Interactive Machine Learning from Humans: Knowledge Sharing via Mutual Feedback

Pallavi Koppol

CMU-CS-23-137
September 2023

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Reid Simmons, Co-Chair
Henny Admoni, Co-Chair
Rayid Ghani
Gonzalo Ramos (Microsoft Research)

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

This work is supported in part by the Office of Naval Research (N00014-18-1-2503).

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Human-in-the-Loop Learning, Learning From Human Feedback, Interactive Machine Learning, Learning from Demonstration, Preference Learning, Active Learning

For my fearless Amma & my poetic Nanna.
Thank you for making any of this possible at all.

Abstract

People regularly interact with human-in-the-loop learning (HiLL) agents that attempt to adapt to their priorities, tastes, and preferences. Examples of such systems include web search engines, movie recommender systems, text prediction, and even large-language model based chat applications. To be adaptive, these HiLL systems must first learn an accurate model of an individual's behavior and preferences.

The ability to learn such a model depends on the quality of the information the HiLL system is able to elicit from the people with whom it interacts, and how well it is then able to leverage that information. Typically, this information is generated via a loop where an agent or model takes actions or makes suggestions that a person responds to with some feedback, and that response is then used to train future behavior. Henceforth, we will refer to this query-feedback pair as an interaction.

We note that the informativeness of a learning interaction is limited by how fully it empowers a person to share their knowledge. We demonstrate that it is possible to improve a person's teaching performance by providing them with (1) more appropriate modalities for sharing feedback (i.e. interaction types) and (2) insight into the context of the learner they are instructing. Our approach therefore moves towards a model that prioritizes a human teacher's ability to provide informative feedback.

To do this, we first formalize the space of interactions that can be used to learn from human feedback and present four interaction archetypes: *Showing, Categorizing, Sorting*, and *Evaluating*. Then, we analyze the effects that these different interaction types may have on learning outcomes via both direct and indirect influences on collected training data. We build on this to contribute a learning approach that enables an algorithmic learner to learn from multiple interaction types based on which would be the most immediately informative. Finally, we develop and evaluate an interaction type-based approach towards bridging the gap between an algorithmic learner and a human teacher's mental model of that learner.

Acknowledgements

I have been able to complete this journey only because of the many people who supplied me with support, advice, affection, and good humor along the way. I am so blessed to have you in my life.

Thank you, first and foremost, to my co-advisors Reid Simmons and Henny Admoni. I could not have asked for a better pair of advisors, and I have benefited tremendously from their expertise, insight, and kindness. Reid, thank you for helping me walk the magical line of thinking deeper while simultaneously trying not to think too much. Some of my favorite meetings with you have been the ones where I came in believing I had nothing to say, and somehow left with deep insights into both my work and myself. Henny, thank you for always making time for me, listening to my myriad ideas and doubts and desires, and sharing with me your unerring ability to find and communicate the underlying narrative of things. I have held on like a lifeline to your years-ago comment that everything will eventually make sense as long as I stay true to myself. I think you were right.

I also want to thank the other members of my committee, Rayid Ghani and Gonzalo Ramos. When I met Rayid, I was halfway through my PhD and trying to figure out how to better align my research with the social good outcomes I wanted to achieve. Thank you, Rayid, for patiently helping me talk through potential career plans, giving me research pointers, and reminding me that everything I accomplish in my PhD is only the beginning. I am also so grateful to have been put in touch with Gonzalo; his forward-thinking and eager approach to research has influenced my own approach to my work. Gonzalo, thank you for helping to guide me through the vast space of human-centered machine learning and for pushing me to think more proactively and less reactively about my work.

I owe a great deal to the collaborators I am lucky to have had during the last several years. In particular, I learned a lot both professionally and personally from Tesca Fitzgerald. Thank you also to Patrick Callaghan, Russell Wong, Yuchen Cui, and Meghna Behari for being some of the funniest, sweetest, and hardest-working people out there.

I am here at CMU largely because of Adam Finkelstein, Manuela Veloso, and Frank Pfenning. Adam was my undergraduate advisor at Princeton, and the one who encouraged me to apply to graduate programs at all. Thank you for believing in me when I couldn't even believe in myself. Manuela, thank you for encouraging me to apply to CMU in particular and for being candid with me about the most pragmatic way to approach graduate school. Thank you Frank, for trusting me to take some extra time off before starting this program; without that grace, I'm not sure if I would have embarked on this journey. I'd also like to thank Barbara Engelhardt and Daniel Suo, both of whom made my undergraduate research experiences enjoyable enough that I decided to pursue research further. And, of course, a huge thank you to Deb Cavlovich and Jenn Landefeld for patiently helping me with all the logistics of achieving a doctoral degree.

The RASL and HARP labs have been my homes here at CMU. Thank you for keeping it real and a little weird and always kind. Mike and Ada, I am so lucky to have started alongside you; thanks for the pep talks, hilarious conversations, and endless commiseration. Michelle and Roshni, thanks for all the shopping trips and walks and laughs. Abhijat, thanks for being one of my most introspective friends and also for always keeping me in the loop so I seem cooler than I am. Reuben, Ben, Stephanie, and Zhi, thank you for years of good advice.

My Pittsburgh family has been the highlight of these past five years. I hardly know where to begin, so I guess I'll start from day one. Misha, thank you for being my first friend here, for being the best office mate I could have ever asked for, and for being there for me through all of the things. Jalani, thank you for somehow always understanding me, for challenging me to think differently, and also for always being down for a long walk; I still think our mothers need to meet. Abhiram, thanks for being one of the sweetest and warmest people that I know, and for making sure I was never working too hard to make gains. Shilpa, thank you for modeling the kind of light-hearted approach to life that I want to embody, for letting me come with you to pick up Bentley, and for making me feel so loved. Siva, thank you for brightening up even the bleakest of times with your goofy pragmatism. Helen, we really made it! Thank you for understanding my PhD feelings better than maybe anyone else, for the trips to the post office and for dreaming about the future. Mark, thank you for being such a steady, comforting presence over the years, and for always being down to cook the most amazing food. Arjun, thank you for being so funny, curious, and brave; I learned a lot from you. Nirav, thank you for delightful meals and the beautiful mornings playing tennis. Arish and Meera, I feel so lucky that Juhi and I happened to meet you both: I will miss our long conversations about everything over some chai. Alex and Angela, thank you for all of the advice and support. Jyotsna and Mohini, thank you for patiently helping me navigate this journey. Roger, Michael, Ellis, and Roie, thank you for the laughs, advice, and ruminations on life. David, thank you for pushing me out of my comfort zone over and over again; also thanks for introducing me to both Marissa and Myra.

Marissa, thank you for being someone who I can talk to about anything. Myra, thank you for being someone who always sees the beauty in things and people, and is always willing to get a little weird. Dorian, thank you for being the best conversationalist at the gym, and giving me excuses to go to the mall. Victoria and Ben, thanks for taking me along with you to Oregon to cook, run, and chat endlessly. Thank you also to Aditya, for a period of regular art sessions. Sam, thank you for being my musical friend and for the long talks.

Cat, I am so grateful for every conversation we've had; thank you for being someone who shares my taste in literature and introspection, and for making me feel so seen. Thank you also for introducing me to Jamie, who is one of the most cheerful, optimistic people I've met. Sara, thank you for the random hallway run-ins and the board games and the commiseration around the PhD. Lisa, thank you for always having inviting and warm tea and snacks and company. Ananya, it's been a joy to reconnect after all these years; thank you for keeping my Sundays lively. Saranya, every time we talk I feel inspired to dig a little deeper and also in awe of your ability to engage so fully with the community around you; please come with me to AVG one day. Giulio, thank you for being one of the most patient and caring people I know, and for creating a space where I can just be myself.

I also need to thank my friends, mentors, and colleagues outside of Pittsburgh, who have been the best long-distance cheerleaders that a person could have asked for. Ella and Lisa, thank you both for being there for me over these long years and for coming to visit me in Pittsburgh. Sarah, Michelle, and Monica, thank you all for being some of the best college roommates and for keeping me company from afar during this long journey. Stephen, thanks for all the long calls about stories and for always knowing how to make me laugh. Jean, something about you always brings out the best in me; I always look forward to Thursday morning calls, and I'm so grateful to have kept in touch across the years and timezones. Min Ju, thanks for keeping me in touch with the bigger picture and showing me what it looks like to tackle life head on. Mariah, Adil, and Kiran, thank you for welcoming me with open arms when I was feeling lost. Rosa, thanks for giving me some of the best career and life advice I've ever received and also for all the amazing conversations. Christoph, thank you for listening to me and believing in me. Felicia, I feel like I've been asking you for advice since high school! Thank you for always making me feel so understood and supported. Vibhaa, my original walko taco buddy, thank you for being there with me through the highs and lows of this whole adventure, and unerringly knowing what I need to hear. Ali, my doctorate journey began with the end of yours but now we can both say we really did it, and read in peace; thank you for always making home feel like home, and also being down to adventure. Juhi, I don't know what I did to get so lucky in this lifetime. For the past five years, you have been the world's best roommate; for the past few decades, you have been my go-to confidante. Thank you for always knowing when to hold my hand.

Finally, I owe everything to my family. Even now, I'm putting them last when they have always put me first. Rajeev Mama, Sunita Atta, Varun, and Vandita, thank you for making me feel so warmly welcomed every time I come home, and always making time to spend with me. Nannamma, Tattayya, Praneeta Atta, Sandeep Mama, Veena Atta, Ravi Mama, Kirti, Ramya, Teju, and Pooji, thanks for always thinking of me, making me laugh, and supporting me. Ammamma, thanks for making sure I managed to graduate from high school and for reminding me that higher education is a luxury and delight. Cheerio, you say so much while saying nothing at all; thanks for letting me hold you and showing me what a good life looks like. Yogi, you are the world's best brother and also the funniest person I know; thank you for being there for me unconditionally and somehow finding ways to make all my fears laughable. Nanna, thank you for patiently listening to me and reminding me to just trust in and enjoy the process. I feel so lucky to get to share this with you. Amma, you are the bravest and most loving person that I know. If I had the courage to undertake this adventure and the capacity to love it even in its hardest moments, it is because of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Intelligent agents such as recommendation engines, large-language models, and assistive devices are becoming fixtures in everyday society due to their growing ability to learn from large-scale data and to personalize based on data from individuals. Approaches to collecting data from people are extremely varied and include asking for annotations on video and images [109], ratings of robot behavior [40], task demonstrations [2], critiques or corrections of proposed trajectories [15, 37], and preferences between options [115]. However, in practice, this richness is often obfuscated and rarely used to meaningfully enhance the flow of information between people and the intelligent systems with which they are interacting.

Let us consider recommender systems as an example, given that hundreds of millions of people interact with them regularly. In the United States of America, 78% of households have a subscription to at least one streaming service and worldwide, Netflix alone has 231 million subscribers as of 2023 [43]. Furthermore, we note that this is all without accounting for the number of people globally interacting with recommendations from non-entertainment sources such as social media, e-commerce, and advertising! A common trajectory of using an entertainment recommendation engine, such as for video content, results in a so-called rabbit hole of content. Initially, recommendations for a user are likely varied and somewhat off-base. Then, as the user interacts with content on the platform more regularly, those recommendations become more refined. They may be quite compelling. At some point, there may be an overwhelming number of recommendations of a particular category (e.g. after watching one romantic comedy, there is a predominance of romantic comedies being recommended)! At this time, the user may begin to feel the presence of the rabbit hole, but options for recourse are unclear and limited: does she simply start negatively rating (via a 'Dislike' or thumbs-down, for example) these suggestions despite the fact that she does enjoy them in smaller quantities? This outcome is negative for both the user of the recommender system, as well as the provider of recommendations as well.

In part, this outcome may be an effect of how information is shared between the user and the recommender system. The machine learning algorithm largely observes what a user is looking at (implicit information), and receives very little in the way of feedback regarding the perceived quality of that content (explicit information). Furthermore, many recommender systems currently use one form of feedback: thumbs up, and thumbs down. However, explicit feedback is extremely valuable and, as we touched upon earlier and will discuss in the remainder of this dissertation, can come in many forms, each of which shares slightly different information. Imagine instead a system where, upon receiving such recommendations, a user could selectively identify which recommendations she likes and dislikes and perhaps even provide insight into why that is! Going one step further, we can imagine a system where at every step there is more of an open and collaborative exchange of information between user and system: after watching a video, perhaps a user can share which parts of it she liked and which she didn't, or whether she'd prefer to watch more of that kind of video or another kind (e.g. she enjoyed watching a cooking video, but would rather have more video essays suggested to her despite watching them less frequently). Such a system may be more likely to lead to value-aligned outcomes, not just in recommender systems but also when it comes to fine-tuning large-language models, assistive technologies, and other systems that regularly interact with individuals.

More generally, understanding the often opaque features of a person's internal world – priorities, preferences, tastes, and beliefs – is a complex, nuanced, and ultimately rewarding endeavor. Doing so affords insights into how a person might perceive, and like to engage with, the world around them. Communicating such information is thus increasingly relevant in the context of the intelligent agents that people engage with regularly. If, ultimately, the goal of these agents is to help people live more meaningfully, efficiently, or joyfully, then it is necessary for them to be able to adapt to an individual's interpretation of what that means.

Often, these intelligent agents interact with the same people repeatedly; as they collect information about these people, they attempt to adapt to their priorities, preferences, and tastes. These are human-in-the-loop learning (HiLL) agents. The entertainment recommender system we discussed previously is an example of such a system, as are other information-sharing platforms and assistive devices. There are other learning paradigms, of course. Some intelligent agents, such as self-driving cars, regularly interact with people who play no role in shaping their development; therefore, a wealth of insights go unheard. However, this still belongs to the umbrella category of learning from human feedback, which contains HiLL as well as less iterative learning paradigms. Other learning paradigms may attempt to adapt to a group, rather than an individual. This dissertation considers only those HiLL systems where an agent is directly learning from and adapting to a single human teacher.

To be adaptive, these HiLL systems must first learn an accurate model of an individual's behavior and preferences. The ability to learn such a model depends in large part on the quality of the feedback the HiLL system is able to elicit and leverage from the people with whom it interacts. Typically, feedback is generated via a loop where an agent or model takes actions or makes suggestions that a person responds to, and that response is then used to train future behavior (e.g. a recommender system serving new recommendations based on a user's viewing history). Henceforth, we will refer to this query-feedback pair as an *interaction*. We refer to the format that this interaction pair takes (e.g. asking for a demonstration, getting a rating, etc.) as an *interaction type*.

In order for such a paradigm to succeed – that is, for the agent to perform in ways that are intended, with minimal externalities – two things are critical. First, that people are fully and effectively able to convey the relevant knowledge that they have. Second, that models are able to appropriately interpret the feedback they are receiving. This dissertation extends the body of work on learning from human feedback by formalizing learning as a joint task with an interaction at its core. We begin by delineating the space of interactions into four primary archetypes (*Showing*, *Categorizing*, *Sorting* and *Evaluating*) and presenting a Markov Decision Process (MDP) formalization of the interactions between people and models. We then build on this to present an active learning approach that uses an expectation of information gain to dynamically reason over the most informative query and interaction type to use in eliciting knowledge from a human teacher. However, people's ability to provide high-quality data can be affected by human factors of an interaction, such as induced cognitive load and perceived usability. We show that measures of performance and accuracy differs significantly between different interaction types. Finally, we investigate how people can understand the implications of the feedback that they provide, and demonstrate that doing so can better align learning outcomes with user expectations.

1.1 Domains of Interest

In this thesis, we focus on learning preferences for behavior in some well-constrained space of safe and appropriate behavior and disregard domains and tasks where blindly adapting to an individual's preferences may not be appropriate.

We are particularly motivated by two broad categories of such agents: recommender systems, such as that presented earlier, and assistive technologies (e.g. assisted feeding). Recommender systems, are commonplace fixtures in society; assistive technologies, remain niche entities with tremendous value. Regardless, the following remains true for any interaction between an individual and an intelligent agent: agents that are capable of eliciting better feedback, or interpreting feedback better, will better be able to unearth individual human preferences. This in turn will have positive downstream effects on the growing number of intelligent systems that have achieved societal penetration.

Currently, for example, users of recommender systems may find themselves pigeon-holed into a small region of the vast space of content available to them, and have limited means of recourse. Content creators may also have a difficult time sharing their creations with the correct target audiences; this is negative for both creator and consumer, and limits the type of content that can be created. One reason for this may be the limited number of interaction mechanisms that exist for consumers to share feedback: they either consume content, or don't, and can occasionally provide a rating. These mechanisms may be insufficient for fully capturing a person's preferences. Meanwhile, assistive technologies that support individuals in activities of daily living can be difficult to use, and rely on human adaptiveness to compensate for their shortcomings. Tailoring their behavior more deftly to the preferences of their users may afford both more autonomy and agency. In order to do this, however, these systems must first be able to learn preferences efficiently.

We further note that there has been a recent interest in learning from human feedback in the context of Reinforcement Learning from Human Feedback [34, 74, 123, 141] for improving the quality of outputs from large language models. Thus far, the primary focus has been on using user preferences between pairs of possible output; however, being able to leverage a wider range of possible feedback avenues and interaction type may be useful in achieving greater resource efficiency. For example, some interaction types may allow for more labels on a single paragraph output, leading to more fine-grained input and evaluations.

We discuss extensions of our research to each of these domains in more detail in Chapter 9.

1.2 Thesis Contributions

The key insight of this thesis is that *the informativeness of a learning interaction is limited by how fully it enables knowledge sharing between agents, namely a human and a learning agent*. Because the quality of the feedback that a human teacher provides depends on both the actions, suggestions, and queries posed by the agent and on the avenues people have to respond to them, it is critical to reason carefully over how an interaction can influence learning outcomes. We hypothesize that **we can significantly improve a lay-person's teaching performance by providing them with more appropriate avenues for both sharing feedback and receiving insight into the context of the learner they are instructing**.

Thus, we contribute a body of work on explicitly learning an individual's preferences that moves towards a model that prioritizes a human teacher's ability to provide informative feedback. We do this by:

1. Developing a Markov Decision Process (MDP) formulation of interactions as well as a novel taxonomy over the space of interactions that can be used to learn from human feedback (Chapter 5)
2. Contributing a learning approach that enables learning from multiple interaction types based on which would be the most immediately informative for the learner based on its current task knowledge (Chapter 6).
3. Analyzing the differing costliness of using each of the different interaction types, and how that may affect downstream outcomes. (Chapter 7).
4. Bridging the gap between the learner and the human teacher's mental model of the learner by using these interactions to provide local feedback from the learner back to the human teacher to guide the learning process(Chapter 8).

Before presenting these contributions, we will first: situate these contributions in the larger narrative of learning preferences within HiLL settings (Chapter 2), present some relevant related work as additional context (Chapter 3), and give more detail on how interactions are particularly important in the HiLL pipeline and why we believe it is so critical to study them (Chapter 4). Finally, in Chapter 9, we will review our contributions, discuss the limitations of our work, and explore extensions to both real-world applications and future research directions.

Chapter 2

An Approach to Interactive Machine Learning from Humans

We seek to develop a human-in-the-loop learning paradigm that prioritizes a human teacher's ability to provide informative feedback. We consider settings where a person interacts with an algorithmic learner in a pedagogical fashion (i.e. they are knowingly attempting to teach), and thereby assumes the role of an expert human teacher. At a high level, our approach towards addressing this can be summarized as follows:

1. The teacher provides information to the algorithmic learner via any of several interaction mechanisms.
2. The algorithmic learner can additionally actively query the teacher for information it believes to be useful, and that it believes the teacher might be able to provide.
3. At appropriate intervals, the algorithmic learner and teacher re-calibrate their models of the others' knowledge, capabilities, and beliefs in order to enable better information sharing.

This series of steps might repeat indefinitely until – and, perhaps, for as long as – the algorithmic learner displays the desired level of proficiency as determined by the teacher.

In the remainder of this chapter, we elucidate each of these steps. Section 2.1 further describes the problem setting of interest. We are particularly interested in tasks where laypersons are knowingly put in the position of being a teacher. Given such a task, Section 2.2 discusses the various interactions via which knowledge can be transferred from teacher to algorithmic learner and how we can meaningfully understand the influences of those interactions. Section 2.3 further explores how an algorithmic learner can support the learning process via an active querying paradigm that is mindful of the differing influences of interactions. Finally, because both algorithmic learner and teacher are participating in a learning process that is temporal in nature, the influence of, and methods of adjusting for, shifting beliefs during the learning process are discussed in Section 2.4.

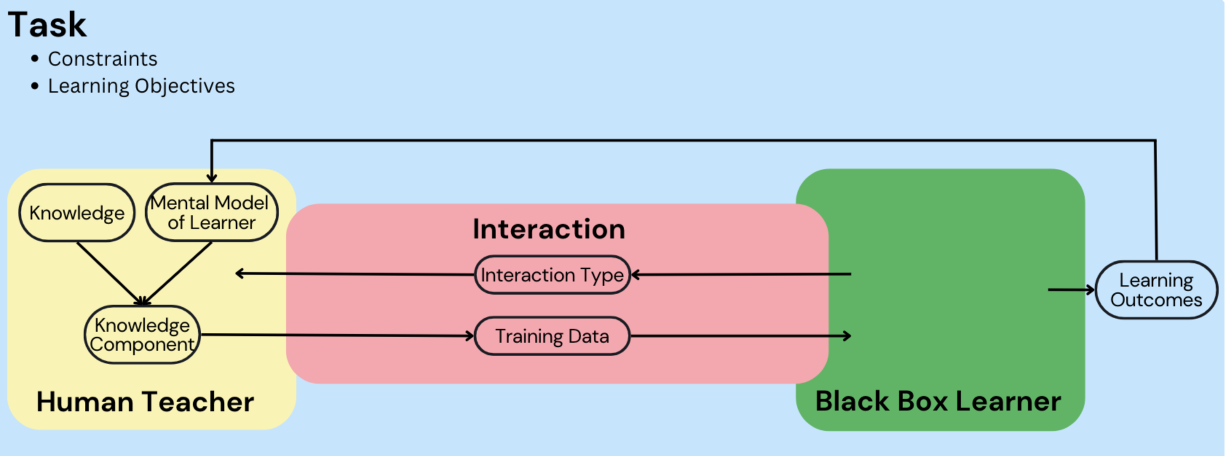


Figure 2.1: We are particularly interested in *how* human teachers share information with learning agents, and *what* information they choose to share. Both of these are influenced by the interactions that are central to the interactive machine learning pipeline. In this dissertation, we explore what interactions entail, as well as the many pathways via which they influence knowledge transfer between human teachers and learning agents.

2.1 Overview

The overarching problem of learning from human feedback manifests in many forms in part because of the variety of roles that both algorithmic learner and human can play.

Online and Offline Learning. An online learner will collect a variably-sized batch of data, perform a model update, and then collect additional feedback data in response to its new performance. In that sense, there is an expectation of an ongoing learning interaction between learner and data source (in our case, a human teacher). In contrast, an offline learner will receive all training data at once and perform no model update between any of the interactions used to collect that training data.

Active and Passive Learning. Passive learning involves a learning agent being trained on a data set that is curated independently of its learning status. On the other hand, active learning enables the learner to independently query for informative data points.

Pedagogical and Pragmatic Teaching. People have been shown to demonstrate tasks differently if they know that a learner is attempting to learn from them, as opposed to if they are asked to complete it as efficiently as possible [48]. Furthermore, if people know they are engaged in a teaching task, they may have different expectations from learners than otherwise.

In this dissertation, we focus on *online, active learning* of *individual preferences* from *pedagogical* human teachers. The domains described in Chapter 1.1 are precisely such settings. We focus on learning individual preferences for a few reasons. First, this allows us to explore how to elicit and leverage knowledge from laypersons as if they were experts. People are the authorities on themselves, especially when it comes to preferences. Generally, they know what they like even if they can't convey that information perfectly. Thus, we can assume the existence of some ground truth preference despite potentially noisy, inconsistent, and uncertain feedback. Second, online and active learning processes have a tighter coupling between model learning and human teaching; this makes evaluating shifts in teaching behavior more tractable, and it positions learning as more obviously a collaborative process. Finally, we are interested in enabling people to teach as effectively as possible and are therefore interested in settings where people are knowingly acting in a pedagogical fashion.

We recognize that there are other challenges to address as well, such as that preferences are often non-stationary and people are not omnipotent. They may not realize that they will like something, because they haven't tried it yet (e.g. I haven't tried watching a dark comedy movie, but I would like it if I tried it. However, I won't tell you that I like it, because I don't know this yet!) Furthermore, the relative strength of their preferences may ebb and flow with time (e.g. if I have watched dark comedies every day for the past week, I may prefer a documentary instead, even if I usually prefer dark comedies to documentaries). While compelling, this is beyond the scope of our work. Even if we consider the set of problems that assume preferences to be stationary (e.g. if we are trying to understand what people currently want, and believe we will not interact with them long enough for their preferences to shift), there will still be learning artefacts that arise because people share knowledge along with biases, inconsistencies, and contextual inferences. Our work addresses these artefacts.

In the upcoming sections, we discuss how to evaluate and mitigate for the ways in which interactions affect how knowledge is shared.

2.2 Sharing Human Knowledge via Learning Interactions

Given such an online, active learning setting with a pedagogical human teacher, we are interested in the interactions that permit knowledge to transfer from teacher to algorithmic learner. In this section, we provide more detail on our approach to analyzing these interactions.

First, in order to discuss interactions, we define the following terminology. Let a query q refer to data that a human teacher U is prompted to respond to by learning agent L. An *annotation* is the feedback that a teacher gives in response to a query. An *interaction type* is the format of a query (e.g., "Showing", "Categorizing", "Sorting", or "Evaluating", as introduced in Section 3.1). An *interaction instance* is a specific query, e.g. "Should the label be 'library' or 'bookshop'?" and its associated annotation. Finally, an *interaction session* is a series of interaction instances. This framing is formalized in more detail in Chapter 5.

In Section 3.1 we present a variety of interaction types present in the literature; still, these are but a handful of the interactions that exist for learning from human feedback. To compare each proposed interaction type against every other would be an intractable endeavor. Fortunately, many of these interaction types share fundamental similarities in how people interface with them. By taxonomizing the space of interaction types along those lines, we can more easily compare clusters of interactions.

Thus, we characterize the growing number of interaction types along the following dimensions:

Query Size. The query size of an interaction type refers to the amount of data that a learner requests feedback on.

Response Size. Similarly, the response size of an interaction type refers to the amount of data contained within an annotation.

Intervention Options. The intervention options that an interaction type affords refers to the granularity with which a human teacher can respond to a query. For example, a *Categorizing* interaction such as requesting a positive or negative label on a trajectory requires that a human teacher label the trajectory in aggregate as good or bad. On the other hand, an *Evaluating* interaction such as credit assignment allows a human teacher to identify specific regions of a trajectory that are good or bad.

Response Choice Space. The response choice space of an interaction type defines the set of possible annotations that a human teacher can give in response to a query posed by the algorithmic learner.

From this, we identify the four interaction archetypes of *Showing*, *Categorizing*, *Sorting*, and *Evaluating*. The aforementioned dimensions and interaction archetypes are formalized with more technical detail in Chapter 5.

We note that our taxonomy only considers explicitly given feedback. Understanding the further implications of the implicit information that interactions allow for is outside the scope of this thesis, though it remains a growing area of research as described in Section 3.1. In Section 2.3, we discuss our approach to, and the importance of, leveraging these distinctions between interaction types in learning from human feedback.

2.3 Learning from Multiple Interaction Types

The four interaction archetypes (*Showing, Categorizing, Sorting*, and *Evaluating*) differ in ways that affect learning outcomes via two pathways: indirectly via how people perceive and respond to these interactions and directly due to how training data is generated. This section presents metrics by which we can measure the influence of teaching performance on training data, an information theoretic formulation for understanding the effects of interactions on training data, and an approach towards leveraging multiple interaction types in one interaction session.

The effects of teaching performance on training data are important to understand, due to potential downstream effects on learning outcomes. While there are many ways to evaluate teaching performance, we focus on:

Cognitive Load. As the cognitive load (i.e. the portion of working memory being utilized) on an individual increases, they grow more easily distracted and tend to have worse task performance [128]. There are both subjective and objective measures of cognitive load.

Performance. We consider task performance (e.g. providing good and efficient data). Typically, when collecting data for machine learning models, we are concerned with the quantity of data we can obtain within a given time and budget constraint. However, we also want the quality of the data to be high.

Usability. The usability of any particular interaction type is a purely subjective measure that may lend insights into how comfortable people are providing feedback via various mechanisms. It is likely that the more usable a person finds an interaction type, the more willing they will be to use it to provide information.

We designed and ran a user study to measure effects along the aforementioned axes, and found that statistically significant differences exist between interaction types on each of these measures. Further details on our experimental setup and analyses can be found in Chapter 7.

Furthermore, even without considering the effects of teaching performance on training data, we can see that the choice of interaction type directly affects the training data that can be collected. This is trivial to see with respect to training data format: the format of data collected via a forced ranking question (i.e. a *Sorting* interaction) will be different than the data collected by asking for a demonstration of a trajectory (i.e. a *Showing* interaction).

However, we can also look at the informativeness of each piece of training data by framing data acquisition as an information-gain problem. We ground the problem of calculating the information gain Φ_i for the optimal query of interaction type i as follows:

$$\begin{aligned}\Phi_i(\omega, s) &= \max_{q \in Q_i(\omega,s)} IG(\omega, C_i(q)) && (2.1)\\ &= H(\omega) - \min_{q \in Q_i(\omega,s)} \mathop{\mathbb{E}}_{c \in C_i(q)} \big[H(\omega|c)\big] && (2.2)\end{aligned}$$

Here, $C(q)$ is a function that defines the choices that a human teacher has in terms of what feedback they can provide given a query q, and $Q(\omega, s)$ defines the set of queries that the agent could pose. More details on this formulation and the functions that comprise it can be found in Section 6.1. We also show that this implies potentially disparate effects of various interaction types on training data, and present our ongoing work on an information-gain based approach towards learning from multiple interaction types in a single session.

An interesting extension of this would be to further incorporate some quantifiable measure of influence on teaching performance into the interaction type selection process. However, to quantify fluctuations in teaching performance in such a fashion is beyond the scope of this thesis.

2.4 Revealing Agent Knowledge via Learning Interactions

Still, if our goal is to make the process of giving and receiving feedback more effective, we cannot simply focus on feedback modalities. In this section, we consider how teaching performance – and thereby, training data – may additionally be influenced by perceived learning outcomes. It is insufficient to consider only the feedback mechanisms available to people because the content of the feedback that people deliver will depend on their mental models of the learner (e.g. a person might give more detailed information to a learner perceived as more competent). There are two primary approaches that we can take towards bridging the gap between the learner's model and the teacher's model of the learner: we can imbue the learner (agent) with a more accurate model of the teacher (human), or we can make the learner more transparent to the teacher.

One approach towards this is to help make the learner more transparent to the teacher. We consider how the interaction types discussed previously can be used to locally expose the black-box model of the learner and give a sense of how information shared by the teacher is being interpreted. We decide to use the interaction types previously discussed because we view them as purely information-sharing mechanisms, and are curious to evaluate the symmetry – or lack thereof – of knowledge transfer from human to agent and vice versa. We discuss this in further detail in Chapter 8.

Another approach towards giving the learner a more accurate model of the teacher might involve developing an active learner that generates queries that are both informative to the model and informative to the human instructor. This would involve building off of our existing information gain formulations of interaction types. Furthermore, we would have to construct and incorporate a model of how a human teacher might interpret the questions being posed by the machine learner (e.g. a teacher might perceive questions that are very similar to each other as indicating that a concept has not been learned). While this is an interesting question, it is beyond the scope of this dissertation.

Together, these two approaches – making the learner more transparent to the teacher, or the teacher more legible to the learner – lend themselves to moving towards learning process that are more bidirectional.

Chapter 3

Related Work

HiLL as we have described it is one manifestation of the larger space of interactive machine learning approaches. Interactive machine learning (IML) can be distinguished from other forms of applied machine learning by virtue of rapid update cycles featuring incremental changes that involve both the algorithmic learner and human teacher [8, 46]. IML is related to Interactive Machine Teaching (IMT, not to be confused with the literature on machine learning systems that attempt to teach their policy to a user known as Machine Teaching [23]) which sees human teachers adopting a three-phase process: they plan what to teach, they explain relevant information, and they review what the learner has understood [107] and the focus is on "the efficacy of of the teachers *given the learners*." [121]. As a result of this paradigm, human teachers have the opportunity to adapt their interactions with an algorithmic learner to better achieve their desired outcomes. Every aspect of this process is an active and broad area of research.

