities may lead to the user deciding not to let the robot attempt the task at all, deciding to supervise the robot in case intervention is needed, or deciding to use a different method for task execution (such as HIRL). Thus, this empowerment strategy has a high degree of robot autonomy, and potentially the lowest amount of user participation.

These descriptions can serve as examples for how to categorize other empowerment strategies along these axis. However, the categorizations presented here still have limitations that need to be addressed in future work for a complete formalization. For instance, even in this two-axis framework, the user in CAIR is arguably only participating in the *teaching* process and not necessarily in the task-completion process once the robot knows how to perform the task. This ambiguity is an important one to clarify, especially to define what a "novel task" is and how empowering a certain approach is for facilitating novel-task executions. Furthermore, the two axes of user participation and robot autonomy are insufficient for a definition of a novel task.

8.1.2 Other Considerations

Along with user participation and robot autonomy, other axes of categorization may be useful for both measuring the user's control over the robot, defining what a novel task is, and measuring qualitative user experience. In this dissertation, I did not focus heavily on the concept of alignment, but it can potentially be used as a way of categorizing different empowerment strategies. "Alignment" as it is commonly referred to in the AI and RLHF literature [123], refers to the alignment of a user's desired outcome with an agents behavior and largely forgoes differentiating between a user's desired behavior and what they specify. In a task-based HRI scenario however, where there are many different means of specifying a task, from language to demonstrations, there may be reason to specify a task different from the desired outcome of the interaction. The interaction presented in Chapter 3 is an example of this. Thus, a dual of alignment could be used to categorize both the alignment of the robot's autonomous behavior to the task specified by the user, as well as the alignment of the overall system, including the user's control, to the users desired outcome. Given these two notions of alignment, an example a novel task could be implied by situations where the overall system aligns with the user's desired outcome, but which differs from what the use specified to the robot. Future work is needed for a holisitic definition of novelty, including differentiating between HIRL algorithms and algorithms such as IODA for example.

A common thread throughout much of the study results presented in this dissertation was a generally positive user experience. Future work could consider user-experience as an axis for measuring empowerment. For example, the qualitative amount of effort a user exerts when interacting with a robot is already a common outcome measure of a study, often using the NASA TLX [260] questionnaire [171, 203, 251].

8.2 User-interactive Robot Foundation Models

As discussed in Chapter 6, a RFM research has been largely aimed at creating a model that can autonomously and independently perform various tasks according to a user's request. Especially as

the performance of RFMs continues to improve [31], largely as a result of algorithmic advacements, "user-interactive RFMs" should also be developed. By user-interactive RFM, I mean an RFM that has been explicitly designed to be able to perform collaborative tasks with people. Furthermore, such user-interactive RFMs should be evaluated holistically, including factors the user's subjective experience and the systems ease-of-use. To get a robot to perform success collaborations with people in terms of both task success and user-experience is by no means trivial. In fact, much research in HRI has been dedicated to improving the experience of human-robot handovers for instance [189]. There are various directions of future work that would assist in the development of user-interactive RFMs.

An important part of developing user-interactive RFMs will be to collect data in a standardized format both of users interacting with robots generally and with users specifically interacting with RFMs. These datasets could be used in the training of user-interactive RFMs. Datasets that consist of user-interactions with other types of foundation models, such as LLMs and VLMs, are already widely available and in active use [304, 127, 122, 287]. Similarly, user-centered evaluations of these models are also crucial for their success. [233] for example uses human-annotation to create a benchmark for evaluating a multi-modal foundation model's video-based reasoning capabilities. Meanwhile, as RFMs are improved to perform tasks based on more abstract user-task specifications, typically tasks involving multiple steps [151], non-user-interactive, simulation-based benchmarks are also being developed to evaluate such capabilities [276]. Future work should evaluate user-interactive RFMs by reporting some of the metrics discussed in Chapter 6, as well as user-studies that have novice-users collaborate with an RFM-based robot.

Another important aspect with which people could be introduced is that of fine-tuning. RFMs are typically fine-tuned, or undergo additional training, when they encounter wholly new environments or tasks. Models such as OpenVLA [134] and Octo [258], have open-source code to allow researchers and general users to fine-tune their models. The fine-tuning process, however, is generally computationally expensive and time consuming. While state-of-the-art techniques such as LoRA [109],

have made fine-tuning less computationally expensive by only updating parts of a larger model, incorporating a human-in-the-loop has the potential to both further speed up this process and allow the user to communicate their preferences in the meantime, as in [306].

8.3 On User Safety and Empowerment

8.3.1 A Limitation of Purely Data-driven Approach

In this dissertation we presented four empowerment strategies. All of these strategies are based on data-driven approaches, with three of them based on RL and one based on large-scale imitation learning. Data-driven approaches use data collected from a problem domain as the primary source of information for making decisions and, more often than not, for training a neural network [150, 57, 116, 5]. Two common downsides of this approach, particularly relevant to giving users more control over their robots, are limitations related to explainability and interperetability of the neural networks decisions [76, 3, 227] and related to safety [182, 52, 110]. Recently in RL, there is an increasing effort to introduce explainable elements into the agent's decision making process [172] as well as safety guarantees, often by combining reinforcement learning with symbolic or formal methods [96, 40, 86, 290]. Future work could combine these methods with the empowerment strategies presented in this work. Safety is particularly important as this work proposes empowering users by giving then more control over their robot as to perform novel and creative tasks. The user executing novel and creative tasks should not mean, however, that the user becomes unsafe as a result.

8.3.2 The Role of Policy and Regulation

In this dissertation, we focused on ways to give control to people such that they can perform tasks that where not necessarily foreseen by the designers of the robot. This approach has implications with regards to consumer-manufacturer relations. For example, robot manufacturers may incentivize to have certain novel used void warranty if they pose a risk to the robot. Such warranties must comply with consumer protection law, and thus be relatively clear about what the consumer can and cannot

do (see [162] for an example in US law). Future work should consider how to regulate and draft consumer protection law that permits end-users to use their robot in novel ways so long as such use does not pose unreasonable risks to the user or others.

As mentioned in Chapter 6 and in this section, the evaluation of RFMs, and the availability of the resulting information about the evaluation, can impact a user's experience with and ability to use an RFM. *Transparency* in both the the training and evaluation process is important for ensuring people can safely and comfortably use RFMs. Various laws requiring companies to disclose details about how AI models were trained have already been passed in places like EU [72, 41] and California [1]. Future work should consider how these laws extend to RFMs, particularly as deployed in home robots, and if additional targeted regulation is needed.

9 Summary

This dissertation presented four empowerment strategies for ensuring novice users of home robots have control over the robot's autonomy and how they interact with the robot. Within these strategies, we presented two HRI theoretic contributions, both of which are problem formulations for giving more control over autonomous robot policies to people. These problem statements, and this dissertation in general, have focused on empowering users to use many different functionalities a robot has to accomplish novel and creative tasks alike. The first problem statement was about ensuring a robot's behavior is predictable during partitioned control with a user. We showed that this predictability can lead to better task and user experience outcomes. The second problem statement, "online behavior modification," directed how one may develop robot policies for a given task such that they style of task completion is something controllable by a user. Online behavior modification affords people both the ability to express their preferences to a robot as well as express their creativity in real-time. From these problem statements, future HRI researchers can develop algorithms and techniques that address them and expand on them.

Along with these problem statements, we also introduced three algorithms that empower users at different levels user involvement in task completion, from a robot autonomously performing a task to task completion being mostly dependent on the user. All of these algorithms were evaluated in user-studies which demonstrated that they improve task success over prior human-centered approaches, and often afford high levels of user satisfaction. The first algorithm, Imaginary Out-of-Distribution Actions (IODA), addressed the PC problem formulation by projecting a robot's state back to an in-distribution state when the user's control brought it out-of-distribution. This allowed the robot's behavior to align with the user's expectations of how the robot should behave, allowing them to leverage that behavior in innovative ways. The second algorithm, Adjustable Control of Reinforcement learning Dynamics (ACORD), addressed the online behavior modification problem by learning latent parameters over behavior features that a person could control in real-time. ACORD's deployment in a painting task showed that people reported high levels of expressivity

when using ACORD while its task performance was still relatively strong compared to vanilla RL. Lastly, Continuous Action-space Interactive Reinforcement learning (CAIR), allowed users to teach robots complex tasks using binary feedback. Together, these algorithms contribute to the field of HRI by providing methods to further empower people to have control over robots.

In addition to single-task human robot interactions, we contribute to the understanding of how to empower users of general purpose robots across a variety of tasks. To do this, we had non-expert users give feedback on real robot foundation model (RFM) performance evaluations. From this study, we learned both about how non-experts use otherwise technical information as well as types of information people would went when using an RFM based robot. By being more informed about a robot's capabilities, people can reduce the risk of unwanted failures and use the robot for the purposes they need without worry. This dissertation also had a couple complementary contributions. Based on our RFM study results, we made several recommendations for RFM and HRI researchers to facilitate human-centered design and evaluation of RFMs. We also introduced Diffusion for Policy Parameters, an algorithm for RFMs which is an alternative to the general policy approach. DPP ensures users have control over task-specific policies and when their behavior affects the underlying RFM.

This dissertation has primarily concerned with developing approaches that enable users to perform novel tasks with minimal effort or robot expertise. An area of future work directly developing techniques that empower end-users to developing frameworks that holistically capture the process and results of designing and deploying such techniques. In other words, extend the control users have to the design process itself. Similarly, the RFM design pipeline from expert designer to novice user should be further studied. This will provide a better understanding of how research can be performed and reported to empower users without sacrificing the rigor or innovation of the research, but rather enhances it. For example, more robust evaluations of RFMs that take into account non-expert information needs can enhance both expert and non-expert understanding of that model's performance. What is learned from such evaluations can be used to develop interfaces for novices

and design principles experts that promote informing and giving control to users.

Future work could also aim to ensure that RFMs are capable of meaningful, transparent, and safe collaboration with people. There are various paths towards achieving this, including AI-related policy initiatives, developing novel RFM-human collaboration studies, and collecting human-robot interaction data sets for RFM training. Overall, my research will continue to work towards the vision of having active and empowered users of home robots. As robots become more and more capable, the potential for people to make use of those robots for arbitrary tasks also increases; it is important that end-users are beneficiaries of this progress.

10 Acronyms

- **IODA**: Imaginary Out-of-Distribution Actions
- **PC**: Partitioned Control
- **ACORD**: Adjustable Control of Reinforcement learning
- **CAIR**: Continuous Action-space Interactive Reinforcement learning
- **DPP**: Diffusion for Policy Parameters
- **RL**: Reinforcement Learning
- **MDP**: Markov Decision Process
- **POMDP**: Partially Observable Markov Decision Process
- **HIRL**: Human-In-the-loop Robot Learning (HIRL)
- **IntRL**: Interactive Reinforcement Learning
- **LfD**: Learning from Demonstration
- **SC**: Shared Control
- **SA**: Shared Autonomy
- **RFM**: Robot Foundation Model
- **SAC**: Soft Actor-Critic
- **OOD**: Out-Of-Distribution
- **TSR**: Task Success Rate
- **FC**: Failure Case
- **ETSR**: Estimated Task Success Rate
- **EFC**: Estimated Failure Case
- **RT-TSR**: Related Task-Task Success Rate
- **RT-FC**: Related Task-Failure Case
- **BF**: Bayes Factor
- **RLHF**: Reinforcement Learning from Human Feedback

References

[1] AB 2013- CHAPTERED. en. URL: https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240AB2013 (visited on 04/28/2025).

[2] Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. User-centered design. In: *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications* 37.4 (2004), pp. 445–456.

[3] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). In: *IEEE Access* 6 (2018), pp. 52138–52160. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2870052. URL: https://ieeexplore.ieee.org/abstract/document/8466590 (visited on 04/28/2025).

[4] Alejandro Gabriel Agostini, Carme Torras, and Florentin Wörgötter. Integrating task planning and interactive learning for robots to work in human environments. eng. In: Accepted: 2011-12-01T13:10:46Z. AAAI Press. Association for the Advancement of Artificial Intelligence, 2011, pp. 2386–2391. URL: https://upcommons.upc.edu/handle/2117/14136 (visited on 04/27/2025).

[5] Kashif Ahmad, Waleed Iqbal, Ammar El-Hassan, Junaid Qadir, Driss Benhaddou, Moussa Ayyash, and Ala Al-Fuqaha. Data-Driven Artificial Intelligence in Education: A Comprehensive Review. In: *IEEE Transactions on Learning Technologies* 17 (2024), pp. 12–31. ISSN: 1939-1382. DOI: 10.1109/TLT.2023.3314610. URL: https://ieeexplore.ieee.org/abstract/document/10247566 (visited on 04/28/2025).

[6] 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*. HRI '12. New York, NY, USA: Association for Computing Machinery, Mar. 2012, pp. 391–398. ISBN: 978-1-4503-1063-5. DOI: 10.1145/2157689.2157815. URL: https://dl.acm.org/doi/10.1145/2157689.2157815 (visited on 04/26/2025).

[7] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. HyperStyle: StyleGAN Inversion With HyperNetworks for Real Image Editing. en. In: 2022, pp. 18511–18521. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Alaluf_HyperStyle_

StyleGAN_Inversion_With_HyperNetworks_for_Real_Image_Editing_CVPR_2022_paper.html (visited on 04/27/2025).

[8] Katherine H. Allen, Audrey K. Balaska, Reuben M. Aronson, Chris Rogers, and Elaine Schaertl Short. Barriers and Benefits: The Path to Accessible Makerspaces. In: *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS '23. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 1–14. ISBN: 9798400702204. DOI: 10.1145/3597638.3608414. URL: https://dl.acm.org/doi/10.1145/3597638.3608414 (visited on 04/03/2025).

[9] Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe Reinforcement Learning via Shielding. arXiv:1708.08611 [cs]. Sept. 2017. DOI: 10.48550/arXiv.1708.08611. URL: http://arxiv.org/abs/1708.08611 (visited on 12/27/2023).

[10] Matthias Althoff, Goran Frehse, and Antoine Girard. Set Propagation Techniques for Reachability Analysis. en. In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.Volume 4, 2021 (May 2021). Publisher: Annual Reviews, pp. 369–395. ISSN: 2573-5144. DOI: 10.1146/annurev-control-071420-081941. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-071420-081941 (visited on 04/28/2025).

[11] Eitan Altman. Constrained Markov Decision Processes: Stochastic Modeling. en. 1st ed. Boca Raton: Routledge, Dec. 2021. ISBN: 978-1-315-14022-3. DOI: 10.1201/9781315140223. URL: https://www.taylorfrancis.com/books/9781315140223 (visited on 06/05/2023).

[12] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. arXiv:1707.01495 [cs]. Feb. 2018. URL: http://arxiv.org/abs/1707.01495 (visited on 09/24/2023).

[13] Riku Arakawa, Sosuke Kobayashi, Yuya Unno, Yuta Tsuboi, and Shin-ichi Maeda. DQN-TAMER: Human-in-the-Loop Reinforcement Learning with Intractable Feedback. In: *arXiv:1810.11748 [cs]* (Oct. 2018). URL: http://arxiv.org/abs/1810.11748 (visited on 02/11/2021).

[14] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. en. In: *Robotics and Autonomous Systems* 57.5 (May 2009),

pp. 469–483. ISSN: 09218890. DOI: 10.1016/j.robot.2008.10.024. URL: https://linkinghub.elsevier.com/retrieve/pii/S0921889008001772 (visited on 12/11/2022).

[15] Reuben M. Aronson and Elaine Schaertl Short. Intentional User Adaptation to Shared Control Assistance. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 4–12. ISBN: 9798400703225. DOI: 10.1145/3610977.3634953. URL: https://dl.acm.org/doi/10.1145/3610977.3634953 (visited on 05/07/2024).

[16] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. In: *Artificial Intelligence* 297 (Aug. 2021), p. 103500. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103500. URL: https://www.sciencedirect.com/science/article/pii/S0004370221000515 (visited on 04/26/2025).

[17] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A Brief Survey of Deep Reinforcement Learning. In: *IEEE Signal Processing Magazine* 34.6 (Nov. 2017). arXiv:1708.05866 [cs, stat], pp. 26–38. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2743240. URL: http://arxiv.org/abs/1708.05866 (visited on 07/19/2023).

