Learning Reward Functions from Diverse Sources of Human Feedback: Optimally Integrating Demonstrations and Preferences

The International Journal of Robotics Research
XX(X):1–20
©The Author(s) 2020
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
ijr.sagepub.com/
SAGE

Erdem Bıyık[1], Dylan P. Losey[2], Malayandi Palan[2], Nicholas C. Landolfi[2], Gleb Shevchuk[2], and Dorsa Sadigh[1,2]

Abstract

Reward functions are a common way to specify the objective of a robot. As designing reward functions can be extremely challenging, a more promising approach is to directly learn reward functions from human teachers. Importantly, data from human teachers can be collected either *passively* or *actively* in a variety of forms: passive data sources include *demonstrations*, (e.g., kinesthetic guidance), whereas *preferences* (e.g., comparative rankings) are actively elicited. Prior research has *independently* applied reward learning to these different data sources. However, there exist many domains where multiple sources are complementary and expressive. Motivated by this general problem, we present a framework to *integrate* multiple sources of information, which are either passively or actively collected from human users. In particular, we present an algorithm that first utilizes user demonstrations to initialize a belief about the reward function, and then actively probes the user with preference queries to zero-in on their true reward. This algorithm not only enables us combine multiple data sources, but it also informs the robot *when* it should leverage each type of information. Further, our approach accounts for the human's *ability* to provide data: yielding user-friendly preference queries which are also theoretically optimal. Our extensive simulated experiments and user studies on a Fetch mobile manipulator demonstrate the superiority and the usability of our integrated framework.

Keywords

Reward Learning, Active Learning, Inverse Reinforcement Learning, Learning from Demonstrations, Preference-based Learning, Human-Robot Interaction

Introduction

When robots enter everyday human environments they need to understand how they should behave. Of course, humans know what the robot should be doing — one promising direction is therefore for robots to *learn* from human experts. Several recent deep learning works embrace this approach, and leverage human demonstrations to try and extrapolate the right robot behavior for interactive tasks.

In order to train a deep neural network, however, the robot needs access to a *large amount* of interaction data. This makes applying such techniques challenging within domains where large amounts of human data are not readily available. Consider domains like robot learning and human-robot interaction in general: here the robot must collect diverse and informative examples of how a human wants the robot to act and respond (Choudhury et al. 2019).

Imagine teaching an autonomous car to safely drive alongside human-driven cars. During training, you demonstrate how to merge in front of a few different vehicles. The autonomous car learns from these demonstrations, and tries to follow your examples as closely as possible. But when the car is deployed, it comes across an aggressive driver, who behaves differently than anything that the robot has seen before — so that matching your demonstrations unintentionally causes the autonomous car to have an accident!

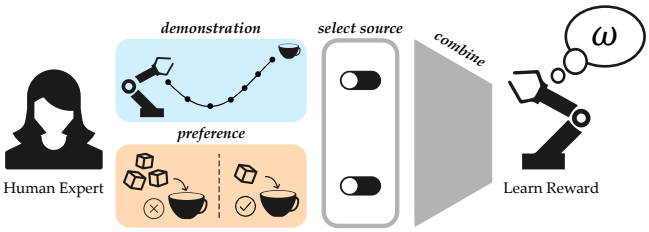

Figure 1. When robots learn from human teachers multiple sources of information are available. We *integrate* demonstrations and preferences, and determine *when* to elicit *what* type of data while accounting for the human-in-the-loop.

Here the autonomous car got it wrong because demonstrations alone failed to convey what you really wanted. Humans are unwilling — and in many cases unable — to provide demonstrations for every possible situation: but it is also difficult for robots to accurately generalize the knowledge learned from just a few expert demonstrations. Hence, the robot needs to leverage the time it has with the

[1]Department of Electrical Engineering, Stanford University
[2]Department of Computer Science, Stanford University

Corresponding author:
Erdem Bıyık
Email: ebiyik@stanford.edu

human teacher as intelligently as possible. Fortunately, there are many different *sources* of human data: passive sources include demonstrations (e.g., kinesthetic teaching) and narrations, whereas preferences (e.g., rankings or queries) and verbal cues can be collected actively. All of these data sources provide information about the human's true reward function, but do so in different ways. Referring to Figure 1, demonstrations provide rich and informative data about the style and timing of the robot's motions. On the other hand, preferences provide explicit and accurate information about a specific aspect of the task.

In this paper, we focus on leveraging these two sources, demonstrations and preferences, to learn what the user wants. Unlike prior work — where the robot *independently* learns from either source — we assert that robots should intelligently *integrate* these different sources when gathering data. Our insight is that:

Multiple data sources are complementary: demonstrations provide a high-level initialization of the human's overall reward functions, while preferences explore specific, fine-grained aspects of it.

We present a unified framework for gathering human data that is collected either passively or actively. Importantly, our approach determines *when* to utilize *what* type of information source, so that the robot can learn efficiently. We draw from work on active learning and inverse reinforcement learning to synthesize human data sources while maximizing information gain.

We emphasize that the robot is gathering data from a human, and thus the robot needs to account for the human's *ability* to provide that data. Returning to our driving example, imagine that an autonomous car wants to determine how quickly it should move while merging. There are a range of possible speeds from 20 to 30 mph, and the car has a uniform prior over this range. A naïve agent might ask a question to divide the potential speeds in half: would you rather merge at 24.9 or 25.1 mph? While another robot might be able to answer this question, to the human user these two choices seem indistinguishable — rendering this question practically useless! Our approach ensures that the robot maintains a model of the human's *ability to provide* data together with the *potential value* of that data. This not only ensures that the robot learns as much as possible from each data source, but it also makes it *easier* for the human to provide their data.

In this paper, we make the following contributions*:

Determining When to Leverage Which Data Source. We explore both passively collected demonstrations and actively collected preference queries. We prove that intelligent robots should collect passive data sources *before* actively probing the human in order to maximize information gain. Moreover, when working with a human, each additional data point has an associated cost (e.g., the time required to provide that data). We therefore introduce an optimal stopping condition so that the robot *stops* gathering data from the human when its expected utility outweighs its cost.

Integrating Multiple Data Sources. We propose a unified framework for reward learning, *DemPref*, that leverages demonstrations and preferences to learn a personalized human reward function. We empirically show that combining both sources of human data leads to a better understanding of what the human wants than relying on either source alone. Under our proposed approach the human demonstrations initialize a high-level belief over what the right behavior is, and the preference queries iteratively fine-tune that belief to minimize robot uncertainty.

Accounting for the Human. When the robot tries to proactively gather data, it must account for the human's ability to provide accurate information. We show that today's state-of-the-art volume removal approach for generating preference queries does not produce easy or intuitive questions. We therefore propose an information theoretic alternative, which maximizes the utility of questions while readily minimizing the human's uncertainty over their answer. This approach naturally results in *user-friendly* questions, and has the same computational complexity as the state-of-the-art volume removal method.

Conducting Simulations and User Studies. We test our proposed approach across multiple simulated environments, and perform two user studies on a 7-DoF Fetch robot arm. We experimentally compare our DemPref algorithm to (a) learning methods that only ever use a single source of data, (b) active learning techniques that ask questions without accounting for the human in-the-loop, and (c) algorithms that leverage multiple data sources but in different orders. We ultimately show that human end-users subjectively prefer our approach, and that DemPref objectively learns reward functions more efficiently than the alternatives.

Overall, this work demonstrates how robots can efficiently learn from humans by synthesizing multiple sources of data. We believe each of these data sources has a role to play in settings where access to data is limited, such as when learning from humans for interactive robotics tasks.

Related Work

Prior work has extensively studied learning reward functions using a single source of information, e.g., ordinal data (Chu and Ghahramani 2005), or human corrections (Bajcsy et al. 2017, 2018; Li et al. 2021b). Other works attempted to incorporate expert assessments of trajectories (Shah and Shah 2020). More related to our work, we will focus on learning from demonstrations and learning from rankings. There has also been a few studies that investigate combining multiple sources of information. Below, we summarize these related works.

Learning reward functions from demonstrations. A large body of work is focused on learning reward functions using a single source of information: collected expert demonstrations. This approach is commonly referred to as inverse reinforcement learning (IRL), where the demonstrations are assumed to be provided by a human expert who is approximately optimizing the reward function (Abbeel and Ng 2004, 2005; Ng et al. 2000; Ramachandran and Amir 2007; Ziebart et al. 2008; Nikolaidis et al. 2015).

*Note that parts of this work have been published at Robotics: Science and Systems (Palan et al. 2019) and the Conference on Robot Learning (Biyik et al. 2019b).