In the remainder of this chapter, we discuss a subset of research related to the interactions that take place during the HiLL pipeline. We are especially interested in IML/IMT settings that may additionally leverage an active machine learner which can intentionally query a human teacher for training data. In particular, we discuss the forms that these interactions can take, the influence of a human teacher's behavior on learning outcomes, and the ways in which model updates can be expressed to, and understood by, human teachers.

3.1 Learning from Human Feedback

Learning interactions between humans and intelligent agents can take on many forms. Cakmak and Thomaz [27] proposed a categorization of interaction queries that correspond to questions that people tend to ask: label, demonstration, and feature queries. Zhang et al. [138] presented a survey on different types of human guidance specifically for deep reinforcement learning and identified four different learning scenarios: standard imitation learning, learning from evaluative feedback, imitation from observation, and learning attention from human. Najar and Chetouani [94] presented a taxonomy of "advice" for RL agents, categorizing it first according to whether it provides contextual or general advice, and then according to whether it indicates feedback, instructions, or constraints.

Additional research into developing intelligent agents that emulate the rich learning interactions people use has involved combining multiple interaction types [25] to better leverage human teachers, leveraging trade-offs between interactions [103], and exploring explicit and implicit information transfer [66]. More recently, this body of research – particularly learning from preferences – has been popularized in the context of Reinforcement Learning from Human Feedback [12, 34, 74, 124, 141]; we discuss this further in Chapter 9.

The research presented in this dissertation extends this body of work by providing a general taxonomy of interaction types as well as a principled framework for representing model-agnostic interactions. Throughout this dissertation, we build on this framework and taxonomy in order to understand the differing informativeness and ease of use of many of the aforementioned interactions. Based on our taxonomy (justified in Chapter 5), we discuss four categories of interactions.

Showing. The teacher provides a demonstration of the agent's expected output. This form of interaction is common in the highly-active research field on Learning from Demonstration [10, 33]. Inverse reinforcement learning (IRL) is a learning from demonstration technique for recovering a reward function from which to train a policy [2, 95]. Behavioral cloning learns a policy directly [13, 116]. Alternatively, the teacher may verbally explain the expected behavior in the form of "advice" indicating what the agent should or should not do in a particular state [79, 94]. While demonstrations can be highly informative, people are limited in the number of examples they can provide and by their expertise.

Categorizing. The teacher provides one or more labels from a predefined set. This type of learning is commonly used for classification and regression. For example, computer vision leverages popular large-scale labeled datasets [42, 45, 83]. In another approach, a human teacher may indicate which object (from a set of candidate objects) is most immediately relevant for a particular task [49]. Labels also include the assignment of rewards to actions, as in bandit problems and reinforcement learning with human feedback [40, 76]. The informativeness of labels is limited by the size of the label set, and people are known to give both overly positive rewards [8] and shifting ratings [101].

Sorting. The teacher indicates their relative preferences over a set of choices presented by the agent. Preference elicitation is an active area of research, especially in recommendation engines [52]. Comparison and ranking-based approaches for learning reward functions are increasingly common [19, 115, 136]. These interactions are precise, and thought to be low user effort. The technique is good at fine-tuning, but the information that can be gained from each query is limited.

Evaluating. The teacher provides granular feedback on an agent's proposed or executed actions. Corrections are feedback on a proposed set of actions either during or after task execution. These can be physical or simulated [15, 65]. These corrections may range from fine-tuned adjustments of the robot's end-effector pose [11, 50] to perturbations of the robot's intended trajectory [14] to changes in the hierarchical structure of the task [56]. Users can also mark good or bad regions of a trajectory via critiques [37]. Credit-assignment interactions, such as the one posed in [66], can be construed as a form of critique where the user is limited to identifying only one good region. Off-switch games can be considered as another special case of critiques where trajectories are segmented into a singular allowed section preceding a singular disallowed section [57].

We note that quality control research for data collection investigates how users, often crowd workers, can provide good data via incentives and task design [73, 80]. Our research differs in that we study how human factors, such as cognitive load and usability, differ between various interaction types. The correlations between cognitive load, usability, and task performance (e.g. providing good data) have been observed and studied throughout cognitive psychology [128], human-computer [85] and human-robot interaction [105].

3.2 Influences on and of Teaching Performance

While "ground-truth" for optimal human performance in HiLL systems may not exist, there are measures that are known to affect human performance. In particular, an increase in *workload* has been correlated with a decrease in task performance [105, 128]. Workload can be measured both in subjective, self-reported measures and in objective task measures [85]. The NASA-TLX survey [60] has been widely adopted in human factors research to measure subjective workload, and consists of several sub-metrics including mental demand, physical demand, temporal demand, performance, effort and frustration. The popular System Usability Scale (SUS) is a validated survey that provides a subjective measure of the *usability* of any given system, reflecting measures such as users' ease and confidence when using a system. A strong, positive association exists between task performance and subjective satisfaction with an interface [98]. Workload and usability have also been found to be non-overlapping measures in an HCI task, which suggests that combining them may provide a more accurate prediction of objective task performance [85]. Our work leverages these as established metrics by which to evaluate teaching performance.

There are many additional axes along which we can discuss the influences of interactions on training data. In Section 4.3, we focus primarily on how *noisiness* in the data, as well as the *quantity* and *distribution* of collected data affect learning outcomes. Human teaching performance has previously been shown to directly influence those particular factors as well.

Noise. We focus on noise introduced via human error (e.g., where a human teacher fails to provide the conventional ground truth). For example, data collected from crowdworkers can be low quality, as workers are incentivized to maximize their own earnings at the potential expense of providing thoughtful labels [63]. Noisiness can also arise from human teachers without adversarial intentions, due to factors such as the amount of precision afforded by a particular user interface [6].

Distribution. Collecting well-distributed data that captures domain shifts is critical for robust models. The availability of crowdworkers suggests the possibility of increased diversity in dataset curation [80], and has already been shown to manifest in more efficient exploration and learning [88]. The teaching interaction may be adapted in response to poorly-distributed training data; for example, tasking a teacher with finding a positive example in an underrepresented region of the state space [82]. However, interaction mechanisms that are not designed to be accessible and usable by a variety of individuals may result in datasets that either eschew or result in low-accuracy feedback from entire swaths of people [132].

Quantity. In HiLL systems, the interaction type being leveraged can affect the rate of data collection, and ultimately limit the amount of data collected. Demonstrations on physical robots, for example, can be difficult and time-consuming to provide; simulated approaches with user-tested interfaces can increase labeling throughput and lead to better learned policies [72, 88].

3.3 Transparency in Algorithmic Learners

We are particularly interested in model transparency in the context of improving teaching performance in HiLL systems. Transparency is critical in such settings because the feedback that people provide is influenced by a learner's current (perceived) policy [87]. Studies have repeatedly shown that having additional insight into a learner's model can influence teaching behavior and increase teaching efficacy [32, 96, 118, 130]. In the remainder of this section, we first discuss existing literature demonstrating the effect of model transparency on teaching performance. We then explore several approaches towards achieving such transparency.

Effects of Model Transparency on Teaching Performance. A human teacher's mental model of an agent's learning status may include assessments of the agent's current knowledge and performance over the task. This mental model can affect various factors of a teacher's task performance including planning, persistence, and satisfaction [68]. As a result, it is important to consider how this mental model may be affected by the agent's performance and the interaction between the teacher and agent. Hedlund et al. [61] found that agent performance can affect a teacher's mental model of both the agent and their own teaching capability. Krening and Feigh [78] showed how teachers perceived an agent trained using verbal demonstrations as being more intelligent and better-performing than the one trained through binary critiques. Furthermore, interactive learning methods lent themselves to more accurate assessments of agent capability as compared to passive, supervised learning [28]. Prior work has investigated the relationship between active learning techniques and model transparency (e.g. by asking questions in regions of greater transparency) [32]. Our research extends this body of work by exploring the relationship between active learning and model transparency through the lens of various learning interactions.

Approaches to Model Transparency. There are number of popular approaches to model transparency, but not all of them are well-suited to the type of rapid, interactive learning we are interested in. Most of these approaches fall into two categories: developing models that are inherently interpretable [113] (e.g. decision trees, linear models), and models that provide post-hoc explanations (e.g. techniques such as LIME [111], or saliency maps [127]). Such approaches may be useful in interpreting complex deep learning models, but may be too heavy-handed for quick iteration cycles between an algorithmic learner and human teacher required of interactive learning settings, particularly when lay-persons are involved. Another axis along which to consider transparency and explainability is whether users are provided with local or global-level insights into the model [64]. Finally, a recent taxonomy grouped explainable AI methods for reinforcement-learning into three categories: feature importance (e.g. directly generating explanations), learning process and MDP (e.g. decompose reward function), and policy-level (e.g. summarize using transitions) [91]. The work presented in this dissertation focuses primarily on local transparency via feature importance methods.

Alongside the methods mentioned previously, we can also consider approaches using different mediums, such as visualization techniques [9], which may be valuable in interactive settings due to their more simple to understand nature [135]. It is challenging to get transparency and explainability correct for many reasons. One of these reasons is that the type of transparency required may vary by the role and intentions of the end-user [7]. Another is that sometimes explanations, like those from post-hoc models, can simply be incorrect [3, 113].

We are particularly interested in techniques that promote model transparency in an active learning setting. While there have been approaches that attempt to give human teachers more insight into how their feedback will be interpreted [16], finding lightweight paradigms by which an algorithmic learner can convey information back to the human teacher remains a relatively open question. In human-robot interaction, implicit behavior and nonverbal cues have long been studied as a way to increase transparency for human-robot teaming [20]. In our research, we are interested in exploring novel interaction paradigms for explicitly communicating a learner's model in an active learning setting.

Chapter 4

Contextualizing the Role of Interactions in HILL

Before we delve further into defining what an interaction entails and how interaction types can affect learning outcomes, we must first understand the larger HiLL paradigm within which interactions are situated. In this chapter, we propose a relationship graph (Fig. 4.1) as an organizing principle for HiLL systems in order to comprehensively analyze the effects of interaction types. At a high level, a primary goal of selecting an *interaction type* is to choose the one that best aligns the *learning outcomes* and *learning objectives* of the overall system. These learning outcomes are dependent on the *training data* obtained by the system; in HiLL settings, this training data comes from a teacher and is thus affected by *teaching performance*. In this section, we will further define each node on this graph. This section is the product of a collaboration with Yuchen Cui and Tesca Fitzgerald. Chapters 5 and 7 will later delve into the labeled edges between nodes.

For our purposes, we abstract away the details of the task (e.g. specific domain, learning model, and interface in which a HiLL system is deployed). In Chapter 5, we justify these assumptions by providing a formalization of interactions that allows us to analyze them separately from any underlying learning models. As a result, the design choices and relationships analyzed in this section may be considered within any domain. To provide concrete examples, however, we ground our discussion in the context of a warehouse robot tasked with restocking items. In this example, the robot collects data from humans to train separate models for object recognition and manipulation.

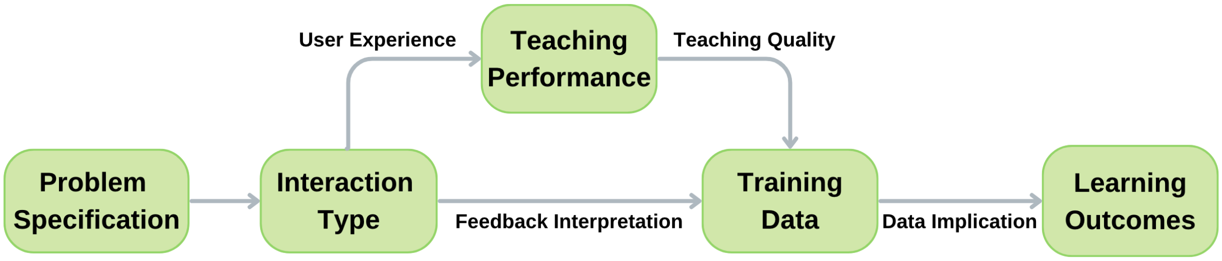


Figure 4.1: This graph outlines several key relationships that affect a HiLL system's ability to meet its problem specification, which consists of the system's learning objectives and constraints.

4.1 Problem Specification

The problem specification consists of the objectives and constraints of a given learning problem. *Learning objectives* describe the goals of designing a HiLL system, which may consist of objectives such as the expected performance on the training, testing, and generalization datasets, output consistency, sample efficiency, (adversarial) robustness [137], and/or explainability [112]. *Constraints* of a HiLL system specify the requirements and limitations, which may include the size of the training data set, physical limits, safety requirements, and so forth.

In the restocking robot domain, for example, the learning objective of the robot's object recognition model is to meet an expected object detection accuracy above some threshold, while also being robust to changes in lighting conditions in the warehouse. Within the robot's manipulation model, the objective is to generalize its grasping model learned from a small set of objects to robustly grasp novel ones, under the constraint that collisions in any form should be avoided.

4.2 Teaching Performance

In a HiLL system, the agent's task is to achieve specified objectives, and the human teacher's task is to provide data that supports the learner. The quality of this training data is critical to the agent's learning outcomes, and is affected by how well the human teacher executes their teaching task. We define *human performance* as a human's ability to provide accurate feedback during a learning interaction. We can more precisely call this *teaching performance*, for our purposes.

For example, if the restocking robot requests feedback on grasping a new object, human performance consists of the human's ability to provide a demonstration, correction, or other indicator that results in a stable grasp of the object. The teacher's ability to provide quality data depends on how they may provide feedback; if the robot requests a ranking between two candidate grasping poses that are equally bad, the teacher may have difficulty deciding between the two and is unable to express feedback about how the robot *should* grasp the object.

4.3 Training Data

Training data is the set of data samples generated by the human teacher through interacting with the learning agent. For the restocking robot, this could consist of object type labels, robot arm trajectories, and/or ratings of trajectories. Training data can have different **data implications**. The *quality* and *quantity* of training data affect learning outcomes in various ways. Two measures of *quality* of a data set are *noisiness* and *distribution*. *Quantity* straightforwardly refers to the number of samples in the data—however, how much data is *enough* is often determined by the complexity of the task itself and the learning objectives of the agent.

Noise. Noise can be introduced during data collection through human error (labeling error). Noisy data often leads to a false measure of training and testing performance [35, 59], the effect of which is specific to the training algorithm [69]. During data collection, especially when crowdsourcing [80], it is important to control such noise in labels through explicitly correcting for labeling bias [122] or modeling noisy labelers [119].

Distribution. This encompasses the diversity, biases, and representativeness of the data sets. ML models may overfit not only to training data, but also to test and generalization data as a result of the research community using identical benchmarks [110]. It is important to design distributions of test and generalization sets that account for domain shifts to better understand generalization errors of ML models [125].

Quantity. Data quantity has proven to be a crucial factor of performance of ML models [58], especially for deep neural networks, as demonstrated on image classification [42] and natural language processing [24]. Small training datasets can lead to overfitting [108]. Performance of deep models on vision tasks increases logarithmically with the volume of training data [126]. Leveraging large amounts of unlabeled data, self-supervised representation learning also improves performance of deep models [93]. We note that there is sometimes a tradeoff between the quality of data and the quantity that can be collected. However, there is also a push to be more resource efficient and focus on collecting fewer, higher quality training data.

4.4 Learning Outcomes

The *outcomes* of HiLL systems are the objective measures of performance of the trained system. In an effective HiLL system, these outcomes should fulfill the previously-described *learning objectives* that define the performance goals of the system. We first consider three common performance-based learning outcomes.

Training performance reflects the model's ability to represent its training data. The exact performance metric is domain-specific. In supervised learning, *training accuracy* is a common metric representing how well the trained model can reproduce the expected output from its training dataset. For reward-based task learning, *policy loss* in the training environment is often used as a performance metric.

Testing performance reflects the model's ability to produce the expected output for inputs that are drawn from the same distribution as, but not included in, the original training dataset. This testing dataset represents the set of problems that the trained model is expected to encounter in its domain.

Generalization performance reflects the model's ability to produce the expected output for inputs that are drawn from a significantly different distribution from the original training dataset. This type of learning outcome may not apply to all learning domains, but is frequently used in domains where the agent learns multiple tasks (e.g. one-/few-shot learning).

In the restocking robot example, the agent's learning outcome for the object recognition task would be its training and testing performance in classifying catalogued items. Meanwhile, generalization performance for its manipulation model might include grasping and accurately reshelving newly-introduced, uncatalogued objects.

4.5 Relationships

This section introduces the major nodes of our proposed relationship graph (Figure 4.1), except for interaction types which will be addressed shortly in Chapter 5. We also discuss the **data implication** edge of the graph and presented characteristics of training data that affect an agent's ability to fulfill its learning outcomes. There also exist the following edges in the graph:

1. **Feedback interpretation**: how the teacher's feedback is synthesized into training data,
2. **Teaching quality**: how the human teacher's experience affects the quality of their feedback (e.g. via imposed cognitive load and other performance metrics), and
3. **User experience**: how the queries posed via this interaction type are perceived by the human teacher.

In our relationship graph, learning outcomes serve as a feedback signal into the problem specification. This signal is analogous to the so-called "gulf of execution" in interaction design [100]; that is, it measures how well the learning outcomes meet the stated learning objectives. The HILL cycle repeats until, at some point, learning outcomes and objectives are aligned. Evaluating this gulf of execution and determining when to pause and re-start the learning process are necessary questions that remain outside the scope of this thesis.

Instead, note that the edges that remain to be discussed all (directly and indirectly) influence the training data that is collected. The remainder of this thesis will focus on these pipelines between interaction types and training data, given that we know training data ultimately affects downstream learning outcomes. We will discuss **interaction types** themselves in Chapter 5. **Feedback interpretation** will be discussed in Section 5.3, while the **teaching quality** and **user experience** edges will be discussed in Chapter 7.

Chapter 5

Formalizing Learning Interactions

In Chapter 4 presented the HiLL paradigm. In the remainder of this thesis, we present how the influences of different interaction types affect downstream learning outcomes. However, the space of interaction types is vast. We enumerated some in Chapter 3.1, but new algorithms and techniques are constantly being developed. It is thus intractable to directly compare each and every extant and proposed interaction type. Still, many interaction types share fundamental similarities in how people interface with them. By taxonomizing the space of interaction types along those lines, we can more tractably compare clusters of interaction types. To this end, this chapter presents a novel taxonomy that characterizes interaction types along four dimensions: the action batch size of the learner's queries, the action batch size of the user's responses, the number of intervention opportunities available to the user per query, and the number of response choices available for a user to select from per query. We also identify four interaction archetypes, termed: *Showing, Categorizing, Sorting, and Evaluating*.

We then use our framework to present an information-gain based overview of how these different interaction types result in nuanced differences in downstream training data.

5.1 Representing Model-Agnostic Interactions

In order to discuss interactions, we define the following terminology. Let a query q refer to data that a human teacher U is prompted to respond to by learning agent L. An *annotation* is the feedback that a teacher gives in response to a query. An *interaction type* is the format of a query (e.g., “Showing”, “Categorizing”, “Sorting”, or “Evaluating”, as introduced in Section 3.1). An *interaction instance* is a specific query, e.g. “Should the label be ‘library’ or ‘bookshop’?” and its associated annotation. Finally, an *interaction session* is a series of interaction instances.

We choose to model interaction sessions as Markov Decision Processes (MDPs) for several reasons. MDPs provide a sequential decision-making paradigm that captures how, in active and passive learning, people provide a series of annotations over the course of an interaction session. Furthermore, with this paradigm, we can treat human teachers as agents making decisions over their own action and state spaces, rather than as oracles in possession of data that is always equally accessible. This allows us to account for imperfect decision-making due to human factors (e.g. cognitive load, usability). Finally, this enables us to analyze interaction types separately from any underlying learning models (e.g. Gaussian process, neural network, Q-learning). The interaction type is a means to obtain data, and the learning model consumes that data. This distinction enables us to discuss interaction types in terms of the user's actions and the learner's actions, to analyze both passive and active data collection, and to assess interactions regardless of learning objectives.

To define this MDP in further detail, we first define a user U as an agent interacting with a learner L via queries. The interactions between U and L can be situated in passive or active learning contexts. Let A_L define the set of actions available to the learner L and $a_t \in A_L$ the action taken at time t. For example, consider a restocking robot tasked with reshelving items. In this case, A_L may consist of the available steering controls needed to construct an appropriate trajectory to reshelve an item. Let $s_t \in S_L$ denote the state at time t (e.g. the position of the robot's end effector). A trajectory is a series of state-action pairs, $\xi = (s_t, a_t)_{t=0}^{T}$, where T is some finite task horizon. This notation holds for one-shot tasks such as accepting or rejecting an image annotation: s_0 is the image, and a_0 is the suggested annotation.

Now, we describe an interaction session I as an MDP $:= (S_U, A_U, \mathcal{T}, \mathcal{R})$. Let A_U define the set of actions available to the user, e.g. the feedback a user can give in a particular interaction type. For example, in a binary feedback interaction, $A_U = \{-1, +1\}$. Note that A_U and A_L need not always be distinct: for demonstrations, the user may have the same action space as the learner (consider, for example, a remote-controlled robot being steered by a user with a joystick). This is sufficient notation for discussing the curation of training sets for passive learning. For active learning, we define the state $\sigma_i \in \mathcal{S}_U$ as the parameterization of L at the ith query, as made visible to the user via means such as model weights. This is distinct from the state of the underlying learner's environment s_t, as discussed previously. The transition $\mathcal{T}$ is a property of L (e.g., gradient descent if the model is a neural network), and can be opaque to the user. The reward $\mathcal{R} : S_U \mapsto \mathbb{R}$ minimizes the difference between the desired and true output of L.

5.2 Features of the Taxonomy of Interaction Types

In this section, we will further explain the column headers in Table 5.1. To do so, we will leverage notation from the MDP formulation we have just presented. We will also provide examples from the reshelving robot task.

Query Size (Actions). The batch size of a query is determined by the number of actions $a \in A_L$ it contains, and is given by $N \cdot T$. A query q consists of one or more trajectories ξ with finite time horizon T such that $q = \{\xi_0, .., \xi_{N-1}\}$, where N is the number of options presented to the user. In the binary feedback interaction we have been referring to, the query presented to the user is a single ($N = 1$) example of a trajectory: $q = \{\xi_0\}$. If we were to instead use a preference interaction, the user would select one of two trajectories presented to them, such that $N = 2$ and $q = \{\xi_0, \xi_1\}$ We note that for demonstrations, the user is not provided with a trajectory (though they may sometimes be provided with a starting state) and is instead asked to provide a trajectory themselves.

Interactions	**Query Size**	**Response Size**	**Intervention Options**	**Response Choice Space**	**References**
Showing					
Demonstrations	0	T	0	$\|A_L\|^T$	[2, 13, 95, 106, 116, 140]
Categorizing					
Labels	T	1	0	$\|A_U\|$	[42, 45, 83]
Binary Feedback	T	1	0	$\|\{-1, +1\}\|$	[40, 76]
Sorting					
Rankings	$T \cdot N$	N	0	$N!$	[19, 52, 136]
Preferences	$T \cdot 2$	2	0	2!	[115]
Evaluating					
Corrections	T	$0 \leq i \leq T$	2^T	$\|A_U\|^i$	[15, 65]
Critiques	T	$0 \leq i \leq T$	2^T	2^T	[37, 57, 66]

Table 5.1: We divide interactions into four clusters. T is the finite time horizon of a presented trajectory (1 in one-shot instances), N is the number of trajectories of length T in a query or response, A_U is the set of user actions, A_L is the set of learner actions, and i is a subset of T.

Response Size (Actions). The number of actions $a \in A_U$ that a user provides in response to a query q is variable in size. In the binary feedback case, the user provides one action: the assignment of either a +1, or a -1. If we were to use a ranking interaction instead, the user would provide n actions by returning a total ordering over the n options presented to them. For example, again using restocking robot example, a user might either rate a potential trajectory for placing an item as good or bad; between five proposed trajectories, they would identify which is the best, second best, third, and so on.

Intervention Options. This quantifies the user's granularity in providing feedback. In both the binary feedback and preference interactions we have used as examples, the expectation is that the user must respond to the entirety of the query q. Therefore, the space of their intervention choices is 0; it is a coarse response. However, some interactions, such as corrections [15] allow the user to select subsets of $q = \{\xi_0\}$ to modify; in the restocking robot case, a user could select the entirety of a reshelving trajectory ξ_0 as good and make no modifications, or adjust some chunk of $i < T$ actions. The user's intervention choice is 2^T because they have the opportunity to intervene at each time step.

Response Choice Space. This is the number of possible responses that a user can provide, given a query q, and is related to A_U. In the binary feedback example, the user can give $|A_U| = |\{-1, +1\}| = 2$ possible responses. More generally, a user has as many response options as they have potential rewards or labels to assign. On the other hand, if we were to use a preference interaction, then $|A_U| = 2$; more generally, in ranking N options, users have $N!$ orderings to choose from.

We recognize that factors such as user interface may affect how users engage with interactions as well. However, this is beyond the scope of this thesis and in subsequent work we simply standardize interaction interfaces as much as possible to minimize any possible effects.

This taxonomy, and the features that comprise it, are largely based on how a user (for our purposes, we refer to a 'user' and a human teacher interchangeably) might perceive and experience differences between interactions. Furthermore, in order to construct this taxonomy, we rely on simplifying assumptions. We exclude the transfer of implicit information as an axis because we believe that people will mostly pay attention to the information that they are explicitly sharing. Additionally, we overlook cognitive shortcuts that individuals might take. For example, when considering Response Choice Space in a sequential decision-making task, users may not be processing all possible trajectories, nor thinking of every word they know in order to caption images. Thus, our big-$\mathcal{O}$ estimates may be too coarse to capture the nuances of a user's *perceived* response space. However, for our purposes, these assumptions suffice to broadly taxonomize the space of possible learning interactions.

5.3 Feedback Interpretation

We can now build on this taxonomy to demonstrate a theoretical example of how different interaction types must necessarily influence training data in distinct ways.

Information gain has been used to select queries in active HiLL systems in studies of individual interaction types, such as when querying a teacher to critique a robot's motion [37] or when querying a teacher to indicate their preference over two proposed actions [18]. Recent work from Jeon et al. [67] proposes a formulation of information gain for finding the best feedback type for reward learning, assuming optimal feedback. We build on these studies and formulate information gain for measuring the effect of interaction types on training data for HiLL systems. Information gain represents the expected change in the model's information entropy (H) resulting from new information. In a HiLL context, this information consists of the training data obtained from one interaction between the teacher and agent.

Each interaction type defines the ways in which the agent may query the teacher for training data and, as a result, defines the number and distribution of possible responses by the human teacher to the agent's query. We ground the problem of calculating the information gain Φ_i for the optimal query of interaction type i as follows:

$$\Phi_i(\omega, s) = \max_{q \in Q_i(\omega,s)} IG(\omega, C_i(q)) \tag{5.1}$$

$$= H(\omega) - \min_{q \in Q_i(\omega,s)} \mathop{\mathbb{E}}_{c \in C_i(q)} \big[H(\omega|c)\big] \tag{5.2}$$

where s is the state in which the query q occurs using interaction type i, and ω represents the random variable for model weights/parameters. This formulation also relies on a function Q that produces a set of queries, and a function C that produces a set of feedback choices, both of which we define later. Here, the notion of state s can be generalized to any input data, such as an image for a visual classification task. In an active learning context, the agent may be able to select the state that maximizes the information gain over ω from its interaction, e.g. by selecting the most informative datapoint to be labeled [70] or changing the behavior of other agents in the environment [114]. Otherwise, the state remains static, and the agent's objective is to select an action query that maximizes the information gain over ω within that state.

Alternatively, information gain can be expressed as the expected Kullback–Leibler divergence of the prior distribution from the posterior belief distribution over model weights:

$$\Phi_i(\omega, s) = \max_{q \in Q_i(\omega,s)} \mathop{\mathbb{E}}_{c \in C_i(q)} \left[D_{\text{KL}}(p(\omega|c)||p(\omega)) \right] \tag{5.3}$$

Both formulations introduce three key, interaction-specific functions: $\mathbf{Q_i}(\omega, \mathbf{s})$, $\mathbf{C_i}(q)$, and $\mathbf{H}(\omega|c)$ (used in Eqn. 2) or $\mathbf{D}_{\text{KL}}(p(\omega|c)||p(\omega))$ (used in Eqn. 3). We describe these functions and their relationship with interaction types in the reminder of this section.

Query: $Q_i(\omega, s)$ A query q is a specific set of data that an agent requests feedback on during a single instance of an interaction. $Q_i(\omega, s)$ is then the set of all possibly queries that can be posed to the teacher, given ω and s. In a **showing** interaction, the agent queries the teacher for an example action, or series of actions (trajectory). Therefore, there is only one possible query in the set $Q_i(\omega, s)$: the agent requests a demonstration from state s without providing any additional information to the teacher. In a **sorting** interaction, the agent's query consists of some n trajectories originating from state s (e.g., the teacher might be asked to order n trajectories with respect to their effectiveness). If we assume that there are k feasible trajectories originating from state s, then $|Q_i(\omega, s)| = \binom{k}{n}$. In both **categorizing** and **evaluating** interactions, which differ on the basis of their *choice space* and *choice implications*, an agent queries the teacher for feedback on a proposed trajectory, and so $|Q_i(\omega, s)| = k$.

Choice Space: $C_i(q)$ Once a query has been selected, the process for expanding a query into a set of possible explicit and implicit choices available to the teacher is also interaction-specific [67, 77]. For example, both a **categorizing** and an **evaluating** interaction consist of querying the teacher by proposing a series of actions (e.g. a motion trajectory for a robot arm, or proposed labels for a set of object images). However, the set of feedback choices available to the teacher in response to an individual query varies by interaction. In the **categorizing** interaction, the teacher may be presented with a set of ± 1 rating choices over the agent's entire proposed sequence of actions. In the **evaluating** interaction, however, the teacher may observe the same sequence of actions but provide feedback at a finer scale, such as ± 1 ratings on *segments* of the agent's manipulation trajectory rather than a single rating over the full trajectory. An alternative **evaluating** interaction may involve providing corrections instead of critiques, enabling the teacher to interrupt the agent's actions in real-time to provide alternative actions. That is, the teacher must choose whether to interrupt the agent's action at each time step, after which they must also choose *what* alternative action the agent should take. Thus, teachers make more feedback choices in response to a single evaluation query than a single categorizing query.

Overall, these examples illustrate the effect of interaction type on the choice set available to the teacher. These effects are apparent both across different interaction types (e.g., the density of the feedback resulting from a categorizing interaction versus an evaluating interaction), as well as within the same interaction type (e.g., critiques and corrections are both evaluating-type interactions, but result in different feedback choices that are available to the teacher).

Furthermore, the likelihood of the choice set containing the optimal choice is dependent on the *quantity* and *quality* of that set. For interaction types that provide an *infinite set* of query responses, such as a **showing** interaction, the teacher may provide feedback from an infinite set of options. In **evaluating** interactions, the teacher is also provided an additional option of whether to provide feedback or not. As a result of the infinite *quantity* of choices, the optimal choice must be contained within this set of options.

For interaction types that provide a *finite set* of query responses, such as **sorting** interactions, the quantity of choices available to the teacher are limited, and so the training data is dependent on the quality of the choices presented to the teacher. The quality of a choice set may be defined by the informativeness of each possible choice, estimated through an information gain formulation [18].

Choice Implications: $H(\omega|c)$ **or** $D_{\mathbf{KL}}(p(\omega|c)||p(\omega))$ The implications of the teacher's choice on the agent's training data is also dependent on the interaction type. In an information gain context, this implication can be represented as the conditional entropy over the model's parameters given the feedback that the teacher did and did not provide [67]. When leaning from **showing** interactions, such as demonstrations, existing work in inverse reinforcement learning typically assumes that the teacher's feedback represents the optimal action that the agent should take and updates the agent's reward model accordingly [2, 106]. The demonstrations may also be used to learn a nonlinear cost function that represents the dynamics of the demonstrated task [47].

In **categorizing** interactions, the teacher's feedback may be used to directly learn a regression model of the reward function that replicates their feedback (as shown by the TAMER framework [75, 133]). By training an action model separately from the reward model, improvements in the action model may be used to guide the agent's queries to improve its reward model [39]. However, feedback does not always reflect the reward of the agent's state. Thomaz et al. [131] show how categorizing feedback not only reflects the teacher's feedback on the agent's prior actions, but also feedback on their expectations of the agent's future behavior. As a result, a key challenge is determining which states and/or state features correspond to the teacher's feedback [75].

In **sorting** interactions, the training implications of the teacher's choice is dependent on the other choices available to them.