[18] Dilip Arumugam, Jun Ki Lee, Sophie Saskin, and Michael L. Littman. Deep Reinforcement Learning from Policy-Dependent Human Feedback. In: *arXiv:1902.04257 [cs, stat]* (Feb. 2019). URL: http://arxiv.org/abs/1902.04257 (visited on 10/05/2020).

[19] Christian Arzate Cruz and Takeo Igarashi. A Survey on Interactive Reinforcement Learning: Design Principles and Open Challenges. en. In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. Eindhoven Netherlands: ACM, July 2020, pp. 1195–1209. ISBN: 978-1-4503-6974-9. DOI: 10.1145/3357236.3395525. URL: https://dl.acm.org/doi/10.1145/3357236.3395525 (visited on 11/19/2020).

[20] Robert B Ash. Information theory. Courier Corporation, 2012.

[21] Sayantan Auddy, Jakob Hollenstein, Matteo Saveriano, Antonio Rodríguez-Sánchez, and Justus Piater. Scalable and Efficient Continual Learning from Demonstration via a Hypernetwork-generated Stable Dynamics Model. arXiv:2311.03600 [cs]. Jan. 2024. DOI: 10.48550/arXiv.2311.03600. URL: http://arxiv.org/abs/2311.03600 (visited on 04/27/2025).

[22] Anthony L. Baker, Elizabeth K. Phillips, Daniel Ullman, and Joseph R. Keebler. Toward an Understanding of Trust Repair in Human-Robot Interaction: Current Research and Future Directions. In: *ACM Trans. Interact. Intell. Syst.* 8.4 (Nov. 2018), 30:1–30:30. ISSN: 2160-6455. DOI: 10.1145/3181671. URL: https://dl.acm.org/doi/10.1145/3181671 (visited on 10/17/2024).

[23] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A Survey of Meta-Reinforcement Learning. arXiv:2301.08028 [cs]. Aug. 2024. DOI: 10.48550/arXiv.2301.08028. URL: http://arxiv.org/abs/2301.08028 (visited on 04/27/2025).

[24] Jacob Beck, Risto Vuorio, Zheng Xiong, and Shimon Whiteson. Recurrent Hypernetworks are Surprisingly Strong in Meta-RL. en. In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 62121–62138. URL: https://proceedings.neurips.cc/paper_files/paper/2023/hash/c3fa3a7d50b34732c6d08f6f66380d75-Abstract-Conference.html (visited on 04/27/2025).

[25] Feryal Behbahani, Kyriacos Shiarlis, Xi Chen, Vitaly Kurin, Sudhanshu Kasewa, Ciprian Stirbu, João Gomes, Supratik Paul, Frans A. Oliehoek, João Messias, and Shimon Whiteson. Learning From Demonstration in the Wild. In: *2019 International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2019, pp. 775–781. DOI: 10.1109/ICRA.2019.8794412. URL: https://ieeexplore.ieee.org/abstract/document/8794412 (visited on 04/26/2025).

[26] Suneel Belkhale and Dorsa Sadigh. MiniVLA: A Better VLA with a Smaller Footprint. 2024. URL: https://github.com/Stanford-ILIAD/openvla-mini.

[27] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA: Association for Computing Machinery, June 2009, pp. 41–48. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553380. URL: https://dl.acm.org/doi/10.1145/1553374.1553380 (visited on 04/28/2025).

[28] Manuel Bied and Mohamed Chetouani. Integrating an Observer in Interactive Reinforcement Learning to Learn Legible Trajectories. In: *2020 29th IEEE International Conference on Robot*

and Human Interactive Communication (RO-MAN). ISSN: 1944-9437. Aug. 2020, pp. 760–767. DOI: 10.1109/RO-MAN47096.2020.9223338.

[29] Erdem Biyik, Nicolas Huynh, Mykel Kochenderfer, and Dorsa Sadigh. Active Preference-Based Gaussian Process Regression for Reward Learning. en. In: *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020. ISBN: 978-0-9923747-6-1. DOI: 10.15607/RSS.2020.XVI.041. URL: http://www.roboticsproceedings.org/rss16/p041.pdf (visited on 08/22/2022).

[30] Erdem Bıyık, Dylan P. Losey, Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. In: *The International Journal of Robotics Research* 41.1 (Jan. 2022). Publisher: SAGE Publications Ltd STM, pp. 45–67. ISSN: 0278-3649. DOI: 10.1177/02783649211041652. URL: https://doi.org/10.1177/02783649211041652 (visited on 06/22/2022).

[31] Kevin Black et al. $03c0_0$: A Vision-Language-Action Flow Model for General Robot Control. arXiv:2410.24164 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2410.24164. URL: http://arxiv.org/abs/2410.24164 (visited on 04/28/2025).

[32] Andreea Bobu, Andi Peng, Pulkit Agrawal, Julie A Shah, and Anca D. Dragan. Aligning Human and Robot Representations. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 42–54. ISBN: 9798400703225. DOI: 10.1145/3610977.3634987. URL: https://dl.acm.org/doi/10.1145/3610977.3634987 (visited on 05/22/2024).

[33] Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D. Dragan. Feature Expansive Reward Learning: Rethinking Human Input. In: *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction* (Mar. 2021). arXiv: 2006.13208, pp. 216–224. DOI: 10.1145/3434073.3444667. URL: http://arxiv.org/abs/2006.13208 (visited on 05/03/2022).

[34] Serena Booth, W. Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. The Perils of Trial-and-Error Reward Design: Misdesign through Overfitting and Invalid Task Specifications. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.5 (June 2023). Number: 5, pp. 5920–5929. ISSN: 2374-3468. DOI: 10.1609/aaai.v37i5.

25733. URL: https://ojs.aaai.org/index.php/AAAI/article/view/25733 (visited on 03/04/2025).

[35] Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. In: *Biometrika* 39.3/4 (1952). Publisher: [Oxford University Press, Biometrika Trust], pp. 324–345. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2334029 (visited on 04/13/2025).

[36] Jake Brawer, Debasmita Ghose, Kate Candon, Meiying Qin, Alessandro Roncone, Marynel Vázquez, and Brian Scassellati. Interactive Policy Shaping for Human-Robot Collaboration with Transparent Matrix Overlays. en. In: *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*. Stockholm Sweden: ACM, Mar. 2023, pp. 525–533. ISBN: 978-1-4503-9964-7. DOI: 10.1145/3568162.3576983. URL: https://dl.acm.org/doi/10.1145/3568162.3576983 (visited on 04/24/2023).

[37] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. arXiv:1606.01540 [cs]. June 2016. URL: http://arxiv.org/abs/1606.01540 (visited on 06/07/2023).

[38] Anthony Brohan et al. RT-1: Robotics Transformer for Real-World Control at Scale. arXiv:2212.06817. Aug. 2023. DOI: 10.48550/arXiv.2212.06817. URL: http://arxiv.org/abs/2212.06817 (visited on 10/17/2024).

[39] Anthony Brohan et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. arXiv:2307.15818 [cs]. July 2023. DOI: 10.48550/arXiv.2307.15818. URL: http://arxiv.org/abs/2307.15818 (visited on 05/07/2024).

[40] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. en. In: *Annual Review of Control, Robotics, and Autonomous Systems* 5.Volume 5, 2022 (May 2022). Publisher: Annual Reviews, pp. 411–444. ISSN: 2573-5144. DOI: 10.1146/annurev-control-042920-020211. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211 (visited on 04/28/2025).

[41] Adam Buick. Copyright and AI training data—transparency to the rescue? In: *Journal of Intellectual Property Law & Practice* 20.3 (Mar. 2025), pp. 182–192. ISSN: 1747-1532. DOI:

10.1093/jiplp/jpae102. URL: https://doi.org/10.1093/jiplp/jpae102 (visited on 04/28/2025).

[42] Baptiste Busch, Jonathan Grizou, Manuel Lopes, and Freek Stulp. Learning Legible Motion from Human–Robot Interactions. en. In: *International Journal of Social Robotics* 9.5 (Nov. 2017), pp. 765–779. ISSN: 1875-4805. DOI: 10.1007/s12369-017-0400-4. URL: https://doi.org/10.1007/s12369-017-0400-4 (visited on 07/24/2023).

[43] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. LeRobot: State-of-the-art Machine Learning for Real-World Robotics in Pytorch. 2024. URL: https://github.com/huggingface/lerobot.

[44] Muffy Calder, Claire Craig, Dave Culley, Richard De Cani, Christl A. Donnelly, Rowan Douglas, Bruce Edmonds, Jonathon Gascoigne, Nigel Gilbert, Caroline Hargrove, Derwen Hinds, David C. Lane, Dervilla Mitchell, Giles Pavey, David Robertson, Bridget Rosewell, Spencer Sherwin, Mark Walport, and Alan Wilson. Computational modelling for decision-making: where, why, what, who and how. en. In: *Royal Society Open Science* 5.6 (June 2018), p. 172096. ISSN: 2054-5703. DOI: 10.1098/rsos.172096. URL: https://royalsocietypublishing.org/doi/10.1098/rsos.172096 (visited on 04/13/2025).

[45] Mehmet Ege Cansev, Honghu Xue, Nils Rottmann, Adna Bliek, Luke E. Miller, Elmar Rueckert, and Philipp Beckerle. Interactive Human–Robot Skill Transfer: A Review of Learning Methods and User Experience. en. In: *Advanced Intelligent Systems* 3.7 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202000247, p. 2000247. ISSN: 2640-4567. DOI: 10.1002/aisy.202000247. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202000247 (visited on 04/28/2025).

[46] Carlos Celemin and Javier Ruiz-del-Solar. An Interactive Framework for Learning Continuous Actions Policies Based on Corrective Feedback. en. In: *Journal of Intelligent & Robotic Systems* 95.1 (July 2019), pp. 77–97. ISSN: 1573-0409. DOI: 10.1007/s10846-018-0839-z. URL: https://doi.org/10.1007/s10846-018-0839-z (visited on 02/22/2021).

[47] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. In: *CoRR* abs/2306.13831 (2023).

[48] 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. en. arXiv:2303.04137 [cs]. Mar. 2024. URL: http://arxiv.org/abs/2303.04137 (visited on 05/07/2024).

[49] Eugenio Chisari, Tim Welschehold, Joschka Boedecker, Wolfram Burgard, and Abhinav Valada. Correct Me If I am Wrong: Interactive Learning for Robotic Manipulation. In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), pp. 3695–3702. ISSN: 2377-3766. DOI: 10.1109/LRA.2022.3145516. URL: https://ieeexplore.ieee.org/abstract/document/9691826 (visited on 04/15/2025).

[50] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. Number: arXiv:1706.03741 arXiv:1706.03741 [cs, stat]. July 2017. URL: http://arxiv.org/abs/1706.03741 (visited on 07/28/2022).

[51] Konstantinos Christofi and Kim Baraka. Uncovering Patterns in Humans that Teach Robots through Demonstrations and Feedback. In: *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 332–336. ISBN: 9798400703232. DOI: 10.1145/3610978.3640740. URL: https://dl.acm.org/doi/10.1145/3610978.3640740 (visited on 03/05/2025).

[52] Jaymari Chua, Yun Li, Shiyi Yang, Chen Wang, and Lina Yao. AI Safety in Generative AI Large Language Models: A Survey. arXiv:2407.18369 [cs]. July 2024. DOI: 10.48550/arXiv.2407.18369. URL: http://arxiv.org/abs/2407.18369 (visited on 04/28/2025).

[53] Matei Ciocarlie, Kaijen Hsiao, Adam Leeper, and David Gossow. Mobile manipulation through an assistive home robot. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ISSN: 2153-0866. Oct. 2012, pp. 5313–5320. DOI: 10.1109/IROS.2012.6385907.

[54] Vanya Cohen, Jason Xinyu Liu, Raymond Mooney, Stefanie Tellex, and David Watkins. A Survey of Robotic Language Grounding: Tradeoffs between Symbols and Embeddings. arXiv:2405.13245 [cs]. June 2024. DOI: 10.48550/arXiv.2405.13245. URL: http://arxiv.org/abs/2405.13245 (visited on 04/27/2025).

[55] Open X.-Embodiment Collaboration et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. arXiv:2310.08864 [cs]. June 2024. DOI: 10.48550/arXiv.2310.08864. URL: http://arxiv.org/abs/2310.08864 (visited on 04/26/2025).

[56] Gabriele Costante, Enrico Bellocchio, Paolo Valigi, and Elisa Ricci. Personalizing vision-based gestural interfaces for HRI with UAVs: a transfer learning approach. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ISSN: 2153-0866. Sept. 2014, pp. 3319–3326. DOI: 10.1109/IROS.2014.6943024. URL: https://ieeexplore.ieee.org/abstract/document/6943024 (visited on 04/28/2025).

[57] Kathrin Cresswell, Margaret Callaghan, Sheraz Khan, Zakariya Sheikh, Hajar Mozaffar, and Aziz Sheikh. Investigating the use of data-driven artificial intelligence in computerised decision support systems for health and social care: A systematic review. EN. In: *Health Informatics Journal* 26.3 (Sept. 2020). Publisher: SAGE Publications Ltd, pp. 2138–2147. ISSN: 1460-4582. DOI: 10.1177/1460458219900452. URL: https://doi.org/10.1177/1460458219900452 (visited on 04/28/2025).

[58] Francisco Cruz, Richard Dazeley, Peter Vamplew, and Ithan Moreira. Explainable robotic systems: understanding goal-driven actions in a reinforcement learning scenario. en. In: *Neural Computing and Applications* (Aug. 2021). ISSN: 1433-3058. DOI: 10.1007/s00521-021-06425-5. URL: https://doi.org/10.1007/s00521-021-06425-5 (visited on 07/24/2023).

[59] Yuchen Cui, Pallavi Koppol, Henny Admoni, Scott Niekum, Reid Simmons, Aaron Steinfeld, and Tesca Fitzgerald. Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning. en. In: vol. 5. ISSN: 1045-0823. Aug. 2021, pp. 4382–4391. DOI: 10.24963/ijcai.2021/599. URL: https://www.ijcai.org/proceedings/2021/599 (visited on 04/14/2025).

[60] Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can Foundation Models Perform Zero-Shot Task Specification For Robot Manipulation? en. In: *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*. ISSN: 2640-3498. PMLR, May 2022, pp. 893–905. URL: https://proceedings.mlr.press/v168/cui22a.html (visited on 04/27/2025).

[61] Devleena Das, Siddhartha Banerjee, and Sonia Chernova. Explainable AI for Robot Failures: Generating Explanations that Improve User Assistance in Fault Recovery. In: *Proceedings of*

the 2021 ACM/IEEE International Conference on Human-Robot Interaction. HRI '21. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 351–360. ISBN: 978-1-4503-8289-2. DOI: 10.1145/3434073.3444657. URL: https://dl.acm.org/doi/10.1145/3434073.3444657 (visited on 07/24/2023).

[62] Shivin Dass, Karl Pertsch, Hejia Zhang, Youngwoon Lee, Joseph Lim, and Stefanos Nikolaidis. PATO: Policy Assisted TeleOperation for Scalable Robot Data Collection. en. In: *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN: 978-0-9923747-9-2. DOI: 10.15607/RSS.2023.XIX.013. URL: http://www.roboticsproceedings.org/rss19/p013.pdf (visited on 08/05/2023).

[63] Joseph DelPreto, Jeffrey I. Lipton, Lindsay Sanneman, Aidan J. Fay, Christopher Fourie, Changhyun Choi, and Daniela Rus. Helping Robots Learn: A Human-Robot Master-Apprentice Model Using Demonstrations via Virtual Reality Teleoperation. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2020, pp. 10226–10233. DOI: 10.1109/ICRA40945.2020.9196754. URL: https://ieeexplore.ieee.org/abstract/document/9196754 (visited on 04/26/2025).

[64] Munjal Desai, Poornima Kaniarasu, Mikhail Medvedev, Aaron Steinfeld, and Holly Yanco. Impact of robot failures and feedback on real-time trust. In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. ISSN: 2167-2148. Mar. 2013, pp. 251–258. DOI: 10.1109/HRI.2013.6483596. URL: https://ieeexplore.ieee.org/abstract/document/6483596 (visited on 10/17/2024).

[65] Maximilian Diehl and Karinne Ramirez-Amaro. A causal-based approach to explain, predict and prevent failures in robotic tasks. In: *Robotics and Autonomous Systems* 162 (Apr. 2023), p. 104376. ISSN: 0921-8890. DOI: 10.1016/j.robot.2023.104376. URL: https://www.sciencedirect.com/science/article/pii/S0921889023000155 (visited on 10/17/2024).