IRL has been successfully applied in a variety of domains. However, it is often too difficult to manually operate robots, especially manipulators with high degrees of freedom (DoF) (Akgun et al. 2012; Dragan and Srinivasa 2012; Javdani et al. 2015; Khurshid and Kuchenbecker 2015). Moreover, even when operating the high DoF of a robot is not an issue, people might have cognitive biases or habits that cause their demonstrations to not align with their actual reward functions. For example, Kwon et al. (2020) have shown that people tend to perform consistently risk-averse or risk-seeking actions in risky situations, depending on their potential losses or gains, even if those actions are suboptimal. As another example from the field of autonomous driving, Basu et al. (2017) suggest that people prefer their autonomous vehicles to be more timid compared to their own demonstrations. These problems show that, even though demonstrations carry an important amount of information about what the humans want, one should either try to learn from suboptimal demonstrations (Brown et al. 2020; Chen et al. 2020) or go beyond demonstrations to properly capture the underlying reward functions.

Learning reward functions from rankings. Another helpful source of information that can be used to learn reward functions is rankings, i.e., when a human expert ranks a set of trajectories in the order of their preference (Brown et al. 2019; Biyik et al. 2019a). A special case of this, which we also adopt in our experiments, is when these rankings are pairwise (Akrour et al. 2012; Lepird et al. 2015; Christiano et al. 2017; Ibarz et al. 2018; Brown and Niekum 2019). In addition to the simulation environments, several works have leveraged pairwise comparison questions for various domains, including exoskeleton gait optimization (Tucker et al. 2020; Li et al. 2021a), autonomous driving (Katz et al. 2021), and trajectory optimization for robots in interactive settings (Cakmak et al. 2011; Biyik et al. 2020).

While having humans provide pairwise comparisons does not suffer from similar problems to collecting demonstrations, each comparison question is much less informative than a demonstration, since comparison queries can provide at most 1 bit of information. Prior works have attempted to tackle this problem by actively generating the comparison questions (Sadigh et al. 2017; Biyik and Sadigh 2018; Basu et al. 2019; Katz et al. 2019; Wilde et al. 2019). While they were able to achieve significant gains in terms of the required number of comparisons, we hypothesize that one can attain even better data-efficiency by leveraging multiple sources of information, even when some sources might not completely align with the true reward functions, e.g., demonstrations as in the driving work by Basu et al. (2017). In addition, these prior works did not account for the human in-the-loop and employed acquisition functions that produce very difficult questions for active question generation. In this work, we propose an alternative approach that generates easy comparison questions for the human while also maximizing the information gained from each question.

Learning reward functions from both demonstrations and preferences. Ibarz et al. (2018) have explored combining demonstrations and preferences, where they take a model-free approach to learn a reward function in the Atari domain. Our motivation, physical autonomous systems, differs from theirs, leading us to a structurally different method. It is difficult and expensive to obtain data from humans controlling physical robots. Hence, model-free approaches are presently impractical. In contrast, we give special attention to data-efficiency. To this end, we (1) assume a simple reward structure that is standard in the IRL literature, and (2) employ active learning methods to generate comparison questions while simultaneously taking into account the ease of the questions. As the resulting training process is not especially time-intensive, we efficiently learn personalized reward functions.

Figure 2. Example of a demonstration (top) and preference query (bottom). During the demonstration the robot is *passive*, and the human teleoperates the robot to produce trajectory ξ^D from scratch. By contrast, the preference query is *active*: the robot chooses two trajectories ξ_1 and ξ_2 to show to the human, and the human answers by selecting their preferred option.

Problem Formulation

Building on prior work, we integrate multiple sources of data to learn the human's reward function. Here we formalize this problem setting, and introduce two forms of human feedback that we will focus on: demonstrations and preferences.

MDP. Let us consider a fully observable dynamical system describing the evolution of the robot, which should ideally behave according to the human's preferences. We formulate this system as a discrete-time Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, f, r, T \rangle$. At time t, $s_t \in \mathcal{S}$ denotes the state of the system and $a_t \in \mathcal{A}$ denotes the robot's action. The robot transitions deterministically to a new state according to its dynamics: $s_{t+1} = f(s_t, a_t)$. At every timestep the robot receives reward $r(s)$, and the task ends after a total of T timesteps.

Trajectory. A trajectory $\xi \in \Xi$ is a finite sequence of state-action pairs, i.e., $\xi = \big((s_t, a_t)\big)_{t=0}^{T}$ over time horizon T. Because the system is deterministic, the trajectory ξ can be more succinctly represented by $\Lambda = (s_0, a_0, a_1, \ldots, a_T)$, the

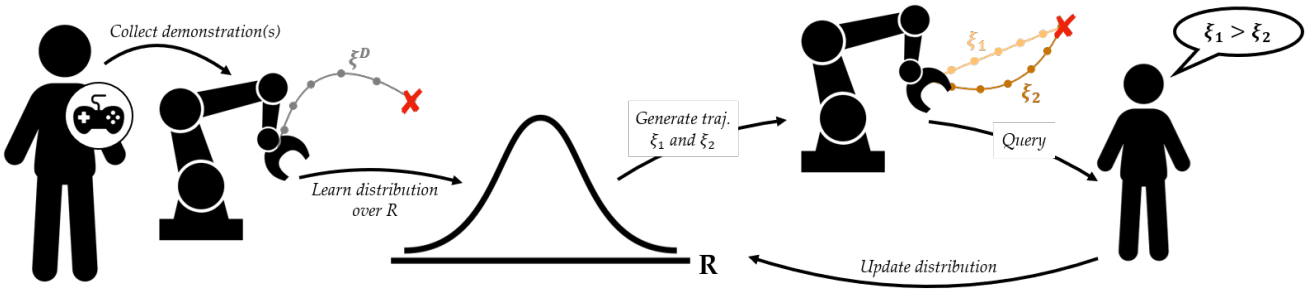

Figure 3. Overview of our DemPref approach. The human starts by providing a set of *high-level* demonstrations (left), which are used to initialize the robot's belief over the human's reward function. The robot then *fine-tunes* this belief by asking questions (right): the robot actively generates a set of trajectories, and asks the human to indicate their favorite.

initial state and sequence of actions in the trajectory. We use ξ and Λ interchangeably when referring to a trajectory, depending on what is most appropriate for the context.

Reward. The reward function captures how the human wants the robot to behave. Similar to related works (Abbeel and Ng 2004; Ng et al. 2000; Ziebart et al. 2008), we assume that the reward is a linear combination of features weighted by ω, so that $r(s) = \omega \cdot \phi(s)$. The reward of a trajectory ξ is based on the cumulative feature counts along that trajectory[†]:

$$R(\xi) = \omega \cdot \sum_{s \in \xi} \phi(s) = \omega \cdot \Phi(\xi) \tag{1}$$

Consistent with prior work (Bobu et al. 2018; Bajcsy et al. 2017; Ziebart et al. 2008), we will assume that the trajectory features $\Phi(\cdot)$ are given: accordingly, to understand what the human wants, the robot must simply learn the human's weights ω. We normalize the weights so that $\|\omega\|_2 \leq 1$.

Demonstrations. One way that the human can convey their reward weights to the robot is by providing demonstrations. Each human demonstration is a trajectory ξ^D, and we denote a data set of L human demonstrations as $\mathcal{D} = \{\xi_1^D, \xi_2^D, \ldots, \xi_L^D\}$. In practice, these demonstrations could be provided by kinesthetic teaching, by teleoperating the robot, or in virtual reality (see Figure 2, top).

Preferences. Another way the human can provide information is by giving feedback about the trajectories the robot shows. We define a preference query $Q = \{\xi_1, \xi_2, \ldots, \xi_K\}$ as a set of K robot trajectories. The human answers this query by picking a trajectory $q \in Q$ that matches their personal preferences (i.e., maximizes their reward function). In practice, the robot could play K different trajectories, and let the human indicate their favorite (see Figure 2, bottom).

Problem. Our overall goal is to accurately and efficiently learn the human's reward function from multiple sources of data. In this paper, we will only focus on demonstrations and preferences. Specifically, we will optimize when to query a user to provide demonstrations, and when to query a user to provide preferences. Our approach should learn the reward parameters ω with the smallest combination of demonstrations and preference queries. One key part of this process is accounting for the human-in-the-loop: we not only consider the informativeness of the queries for the robot, but we also ensure that the queries are intuitive and can easily be answered by the human user.