A pairwise preference between two actions may be interpreted as a loss function representing the margin between the agent's predicted preference over the two options (according to its reward function) and the human's actual preferences [34]. As a result, the objective of the model is not necessarily to estimate an action's reward itself, but rather, to learn a reward function that preserves the relative ranking of one action over another [84]. Learning from relative rankings has an added benefit: by removing the assumption that either of the ranked options is optimal, the model can learn a reward function that exceeds the performance of the teacher [23]. Since the strength of the teacher's preference is unknown, it may be beneficial to provide an option to indicate equal preference between two options rather than force the teacher to indicate a preference [62].

In **evaluating** interactions, the teacher provides feedback over a series of proposed or executed actions by the agent. This feedback must be considered with respect to the actions before and after the teacher's feedback. For example, Celemin and Ruiz-del Solar [30] presents a method for approximating the magnitude of the teacher's binary feedback based on the variability of that feedback over time. In corrective interactions, the teacher's feedback can be interpreted as an alternate demonstration that results in higher reward or performance than the agent's originally proposed action. This correction can be used to update the agent's behavior in real-time [14] or interpreted as a singular sample of the desired change in the agent's model (reinforced through additional corrections) [50].

We recognize that our lens focuses on the training implications of a human teacher's explicit responses to an agent's queries. However, a teacher may provide additional data that may be incorporated into the agent's training process. For example, they may also reveal additional, implicit information via gestures [20], facial expressions [38], gaze [139], or other social cues. The teacher's lack of explicit feedback in some states may also provide implicit data, such as when ignoring a web link or skipping a video [17]. Interpreting a teacher's silence as positive feedback may speed up learning [29]; however, the direction and magnitude of reward implied by a teacher's silence is likely to be domain-specific. Furthermore, higher-level information about the task may be learned implicitly through multiple interactions [97]. Leveraging both implicit and explicit information may result in increased informativeness of an interaction without asking of any additional effort from the teacher.

Chapter 6

Leveraging Multiple Interaction Types for Better Learning Outcomes

We have shown that different interaction types convey nuanced explicit and implicit information that affects the training data that can be collected. In the remainder of this thesis, we will continue to frame training data as resulting from *choices* made by a human teacher in response to an agent's query. This perspective underscores the importance of developing learning approaches that support learner efficacy while being more attuned to human teachers. A first step in this direction is to be able to learn dynamically from multiple interaction types.

This chapter presents work on an active learning algorithm that allows an agent to do exactly this by reasoning over the response choices made by a teacher in conjunction with task understanding. We present a unifying formalism across different interaction types, an extension of our previous information-gain formulation that can be used to learn from multiple interaction types in one session, an algorithm that leverages that information gain formulation, and evaluations of our approach. The work presented in this chapter is in collaboration with Tesca Fitzgerald, Patrick Callaghan, and Russell Wong; the major contributions from this thesis lie in the theoretical construction of the optimization function, the formulation of the query, choice spaces, and implications for the interaction types, as well as contributions to the evaluations.

6.1 INQUIRE: INteractive Querying for User-aware Informative REasoning

This work builds upon the framework presented in Section 5.3 in order to develop an algorithm that can dynamically learn from multiple interaction types in one setting. Such an algorithm could improve both the quality and quantity of data collected by acquiring a more diverse set of examples, as well as querying a human teacher more efficiently. Apart from a few exceptions [19, 26, 103], most prior work in active learning assumes that a single interaction type is used throughout a learning session. However, it is likely that the optimal interaction type varies with: the learning agent's changing knowledge of the task, the states from which the learning agent can query the human teacher, and costs specific to the domain and interaction (e.g. the effort needed to provide a response [77]).

We developed INQUIRE (Alg. 1), an active learning system that optimizes over both the interaction type and content of its queries based on its current task knowledge and needs. At a high level, INQUIRE repeatedly: computes an optimal query and interaction type based on the expected information gain of that query given its current task knowledge, poses that query to a teacher and stores their feedback, and finally updates its task knowledge to maximize the likelihood of all of the feedback it has received thus far.

The remainder of this section further details the underlying information-gain formulations and optimization functions on which INQUIRE is built, provides an overview of the algorithm, and demonstrates in evaluations across domains and baselines that INQUIRE improves task performance (particularly when an agent may need to handle repeated states that may become low-information if using a single interaction type). In addition, INQUIRE includes a cost metric that can be used to represent constraints such as the cognitive load on a teacher or the difficulty of providing a response for a given task. In Chapter 7, we will further evaluate the costs of different interaction types on teaching performance.

6.1.1 Problem Setup

We consider sequential decision-making problems that are both deterministic and fully observable. Real-world examples of such problems might be assistive feeding tasks, and longitudinal models of recommender systems. We are interested in leveraging feedback from a human teacher in order to recover a reward function that leads to desirable agent behavior in a particular task domain. To do so, we make the standard assumption that a reward is given by a linear combination of feature weights, $r(t) = \phi(t) \cdot \omega$. This reduces the reward-learning problem to learning a distribution $\mathcal{W}$ over the set of possible feature weights. At the outset, we can assume a uniform prior. Our learning algorithm will be able to utilize one specific interaction type from each of the four interaction archetypes (*Showing*, *Categorizing*, *Sorting*, and *Evaluating*) we have been discussing thus far:

Demonstration. This is a *Showing* type interaction, wherein the agent presents a starting state s and requests a trajectory $t \in \mathcal{T}(s)$ from a human teacher. Here, $\mathcal{T}(s)$ represents all possible trajectories originating from s. Thus the agent's query space can be represented as $Q_{\text{demo}} = \{\mathcal{T}(s), \forall s\}$. Correspondingly, the human teacher's choice space is given by $C_{\text{demo}}(q) = \mathcal{T}(s)$ because they can provide any trajectory t within the given trajectory space.

Preference. This is a *Sorting* type interaction, where there are only two candidates presented to a human teacher. Here, the agent's query space is defined as $Q_{\text{pref}} = \mathcal{T}(s) \times \mathcal{T}(s)$. This is because any individual query consists of two possible trajectories, i.e. $q_{\text{pref}} = \{t_a, t_b\} | t_a, t_b \in \mathcal{T}(s)$. The corresponding choice space for the human teacher is therefore $C_{\text{pref}}(q) = q = \{t_a, t_b\}$.

Correction. This is an *Evaluating* type interaction wherein the agent demonstrates a possible trajectory, and the human teacher suggests corrections to that trajectory. The agent's query space is given by $Q_{\text{corr}} = \mathcal{T}(s)$ from which it demonstrates one $q \in \mathcal{T}(s)$. The teacher's corresponding choice space is $C_{\text{corr}}(q) = \mathcal{T}(s)$.

Table 6.1: Each interaction involves separate query spaces, choice spaces, and choice implications.

	Query Space $Q_i(s)$	**Query** $q \in Q_i(s)$	**Choice Space** $C_i(q)$	**Choice Implication** $c \in C_i(q) \implies (c^+, c^-)$
Demo.	$\{T\}$	T	T	$c^+ : t \in T \quad c^- : T \setminus t$
Pref.	T x T	$\{t_0, t_1\}, t_0, t_1 \in T$	$\{t_0, t_1\}$	$c^+ : t \in q \quad c^- : q \setminus c^+$
Corr.	T	$t \in T$	T	$c^+ : t' \in T \quad c^- : q$
Binary	T	$t \in T$	$\{0, 1\}$	$c = 0 \implies c^+ : T \setminus q \quad c^- : q$ $c = 1 \implies c^+ : q \quad c^- : T \setminus q$

```
Algorithm 1 INQUIRE - Overview
Input: Set of query states $S$
Parameters: $K$ (# of queries), $\mathcal{I}$ (interaction types)
Output: Weight vector $\omega^*$

$\mathbf{F} \leftarrow \{\}$
$\mathbf{\Omega} \leftarrow M$ random initial weight vectors
for $K$ iterations do
    $s \leftarrow$ next query state in $S$
    $q_i^* \leftarrow$ generate_query($s, \mathcal{I}, \mathbf{\Omega}$)        (Alg. 2)
    $\mathbf{F} \leftarrow \mathbf{F} \cup \{$query_teacher($q_i^*$)$\}$
    $\mathbf{\Omega} \leftarrow$ update_weights($\mathbf{F}$)
$\omega^* \leftarrow$ mean($\mathbf{\Omega}$)
return $\omega^*$
```

```
Algorithm 2 INQUIRE - Generate Query
Input: $s$ (state), $\mathcal{I}$ (interaction types), $\mathbf{\Omega}$ (weight samples)
Output: Query $q^*$
$\mathbf{T} \leftarrow$ uniformly_sample_trajectories($s$)
Compute $\mathbf{E}$ : $\{\mathbf{E}_{t,t',\omega}, \forall t, t' \in \mathbf{T}, \omega \in \mathbf{\Omega}\}$   (Eq. 6.6)
for each interaction type $i \in \mathcal{I}$ do
    $\mathbf{Q} \leftarrow Q_i(s)$                              (See Table 1)
    $\mathbf{C} \leftarrow \{C_i(q), \forall q \in \mathbf{Q}\}$                 (See Table 1)
    Compute info gain matrix $\mathbf{G}^{(\mathbf{i})}$ from $\mathbf{E}$ (Eq. 6.11)
    $q \leftarrow \arg\max_{q'} \sum_{c \in \mathbf{C}_{q'}, \omega \in \mathbf{\Omega}} \mathbf{G}^{(\mathbf{i})}_{q',c,\omega}$
    $g \leftarrow \frac{1}{\log(\lambda_i)} \sum_{c \in \mathbf{C}_q, \omega \in \mathbf{\Omega}} \mathbf{G}^{(\mathbf{i})}_{q,c,\omega}$
    if information gain $g > g^*$ then
        $g^* \leftarrow g$
        $q^* \leftarrow q$    {Store query with highest info. gain}
return $q^*$
```

Binary Feedback. This is a *Categorizing* type interaction wherein an agent proposes a trajectory and the human teacher can only provide a reward signal of ± 1. Here, the agent's query space is $Q_{\text{bnry}} = \mathcal{T}(s)$ and the teacher's corresponding choice space is $C_{\text{bnry}}(q) = \{-1, +1\}$.

Let us also further define the notion of *choice implications*, which we discussed at length in Section 5.3. We decompose a teacher's choice $c \in C(q)$ into a set of accepted and rejected trajectories, c^+ and c^-. This decomposition allows us to calculate information gain, which we will discuss and use in a later section. Furthermore, we note that the set of all possible trajectories starting at state s, i.e. $\mathcal{T}(s)$ is potentially infinite. In order to perform tractable computations, we instead use the approximated set T, which contains N trajectories sampled from the starting state s. Table 6.1 summarizes the previous interaction type definitions and their associated implications.

6.1.2 Updating Weights Given Feedback

We will first briefly discuss how INQUIRE updates feature weights based on feedback received from a human teacher, before proceeding to talk about how to generate an optimal query. In order to learn from multiple interaction types, we keep a set of all feedback that has been received thus far. We denote this set **F**. We then want to update ω such that the likelihood of **F** is maximized:

$$\omega^* = \arg\max_{\omega} \prod_{c \in \mathbf{F}} P(c|\omega) \tag{6.1}$$

$$= \arg\max_{\omega} \prod_{c \in \mathbf{F}} \frac{\sum_{t \in c^+} e^{\beta \cdot \phi(t) \cdot \omega}}{\sum_{t \in c^+ \cup c^-} e^{\beta \cdot \phi(t) \cdot \omega}} \tag{6.2}$$

Note that this expansion is a result of using the common Boltzmann-rational choice model:

$$P(c|\omega) = \frac{\sum_{t \in c^+} e^{\beta \cdot \phi(t) \cdot \omega}}{\sum_{t \in c^+ \cup c^-} e^{\beta \cdot \phi(t) \cdot \omega}} \tag{6.3}$$

In this equation, $\phi(t)$ gives the sum over the features weights of all states that comprise a given trajectory t. β is a parameter representing the expected optimality of the teacher's feedback with respect to ω. The particular gradient update we use to adjust ω after every piece of feedback can be found, with its associated derivation, in [51].

6.1.3 Optimal Query Generation.

We build off of the framing presented in Section 5.3 and solve the following optimization problem for a given interaction type i:

$$q_i^* = \arg\max_{q \in Q_i(s)} \mathbb{E}_{c|C_i(q)} \left[\text{IG}(\mathcal{W} \mid c) \right] \tag{6.4}$$

$$= \arg\max_{q \in Q_i(s)} \sum_{c \in C_i(q)} \sum_{w \in \Omega} \left[P(c|w) \cdot \log \frac{M \cdot P(c|w)}{\sum_{w' \in \Omega} P(c|w')} \right] \tag{6.5}$$

In other words, we frame an optimal query as one that greedily maximizes the agent's expected information gain over $\mathcal{W}$ after receiving feedback from the user. In this equation, $Q_i(s)$ is a function that returns the query space for interaction type i at state s, and $C_i(q)$ returns the choice space corresponding to a particular query $q \in Q_i(s)$.

It requires a significant amount of effort to make computing Eq. 6.4 tractable, and this effort is the work of our collaborators, especially Tesca Ftizgerald, on [51]. We do not delve into the derivations or justifications of the following equations here; suffice it to say that they are necessary for understanding Alg. 1 and Alg. 2 and for INQUIRE to function. In the remainder of this subsection, we provide a brief overview of the approaches used to make INQUIRE computationally feasible, and the equations needed to understand Alg. 2. We strongly encourage reading [51] for additional context and details.

Because there are potentially infinite trajectories that originate at any starting state s (i.e. $\mathcal{T}(s)$), we must instead use the approximated set T that consists of M sampled trajectories (and let Ω be that set of sampled trajectories). To further increase tractability, we reformulate Eq. 6.5 as a computation over a series of tensors including a tensor of exponentiated rewards (Eq. 6.6), a series of probability tensors(Eq. 6.7 - Eq. 6.10), and an information gain tensor (Eg. 6.11).

$$\mathbf{E}_{t,t',\omega} = e^{\beta\cdot\phi(t')\cdot\omega} \qquad \Longrightarrow \qquad \left[\mathbf{E}+\mathbf{E}^{\mathbf{T}}\right]_{t,t',\omega} = e^{\beta\cdot\phi(t')\cdot\omega} + e^{\beta\cdot\phi(t)\cdot\omega} \tag{6.6}$$

$$\mathbf{P}^{(\text{demo})}_{q,c,\omega} = \left[\mathbf{E}_0 \oslash \sum_{t\in T} \mathbf{E^T}_t\right]_{c,\omega} \qquad \text{(since } |Q| = 1 \text{ for demonstrations)} \tag{6.7}$$

$$\mathbf{P}^{(\text{pref})}_{q,c,\omega} = \left[\left(\mathbf{E}\oslash(\mathbf{E}+\mathbf{E}^{\mathbf{T}})\right)^{\mathbf{T}}, \mathbf{E}\oslash(\mathbf{E}+\mathbf{E}^{\mathbf{T}})\right]_{c,q_0,q_1,\omega} \qquad (c\in\{0,1\} \text{ for prefs.}) \tag{6.8}$$

$$\mathbf{P}^{(\text{corr})}_{q,c,\omega} = \left[\mathbf{E}\oslash(\mathbf{E}+\mathbf{E}^{\mathbf{T}})\right]_{q,c,\omega} \tag{6.9}$$

$$\mathbf{P}^{(\text{bnry})}_{q,c,\omega} = \left[1 - \left(\mathbf{E}_0 \oslash \alpha \sum_{t\in T} \mathbf{E^T}_t\right), \mathbf{E}_0 \oslash \alpha \sum_{t\in T} \mathbf{E^T}_t\right]_{c,q,\omega} \qquad (c\in\{0,1\} \text{ for binary rewards}) \tag{6.10}$$

Here, $\mathbf{P}^{(i)}_{q,c,\omega}$ represents the probability that a teacher selects $c \in C(q)$ as their response to q, given the sampled weights from $\mathbf{\Omega}$. Further note that $\oslash$ represents an element-wise division of two matrices (i.e., $(\mathbf{A}\oslash\mathbf{B})_{ij} = \mathbf{A}_{ij}/\mathbf{B}_{ij}$) and α is a normalization factor such that $\sum_c \mathbf{P}^{(\text{bnry})}_{q,c,\omega} = 1$.

Given $\mathbf{E}$ and $\mathbf{P}$, as well as our set of trajectory samples, we can directly compute for the optimal query q_i^* for an interaction type i.

$$\mathbf{G}^{(i)}_{q,c,\omega} = \mathbf{P}^{(i)}_{q,c,\omega} \cdot \log\left(\frac{M\cdot \mathbf{P}^{(i)}_{q,c,\omega}}{\sum_{\omega'\in\Omega} \mathbf{P}^{(i)}_{q,c,\omega'}}\right) \qquad\qquad q_i^* = \arg\max_q \sum_{c,\omega} \mathbf{G}^{(\mathrm{i})}_{q,c,\omega} \tag{6.11}$$

Finally, we can use this to directly solve for the optimal interaction type as well. We can perform both *unweighted* and *cost-weighted* optimizations using the parameter λ_i. Costs can be user-defined to account for factors such as the expected time to answer a query; eventually, more difficult to quantify factors such as cognitive load may be usable as well. To perform an *unweighted* optimization, let λ_i be constant across all interaction types.

$$i^* = \arg\max_{i\in\mathcal{I}} \frac{1}{\log(\lambda_i)} \sum_{c,\omega} \mathbf{G}^{(\mathrm{i})}_{q_i^*,c,\omega} \tag{6.12}$$

These definitions are sufficient for understanding the summary provided in Alg. 2. Again, we thank our collaborators for their efforts in deriving these equations, and refer readers to [51] for more details.

6.2 Oracle Implementation

To evaluate INQUIRE in a controlled fashion, we leverage an oracle teacher. The oracle teacher, similar to INQUIRE, requires its own set of trajectory samples. It then selects a response to a query via one of three mechanisms: returning the highest-reward trajectory from its choice space (demonstrations/preferences), rejection sampling of trajectories followed by selection of the trajectory with the highest reward-to-distance ratio from the queried trajectory (corrections), and returning whether a query meets or exceeds a reward threshold (binary reward). The use of an oracle agent, as opposed to a person, allows us to have more control over our evaluations, and to understand the limitations of our approach. Furthermore, it allows us to disregard confounds from inconsistent or suboptimal teaching behavior and purely evaluate our approach as a proof-of-concept. Implementation details are presented below, and we thank our collaborator Russell Wong for both implementing this oracle agent and writing the text used in this section (also found in the Appendix of [51]).

When responding to a query, the oracle requires its own set of trajectory samples. Similar to INQUIRE, we derive this set by uniformly sampling N trajectories; however, the two sample sets are kept separate, and so we distinguish the oracle's trajectory set as T' (resampled for each query state).

Demonstration/Preferences The oracle returns the highest-reward trajectory (according to ω^*) from a uniformly-sampled trajectory set T' (for demonstrations) or from the pair of queried trajectories $C(q)$ (for preferences):

$$\text{Oracle}_{\text{demo}}(q) = \arg\max_{t \in T'} \left(\phi(t) \cdot \omega^*\right) \qquad \text{Oracle}_{\text{pref}}(q) = \arg\max_{t \in C(q)} \left(\phi(t) \cdot \omega^*\right) \quad (6.13)$$

Corrections The oracle produces T' by performing rejection sampling; it uniformly samples trajectories and accepts only those with a reward greater than or equal to the queried trajectory q until T' contains N trajectories:

$$\forall t \in T', \phi(t) \cdot \omega^* \geq \phi(q) \cdot \omega^* \quad (6.14)$$

After producing this trajectory set, the oracle selects the trajectory with the highest ratio of reward-to-distance from the queried trajectory:

$$\text{Oracle}_{\text{corr}}(q) = \arg\max_{t \in T'} \frac{\Delta_r(q,t)}{\Delta_d(q,t)} \quad (6.15)$$

$$\Delta_r(q,t) = \min_{t' \in T'} \frac{\phi(t) \cdot \omega^* - \phi(q) \cdot \omega^*}{\phi(t') \cdot \omega^* - \phi(q) \cdot \omega^*} \qquad \Delta_d(q,t) = \min_{t' \in T'} \frac{e^{\delta(t,q)}}{e^{\delta(t',q)}} \quad (6.16)$$

The distance metric δ between two trajectories is domain-specific. We will provide additional details on these distance metrics in Section 6.3.

Binary Reward The oracle produces T' by uniformly sampling N trajectories and produces a cumulative distribution R over ground-truth rewards for T'. It then selects a positive or negative reward indicating whether the agent's query q meets or exceeds a threshold percentile

α with respect to the sampled trajectories:

$$R = \{\omega^* \cdot \phi(t), \forall t \in T'\} \qquad \text{Oracle}_{\text{bnry}}(q) = \begin{cases} + & R(\omega^* \cdot \phi(q)) \geq \alpha \\ - & \text{otherwise} \end{cases} \tag{6.17}$$

We set $\alpha = 0.75$ in our experiments.

6.3 Results

We simulate four types of learning problems in robotics using the oracle teacher described in Section 6.2. The **Parameter Estimation** domain involves directly estimating a randomly-initialized, ground truth weight vector ω^* containing 8 parameters. The **Linear Dynamical System** domain, inspired by [18], simulates a controls problem and involves learning 8 parameters. The **Lunar Lander** domain [21] simulates a controls problem involving 4 parameters. The **Pizza Arrangement** domain simulates a preference-learning problem involving 4 parameters. Each domain (except for Parameter Estimation) has a *static*-state and *changing*-state condition indicating whether the robot must formulate all queries from the same query state or not, respectively.

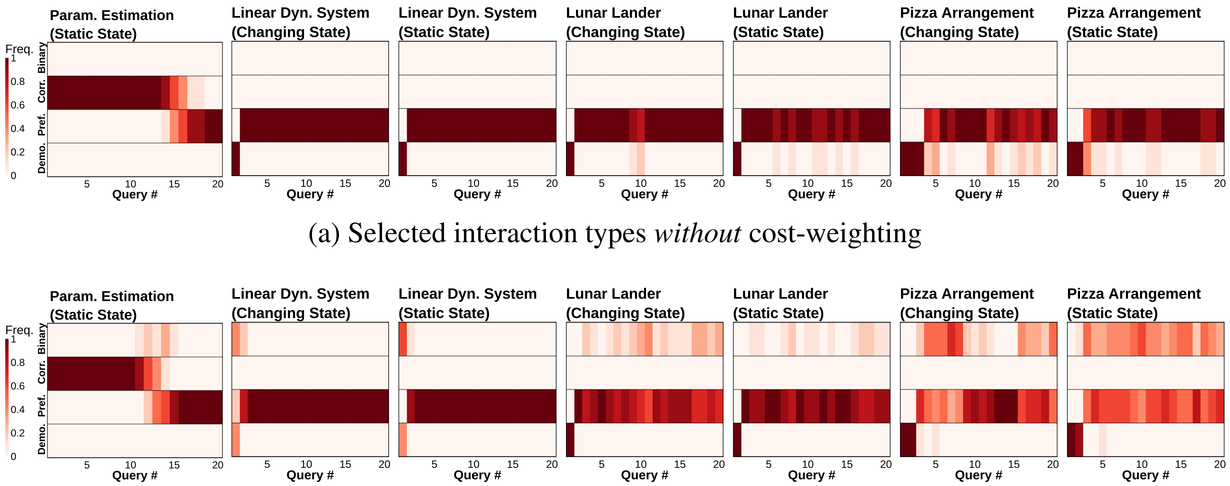
In the Parameter Estimation domain, we define the distance metric δ between two trajectories as the angular distance between the two parameter vectors. In the Linear Dynamical System and Lunar Lander domains, we define δ as the normalized distance between the two trajectories' aligned x and y poses over time. We use the DTW-Python package [53] to align trajectories via Dynamic Time Warping and return their normalized distances. In the Pizza Arrangement domain, we define δ as the Euclidean distance between two toppings. For the full evaluation procedure and oracle implementation details for each domain please see [51]. We thank our collaborators on this paper for contributing implementations and write ups of these domains and evaluations, which are presented in the remainder of this section.

Note that we set $\beta = 20$ across all interaction types (determined by empirical evaluation).

6.3.1 Query Selection

We first analyze how INQUIRE selects queries. Figure 6.1 reflects the changes in interaction types selected by INQUIRE over time. Figure 6.1a first reports these interaction selections in an unweighted query optimization setting, where all interaction types are assumed to be equally costly. In the parameter optimization domain, INQUIRE requests corrections in the first 14-18 queries and then requests preferences as the remaining queries. Demonstrations were not enabled in this domain. In all other domains, INQUIRE requests a demonstration as its first query, then immediately switches to requesting preferences for the remaining queries (occasionally alternating between preferences and demonstrations in the Lunar Lander domain).

After assigning different cost values to each interaction type, INQUIRE chooses more diverse interaction types in order to maximize its information-to-cost ratio. We assign a cost of 20 to each demonstration, 15 to each correction, 10 to each preference, and 5 to each binary query; these numbers are somewhat arbitrary, but the ordering is based on the number of actions a human teacher would need to provide in each format. As shown in Figure 6.1b, this typically results in



(a) Selected interaction types *without* cost-weighting

(b) Selected interaction types *with* cost-weighting

Figure 6.1: Heatmaps illustrating how INQUIRE selects different interaction types as it learns more over time. These selections differ when deriving unweighted (top) or cost-weighted (bottom) information gain estimations. In the cost-weighted setting (bottom), INQUIRE selects more low-cost binary queries than it does in the unweighted setting (top).

INQUIRE posing more binary queries due to their relatively low cost. This pivot toward binary queries may occur at the start (as seen in the linear dynamical system), middle (as seen in the parameter estimation domain), or interspersed throughout the learning process (as seen in the Lunar Lander domain).

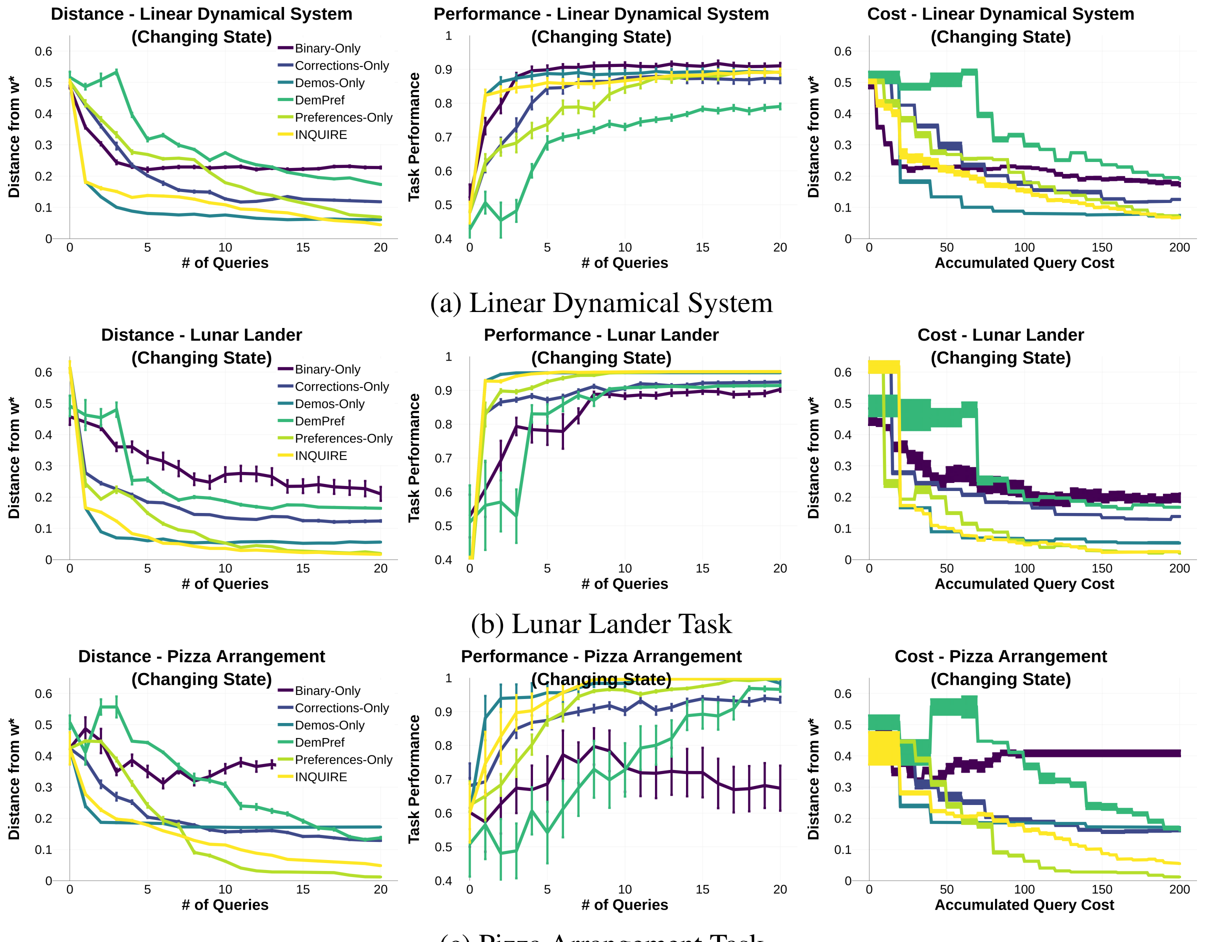
6.3.2 Learning Performance

We now analyze the effect of INQUIRE's interaction type selections on its learning performance and compare to two types of baselines. The first, DemPref [103], learns from 3 demonstrations and then learns from preference queries by using a volume removal objective function. As our second baseline, we compare INQUIRE against agents that use only one form of interaction: demonstrations, preferences, corrections, or binary reward. Note that the preference-only agent is formulated according to [18] and thus represents this baseline method.

We first consider the changing-state formulation of each domain, where the robot is presented with a new state for each query. Since the Parameter Estimation domain does not contain states, we exclude it from this first set of results. Figure 6.2 illustrates this learning performance in the Linear Dynamical System and Lunar Lander domains according to three key metrics.

Distance measures the angular distance between the ground truth feature weights (ω^*) and the algorithm's estimated feature weight $\tilde{\omega}$ after each query.

Performance measures the task reward achieved using a trajectory optimized according to $\tilde{\omega}$ (the algorithm's estimated feature weight after each query). Performance is scaled between 0-1, with 0 and 1 representing the worst and best possible task rewards according to ω^*, respectively. Note that INQUIRE's distance and performance metrics are achieved in the unweighted condition.



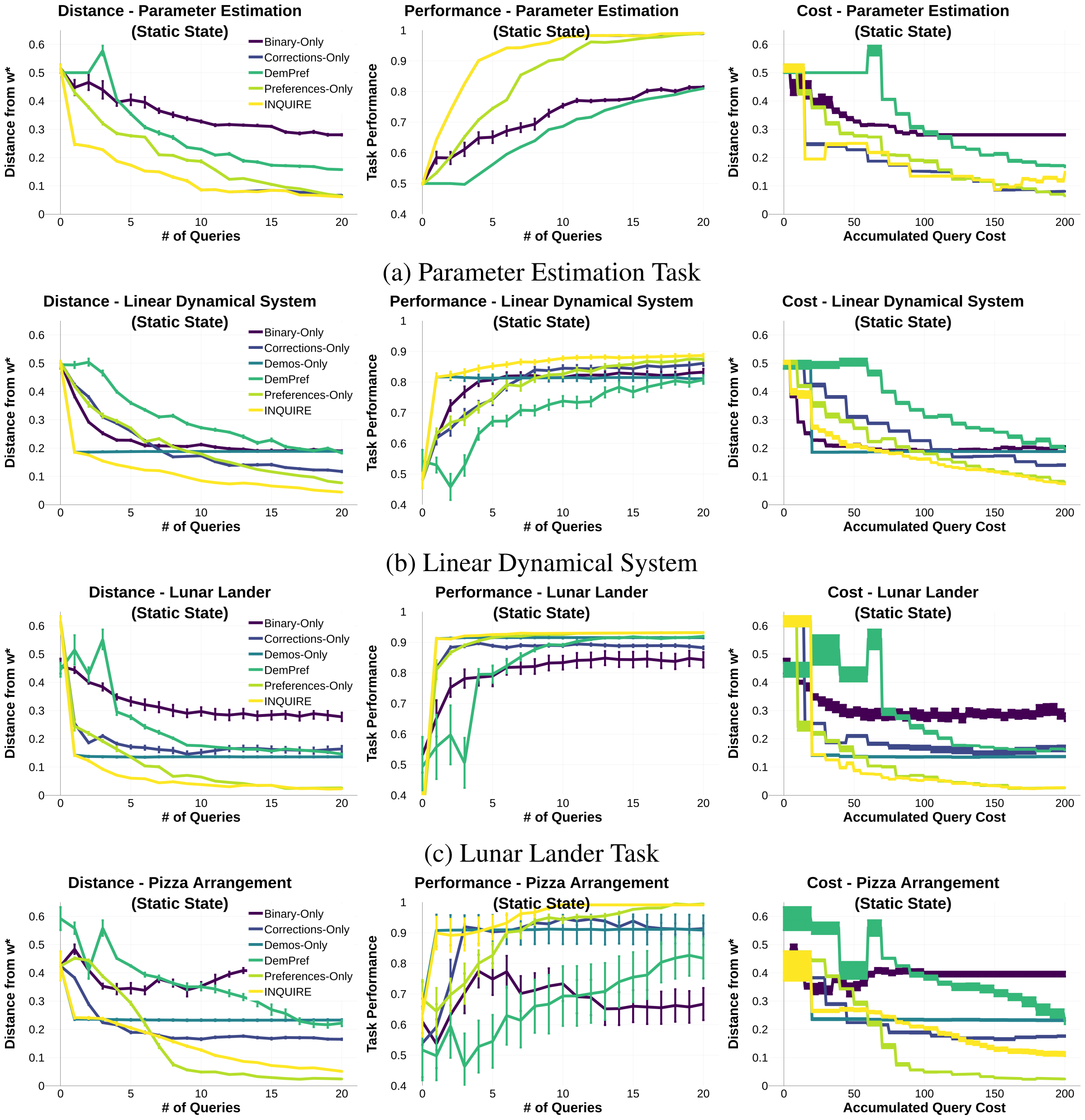
(a) Linear Dynamical System

(b) Lunar Lander Task

(c) Pizza Arrangement Task

Figure 6.2: Metrics for the *changing state* condition in which the robot's initial state changes with each query. Error bars/regions represent variance across multiple evaluation runs with randomized query states and initial weights. Cost metrics are cut off after 20 queries for the *binary-only* method in (c) due to extensive computation times.