[66] Johnny van Doorn, Don van den Bergh, Udo Böhm, Fabian Dablander, Koen Derks, Tim Draws, Alexander Etz, Nathan J. Evans, Quentin F. Gronau, Julia M. Haaf, Max Hinne, Šimon Kucharský, Alexander Ly, Maarten Marsman, Dora Matzke, Akash R. Komarlu Narendra Gupta, Alexandra Sarafoglou, Angelika Stefan, Jan G. Voelkel, and Eric-Jan Wagenmakers. The JASP guidelines for conducting and reporting a Bayesian analysis. en. In: *Psychonomic Bulletin & Review* 28.3 (June 2021), pp. 813–826. ISSN: 1531-5320. DOI: 10.3758/

s13423-020-01798-5. URL: https://doi.org/10.3758/s13423-020-01798-5 (visited on 01/31/2023).

[67] Anca Dragan and Siddhartha Srinivasa. Generating Legible Motion. en. In: *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, June 2013. ISBN: 978-981-07-3937-9. DOI: 10.15607/RSS.2013.IX.024. URL: http://www.roboticsproceedings.org/rss09/p24.pdf (visited on 07/24/2023).

[68] Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A Vision-Language-Model for Detecting and Reasoning Over Failures in Robotic Manipulation. arXiv:2410.00371. Oct. 2024. DOI: 10.48550/arXiv.2410.00371. URL: http://arxiv.org/abs/2410.00371 (visited on 10/17/2024).

[69] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-Shot Imitation Learning. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/ba3866600c3540f67c1e9575e213be0a-Abstract.html (visited on 04/26/2025).

[70] Burak Ercan, Onur Eker, Canberk Saglam, Aykut Erdem, and Erkut Erdem. HyperE2VID: Improving Event-Based Video Reconstruction via Hypernetworks. In: *IEEE Transactions on Image Processing* 33 (2024), pp. 1826–1837. ISSN: 1941-0042. DOI: 10.1109/TIP.2024.3372460. URL: https://ieeexplore.ieee.org/abstract/document/10462903 (visited on 04/27/2025).

[71] Haritheja Etukuru, Norihito Naka, Zijin Hu, Seungjae Lee, Julian Mehu, Aaron Edsinger, Chris Paxton, Soumith Chintala, Lerrel Pinto, and Nur Muhammad Mahi Shafiullah. Robot Utility Models: General Policies for Zero-Shot Deployment in New Environments. arXiv:2409.05865 [cs]. Sept. 2024. DOI: 10.48550/arXiv.2409.05865. URL: http://arxiv.org/abs/2409.05865 (visited on 04/26/2025).

[72] EU Artificial Intelligence Act — Up-to-date developments and analyses of the EU AI Act. en-US. URL: https://artificialintelligenceact.eu/ (visited on 04/28/2025).

[73] Bisni Fahad Mon, Asma Wasfi, Mohammad Hayajneh, Ahmad Slim, and Najah Abu Ali. Reinforcement Learning in Education: A Literature Review. en. In: *Informatics* 10.3 (Sept.

2023). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 74. ISSN: 2227-9709. DOI: 10.3390/informatics10030074. URL: https://www.mdpi.com/2227-9709/10/3/74 (visited on 04/14/2025).

[74] Miguel Faria, Francisco S. Melo, and Ana Paiva. Understanding Robots: Making Robots More Legible in Multi-Party Interactions. In: *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. ISSN: 1944-9437. Aug. 2021, pp. 1031–1036. DOI: 10.1109/RO-MAN50785.2021.9515485.

[75] Taylor Kessler Faulkner, Elaine Schaertl Short, and Andrea Lockerd Thomaz. Policy Shaping with Supervisory Attention Driven Exploration. en. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 842–847. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8594312. URL: https://ieeexplore.ieee.org/document/8594312/ (visited on 09/13/2020).

[76] Carlos Fernández-Loría, Foster Provost, and Xintian Han. Explaining Data-Driven Decisions made by AI Systems: The Counterfactual Approach. arXiv:2001.07417 [cs]. Oct. 2021. DOI: 10.48550/arXiv.2001.07417. URL: http://arxiv.org/abs/2001.07417 (visited on 04/28/2025).

[77] Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit Behavioral Cloning. en. In: *Proceedings of the 5th Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Jan. 2022, pp. 158–168. URL: https://proceedings.mlr.press/v164/florence22a.html (visited on 04/26/2025).

[78] Matthew Fontaine and Stefanos Nikolaidis. A Quality Diversity Approach to Automatically Generating Human-Robot Interaction Scenarios in Shared Autonomy. arXiv:2012.04283 [cs]. June 2021. DOI: 10.48550/arXiv.2012.04283. URL: http://arxiv.org/abs/2012.04283 (visited on 07/20/2023).

[79] Matthew Fontaine and Stefanos Nikolaidis. Differentiable Quality Diversity. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 10040–10052. URL: https://proceedings.neurips.cc/paper/2021/hash/532923f11ac97d3e7cb0130315b067dc-Abstract.html (visited on 03/05/2024).

[80] Franka Robotics. URL: https://franka.de/.

[81] Tyler Frasca, Bradley Oosterveld, Meia Chita-Tegmark, and Matthias Scheutz. Enabling Fast Instruction-Based Modification of Learned Robot Skills. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.7 (May 2021). Number: 7, pp. 6075–6083. ISSN: 2374-3468. DOI: 10.1609/aaai.v35i7.16757. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16757 (visited on 04/26/2025).

[82] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – A Differentiable Physics Engine for Large Scale Rigid Body Simulation. In: *arXiv:2106.13281 [cs]* (June 2021). arXiv: 2106.13281. URL: http://arxiv.org/abs/2106.13281 (visited on 04/11/2022).

[83] Justin Fu, Katie Luo, and Sergey Levine. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning. arXiv:1710.11248 [cs]. Aug. 2018. DOI: 10.48550/arXiv.1710.11248. URL: http://arxiv.org/abs/1710.11248 (visited on 04/26/2025).

[84] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods. In: *arXiv:1802.09477 [cs, stat]* (Oct. 2018). URL: http://arxiv.org/abs/1802.09477 (visited on 02/25/2021).

[85] Scott Fujimoto, David Meger, and Doina Precup. Off-Policy Deep Reinforcement Learning without Exploration. In: *arXiv:1812.02900 [cs, stat]* (Aug. 2019). URL: http://arxiv.org/abs/1812.02900 (visited on 10/08/2021).

[86] Nathan Fulton and André Platzer. Safe Reinforcement Learning via Formal Methods: Toward Safe Control Through Proof and Learning. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). Number: 1. ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.12107. URL: https://ojs.aaai.org/index.php/AAAI/article/view/12107 (visited on 04/28/2025).

[87] Javier Garcıa and Fernando Fernandez. A Comprehensive Survey on Safe Reinforcement Learning. en. In: ().

[88] Ali Ghadirzadeh, Xi Chen, Wenjie Yin, Zhengrong Yi, Mårten Björkman, and Danica Kragic. Human-Centered Collaborative Robots With Deep Reinforcement Learning. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 566–571. ISSN: 2377-3766. DOI: 10.1109/LRA.2020.3047730. URL: https://ieeexplore.ieee.org/abstract/document/9309387 (visited on 04/14/2025).

[89] Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Tan, Lawrence Chen, Quan Vuong, Ted Xiao, Pannag Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. en. In: *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, July 2024. ISBN: 9798990284807. DOI: 10.15607/RSS.2024.XX.090. URL: http://www.roboticsproceedings.org/rss20/p090.pdf (visited on 10/17/2024).

[90] Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. en. In: (). URL: https://octo-models.github.io.

[91] Kevin A. Gluck and John E. Laird, eds. Interactive Task Learning: Humans, Robots, and Agents Acquiring New Tasks through Natural Interactions. The MIT Press, Sept. 2019. ISBN: 978-0-262-34942-0. DOI: 10.7551/mitpress/11956.001.0001. URL: https://doi.org/10.7551/mitpress/11956.001.0001 (visited on 04/27/2025).

[92] Miyu Goko, Motonari Kambara, Daichi Saito, Seitaro Otsuki, and Komei Sugiura. Task Success Prediction for Open-Vocabulary Manipulation Based on Multi-Level Aligned Representations. en. In: Sept. 2024. URL: https://openreview.net/forum?id=QtCtY8zl2T (visited on 10/17/2024).

[93] Deepak Gopinath, Siddarth Jain, and Brenna D. Argall. Human-in-the-Loop Optimization of Shared Autonomy in Assistive Robotics. In: *IEEE Robotics and Automation Letters* 2.1 (Jan. 2017). Conference Name: IEEE Robotics and Automation Letters, pp. 247–254. ISSN: 2377-3766. DOI: 10.1109/LRA.2016.2593928.

[94] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy Shaping: Integrating Human Feedback with Reinforcement Learning. In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 2625–2633. (Visited on 09/13/2020).

[95] Miguel Grinberg. Flask web development: developing web applications with python. " O'Reilly Media, Inc.", 2018.

[96] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A Review of Safe Reinforcement Learning: Methods, Theories, and Applications. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (Dec. 2024), pp. 11216–11235. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2024.3457538. URL: https://ieeexplore.ieee.org/abstract/document/10675394 (visited on 04/28/2025).

[97] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. en. In: *Proceedings of The 6th Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Mar. 2023, pp. 175–187. URL: https://proceedings.mlr.press/v205/guhur23a.html (visited on 04/26/2025).

[98] David Ha, Andrew Dai, and Quoc V. Le. HyperNetworks. arXiv:1609.09106 [cs]. Dec. 2016. DOI: 10.48550/arXiv.1609.09106. URL: http://arxiv.org/abs/1609.09106 (visited on 04/27/2025).

[99] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In: *arXiv:1801.01290 [cs, stat]* (Aug. 2018). URL: http://arxiv.org/abs/1801.01290 (visited on 02/11/2021).

[100] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications. In: *arXiv:1812.05905 [cs, stat]* (Jan. 2019). URL: http://arxiv.org/abs/1812.05905 (visited on 06/08/2021).

[101] Tom Haider, Karsten Roscher, Felippe Schmoeller da Roza, and Stephan Günnemann. Out-of-Distribution Detection for Reinforcement Learning Agents with Probabilistic Dynamics Models. In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '23. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2023, pp. 851–859. ISBN: 978-1-4503-9432-1. (Visited on 06/19/2023).

[102] Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. BAKU: An Efficient Transformer for Multi-Task Policy Learning. en. In: Nov. 2024. URL: https://openreview.net/forum?id=uFXGsiYkkX&noteId=SowL7B05mC (visited on 01/01/2025).

[103] Josiah P. Hanna, Siddharth Desai, Haresh Karnan, Garrett Warnell, and Peter Stone. Grounded action transformation for sim-to-real reinforcement learning. en. In: *Machine Learning* 110.9 (Sept. 2021), pp. 2469–2499. ISSN: 1573-0565. DOI: 10.1007/s10994-021-05982-z. URL: https://doi.org/10.1007/s10994-021-05982-z (visited on 03/04/2025).

[104] Keliang He, Morteza Lahijanian, Lydia E. Kavraki, and Moshe Y. Vardi. Towards manipulation planning with temporal logic specifications. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 1050-4729. May 2015, pp. 346–352. DOI: 10.1109/ICRA.2015.7139022. URL: https://ieeexplore.ieee.org/abstract/document/7139022 (visited on 04/27/2025).

[105] Shashank Hegde, Zhehui Huang, and Gaurav S. Sukhatme. HyperPPO: A scalable method for finding small policies for robotic control. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 10821–10828. DOI: 10.1109/ICRA57147.2024.10610861. URL: https://ieeexplore.ieee.org/abstract/document/10610861 (visited on 04/27/2025).

[106] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs, stat]. Dec. 2020. DOI: 10.48550/arXiv.2006.11239. URL: http://arxiv.org/abs/2006.11239 (visited on 05/07/2024).

[107] Shanee Honig and Tal Oron-Gilad. Understanding and Resolving Failures in Human-Robot Interaction: Literature Review and Model Development. en. In: *Frontiers in Psychology* 9 (June 2018), p. 861. DOI: 10.3389/fpsyg.2018.00861. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC6013580/ (visited on 10/17/2024).

[108] Long-Jing Hsu, Weslie Khoo, Manasi Swaminathan, Kyrie Jig Amon, Rasika Muralidharan, Hiroki Satov, Min Min Thant, Anna S. Kim, Katherine M Tsui, David J Crandall, and Selma Šabanović. Let's Talk About You: Development and Evaluation of an Autonomous Robot to Support Ikigai Reflection in Older Adults. In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. ISSN: 1944-9437. Aug. 2024, pp. 1323–1330. DOI: 10.1109/RO-MAN60168.2024.10731264. URL: https://ieeexplore.ieee.org/abstract/document/10731264 (visited on 04/03/2025).

[109] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models.

arXiv:2106.09685 [cs]. Oct. 2021. DOI: 10.48550/arXiv.2106.09685. URL: http://arxiv.org/abs/2106.09685 (visited on 04/28/2025).

[110] Yupeng Hu, Wenxin Kuang, Zheng Qin, Kenli Li, Jiliang Zhang, Yansong Gao, Wenjia Li, and Keqin Li. Artificial Intelligence Security: Threats and Countermeasures. In: *ACM Comput. Surv.* 55.1 (Nov. 2021), 20:1–20:36. ISSN: 0360-0300. DOI: 10.1145/3487890. URL: https://dl.acm.org/doi/10.1145/3487890 (visited on 04/28/2025).

[111] Jindan Huang, Reuben M. Aronson, and Elaine Schaertl Short. Modeling Variation in Human Feedback with User Inputs: An Exploratory Methodology. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 303–312. ISBN: 9798400703225. DOI: 10.1145/3610977.3634925. URL: https://dl.acm.org/doi/10.1145/3610977.3634925 (visited on 10/16/2024).

[112] Jindan Huang, Isaac Sheidlower, Reuben M. Aronson, and Elaine Schaertl Short. On the Effect of Robot Errors on Human Teaching Dynamics. arXiv:2409.09827. Sept. 2024. DOI: 10.48550/arXiv.2409.09827. URL: http://arxiv.org/abs/2409.09827 (visited on 10/17/2024).

[113] Yizhou Huang, Kevin Xie, Homanga Bharadhwaj, and Florian Shkurti. Continual Model-Based Reinforcement Learning with Hypernetworks. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2021, pp. 799–805. DOI: 10.1109/ICRA48506.2021.9560793. URL: https://ieeexplore.ieee.org/abstract/document/9560793 (visited on 04/27/2025).

[114] Donald Joseph Hejna Iii and Dorsa Sadigh. Few-Shot Preference Learning for Human-in-the-Loop RL. en. In: *Proceedings of The 6th Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Mar. 2023, pp. 2014–2025. URL: https://proceedings.mlr.press/v205/iii23a.html (visited on 04/28/2025).

[115] Alessandro Iucci, Alberto Hata, Ahmad Terra, Rafia Inam, and Iolanda Leite. Explainable Reinforcement Learning for Human-Robot Collaboration. In: *2021 20th International Conference on Advanced Robotics (ICAR)*. Dec. 2021, pp. 927–934. DOI: 10.1109/ICAR53236.2021.9659472. URL: https://ieeexplore.ieee.org/document/9659472/?arnumber=9659472 (visited on 10/17/2024).

[116] Pallavi Jain, Vandana Tripathi, Ravisankar Malladi, and Alex Khang. Data-Driven Artificial Intelligence (AI) Models in the Workforce Development Planning. In: *Designing Workforce Management Systems for Industry 4.0*. Num Pages: 18. CRC Press, 2023. ISBN: 978-1-00-335707-0.

[117] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment. arXiv:1909.12271. Sept. 2019. DOI: 10.48550/arXiv.1909.12271. URL: http://arxiv.org/abs/1909.12271 (visited on 10/17/2024).

[118] Noémie Jaquier, Michael C Welle, Andrej Gams, Kunpeng Yao, Bernardo Fichera, Aude Billard, Aleš Ude, Tamim Asfour, and Danica Kragic. Transfer learning in robotics: An upcoming breakthrough? A review of promises and challenges. EN. In: *The International Journal of Robotics Research* 44.3 (Mar. 2025). Publisher: SAGE Publications Ltd STM, pp. 465–485. ISSN: 0278-3649. DOI: 10.1177/02783649241273565. URL: https://doi.org/10.1177/02783649241273565 (visited on 04/28/2025).