DemPref: Learning Rewards from Demonstrations and Preferences

In this section we overview our approach for integrating demonstrations and preferences to efficiently learn the human's reward function. Intuitively, demonstrations provide an informative, *high-level* understanding of what behavior the human wants; however, these demonstrations are often noisy, and may fail to cover some aspects of the reward function. By contrast, preferences are *fine-grained*: they isolate specific, ambiguous aspects of the human's reward, and reduce the robot's uncertainty over these regions. It therefore makes sense for the robot to start with high-level demonstrations before moving to fine-grained preferences. Indeed — as we will show in Theorem 2 — starting with demonstrations and then shifting to preferences is the most efficient order for gathering data. Our DemPref algorithm leverages this insight to combine high-level demonstrations and low-level preference queries (see Figure 3).

Initializing a Belief from Offline Demonstrations

DemPref starts with a set of offline trajectory demonstrations $\mathcal{D}$. These demonstrations are collected *passively*: the robot lets the human show their desired behavior, and does not interfere or probe the user. We leverage these passive human demonstrations to initialize an informative but imprecise prior over the true reward weights ω.

Belief. Let the belief b be a probability distribution over ω. We initialize b using the trajectory demonstrations, so that $b^0(\omega) = P(\omega \mid \mathcal{D})$. Applying Bayes' Theorem:

$$\begin{aligned} b^0(\omega) &\propto P(\mathcal{D} \mid \omega) P(\omega) \\ &\propto P(\xi_1^D, \xi_2^D, \ldots, \xi_L^D \mid \omega) P(\omega) \end{aligned} \tag{2}$$

We assume that the trajectory demonstrations are conditionally independent, i.e., the human does not consider their previous demonstrations when providing a new demonstration. Hence, Equation (2) becomes:

$$b^0(\omega) \propto P(\omega) \prod_{\xi^D \in \mathcal{D}} P(\xi^D \mid \omega) \tag{3}$$

In order to evaluate Equation (3), we need a model of $P(\xi^D \mid \omega)$ — in other words, how likely is the demonstrated trajectory ξ^D given that the human's reward weights are ω?

[†]More generally, the trajectory features $\Phi(\xi)$ can be defined as any function over the entire trajectory ξ.

Boltzmann Rational Model. DemPref is not tied to any specific choice of the human model in Equation (3), but we do want to highlight the Boltzmann rational model that is commonly used in inverse reinforcement learning (Ziebart et al. 2008; Ramachandran and Amir 2007). Under this particular model, the probability of a human demonstration is related to the reward associated with that trajectory:

$$\begin{aligned} P(\xi^D \mid \omega) &\propto \exp\left(\beta^D R(\xi^D)\right) \\ &= \exp\left(\beta^D \omega \cdot \Phi(\xi^D)\right) \end{aligned} \tag{4}$$

Here $\beta^D \geq 0$ is a temperature hyperparameter, commonly referred to as the *rationality coefficient*, that expresses how noisy the human demonstrations are, and we substituted Equation (1) for R. Leveraging this human model, the initial belief over ω given the offline demonstrations becomes:

$$b^0(\omega) \propto \exp\left(\beta^D \omega \cdot \sum_{\xi^D \in \mathcal{D}} \Phi(\xi^D)\right) P(\omega) \tag{5}$$

Summary. Human demonstrations provide an informative but imprecise understanding of ω. Because these demonstrations are collected passively, the robot does not have an opportunity to investigate aspects of ω that it is unsure about. We therefore leveraged these demonstrations to initialize b^0, which we treat as a high-level *prior* over the human's reward. Next, we will introduce proactive questions to remove uncertainty and obtain a fine-grained posterior.

Updating the Belief with Proactive Queries

After initialization, DemPref iteratively performs two main tasks: *actively* choosing the right preference query Q to ask, and applying the human's answer to update b. In this section we focus on the second task: updating the robot's belief b. We will explore how robots should proactively choose the right question in the subsequent section.

Posterior. The robot asks a new question at each iteration $i \in \{0, \ldots\}$. Let Q_i denote the i-th preference query, and let q_i be the human's response to this query. Again applying Bayes' Theorem, the robot's posterior over ω becomes:

$$b^{i+1}(\omega) \propto P(q_0, \ldots, q_i \mid Q_0, \ldots, Q_i, \omega) \cdot b^0(\omega), \tag{6}$$

where b^0 is the prior initialized using human demonstrations. We assume that the human's responses q are conditionally independent, i.e., only based on the current preference query and reward weights. Equation (6) then simplifies to:

$$b^{i+1}(\omega) \propto \prod_{j=0}^{i} P(q_j \mid Q_j, \omega) \cdot b^0(\omega) \tag{7}$$

We can equivalently write the robot's posterior over ω after asking $i+1$ questions as:

$$b^{i+1}(\omega) \propto P(q_i \mid Q_i, \omega) \cdot b^i(\omega) \tag{8}$$

Human Model. In Equation (8), $P(q \mid Q, \omega)$ denotes the probability that a human with reward weights ω will answer query Q by selecting trajectory $q \in Q$. Put another way, this likelihood function is a probabilistic human model. Our DemPref approach is agnostic to the specific choice of $P(q \mid Q, \omega)$ — we test different human models in our experiments. For now, we simply want to highlight that this human model defines the way users respond to queries.

Choosing Queries. We have covered how the robot can update its understanding of ω given the human's answers; but how does the robot choose the right questions in the first place? Unlike demonstrations — where the robot is passive — here the robot is *active*, and purposely probes the human to get fine-grained information about specific parts of ω that are unclear. At the same time, the robot needs to remember that a human is answering these questions, and so the options need to be easy and intuitive for the human to respond to. Proactively choosing intuitive queries is the most challenging part of the DemPref approach. Accordingly, we will explore methods for actively generating queries Q in the next section, before returning to finalize our DemPref algorithm.

Asking Easy Questions

Now that we have overviewed DemPref — which integrates passive human demonstrations and proactive preference queries — we will focus on how the robot chooses these queries. We take an *active learning* approach: the robot selects queries to accurately fine-tune its estimate of ω in a data-efficient manner, minimizing the total number of questions the human must answer.

Overview. We present two methods for active preference-based reward learning: *Volume Removal* and *Information Gain*. Volume removal is a state-of-the-art approach where the robot solves a submodular optimization problem to choose which questions to ask. However, this approach sometimes fails to generate *informative* queries, and also does not consider the ease and intuitiveness of every query for the human-in-the-loop. This can lead to queries that are *difficult* for the human to answer, e.g., two queries that are equally good (or bad) from the human's perspective.

We *resolve* this issue with the second method, information gain: here the robot balances (a) how much information it will get from a correct answer against (b) the human's ability to answer that question confidently. We end the section by describing a *set of tools* that can be used to enhance either method, including an optimal condition for determining when the robot should stop asking questions.

Greedy Robot. The robot should ask questions that provide accurate, fine-grained information about ω. Ideally, the robot will find the best *adaptive sequence* of queries to clarify the human's reward. Unfortunately, reasoning about an adaptive sequence of queries is — in general — NP-hard (Ailon 2012). We therefore proceed in a *greedy* fashion: at every iteration i, the robot chooses Q_i while thinking only about the next posterior b^{i+1} in Equation (8).

Choosing Queries with Volume Removal

Maximizing volume removal is a state-of-the-art strategy for selecting queries. The method attempts to generate the most-informative queries by finding the Q_i that maximizes the expected difference between the prior and *unnormalized* posterior (Sadigh et al. 2017; Biyik and Sadigh 2018; Biyik et al. 2019a). Formally, the method generates a query of

$K \geq 2$ trajectories at iteration i by solving:

$$\underset{Q_i=\{\Lambda_1,\ldots,\Lambda_K\}}{\arg\max} \mathbb{E}_{q_i}\left[\int_{\|\omega\|_2\leq 1} \left(b^i(\omega) - b^i(\omega)P(q_i\,|\,Q_i,\omega)\right) d\omega\right]$$

where the prior is on the left and the unnormalized posterior from Equation (8) is on the right. This optimization problem can equivalently be written as:

$$Q_i^* = \underset{Q_i=\{\Lambda_1,\ldots,\Lambda_K\}}{\arg\max} \mathbb{E}_{q_i}\mathbb{E}_{b^i}\left[1 - P(q_i\,|\,Q_i,\omega)\right] \quad (9)$$

The distribution b^i can get very complex and thus — to tractably compute the expectations in Equation (9) — we are forced to leverage sampling. Letting Ω denote a set of M samples drawn from the prior b^i, and $\doteq$ denote asymptotic equality as the number of samples $M \to \infty$, the optimization problem in Equation (9) becomes:

$$Q_i^* \doteq \underset{Q_i=\{\Lambda_1,\ldots,\Lambda_K\}}{\arg\min} \sum_{q_i\in Q_i}\left(\sum_{\omega\in\Omega} P(q_i \mid Q_i,\omega)\right)^2 \quad (10)$$

Intuition. When solving Equation (10), the robot looks for queries Q_i where each answer $q_i \in Q_i$ is equally likely given the current belief over ω. These questions appear useful because the robot is maximally uncertain about which trajectory the human will prefer.