Cost-vs-Distance measures the relationship between the cumulative cost of each query and the resulting distance between $\tilde{\omega}$ and ω^* after each query. INQUIRE's metrics in this graph are achieved in the cost-weighted condition.



(a) Parameter Estimation Task

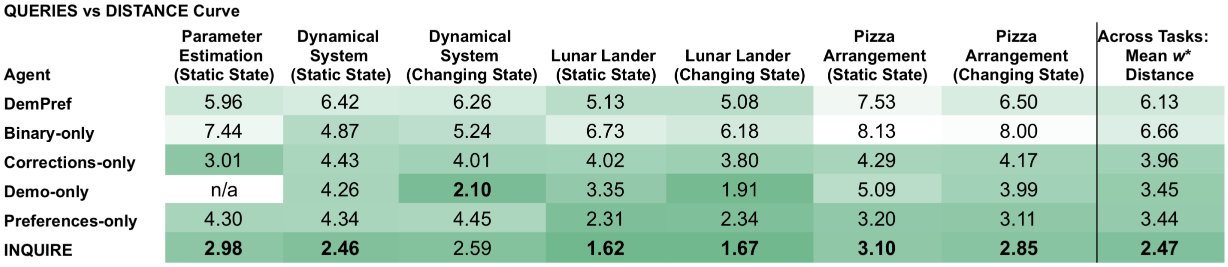
(b) Linear Dynamical System

(c) Lunar Lander Task

(d) Pizza Arrangement Task

Figure 6.3: Metrics for the *static state* condition in which the robot is presented with the same state for all 20 queries. Error bars/regions represent variance across multiple evaluation runs with randomized query states and initial weights. Cost metrics are cut off after 20 queries for the *binary-only* method in (a) and (d) due to extensive computation times.

Figure 6.3 presents the same three metrics for the *static-state* condition in which all 20 queries must be selected from the same initial state. Finally, we quantify these graphs by reporting the area-under-the-curve (AUC) metrics for the distance, performance, and cost curves across all tasks. These metrics are shown in Figure 6.4 - 6.9. The AUC metrics indicate that, compared to the baseline methods, INQUIRE results in the best average learning performance (measured both by the distance and performance plots in Figures 6.2-6.3) across all domains and dominates learning performance in the static-state domains. INQUIRE also results in the best average distance-to-cost ratio across all domains.



QUERIES vs DISTANCE Curve

Agent	**Parameter Estimation (Static State)**	**Dynamical System (Static State)**	**Dynamical System (Changing State)**	**Lunar Lander (Static State)**	**Lunar Lander (Changing State)**	**Pizza Arrangement (Static State)**	**Pizza Arrangement (Changing State)**	**Across Tasks: Mean *w** Distance**
DemPref	5.96	6.42	6.26	5.13	5.08	7.53	6.50	6.13
Binary-only	7.44	4.87	5.24	6.73	6.18	8.13	8.00	6.66
Corrections-only	3.01	4.43	4.01	4.02	3.80	4.29	4.17	3.96
Demo-only	n/a	4.26	**2.10**	3.35	1.91	5.09	3.99	3.45
Preferences-only	4.30	4.34	4.45	2.31	2.34	3.20	3.11	3.44
INQUIRE	**2.98**	**2.46**	2.59	**1.62**	**1.67**	**3.10**	**2.85**	**2.47**

Figure 6.4: AUC values for the distance plots in Figs 6.2-6.3. Darker cells indicate lower (better) values.

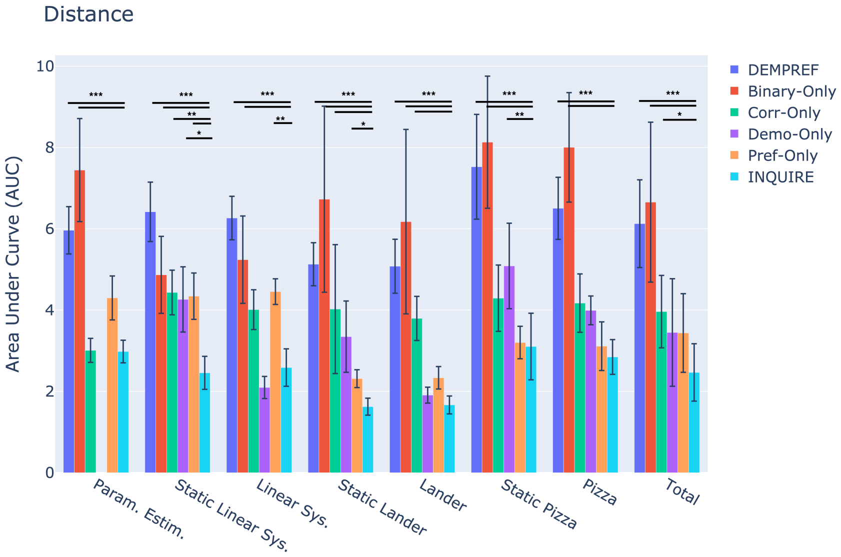


Figure 6.5: Visualizing Fig. 6.4, with statistical significance noted. (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$)

QUERIES vs PERFORMANCE Curve

Agent	Parameter Estimation (Static State)	Dynamical System (Static State)	Dynamical System (Changing State)	Lunar Lander (Static State)	Lunar Lander (Changing State)	Pizza Arrangement (Static State)	Pizza Arrangement (Changing State)	Across Tasks: Mean Performance
DemPref	13.95	14.69	14.47	17.22	17.37	14.03	15.53	15.32
Binary-only	15.01	16.54	18.37	16.85	17.37	14.29	14.65	16.15
Corrections-only	19.14	16.53	17.19	18.02	18.33	18.49	18.56	18.04
Demo-only	n/a	16.76	**18.15**	18.61	19.32	18.86	**20.10**	18.63
Preferences-only	17.74	16.55	16.74	18.63	19.05	18.77	18.89	18.05
INQUIRE	**19.15**	**17.81**	17.83	**18.86**	**19.33**	**19.88**	19.79	**18.95**

Figure 6.6: AUC values for the performance plots in Figs 6.2-6.3. Darker cells indicate higher (better) values.

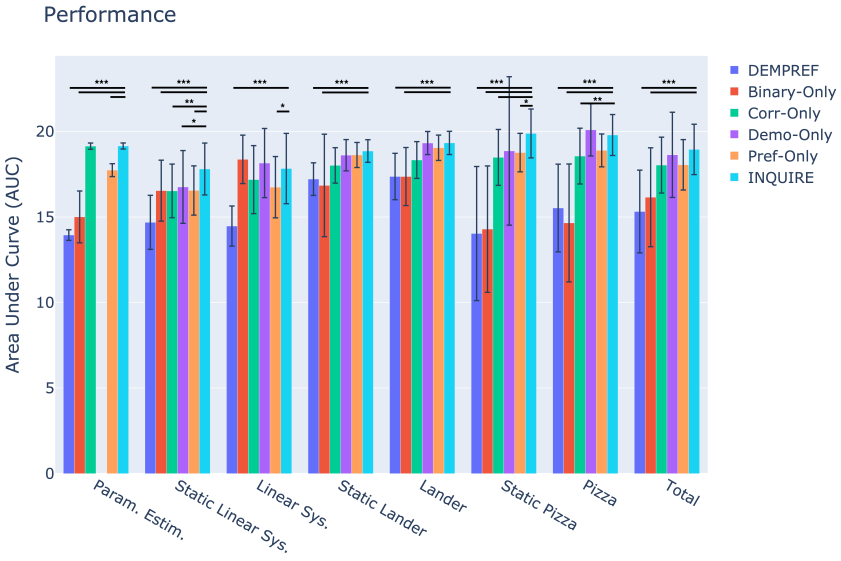


Figure 6.7: Visualizing Fig. 6.6, with statistical significance noted. (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$)

COST vs DISTANCE Curve

Agent	Parameter Estimation (Static State)	Dynamical System (Static State)	Dynamical System (Changing State)	Lunar Lander (Static State)	Lunar Lander (Changing State)	Pizza Arrangement (Static State)	Pizza Arrangement (Changing State)	Across Tasks: Mean Cost/Distance
DemPref	68.26	71.59	70.52	59.16	58.41	82.21	74.20	69.19
Binary-only	64.15	43.18	44.85	61.30	49.84	78.62	79.20	60.16
Corrections-only	**36.39**	51.68	46.02	41.91	42.57	45.13	46.63	44.33
Demo-only	n/a	43.99	**27.94**	37.09	26.07	50.82	42.26	38.03
Preferences-only	42.41	42.73	43.92	**22.90**	**23.18**	**31.81**	**31.01**	33.99
INQUIRE	37.68	**36.39**	35.92	22.98	23.71	33.99	36.48	**32.45**

Figure 6.8: AUC values for the cost plots in Figs 6.2-6.3. Darker cells indicate lower (better) values.

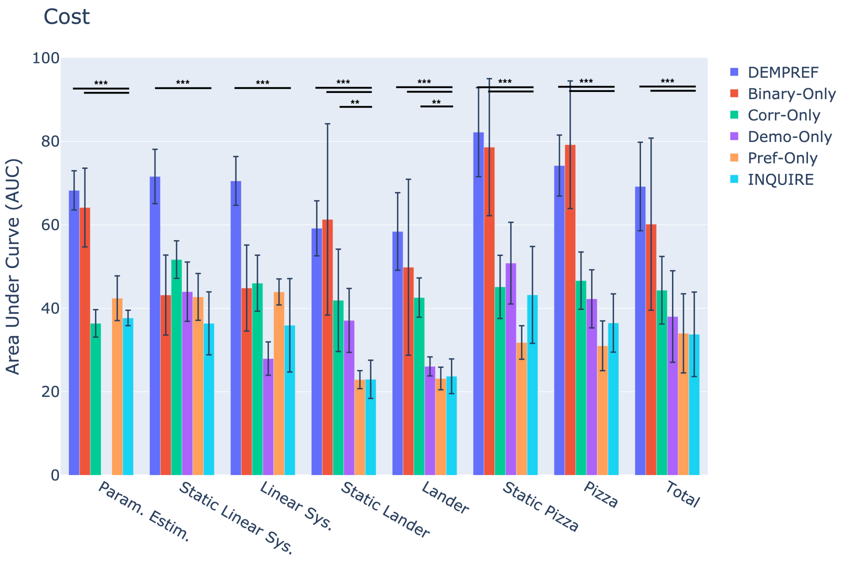


Figure 6.9: Visualizing Fig 6.8, with statistical significance noted. (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$)

6.4 Discussion

INQUIRE's performance in our evaluations underscores the value of learning agents that dynamically reason over not just the optimal query, but also the optimal interaction type to use given their current state. In every domain, whether in the static or changing state formulation, INQUIRE leveraged multiple types of feedback. This often took the form of demonstrations being followed by preferences – although binary feedback was fairly common when costs were involved. However the nature of how and when query types change differs between domains, as we would expect. We can see this easily in the unweighted evaluations. For example, in the Parameter Estimation domain, INQUIRE made between 14 and 18 correction queries before finding preference queries to be more informative. On the other hand, in the Linear Dynamical System domain, it asked only one demonstration query before switching to preferences. We see that across domains, INQUIRE outperforms fixed-strategy approaches such as DemPref [103] due to its dual optimization of both query and interaction type given the learning agent's current model of the task reward.

The largest benefits of using an algorithm like INQUIRE appear to be in the static state setting where it consistently outperforms other single-interaction and fixed-ratio approaches (Figure 6.3). This further highlights the benefit of optimizing over both query and interaction type. Repeatedly requesting a demonstration from a single state may be unlikely to yield significantly different information between iterations, for example. Using a variety of interactions allows for the learner to avoid these likely redundancies and likely contributes to INQUIRE's improved performance over other baselines. In changing-state settings, INQUIRE is sometimes outperformed by single-interaction approaches which benefit from the stochasticity inherent in the changing starting state.

Finally, we introduce in INQUIRE a cost metric that we have shown to minimize the cost-to-distance ratio across domains. Ultimately, queries that are informative can only be so if a human teacher is able to answer them without much difficulty. This cost metric can be set to better achieve an informativeness-to-cost ratio. For example, cost could be the average time taken to respond to particular queries or their associated cognitive load [77].

There a few limitations in this work. First, we did not conduct any human studies and instead opted to use an oracle agent for our evaluations. Feedback from people is often suboptimal, and that suboptimality may differ across interaction types. Future work should further investigate this relationship between suboptimality and interaction types. The β parameter in our current formulation can be used to quantify this. Furthermore, INQUIRE operates only within the framework of sequential decision making tasks. We can envision expanding this approach to non-sequential tasks such as labeling, where crowdwork and human feedback are commonly used for training models.

INQUIRE is an active learning approach that enables a learning agent to dynamically reason over both what query to ask and which interaction type to use, given its current model of the task reward. This allows it to adjust its learning based on task proficiency. In order to achieve this, we use a unifying information gain formulation across interaction types. Our evaluations demonstrate the efficacy of this approach, and show that it outperforms baselines across a variety of tasks domains and state configurations.

Chapter 7

Interaction Considerations in Learning from Humans

Given the interaction archetypes outlined in Chapter 5, we revisit the HIL relationship graph (Figure 4.1) presented in Chapter 4. In particular, we consider the relationship between interaction types and teaching performance ("User Experience") and training data ("Teaching Quality").

Approaches to collecting data from people include asking for annotations on video and images [109], ratings of behavior [40], task demonstrations [2], critiques or corrections of proposed trajectories [15, 37], and preferences between options [115]. Distinctions between these techniques have led to a growing body of work on understanding them relative to each other. Different interaction types can be used in combination to accelerate learning [103], better leverage people as teachers [25], and differ in the amount of implicit information they encode [66]. Learning interactions are often selected based on how informative they are for a learner, without examining how human teachers may differently perceive and respond to those interactions. However, people's experiences with different interactions can affect the data they provide.

Teaching performance is important to consider because, despite the assumption that people are expert teachers, they are also flawed teachers. In order to collect high-quality data, whether via active learning or curated training sets, i.e. passive learning, it is necessary to leverage data collection processes that accommodate people's limitations. People provide noisy data, are biased towards providing positive rewards, and get fatigued [8]. Several of the shortcomings in people's teaching capabilities may relate to the fact that as the cognitive load (i.e. the portion of working memory being utilized) on an individual increases, they grow more easily distracted and have worse task performance [128]. Interaction design can be used to modulate cognitive load in human learners [31]; that is, the way a task is presented can affect how burdensome it is and may ultimately affect the quality of the data it produces.

In this chapter, we revisit the interaction archetypes presented in Chapter 5 (*Showing, Categorizing, Sorting,* and *Evaluating*). We present the design and results of a user study constructed to identify differences between interaction types in terms of human factors related to data quality, such as cognitive load, confidence, and subjective usability. We find that *Evaluating* interaction types, where people identify good behavior, are the most cognitively loading and least usable in both of the study's task domains. *Categorizing* (i.e., assigning a positive or negative reward) and *Showing* (i.e., giving demonstrations) are less cognitively loading and more usable. While

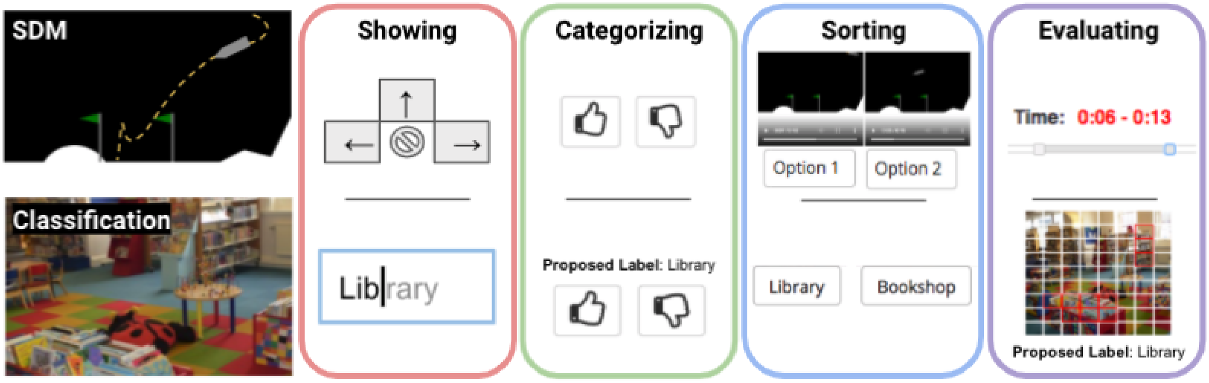


Figure 7.1: We identify and evaluate four interaction archetypes for sequential decision making (SDM, top) and Classification (bottom) tasks.

quantifying human cognitive-effort such that it can be used in an objective function is a complex problem that remains outside the scope of this thesis, our findings provide initial insights that we were able to use when evaluating INQUIRE with cost-awareness (Chapter 6) and underscore the reality that interaction types do affect teaching performance and thereby affect the training data that is ultimately collected (Chapter 4).

7.1 Effects of Interactions on Teaching Performance

We designed a mixed-design user study to find empirical differences in cognitive load, performance and usability between interaction types. Our within-subjects independent variable, interaction type, had four levels: *Showing, Categorizing, Sorting*, and *Evaluating*. Our between-subjects independent variable, task domain, had two levels: Sequential Decision Making (henceforth SDM), and Classification.

To enable comparisons between interaction types, we selected similarly complex examples from each cluster and minimized presentation differences. We made the assumption that salient differences in user attitudes manifest even between low-complexity interactions (e.g. differences would be present between reward-punishment and 2-way preference comparisons, not just rating scales and N-way rankings). We chose the lowest-complexity, non-trivial examples of each interaction type: demonstrations with a manageable $|A_L| = 4$ for *Showing*, reward & punishment for *Categorizing*, preference comparisons for *Sorting*, and credit assignment (a subcategory of critiques) for *Evaluating* (Figure 7.1 [1]). We also standardized the interaction interface (e.g. the number of buttons, duration of tasks, available controls) as much as possible to minimize their impact on user attitudes.

The *SDM* task involved piloting a lunar lander to land upright between flag posts. Participants supplied or responded to a trajectory. We manually created trajectories to show to participants, and ensured an equal distribution of successful and failed trajectories. For *Showing*, participants used keyboard inputs ($|A_L| = 4$) to provide example trajectories. For *Categorizing*, participants labeled a video of a potential lunar lander trajectory with a thumbs-up or thumbs-down. For *Sorting*, participants were shown two videos of potential trajectories for a lunar lander, and chose the better one. Finally for *Evaluating*, participants were given one video of a potential lunar lander trajectory, and used a double-ended slider to select the best portion of the trajectory.

The *Classification* task consisted of a series of images to be annotated. Users provided or responded to one-word captions. We used 20 images from Pascal VOC 2012 [45] by randomly selecting one image from each of its classes. Captions to be evaluated were generated by a Keras InceptionV3 [129] model trained on ImageNet [42]. For *Showing*, participants were given an image and typed a caption of their own choosing into a textbox. For *Categorizing*, participants labeled an image-caption pair as thumbs-up or thumbs-down. For *Sorting* participants were shown two potential captions for a given image, and chose the one they felt was better. Finally, for *Evaluating*, participants were given one image-caption pair, and used a grid to select the parts of the image that best justified the proposed caption.

7.1.1 Hypotheses

We hypothesize that interaction types are not interchangeable with respect to their human factors:

H1 Cognitive load differs between interaction types

H2 Task completion times differ between interaction types

H3 User confidence varies between interaction types

H4 Subjective usability differs between interaction types

H5 Preferred interaction types differ between tasks

Our study is designed to identify significant differences, not to find causal relationships, but we expect that as Response Choice Space and Response Size increase, cognitive load will increase, while usability and performance suffer.

7.1.2 Measures

We collected metrics on cognitive load (M1, M2), performance (M3, M4), and usability (M5-M9), as well as participants' responses, and any button toggles or video replays. Participants had the opportunity to provide additional feedback, and were asked to report their age and gender.

M1 - Secondary task performance. During each interaction, participants pressed a key every time a color-changing circle turned pink (Figure 7.2). The longer the participants' reaction time, the greater the cognitive load [41].

M2 - Paas subjective rating scale. After each interaction section, participants responded to the prompt "*How much mental effort did this interaction type demand?*" using a 9-point Likert scale [102] .

M3 - Primary question response time. We recorded the time between when the participant was given the stimulus (e.g. began the lunar lander game, started to view presented trajectories, or was first presented with an image to label) and when they submitted their response.

M4 - Self-reported confidence per query. For each query, participants responded to the prompt "*How confident are you in your answer to the primary question (not the color-changing circle) above?*" with a 4-point Likert scale. We did not include a neutral option.

M5 - Frustration. After each interaction section, participants responded to the NASA TLX [60] prompt *"How insecure, discouraged, irritated, stressed, and annoyed were you?"* on a 9-point Likert scale.

M6 - Complexity. After each interaction section, participants responded to the System Usability Scale (SUS) [22] prompt: *"I found this interaction type unnecessarily complex"* with a 5-point Likert scale.

M7 - Ease of Use. After each interaction section, participants responded to the SUS prompt: *"I thought this interaction type was easy to use"* with a 5-point Likert scale.

M8 - Overall Confidence. After each interaction section, participants responded to the SUS prompt: *"I felt very confident using this interaction type"* with a 5-point Likert scale.

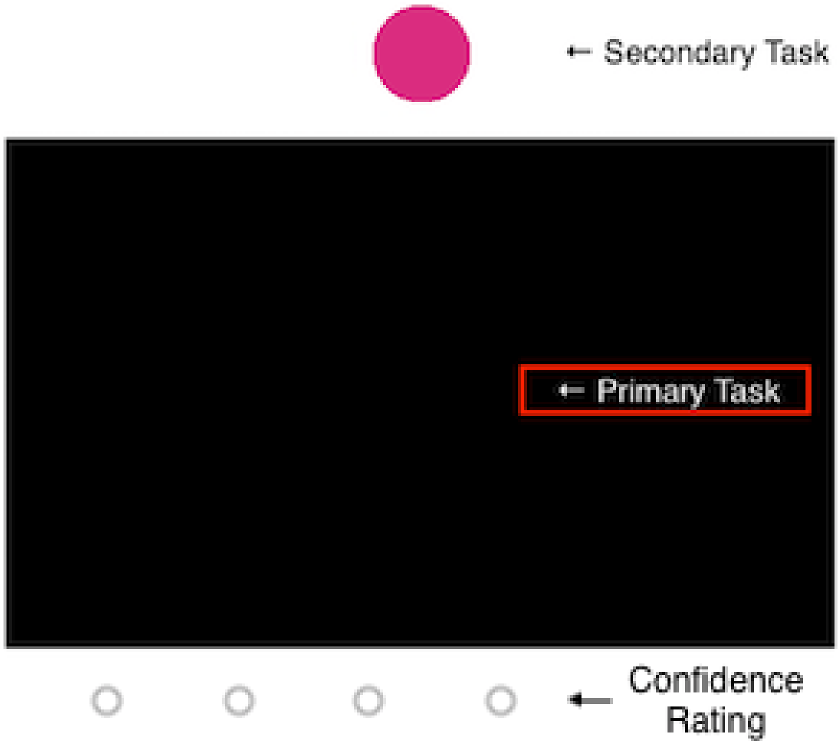


Figure 7.2: Participants responded to primary tasks described in Section 7.1.2, a secondary task (M1), and a confidence assessment (M4).

M9 - Forced Ranking. At the study's conclusion, users answered *"My nth choice interaction type would be . . . "* for their first, second, third, and fourth choice interactions.

7.1.3 Procedure

Participants were fully counterbalanced between all orderings of interaction types within a task domain. Participants were given instructions describing the study. They then practiced responding to the secondary task. At the beginning of each interaction type's section, participants were presented with instructions describing the interaction, and an example of a good response. Participants then practiced the interaction, including the secondary task and confidence assessment. Each interaction type's section comprised five questions presented in a sequential, but randomized, order.

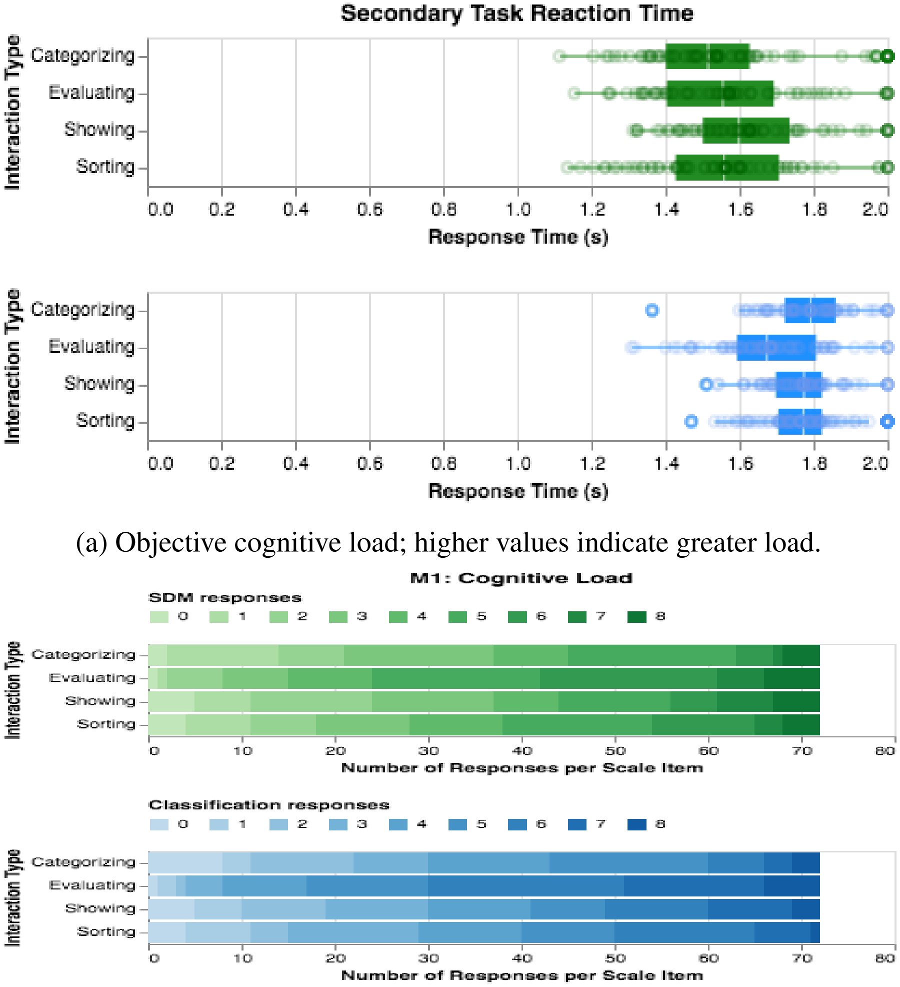
7.2 Results

We collected data from 150 Prolific workers over the age of 18 and with approval ratings $\geq 98\%$. Partial or duplicate task completions were discarded, leaving us with 144 participants, 72 per task domain. 61.1% of the participants self-identified as male, 37.5% as female, and 1.39% as non-binary. Their ages ranged from 18 to 70 (*M* = 26.71, *SD* = 9.45). This study and recruitment procedure was approved by our Institutional Review Board. We analyzed the effects of interaction types in each domain separately, and opted not to evaluate interaction effects between domains for two reasons. First, we use Likert-type scale data which is subject to interpersonal variance and cannot be reliably compared between separate populations. Second, our goal is not to identify differences between these specific domains, but to show that task domain can affect participants' preferences.

We analyzed all ordinal data using a Friedman Test followed by a post-hoc Wilcoxon signed-rank test (Bonferonni correction $\alpha = 0.0083$). Numerical data was analyzed with a one-way repeated measures ANOVA and post-hoc pairwise Tukey analyses. We used $\alpha = 0.05$ for our analyses. Because we used only portions of NASA-TLX and SUS to avoid participant fatigue, we treat each question as an individual item.

H1: Cognitive load differs between interaction types. A one-way repeated measures ANOVA revealed a statistically significant difference in secondary task reaction times (M1, Figure 7.3a) between interaction types ($F(3, 213) = 6.57, p < 0.001$ in SDM, and $F(3, 213) = 20.04, p < 0.001$ in Classification). In SDM, secondary reaction time was significantly longer in *Showing* as compared to *Categorizing* ($p < 0.05$). In Classification, secondary task reaction time during *Evaluating* was significantly less than in *Showing*, *Sorting* or *Categorizing* ($p < 0.01$). For completeness, we repeated this analysis after performing outlier rejection for samples more than three standard deviations from the mean: no differences were found in our results.

Differences were also found in participants' subjective assessments of the mental effort each interaction type required (M2, Figure 7.3b) in both domains ($\chi^2(3) = 1.3 \times 10^{-13}, p < 0.001$ in SDM, and $\chi^2(3) = 4.44 \times 10^{-15}, p < 0.001$ in Classification). In SDM, participants felt that



(a) Objective cognitive load; higher values indicate greater load.

(b) Subjective cognitive load; darker values indicate greater load.

Figure 7.3: Cognitive load (H1) was measured both objectively (secondary task reaction time) and subjectively.

Sorting was significantly harder than *Categorizing* ($p < 0.006$), and that *Evaluating* was significantly harder than *Showing*, *Sorting*, and *Categorizing* ($p < 0.001$ in all cases). In Classification, participants rated *Evaluating* as significantly harder than *Showing*, *Sorting*, and *Categorizing* ($p < 0.001$ in all cases). *The data supports H1*.

H2: Task completion times differ between interaction types. Statistically significant differences were found in response times (M3, Figure 7.4) between interaction types in both domains ($F(3, 213) = 28.79, p < 0.001$ in SDM, $F(3, 213) = 166.29, p < 0.001$ in Classification). In SDM, response times were significantly greater in *Sorting* as compared to *Showing* and *Categorizing*, and in *Evaluating* as compared to any other interaction type ($p < 0.01$ for both). In Classification, participants' response times to *Evaluating* were significantly greater than to any other interaction type ($p < 0.01$ in all cases). Figure 7.4 demonstrates these findings visually. When we repeated this analysis after performing outlier rejection for samples more than three standard deviations from the mean, our results largely stayed the same: we additionally found that *Showing* interactions in the Classification domain took significantly longer than *Categorizing* $(\alpha < 0.05)$. *The data supports H2.*

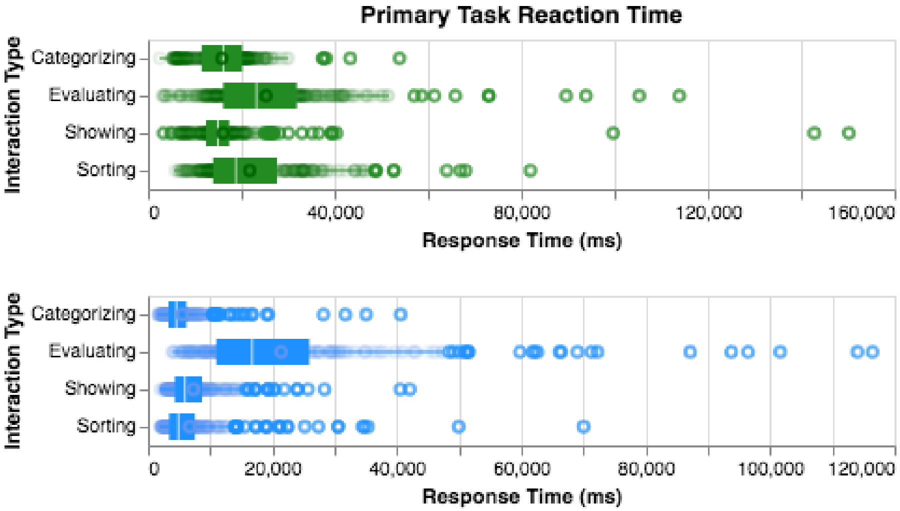


Figure 7.4: Time taken to complete the primary interaction task (H2).