[119] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. In: *The International Journal of Robotics Research* 37.7 (June 2018). Publisher: SAGE Publications Ltd STM, pp. 717–742. ISSN: 0278-3649. DOI: 10.1177/0278364918776060. URL: https://doi.org/10.1177/0278364918776060 (visited on 08/22/2022).

[120] Shervin Javdani, Siddhartha Srinivasa, and Andrew Bagnell. Shared Autonomy via Hindsight Optimization. en. In: *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, July 2015. ISBN: 978-0-9923747-1-6. DOI: 10.15607/RSS.2015.XI.032. URL: http://www.roboticsproceedings.org/rss11/p32.pdf (visited on 12/01/2022).

[121] Hong Jun Jeon, Dylan P. Losey, and Dorsa Sadigh. Shared Autonomy with Learned Latent Actions. Number: arXiv:2005.03210 arXiv:2005.03210 [cs]. May 2020. URL: http://arxiv.org/abs/2005.03210 (visited on 06/23/2022).

[122] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. en. In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 24678–24704. URL: https://proceedings.neurips.cc/paper_

files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html (visited on 04/28/2025).

[123] Jiaming Ji et al. AI Alignment: A Comprehensive Survey. arXiv:2310.19852 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2310.19852. URL: http://arxiv.org/abs/2310.19852 (visited on 04/14/2025).

[124] Wanxin Jin, Todd D. Murphey, Zehui Lu, and Shaoshuai Mou. Learning From Human Directional Corrections. In: *IEEE Transactions on Robotics* 39.1 (Feb. 2023), pp. 625–644. ISSN: 1941-0468. DOI: 10.1109/TRO.2022.3190221. URL: https://ieeexplore.ieee.org/abstract/document/9852712 (visited on 04/15/2025).

[125] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement Learning: A Survey. arXiv:cs/9605103. Apr. 1996. DOI: 10.48550/arXiv.cs/9605103. URL: http://arxiv.org/abs/cs/9605103 (visited on 07/19/2023).

[126] Shivaram Kalyanakrishnan and Peter Stone. An Empirical Analysis of Value Function-Based and Policy Search Reinforcement Learning. en. In: (2009).

[127] Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. HAGRID: A Human-LLM Collaborative Dataset for Generative Information-Seeking with Attribution. arXiv:2307.16883 [cs]. July 2023. DOI: 10.48550/arXiv.2307.16883. URL: http://arxiv.org/abs/2307.16883 (visited on 04/28/2025).

[128] Anssi Kanervisto, Christian Scheller, and Ville Hautamäki. Action Space Shaping in Deep Reinforcement Learning. In: *arXiv:2004.00980 [cs]* (May 2020). URL: http://arxiv.org/abs/2004.00980 (visited on 08/30/2021).

[129] Parham M. Kebria, Hamid Abdi, Mohsen Moradi Dalvand, Abbas Khosravi, and Saeid Nahavandi. Control Methods for Internet-Based Teleoperation Systems: A Review. In: *IEEE Transactions on Human-Machine Systems* 49.1 (Feb. 2019). Conference Name: IEEE Transactions on Human-Machine Systems, pp. 32–46. ISSN: 2168-2305. DOI: 10.1109/THMS.2018.2878815.

[130] David Kent, Carl Saldanha, and Sonia Chernova. A Comparison of Remote Robot Teleoperation Interfaces for General Object Manipulation. In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '17. New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 371–379. ISBN: 978-1-4503-4336-7. DOI:

10.1145/2909824.3020249. URL: https://doi.org/10.1145/2909824.3020249 (visited on 08/22/2022).

[131] Taylor Kessler Faulkner, Reymundo A. Gutierrez, Elaine Schaertl Short, Guy Hoffman, and Andrea L. Thomaz. Active Attention-Modified Policy Shaping: Socially Interactive Agents Track. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2019, pp. 728–736. ISBN: 978-1-4503-6309-9. (Visited on 03/05/2025).

[132] Alexander Khazatsky et al. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. arXiv:2403.12945 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2403.12945. URL: http://arxiv.org/abs/2403.12945 (visited on 05/21/2024).

[133] Joseph Kim, Christian Muise, Ankit Jayesh Shah, Shubham Agarwal, and Julie A. Shah. Bayesian Inference of Linear Temporal Logic Specifications for Contrastive Explanations. en. In: *MIT web domain* (2019). Accepted: 2021-11-04T13:51:00Z Publisher: International Joint Conferences on Artificial Intelligence. URL: https://dspace.mit.edu/handle/1721.1/137327 (visited on 04/27/2025).

[134] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. arXiv:2406.09246. Sept. 2024. DOI: 10.48550/arXiv.2406.09246. URL: http://arxiv.org/abs/2406.09246 (visited on 10/17/2024).

[135] Su Kyoung Kim, Elsa Andrea Kirchner, Arne Stefes, and Frank Kirchner. Intrinsic interactive reinforcement learning – Using error-related potentials for real world human-robot interaction. en. In: *Scientific Reports* 7.1 (Dec. 2017), p. 17562. ISSN: 2045-2322. DOI: 10.1038/s41598-017-17682-7. URL: http://www.nature.com/articles/s41598-017-17682-7 (visited on 02/22/2021).

[136] Kinova Robotics. URL: www.kinovarobotics.com.

[137] W. Bradley Knox, Stephane Hatgis-Kessell, Sigurdur Orn Adalgeirsson, Serena Booth, Anca Dragan, Peter Stone, and Scott Niekum. Learning Optimal Advantage from Preferences and

Mistaking It for Reward. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.9 (Mar. 2024). Number: 9, pp. 10066–10073. ISSN: 2374-3468. DOI: 10.1609/aaai.v38i9.28870. URL: https://ojs.aaai.org/index.php/AAAI/article/view/28870 (visited on 04/13/2025).

[138] W. Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: the TAMER framework. en. In: *Proceedings of the fifth international conference on Knowledge capture - K-CAP '09*. Redondo Beach, California, USA: ACM Press, 2009, p. 9. ISBN: 978-1-60558-658-8. DOI: 10.1145/1597735.1597738. URL: http://portal.acm.org/citation.cfm?doid=1597735.1597738 (visited on 09/13/2020).

[139] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. Sept. 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727.

[140] Mika Koverola, Anton Kunnari, Jukka Sundvall, and Michael Laakasuo. General Attitudes Towards Robots Scale (GAToRS): A New Instrument for Social Surveys. en. In: *International Journal of Social Robotics* 14.7 (Sept. 2022), pp. 1559–1581. ISSN: 1875-4805. DOI: 10.1007/s12369-022-00880-3. URL: https://doi.org/10.1007/s12369-022-00880-3 (visited on 12/09/2024).

[141] Hadas Kress-Gazit, Kerstin Eder, Guy Hoffman, Henny Admoni, Brenna Argall, Ruediger Ehlers, Christoffer Heckman, Nils Jansen, Ross Knepper, Jan Křetínský, Shelly Levy-Tzedek, Jamy Li, Todd Murphey, Laurel Riek, and Dorsa Sadigh. Formalizing and Guaranteeing* Human-Robot Interaction. In: *Communications of the ACM* 64.9 (Sept. 2021). arXiv:2006.16732 [cs], pp. 78–84. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3433637. URL: http://arxiv.org/abs/2006.16732 (visited on 04/13/2025).

[142] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. In: *arXiv:1906.00949 [cs, stat]* (Nov. 2019). arXiv: 1906.00949. URL: http://arxiv.org/abs/1906.00949 (visited on 02/25/2022).

[143] Li-Cheng Lan, Huan Zhang, and Cho-Jui Hsieh. Can Agents Run Relay Race with Strangers? Generalization of RL to Out-of-Distribution Trajectories. en. In: Sept. 2022. URL: https://openreview.net/forum?id=ipflrGaf7ry (visited on 07/24/2023).

[144] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. en. In: *Nature* 521.7553 (May 2015). Publisher: Nature Publishing Group, pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 03/04/2025).

[145] Jin Sol Lee, Youngjib Ham, Hangue Park, and Jeonghee Kim. Challenges, tasks, and opportunities in teleoperation of excavator toward human-in-the-loop construction automation. In: *Automation in Construction* 135 (Mar. 2022), p. 104119. ISSN: 0926-5805. DOI: 10.1016/j.autcon.2021.104119. URL: https://www.sciencedirect.com/science/article/pii/S0926580521005707 (visited on 03/05/2025).

[146] Jie Lei, Mingli Song, Ze-Nian Li, and Chun Chen. Whole-body humanoid robot imitation with pose similarity evaluation. In: *Signal Processing* 108 (Mar. 2015), pp. 136–146. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2014.08.030. URL: https://www.sciencedirect.com/science/article/pii/S0165168414003946 (visited on 04/27/2025).

[147] Sergey Levine and Vladlen Koltun. Guided Policy Search. en. In: *Proceedings of the 30th International Conference on Machine Learning*. ISSN: 1938-7228. PMLR, May 2013, pp. 1–9. URL: https://proceedings.mlr.press/v28/levine13.html (visited on 05/09/2024).

[148] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. arXiv:1603.02199 [cs]. Aug. 2016. DOI: 10.48550/arXiv.1603.02199. URL: http://arxiv.org/abs/1603.02199 (visited on 03/04/2025).

[149] Boyi Li, Philipp Wu, Pieter Abbeel, and Jitendra Malik. Interactive Task Planning with Language Models. arXiv:2310.10645 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2310.10645. URL: http://arxiv.org/abs/2310.10645 (visited on 04/27/2025).

[150] Xiao-Hui Li, Caleb Chen Cao, Yuhan Shi, Wei Bai, Han Gao, Luyu Qiu, Cong Wang, Yuanyuan Gao, Shenjia Zhang, Xun Xue, and Lei Chen. A Survey of Data-Driven and Knowledge-Aware eXplainable AI. In: *IEEE Transactions on Knowledge and Data Engineering* 34.1 (Jan. 2022), pp. 29–49. ISSN: 1558-2191. DOI: 10.1109/TKDE.2020.2983930. URL: https://ieeexplore.ieee.org/abstract/document/9050829 (visited on 04/28/2025).

[151] Jacky Liang et al. Learning to Learn Faster from Human Feedback with Language Model Predictive Control. en. In: *Robotics: Science and Systems XX*. Robotics: Science and Systems

Foundation, July 2024. ISBN: 9798990284807. DOI: 10.15607/RSS.2024.XX.125. URL: http://www.roboticsproceedings.org/rss20/p125.pdf (visited on 04/28/2025).

[152] Christina Lichtenthäler and Alexandra Kirsch. Legibility of Robot Behavior : A Literature Review. Apr. 2016. URL: https://hal.science/hal-01306977 (visited on 05/24/2024).

[153] Bo Liu, Xuesu Xiao, and Peter Stone. A Lifelong Learning Approach to Mobile Robot Navigation. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 1090–1096. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3056373. URL: https://ieeexplore.ieee.org/abstract/document/9345478 (visited on 04/27/2025).

[154] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., May 2024, pp. 44776–44791. (Visited on 10/17/2024).

[155] 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. en. arXiv:2211.08416 [cs]. July 2023. URL: http://arxiv.org/abs/2211.08416 (visited on 05/14/2024).

[156] Jason Xinyu Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. Grounding complex natural language commands for temporal tasks in unseen environments. In: *Conference on Robot Learning*. PMLR, 2023, pp. 1084–1110. URL: https://proceedings.mlr.press/v229/liu23d.html (visited on 04/27/2025).

[157] Yueyue Liu, Zhijun Li, Huaping Liu, and Zhen Kan. Skill transfer learning for autonomous robots and human–robot cooperation: A survey. In: *Robotics and Autonomous Systems* 128 (June 2020), p. 103515. ISSN: 0921-8890. DOI: 10.1016/j.robot.2020.103515. URL: https://www.sciencedirect.com/science/article/pii/S0921889019309972 (visited on 04/28/2025).

[158] Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. en. In: Aug. 2023. URL: https://openreview.net/forum?id=8yTS_nAILxt (visited on 10/17/2024).

[159] Björn Lütjens, Michael Everett, and Jonathan P. How. Safe Reinforcement Learning With Model Uncertainty Estimates. In: *2019 International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2019, pp. 8662–8668. DOI: 10.1109/ICRA.2019.8793611.

[160] 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:2310.12931 [cs]. Apr. 2024. DOI: 10.48550/arXiv.2310.12931. URL: http://arxiv.org/abs/2310.12931 (visited on 05/07/2024).

[161] James MacGlashan, Mark K. Ho, Robert Loftin, Bei Peng, David Roberts, Matthew E. Taylor, and Michael L. Littman. Interactive Learning from Policy-Dependent Human Feedback. In: *arXiv:1701.06049 [cs]* (Jan. 2017). URL: http://arxiv.org/abs/1701.06049 (visited on 10/05/2020).

[162] Magnuson Moss Warranty-Federal Trade Commission Improvements Act. en. July 2013. URL: https://www.ftc.gov/legal-library/browse/statutes/magnuson-moss-warranty-federal-trade-commission-improvements-act (visited on 04/28/2025).

[163] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient Multi-task Fine-tuning for Transformers via Shared Hypernetworks. arXiv:2106.04489 [cs]. June 2021. DOI: 10.48550/arXiv.2106.04489. URL: http://arxiv.org/abs/2106.04489 (visited on 04/27/2025).

[164] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In: *arXiv preprint arXiv:2108.03298*. 2021.

[165] Ji-Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. The state of user-centered design practice. In: *Commun. ACM* 48.3 (Mar. 2005), pp. 105–109. ISSN: 0001-0782. DOI: 10.1145/1047671.1047677. URL: https://dl.acm.org/doi/10.1145/1047671.1047677 (visited on 03/31/2025).

[166] P. Marayong, Ming Li, A.M. Okamura, and G.D. Hager. Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Vol. 2. ISSN: 1050-4729. Sept.

2003, 1954–1959 vol.2. DOI: 10.1109/ROBOT.2003.1241880. URL: https://ieeexplore.ieee.org/document/1241880 (visited on 03/05/2025).

[167] Gabriel B. Margolis and Pulkit Agrawal. Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior. arXiv:2212.03238 [cs, eess]. Dec. 2022. URL: http://arxiv.org/abs/2212.03238 (visited on 09/11/2023).

[168] Daniel Marta, Christian Pek, Gaspar I. Melsión, Jana Tumova, and Iolanda Leite. Human-Feedback Shield Synthesis for Perceived Safety in Deep Reinforcement Learning. In: *IEEE Robotics and Automation Letters* 7.1 (Jan. 2022). Conference Name: IEEE Robotics and Automation Letters, pp. 406–413. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3128237. URL: https://ieeexplore.ieee.org/document/9616473 (visited on 12/27/2023).

[169] Marcus Mast, Michael Burmester, Katja Krüger, Sascha Fatikow, Georg Arbeiter, Birgit Graf, Gernot Kronreif, Lucia Pigini, David Facal, and Renxi Qiu. User-centered design of a dynamic-autonomy remote interaction concept for manipulation-capable robots to assist elderly people in the home. In: *J. Hum.-Robot Interact.* 1.1 (July 2012), pp. 96–118. DOI: 10.5898/JHRI.1.1.Mast. URL: https://dl.acm.org/doi/10.5898/JHRI.1.1.Mast (visited on 04/03/2025).

[170] Jan Matas, Stephen James, and Andrew J. Davison. Sim-to-Real Reinforcement Learning for Deformable Object Manipulation. en. In: *Proceedings of The 2nd Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Oct. 2018, pp. 734–743. URL: https://proceedings.mlr.press/v87/matas18a.html (visited on 03/04/2025).

[171] Amirhossein H. Memar and Ehsan T. Esfahani. Objective Assessment of Human Workload in Physical Human-robot Cooperation Using Brain Monitoring. In: *J. Hum.-Robot Interact.* 9.2 (Dec. 2019), 13:1–13:21. DOI: 10.1145/3368854. URL: https://dl.acm.org/doi/10.1145/3368854 (visited on 04/28/2025).

[172] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. Explainable Reinforcement Learning: A Survey and Comparative Review. In: *ACM Comput. Surv.* 56.7 (Apr. 2024), 168:1–168:36. ISSN: 0360-0300. DOI: 10.1145/3616864. URL: https://dl.acm.org/doi/10.1145/3616864 (visited on 04/28/2025).

[173] Cristian Millian-Arias, Bruno Fernandes, Francisco Cruz, Richard Dazeley, and Sergio Fernandes. A Robust Approach for Continuous Interactive Reinforcement Learning. DOI: 10.