When Does This Fail? Although prior works have shown that volume removal can work in practice, we here identify two key shortcomings. First, we point out a failure case: the robot may solve for questions where the answers are equally likely but *uninformative* about the human's reward. Second, the robot does not consider the human's ability to answer when choosing questions — and this leads to *challenging*, indistinguishable queries that are hard to answer!

Uninformative Queries. The optimization problem used to identify volume removal queries fails to capture our original goal of generating informative queries. Consider a trivial query where all options are identical: $Q_i = \{\xi_A, \xi_A, \ldots, \xi_A\}$. Regardless of which answer q the human chooses, here the robot gets no information about the right reward function; put another way, $b^{i+1} = b^i$. Asking a trivial query is a waste of the human's time — but we find that this uninformative question is actually a best-case solution to Equation (9).

Theorem 1. *The trivial query $Q = \{\xi_A, \xi_A, \ldots, \xi_A\}$ (for any $\xi_A \in \Xi$) is a global solution to Equation (9).*

Proof. For a given Q and ω, $\sum_q P(q \mid Q,\omega) = 1$. Thus, we can upper bound the objective in Equation (9) as follows:

$$\mathbb{E}_{q_i}\mathbb{E}_{b^i}[1 - P(q_i \mid Q_i,\omega)]$$
$$= 1 - \mathbb{E}_{b^i}\left[\sum_{q_i\in Q_i} P(q_i \mid Q_i,\omega)^2\right] \leq 1 - 1/K$$

recalling that K is the total number of options in Q_i. For the trivial query $Q = \{\xi_A, \xi_A, \ldots, \xi_A\}$, the objective in Equation (9) has value $\mathbb{E}_{q_n}\mathbb{E}_{b^i}[1 - P(q_i \mid Q,\omega)] = 1 - 1/K$. This is equal to the upper bound on the objective, and thus the trivial, uninformative query of identical options is a global solution to Equation (9). □

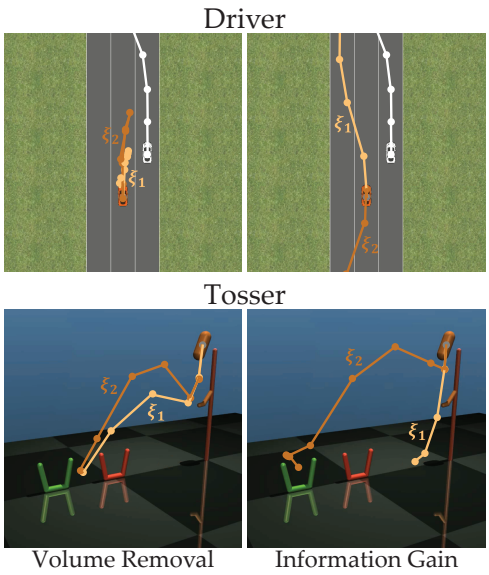

Figure 4. Sample queries generated with the volume removal and information gain methods on Driver and Tosser tasks. Volume removal generates queries that are difficult, because the options are almost equally good or equally bad.

Challenging Queries. Volume removal prioritizes questions where each answer is equally likely. Even when the options are not identical (as in a trivial query), the questions may still be very challenging for the user to answer. We explain this issue through a concrete example (also see Fig. 4):

Example 1. Let the robot query the human while providing $K = 2$ different answer options, ξ_1 and ξ_2.

Question A. Here the robot asks a question where both options are equally good choices. Consider query Q_A such that $P(q = \xi_1 \mid Q_A,\omega) = P(q = \xi_2 \mid Q_A,\omega)\ \forall\omega \in \Omega$. Responding to Q_A is difficult for the human, since both options ξ_1 and ξ_2 equally capture their reward function.

Question B. Alternatively, this robot asks a question where only one option matches the human's true reward. Consider a query Q_B such that:

$$P(q = \xi_1 \mid Q_B,\omega) \approx 1 \quad \forall\omega \in \Omega^{(1)}$$
$$P(q = \xi_2 \mid Q_B,\omega) \approx 1 \quad \forall\omega \in \Omega^{(2)}$$
$$\Omega^{(1)} \cup \Omega^{(2)} = \Omega, \quad |\Omega^{(1)}| = |\Omega^{(2)}|$$

If the human's weights ω lie in $\Omega^{(1)}$, the human will always answer with ξ_1, and — conversely — if the true ω lies in $\Omega^{(2)}$, the human will always select ξ_2. Intuitively, this query is easy for the human: regardless of what they want, one option stands out when answering the question.

Incorporating the Human. Looking at Example 1, it seems clear that the robot should ask question Q_B. Not only does Q_A fail to provide any information about the human's reward (because their response could be equally well explained by any ω), but it is also hard for the human to answer (since both options seem equally viable). Unfortunately, when maximizing volume removal the robot thinks Q_A is *just as good* as Q_B: they are both global solutions to its optimization problem! Here volume removal gets it wrong because it fails to take the human into consideration. Asking questions based only on how uncertain the robot is about the human's answer

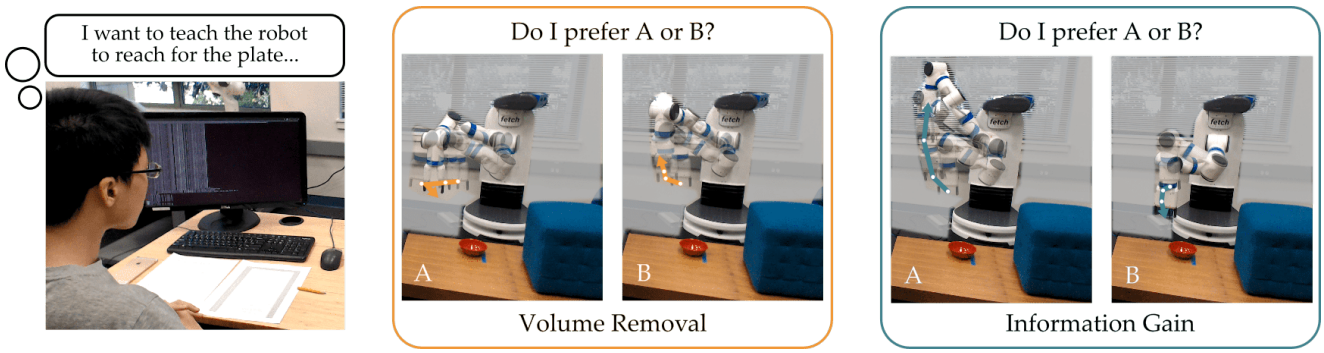

Figure 5. Comparing preference queries that do not account for the human's ability to answer to queries generated using our information gain approach. Here the robot is attempting to learn the user's reward function, and demonstrates two possible trajectories. The user should select the trajectory that better aligns with their own preferences. While the trajectories produced by the state-of-the-art volume removal method are almost indistinguishable, our information theoretic approach results in questions that are easy to answer, which eventually increase the robot's overall learning efficiency.

can naturally lead to confusing, uninformative queries. Fig. 4 demonstrates some of these hard queries generated by the volume removal formulation.

Choosing Queries with Information Gain

To address the shortcomings of volume removal, we propose an information gain approach for choosing DemPref queries. We demonstrate that this approach ensures the robot not only asks questions which provide the most information about ω, but these questions are also *easy* for the human to answer. We emphasize that encouraging easy questions is not a heuristic addition: instead, the robot recognizes that picking queries which the human can accurately answer is necessary for overall performance. Accounting for the human-in-the-loop is therefore part of the optimal information gain solution.