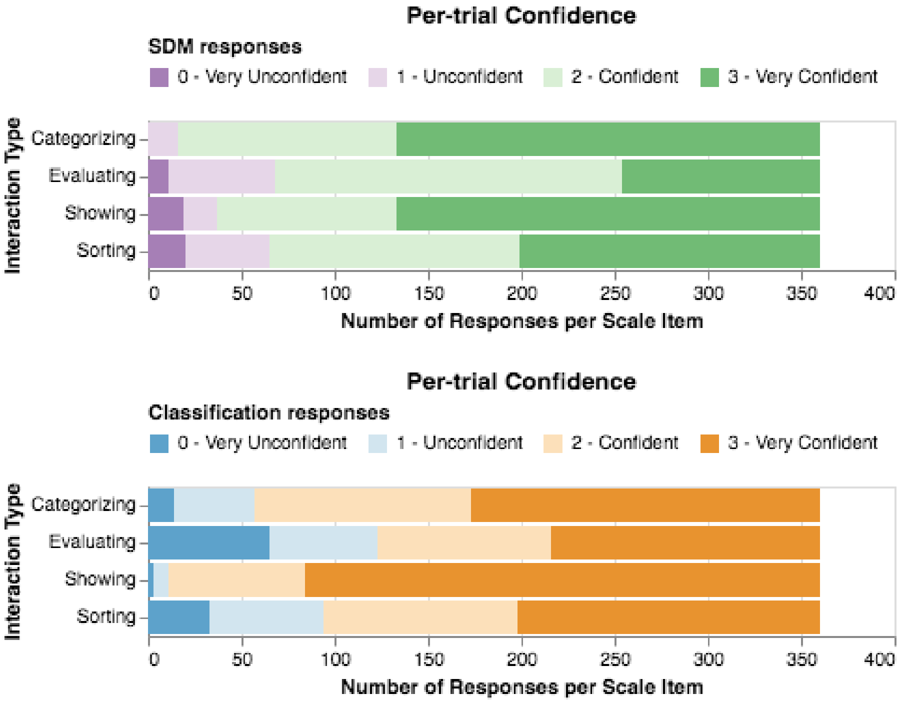


Figure 7.5: Per-trial confidence in response quality (H3).

H3: User confidence varies between interaction types. Median per-trial confidence scores (M4, Figure 7.5) were significantly different between interaction types in both domains ($\chi^2(3) = 49.24, p < 0.001$ in SDM, and $\chi^2(3) = 102.91, p < 0.001$ in Classification). In SDM, participants were significantly more confident in their responses to *Showing* and *Categorizing* than *Sorting* or *Evaluating* ($p < 0.01$ in all cases). In Classification, participants were more confident in their responses to *Showing* than to any other interaction ($p < 0.01$ in all cases). They were also more confident in their responses to *Categorizing* than to *Sorting* and *Evaluating* ($p < 0.01$ in both cases). *The data supports H3.*

H4: Subjective usability differs between interaction types. Significant differences were found in participants' ratings of frustration ($\chi^2(3) = 27.07, p < 0.001$ in SDM, $\chi^2(3) = 50.94, p < 0.001$ in Classification), perceptions of complexity ($\chi^2(3) = 41.68, p < 0.001$ in SDM, and $\chi^2(3) = 69.30, p < 0.001$ in Classification), ease of use ($\chi^2(3) = 33.14, p < 0.001$ in SDM, and $\chi^2(3) = 54.19, p < 0.001$ in Classification), and confidence with the interaction type ($\chi^2(3) = 28.66, p < 0.001$ in SDM, and $\chi^2(3) = 55.63, p < 0.001$ in Classification). These results, corresponding to M5 through M8, are shown in Figures 7.6 through 7.9.

Participants felt more frustrated by *Evaluating* than any other interaction in both task domains ($p < 0.00145$ in all cases). They perceived *Evaluating* as more unnecessarily complex than any other interaction in both task domains as well ($p < 0.001$ in all cases); in Classification, they also perceived *Sorting* as unnecessarily more complex than *Categorizing* ($p < 0.00785$). Correspondingly, participants found *Evaluating* to be less easy to use than any other interaction type in both task domains ($p < 0.001$ in all cases). In SDM, participants were more confident with *Showing*, *Sorting*, and *Categorizing* over *Evaluating* ($p < 0.001$ in all cases). In Classification, participants felt more confident using *Showing* than *Sorting* ($p < 0.00230$) or *Evaluating* ($p < 0.001$). They also felt more confident using either *Sorting* or *Categorizing* over *Evaluating* ($p < 0.001$ in both cases). *The data supports H4.*

H5: Preferred interaction types differ between tasks. We tallied participants' preferred interaction types (M9) using a Condorcet method. We found that in SDM, participants preferred *Showing*, *Sorting*, *Categorizing*, and then *Evaluating*. In Classification, they preferred *Categorizing*, *Sorting*, *Showing*, and then *Evaluating*. *The data supports H5.*

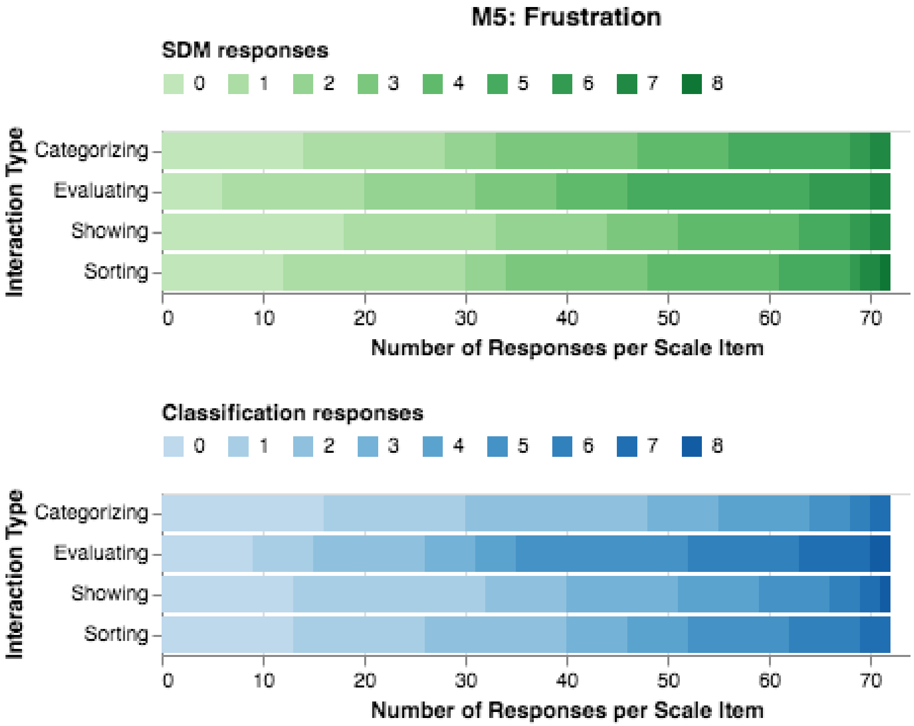


Figure 7.6: Responses to subjective measures of frustration (H4). Darker colors denote greater frustration.

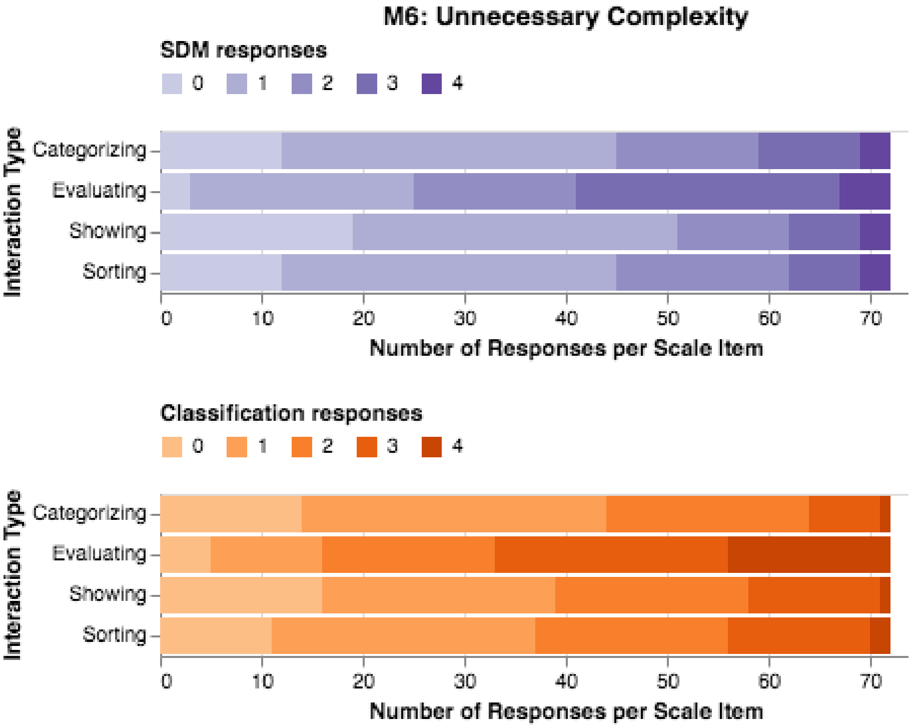


Figure 7.7: Responses to subjective measures of complexity (H4). Darker colors denote higher perceived complexity.

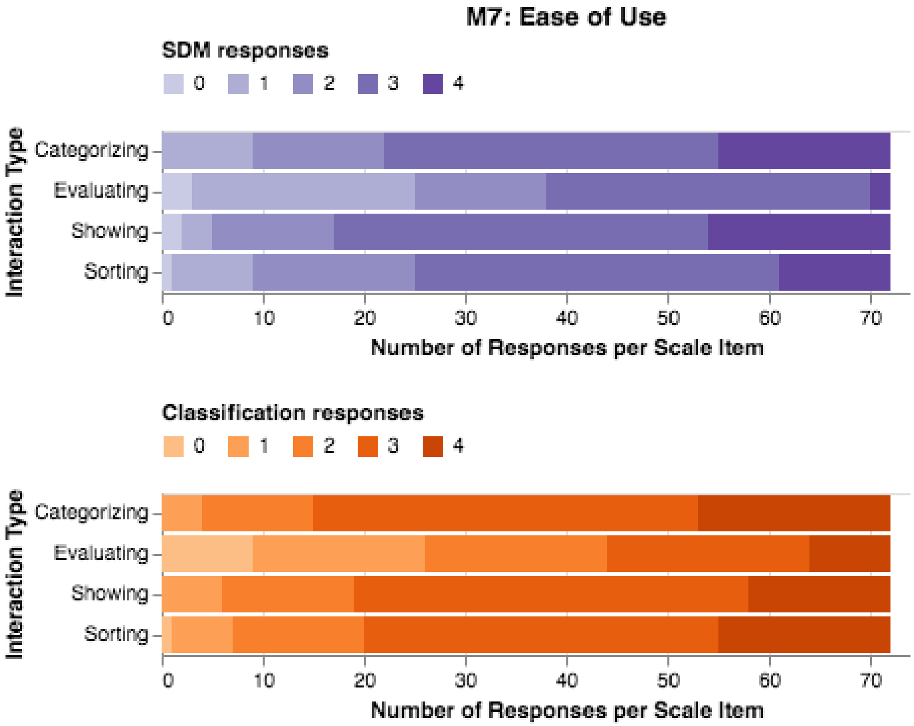


Figure 7.8: Responses to subjective measures of ease of use (H4). Darker colors denote greater ease of use.

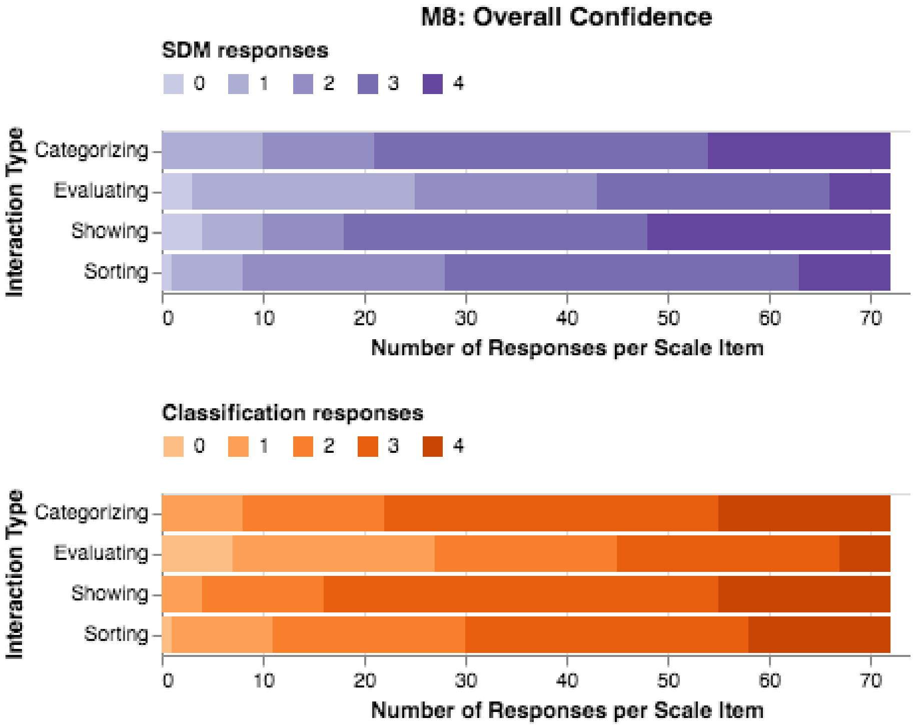


Figure 7.9: Responses to subjective measures of confidence (H4). Darker colors denote greater perceived confidence.

7.3 Discussion

Overall, our results show that interaction types are differently cognitively loading and usable, and may variably impact performance as estimated via task completion times and self-assessed confidence. Participants rated *Evaluating* interactions as requiring the most cognitive effort, being the most frustrating, the most unnecessarily complex, least easy to use, and inspiring the least confidence. Objectively, they also took the longest time to complete *Evaluating* tasks. In SDM, *Sorting* took longer than *Showing* and *Categorizing*. In Classification, participants felt *Sorting* was more unnecessarily complex than *Categorizing* and were less confident using it than *Showing*. This suggests that *Categorizing* is preferable to *Sorting*, which is preferable to *Evaluating*. This corresponds to our expectation that as an interaction's Response Choice Space and Response Size increases, its usability decreases.

Unexpectedly, *Showing*, has the largest Response Choice Space and was among the easiest to use. This may be due to cognitive shortcuts: when guiding the lunar lander, users may not be processing all possible trajectories, nor are they thinking of every word they know in order to caption images. Thus, our big-$\mathcal{O}$ estimates may have been too coarse to capture the nuances of a user's *perceived* response space.

We also found a disagreement between participants' subjective assessment of mental effort and their objective secondary task performance, as in prior work [41]. This could indicate that there are additional, unknown factors that affect perceived mental effort. We did also observe a relationship between participants' primary task reaction times and subjective assessments of cognitive load, indicating that they took longer on cognitively loading tasks. This may have given them more opportunities to respond to the secondary task, influencing their reaction times.

Pre-existing notions that interaction types such as *Evaluating* and *Sorting* might be more user-friendly than others (particularly *Showing*), because they require fewer inputs from a user, were not supported in the two domains we evaluated. Furthermore, differences existed in participants' preferred interactions between the task domains, despite our standardization of interaction types within and between them. Future work is required to understand how properties of a task domain influence interactions.

This work is one step towards developing a principled understanding of the algorithmic and human-factors components of learning interactions. In particular, it is a necessary step towards understanding the trade-off between the expected informativeness of a learning interaction, and a user's ability to provide high quality feedback. As data-gathering needs increase in scale and across domains, understanding this relationship will expand our ability to design learning interactions that not only accommodate the needs of learning agents, but also leverage the capabilities of human teachers.

[1] The Classification image in Figure 7.1 is ”Inside Whitehaven library” by librariesteam and is licensed with CC BY 2.0 [36].

Chapter 8

Model Transparency to Guide Human Teaching

Thus far, we have investigated how an algorithmic learner can reason over interactions and the implications that the choice of interaction may have on ultimate learning outcomes via both the direct learning route and the indirect effect on the human teacher. However, the feedback that a human teacher gives is additionally influenced by the teacher's understanding of the learner's model.

This effect has been previously studied in the context of interactive learning; transparency into the learner's process led to human teachers providing less redundant, and more relevant, feedback [130]. This may feel familiar due to similarities to human-to-human teaching: if a teacher is trying to impart knowledge to a student and that student doesn't properly convey their mastery of the material, the teacher might either spend more time covering foundations under the mistaken belief that the student is lost or may move ahead to advanced topics without realizing the student needs more time to learn the basics.

Ultimately, teaching is a *joint task* wherein two entities must collaborate to ensure a successful and efficient transfer of knowledge. While collaboration can take many forms, we focus on mutual information sharing with the belief that doing so will help teachers select an optimal information sharing strategy and thereby improve the learning process and its outcomes. In interactive learning contexts, we propose that *the common interaction types via which human teachers are asked to share information with learning agents are merely avenues for information transfer and can therefore be equivalently used to share information from learning agent back to the human teacher as well.* The contributions of this chapter include: an algorithmic approach to generating feedback inspired by common approaches to learning from human feedback, a task setting in which to explore these algorithms, and the results of both an online user study and an in-person talk-aloud protocol to assess these algorithms.

8.1 Algorithms for Feedback

While there are many approaches to model transparency and XAI, they are not all equally well-suited to the type of collaborative, interactive learning we are interested in. Techniques such as

LIME [111] or saliency maps [3] may be useful in interpreting complex deep learning models, but too dense for repeated interactions between an algorithmic learner and human teacher, as would be necessary in the collaborative, interactive scenarios in which we are interested. Other approaches towards transparency in active learning have attempted to provide insight into how feedback will be interpreted [16], but finding lightweight paradigms by which an algorithmic learner can convey information back to the human teacher remains a relatively open question. Furthermore, there exists a wealth of research into implicit information transfer, particularly in human-robot interaction [20, 66]. In our research, we are interested in exploring novel, disembodied interaction paradigms for explicitly communicating a learner's model in an interactive learning setting.

We begin by defining such an interactive learning setting between a human teacher H and an algorithmic learner L. The learner L is attempting to discern a rule $r \in R$. Here, R is a finite set of possible rules and each $r \in R$ can be described as a vector consisting of d discrete features. We can envision this mapping to scenarios such as a recommender system mapping a new user to a particular set of known preferences that best describe that user's tastes. Let us further formalize this interactive learning environment such that at each time step t, the teacher H takes an action a_t and the learner L responds to a_t with some feedback f_t. We will further detail an instantiation of this interactive learning setting in Section 8.2 when we describe our evaluation setup.

8.1.1 Enforcing Relevance to Teaching

In order for a teacher to perceive feedback as useful – or, at the very least, not confusing – it must feel germane to the context in which it is given. In order to ensure this, we make a few assumptions. First, we assume that when taking any action a_t, a teacher is likely trying to focus on teaching the learner L the value of a single feature j of the true rule r^*. We define this as r^*_j . For example, in the task we will describe shortly, j might be associated with the discrete feature "primary class" and can take on values, denoted by $\mathcal{V}(j)$, such as "number", "shape, "color", or "fill" (see Section 8.2 and Figure 8.2 for more information). Then, we further assume that the learner's feedback f_t will feel most relevant if it can correctly identify the particular feature of interest j and share information about its belief over R given that feature. Note that both j and r^* are unknown to the learner L.

In order to estimate j, we maintain a separate relative frequency histogram of each possible value for each j (note that we do not use a probability density function because we are operating with a discrete feature space). We construct this by considering the set of valid rules remaining in the hypothesis set, and how many of those rules have $j = v$ for all possible values v. Henceforth, let us define these distributions to be ω_j. At each iteration t, we compute the KL-divergence between ω_j at times t and $t - 1$ for all j. We then assume that the w_j with the largest KL-divergence between time steps is the most relevant feature to the action a_t that was just taken. These assumptions are encoded in Equation 8.1 below.

$$
\begin{aligned}
j_t^* &= \arg\max_j D_{\text{KL}}(\omega_{j,t-1}||\omega_{j,t}) \\
&= \arg\max_j \sum_{v \in \mathcal{V}(j)} \omega_{j,t-1}(v) \log(\frac{\omega_{j,t-1}(v)}{\omega_{j,t}(v)})
\end{aligned}
\tag{8.1}
$$

8.1.2 Feedback Implementations

We implement four different types of feedback based on the interaction types discussed in Chapter 5. Each of these implementations is informed by both (1) the explicit presentation of the information as discussed in that chapter, and (2) the implicit information encoded (e.g. the implications of the choice made by the teacher given all possible response choices, as also discussed in the same chapter as well as Chapter 6).

We opt to implement an agent that provides feedback regarding the values of a particular feature of interest as discussed and computed in the previous section. This decision was made because give feedback on each rule individually seems neither realistic to how people share and receive information nor scalable to larger or continuous problem spaces. Furthermore, to do so, we would need to compute the probability of some $r \in R$ being the true rule r^*, given all the actions taken by the teacher so far and the most likely current feature of interest j as defined in the previous section, i.e. $p(r = r^*|j; a_0, a_1, ...a_t)$. Instead, we can evaluate the likelihood of observing $a_0, a_1, ...a_t$ under each possible r for that given j using the relative frequency histogram that we discussed in Section 8.1.1. In other words, because we have a discrete rule space, this relative frequency histogram gives us the number of rules remaining (i.e. rules that could explain the actions taken so far) separated by what value they take on for the feature j.

To ground the feedback types we will be discussing in the remainder of this section, we briefly allude to our task formulation (discussed in detail in Section 8.2). We will use the running example of a task where the action a_t a teacher takes is to play a card with certain properties (e.g. “Number”, “Fill”, “Color”, and “Shape”) into one of three bins and the learning agent must identify the underlying sorting rule. Thus, $j \in \{$Number, Fill, Color, and Shape$\}$ and $\mathcal{V}(j)$ takes on respective values as shown in Figure 8.1.

Preferences. Traditionally, in a *preference* query in the active learning context, a teacher selects one option from a set (often a pair) of candidates proposed by the learner. We invert this paradigm by having the learner present two candidate hypotheses over some feature j and indicate which of those two it believes is more plausible. In order to standardize interaction formats with prior work and between one another, we do not allow the learner to say both candidates are equally plausible. Thus, given a feature j, the format of a *preference* type feedback is given by:

$$
f_{\textbf{pref}}(t) = \{\phi^a_{j,t}, \phi^b_{j,t}\} \tag{8.2}
$$

Here, j is given by Equation 8.2, and t denotes the current time step of the interactive learning process. We convert this to a human-legible sentence of the form “Between $\phi^a_{j,t}$and $\phi^b_{j,t}$, I think

$\phi^a_{j,t}$ is the feature more likely associated with j". For example, the agent might share, "Between "Number" cards and "Fill" cards, I think "Number" is more likely the feature associated with how cards are sorted."

However, we observe that when people share preferences in this format, there is an implicit understanding that the secondary option is a viable contender. In accordance with Gricean maxims [54] it would be unusual to share a preference between something you strongly prefer, and something that is irrelevant. Therefore, we let:

$$\phi^a_{j,t} = \arg\max_{v \in \mathcal{V}(j)} (w_j(v) | a_0, ...a_t) \tag{8.3}$$

$$\phi^b_{j,t} = \arg\max_{v \in \mathcal{V}(j) \setminus a} (w_j | a_0, ...a_t) \tag{8.4}$$

In other words, our preference tuple is comprised of the most and second most probable values v of the feature of interest j.

Binary Feedback. This is possibly the most commonly encountered form of teaching interaction in the traditional setting: users may give thumbs up or thumbs down on videos, images, or trajectories as in our INQUIRE setting (Section 6.1). In the inverted setting where the learner is providing feedback, the learner instead provides a positive or negative assessment of the likelihood of a feature taking on a particular value. In other words, this feedback is comprised of:

$$f_{\textbf{bnry}}(t) = \{\phi^a_{j,t}, r \in \{+, -\}\} \tag{8.5}$$

Implicit in the presentation of any positive or negative label is a certain degree of confidence in the label. Our learning agent has $\alpha^+ = 0.95$ and $\alpha^- = 0.05$ such that it will only assign a value a positive label if the likelihood of that value for feature j exceeds α^+ and, similarly, will only assign a negative label if the likelihood is below α^-. In pilot tests, people preferred positively oriented feedback. As a result, if at time t there exists a value that exceeds α^+ and a value that is below α^-, we make the decision to share the positive label. In the event that no values have crossed either α^+ or α^-, we default to sharing the most-likely value. This can be written as follows:

$$v^- = \arg\max_{v \in \mathcal{V}(j)} (w_j(v) | a_0, ...a_t) \tag{8.6}$$

$$v^+ = \arg\min_{v \in \mathcal{V}(j)} (w_j(v) | a_0, ...a_t) \tag{8.7}$$

$$f_{\textbf{bnry}}(t) = \begin{cases} \{\phi^-_{j,t}, -\}, & \text{if } p(v^-) < \alpha^- \text{and } p(v^+) \leq \alpha^+ \\ \{\phi^+_{j,t}, +\}, & \text{otherwise} \end{cases} \tag{8.8}$$

This can then be translated simply into a human-legible in the following way: "I think that v^- is not j," or "I think that v^+ is j." For example, the agent in our task might share: "I think that "Color" is not how cards are sorted."

Credit Assignment. A *credit assignment* query traditionally asks a human teacher to identify which feature(s) of an input best justify the associated output (often a label). Correspondingly, we define *credit assignment* feedback as identifying which feature of an action the learning agent believes to be most associated with the observed outcome state of that action. The format of this feedback type is therefore defined by:

$$f_{\textbf{credit}}(t) = \{\phi^a_{j,t}\} \tag{8.9}$$

Given the nature of the feedback, we let $\phi^a_{j,t}$ take on the most likely value associated with j as follows:

$$\phi^a_{j,t} = \arg\max_{v \in \mathcal{V}(j)} (w_j(v)|a_0, ...a_t) \tag{8.10}$$

Where j and t are defined as above. The sentence we construct from this is of the form: “I think a_t because it is $\phi^a_{j,t}$”. Our running example might be phrased as, “I think the “Red, Squiggle, One, Ellipse” card was placed into Bin 1 because it is a “One” card.” Unlike other feedback types, the nature of the *credit assignment* feedback (a member of the *Evaluation* archetype of interactions) necessitates that it directly relates to the most recent action taken rather than expressing itself as a general claim.

Showing. This feedback type when used as a querying mechanism in active learning settings traditionally involves the learning agent giving the human teacher a starting state from which it wants an example of a trajectory due to some degree of uncertainty, and nothing more. Indeed, this is how we implemented it in INQUIRE (see Section 6.1). In order to invert this paradigm, we construct a *showing* feedback type that simply provides a human teacher with a likely, but not certain, feature of interest and a less-confident, more probing framing. Like *credit assignment*, the format of this feedback is given by:

$$f_{\textbf{show}}(t) = \{\phi^a_{j,t}\} \tag{8.11}$$

because the feedback type is comprised only of one particular value of our feature of interest. However, we define $\phi^a_{j,t}$ to be the second-most likely value associated with j, if such a value exists, as follows:

$$v_1 = \arg\max_{v \in \mathcal{V}(j)} (w_j(v)|a_0, ...a_t) \tag{8.12}$$

$$\phi^a_{j,t} = \arg\max_{v \in \mathcal{V}(j) \setminus v_1} (w_j(v)|a_0, ...a_t) \tag{8.13}$$

Note that this is functionally identical to finding the second-most likely value for use in the *preference* feedback type. We translate into a sentence of the general form: “I am wondering what action would be associated with $\phi^a_{j,t}$.” For example, “I am wondering where a ”One” card would go.” We initially piloted a more confident sounding statement (e.g. ”I am thinking about...”) but that was received poorly; we also trialed using the most and least likely values of j, but using the second-most likely value was generally considered the most effective version. We note that it is possible that this feedback type may have benefited from using a maximum information-gain approach to identify features of interest given its somewhat information-seeking nature, but for consistency and for the interest of relevance, we used the maximum KL-divergence approach instead.

8.2 Set-HiLL Task Formulation

We seek to jointly evaluate a teacher's ability to convey information to a learning agent, as well as the agent's ability to convey their understanding back to the teacher. To achieve these goals, we build on the setup presented in [71] to construct a task inspired by the card game Set which involves a specialized deck as depicted in Figure 8.1. In our Set-HiLL variant, ask human teachers to sort cards into one of three bins based on properties such as color, shape, number and fill. The particular sorting rule they are asked to use takes the place of a preference that a person might be attempting to teach in a recommendation or assistive technology setting and enables us to evaluate teaching and learning efficacy against a known, static ground truth. We encode complexity into the sorting rule by having it be based primary on one feature (e.g. number), and also including an exception value (e.g. cards where the *Shape* is *Diamond*) which overrules the primary sorting feature.

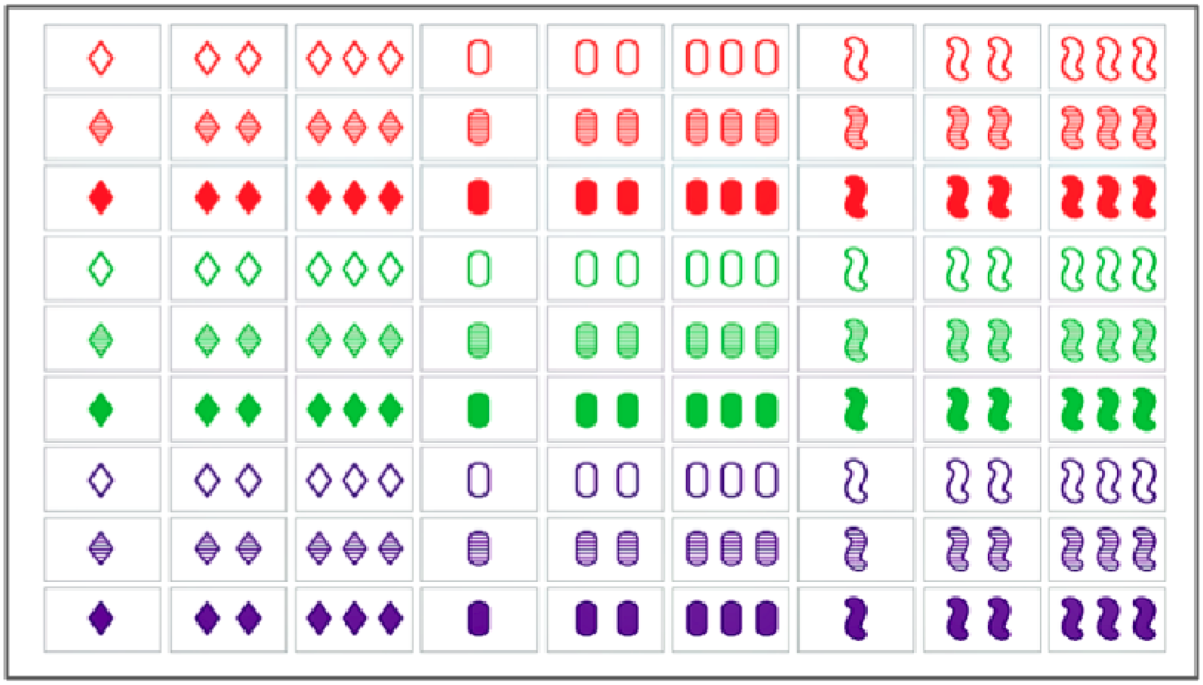


Figure 8.1: Set cards can be grouped based on four different classes of properties: color (red, green, or purple), shape (diamond, oval, or squiggle), pattern (open, striped, or solid), and number (one, two, or three).