1145/3406499.3418769. URL: https://dl-acm-org.ezproxy.library.tufts.edu/doi/epdf/10.1145/3406499.3418769 (visited on 02/22/2021).

[174] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs]. Dec. 2013. DOI: 10.48550/arXiv.1312.5602. URL: http://arxiv.org/abs/1312.5602 (visited on 03/04/2025).

[175] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. en. In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14236. URL: http://www.nature.com/articles/nature14236 (visited on 09/29/2020).

[176] Nina M. Moorman, Nakul Gopalan, Aman Singh, Erin Botti, Mariah Schrum, Chuxuan Yang, Lakshmi Seelam, and Matthew Gombolay. Investigating the Impact of Experience on a User's Ability to Perform Hierarchical Abstraction. In: *Proceedings of Robotics: Science and Systems*. Daegu, Republic of Korea, July 2023. DOI: 10.15607/RSS.2023.XIX.004.

[177] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. en. arXiv:1504.04909 [cs, q-bio]. Apr. 2015. URL: http://arxiv.org/abs/1504.04909 (visited on 05/09/2024).

[178] Christopher Mower, Joao Moura, and Sethu Vijayakumar. Skill-based Shared Control. en. In: *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, July 2021. ISBN: 978-0-9923747-7-8. DOI: 10.15607/RSS.2021.XVII.028. URL: http://www.roboticsproceedings.org/rss17/p028.pdf (visited on 09/24/2023).

[179] Fabio Muratore, Felix Treede, Michael Gienger, and Jan Peters. Domain Randomization for Simulation-Based Policy Optimization with Transferability Assessment. en. In: *Proceedings of The 2nd Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Oct. 2018, pp. 700–713. URL: https://proceedings.mlr.press/v87/muratore18a.html (visited on 09/24/2023).

[180] Vivek Myers, Erdem Biyik, Nima Anari, and Dorsa Sadigh. Learning Multimodal Rewards from Rankings. en. In: *Proceedings of the 5th Conference on Robot Learning*. ISSN: 2640-3498.

PMLR, Jan. 2022, pp. 342–352. URL: https://proceedings.mlr.press/v164/myers22a.html (visited on 08/22/2022).

[181] Vivek Myers, Chunyuan Zheng, Oier Mees, Kuan Fang, and Sergey Levine. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. en. In: Sept. 2024. URL: https://openreview.net/forum?id=qUSa3F79am (visited on 10/17/2024).

[182] Subash Neupane, Shaswata Mitra, Ivan A. Fernandez, Swayamjit Saha, Sudip Mittal, Jingdao Chen, Nisha Pillai, and Shahram Rahimi. Security Considerations in AI-Robotics: A Survey of Current Methods, Challenges, and Opportunities. In: *IEEE Access* 12 (2024), pp. 22072–22097. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3363657. URL: https://ieeexplore.ieee.org/abstract/document/10423748 (visited on 04/28/2025).

[183] Benjamin A. Newman, Reuben M. Aronson, Siddhartha S. Srinivasa, Kris Kitani, and Henny Admoni. HARMONIC: A multimodal dataset of assistive human–robot collaboration. en. In: *The International Journal of Robotics Research* 41.1 (Jan. 2022). Publisher: SAGE Publications Ltd STM, pp. 3–11. ISSN: 0278-3649. DOI: 10.1177/02783649211050677. URL: https://doi.org/10.1177/02783649211050677 (visited on 01/28/2023).

[184] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In: *In Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann, 1999, pp. 278–287.

[185] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. arXiv:2102.09672 [cs, stat]. Feb. 2021. DOI: 10.48550/arXiv.2102.09672. URL: http://arxiv.org/abs/2102.09672 (visited on 05/13/2024).

[186] Monica N. Nicolescu and Maja J. Mataric. Natural methods for robot task learning: instructive demonstrations, generalization and practice. In: *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. AAMAS '03. New York, NY, USA: Association for Computing Machinery, July 2003, pp. 241–248. ISBN: 978-1-58113-683-8. DOI: 10.1145/860575.860614. URL: https://dl.acm.org/doi/10.1145/860575.860614 (visited on 04/26/2025).

[187] Abby O'Neill et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration0. In: *2024 IEEE International Conference on Robotics*

and Automation (ICRA). May 2024, pp. 6892–6903. DOI: 10.1109/ICRA57147.2024.10611477. URL: https://ieeexplore.ieee.org/document/10611477 (visited on 10/17/2024).

[188] OpenAI et al. GPT-4 Technical Report. arXiv:2303.08774 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2303.08774. URL: http://arxiv.org/abs/2303.08774 (visited on 05/07/2024).

[189] Valerio Ortenzi, Akansel Cosgun, Tommaso Pardi, Wesley P. Chan, Elizabeth Croft, and Dana Kulić. Object Handovers: A Review for Robotics. In: *IEEE Transactions on Robotics* 37.6 (Dec. 2021), pp. 1855–1873. ISSN: 1941-0468. DOI: 10.1109/TRO.2021.3075365. URL: https://ieeexplore.ieee.org/abstract/document/9444288 (visited on 04/28/2025).

[190] Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information. en. In: *Neural Networks* 152 (Aug. 2022), pp. 90–104. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2022.04.009. URL: https://www.sciencedirect.com/science/article/pii/S0893608022001393 (visited on 01/02/2023).

[191] Błażej Osiński, Adam Jakubowski, Paweł Ziecina, Piotr Miłoś, Christopher Galias, Silviu Homoceanu, and Henryk Michalewski. Simulation-Based Reinforcement Learning for Real-World Autonomous Driving. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2020, pp. 6411–6418. DOI: 10.1109/ICRA40945.2020.9196730. URL: https://ieeexplore.ieee.org/abstract/document/9196730?casa_token=Vm-rgJU6AloAAAAA:URquLkIIAdXNCk_Tn81Uk5zC8cYxMDAIufCm050NAIqreSFIHsvV2-yPoq_fXAkZX3Et2HhM1tI (visited on 03/04/2025).

[192] Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. en. In: Sept. 2019. URL: https://openreview.net/forum?id=SJgwNerKvB (visited on 04/27/2025).

[193] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. en. In: *Advances in Neural Information Processing Systems* 35 (Dec. 2022), pp. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html (visited on 04/13/2025).

[194] Rohan Paleja, Muyleng Ghuy, Nadun Ranawaka Arachchige, Reed Jensen, and Matthew Gombolay. The Utility of Explainable AI in Ad Hoc Human-Machine Teaming. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 610–623. URL: https://proceedings.neurips.cc/paper/2021/hash/05d74c48b5b30514d8e9bd60320fc8f6-Abstract.html (visited on 07/24/2023).

[195] Jiayi Pan, Glen Chou, and Dmitry Berenson. Data-Efficient Learning of Natural Language to Linear Temporal Logic Translators for Robot Task Specification. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. May 2023, pp. 11554–11561. DOI: 10.1109/ICRA48891.2023.10161125. URL: https://ieeexplore.ieee.org/abstract/document/10161125 (visited on 04/27/2025).

[196] Ravi Pandya, Michelle Zhao, Changliu Liu, Reid Simmons, and Henny Admoni. Multi-Agent Strategy Explanations for Human-Robot Collaboration. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 17351–17357. DOI: 10.1109/ICRA57147.2024.10610720. URL: https://ieeexplore.ieee.org/abstract/document/10610720 (visited on 10/17/2024).

[197] Andrea Papenmeier, Dagmar Kern, Daniel Hienert, Yvonne Kammerer, and Christin Seifert. How Accurate Does It Feel? – Human Perception of Different Types of Classification Mistakes. en. In: *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–13. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3501915. URL: https://dl.acm.org/doi/10.1145/3491102.3501915 (visited on 01/22/2025).

[198] Hae Won Park, Ishaan Grover, Samuel Spaulding, Louis Gomez, and Cynthia Breazeal. A Model-Free Affective Reinforcement Learning Approach to Personalization of an Autonomous Social Robot Companion for Early Literacy Education. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019). Number: 01, pp. 687–694. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.3301687. URL: https://ojs.aaai.org/index.php/AAAI/article/view/3846 (visited on 04/14/2025).

[199] Carolina Passenberg, Angelika Peer, and Martin Buss. A survey of environment-, operator-, and task-adapted controllers for teleoperation systems. en. In: *Mechatronics* 20.7 (Oct. 2010), pp. 787–801. ISSN: 09574158. DOI: 10.1016/j.mechatronics.2010.04.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957415810000735 (visited on 08/22/2022).

[200] Benjamin Pitzer, Michael Styer, Christian Bersch, Charles DuHadway, and Jan Becker. Towards perceptual shared autonomy for robotic mobile manipulation. In: *2011 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. May 2011, pp. 6245–6251. DOI: 10.1109/ICRA.2011.5980259.

[201] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter Space Noise for Exploration. arXiv:1706.01905 [cs, stat]. Jan. 2018. DOI: 10.48550/arXiv.1706.01905. URL: http://arxiv.org/abs/1706.01905 (visited on 05/09/2024).

[202] Elisa Prati, Margherita Peruzzini, Marcello Pellicciari, and Roberto Raffaeli. How to include User eXperience in the design of Human-Robot Interaction. In: *Robotics and Computer-Integrated Manufacturing* 68 (Apr. 2021), p. 102072. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2020.102072. URL: https://www.sciencedirect.com/science/article/pii/S0736584520302805 (visited on 04/03/2025).

[203] Matthew S. Prewett, Ryan C. Johnson, Kristin N. Saboe, Linda R. Elliott, and Michael D. Coovert. Managing workload in human–robot interaction: A review of empirical studies. In: *Computers in Human Behavior*. Advancing Educational Research on Computer-supported Collaborative Learning (CSCL) through the use of gStudy CSCL Tools 26.5 (Sept. 2010), pp. 840–856. ISSN: 0747-5632. DOI: 10.1016/j.chb.2010.03.010. URL: https://www.sciencedirect.com/science/article/pii/S0747563210000506 (visited on 04/28/2025).

[204] Prolific. URL: www.prolific.com.

[205] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. DexMV: Imitation Learning for00a0Dexterous Manipulation from00a0Human Videos. en. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Cham: Springer Nature Switzerland, 2022, pp. 570–587. ISBN: 978-3-031-19842-7. DOI: 10.1007/978-3-031-19842-7_33.

[206] Marco Ramacciotti, Mario Milazzo, Fabio Leoni, Stefano Roccella, and Cesare Stefanini. A novel shared control algorithm for industrial robots. In: *International Journal of Advanced Robotic Systems* 13.6 (Dec. 2016). Publisher: SAGE Publications, p. 1729881416682701. ISSN: 1729-8806. DOI: 10.1177/1729881416682701. URL: https://doi.org/10.1177/1729881416682701 (visited on 03/05/2025).

[207] Ellis Ratner, Dylan Hadfield-Menell, and Anca D. Dragan. Simplifying Reward Design through Divide-and-Conquer. Number: arXiv:1806.02501 arXiv:1806.02501 [cs]. June 2018. URL: http://arxiv.org/abs/1806.02501 (visited on 06/22/2022).

[208] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent Advances in Robot Learning from Demonstration. en. In: *Annual Review of Control, Robotics, and Autonomous Systems* 3.Volume 3, 2020 (May 2020). Publisher: Annual Reviews, pp. 297–330. ISSN: 2573-5144. DOI: 10.1146/annurev-control-100819-063206. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-100819-063206 (visited on 10/07/2024).

[209] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Shared Autonomy via Deep Reinforcement Learning. Number: arXiv:1802.01744 arXiv:1802.01744 [cs]. May 2018. URL: http://arxiv.org/abs/1802.01744 (visited on 06/23/2022).

[210] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal Diffusion Transformer: Learning Versatile Behavior from Multimodal Goals. In: *Proceedings of Robotics: Science and Systems*. Delft, Netherlands, July 2024. DOI: 10.15607/RSS.2024.XX.121.

[211] Alessandro Roncone, Olivier Mangin, and Brian Scassellati. Transparent role assignment and task allocation in human robot collaboration. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. May 2017, pp. 1014–1021. DOI: 10.1109/ICRA.2017.7989122. URL: https://ieeexplore.ieee.org/abstract/document/7989122 (visited on 04/14/2025).

[212] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597 [cs]. May 2015. DOI: 10.48550/arXiv.1505.04597. URL: http://arxiv.org/abs/1505.04597 (visited on 05/13/2024).

[213] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep One-Class Classification. en. In: *Proceedings of the 35th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2018, pp. 4393–4402. URL: https://proceedings.mlr.press/v80/ruff18a.html (visited on 08/06/2023).

[214] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models. en. In: 2024, pp. 6527–6536. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Ruiz_HyperDreamBooth_HyperNetworks_for_Fast_Personalization_of_Text-to-Image_Models_CVPR_2024_paper.html (visited on 04/27/2025).

[215] Gery W. Ryan and H. Russell Bernard. Techniques to Identify Themes. en. In: *Field Methods* 15.1 (Feb. 2003). Publisher: SAGE Publications Inc, pp. 85–109. ISSN: 1525-822X. DOI: 10.1177/1525822X02239569. URL: https://doi.org/10.1177/1525822X02239569 (visited on 01/20/2025).

[216] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active Preference-Based Learning of Reward Functions. en. 2017. ISBN: 978-0-9923747-3-0. DOI: 10.15607/rss.2017.xiii.053. URL: https://escholarship.org/uc/item/88k894w7 (visited on 04/13/2025).

[217] Tatsuya Sakai and Takayuki Nagai. Explainable autonomous robots: a survey and perspective. In: *Advanced Robotics* 36.5-6 (Mar. 2022). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/016918 pp. 219–238. ISSN: 0169-1864. DOI: 10.1080/01691864.2022.2029720. URL: https://doi.org/10.1080/01691864.2022.2029720 (visited on 07/24/2023).

[218] Maha Salem, Gabriella Lakatos, Farshid Amirabdollahian, and Kerstin Dautenhahn. Would You Trust a (Faulty) Robot?: Effects of Error, Task Type and Personality on Human-Robot Cooperation and Trust. en. In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. Portland Oregon USA: ACM, Mar. 2015, pp. 141–148. ISBN: 978-1-4503-2883-8. DOI: 10.1145/2696454.2696497. URL: https://dl.acm.org/doi/10.1145/2696454.2696497 (visited on 01/22/2025).

[219] Erica Salvato, Gianfranco Fenu, Eric Medvet, and Felice Andrea Pellegrino. Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning. In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 153171–153187. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3126658. URL: https://ieeexplore.ieee.org/abstract/document/9606868 (visited on 03/04/2025).

[220] Elad Sarafian, Shai Keynan, and Sarit Kraus. Recomposing the Reinforcement Learning Building Blocks with Hypernetworks. en. In: *Proceedings of the 38th International Conference*

on Machine Learning. ISSN: 2640-3498. PMLR, July 2021, pp. 9301–9312. URL: https://proceedings.mlr.press/v139/sarafian21a.html (visited on 04/27/2025).

[221] Fumihiro Sasaki and Ryota Yamashina. Behavioral Cloning from Noisy Demonstrations. en. In: Oct. 2020. URL: https://openreview.net/forum?id=zrT3HcsWSAt (visited on 04/26/2025).

[222] Sven R. Schepp, Jakob Thumm, Stefan B. Liu, and Matthias Althoff. SaRA: A Tool for Safe Human-Robot Coexistence and Collaboration through Reachability Analysis. In: *2022 International Conference on Robotics and Automation (ICRA)*. May 2022, pp. 4312–4317. DOI: 10.1109/ICRA46639.2022.9811952. URL: https://ieeexplore.ieee.org/abstract/document/9811952 (visited on 04/28/2025).

[223] Matthias Scheutz, Paul Schermerhorn, and James Kramer. The utility of affect expression in natural language interactions in joint human-robot tasks. en. In: *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. Salt Lake City Utah USA: ACM, Mar. 2006, pp. 226–233. ISBN: 978-1-59593-294-5. DOI: 10.1145/1121241.1121281. URL: https://dl.acm.org/doi/10.1145/1121241.1121281 (visited on 04/27/2025).

[224] Mariah L. Schrum, Erin Hedlund-Botti, Nina Moorman, and Matthew C. Gombolay. MIND MELD: Personalized Meta-Learning for Robot-Centric Imitation Learning. In: *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '22. Sapporo, Hokkaido, Japan: IEEE Press, Mar. 2022, pp. 157–165. (Visited on 07/19/2023).