Information Gain. At each iteration, we find the query Q_i that maximizes the expected information gain about ω. We do so by solving the following optimization problem:

$$\begin{aligned} Q_i^* &= \arg\max_{Q_i} I(\omega; q_i \mid Q_i, b^i) \\ &= \arg\max_{Q_i} H(\omega \mid Q_i, b^i) - \mathbb{E}_{q_i} H(\omega \mid q_i, Q_i, b^i), \end{aligned} \quad (11)$$

where I is the mutual information and H is Shannon's information entropy (Cover and Thomas 2012). Approximating the expectations via sampling, we re-write this optimization problem below (see the Appendix for the full derivation):

$$\begin{aligned} Q_i^* \doteq \arg\max_{Q_i=\{\Lambda_1,\dots,\Lambda_K\}} \frac{1}{M} \sum_{q_i \in Q_i} \sum_{\omega \in \Omega} \Bigg(P(q_i \mid Q_i, \omega) \\ \log_2 \left(\frac{M \cdot P(q_i \mid Q_i, \omega)}{\sum_{\omega' \in \Omega} P(q_i \mid Q_i, \omega')} \right) \Bigg) \end{aligned} \quad (12)$$

Intuition. To see why accounting for the human is naturally part of the information gain solution, re-write Equation (11):

$$Q_i^* = \arg\max_{Q_i} \; H(q_i \mid Q_i, b^i) - \mathbb{E}_\omega H(q_i \mid \omega, Q_i) \quad (13)$$

Here the first term in Equation (13) is the *robot's uncertainty* over the human's response: given a query Q_i and the robot's understanding of ω, how confidently can the robot predict the human's answer? The second entropy term captures the *human's uncertainty* when answering: given a query and their true reward, how confidently will they choose option q_i? Optimizing for information gain with Equations (11) or (12) naturally considers both robot and human uncertainty, and favors questions where (a) the robot is unsure how the human will answer but (b) the human can answer easily. We contrast this to volume removal, where the robot purely focused on questions where the human's answer was unpredictable.

Why Does This Work? To highlight the advantages of this method, let us revisit the shortcomings of volume removal. Below we show how information gain successfully addresses the problems described in Theorem 1 and Example 1. Further, we emphasize that the computational complexity of computing objective (12) is equivalent — in order — to the volume removal objective from Equation (10). Thus, information gain avoids the previous failures while being at least as computationally tractable.

Uninformative Queries. Recall from Theorem 1 that any trivial query $Q = \{\xi_A, \dots, \xi_A\}$ is a global solution for volume removal. In reality, we know that this query is a worst-case choice: no matter how the human answers, the robot will gain no insight into ω. Information gain ensures that the robot will not ask trivial queries: under Equation (11), $Q = \{\xi_A, \dots, \xi_A\}$ is actually the *global minimum*!

Challenging Questions. Revisiting Example 1, we remember that Q_B was a much easier question for the human to answer, but volume removal values Q_A as highly as Q_B. Under information gain, the *robot* is equally uncertain about how the human will answer Q_A and Q_B, and so the first term in Equation (13) is the same for both. But the information gain robot additionally recognizes that the *human* is very uncertain when answering Q_A: here Q_A attains the global maximum of the second term while Q_B attains the global minimum! Thus, the overall value of Q_B is higher and — as desired — the robot recognizes that Q_B is a better question.

Why Demonstrations First? Now that we have a user-friendly strategy for generating queries, it remains to determine *in what order* the robot should leverage the demonstrations and the preferences.

Recall that demonstrations provide coarse, high-level information, while preference queries hone-in on isolated aspects of the human's reward function. Intuitively, it seems

like we should start with high-level demonstrations before probing low-level preferences: but is this really the right order of collecting data? What about the alternative — a robot that instead waits to obtain demonstrations until after asking questions?

When leveraging information gain to generate queries, we here prove that the robot will gain *at least as much* information about the human's preferences as any other order of demonstrations and queries. Put another way, starting with demonstrations *in the worst case* is just as good as any other order; and *in the best case* we obtain more information.

Theorem 2. *Under the Boltzmann rational human model for demonstrations presented in Equation (4), our DemPref approach (Algorithm 1) — where preference queries are actively generated after collecting demonstrations — results in at least as much information about the human's preferences as would be obtained by reversing the order of queries and demonstrations.*

Proof. Let Q_i^* be the information gain query *after collecting demonstrations*. From Equation (11), $Q_i^* = \arg\max_{Q_i} I(\omega; q_i \mid Q_i, b^i)$. We let q_i^* denote the human's response to query Q_i^*. Similarly, let $\tilde{Q}_i$ be the information gain query *before collecting demonstrations*, so that $\tilde{Q}_i = \arg\max_{Q_i} I(\omega; q_i \mid Q_i, (\tilde{Q}_j, \tilde{q}_j)_{j=0}^{i-1})$. Again, we let $\tilde{q}_i$ denote the human's response to query $\tilde{Q}_i$. We can now compare the overall information gain for each order of questions and demonstrations:

$$
\begin{aligned}
&I\Big(\omega; (\mathcal{D}, q_1^*, q_2^*, \ldots) \mid (Q_1^*, Q_2^*, \ldots)\Big) \\
&\quad = I(\omega; \mathcal{D}) + I\left(\omega; (q_1^*, q_2^*, \ldots) \mid (b^0, Q_1^*, Q_2^*, \ldots)\right) \\
&\quad \geq I(\omega; \mathcal{D}) + I\Big(\omega; (\tilde{q}_1, \tilde{q}_2, \ldots) \mid (b^0, \tilde{Q}_1, \tilde{Q}_2, \ldots)\Big) \\
&\quad = I\Big(\omega; (\tilde{q}_1, \tilde{q}_2, \ldots, \mathcal{D}) \mid (\tilde{Q}_1, \tilde{Q}_2, \ldots)\Big) \qquad \square
\end{aligned}
$$

Intuition. We can explain Theorem 2 through two main insights. First, the information gain from a passively collected demonstration is the same regardless of when that demonstration is provided. Second, proactively generating questions based on a prior leads to more incisive queries than choosing questions from scratch. In fact, Theorem 2 can be generalized to show that active information resources should be utilized after passive resources.

Bounded Regret. At the start of this section we mentioned that — instead of looking for the optimal sequence of future questions — our DemPref robot will greedily choose the best query at the current iteration. Prior work has shown that this greedy approach is reasonable for volume removal, where it is guaranteed to have bounded sub-optimality in terms of the volume removed (Sadigh et al. 2017). However, this volume is defined in terms of the unnormalized distribution, and so this result does not say much about the learning performance. Unfortunately, the information gain also does not provide theoretical guarantees, as it is only submodular, but not *adaptive* submodular.

Useful Tools & Extensions

We introduced how robots can generate proactive questions to maximize volume removal or information gain. Below we highlight some additional tools that designers can leverage to improve the computational performance and applicability of these methods. In particular, we draw the reader's attention to an optimal stopping condition, which tells the DemPref robot when to stop asking the human questions.

Optimal Stopping. We propose a novel extension — specifically for information gain — that tells the robot when to stop asking questions. Intuitively, the DemPref querying process should end when the robot's questions become more costly to the human than informative to the robot.

Let each query Q have an associated cost $c(Q) \in \mathbb{R}^+$. This function captures the *cost* of a question: e.g., the amount of time it takes for the human to answer, the number of similar questions that the human has already seen, or even the interpretability of the question itself. We subtract this cost from our information gain objective in Equation (11), so that the robot maximizes information gain while biasing its search towards low-cost questions:

$$\max_{Q_i = \{\Lambda_1, \ldots, \Lambda_K\}} I(\omega; q_i \mid Q_i, b^i) - c(Q_i) \tag{14}$$

Importantly, Theorem 2 still holds with this extension to the model, since the costs depend only on the query. And now that we have introduced a cost into the query selection problem, the robot can reason about when its questions are becoming prohibitively expensive or redundant. We find that the best time to stop asking questions in expectation is when their cost exceeds their value:

Theorem 3. *A robot using information gain to perform active preference-based learning should stop asking questions if and only if the global solution to Equation (14) is negative at the current iteration.*

See the Appendix for our proof. We emphasize that this result is valid only for information gain, and adapting Theorem 3 to volume removal is not trivial.

The decision to terminate our DemPref algorithm is now fairly straightforward. At each iteration i, we search for the question Q_i that maximizes the trade-off between information gain and cost. If the value of Equation (14) is non-negative, the robot shows this query to the human and elicits their response; if not, the robot cannot find any sufficiently important questions to ask, and the process ends. This automatic stopping procedure makes the active learning process more user-friendly by ensuring that the user does not have to respond to unnecessary or redundant queries.

Other Potential Extensions. We have previously developed several tools to improve the computational efficiency of volume removal, or to extend volume removal to better accommodate human users. These tools include batch optimization (Biyik and Sadigh 2018; Bıyık et al. 2019), iterated correction (Palan et al. 2019), and dynamically changing reward functions (Basu et al. 2019). Importantly, the listed tools are *agnostic* to the details of volume removal: they simply require (a) the query generation algorithm to operate in a greedy manner while (b) maintaining a belief over ω. Our proposed information gain approach for generating easy queries satisfies both of these requirements. Hence, these prior extensions to volume removal can also be straightforwardly applied to information gain.