This rule $r*$ is provided to the teacher at the outset of the task, and is sampled from a finite set of possible rules $\mathcal{R}$. We will further expand on rule construction shortly. For now, suffice it to say that a rule $r \in \mathcal{R}$ dictates which cards can be placed into which bins. For example, a rule might be: *"'Two' cards go in Bin 1, 'One' cards go in Bin 2, 'Three' cards go in Bin 3. 'Diamond' cards also go in Bin 3 regardless of number because they are an exception."*

Teaching and learning of this rule commences in a turn-based fashion. First, the teacher selects a card to play (i.e. place into the appropriate bin, based on $r*$). The algorithmic learner observes the card selection and placement, and then shares some feedback with the teacher. The teacher can then select another card to play, and so on. The game continues in this fashion until the teacher believes that the agent has learned $r*$ and the teaching process should end.

8.2.1 Rule Construction

We design our rules to be easily understandable while also being complex enough that teaching them would be a non-trivial task. We note that in real-world settings people often have deeply-held personal beliefs that they cannot easily articulate or share with the learning system, but that are easily recognizable to them (e.g. they know how to act in accordance with their underlying beliefs and preferences).

However, to make the problem tractable and minimize noise, we had to make some assumptions and place some constraints on the construction of these rules. The most salient of these is that while we recognize that in reality people's preferences and beliefs are non-stationary and evolving, we need to use a static rule as a known, ground truth in order to evaluate learning and teaching efficacy. We liken this to a hyper-local version of the real-world problem wherein we assume a person already has a refined understanding of their own taste and we are looking at their interactions with a learner in a short enough time-frame that we would not expect to see major deviations in that taste.

Finally, we also assume that people will try to make sense of the rule by decomposing it into its constituent components. These correspond to the features of rules that we discuss throughout Section 8.1. We corroborate this assumption via our talk-aloud study, discussed in more detail in Section 8.4. Using the rule, *"'Two' cards go in Bin 1, 'One' cards go in Bin 2, 'Three' cards go in Bin 3. 'Diamond' cards also go in Bin 3 regardless of number because they are an exception."* as a guiding example, let us consider the features that describe a rule:

1. **Primary Class.** This can take on any value in {*number, color, shape, fill*}. In our example, cards are primarily sorted into bins based on their *number*.
2. **Bin 1 - Primary Value.** Similar features exist for Bin 2 and Bin 3 as well. Each of these features takes on a value in relation to the Primary Class feature. For example, in our example, Bin 1 - Primary Value would be assigned the value *'Two'*. However, we can imagine that in another rule, where cards are primarily sorted by color, it might take on a value such as *'Red'*. The full list of class and value pairings can be found in Figure 8.2.
3. **Exception Class.** Similar to the Primary Class feature, Exception Class can take on any value in {*number, color, shape, fill*}. In our example, it takes on the value of *shape*.
4. **Exception Value.** This is similar to the three Bin n - Primary Value features, in that it is parameterized by the Exception Class. In our example, it takes on the value of *'Diamond'*.

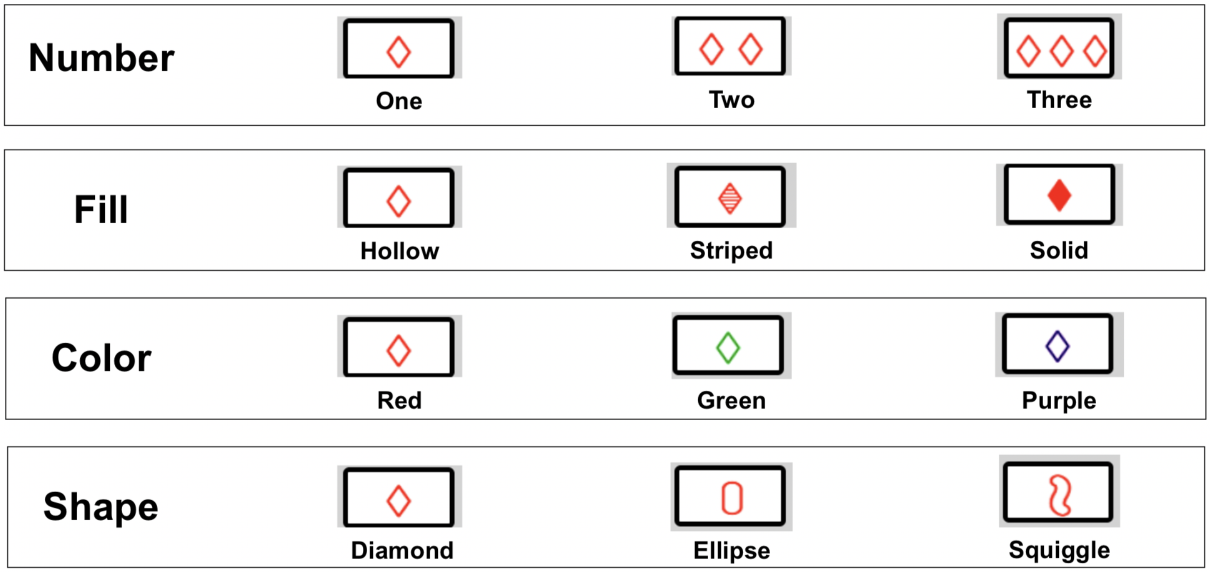


Figure 8.2: *Set* cards have four different classes of features: number, fill, color, and shape. Each of these has three possible values, as shown above.

5. **Exception Bin.** This feature specifies the bin in which the exception is housed, and therefore can be assigned a value in {*Bin 1, Bin 2, Bin 3*}. In our example, it is *Bin 3*.

8.2.2 Evaluation Criteria: Accuracy, Efficiency, and Influence

Our fundamental goal is to develop HiLL systems that both improve the quality of learning outcomes and minimize teaching burden. Therefore, we need to simultaneously evaluate the experiences of both the algorithmic learner and the human teacher. While there are many ways to evaluate the outcomes of different feedback approaches, we will focus on the following three dimensions: *accuracy, efficiency*, and *influence*.

We describe the motivations behind the selection of these categories as follows:

1. **Accuracy.** The accuracy of a learned model is central to its ability to fulfill its intended function. In our setting, we can calculate model accuracy as how often the algorithmic learner is able to identify the correct rule from a larger set of rules. However, we additionally define teaching accuracy as a representation of *how well a teacher understands the state of the learning agent*. Teaching accuracy can lead to: training efficiency (e.g. by more easily finding the most relevant examples), more value-aligned outcomes, and greater confidence in learner performance.
2. **Efficiency.** Training any sort of intelligent system can be costly with respect to both time and human effort. Minimizing such costs will lead to faster development cycles, as well as make personalized training more accessible to more people. Therefore, in developing HiLL systems that are designed with people in mind, it is critical to understand how quickly an algorithmic learner is able to converge on an intended goal, and how much teaching effort was required in the process.

	Quantitative	*Qualitative*
Accuracy	How long does it take a human teacher to perceive an algorithmic learner as proficient? How well do their expectations of the learner's proficiency align with its true capabilities?	How do users interpret what the model is getting wrong? Are they able to correctly identify mistaken beliefs?
Efficiency	How quickly is the algorithmic learner able to learn the rule being taught (e.g. number of interactions and/or duration of teaching session)?	How are different approaches perceived (e.g. with respect to usability, legibility, or informativeness) by human teachers?
Influence	What is people's self-reported confidence in their teaching given different types of feedback? How valuable do they self-perceive feedback to be for their teaching process?	How often do people find feedback to be helpful and useful to their teaching process? Why terminate learning when they do? What does the model know, or not know? How often does feedback induce a change in teaching behavior?

Table 8.1: We are interested in both objective and subjective measures of teaching and learning performance, along three primary dimensions.

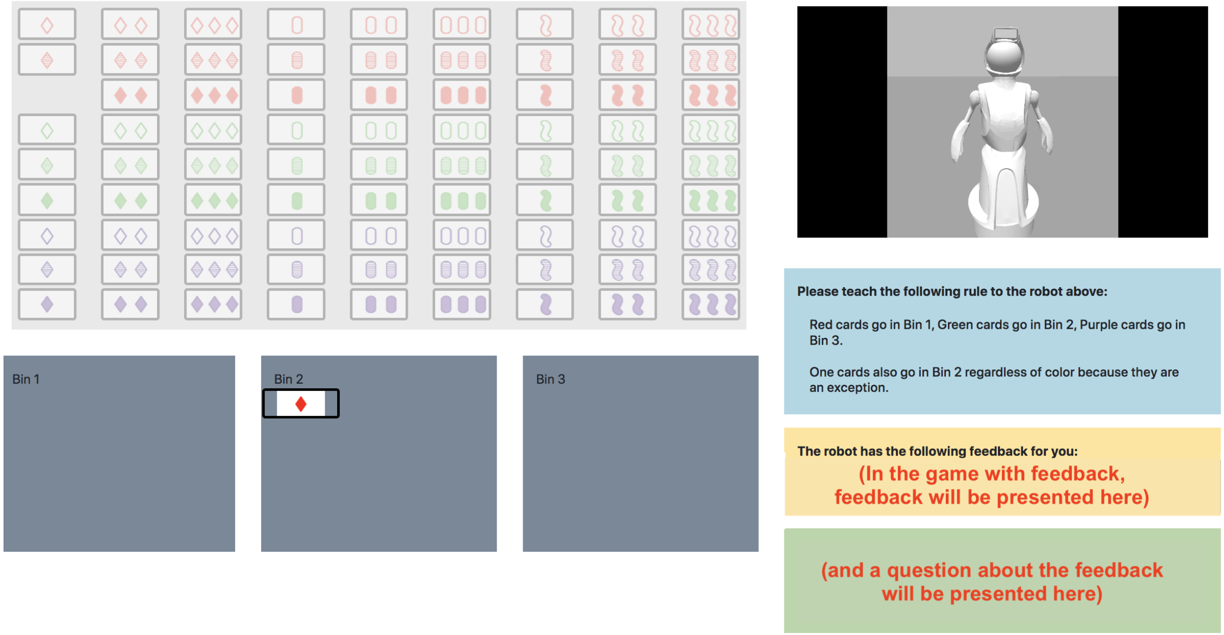
3. **Influence.** We hypothesize that the nature of collaborative learning makes it more efficient than interactive, non-collaborative learning. First, we will analyze how techniques that are commonly used in learning from humans might be able to be used to convey information from an algorithmic learner back to the human teacher. While it is important to understand the effects of these techniques on accuracy and efficiency, we are also interested in how they might differently influence teaching techniques and decisions, if at all. Understanding how different techniques impact human teaching approaches can give us further insight into how people attempt to share knowledge with algorithmic learners.

An overview of the metrics we use in our evaluations can be found in Table 8.1. In order to evaluate both the quantitative and qualitative metrics listed, we concurrently designed and ran both an online user study and an in-person talk-aloud study. These will be described in further detail in the subsequent sections.

8.3 Evaluation: Online Between-Subjects User Study

We designed a between-subjects user study to find empirical differences in quantitative metrics relevant to *accuracy, efficiency,* and *influence*. The between-subjects independent variable, the interaction type used by the learner when presenting feedback, had four levels: *showing*, *preference*, *binary feedback*, and *credit assignment*. In the control condition, the learning agent did not provide any feedback.

We have already discussed in Section 8.1.2, the ways in which we standardized the generation of feedback across interaction types. In order to further control for any confounding variables, we kept the study interface (see Figure 8.3) the same across all conditions. Participants were presented with a grid of all possible *Set* cards, grouped by color, shape, fill, and number. They were given a rule, as described in Section 8.2.1, and tasked with selecting cards from the grid to be placed into the appropriate bins. We minimized participant error by having them select a



Please teach the following rule to the robot above:

Red cards go in Bin 1, Green cards go in Bin 2, Purple cards go in Bin 3.

One cards also go in Bin 2 regardless of color because they are an exception.

The robot has the following feedback for you:

(In the game with feedback, feedback will be presented here)

(and a question about the feedback will be presented here)

Figure 8.3: The layout for both the online and in-person user studies featured all possible cards, three bins into which those cards could be placed, a box containing the rule to be taught, a box containing feedback from the agent, and a box with a Likert Scale to use in evaluating the feedback. We also included a GIF of a Quori robot in a neutral pose.

card to play, but then auto-sorting those cards according to the rule. In order to further minimize the number of accidental clicks, as participants hovered over the various cards in the grid, the corresponding bin for that card would be highlighted as a visual reminder of the rule. Participants were reminded that cards being auto-placed into bins was merely an assistive short-cut and should *not* be interpreted as the learner playing the card. In conditions with feedback, participants received feedback after every card played. They were then asked to evaluate the helpfulness of this feedback prior to either playing another card.

In all conditions, participants were instructed to terminate the learning process if they felt confident in the learner's understanding of the rule. In order to incentivize teachers to terminate precisely when the agent had converged, we introduced a bonus scheme. Participants received an extra $\$1.00$ if they terminated at the correct time, $\$0.75$ if they were too early or too late by one card, and $\$0.25$ if they were too early or too late by two cards. If a participant incorrectly terminated when the learning agent had not yet converged upon the true rule, they were asked to continue playing cards until they reached a successful termination state. The bonus was only given based on their first attempt to terminate.

8.3.1 Hypotheses and Measures

As this is an exploratory study into the symmetry – or lack thereof – in sharing information with and receiving information from a learning agent, our hypotheses are few and simple. We have two sets of hypotheses: first, we believe that the feedback we have generated will be effective, given that we know it is effective as a teaching instrument for algorithmic learners in active learning settings featuring human teachers.

H1 Teaching with feedback will be more accurate than teaching without feedback.

H2 Teaching with feedback will be more efficient than teaching without feedback.

Second, we assume symmetry from the active learning setting: we hypothesize that there will be a difference between receiving feedback from different interaction types, just as there is an experienced difference in sharing information.

H3 Different interaction types will differently affect teaching accuracy.

H4 Different interaction types will differently affect teaching efficiency.

H5 Human teachers will be differently influenced by each interaction type for feedback (e.g. with respect to their confidence in their own teaching, and the subjective experience of teaching).

In order to evaluate these hypotheses, we collected data on accuracy (M1 - M2), efficiency (M3 - M4), and influence (M5 - M7). Participants had the opportunity to provide additional free-response feedback after every trial, as well as demographic information at the end of the study. We did collect some additional metrics, such as the helpfulness of each individual piece of feedback, that we chose not to analyze for the online study due to high levels of variability between individuals; further discussion on that can be found in Section 8.4 where we discuss our concurrently designed talk-aloud study.

M1 **Accuracy of Perceived Convergence.** The number of cards that the teacher gave the agent before believing the agent to have converged and terminating, relative to when the agent actually converged. Participants were encouraged to terminate when they first felt the agent had learned the rule via a bonus reward scheme described in the section 8.3.

M2 **Failure Rate.** The number of instances where termination occurred prior to the learning agent converging upon the true rule.

M3 **Number of Cards to Converge.** The number of cards the learning agent needed to see before converging upon the true rule.

M4 **Time to Select Each Card** The average amount of time (ms) that a teacher took to select a card to play next. Longer amounts of time can indicate higher cognitive load [128].

M5 **Confidence at Termination.** Upon terminating learning, participants were asked *"How confident are you that the robot has correctly learned the rule?"* with a 5-pt Likert Scale.

M6 **Frustration.** After the successful completion of each trial, participants were asked to respond to the prompt, *"I found teaching this robot to be..."* on a 5-pt Likert Scale with ends at "Extremely Frustrating" and "Extremely Pleasant."

M7 **Ease of Use.** After the successful completion of each trial, participants were asked to respond to the prompt, "*I found it easy to teach this rule.*" on a 5-pt Likert Scale.

M8 Utility of Feedback. After the completing the trial with feedback, participants were asked to respond to the prompt, "*I used the robot's feedback when deciding what card to place next.*" on a 5-pt Likert Scale.

8.3.2 Procedure

Each participant encountered two conditions: one without any feedback, and one where the learner presented feedback via one of the four aforementioned interaction types. In the conditions with feedback, feedback was given after every card played. Participants were fully counterbalanced between each of the interaction types, and between whether they received the no-feedback condition first or second. We additionally had a control condition wherein participants received the no-feedback condition in the first and second rounds.

All participants were first given a set of instructions describing the task. We then presented them with a tutorial game with no feedback because we wanted the participants to: (1) familiarize themselves with the task and the interface, and (2) establish a prior on the otherwise black-box learner. To enable this second point, the tutorial game ended automatically when the learner had learned the rule. In all other rounds, the participant was responsible for determining that learning had concluded. If they were correct, they continued to a brief survey about their experience; if they were incorrect, the game continued until they were correct. In the condition with feedback, participants were additionally required to evaluate every instance of feedback received. At the conclusion of the study, participants were asked to indicate which of the two conditions they preferred, and to provide demographic information.

8.3.3 Results

We collected data from 191 Prolific workers over the age of 18 and with an approval rating of 100%. Participants were required to be fluent in English. We also excluded anyone with colorblindness, due to the nature of the Set cards used in our task. We discarded partial data as well as data from anyone who failed the attention check, and balanced the number of participants in each condition. This left us with data from 170 participants, 34 for each interaction type and also the control condition where participants saw no feedback in either round. This study and recruitment procedure was approved by our Institutional Review Board.

We analyzed all ordinal (e.g. Likert scale) data using a Wilcoxon signed-rank test (Bonferonni correction $\alpha = 0.0125$). Numerical data was tested for normality using a Shapiro-Wilk test and, when normal, was analyzed with a paired t-test; otherwise, it was also analyzed with a two-sided Wilcoxon signed-rank test. We treat each Likert question as an individual item. Furthermore, we note that one limitation of our between-subjects design is that Likert item comparisons are highly subjective and therefore prone to interpersonal variance. For completeness, we performed our analysis with and without outlier rejection for samples more than three standard deviations from the mean; no differences were found.

Overall, the findings of our online study suggest that while feedback is often – though not always – helpful, there doesn't seem to be a statistically significant effect in the efficacy of different interaction types. This contrasts with our expected findings and can be explained, in part, by the findings of the talk-aloud study we describe in the following section.

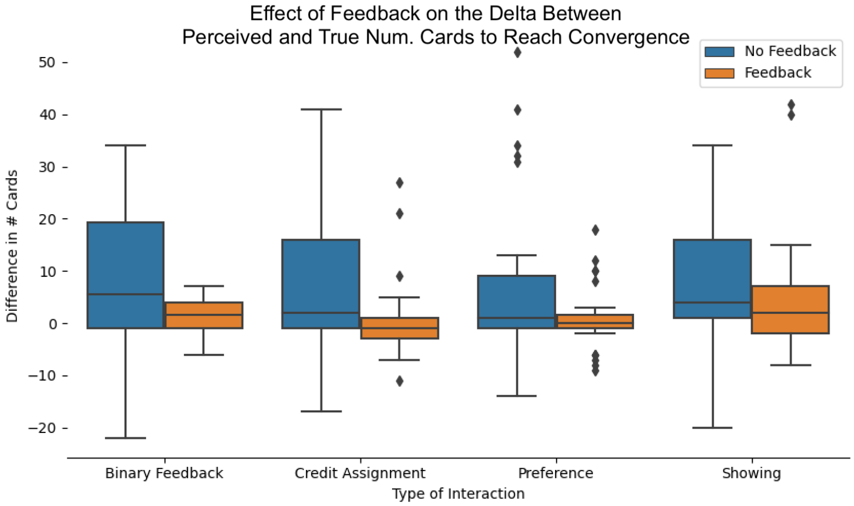


Figure 8.4: In general, feedback helped to narrow the gap between human teachers' perceptions of convergence and when the agent actually converged. However, statistically significant differences were found only with respect to *credit assignment* and *binary feedback*.

H1: Teaching with feedback will be more accurate than teaching without feedback. This hypothesis was partially supported; teachers seemed to see an improvement in the accuracy of their mental model of the learning agent. Statistically significant differences were found with respect to the difference in the number of cards at which the user perceived the agent to have learned versus when the agent actually converged (M1) for *binary* ($p < 0.005$, Z = 94.5) and *credit assignment* ($p < 0.01, Z = 97.5$). Because we use a Bonferonni correction, we do not consider the difference between no-feedback and *showing* or *preference* to be statistically significant ($p < 0.05$ for both).

We did not find any statistically significant differences in failure rate between no-feedback and any of the interaction type conditions. Given our corrected analysis, we do not consider the difference between no-feedback *credit assignment* ($p < 0.05$) to be statistically significant. However, we notice a trend for people to terminate slightly too early in the *credit assignment* condition, perhaps indicating some degree of overconfidence.

H2: Teaching with feedback will be more efficient than teaching without feedback. This hypothesis was somewhat supported. Statistically significant differences from the no-feedback condition were found with respect to the the number of cards needed for the agent to learn the rule (M3) in the case of *preference* ($p < 0.01, Z = 91.5$), *combined binary* ($p < 0.005, Z = 81.5$), and *credit assignment* ($p < 0.005, Z = 91.5$). However, there was no statistically significant difference in the number of cards needed for the agent to converge between the no-feedback and *showing* feedback conditions.

Furthermore, statistically significant differences were found with respect to the average duration (ms) needed to select a card to play (M4) in the case of *binary* ($p < 0.001, T = 47.0$), *credit assignment* ($p < 0.001, T = 82.0$) and *showing* ($p < 0.001, T = 84.0$). We note that *showing* becomes statistically significant when outlier rejection is performed. There was no statistically

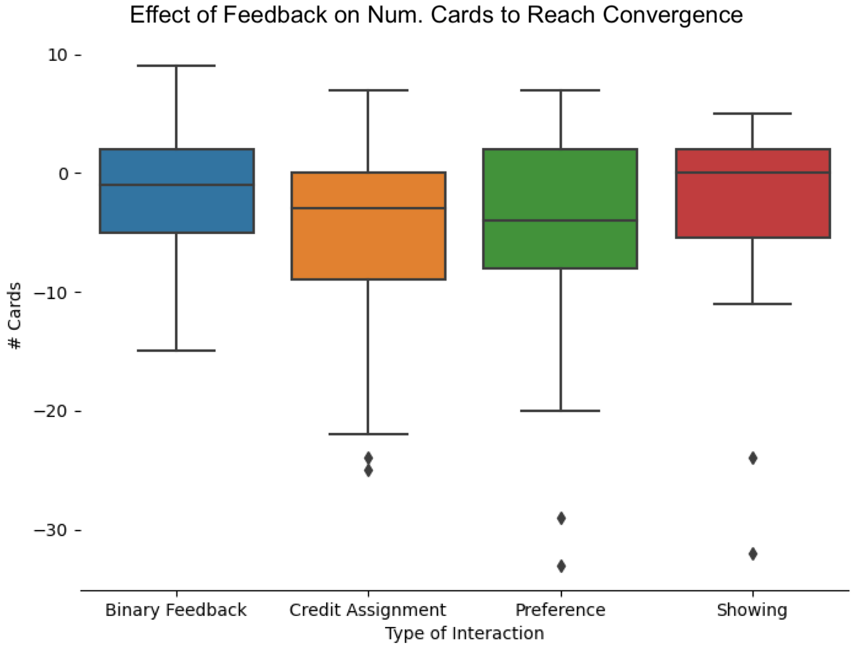


Figure 8.5: This figure plots the *difference* between the number of cards needed for the learner to converge on the true rule in the conditions with and without feedback. The presence of feedback generally appears to have reduced the number of cards needed for the learner to converge, implying that teaching efficacy was increased. Statistically significant results were found for every interaction type other than *showing*, which indicates that it's not just the presence of feedback, but also the type of feedback, that accounts for this difference. However, across interaction types, there were no significant differences.

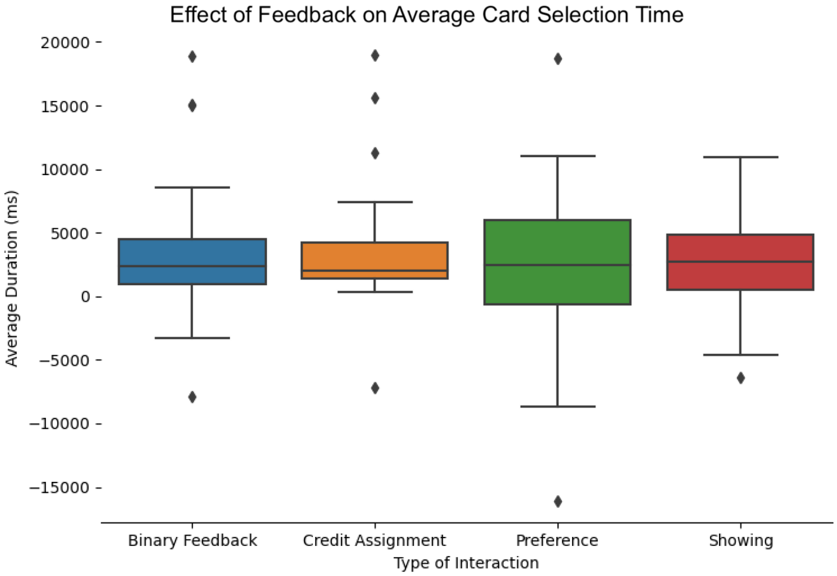


Figure 8.6: This figure again plots the *difference* between the average amount of time taken by each user to select a card to play in the conditions with and without feedback. Feedback generally seems to increase the amount of time teachers spend in selecting their next action, as compared to when they receive no feedback from the learner. This may in part be due to the increased cognitive overhead of updating their model of the learner, and deciding what information to share in light of that model.

significant difference between no-feedback and *preferences* given our use of a Bonferonni correction ($p < 0.05$).

H3: Different interaction types will differently affect teaching accuracy. This hypothesis was not supported given the results of our online study. We did not find a statistically significant difference from the no-feedback condition in the difference in the number of cards at which the user perceived the agent to have learned versus when the agent actually learned (M1) across interaction types, nor in the failure rate (M2).

H4: Different interaction types will differently affect teaching efficiency. This hypothesis was not supported given the results of our online study. We did not find a statistically significant difference across interaction types in the delta between no-feedback and feedback for number of cards needed for the agent to converge (M3), or the average duration (ms) needed to select a card to play (M4).

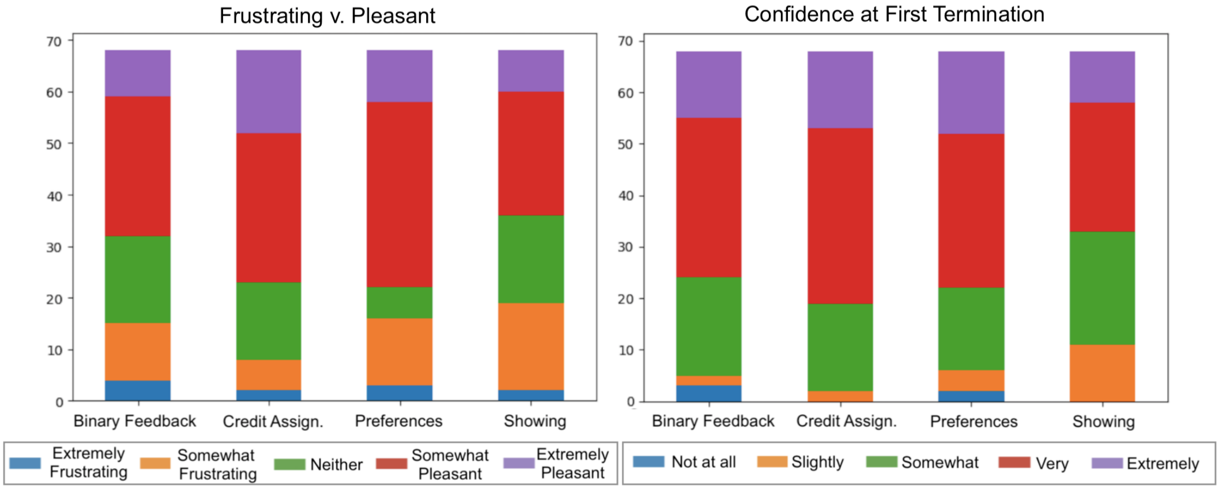


Figure 8.7: We did not find statistically significant differences from the no-feedback condition across interaction types in the difference in whether participants perceived teaching the robot to be frustrating or pleasant, or their self-reported confidence at first termination. As expected, most of the responses lie between the two extremes of the Likert scale, with a positive bias.

H5: Human teachers will be differently influenced by each interaction type for feedback (e.g. with respect to their confidence in their own teaching, and the subjective experience of teaching). This hypothesis was not supported given the results of our online study. We did not find statistically significant differences from the no-feedback condition across interaction types in the difference in: self-reported confidence at the first termination (M5), whether participants perceived teaching the robot to be frustrating or pleasant (M6), or how easy participants found it to teach the robot (M7). Neither did we find a statistically significant difference in whether participants felt they used the feedback in deciding what card to place next (M8).

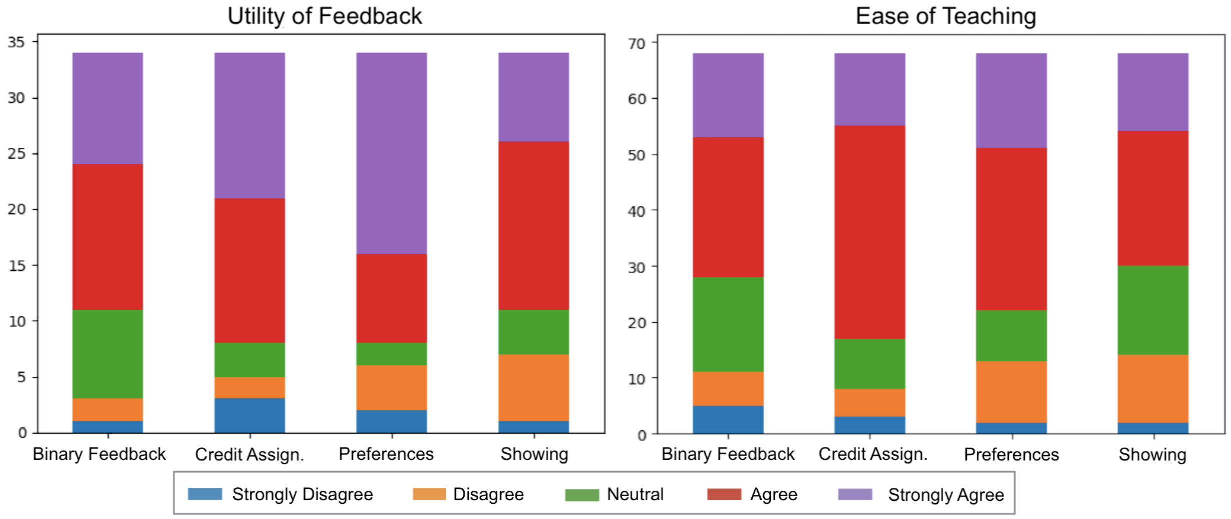


Figure 8.8: We did not find statistically significant differences from the no-feedback condition across interaction types in the difference in whether participants perceived feedback to be useful in determining their next steps, or in how easy they found it to teach the robot However, the data show interesting trends such as that preferences were thought to be extremely useful in determining next steps as compared to other interaction types.

8.4 Evaluation: In-Person Talk-Aloud Protocol

While the online study, presented in section 8.3, allowed us to easily collect and compute aggregate statistics on quantitative learning and teaching efficacy, it is only one part of the picture. To more fully understand how human teachers respond to these different forms of feedback, we wanted to obtain free-form, unprompted opinions and reactions from people. However, to perform in-person interviews with nearly two-hundred participants would be an extremely time-consuming task. Therefore, we decided to augment our online study with a smaller in-person cohort who would participant in an unstructured talk-aloud study.

Our talk-aloud study is largely the same as the online study, with a few important adjustments. First, we redesigned it as a within-subjects user study, such that every participant sees every interaction type that the learner can use when presenting feedback. Furthermore, we removed the no-feedback control condition because we did not run statistical tests given the small sample size and aimed to minimize participant fatigue.

This talk-aloud study also featured a new condition wherein participants had the opportunity to choose, at each turn, what format the learning agent should use when presenting feedback. Thus, the within-subjects independent variable had five levels: *demonstration, preference, binary feedback, credit assignment,* and *choose-your-own.* We counter-balanced the first four levels using a Balanced Latin Squares approach. However, the *choose-your-own* condition always appeared last, after participants were exposed to each of the interaction type conditions.

8.4.1 Procedure

The procedure for this study was similar to the online-study procedure, as described in Section 8.3.2. However, participants encountered five different feedback conditions and did not encounter any no-feedback condition (apart from the tutorial game). The entire study, including the tutorial, took between 45 and 90 minutes. Participants were instructed to narrate their thoughts (e.g. feelings, strategies, questions) throughout the process. If participants were silent or did not explain a decision, the interviewer prompted them with a neutral phrase such as, "Can you describe what you're thinking at the moment?" Finally, at the conclusion of the study, participants were asked to provide a forced ranking between the *demonstration, preference, binary feedback,* and *credit assignment* feedback types. We captured both screen and audio recordings of participants going through the experiment.