[225] Mariah L. Schrum, Michael Johnson, Muyleng Ghuy, and Matthew C. Gombolay. Four Years in Review: Statistical Practices of Likert Scales in Human-Robot Interaction Studies. en. In: *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. Cambridge United Kingdom: ACM, Mar. 2020, pp. 43–52. ISBN: 978-1-4503-7057-8. DOI: 10.1145/3371382.3380739. URL: https://dl.acm.org/doi/10.1145/3371382.3380739 (visited on 12/09/2024).

[226] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. In: *arXiv:1707.06347 [cs]* (Aug. 2017). URL: http://arxiv.org/abs/1707.06347 (visited on 02/25/2021).

[227] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. en. In: *Data Mining*

and Knowledge Discovery 38.5 (Sept. 2024), pp. 3043–3101. ISSN: 1573-756X. DOI: 10.1007/s10618-022-00867-8. URL: https://doi.org/10.1007/s10618-022-00867-8 (visited on 04/28/2025).

[228] Mario Selvaggio, Marco Cognetti, Stefanos Nikolaidis, Serena Ivaldi, and Bruno Siciliano. Autonomy in Physical Human-Robot Interaction: A Brief Survey. In: *IEEE Robotics and Automation Letters* 6.4 (Oct. 2021). Conference Name: IEEE Robotics and Automation Letters, pp. 7989–7996. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3100603.

[229] Stela H. Seo, Denise Geiskkovitch, Masayuki Nakane, Corey King, and James E. Young. Poor Thing! Would You Feel Sorry for a Simulated Robot? A comparison of empathy toward a physical and a simulated robot. In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. HRI '15. New York, NY, USA: Association for Computing Machinery, Mar. 2015, pp. 125–132. ISBN: 978-1-4503-2883-8. DOI: 10.1145/2696454.2696471. URL: https://dl.acm.org/doi/10.1145/2696454.2696471 (visited on 01/05/2025).

[230] Rossitza Setchi, Maryam Banitalebi Dehkordi, and Juwairiya Siraj Khan. Explainable Robotics in Human-Robot Interactions. In: *Procedia Computer Science*. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020 176 (Jan. 2020), pp. 3057–3066. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.09.198. URL: https://www.sciencedirect.com/science/article/pii/S1877050920321001 (visited on 05/14/2024).

[231] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On Bringing Robots Home. arXiv:2311.16098. Nov. 2023. DOI: 10.48550/arXiv.2311.16098. URL: http://arxiv.org/abs/2311.16098 (visited on 10/17/2024).

[232] Ankit Shah, Pritish Kamath, Julie A Shah, and Shen Li. Bayesian Inference of Temporal Task Specifications from Demonstrations. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/13168e6a2e6c84b4b7de9390c0ef5ec5-Abstract.html (visited on 04/27/2025).

[233] Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. TOMATO: Assessing Visual Temporal Reasoning Capabilities in Multimodal Foundation Models. en. In: Oct. 2024. URL: https://openreview.net/forum?id=fCi4o83Mfs (visited on 04/28/2025).

[234] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. VideoDex: Learning Dexterity from Internet Videos. en. In: *Proceedings of The 6th Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Mar. 2023, pp. 654–665. URL: https://proceedings.mlr.press/v205/shaw23a.html (visited on 04/26/2025).

[235] Isaac Sheidlower, Reuben Aronson, and Elaine Short. Modifying RL Policies with Imagined Actions: How Predictable Policies Can Enable Users to Perform Novel Tasks. en. In: *Proceedings of the AAAI Symposium Series* 2.1 (2023). Number: 1, pp. 192–197. ISSN: 2994-4317. DOI: 10.1609/aaaiss.v2i1.27670. URL: https://ojs.aaai.org/index.php/AAAI-SS/article/view/27670 (visited on 06/03/2024).

[236] Isaac Sheidlower, Emma Bethel, Douglas Lilly, Reuben M. Aronson, and Elaine Schaertl Short. Imagining In-distribution States: How Predictable Robot Behavior Can Enable User Control Over Learned Policies. In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. ISSN: 1944-9437. Aug. 2024, pp. 1308–1315. DOI: 10.1109/RO-MAN60168.2024.10731233. URL: https://ieeexplore.ieee.org/abstract/document/10731233 (visited on 04/07/2025).

[237] Isaac Sheidlower, Allison Moore, and Elaine Short. Keeping Humans in the Loop: Teaching via Feedback in Continuous Action Space Environments. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2022, pp. 863–870. DOI: 10.1109/IROS47612.2022.9982282.

[238] Isaac Sheidlower, Mavis Murdock, Emma Bethel, Reuben M. Aronson, and Elaine Schaertl Short. Online Behavior Modification for Expressive User Control of RL-Trained Robots. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 639–648. ISBN: 9798400703225. DOI: 10.1145/3610977.3634947. URL: https://dl.acm.org/doi/10.1145/3610977.3634947 (visited on 05/22/2024).

[239] Isaac Sheidlower, Elaine Schaertl Short, and Allison Moore. Environment Guided Interactive Reinforcement Learning: Learning from Binary Feedback in High-Dimensional Robot Task Environments. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2022, pp. 1726–1728. ISBN: 978-1-4503-9213-6. (Visited on 11/20/2022).

[240] Isaac S. Sheidlower, Reuben M. Aronson, and Elaine Short. Towards Interpretable Foundation Models of Robot Behavior: A Task Specific Policy Generation Approach. en. In: Aug. 2024. URL: https://openreview.net/forum?id=umkqPTUDED (visited on 10/11/2024).

[241] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. en. In: *Proceedings of The 6th Conference on Robot Learning*. ISSN: 2640-3498. PMLR, Mar. 2023, pp. 785–799. URL: https://proceedings.mlr.press/v205/shridhar23a.html (visited on 10/21/2024).

[242] Yash Shukla, Bharat Kesari, Shivam Goel, Robert Wright, and Jivko Sinapov. A Framework for Few-Shot Policy Transfer Through Observation Mapping and Behavior Cloning. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2023, pp. 7104–7110. DOI: 10.1109/IROS55552.2023.10342477. URL: https://ieeexplore.ieee.org/abstract/document/10342477 (visited on 04/28/2025).

[243] Yash Shukla, Abhishek Kulkarni, Robert Wright, Alvaro Velasquez, and Jivko Sinapov. Automaton-Guided Curriculum Generation for Reinforcement Learning Agents. en. In: *Proceedings of the International Conference on Automated Planning and Scheduling* 33 (July 2023), pp. 605–613. ISSN: 2334-0843. DOI: 10.1609/icaps.v33i1.27242. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/27242 (visited on 04/28/2025).

[244] Weiyong Si, Ning Wang, and Chenguang Yang. A review on manipulation skill acquisition through teleoperation-based learning from demonstration. en. In: *Cognitive Computation and Systems* 3.1 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/ccs2.12005, pp. 1–16. ISSN: 2517-7567. DOI: 10.1049/ccs2.12005. URL: https://onlinelibrary.wiley.com/doi/abs/10.1049/ccs2.12005 (visited on 04/26/2025).

[245] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot,

Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. en. In: *Nature* 529.7587 (Jan. 2016). Publisher: Nature Publishing Group, pp. 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961. URL: https://www.nature.com/articles/nature16961 (visited on 04/27/2025).

[246] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and Characterizing Reward Gaming. en. In: *Advances in Neural Information Processing Systems* 35 (Dec. 2022), pp. 9460–9471. URL: https://proceedings.neurips.cc/paper_files/paper/2022/hash/3d719fee332caa23d5038b8a90e81796-Abstract-Conference.html (visited on 03/04/2025).

[247] Matthijs T. J. Spaan. Partially Observable Markov Decision Processes. en. In: *Reinforcement Learning: State-of-the-Art*. Ed. by Marco Wiering and Martijn van Otterlo. Berlin, Heidelberg: Springer, 2012, pp. 387–414. ISBN: 978-3-642-27645-3. DOI: 10.1007/978-3-642-27645-3_12. URL: https://doi.org/10.1007/978-3-642-27645-3_12 (visited on 04/14/2025).

[248] James Staley, Elaine Short, Shivam Goel, and Yash Shukla. Agent-Centric Human Demonstrations Train World Models. en. In: Nov. 2024. URL: https://openreview.net/forum?id=P1RnrmDCnc (visited on 04/26/2025).

[249] Stanford Artificial Intelligence Laboratory et al. Robotic Operating System. May 2018. URL: https://www.ros.org.

[250] Kristen Stubbs, Pamela J. Hinds, and David Wettergreen. Autonomy and Common Ground in Human-Robot Interaction: A Field Study. In: *IEEE Intelligent Systems* 22.2 (Mar. 2007). Conference Name: IEEE Intelligent Systems, pp. 42–50. ISSN: 1941-1294. DOI: 10.1109/MIS.2007.21. URL: https://ieeexplore.ieee.org/abstract/document/4136857 (visited on 10/17/2024).

[251] Bingyi Su, SeHee Jung, Lu Lu, Hanwen Wang, Liwei Qing, and Xu Xu. Exploring the impact of human-robot interaction on workers' mental stress in collaborative assembly tasks. In: *Applied Ergonomics* 116 (Apr. 2024), p. 104224. ISSN: 0003-6870. DOI: 10.1016/j.

apergo.2024.104224. URL: https://www.sciencedirect.com/science/article/pii/S0003687024000012 (visited on 04/28/2025).

[252] Lingfeng Sun, Devesh K. Jha, Chiori Hori, Siddarth Jain, Radu Corcodel, Xinghao Zhu, Masayoshi Tomizuka, and Diego Romeres. Interactive Planning Using Large Language Models for Partially Observable Robotic Tasks. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 14054–14061. DOI: 10.1109/ICRA57147.2024.10610981. URL: https://ieeexplore.ieee.org/abstract/document/10610981 (visited on 04/27/2025).

[253] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-Distribution Detection with Deep Nearest Neighbors. en. In: *Proceedings of the 39th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, June 2022, pp. 20827–20840. URL: https://proceedings.mlr.press/v162/sun22d.html (visited on 07/24/2023).

[254] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction, 2nd ed. Reinforcement learning: An introduction, 2nd ed. Pages: xxii, 526. Cambridge, MA, US: The MIT Press, 2018. ISBN: 978-0-262-03924-6.

[255] Aaquib Tabrez, Shivendra Agrawal, and Bradley Hayes. Explanation-Based Reward Coaching to Improve Human Performance via Reinforcement Learning. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. ISSN: 2167-2148. Mar. 2019, pp. 249–257. DOI: 10.1109/HRI.2019.8673104. URL: https://ieeexplore.ieee.org/abstract/document/8673104 (visited on 10/17/2024).

[256] Matthew E. Taylor, Shimon Whiteson, and Peter Stone. Transfer via inter-task mappings in policy search reinforcement learning. en. In: *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. Honolulu Hawaii: ACM, May 2007, pp. 1–8. ISBN: 978-81-904262-7-5. DOI: 10.1145/1329125.1329170. URL: https://dl.acm.org/doi/10.1145/1329125.1329170 (visited on 05/09/2024).

[257] ALOHA 2 Team et al. ALOHA 2: An Enhanced Low-Cost Hardware for Bimanual Teleoperation. arXiv:2405.02292. Feb. 2024. DOI: 10.48550/arXiv.2405.02292. URL: http://arxiv.org/abs/2405.02292 (visited on 10/17/2024).

[258] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence

Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. arXiv:2405.12213. May 2024. DOI: 10.48550/arXiv.2405.12213. URL: http://arxiv.org/abs/2405.12213 (visited on 10/17/2024).

[259] Bryon Tjanaka, Matthew C. Fontaine, Yulun Zhang, Sam Sommerer, Nathan Dennler, and Stefanos Nikolaidis. pyribs: A bare-bones Python library for quality diversity optimization. Publication Title: GitHub repository. 2021. URL: https://github.com/icaros-usc/pyribs.

[260] TLX @ NASA Ames - Home. URL: https://humansystems.arc.nasa.gov/groups/tlx/ (visited on 01/28/2023).

[261] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. arXiv:1703.06907 [cs]. Mar. 2017. DOI: 10.48550/arXiv.1703.06907. URL: http://arxiv.org/abs/1703.06907 (visited on 09/24/2023).

[262] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral Cloning from Observation. In: *arXiv:1805.01954 [cs]* (May 2018). URL: http://arxiv.org/abs/1805.01954 (visited on 10/19/2020).

[263] Hugo Touvron et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs]. July 2023. DOI: 10.48550/arXiv.2307.09288. URL: http://arxiv.org/abs/2307.09288 (visited on 04/13/2025).

[264] Charles Vanover, Paul Mihas, and Johnny Saldana. Analyzing and Interpreting Qualitative Research: After the Interview. en. Google-Books-ID: 0xIoEAAAQBAJ. SAGE Publications, Apr. 2021. ISBN: 978-1-5443-9588-3.

[265] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (visited on 10/16/2024).

[266] Viswanath Venkatesh, Michael G. Morris, Gordon B. Davis, and Fred D. Davis. User Acceptance of Information Technology: Toward a Unified View. In: *MIS Quarterly* 27.3 (2003). Publisher: Management Information Systems Research Center, University of Minnesota, pp. 425–

478. ISSN: 0276-7783. DOI: 10.2307/30036540. URL: https://www.jstor.org/stable/30036540 (visited on 01/28/2023).

[267] Ngo Anh Vien, Wolfgang Ertel, and Tae Choong Chung. Learning via human feedback in continuous state and action spaces. en. In: *Applied Intelligence* 39.2 (Sept. 2013), pp. 267–278. ISSN: 1573-7497. DOI: 10.1007/s10489-012-0412-6. URL: https://doi.org/10.1007/s10489-012-0412-6 (visited on 02/22/2021).

[268] Dinh-Son Vu, Ulysse Cote Allard, Clement Gosselin, Francois Routhier, Benoit Gosselin, and Alexandre Campeau-Lecours. Intuitive adaptive orientation control of assistive robots for people living with upper limb disabilities. eng. In: *IEEE ... International Conference on Rehabilitation Robotics: [proceedings]* 2017 (July 2017), pp. 795–800. ISSN: 1945-7901. DOI: 10.1109/ICORR.2017.8009345.

[269] Nikola Vulin, Sammy Christen, Stefan Stevsic, and Otmar Hilliges. Improved Learning of Robot Manipulation Tasks via Tactile Intrinsic Motivation. In: *IEEE Robotics and Automation Letters* (2021), pp. 1–1. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3061308. URL: http://arxiv.org/abs/2102.11051 (visited on 02/25/2021).

[270] Joshua Wainer, David J. Feil-seifer, Dylan A. Shell, and Maja J. Mataric. The role of physical embodiment in human-robot interaction. In: *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication.* ISSN: 1944-9437. Sept. 2006, pp. 117–122. DOI: 10.1109/ROMAN.2006.314404. URL: https://ieeexplore.ieee.org/abstract/document/4107795 (visited on 01/05/2025).

[271] Nick Walker, Kevin Weatherwax, Julian Allchin, Leila Takayama, and Maya Cakmak. Human Perceptions of a Curious Robot that Performs Off-Task Actions. en. In: *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction.* Cambridge United Kingdom: ACM, Mar. 2020, pp. 529–538. ISBN: 978-1-4503-6746-2. DOI: 10.1145/3319502.3374821. URL: https://dl.acm.org/doi/10.1145/3319502.3374821 (visited on 07/20/2021).

[272] Shuangge Wang, Anjiabei Wang, Sofiya Goncharova, Brian Scassellati, and Tesca Fitzgerald. Effects of Robot Competency and Motion Legibility on Human Correction Feedback. In: *Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction.* HRI '25. Melbourne, Australia: IEEE Press, Mar. 2025, pp. 789–799. (Visited on 04/15/2025).

[273] Wenshuo Wang, Xiaoxiang Na, Dongpu Cao, Jianwei Gong, Junqiang Xi, Yang Xing, and Fei-Yue Wang. Decision-making in driver-automation shared control: A review and perspectives. In: *IEEE/CAA Journal of Automatica Sinica* 7.5 (Sept. 2020). Conference Name: IEEE/CAA Journal of Automatica Sinica, pp. 1289–1307. ISSN: 2329-9274. DOI: 10.1109/JAS.2020.1003294.

[274] Xin Wang, Yudong Chen, and Wenwu Zhu. A Survey on Curriculum Learning. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (Sept. 2022), pp. 4555–4576. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3069908. URL: https://ieeexplore.ieee.org/abstract/document/9392296 (visited on 04/28/2025).