Algorithm 1 DemPref with a Human-in-the-Loop

Collect human demonstrations: $\mathcal{D} = \{\xi_1^D, \xi_2^D, \ldots, \xi_L^D\}$
Initialize belief over the human's reward weights ω:

$$b^0(\omega) \propto \exp\left(\beta^D \omega \cdot \sum_{\xi^D \in \mathcal{D}} \Phi(\xi^D)\right) P(\omega)$$

for $i \leftarrow 0, 1, \ldots$ **do**
    Choose proactive question Q_i:

$$Q_i \leftarrow \arg\max_{Q} \; I(\omega; q \mid Q, b^i) - c(Q)$$

    if $I(\omega; q \mid Q_i, b^i) - c(Q_i) < 0$ **then**
        return b^i
    end if
    Elicit human's answer q_i to query Q_i
    Update belief over ω given query and response:

$$b^{i+1}(\omega) \propto P(q_i \mid Q_i, \omega) b^i(\omega)$$

end for

Algorithm

We present the complete DemPref pseudocode in Algorithm 1. This algorithm involves two main steps: first, the robot uses the human's offline trajectory demonstrations $\mathcal{D}$ to initialize a high-level understanding of the human's preferred reward. Next, the robot actively generates user-friendly questions Q to fine-tune its belief b over ω. These questions can be selected using volume removal or information gain objectives (we highlight the information gain approach in Algorithm 1). As the robot asks questions and obtains a precise understanding of what the human wants, the informative value of new queries decreases: eventually, asking new questions becomes suboptimal, and the DemPref algorithm terminates.

Advantages. We conclude our presentation of DemPref by summarizing its two main contributions:

1. The robot learns the human's reward by synthesizing two types of information: high-level demonstrations and fine-grained preference queries.

2. The robot generates questions while accounting for the human's ability to respond, naturally leading to user-friendly and informative queries.

Experiments

We conduct five sets of experiments to assess the performance of DemPref under various metrics[‡]. We start by describing the simulation domains and the user study environment, and introducing the human choice models we employed for preference-based learning. Each subsequent subsection presents a set of experiments and tests the relevant hypotheses.

Simulation Domains. In each experiment, we use a subset of the following domains, shown in Figures 4 and 6, as well as a linear dynamical system:

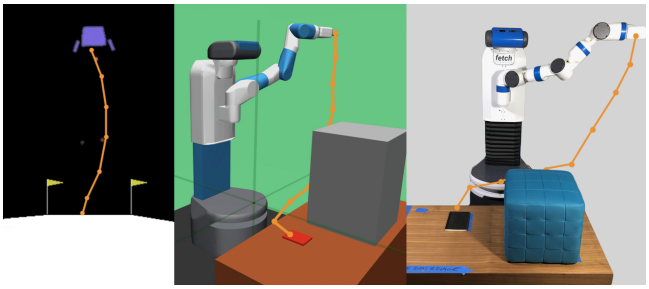

Figure 6. Views from simulation domains, with a demonstration in orange: **(a)** Lunar Lander, **(b)** Fetch (simulated), **(c)** Fetch (physical).

Linear Dynamical System (LDS): We use a linear dynamical system with six dimensional state and three dimensional action spaces. State values are directly used as state features without any transformation.

Driver: We use a 2D driving simulator (Sadigh et al. 2016), where the agent has to safely drive down a highway. The trajectory features correspond to the distance of the agent from the center of the lane, its speed, heading angle, and minimum distance to other vehicles during the trajectory (white in Figure 4 (top)).

Tosser: We use a Tosser robot simulation built in MuJoCo (Todorov et al. 2012) that tosses a capsule-shaped object into baskets. The trajectory features are the maximum horizontal distance forward traveled by the object, the maximum altitude of the object, the number of flips the object does, and the object's final distance to the closest basket.

Lunar Lander: We use the continuous LunarLander environment from OpenAI Gym (Brockman et al. 2016), where the lander has to safely reach the landing pad. The trajectory features correspond to the lander's average distance from the landing pad, its angle, its velocity, and its final distance to the landing pad.

Fetch: Inspired by Bajcsy et al. (2017), we use a modification of the Fetch Reach environment from OpenAI Gym (built on top of MuJoCo), where the robot has to reach a goal with its arm, while keeping its arm as low-down as possible (see Figure 6). The trajectory features correspond to the robot gripper's average distance to the goal, its average height from the table, and its average distance to a box obstacle in the domain.

For our user studies, we employ a version of the Fetch environment with the physical Fetch robot (see Figure 6) (Wise et al. 2016).

Human Choice Models. We require a probabilistic model for the human's choice q in a query Q conditioned on their reward parameters ω. Below, we discuss two specific models that we use in our experiments.

Standard Model Using Strict Preference Queries. Previous work demonstrated the importance of modeling imperfect human responses (Kulesza et al. 2014). We model a noisily optimal human as selecting ξ_k from a *strict* preference query

[‡]Unless otherwise noted, we adopt $\beta^D = 0.02$, constant $c(Q)$ for $\forall Q$, and assume a uniform prior over reward parameters ω, i.e., $P(\omega)$ is constant for any $\|\omega\|_2 \leq 1$. We use Metropolis-Hastings algorithm for sampling the set Ω from belief distribution over ω

Q by

$$P(q = \xi_k \mid Q, \omega) = \frac{\exp(R(\xi_k))}{\sum_{\xi \in Q} \exp(R(\xi))}, \quad (15)$$

where we call the query strict because the human is required to select one of the trajectories. This model, backed by neuroscience and psychology (Daw et al. 2006; Luce 2012; Ben-Akiva et al. 1985; Lucas et al. 2009), is routinely used (Biyik et al. 2019a; Guo and Sanner 2010; Viappiani and Boutilier 2010).

Extended Model Using Weak Preference Queries. We generalize this preference model to include an "About Equal" option for queries between two trajectories. We denote this option by Υ and define a *weak preference query* $Q^+ := Q \cup \{\Upsilon\}$ when $K = 2$.

Building on prior work by Krishnan (1977), we incorporate the information from the "About Equal" option by introducing a minimum perceivable difference parameter $\delta \geq 0$, and defining:

$$\begin{aligned} &P(q = \Upsilon \mid Q^+, \omega) = \\ &\quad (\exp(2\delta) - 1)\, P(q = \xi_1 \mid Q^+, \omega) P(q = \xi_2 \mid Q^+, \omega)\,, \\ &P(q = \xi_k \mid Q^+, \omega) = \\ &\quad \frac{1}{1 + \exp(\delta + R(\xi_{k'}) - R(\xi_k))}, \{\xi_k, \xi_{k'}\} = Q^+ \setminus \{\Upsilon\}. \end{aligned} \quad (16)$$

Notice that Equation (16) reduces to Equation (15) when $\delta = 0$; in which case we model the human as always perceiving the difference in options. All derivations in earlier sections hold with weak preference queries. In particular, we include a discussion of extending our formulation to the case where δ is user-specific and unknown in the Appendix. The additional parameter causes no trouble in practice. For all our experiments, we set $K = 2$, and $\delta = 1$ (whenever relevant).

We note that there are alternative choice models compatible with our framework for weak preferences (e.g., (Holladay et al. 2016)). Additionally, one may generalize the weak preference queries to $K > 2$, though it complicates the choice model as the user must specify which of the trajectories create uncertainty.

Evaluation Metric. To judge convergence of inferred reward parameters to true parameters in simulations, we adopt the *alignment metric* from Sadigh et al. (2017):

$$m = \frac{1}{M} \sum_{\bar{\omega} \in \Omega} \frac{\omega^* \cdot \bar{\omega}}{\|\omega^*\|_2 \|\bar{\omega}\|_2}, \quad (17)$$

where ω^* is the true reward parameters.

We are now ready to present our five sets of experiments each of which demonstrates a different aspect of the proposed DemPref framework:

1. The utility of initializing with demonstrations,
2. The advantages preference queries provide over using only demonstrations,
3. The advantages of information gain formulation over volume removal,
4. The order of demonstrations and preferences, and
5. Optimal stopping condition under the information gain objective.