8.4.2 Data Overview and Coding

We collected data from 8 participants, sourced through a combination of word-of-mouth recruiting and via Carnegie Mellon University's Center for Behavioral and Decision Research recruitment portal. There were a range of ages represented: of these participants, three were in the 18-24 age range, two were in the 25-34 age range, one was between 25-44 and two were between 45-54. Half of the participants self-identified as men, three self-identified as women, and one preferred not to share. Furthermore, familiarity with artificial intelligence and robotics varied from not at all familiar to extremely.

In order to do a thematic analysis [92], two coders each transcribed half of the recordings collected. Afterwards, they coded one of those recordings together in two-passes. The first pass was used to generate low-level notes and observations; the second pass was then used to synthesize these into higher-level codes. Both coders then independently coded another transcript using these high-level codes. However, this yielded a Cohen's Kappa of 0.658 which, while considered substantial agreement, was lower than desired. Therefore, the coders convened to review their coding approaches as well as the code-bank they had previously created. Afterwards, they independently coded a third transcript, and this time achieved a Cohen's Kappa score of 0.786. Given this sufficiently high inter-coder reliability (ICR) score [90], the coders then split the remaining transcripts between themselves and completed the coding process separately. The final code-bank contained 15 high-level codes.

After all of the transcripts were coded, they were analyzed using affinity diagramming. We found a natural separation between observations related to how participants opted to teach the learning agent and observations that served as a meta-analysis of the different interaction types for feedback, moments of personification, and mistakes or misinterpretations of the system. We discuss each of these in more detail in the remainder of this chapter.

8.4.3 Findings

Our study revealed a relationship between using different interaction types for feedback, how participants' build their mental models of learning agents, and participants' own teaching personas. This relationship graph is detailed in Figure 8.9, and will be discussed in more detail in the remainder of this section. We discover that participants largely use feedback for two distinct, but related, purposes: first, to identify what the learning agent knows, and second, to determine what teaching action they should take next. Furthermore, we found that *teaching personas* seem to have large effects on how participants engage with feedback from the learning agent. We define a teaching persona to encompass an individual's preconceived notions, strategies and beliefs about themselves as a teacher, the agent as a learner, and how the teaching process should and will unfold.

This may in-part explain the findings from the online study; across a population of users with different teaching personas, we would expect to find any differences between interaction types to be washed out. Our in-person participants, however, demonstrated strong preferences between interaction types. In this section, we will first describe the aforementioned components of training a learning agent and teaching personas. Then, we will describe the complex teaching/training process that unfolds around these. Finally, we will discuss how these are informed by, and inform, participants' commentary on the task and the different interaction types for feedback.

Participants constructed a model of the learning agent involving both what the learning agent knows, and what it needs to know. Together, these two components comprise the human teacher's model of the learning agent. They also respectively correspond to the Gulf of Evaluation and Gulf of Execution often discussed in the Human-Computer Interaction literature [99]. We underscore that because the agent is a black-box to the participants, these are not mutually exclusive categories: knowing what the agent does *not know* does not necessarily tell you what it does, and vice versa. We further describe our observations below:

1. **Feedback that reveals what the learning agent knows can be used to assess progress and teaching efficacy.** Participants often relied on feedback in order to determine what the learning agent had deduced so far, and what it was paying attention to in their teaching. Our experimental setup was designed such that they had no other cues to rely on, to simulate commonly used machine learning and artificial intelligence training schemes in crowdsourced data collection processes.

   1a. **Participants at times opted to use feedback purely as a tool for evaluation, e.g. via testing and confirmation-seeking behaviors.** This would often manifest as playing cards to "reinforce" something they had already taught and checking the feedback against their expectations. Some participants demonstrated this behavior in response to the learning agent appearing to understand the rule earlier than they expected; for example, one participant said they "...can't tell if [this was a] fluke" prior to engaging in testing behavior. Some participants were more biased towards this view of feedback than others, and would rate feedback as more helpful when they perceived it as correct. Others switched to testing behaviors as learning progressed; either when they experienced confusion about what the learning agent had understood, or when teaching had progressed sufficiently and they wanted to validate learning before terminating the session.

2. **Feedback that directly asks about a region of uncertainty, reveals more of the agent's state, or is perceived as inaccurate can be used to identify next teaching steps.** Participants alternatively used feedback as a guide to inform their teaching strategy. For ex-

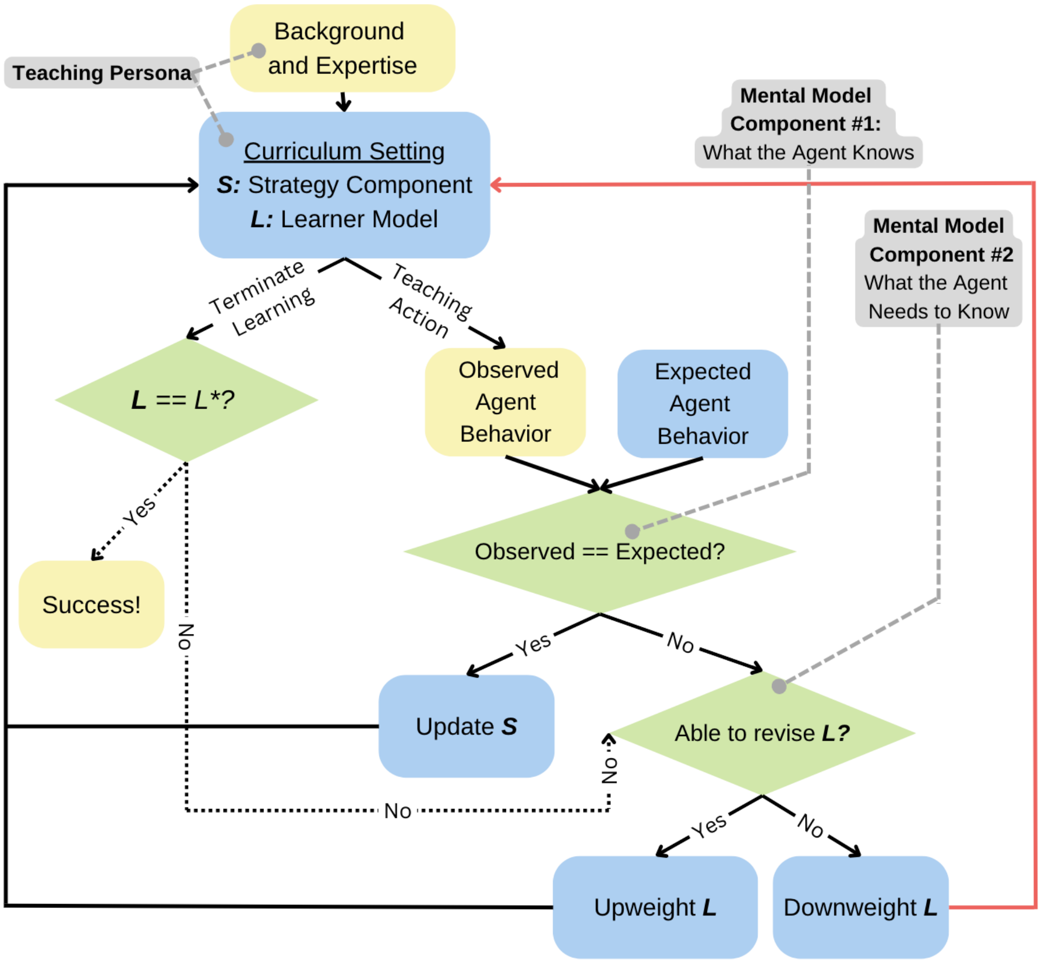


Figure 8.9: We found a relationship between how participants' teaching personas influenced their strategies and approaches to forming mental models of a learning agent. Furthermore, we found that when feedback was unhelpful in building those mental models, participants were increasingly likely to rely solely on their teaching personas and initial strategies and less likely to engage with the learning agent's feedback. For example, a *learner-led* teacher might have a low weight on **S** and a high weight on *L*; in contrast, a *curriculum-driven* teacher might have a high weight on **S** and a low weight on *L*.

ample, the *showing* type feedback was commonly perceived as a request for information; this aligns with its intended formulation, as described in Section 8.1. Similarly, three of the participants noted that the *preference* type feedback was informative in helping them understand what information they might need to reinforce, or what teaching action to take next. Finally, feedback perceived as incorrect was sometimes effective in helping participants identify their own confirmation biases; upon receiving such feedback, participants would express surprise at the unexpected feedback, review their teaching to understand how the learning agent reached its conclusion, and finally be able to identify the gap between what they had been intending to teach versus what they had actually been teaching.

Participants opted to teach rules hierarchically, and used feedback cues to adjust their strategy. This finding corroborates the decomposition of rules we used to generate feedback, as discussed in Section 8.1.1. Rules were perceived by participants as being composed of: a primary rule (or "general" rule as many referred to it), exception, and Bins. In order of occurrence, participants leveraged the following strategies: (1) teaching the primary rule first, (2) teaching the exception rule first, and (3) teaching by Bins, one by one. When using the first hierarchy, which was the most common approach, participants still tended to teach the rules Bin by Bin. However, they would ignore the exception rule, and return to teaching it after they felt the learning agent understood the primary rule for each Bin (e.g. "Diamond" cards belong in Bin 1, "Squiggle" cards belong in Bin 2, and "Ellipse" cards belong in Bin 3).

However, at a more granular level, participants would state they were "choosing cards randomly." In other words, participants could generally identify what semantic component of the rule they wanted to teach (e.g. "Diamond" cards go in Bin 1), but would then choose a random card from the set of cards that focus on that component (e.g. choosing any color "Diamond" card). As teaching progressed, participants would try to provide good coverage of cards played and also optimize for cards that the learner had not seen before.

Participants also demonstrated some interesting teaching behavior in response to feedback from the learning agent. When confronted with feedback demonstrating an incorrect understanding of the true rule, participants often selected their next card using two different types of counterfactual reasoning. For example, consider feedback that suggests "Red" cards go in Bin 1, when actually "Diamond" cards go in Bin 1. In a *same-bin counterfactual*, participants might choose to play a "Purple" card in Bin 1 to teach the learning agent that color does not matter. Alternatively, in an *across-bin counterfactual*, participants might choose to play a "Red" card in a different bin, to disprove the hypothesis that all "Red" cards belong in Bin 1. Participants often demonstrated both types of counterfactual behavior, instead of strongly biasing towards one or the other. Reasons for selecting one type of counterfactual over another are, at this time, unclear; further study might reveal the nuances between them. However, if participants received feedback that they interpreted as an accurate understanding of the component of the rule they were currently trying to teach, they would then choose to progress to the next phase of their teaching strategy (e.g. move from teaching the primary rule to teaching the exception rule).

Failures to assimilate feedback into the working model of the learning agent resulted in poor learning outcomes. We consider bad outcomes to include ones where the human teacher either cannot identify what the learner knows, or cannot identify which teaching action they should take next. We identified three primary types of bad outcomes: (1) early terminations, (2) decreasing confidence, and (3) unchecked confirmation bias. We consider an early termination one where a participant incorrectly believes that the learning agent has learned the rule and chooses to terminate learning, when in actuality the learning agent has yet to converge. Decreasing confidence appears in the latter half of teaching, as participants may feel confident in the initial part of teaching and start to lose confidence if they don't receive the feedback they expect. Finally, confirmation bias is a common cognitive shortcoming in people. In our games, participants would sometimes be unable to recognize that the sequence of cards they chose to sort could be explained by multiple rules. Often, feedback was helpful in recognizing this bias. However, at times participants were unable to understand the feedback; this would lead to an inability to understand what teaching actions to take. These poor learning outcomes usually led participants to disengage from the interactive learning process by ignoring or dismissing the agent's feedback and relying more heavily on their own teaching persona (e.g. preconceived notions, strategies and beliefs). This is in contrast to the paradigm discussed previously, wherein participants were able to progress through, and adjust, their strategy by interpreting and responding to the agent's feedback.

We note that all of these outcomes are inextricably correlated with negative emotions such as confusion and frustration. Confusion may have multiple sources, but we primarily observed it as a response to either an inability to interpret the content or relevance of a piece of feedback (e.g. not understanding why the learning agent gave a particular piece of feedback in response to their most recent teaching action), or an inability to know how to respond to the feedback (e.g. recognizing they had led the learning agent astray, but not knowing how to take repairing actions). This generally progressed to frustration, at which point participants would express that feedback consistently has no relevance to their teaching, start to play cards randomly, and ascribe traits such as "resistance" and "obstinacy" to the learning agent. We further observe that some instances of participants not finding feedback relevant is correlated with confirmation bias: it may not be relevant to the rule they are trying to teach, but relevant to the majority of rules remaining as viable hypotheses.

When possible, participants used multiple feedback types to compensate for any confusion, as well as their shifting desires for detailed feedback. The majority of participants used more than one type of feedback in the *choose-your-own* condition. The most common approach was to use a combination of the *binary feedback* and *credit assignment* feedback types. Common reasons cited for this behavior included using *binary feedback* to highlight one feature of importance, and using *credit assignment* for more granular feedback. However, two participants started with *preference* feedback and switched to either *credit assignment* or *binary feedback* as learning progressed. Interestingly, their motivations were different: the participant who switched from *preference* to *credit assignment* seemed to do so because they were transitioning from a teaching stance to an evaluating stance as described by their statement "...I think, if I choose credit assignment it will be more explicit about why I put that card there and... see if I can determine that it's learning the rule then." On the other hand, the participant who switched to *binary feedback* did so when they were feeling confused about the *preference* feedback they were receiving. We also note that one participant opted to receive the *showing* feedback in the beginning, and then switched to *credit assignment* towards the end; this participant justified their selections by observing that they felt *showing* was valuable for helping them to identify next steps, whereas *credit assignment* was then useful to test progress. Several participants also noted that while they enjoyed being able to choose their own feedback, it did add some complexity to the task. This is likely due to the increased cognitive load of having to additionally infer which feedback is likely to contain helpful content.

In general, participants felt most positively about *credit assignment* feedback, and most poorly about *showing* feedback. When asked to rank each of the interaction types for feedback in order of their most, to least preferred, half of the participants gave the following order: *credit assignment, preference, combined binary,* and *showing*. This was also the ordering given by tallying participants' votes using a Condorcet method. The other four participants were very varied. All had *showing* as their least favorite; two of them had *preference* as their first choice, while the other two had *combined binary* as their first choice. In each of those subsets, half had *credit assignment* as their second choice, while the other half had it as their third. This reveals both a strong trend, as well as a sizeable degree of individual variation. In this subsection, we will first give an overview of subjective perspectives on each of the interaction types for feedback, and then provide an emergent hypothesis for what may drive this individual variation.

1. ***Showing* feedback felt vague and difficult to interpret.** Half of the participants explicitly shared that feedback of the *showing* format felt unclear, vague, or unhelpful. Every participant ranked it as their least preferred interaction type. However, two participants expressed that the feedback could be helpful, because it shows what elements of a card the learning agent is thinking about. Furthermore, as we will discuss shortly, this interaction type seemed to lend itself the most to personification of the learning agent.
2. ***Preference* feedback was perceived as providing a lot of information, but became less helpful towards the end of teaching.** Participants praised the *preference* type feedback for providing specific information and insight into the top two options the agent is considering (we note that this corroborates the implicit information we chose to encode into the feedback, as described in Section 8.1). However, one-fourth of the participants expressed that this feedback type seemed to become less helpful as time went on. We believe that this may be because as teaching progresses, participants desire extremely specific feedback; because the *preference* type feedback always compares two options, there may be uncertainty as to whether the agent has fully determined the rule.
3. ***Combined Binary* feedback was controversial: some people found it easy to understand, others found it difficult.** Participants who liked this feedback type did so because they found it straightforward and easy to understand. Three of the participants independently observed that the *combined binary* feedback was helpful early on in the teaching task. However, this feedback was sometimes perceived as "non-intuitive" because it would "jump-around." This is likely because the algorithm is designed to provide both positive and negative feedback after a certain confidence threshold is reached; so, a teacher may play a "Red" card in Bin 1 expecting that the agent will associate "Red" with Bin 1, but the feedback might be "'Purple' cards do not go in Bin 3." At least one participant actually liked this property, and shared that it was helpful when the agent extrapolated information that the participant had not explicitly shared.
4. ***Credit Assignment* feedback was liked by every participant, and often described as being "explicit".** As mentioned previously, this feedback type was generally the most popular kind. Participants described it was explicit, clear, straightforward, and detail-oriented. This feedback type was particularly preferred by participants who often exhibited testing behaviors and strong teaching personas (e.g. strongly identified themselves as a teacher).

Participants' strategies, responses to feedback, and attitudes towards different interaction types were heavily influenced by their teaching personas. Half of the participants strongly

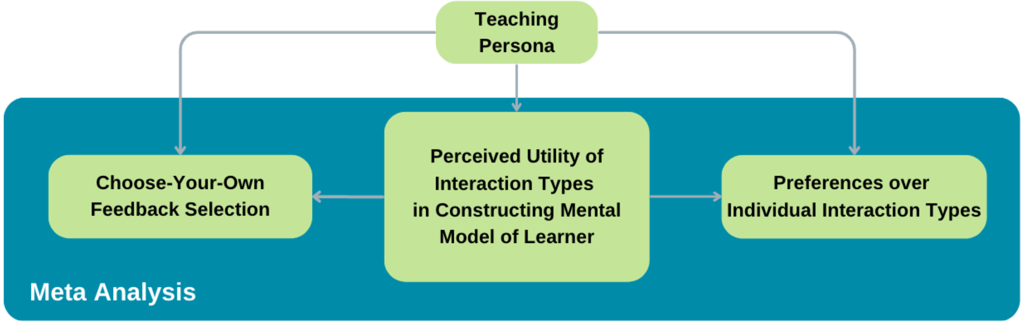

Teaching Persona
Choose-Your-Own Feedback Selection
Perceived Utility of Interaction Types in Constructing Mental Model of Learner
Preferences over Individual Interaction Types
Meta Analysis

Figure 8.10: Participants' teaching personas influenced how they perceived and leveraged different interaction types, which in turn affects how they choose to engage with feedback in the *choose-your-own* session.

identified with being the teacher (*curriculum-driven* teachers), whereas the other half were generally more open to having their teaching process led by the learning agent (*learner-led* teachers). We were unable to identify any predictor of which category a person would belong to; it may simply be a feature of the variability of individual personalities. However, this division had strong influences on how a participant would interpret and respond to the learning agent's feedback (Figure 8.10, Figure 8.9). Participants who strongly identified as the teacher often had a myopic view of feedback and used it mostly as an evaluation tool; they would often stick to their strategies regardless of the feedback received, and express frustration if the learning agent seemed not to understand their intentions. The other set of participants were more open to adjusting their teaching strategy in response to feedback from the learner, and were more likely to interpret inaccurate feedback as a reflection of their prior teaching than a reflection on the learning agent. This is perhaps best exemplified by the sole participant who did not like the *credit assignment* feedback. This participant felt that it was confusing, and that the agent could have been giving feedback about "any of the features." That participant preferred the *combined binary* and *preference* feedback types, citing that they were more helpful in sharing what the agent was "thinking."

Participants perceive some interaction types as more conducive towards assisting with identifying learner progress, whereas others as more conducive towards identifying next teaching steps to be taken. While all interaction types are designed to provide information about both components of constructing a model of the learning agent, participants may evaluate them on how well they enable each component. This may be a result of different teaching personas, and the explicit and implicit information conveyed by each type of feedback. By their construction, we would expect that *showing* and *preferences* would be most valuable for identifying next steps because they discuss alternative hypotheses, while *combined binary* and *credit assignment* would be best for assessing progress because they only share information about what they are most confident about.

In practice, this hypothesis was somewhat upheld for *showing, preference,* and *credit assignment* type feedback. Participants often used *credit assignment* to identify progress, and to receive detailed feedback; as mentioned previously, many participants used the *credit assignment* feedback as part of the *choose-your-own* condition. One participant explicitly mentioned that they liked this feedback type because it “lets me, as teacher, determine the pace of the subjects.” Participants also seemed to find the *preference* feedback valuable for identifying next steps to take, citing reasons such as that it “...shows what [the participant] needs to reinforce.” While most participants did not enjoy the *showing* feedback, there was one participant who began the *choose-your-own* condition with it, because they felt it was the most helpful in identifying next steps by providing context as to what the learning agent was observing and trying to understand. Furthermore, at least one other participant also responded to the *showing* feedback as if it were a request for information; two others were able to identify and rectify confirmation bias as a result. Others, who didn’t enjoy it, would make statements such as “[*showing*] doesn’t align with my teaching strategy.” The *combined binary* feedback, however, had mixed perceptions. Most participants seemed to view it as more useful for identifying next steps, sharing things like “it has its own pace” and “helpful when it extrapolates.” A few others found it more suited for identifying progress, citing that it was “straightforward,” and “easy to understand.”

In general, we found that participants who felt very strongly about themselves as teachers tended to prefer *credit assignment*, and *combined binary* feedback. Another set of participants, who were more willing to be led by the learning agent, preferred *preference* and *combined binary* feedback. While no one strongly preferred *showing*, it was used only once in the case of a participant who wanted to let the learning agent initially help lead the teaching process. It is possible that these lie on some sort of spectrum, with *credit assignment* and *showing* at either end, and *preference* and *combined binary* feedback in the middle. The salient difference between *preference* and *combined binary* feedback, then, would be the amount of information shared. Because *preference* encodes more information, it may feel less straightforward and therefore explain the bias most participants have towards *combined binary* feedback instead. It is also possible that the simple nature of this task minimizes differences between interactions; however, as strong subjective opinions have been found to emerge, matching feedback to individual personalities and desires is likely a promising area for future research.

Personification of the learning agent. Some participants were more likely to personify the learning agent than others, regardless of the interaction type being used to provide feedback. However, there were some trends. The *Showing* type feedback was, on multiple occasions, referred to as a “philosophizer”, likely due to its use of the phrase “I’m wondering if...”. Some participants felt that the *preference* type feedback felt as if the robot was “thinking too much.” We also observed that participants tended to ascribe a male gender to the robot. Sentiments about agent feedback may affect perceptions of physical manifestations of the agent as well. One participant stated “Don’t you think how it’s interesting how we see this robot profile’s backside and not its face, it doesn’t even look like it’s watching” after expressing increased frustration about the agent’s “inaccurate and gibberish” feedback as compared to previous rounds. This suggests a perception of the agent as turning away from, or not participating reasonably in the shared teaching-learning task. It should be noted that in actuality, the robot was facing the participant at all times.

“One card in Bin 1?” and other mistakes and misconceptions. Most participants did not experience major difficulties when participating in the study. However, two participants struggled with the auto-placement of cards. These participants made comments that indicated they believed the learning agent to be playing those cards (e.g. “So ”Squiggle”... so the ”One” card ”Squiggle” went into... Box 2, which is good.”). One participant expressed confusion over the convention of calling Set cards with two shapes on them “Two” cards and briefly erroneously believed the statement “‘Two” cards go in Bin 1” to mean only two cards could belong in Bin 1. Another participant voiced that the appearance of the button allowing them to terminate learning (this button appears after the first card has been played) appeared as a request for them to terminate. Finally, several participants indicated that the final *choose-your-own* condition was a bit difficult because they had trouble remembering which feedback type was which. Many of these participants either asked for, or were proactively given, a refresher on each of the feedback types. Despite this refresher, there was at least one participant who shared that they had erroneously selected one type of feedback when they had intended to select another.

8.5 Discussion

Our evaluations in two separate studies underscore the need for thoughtfully pairing interaction types for sharing feedback with the needs of human teachers. Most, though not all, inversions of interaction types for learning from human feedback as transparency mechanisms yield significant improvements in teaching and learning outcomes. Our takeaway is perhaps most saliently demonstrated by the fact that the results of our between-subjects online study (each participant saw a *no-feedback* and a single *feedback* condition using one of the four implemented interaction types) did not show statistically significant differences between different interaction types; this may initially seem to indicate that, for a human teacher receiving feedback, they are all interchangeable. However, in our in-person within-subjects study (each participant saw all four interaction types, as well as a *choose-your-own* condition), participants expressed strong preferences between the different interaction types. This suggests that, at a population level, individual differences balance out attitudes towards different interaction types. However, for an individual, there may very well be some interaction types that are better or worse suited to their particular teaching persona.

This is further illustrated by the fact that, in the online study, the *Showing* interaction type did not show statistically significant improvements over the baseline condition of no-feedback. However, in our smaller talk-aloud study, at least one participant still chose to use the *Showing* feedback; this implicitly reveals that for some people there is some utility to this type of interaction that may not have been captured in the online study. This participant later shared that they felt the *Showing* mechanism was helpful initially for identifying what the learning agent was trying to learn. However, other participants viewed this interaction type as difficult to understand.

Indeed, in the diversity of ways in which participants navigated the *choose-your-own* condition in our talk-aloud study, we see that different human teachers have different needs, as well as expectations of a learning agent. Misalignment between those needs and expectations can lead to negative learning outcomes, such as confirmation bias, or miscalibrated confidence in the learning agent’s model of the task. Even interaction types that were generally popular, such as the *credit assignment*, were not universally beloved; at least one participant expressed a distaste and confusion around this interaction type. We note that this may also be influenced by the particular format we chose for these interactions, and further research should explore alternative formats. It is also interesting to note that, in general, people most preferred the *credit assignment* style of receiving feedback; however, as we discuss in Chapter 7, people uniformly dislike *credit assignment* when it comes to providing feedback. This asymmetry warrants further exploration. Ultimately, we must move away from a one-size fits all approach, and begin to understand how to dynamically adapt learning processes to suit not just algorithmic learners, but human teachers as well.

Chapter 9

Discussion

In this dissertation, we have focused on the role of interactions in learning from human feedback and other HiLL paradigms. We first constructed a principled, MDP-based approach for describing and comparing interactions against each other. Using this, we contributed a taxonomy of interactions used in learning from human feedback comprised of four primary archetypes: *Showing* (e.g. demonstrations), *Sorting* (e.g. preferences and rankings), *Categorizing* (e.g. binary feedback and label assignment), and *Evaluating* (e.g. credit assignment and corrections). We then showed that the expected information gain of queries of each of these interaction types differs, and developed an active learning algorithm, INQUIRE, that can dynamically pose queries across multiple interaction types in order to ask the most informative query given a learning agent's current task knowledge. However, we cannot solely optimize for informativeness of a query because there are also differing costs associated with each interaction type; there exists a trade-off between the informativeness and the costliness of any interaction, and algorithms we design must take this into consideration. In Chapter 7, we measure some of the different costs with respect to cognitive load, performance, and usability and show that there are exist significant differences in cost across interaction types.

We also observed that the information shared by a human teacher to a learning agent, which we have shown to be strongly affected by the interaction type used, is only one part of the knowledge-sharing picture. It strictly controls how information can be shared, both explicitly and implicitly; however, ultimately the human teacher makes a determination of *what* should be shared. And, this determination is influenced by their perceptions and expected outcomes of the learning agent that they are training. With this in mind, we evaluated the symmetry of information sharing by implementing one of each of the interaction archetypes for learning from human feedback as a vehicle for sharing information about a learning agent's model of a task back to a human teacher. Interestingly, our results were mixed. On balance, most of the interaction types were effective in providing feedback about the learner's model of the task but the type of feedback that was helpful varied strongly from person to person and, across a wide population of participants, there were no significant differences between the efficacy or accuracy enabled by different interaction types.

The work presented in this thesis is one step towards machine learning systems that are teachable – not just by machine learning and artificial intelligence experts, but by laypersons with a variety of needs, desires, and expertise. In the remainder of this chapter, we will discuss some of the limitations of the work we have presented, possible real-world applications of this work and its extensions, and promising directions for additional research.

9.1 Limitations

In order to answer the research questions we posed in this dissertation, we made several simplifying assumptions. These are certainly limitations of our work, and any extensions which loosen these assumptions would be an important next step. One of the large assumptions of this body of work is the simplifying assumption that salient differences between interaction archetypes would manifest even in the most simple examples (e.g. preferences for a *Sorting* interaction, rather than providing a fully ranked list of n candidates).

While this seems to be true for how people perceive being asked to provide feedback in these formats, we must consider two alternative options. First, it may be the case that there exists a non-linear relationship such that differences between interaction types may lessen rather than increase, or stay the same, based on complexity of the interaction and/or the task. Second, we have seen that the presence of these differences varies based on how the interaction is being used (e.g. as a means to provide, versus receive, information) and should take this into further consideration in constructing more dynamic interactive learning systems.

Additionally, we only briefly looked at classification tasks; however, a vast majority of the tasks that people want to use machine learning and artificial intelligent systems for reduce to classification problems. As such, it would be prudent to further investigate our approaches and findings in that context as well. Our work, particularly that described in Chapter 8, relies heavily on toy problems; extending this to larger and more complex domains (possibly continuous, as in our work with INQUIRE) would be a necessary next step towards making our findings ready for real-world adoption.

In the remainder of this chapter, we present in more detail possible extensions to the work we have presented thus far – both with respect to domains in the real world (Section 9.2), as well as research questions (Section 9.3) that should be pursued further.

9.2 Extension to Real-World Scenarios

The examples we have provided throughout this thesis have relied on smaller toy problems to make solving the presented research questions more tractable. In this section, we give a brief overview of a few real-world domains where such work is immediately or near-term applicable. The research presented herein has to do with making the interactions between humans and intelligent systems more seamless, personalized, and easy to use; the domain presented subsequently therefore are all similar in that they focus on use-cases where individuals are repeatedly interacting with – from their perspective – a single, static learning agent.

9.2.1 Reinforcement Learning from Human Feedback

As of 2022, with the commercial release of large language models (LLMs) such as OpenAI's ChatGPT and Google Bard, there has been a surge of interest in reinforcement learning from human feedback (RLHF) [12, 34, 74, 124, 141]. One way in which RLHF is used is to have human coders evaluate two pieces of text generated by a LLM and select which one they believe to be better, or contain more accurate information, or sound more friendly, or any other metric of choice. This form of preference-based learning is currently the most popular approach used for RLHF. Our research suggests that other interaction types, or more dynamic approaches, may ultimately prove more fruitful and sustainable.

Passages of text are extremely information dense and evaluating between two (or more) options may be non-trivial. For example, if one seems more friendly and helpful but provides incorrect information, is that better or worse than a passage that has correct information but is constructed in a garbled or unfriendly way? Researchers are finding that crowdworkers paid to evaluate these passages err on the side of the former [12]. However, this clearly has shortcomings, and LLMs have gained notoriety for their hallucinations and inaccuracies [89]. One could argue that by collecting sufficient data, we can discern these nuances. However, human labor is costly and their energy finite. Therefore, more granular forms of interaction, such as those belonging to the *Evaluating* class may prove more effective and valuable. A *credit assignment* approach could have evaluators assess which portions of a passage are good, and which are not. Or, using *corrections*, they might be able to modify an incorrect statement into something correct, or an unfriendly excerpt into something more genial.

However, even if using varied forms of interactions for collecting data, the fact remains that LLMs must be trained on massive quantities of data, and collecting data from people is expensive. As a result, we must prioritize being extremely resource efficient by knowing the right times to ask for human feedback and by knowing what to ask for at those times. The structure of this problem is similar to that posed in Chapter 6. Of course, the scale of the models and the type of data involved are quite different. Still, a similar approach to approximating the expected information gain from receiving a piece of human feedback given the current state of the model may be an appropriate avenue of investigation to address this problem.

Somewhat relatedly, there have been recent reports of crowdworkers using these LLMs to solve online studies on sites such as Amazon Mechanical Turk. This is a concerning problem for the field, as the quality of model development depends in large part on the quality of data collected. In essence, this reduces to a model training on itself, which eventually leads to a phenomenon known as model collapse [120]. Therefore, it is critical to retain high-value data generated authentically by human teachers. However, the appeal of LLMs for crowdwork is clear: with significantly less effort, crowdworkers can respond to significantly more postings and thereby increase their earned wages. We observe that there are some interaction types that are more prone to abuse of this nature than others. For example, generation of text can now be automated in large part. However, asking crowdworkers to provide rankings over a set of items, assign labels from a label bank, or provide granular edits to a piece of text may be slightly more robust to the emergence of LLMs.