[275] Yixiao Wang, Yifei Zhang, Mingxiao Huo, Thomas Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu, Pengliang Ji, Wei Zhan, Mingyu Ding, and Masayoshi Tomizuka. Sparse Diffusion Policy: A Sparse, Reusable, and Flexible Policy for Robot Learning. en. In: Sept. 2024. URL: https://openreview.net/forum?id=zeYaLS2tw5 (visited on 10/17/2024).

[276] Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. LADEV: A Language-Driven Testing and Evaluation Platform for Vision-Language-Action Models in Robotic Manipulation. arXiv:2410.05191 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2410.05191. URL: http://arxiv.org/abs/2410.05191 (visited on 04/28/2025).

[277] Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. Towards Testing and Evaluating Vision-Language-Action Models for Robotic Manipulation: An Empirical Study. arXiv:2409.12894. Sept. 2024. URL: http://arxiv.org/abs/2409.12894 (visited on 10/17/2024).

[278] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. In: *arXiv:1709.10163 [cs]* (Jan. 2018). URL: http://arxiv.org/abs/1709.10163 (visited on 09/29/2020).

[279] Christopher Watkins. (PDF) Technical Note: Q-Learning. en. Publication Title: ResearchGate. DOI: 10.1007/BF00992698. URL: https://www.researchgate.net/publication/220344150_Technical_Note_Q-Learning (visited on 09/22/2020).

[280] Christopher J.C.H. Watkins and Peter Dayan. Technical Note: Q-Learning. en. In: *Machine Learning* 8.3 (May 1992), pp. 279–292. ISSN: 1573-0565. DOI: 10.1023/A:1022676722315. URL: https://doi.org/10.1023/A:1022676722315 (visited on 02/24/2022).

[281] Sanne van Waveren, Christian Pek, Jana Tumova, and Iolanda Leite. Correct Me If I'm Wrong: Using Non-Experts to Repair Reinforcement Learning Policies. In: *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '22. Sapporo, Hokkaido, Japan: IEEE Press, Mar. 2022, pp. 493–501. (Visited on 06/22/2022).

[282] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. arXiv:2409.12514 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2409.12514. URL: http://arxiv.org/abs/2409.12514 (visited on 01/02/2025).

[283] Chelsea C. White and Douglas J. White. Markov decision processes. In: *European Journal of Operational Research* 39.1 (Mar. 1989), pp. 1–16. ISSN: 0377-2217. DOI: 10.1016/0377-2217(89)90348-2. URL: https://www.sciencedirect.com/science/article/pii/0377221789903482 (visited on 04/14/2025).

[284] WidowX 250 S. en. URL: https://www.trossenrobotics.com/widowx-250 (visited on 10/17/2024).

[285] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. In: *The Journal of Machine Learning Research* 18.1 (Jan. 2017), pp. 4945–4990. ISSN: 1532-4435.

[286] Jinseok Woo, Yasuhiro Ohyama, and Naoyuki Kubota. Robot Partner Development Platform for Human-Robot Interaction Based on a User-Centered Design Approach. en. In: *Applied Sciences* 10.22 (Jan. 2020). Number: 22 Publisher: Multidisciplinary Digital Publishing Institute, p. 7992. ISSN: 2076-3417. DOI: 10.3390/app10227992. URL: https://www.mdpi.com/2076-3417/10/22/7992 (visited on 04/03/2025).

[287] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis. arXiv:2306.09341 [cs]. Sept. 2023. DOI: 10.48550/arXiv.2306.09341. URL: http://arxiv.org/abs/2306.09341 (visited on 04/28/2025).

[288] Peter R. Wurman et al. Outracing champion Gran Turismo drivers with deep reinforcement learning. en. In: *Nature* 602.7896 (Feb. 2022). Publisher: Nature Publishing Group, pp. 223–

228. ISSN: 1476-4687. DOI: 10.1038/s41586-021-04357-7. URL: https://www.nature.com/articles/s41586-021-04357-7 (visited on 04/27/2025).

[289] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-Pack: Packaged Resources To Advance General Chinese Embedding. _eprint: 2309.07597. 2023.

[290] Tengyu Xu, Yingbin Liang, and Guanghui Lan. CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee. en. In: *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2021, pp. 11480–11491. URL: https://proceedings.mlr.press/v139/xu21a.html (visited on 04/28/2025).

[291] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized Out-of-Distribution Detection: A Survey. arXiv:2110.11334 [cs]. Aug. 2022. URL: http://arxiv.org/abs/2110.11334 (visited on 07/24/2023).

[292] Shuo Yang, Wei Zhang, Weizhi Lu, Hesheng Wang, and Yibin Li. Learning Actions from Human Demonstration Video for Robotic Manipulation. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Nov. 2019, pp. 1805–1811. DOI: 10.1109/IROS40897.2019.8968278. URL: https://ieeexplore.ieee.org/abstract/document/8968278 (visited on 04/26/2025).

[293] Takuma Yoneda, Luzhe Sun, and Ge Yang, Bradly Stadie, and Matthew Walter. To the Noise and Back: Diffusion for Shared Autonomy. arXiv:2302.12244 [cs]. June 2023. DOI: 10.48550/arXiv.2302.12244. URL: http://arxiv.org/abs/2302.12244 (visited on 10/30/2023).

[294] Heng You, Tianpei Yang, Yan Zheng, Jianye Hao, and Matthew E. Taylor. Cross-domain adaptive transfer reinforcement learning based on state-action correspondence. en. In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. ISSN: 2640-3498. PMLR, Aug. 2022, pp. 2299–2309. URL: https://proceedings.mlr.press/v180/you22a.html (visited on 04/27/2025).

[295] Hang Yu, Reuben M. Aronson, Katherine H. Allen, and Elaine Schaertl Short. From "Thumbs Up" to "10 out of 10": Reconsidering Scalar Feedback in Interactive Reinforcement Learning. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2023, pp. 4121–4128. DOI: 10.1109/IROS55552.2023.10342458. URL: https://ieeexplore.ieee.org/abstract/document/10342458 (visited on 03/05/2025).

[296] Hang Yu, Qidi Fang, Shijie Fang, Reuben M. Aronson, and Elaine Schaertl Short. How Much Progress Did I Make? An Unexplored Human Feedback Signal for Teaching Robots. In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. ISSN: 1944-9437. Aug. 2024, pp. 1739–1746. DOI: 10.1109/RO-MAN60168.2024.10731359. URL: https://ieeexplore.ieee.org/abstract/document/10731359 (visited on 03/05/2025).

[297] Hongxiang Yu, Anzhe Chen, Kechun Xu, Zhongxiang Zhou, Wei Jing, Yue Wang, and Rong Xiong. A Hyper-Network Based End-to-End Visual Servoing With Arbitrary Desired Poses. In: *IEEE Robotics and Automation Letters* 8.8 (Aug. 2023), pp. 4769–4776. ISSN: 2377-3766. DOI: 10.1109/LRA.2023.3288382. URL: https://ieeexplore.ieee.org/abstract/document/10158789 (visited on 04/27/2025).

[298] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. arXiv:1910.10897. June 2021. DOI: 10.48550/arXiv.1910.10897. URL: http://arxiv.org/abs/1910.10897 (visited on 10/17/2024).

[299] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-Language Models for Vision Tasks: A Survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.8 (Aug. 2024). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 5625–5644. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2024.3369699. URL: https://ieeexplore.ieee.org/document/10445007 (visited on 10/17/2024).

[300] Ziyi Zhang, Sara Willner-Giwerc, Jivko Sinapov, Jennifer Cross, and Chris Rogers. An Interactive Robot Platform for Introducing Reinforcement Learning to K-12 Students. en. In: *Robotics in Education*. Ed. by Munir Merdan, Wilfried Lepuschitz, Gottfried Koppensteiner, Richard Balogh, and David Obdržálek. Cham: Springer International Publishing, 2022, pp. 288–301. ISBN: 978-3-030-82544-7. DOI: 10.1007/978-3-030-82544-7_27.

[301] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. arXiv:2304.13705. Apr. 2023. DOI: 10.48550/arXiv.2304.13705. URL: http://arxiv.org/abs/2304.13705 (visited on 10/17/2024).

[302] Wayne Xin Zhao et al. A Survey of Large Language Models. arXiv:2303.18223. Oct. 2024. URL: http://arxiv.org/abs/2303.18223 (visited on 10/17/2024).

[303] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. Dec. 2020, pp. 737–744. DOI: 10.1109/SSCI47803.2020.9308468. URL: https://ieeexplore.ieee.org/abstract/document/9308468 (visited on 03/04/2025).

[304] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset. arXiv:2309.11998 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2309.11998. URL: http://arxiv.org/abs/2309.11998 (visited on 04/28/2025).

[305] Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. Transfer Learning in Deep Reinforcement Learning: A Survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.11 (Nov. 2023), pp. 13344–13362. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2023.3292075. URL: https://ieeexplore.ieee.org/abstract/document/10172347 (visited on 04/28/2025).

[306] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593 [cs]. Jan. 2020. DOI: 10.48550/arXiv.1909.08593. URL: http://arxiv.org/abs/1909.08593 (visited on 04/28/2025).

A Appendix

A.1 Customizing Behaviors

A.1.1 ACORD Hyperparameters

Hyperparameters for ACORD as it was used for both environments: painting (for user study) and bipedal walker (simulation).

ACORD Specific Parameters	Paint, BW
All Discriminator Networks	
optimizer	Adam
learning rate	$1 \cdot 10^{-3}$
replay buffer size	10^6
number of hidden layers	2
number of hidden units per layer	(256, 256)
number of samples per minibatch	256
FC 1 layer nonlinearity	ReLU
FC 2 layer nonlinearity	$\frac{1}{2}$tanh
Output nonlinearity	sigmoid
Output σ (minimum, maximum) value	(.1, 1)
λ scale	10
Discriminator Update Parameters	
gradient update every x steps	250, 1000
Reward Function Parameters	
R_{env} reward scale	2, 12
c	15, 5
discriminator based reward scale	12

Table 6: ACORD parameters

SAC Parameters	Paint, BW
Shared Between Actor and Critic Networks	
optimizer	Adam
learning rate	$3 \cdot 10^{-3}$
discount (γ)	0.99
replay buffer size	10^5
number of hidden layers (all networks)	2
number of hidden units per layer	(400,300)
number of samples per minibatch	256
nonlinearity	ReLU
SAC Update Parameters	
target smoothing coefficient (τ)	0.02, 0.005
target update interval	1
gradient steps per environment step	1
entropy optimizer	Adam
entropy optimizer learning rate	$3 \cdot 10^{-3}$

Table 7: SAC parameters

A.1.2 Post-Condition Survey Results in Table Format

	Fun	Desire to Repeat	Likeability	Control	Expressiveness	Mental Demand	Degree of Hard Work
Very High	47.82	43.48	34.78	30.43	17.39	4.35	4.35
High	34.78	26.09	43.48	52.17	39.13	0.0	8.70
Neutral	17.39	21.74	17.39	17.39	26.09	30.43	21.74
Low	0.0	8.69	4.35	0.0	13.04	34.78	43.48
Very Low	0.0	0.0	0.0	0.0	4.35	30.43	21.74

Table 8: ACORD Post-Condition Likert Scale Data

	Fun	Desire to Repeat	Likeability	Control	Expressiveness	Mental Demand	Degree of Hard Work
Very High	39.13	34.78	34.78	8.7	4.35	0.0	4.35
High	47.83	43.48	52.17	65.22	56.52	21.74	13.04
Neutral	13.04	21.74	13.04	13.04	26.09	13.04	21.74
Low	0.0	0.0	0.0	13.04	8.7	34.78	30.43
Very Low	0.0	0.0	0.0	0.0	4.35	30.43	30.43

Table 9: SA Post-Condition Likert Scale Data

	Fun	Desire to Repeat	Likeability	Control	Expressiveness	Mental Demand	Degree of Hard Work
Very High	21.74	21.74	26.09	4.35	8.7	4.35	0.0
High	47.83	43.48	47.83	26.09	17.39	0.0	0.0
Neutral	17.39	17.39	17.39	26.09	17.39	0.0	0.0
Low	8.7	17.39	4.35	26.09	43.48	8.7	8.7
Very Low	4.35	0.0	4.35	17.39	13.04	86.96	91.3

Table 10: RL Post-Condition Likert Scale Data

A.1.3 Paintings

Paintings from all participants in each condition. Each row represents all the paintings of a given participant.

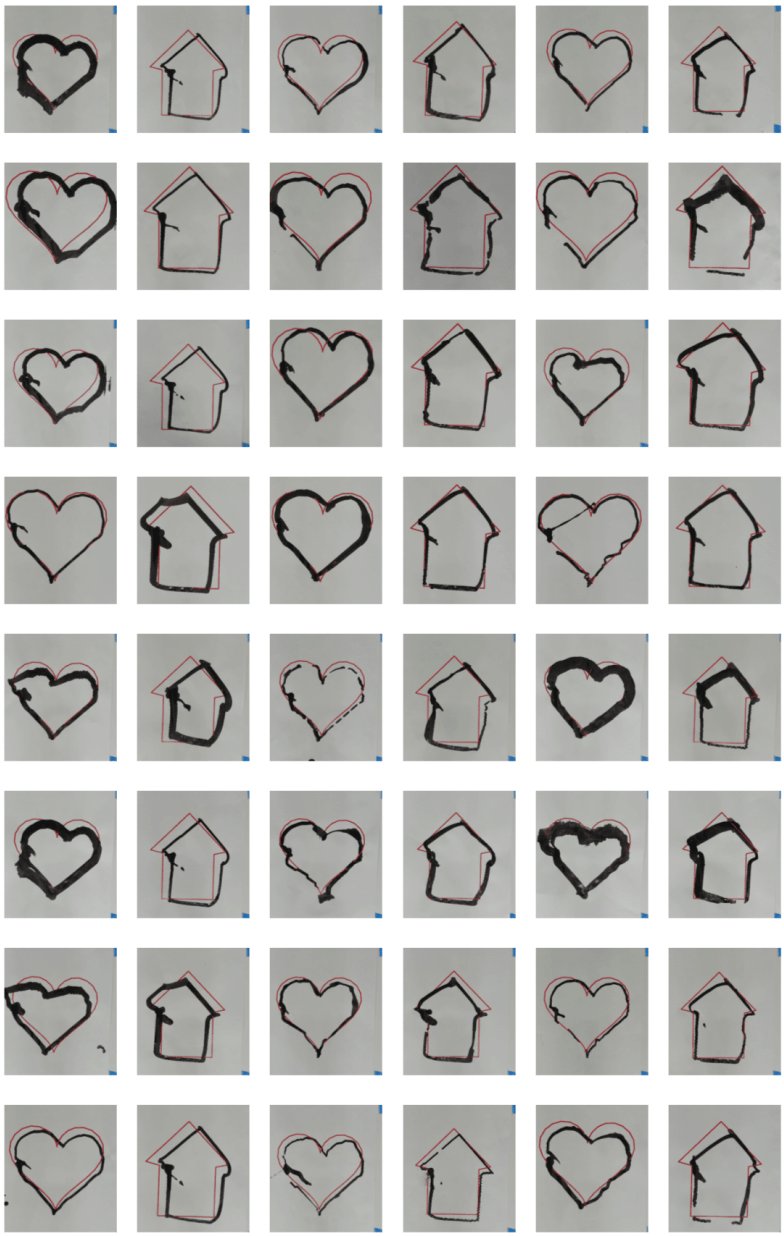


Figure 33: All paintings from all participants and conditions.

A.2 Novice-Friendly Teaching

A.2.1 CAIR Hyperparameters

We used the same hyperparameters for both simulated environments. Hyperparameters where chosen via a grid search over commonly used hyperparameter values for Soft Actor Critic.

Table 11: Teach Hyperparameters

Parameter	Value
Shared Between Actor and Critic	
optimizer	Adam
learning rate	$1 \cdot 10^{-3}$
discount (γ)	0.99
replay buffer size	10^5
number of hidden layers (all networks)	2
number of hidden units per layer	256
number of samples per minibatch	256
nonlinearity	ReLU
reward scale	10
SAC	
target smoothing coefficient (τ)	0.005
target update interval	1
gradient steps per environment step	1

Table 12: Env Hyperparameters

Parameter	Value
Shared Between Actor and Critic	
optimizer	Adam
learning rate	$1 \cdot 10^{-3}$
discount (γ)	0.99
replay buffer size	10^6
number of hidden layers	2
number of hidden units per layer	256
number of samples per minibatch	256
nonlinearity	ReLU
reward scale	2
General SAC	
target smoothing coefficient (τ)	0.005
target update interval	1
gradient steps per environment step	1

A.2.2 Time Spent in Each Enviornment

The time spent in each simulation environment when rendered in real time.