Initializing with Demonstrations

We first investigate whether initializing the learning framework with user demonstrations is helpful. Specifically, we test the following hypotheses:

H1. *DemPref accelerates learning by initializing the prior belief b^0 using user demonstrations.*

H2. *The convergence of DemPref improves with the number of demonstrations used to initialize the algorithm.*

To test these two claims, we perform simulation experiments in Driver, Lunar Lander and Fetch environments. For each environment, we simulate a human user with hand-tuned reward function parameters ω, which gives reasonable performance. We generate demonstrations by applying model predictive control (MPC) to solve: $\max_\Lambda \omega^* \cdot \Phi(\Lambda)$. After initializing the belief with varying number of such demonstrations ($|\mathcal{D}| \in \{0, 1, 3\}$), the simulated users in each environment respond to 25 strict preference queries according to Equation (15), each of which is actively synthesized with the volume removal optimization. We repeat the same procedure for 8 times to obtain confidence bounds.

The results of the experiment are presented in Figure 7. On all three environments, initializing with demonstrations improves the convergence rate of the preference-based algorithm significantly; to match the m value attained by DemPref with only one demonstration in 10 preference queries, it takes the pure preference-based algorithm, i.e., without any demonstrations, 30 queries on Driver, 35 queries on Lander, and 20 queries on Fetch. These results provide strong evidence in favor of **H1**.

The results regarding **H2** are more complicated. Initializing with three instead of one demonstration improves convergence significantly only on the Driver and Lunar Lander domains. (The improvement on Driver is only at the early stages of the algorithm, when fewer than 10 preferences are used.) However, on the Fetch domain, initializing with three instead of one demonstration hurts the performance of the algorithm. (Although, we do note that the results from using three demonstrations are still an improvement over the results from not initializing with demonstrations). This is unsurprising. It is much harder to provide good demonstrations on the Fetch environment than on the Driver or Lunar Lander environments, and therefore the demonstrations are of lower quality. Using more demonstrations when they are of lower quality leads to the prior being more concentrated further away from the true reward function, and can cause the preference-based learning algorithm to slow down.

In practice, we find that using a single demonstration to initialize the algorithm leads to reliable improvements in convergence, regardless of the complexity of the domain.

DemPref vs IRL

Next, we analyze if preference elicitation improves learning performance. To do that, we conduct a within-subjects user study where we compare our DemPref algorithm with Bayesian IRL (Ramachandran and Amir 2007). The hypotheses we are testing are:

H3. *The robot which uses the reward function learned by DemPref will be more successful at the task (as evaluated by the users) than the IRL counterpart.*

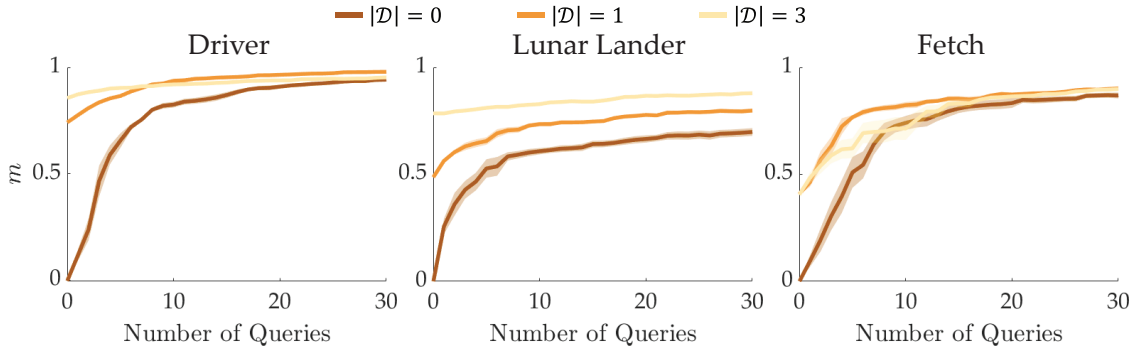

Figure 7. The results of our first experiment, investigating whether initializing with demonstrations improves the learning rate of the algorithm, on three domains. On the Driver, Lunar Lander, and Fetch environments, initializing with one demonstration improved the rate of convergence significantly.

H4. *Participants will prefer to use the DemPref framework as opposed to the IRL framework.*

For these evaluations, we use the Fetch domain with the physical Fetch robot. Participants were told that their goal was to get the robot's end-effector as close as possible to the goal, while (1) avoiding collisions with the block obstacle and (2) keeping the robot's end-effector low to the ground (so as to avoid, for example, knocking over objects around it). Participants provided demonstrations via teleoperation (using end-effector control) on a keyboard interface; each user was given some time to familiarize themselves with the teleoperation system before beginning the experiment.

Participants trained the robot using two different systems. (1) IRL: Bayesian IRL with 5 demonstrations. (2) DemPref: our DemPref framework (with the volume removal objective) with 1 demonstration and 15 proactive preference queries[§]. We counter-balanced across which system was used first, to minimize the impact of familiarity bias with our teleoperation system.

After training, the robot was trained in simulation using Proximal Policy Optimization (PPO) with the reward function learned from each system (Schulman et al. 2017). To ensure that the robot was not simply overfitting to the training domain, we used different variants of the domain for training and testing the robot. We used two different test domains (and counter-balanced across them) to increase the robustness of our results against the specific testing domain. Figure 8 (left) illustrates one of our testing domains. We rolled out three trajectories in the test domains for each algorithm on the physical Fetch. After observing each set of trajectories, the users were asked to rate the following statements on a 7-point Likert scale:

1. The robot accomplished the task well. (Accomplished)
2. The robot did what I wanted. (Did Wanted)
3. It was easy to train the robot with this system. (Easy)
4. I would want to use this system to train a robot in the future. (Would Use Again)

They were also asked two comparison questions:

1. Which robot accomplished the task better? (Better at Task)
2. Which system would you prefer to use if you had to train a robot to accomplish a similar task? (Preferred)

They were finally asked for general comments.

For this user study, we recruited 15 participants (11 male, 4 female), six of whom had prior experience in robotics but none of whom had any prior exposure to our system.

We present our results in Figure 8 (right). When asked which robot accomplished the task better, users preferred the DemPref system by a significant margin ($p < 0.05$, Wilcoxon paired signed-rank test); similarly, when asked which system they would prefer to use in the future if they had to train the robot, users preferred the DemPref system by a significant margin ($p < 0.05$). This provides strong evidence in favor of both **H3** and **H4**.

As expected, many users struggled to teleoperate the robot. Several users made explicit note of this fact in their comments: "I had a hard time controlling the robot", "I found the [IRL system] difficult as someone who [is not] kinetically gifted!", "Would be nice if the controller for the [robot] was easier to use." Given that the robot that employs IRL was only trained on these demonstrations, it is perhaps unsurprising that DemPref outperforms IRL on the task.

We were however surprised by the extent to which the IRL-powered robot fared poorly: in many cases, it did not even attempt to reach for the goal. Upon further investigation, we discovered that IRL was prone to, in essence, "overfitting" to the training domain. In several cases, IRL had overweighted the users' preference for obstacle avoidance. This proved to be an issue in one of our test domains where the obstacle is closer to the robot than in the training domain. Here, the robot does not even try to reach for the goal since the loss in value (as measured by the learned reward function) from going near the obstacle is greater than the gain in value from reaching for the goal. Figure 8 (left) shows this test domain and illustrates, for a specific user, a trajectory generated according to reward function learned by each of IRL and DemPref.

While we expect that IRL would overcome these issues with more careful feature engineering and increased diversity of the training domains, it is worth noting DemPref was affected much less by these issues. These results suggest preference-based learning methods may be more robust to poor feature engineering and a lack of training diversity

[§]The number of demonstrations and preferences used in each system were chosen such that a simulated agent achieves similar convergence to the true reward on both systems.

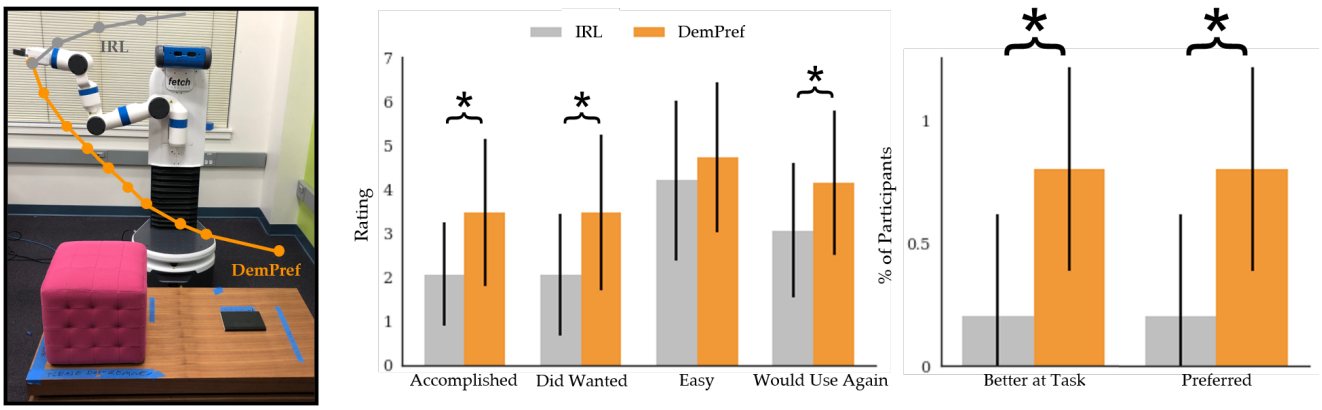

Figure 8. (Left) Our testing domain, with two trajectories generated according to the reward functions learned by IRL and DemPref from a specific user in our study. (Right) The results of our usability study – the error bars correspond to standard deviation and significant results are marked with an asterisk. We find that users rated the robot trained with DemPref as significantly better at accomplishing the task and preferred to use our method for training the robot significantly more than they did IRL. However, we did not find evidence to suggest that users found our method easier to use than standard IRL.

than IRL; however, a rigorous evaluation of these claims is beyond the scope of this paper.