Our suggestions above mostly have to do with the training process for these LLMs. We note that when individual laypersons interact with these LLM systems, they exhibit teaching behaviors congruous with those that we have seen during the course of our research. For example, an individual might as a question and, upon receiving a response from the agent, provide clarification or ask for an amendment of the feedback; this is, again, an *Evaluating* type interaction. Or, they might say something along the lines of 'That's great!' or 'That's not what I wanted.' which we can interpret as binary feedback signals. We can also see this when it comes to more sophisticated prompt engineering: techniques such as Chain-of-thought prompting ([134]) may be considered providing a rigorous step-by-step (action-by-action) demonstration of a single trajectory, or an example of a *Showing* type interaction.

9.2.2 Recommender Systems

Many people interact with some sort of recommendation agent on a near-daily basis whether that's through social media consumption, looking for a movie or show to watch on a streaming service, or having ads suggested to them based on their search and purchasing history. This is an everyday example of a HiLL paradigm involving laypersons: an algorithm makes a suggestion, the layperson (e.g. human teacher in this scenario), presents feedback on that suggestion, and the algorithm uses that feedback to update its model of the teacher's underlying rewards, beliefs, or preferences. However, this scenario is distinct from those that we studied in this dissertation in that these laypersons are generally not acting pedagogically. As a result, cognitive biases such as hyperbolic discounting [5] – for example, the tendency of an individual to choose things that are good for them in the short term rather than what they want for their long term selves – become much more necessary to account for and manage.

Recommender systems traditionally fall into two categories, content-based [86, 104] or collaborative filtering-based approaches [44, 117], but both of these approaches are ultimately reliant upon the data captured by repeated interactions between customers (human teachers, teaching about their own preferences) and the recommender system. Our research is applicable for both paradigms: in the first, content is generated based on users who are similar to each other and, in the second, content is generated based on the content that a user has engaged with itself. For our purposes, this is merely a difference in data format. Furthermore, while most recommender systems right now are treated as bandit problems, there has been some interest in modeling these problems in a more longitudinal fashion [4, 142], which suits the approaches we have been exploring in this dissertation.

The work we presented in this dissertation is highly relevant to recommender systems even now, particularly the sections pertaining to sharing knowledge from teacher to agent. We note that recommender systems are fairly regularly sharing what they think about a human teacher (e.g. the suggestions they make implicitly indicate the content they believe the user may enjoy, and row titles such as 'Because you liked *Spiderman*' are more explicit credit-assignment-like statements). We can superficially observe the ways in which interactions have been experimented with in these platforms: services such as YouTube and Netflix used to have a 5-star rating (a *Categorizing* type interaction) and has since moved to a Like/Dislike paradigm (still a *Categorizing* type interaction, but of minimal complexity). We can imagine augmenting existing approaches with more tooling to allow laypersons to share their feedback: curating rows of movies, for example, by having the ability to remove movies from the row that they feel do not fit either the row or their taste (an *Evaluating* type interaction). Or, they could participate in a game where they rank content they have seen against previous content they have seen (a *Ranking* type interaction). And, some systems already let people self-identify movies they have enjoyed (a *Showing* type interaction) when setting up an account in order to mitigate the cold-start problem [81]. This cold-start problem relates to something we discussed in Chapter 6: the possible reward space at the outset of teaching is often massive, and we need to find an effective way of narrowing down to the region of the reward space that contains the true reward.

It is likely that only some power-users would want to be able to use any and all interaction types of their choosing. For many, the degrees of freedom may be more frustrating than empowering, or go unused altogether. After all, many people engage with recommender systems in order to relax in a more passive fashion, and may not want to assume the role of a teacher and the burden which is implicitly imposes. However, we can imagine, based on our findings in Chapters 7 and 8, that there may be different interaction formats that work better for different people. We can imagine a potential interface wherein different people can engage with the same recommender in different ways to better suit their individual desires. Or, there could be a 'pedagogical mode' that people can enter, if they choose, wherein they can more interactively teach the recommender system about their preferences in a game-like fashion (e.g. answering pairs of preference questions) and thereby help mitigate the implicit influence of hyperbolic discounting.

9.2.3 Assistive Technologies

Artificial intelligence and machine learning solutions can be effective tools in assisting caregivers for the growing 65+ population in the United States, particularly for caregivers of those with cognitive impairments, as well as for the recipients of care themselves. Today, it is estimated that 70% of Americans who are 65 will require some form of long-term care in their life [1]. However, many families cannot afford the costs of long-term care such as assisted living and nursing home facilities, and may opt to provide at-home care [55]. At-home assistive technology solutions can and should both help to relieve caregiver burden and facilitate agency for elder Americans with MCI. The research presented in this dissertation may be particularly suitable for the development of assistive technologies that support the agency of their users.

In general, we can envision at-home social robots that can learn to adapt to the preferences of an individual in need of care (henceforth, user) in ways that feel more interactive and, therefore, provide more agency. Participants in the talk-aloud study we described in Chapter 8 observed that some forms of interactions made them feel more in control of the teaching and learning process than others. Many noted different levels of frustration with different interaction types for feedback. And, those differences were not consistent across a population of people; everyone had their own unique preferences. By understanding how an individual likes to share information, and to receive information from an agent, we can diminish frustration and increase communication efficacy. This is particularly crucial in the case of services such as at-home robots, where the agent is entering into someone's living space.

There are two primary types of learning that laypersons may be interested in: the ability to teach an intelligent agent a novel task from scratch, or to adjust its pre-existing behavior to suit their preferences, needs, or desires. However, most people interested in the assistive technologies are likely to be interested in the latter. Many household tasks are shared: eating, cooking, cleaning, etc. For example, an at-home robot may have a default approach to cleaning a room, putting away dishes, or sorting objects that a user may want to modify in accordance with their own home, lifestyle, and opinions. To be able to convey these opinions is critical for a sense of agency, as well as for these robots to provide the support that they are designed to offer. This may be particularly important for individuals from underrepresented backgrounds, who may operate with cultural norms and contexts that are somewhat out of distribution of the larger population. Learning systems that can adapt to the teaching preferences and capabilities of a variety of users will be more able to easily surface such information.

9.3 Future Work

We now discuss research questions that should be further investigated in order to build on the work we have presented in this thesis. As it stands, our results are foundational: addressing the questions below will be vital to extending this world to the real-world scenarios outlined in the previous section. Aside from the practicality of making this research widely implementable and valuable, there are interesting questions about the nature of the relationship between humans and the machine learning and artificial intelligence systems they use underpinning these research questions. Perhaps most saliently, they all seek to advance a single question: *how can learning agents better understand people, in the service of addressing human needs?* In the pursuit of an answer to this question, we propose a variety of research topics including: better simulations of human behavior, creating learning interactions that feel appropriately timed, understanding pedagogy in the context of learning agents, and exploring what it would ultimately take to create a dynamic and user-friendly way for laypersons to train machine learning and artificial intelligence systems to suit their individual needs.

9.3.1 Noisy Oracle Agents

While many of the evaluations presented in this thesis have involved human subjects, it is often desirable to test algorithmic approaches on simulated oracles. We leveraged such an oracle in order to evaluate the INQUIRE algorithm (Chapter 6). However, the oracle we constructed is an optimal agent – as are many oracles used for testing purposes, hence the moniker 'oracle.' This is a limiting assumption because, in reality, human feedback is noisy; people often do not act as expected, and there are often salient differences between the actions of any two individuals.

This point is underscored by our nuanced findings in Chapter 8, where a between-subjects aggregate study washes out individual preferences between interaction types whereas a talk-aloud within-subjects approach reveals strongly held individual beliefs about interaction types. In addition, Chapter 7 further reinforces that people have statistically significantly different reactions (e.g. with respect to cognitive load, time taken to perform the task, frustration) to providing information via different interaction types.

Therefore, we propose an extension of our work on oracle construction for INQUIRE wherein we develop a noisy-oracle. This noisy-oracle would have some degree of noise built in, and supported across interaction types. This alone is a non-trivial problem, because different interaction type are likely to encode noise in different ways as a result of the different choices and implications present for a human – and now noisy-oracle – teacher. Taking this one step further, we would want to be able to empirically evaluate how to vary this noisiness parameter across interaction types. We could then use this to re-evaluate INQUIRE against a noisy-oracle, and not just an optimal one. Furthermore, we could run a small human subjects study in order to determine how to best model this noise.

We envision this noisy-oracle to be an open-source implementation that can be used by others working in the active learning space for standardization purposes in evaluation. As of now, the field of interactions for active and interactive learning lack standardized benchmarks and evaluation frameworks; some of the work presented in this thesis has been working towards establishing such norms, and a benchmark noisy-oracle agent be a logical next step in advancing this research.

9.3.2 Cadence of Feedback

Chapter 8 presented an approach for generating local feedback about a learning agent's model of a task for use in an interactive learning setting; in our setting, feedback was generated and presented after every action taken by a human teacher. However, a more nuanced understanding of when to generate and provide this feedback would be valuable. For example, depending on the nature of the task, it may be more frustrating than informative to receive feedback after every piece of information that a human teacher shares. On the other hand, if the interval between feedback is too large, the gap between the algorithmic learner's model and the human teacher's understanding of that model may grow wider. Furthermore, the nature of when to provide feedback is likely dynamic, and dependent on the progression of learning. Our talk-aloud study showed that participants often used feedback to determine when to move on to the next phase in their teaching strategy; this indicates that feedback may be particularly valuable at critical teaching junctures.

We can envision three potential cadences of paraphrased feedback: *learner-model based*, *fixed interval*, and *on-demand*.

Learner-model based. We use model confidence in the prediction of the human teacher's rule as a way to gauge when to provide feedback. Some ways to leverage confidence in this way might be to: provide feedback whenever a new rule becomes the likely hypothesis, or to provide feedback more regularly whenever there is a sufficiently sizeable change in entropy over model predictions.

Fixed-interval. We provide feedback at a regular, predetermined frequency. Based on our studies in Chapter 8, we have a sense of approximately how long learning sessions tend to last. We can use this to determine how often feedback should be presented. For example, it can be provided when we think we are approximately 20%, 40%, 60%, 80% and nearing 100% through a learning session. Of course, expanding to new domains will necessitate additional empirical evaluation.

On-demand. We allow human teachers to request and receive feedback whenever they desire. That is, prior to sharing information, a human teacher can choose for themselves whether they want additional insight into the algorithmic learner's model. We note that collecting this data can also pave the way for a future *user-model* based approach, wherein the algorithmic learner anticipates when a human teacher might request feedback.

We can then evaluate which approach users prefer, and how learning performance fares as a result via a combination of qualitative and quantitative metrics. Metrics related to accuracy, efficiency, and influence are likely to be valuable in assessing each of these approaches. This project will predominantly be the technical contribution of a learner-model based approach to timing feedback with a smaller user study assessment.

9.3.3 Understanding Teaching Personas

Chapters 7 and 8 involved in-depth examinations of how people give and respond to feedback across the four primary interaction archetypes: *Showing, Sorting, Categorizing*, and *Evaluating.* Together, these chapters have highlighted the ways in which human teachers are similar and display group-level trends in providing information; they have also highlighted the ways in which different teachers may have different perceptions of the learner with whom they are interacting.

One of the salient ways that this manifested, as discussed in Chapter 8, is the notion of *teaching personas.* These personas encode individual preferences (e.g. having a strong *a priori* teaching strategy, versus having their teaching influenced strongly by the learning agent they are teaching) as well as the context of a particular learning interaction: is the learning agent perceived as "performing well" by the human teacher? How long has learning been progressing? Is the agent consistently underperforming in a specific area? As we saw in Chapters 5 and 6, the expected informativeness of a query of a particular information type for a learning agent varies with that agent's task knowledge; it stands to reason that, similarly, receiving feedback of a particular information type for a human teacher varies with that teacher's perception of the teaching task. Further analysis on a larger population of human subject may allow us to draw more concrete conclusions regarding how these teaching personas form, manifest, and change over the course of a teaching task.

In addition, it is possible that this dynamic approach to attitudes about interaction types carries over to how human teachers give feedback as well. There are much statistical stronger trends, and therefore likely less individual variance, in that data (likely because, particularly in tasks with an objective ground truth answer, and where the interaction format is tightly constrained, there is less room for individual expression of personality); however, this does not mean that there is no effect. Understanding how teaching personas may, or may not, emerge in presenting information would additionally be an important stepping stone towards the ultimate goal of this line of research: user-friendly interactive learning from lay persons.

9.4 Conclusion

Altogether, the work presented in this thesis demonstrates that interaction types themselves play a critical role in the transfer of information between learning agents and human teachers in active learning settings. We have shown that asymmetries exist between different types of interactions and their effects on both teaching and learning performance. In addition, we found that when human teachers receive feedback from a learning agent – thereby shifting from an active learning to an interactive learning setting – they exhibit preferences at the individual levels and these preferences shift during the teaching process.

Given our findings, we provide four recommendations to designers of machine learning systems that learn from human feedback. These findings have to do with how to (1) enable human teachers to communicate knowledge effectively, (2) provide information that helps human teachers determine what knowledge to share, (3) adapt algorithmic learner feedback to different human teachers, and (4) keep human teachers invested in actively engaging with algorithmic learners.

First, different interaction types enable human teachers to share different types of knowledge. Machine learning designers should, to the best of their ability, understand which interaction types will be most valuable given their particular task, model, and learning objectives. Alternatively, they can use an approach that dynamically adjusts for this during a learning session.

Second, model transparency enables human teachers to teach more effectively by helping them build a model of both what the learning agent knows and needs to know. This should therefore be highly prioritized when building systems designed to iteratively learn from people.

Third, in order to achieve this model transparency in an interactive setting, *binary* feedback works for both *learner-led* and *curriculum-led* individuals, and can therefore serve as a first-pass approach. However, being able to accurately identify a human teacher's current teaching persona and adapt feedback to it (e.g. providing *credit assignment* feedback to a *curriculum-led* teacher) may yield better performance outcomes. We therefore encourage designers of these systems to better understand the likely teaching dynamics present in their particular HiLL task.

Finally, we note that the willingness of a human teacher to be *learner-led* can vary over the course of repeated interactions with a learning agent and care should be taken to encourage this. Teachers who are somewhat *learner-led* are less likely to experience effects such as confirmation bias in their teaching, leading to greater teaching efficacy and a better teaching experience. Furthermore, navigating the balance between *curriculum-led* and *learner-led* teaching can help to frame learning as a joint task in which human teachers are equally capable of directing and driving the learning process. As we move towards a world where people are increasingly sharing knowledge with learning systems to accomplish goals (these tasks can range from the common task of receiving better movie recommendations to no-code application development!), it is important that people are equipped with the ability to best share this knowledge. It is our hope that these recommendations will help make this possible.

Ultimately, our findings suggests that the optimal interactive teaching process involves dynamically adjusting the interaction between teacher and learner to account for both the teacher's preferences and the learner's task knowledge. Such an undertaking is far-reaching and would have to build on work proposed in Sections 9.3.2 and 9.3.3, combine those findings with those from Chapter 7, and then modify an algorithm such as INQUIRE. Still, we can imagine that doing so would be worth the effort. An interactive learning system that can dynamically adapt to the teaching and learning of the people and algorithms involved would make it easier for lay persons to train, or fine-tune, their own models; as we discussed in Section 9.2, this has several real-world benefits. This thesis is one contribution towards the realization of such a system.

Bibliography

[1] How much care will you need? — acl administration for community living. URL `https://acl.gov/ltc/basic-needs/how-much-care-will-you-need`. 9.2.3

[2] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International conference on Machine learning*, page 1. ACM, 2004. 1, 3.1, 5.3, 7

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018. 3.3, 8.1

[4] M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2022. 9.2.2

[5] George Ainslie and Nick Haslam. Hyperbolic discounting. 1992. 9.2.2

[6] Ahmet Aker, Mahmoud El-Haj, M-Dyaa Albakour, Udo Kruschwitz, et al. Assessing crowdsourcing quality through objective tasks. In *International Conference on Language Resources and Evaluation*, pages 1456–1461, 2012. 3.2

[7] Kasun Amarasinghe, Kit T Rodolfa, Hemank Lamba, and Rayid Ghani. Explainable machine learning for public policy: Use cases, gaps, and research directions. *Data & Policy*, 5:e5, 2023. 3.3

[8] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4): 105–120, 2014. 3, 3.1, 7

[9] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 337–346, 2015. 3.3

[10] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009. 3.1

[11] Brenna D Argall, Eric L Sauser, and Aude G Billard. Tactile guidance for policy refinement and reuse. In *IEEE Intl. Conf. on Development and Learning*, pages 7–12, 2010. 3.1

[12] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma,

Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 3.1, 9.2.1

[13] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129, 1995. 3.1

[14] Andrea Bajcsy, Dylan Losey, Marcia O'Malley, and Anca Dragan. Learning robot objectives from physical human interaction. *Proceedings of Machine Learning Research*, 78: 217–226, 2017. 3.1, 5.3

[15] Andrea Bajcsy, Dylan P Losey, Marcia K O'Malley, and Anca D Dragan. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 141–149. ACM, 2018. 1, 3.1, 5.2, 7

[16] Chandrayee Basu, Mukesh Singhal, and Anca D Dragan. Learning from richer human guidance: Augmenting comparison-based learning with feature queries. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 132–140. ACM, 2018. 3.3, 8.1

[17] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. A generic coordinate descent framework for learning from implicit feedback. In *International Conference on World Wide Web*, 2017. 5.3

[18] Erdem Biyik, Malayandi Palan, Nicholas C. Landolfi, Dylan P. Losey, and Dorsa Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In *Conference on Robot Learning (CoRL)*, pages 1177–1190, 2020. 5.3, 5.3, 6.3, 6.3.2

[19] Erdem Bıyık, Malayandi Palan, Nicholas C Landolfi, Dylan P Losey, Dorsa Sadigh, et al. Asking easy questions: A user-friendly approach to active reward learning. In *Conference on Robot Learning*, pages 1177–1190, 2020. 3.1, 6.1

[20] Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ international conference on intelligent robots and systems*, pages 708–713. IEEE, 2005. 3.3, 5.3, 8.1

[21] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. corr abs/1606.01540 (2016). *arXiv preprint arXiv:1606.01540*, 2016. 6.3

[22] John Brooke. Sus: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, editors, *Usability Evaluation in Industry*. London: Taylor and Francis, 1996. 7.1.2

[23] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning (ICML)*, pages 783–792. PMLR, 2019. 3, 5.3

[24] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Lan-

guage models are few-shot learners. 33:1877–1901, 2020. 4.3

[25] Kalesha Bullard, Andrea L Thomaz, and Sonia Chernova. Towards intelligent arbitration of diverse active learning queries. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6049–6056. IEEE, 2018. 3.1, 7

[26] Kalesha Bullard, Yannick Schroecker, and Sonia Chernova. Active learning within constrained environments through imitation of an expert questioner. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019. 6.1

[27] Maya Cakmak and Andrea L Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012. 3.1

[28] Maya Cakmak, Crystal Chao, and Andrea L Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010. 3.3

[29] Thomas Cederborg, Ishaan Grover, Charles L Isbell Jr, and Andrea Lockerd Thomaz. Policy shaping with human teachers. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3366–3372, 2015. 5.3

[30] Carlos Celemin and Javier Ruiz-del Solar. An interactive framework for learning continuous actions policies based on corrective feedback. *Journal of Intelligent & Robotic Systems*, 95(1):77–97, 2019. 5.3

[31] Paul Chandler and John Sweller. Cognitive load theory and the format of instruction. *Cognition and instruction*, 8(4):293–332, 1991. 7

[32] Crystal Chao, Maya Cakmak, and Andrea L Thomaz. Transparent active learning for robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 317–324. IEEE, 2010. 3.3, 3.3

[33] Sonia Chernova and Andrea L Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014. 3.1

[34] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Conf. on Neural Information Processing Systems*, volume 30, pages 4299–4307, 2017. URL `https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf`. 1.1, 3.1, 5.3, 9.2.1

[35] Corinna Cortes, Lawrence D Jackel, and Wan-Ping Chiang. Limits on learning machine accuracy imposed by data quality. *Conference on Neural Information Processing Systems*, 7:239–246, 1994. 4.3

[36] Creative Commons. Attribution 2.0 generic (cc by 2.0), 2004. URL `"https://creativecommons.org/licenses/by/2.0/"`. [Online; accessed 11-May-2021]. 1

[37] Yuchen Cui and Scott Niekum. Active reward learning from critiques. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6907–6914. IEEE, 2018. 1, 3.1, 5.3, 7

[38] Yuchen Cui, Qiping Zhang, Alessandro Allievi, Peter Stone, Scott Niekum, and W Bradley Knox. The empathic framework for task learning from implicit human feedback. In *Conference on Robot Learning (CoRL)*, 2020. 5.3

[39] Christian Daniel, Malte Viering, Jan Metz, Oliver Kroemer, and Jan Peters. Active reward learning. In *Robotics: Science and Systems*, 2014. 5.3

[40] Christian Daniel, Oliver Kroemer, Malte Viering, Jan Metz, and Jan Peters. Active reward learning with a novel acquisition function. *Autonomous Robots*, 39(3):389–405, 2015. 1, 3.1, 7

[41] Krista E DeLeeuw and Richard E Mayer. A comparison of three measures of cognitive load: Evidence for separable measures of intrinsic, extraneous, and germane load. *Journal of educational psychology*, 100(1):223, 2008. 7.1.2, 7.3

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3.1, 4.3, 7.1

[43] Ana Durrani. The average american spends over 13 hours a day using digital media-here's what they're streaming. *Forbes*, 2023. 1

[44] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, 4(2): 81–173, 2011. 9.2.2

[45] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 3.1, 7.1

[46] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *International Conference on Intelligent User Interfaces*, pages 39–45, 2003. 3

[47] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning (ICML)*, pages 49–58, 2016. 5.3

[48] Jaime Fisac, Monica Gates, Jessica Hamrick, Chang Liu, Dylan Hadfield-Menell, Malayandi Palaniappan, Dhruv Malik, Shankar Sastry, Thomas Griffiths, and Anca Dragan. Pragmatic-pedagogic value alignment. In *International Symposium on Robotics Research*, pages 49–57. 2020. 2.1

[49] Tesca Fitzgerald, Ashok Goel, and Andrea Thomaz. Human-guided object mapping for task transfer. *ACM Transactions on Human-Robot Interaction*, 7(2):1–24, 2018. 3.1

[50] Tesca Fitzgerald, Elaine Short, Ashok Goel, and Andrea Thomaz. Human-guided trajectory adaptation for tool transfer. In *Intl. Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1350–1358, 2019. 3.1, 5.3

[51] Tesca Fitzgerald, Pallavi Koppol, Patrick Callaghan, Russell Quinlan Jun Hei Wong, Reid Simmons, Oliver Kroemer, and Henny Admoni. Inquire: Interactive querying for user-aware informative reasoning. In *6th Annual Conference on Robot Learning*, 2022. 6.1.2, 6.1.3, 6.1.3, 6.2, 6.3

[52] Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning and Ranking by Pairwise Comparison*, pages 65–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-14125-6. doi: 10.1007/978-3-642-14125-6_4. URL `https://doi.org/10.1007/978-3-642-14125-6_4`. 3.1

[53] Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: the dtw package. *Journal of statistical Software*, 31:1–24, 2009. 6.3

[54] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975. 8.1.2

[55] Jack M Guralnik, Lisa Alecxih, Laurence G Branch, and Joshua M Wiener. Medical and long-term care costs when older persons become more dependent. *American Journal of Public Health*, 92(8):1244–1245, 2002. 9.2.3

[56] Reymundo A Gutierrez, Vivian Chu, Andrea L Thomaz, and Scott Niekum. Incremental task modification via corrective demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1126–1133, 2018. 3.1

[57] Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The off-switch game. *arXiv preprint arXiv:1611.08219*, 2016. 3.1

[58] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. 4.3

[59] Ronny Hänsch and Olaf Hellwich. The truth about ground truth: Label noise in human-generated reference data. In *International Geoscience and Remote Sensing Symposium*, pages 5594–5597, 2019. 4.3

[60] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988. 3.2, 7.1.2

[61] Erin Hedlund, Michael Johnson, and Matthew Gombolay. The effects of a robot's performance on human teachers for learning from demonstration tasks. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 207–215, 2021. 3.3

[62] Rachel Holladay, Shervin Javdani, Anca Dragan, and Siddhartha Srinivasa. Active comparison based learning incorporating user uncertainty and noise. In *Workshop on Model Learning for Human-Robot Communication*, 2016. 5.3

[63] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, 2009. 3.2

[64] Sheikh Rabiul Islam, William Eberle, Sheikh Khaled Ghafoor, and Mohiuddin Ahmed. Explainable artificial intelligence approaches: A survey. *arXiv preprint arXiv:2101.09429*, 2021. 3.3

[65] Ashesh Jain, Shikhar Sharma, Thorsten Joachims, and Ashutosh Saxena. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research*, 34(10):1296–1313, 2015. 3.1

[66] Hong Jun Jeon, Smitha Milli, and Anca D Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. *arXiv preprint arXiv:2002.04833*, 2020. 3.1, 3.1,

7, 8.1

[67] Hong Jun Jeon, Smitha Milli, and Anca D Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Conference on Neural Information Processing Systems*, 2020. 5.3, 5.3, 5.3

[68] Hueyching Janice Jih and Thomas Charles Reeves. Mental models: A research focus for interactive learning systems. *Educational Technology Research and Development*, 40(3): 39–53, 1992. 3.3

[69] Elias Kalapanidas, Nikolaos Avouris, Marian Craciun, and Daniel Neagu. Machine learning algorithms: a study on noise sensitivity. In *Balcan Conference in Informatics*, pages 356–365, 2003. 4.3

[70] Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 877–882, 2007. 5.3

[71] Roshni Kaushik and Reid Simmons. Affective robot behavior improves learning in a sorting game. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 436–441. IEEE, 2022. 8.2

[72] David Kent, Carl Saldanha, and Sonia Chernova. A comparison of remote robot teleoperation interfaces for general object manipulation. In *International Conference on Human-Robot Interaction (HRI)*, pages 371–379, 2017. 3.2

[73] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318, 2013. 3.1

[74] W Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE international conference on development and learning*, pages 292–297. IEEE, 2008. 1.1, 3.1, 9.2.1

[75] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Intl. Conf. on Knowledge Capture*, pages 9–16, 2009. 5.3

[76] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. 3.1

[77] Pallavi Koppol, Henny Admoni, and Reid Simmons. Interaction considerations in learning from humans. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 5.3, 6.1, 6.4

[78] Samantha Krening and Karen M Feigh. Interaction algorithm effect on human experience with reinforcement learning. *ACM Transactions on Human-Robot Interaction (THRI)*, 7 (2):1–22, 2018. 3.3

[79] Samantha Krening, Brent Harrison, Karen M Feigh, Charles Lee Isbell, Mark Riedl, and Andrea Thomaz. Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55, 2016. 3.1

[80] Matthew Lease. On quality control and machine learning in crowdsourcing. *Human*

Computation, 11(11), 2011. 3.1, 3.2, 4.3

[81] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert systems with applications*, 41(4):2065–2073, 2014. 9.2.2

[82] Christopher Lin, Mausam Mausam, and Daniel Weld. Active learning with unbalanced classes and example-generation queries. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 6, 2018. 3.2

[83] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3.1

[84] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Rankiqa: Learning from rankings for no-reference image quality assessment. In *International Conference on Computer Vision (ICCV)*, pages 1040–1049, 2017. 5.3

[85] Luca Longo. Experienced mental workload, perception of usability, their interaction and impact on task performance. *PloS one*, 13(8):e0199661, 2018. 3.1, 3.2

[86] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, pages 73–105, 2011. 9.2.2

[87] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*, pages 2285–2294. PMLR, 2017. 3.3

[88] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning (CoRL)*, pages 879–893, 2018. 3.2, 3.2

[89] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020. 9.2.1

[90] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3): 276–282, 2012. 8.4.2

[91] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. A survey of explainable reinforcement learning. *arXiv preprint arXiv:2202.08434*, 2022. 3.3

[92] Matthew B Miles and A Michael Huberman. *Qualitative data analysis: An expanded sourcebook*. sage, 1994. 8.4.2

[93] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4.3

[94] Anis Najar and Mohamed Chetouani. Reinforcement learning with human advice: a survey. In *Frontiers in Robotics and AI*, 2021. 3.1, 3.1

[95] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In

ICML, volume 1, page 2, 2000. 3.1

[96] Felicia Ng, Jina Suh, and Gonzalo Ramos. Understanding and supporting knowledge decomposition for machine teaching. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 1183–1194, 2020. 3.3

[97] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. Learning grounded finite-state representations from unstructured demonstrations. *International Journal of Robotics Research*, 34(2):131–157, 2015. 5.3

[98] Jakob Nielsen and Jonathan Levy. Measuring usability: Preference vs. performance. *Communications of the ACM*, 37(4):66–75, April 1994. ISSN 0001-0782. doi: 10.1145/175276.175282. URL `https://doi.org/10.1145/175276.175282`. 3.2

[99] Donald A Norman. Cognitive engineering. *User centered system design*, 31(61):2, 1986. 8.4.3

[100] Donald A Norman. *The psychology of everyday things.* Basic books, 1988. 4.5

[101] Kieran O'Connor and Amar Cheema. Do evaluations rise with experience? *Psychological Science*, 29(5):779–790, 2018. 3.1

[102] Fred G Paas. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of educational psychology*, 84(4):429, 1992. 7.1.2

[103] Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences. In *Robotics: Science and Systems (RSS)*, 2019. 3.1, 6.1, 6.3.2, 6.4, 7

[104] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer, 2007. 9.2.2

[105] 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. *Computers in Human Behavior*, 26(5):840–856, 2010. 3.1, 3.2

[106] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007. 5.3

[107] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human–Computer Interaction*, 35(5-6):413–451, 2020. 3

[108] Sarunas J Raudys, Anil K Jain, et al. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991. 4.3

[109] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017. 1, 7

[110] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet

classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pages 5389–5400. PMLR, 2019. 4.3

[111] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 3.3, 8.1

[112] Avi Rosenfeld and Ariella Richardson. Explainability in human–agent systems. *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 33(6): 673–705, 2019. 4.1

[113] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019. 3.3

[114] Dorsa Sadigh, S Shankar Sastry, Sanjit A Seshia, and Anca Dragan. Information gathering actions over human internal state. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 66–73, 2016. 5.3

[115] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems (RSS)*, 2017. 1, 3.1, 7

[116] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999. 3.1

[117] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer, 2007. 9.2.2

[118] Aran Sena, Yuchen Zhao, and Matthew Jacob William Howard. Teaching human teachers to teach robot learners. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018. 3.3

[119] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *ACM Intl. Conf. on Knowledge Discovery and Data Mining*, pages 614–622, 2008. 4.3

[120] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arxiv:2305.17493*, 2023. 9.2.1

[121] Patrice Y Simard, Saleema Amershi, David M Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, et al. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742*, 2017. 3

[122] Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008. 4.3

[123] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan

Schultz, and Michael Goodrich. Common metrics for human-robot interaction. In *ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 33–40, 2006. 1.1

[124] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020. 3.1, 9.2.1

[125] Adarsh Subbaswamy and Suchi Saria. From development to deployment: dataset shift, causality, and shift-stable models in health ai. *Biostatistics*, 21(2):345–352, 2020. 4.3

[126] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *International Conference on Computer Vision (ICCV)*, pages 843–852, 2017. 4.3

[127] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 3.3

[128] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285, 1988. 2.3, 3.1, 3.2, 7, 8.3.1

[129] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 7.1

[130] Andrea L Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, 2008. 3.3, 8

[131] Andrea L Thomaz, Cynthia Breazeal, et al. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI Conference on Artificial Intelligence*, volume 6, pages 1000–1005, 2006. 5.3

[132] Aditya Vashistha, Pooja Sethi, and Richard Anderson. Bspeak: An accessible crowdsourcing marketplace for low-income blind people. In *CHI Conference on Human Factors in Computing Systems*, 2018. 3.2

[133] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018. 5.3

[134] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 9.2.1

[135] Joseph Jay Williams, Tania Lombrozo, and Bob Rehder. Why does explaining help learning? insight from an explanation impairment effect. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010. 3.3

[136] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning*

Research, 18(1):4945–4990, 2017. 3.1

[137] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, pages 7472–7482. PMLR, 2019. 4.1

[138] Ruohan Zhang, Faraz Torabi, Lin Guan, Dana H Ballard, and Peter Stone. Leveraging human guidance for deep reinforcement learning tasks. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019. 3.1

[139] Ruohan Zhang, Akanksha Saran, Bo Liu, Yifeng Zhu, Sihang Guo, Scott Niekum, Dana Ballard, and Mary Hayhoe. Human gaze assisted artificial intelligence: A review. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2020, page 4951. NIH Public Access, 2020. 5.3

[140] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[141] 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 preprint arXiv:1909.08593*, 2019. 1.1, 3.1, 9.2.1

[142] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2810–2818, 2019. 9.2.2