Table 13: BW state-of-the-art Baseline Comparison

BW	50	100	150	200	250	300
Time (min)	4.16-48.33	8.33-96.66	12.5-145	16.67-193.33	20.83-241.67	25-290
Percent of Environment Solved						
SAC	**13%**	13%	13%	13%	13%	13%
SAC-AE	5%	11%	17%	27%	38%	**55%**
td3	5%	5%	5%	11%	13%	22%
CAIR	11 %	**22%**	**44%**	**57%**	**50%**	53%

Table 14: RPM state-of-the-art Baseline Comparison

RPM	250	500	750	1000	1250	1500	1750	2000
Time (min)	31.25	62.5	93.75	125	156.25	187.5	218.75	250
Percent of Environment Solved								
SAC	2%	2%	2%	2%	2%	2%	2%	2%
HERx6	5%	7%	10%	18%	17%	30%	40%	53%
SAC-AE	2%	2%	2%	2%	2%	2%	2%	2%
td3	5%	5%	5%	5%	5%	5%	5%	5%
CAIR	**62%**	**58%**	**62%**	**70%**	**73%**	**78%**	**82%**	**80%**

A.3 Informing Users

A.3.1 Task list

Here is a list of all participant facing tasks used in the study. In the study, the task description was appended to the prefix "You want the robot to." The failure case (FC) description was appended to the prefix "The robot estimates it may fail by" for EFC and "The robot has previously failed this similar task by" for RTFC. The TSR information was appended to the prefix "The robot estimates it can successfully complete the task" for ETSR and "The robot has previously succeeded in this similar task" for RT-TSR.

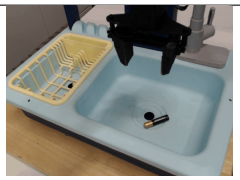
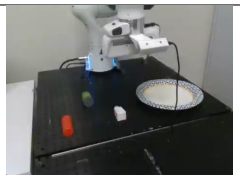
Task (Model)	Task Image	Task Description	Task Information and Similar Task
Stack cups (OpenVLA)		Stack the blue cup face-up on top of the upside down pink cup.	**TSR:** 4/10 times, or 40% **FC**: Picking up the cup but dropping it in the sink, missing the pink cup. **Similar Task:** Lift pan lid
Put away soup (Baku)		Move the can of soup to the refrigerator.	**TSR:4/5 times, or 80%** **FC**: Grasping the can but then halting, failing to put it in the refrigerator. **Similar Task:** Put away bottle
Close oven door (Baku)		Close the oven door.	**TSR:** 4/5 times, or 80% **FC**: Not grasping the handle, leaving the oven door open. **Similar Task:** Make toast
Remove battery (OpenVLA)		Lift and remove the battery from sink.	**TSR:** 7/10 times, or 70% **FC**: Not grasping the battery and then pushes the battery around in the sink. **Similar Task:** Lift pan lid
Move salt (OpenVLA)		Move the white salt shaker onto the plate.	**TSR:** 9/12 times, or 75% **FC**: Trying to pick up the wrong object. **Similar Task:** Put carrot on plate
Lift orange (Baku)		Lift up the orange out of the bowl.	**TSR:** 4/5 times, or 80% **FC**: Not grasping the orange, leaving it in the bowl. **Similar Task:** Remove battery
Put eggplant in pot (OpenVLA)		Take the eggplant out of the sink and put it into the pot.	**TSR:** 7/10 times, or 70% **FC**: Missing the eggplant and hitting the sink instead. **Similar Task:** Put coke in basket

Table 15: List of participant facing tasks used in the online study.

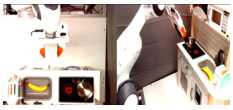
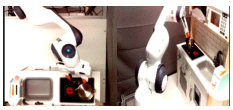
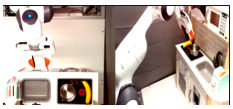
Task	Task Image	Task Description	Task Information and Similar Task
Lift pan lid (Baku)		Lift up the lid of the pan off the pan.	**TSR:** 4/5 times, or 80% **FC**: Partly grasping and then dropping the pan lid handle, causing the pan to shake as the lid falls back on top. **Similar Task:** Remove battery
Move banana to sink (MDT)		Move the banana from the stove and place it into the sink.	**TSR:** 4/5 times, or 80% **FC**: Not picking up the banana, putting nothing into the sink. **Similar Task:** Move banana to stove
Move banana to stove (MDT)		Move the banana from the sink to the stove.	**TSR:** 4/5 times, or 80% **FC**: Being unable to pick up the banana from the sink at the correct location and may push it's gripper into the sink. **Similar Task:** Move banana to sink
Move pot to sink (MDT)		Move the pot from the stove to the sink	**TSR:** 4/5 times, or 80% **FC**: Being unable to grasp the pot, moving it around on the stove and towards the edge. **Similar Task:** Move banana to sink
Put away coke (Baku)		Pick up the coke can and put it in the basket.	**TSR:** 3/5 times, or 60% **FC**: Dropping the coke can and missing the basket, causing it to fall on the counter. **Similar Task:** Put eggplant in pot
Put carrot on plate (OpenVLA)		Pick up the carrot from the sink and put it on the plate.	**TSR:** 4/10 times, or 40% **FC**: Being unable to grasp the carrot and/or by knocking over the plate. **Similar Task:** Move salt
Wipe board (Baku)		Use the towel to wipe down the cutting board.	**TSR:** 5/5 times, or 100% **FC**: Not available. **Similar Task:** Make toast
Make toast (MDT)		Push down the toaster lever to make toast	**TSR:** 1/5 times, or 20% **FC**: Not being able to push down the lever. **Similar Task:** Close oven door
Put away bottle (Baku)		Move the green bottle of tea to the refrigerator door.	**TSR:** 1/5 times, or 20% **FC**: Attempting to pick up an object that is not the green tea. **Similar Task:** Put away soup
(example task) Put away ball (TinyVLA)		Put the tennis ball in the tube.	**TSR:** 4/5 times, or 80% **FC**: Knocking over the tube. **Similar Task:** (example) Put carrot in bowl **RT-TSR**: 3/10 times, or 30% **RT-FC**: Being unable to the grasp the carrot.

Table 16: List of participant facing tasks used in the online study (continued).

A.3.2 Codebooks

Theme	Code	Description
Trust	Trusted estimates	Someone who explicitly indicated that they trusted the estimates or thought the estimates could be relied on
Trust	Skeptical of estimates	Someone who expressed skepticism over if the estimates could be relied on or not/their value.
Trust	Performance threshold	Someone who mentioned a numeric threshold or some threshold of reported performance to where they could make decisions about the task.
Strategy	All info	Mentions all the information was useful
Strategy	Ordered preference	Mentions how they relied one or more types of info but then also used the others as backup or support
Strategy	Mainly used estimated TSR	Someone who more or less just says they used estimated TSR only
Strategy	Past performance	Someone who says they used the previous tasks or data to help them decide (implies sort of meta-learning the study)
Strategy	Similarity	Someone who mentions the value of the real data was based on how similar the task was
Strategy	Risk calculation	Someone who mentions trying to estimate failure risk
Strategy	Compared real and estimates	Explicitly compares real and estimates
Preference	Preferred estimates	Explicitly mentions they preferred estimates
Preference	Preferred real data	Explicitly mentions they preferred real data
Preference	Preferred success rates	Explicitly mentions they preferred success rates
Preference	Preferred failures	Explicitly mentions they preferred failures

Table 17: Thematic codebook developed for the responses to the post-study open-response questions: "How did you use each type of information to make your decision? What made certain information more useful than other information?"

Theme	Code	Description
Robot capability	Robot capability	Wanting to know more about a robot's general specifications, sensing capabilities, or algorithmic ability.
Robot capability	Speed	Wanting to know how fast the robot can perform the task.
Failure	Failure degree	Wanting to know about how bad a failure could be/about the range of possible failures.
Failure	Failure rate and cases	Wanting to know about failure rate as opposed to success rate; wanting to know what in the environment could be damaged or affected because of the robot executing the task; failures unrelated to the requested task.
Robustness	Environment factors	Wanting to know more about the environment: its dynamics and how it may affect the robots performance.
Robustness	Robustness to task	Wanting to know about how robust the robot is to minor variations in the requested task.
Robustness	Robustness to environment	Wanting to know about how robust the robot is to external or environmental factors separate from the task.
Robustness	Failure recovery	How capable the robot is at recovering from failures
Task	Task difficulty	Wanting some sort of measure or assessment of task difficulty or complexity; wanting to know something about how good the robot is relative to human performance on the task.
Task	Task type	General info about the nature of the task; wanting to know the horizon of the task, as in if it is a short or long task/single step vs multi-step.
Task	Performance on requested task	Wanting to know about how good the robot is on the actual task (not like estimated success rate, but like actual performance).
Task	Watch repeated attempts	Expressing a desire to watch the robot attempt the task one or more times.
Learning	Ability to learn	Wanting to know how well the robot can adapt/learn as it executes a task or attempts a task multiple times.
Learning	Learning experience	Wanting to know how the robot was trained or how it previously learned to do the task.
Miscellaneous	Human collaboration	Wanting to know how well the robot could collaborate with a human or handle a human intervention; wanting to know about the robots ability to learn from people.
Miscellaneous	Satisfied	The participant does not express a desire for more information or is already satisfied with the provided information.
Miscellaneous	More info	Expressing a desire for more of the four information types already provided.
Miscellaneous	Deployment duration	Wanting to know about how long the robot has been deployed for or its deployment history.

Table 18: Thematic codebook developed for the responses to the post-study open-response question: "What other types of information may be useful to you to decide when a robot can or cannot reliably perform a task?"

A.3.3 Code Counts

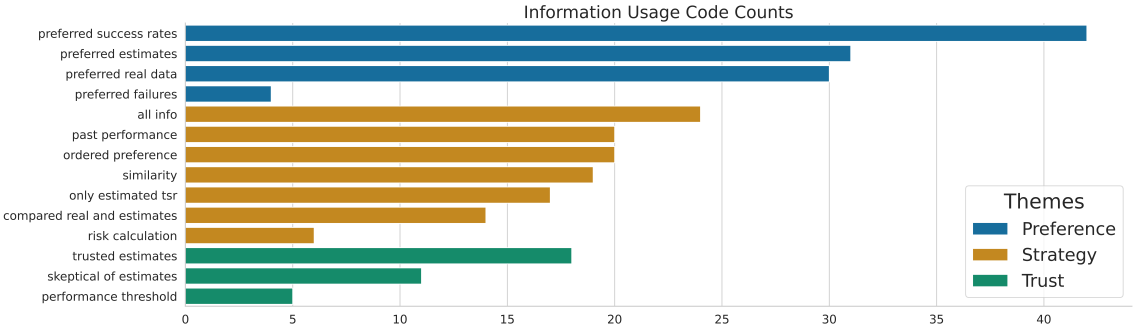


Figure 34: Information usage code counts from online study.

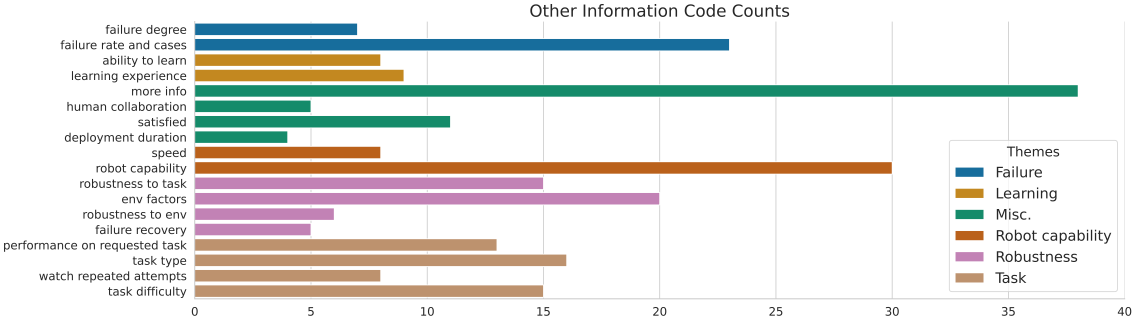


Figure 35: Other information code counts from online study.

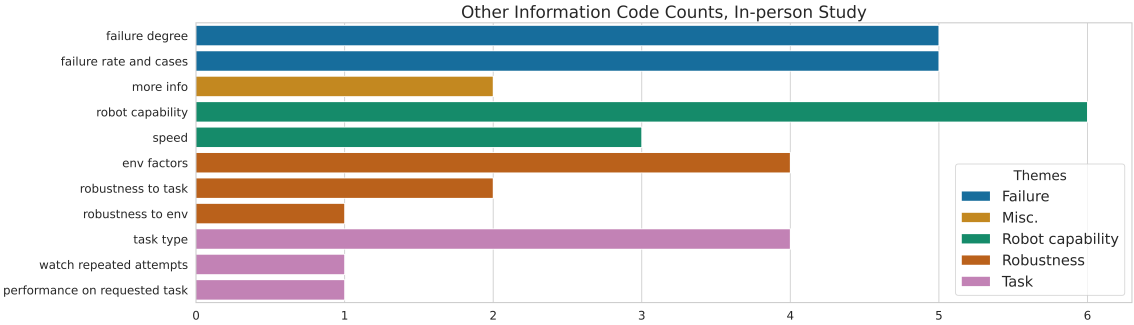


Figure 36: Other information code counts from in-person study.

A.3.4 List of Questions Used

Here is a list all of the questions asked to participants in the online and offline studies (excluding consent and background). Questions labeled "Likert" indicate a 5-point Likert question ranging from "strongly disagree" to "strongly agree."

Online study pre-task execution questions These are the questions asked to participants before seeing the robot attempt a task. Participants could see the description of the task, the task image, and the performance information presented in an information box.

- What is the reported robot's estimate of its success rate for this task? (manipulation check)
  response options: 0-25%, 26-50%, 51-75%, 76-100%, Information not available
- I am confident the robot will be able to successfully perform this task. (Likert)
- I will be surprised if the robot fails. (Likert)
- I would feel comfortable with this robot performing this task in my own home. (Likert)
- I trust the robot to do the task on its own. (Likert)
- I would spend time improving the robot before letting it attempt this task. (Likert)
- I have enough information to assess the robot's ability to perform this task. (Likert)

Online study post-task execution questions These are the questions asked to participants after seeing the robot attempt a task. Participants could still see the description of the task and performance information presented in an information box. They could also freely rewind and re-watch the task execution video.

- The robot succeeded at the requested task. (Likert)
- I was surprised by the robot's behavior. (Likert)
- I received sufficient information to predict this outcome. (Likert)
- I would let the robot do the task again. (Likert)
- Do you have any other comments about the task, information, or robot behavior? (optional) (open-response)

Online study post-study questions These are the questions asked after participants experienced all 16 robot tasks.

- Please rank each type of information based on how useful it was. (drop-down menu ranking)
- Estimated task success rate for a requested task is useful for determining whether or not I want to use the robot. (Likert)
- Estimated failure cases for a requested task are useful for determining whether or not I want to use the robot. (Likert)
- Real task success rate for a similar task is useful for determining whether or not I want to use the robot. (Likert)
- Real task failure examples for a similar task are useful for determining whether or not I want to use the robot. (Likert)

- How did you use each type of information to make your decision? What made certain information more useful than other information? (open-response)
- What other types of information may be useful to you to decide when a robot can or cannot reliably perform a task? (open-response)

In-Person follow-up study questions

- Please rank each type of information based on how useful it was. (drop-down menu ranking)
- Estimated task success rate for a requested task is useful for determining whether or not I want to use the robot. (Likert)
- Estimated failure cases for a requested task are useful for determining whether or not I want to use the robot. (Likert)
- Real task success rate for a similar task is useful for determining whether or not I want to use the robot. (Likert)
- Real task failure examples for a similar task are useful for determining whether or not I want to use the robot. (Likert)
- What other types of information may be useful to you to decide when a robot can or cannot reliably perform a task? (open-response)
- How would you decide whether or not the robot was good enough at a task to want one? (open-response)