It is interesting that despite the challenges that users faced with teleoperating the robot, they did not rate the DemPref system as being "easier" to use than the IRL system ($p = 0.297$). Several users specifically referred to the time it took to generate each query ($\sim$45 seconds) as negatively impacting their experience with the DemPref system: "I wish it was faster to generate the preference [queries]", "The [DemPref system] will be even better if time cost is less." Additionally, one user expressed difficulty in evaluating the preference queries themselves, commenting "It was tricky to understand/infer what the preferences were [asking]. Would be nice to communicate that somehow to the user (e.g. which [trajectory] avoids collision better)!", which highlights the fact that volume removal formulation may generate queries that are extremely difficult for the humans. Hence, we analyze in the next subsection how information gain objective improves the experience for the users.

Information Gain vs Volume Removal

To investigate the performance and user-friendliness of the information gain and volume removal methods for preference-based learning, we conduct experiments with simulated users in LDS, Driver, Tosser and Fetch environments; and real user studies in Driver, Tosser and Fetch (with the physical robot). We are particularly interested in the following three hypotheses:

H5. *Information gain formulation outperforms volume removal in terms of data-efficiency.*

H6. *Information gain queries are easier and more intuitive for the human than those from volume removal.*

H7. *A user's preference aligns best with reward parameters learned via information gain.*

To enable faster computation, we discretized the search space of the optimization problems by generating $500,000$ random queries and precomputing their trajectory features. Each call to an optimization problem then performs a loop over this discrete set.

In simulation experiments, we learn the randomly generated reward functions via both strict and weak preference queries where the "About Equal" option is absent and present, respectively. We repeat each experiment 100 times to obtain confidence bounds. Figure 9 shows the alignment value against query number for the 4 different tasks. Even though the "About Equal" option improves the performance of volume removal by preventing the trivial query, $Q = \{\xi_A, \xi_A, \dots\}$, from being a global optimum, information gain gives a significant improvement on the learning rate both with and without the "About Equal" option in all environments[¶]. These results strongly support **H5**.

The numbers given within Figure 10 count the wrong answers and "About Equal" choices made by the simulated users. The information gain formulation significantly improves over volume removal. Moreover, weak preference queries consistently decrease the number of wrong answers, which can be one reason why it performs better than strict queries[‖]. Figure 10 also shows when the wrong responses are given. While wrong answer ratios are higher with volume removal formulation, it can be seen that information gain reduces wrong answers especially in early queries, which leads to faster learning. These results support **H6**.

In the user studies for this part, we used Driver and Tosser environments in simulation and the Fetch environment with the physical robot. We began by asking participants to rank a set of features (described in plain language) to encourage each user to be consistent in their preferences. Subsequently, we queried each participant with a sequence of 30 questions generated actively; 15 from volume removal and 15 via information gain. We prevent bias by randomizing the sequence of questions for each user and experiment: the user does not know which algorithm generates a question.

Participants responded to a 7-point Likert scale survey after each question:

[¶] See the Appendix for results without query space discretization.

[‖] Another possible explanation is the information acquired by the "About Equal" responses. We analyze this in the Appendix by comparing the results with what would happen if this information was discarded.

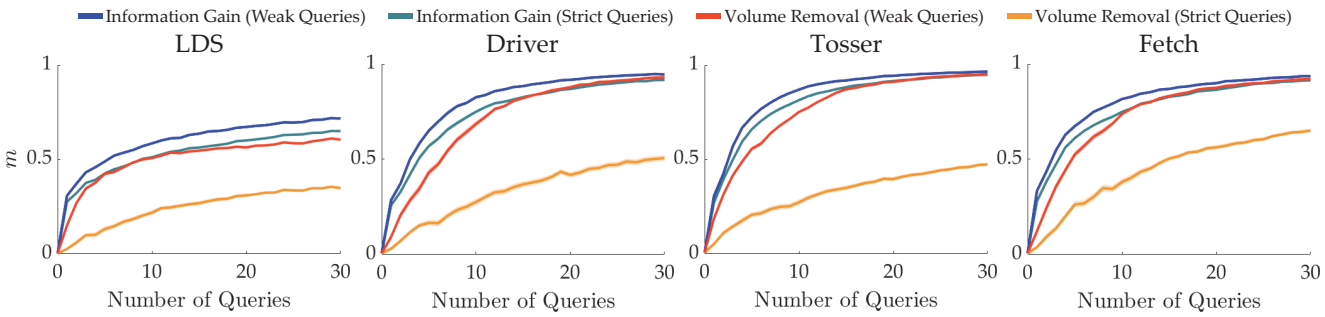

Figure 9. Alignment values are plotted (mean $\pm$ standard error) to compare information gain and volume removal formulations. Standard errors are so small that they are mostly invisible in the plots. Dashed lines show the weak preference query variants. Information gain provides a significant increase in learning rate in all cases. While weak preference queries lead to a large amount of improvement under volume removal, information gain formulation is still superior in terms of the convergence rate.

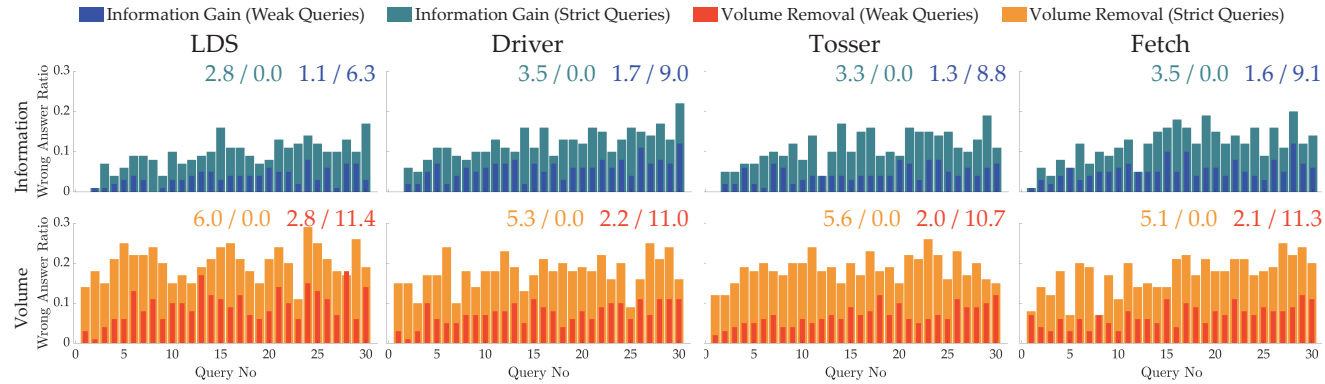

Figure 10. Wrong answer ratios on different queries are shown. The numbers at top show the average number of wrong responses and "About Equal" choices, respectively, for both strict and weak queries. Information gain formulation yields smaller numbers of wrong and "About Equal" answers, especially in the early stages.

1. It was easy to choose between the trajectories that the robot showed me.

They were also asked the Yes/No question:

1. Can you tell the difference between the options presented?

In concluding the Tosser and Driver experiments, we showed participants two trajectories: one optimized using reward parameters from information gain (trajectory A) and one optimized using reward parameters from volume removal (trajectory B)**. Participants responded to a 7-point Likert scale survey:

1. Trajectory A better aligns with my preferences than trajectory B.

We recruited 15 participants (8 female, 7 male) for the simulations (Driver and Tosser) and 12 for the Fetch (6 female, 6 male). We used strict preference queries. A video demonstration of these user studies is available at http://youtu.be/JIs43cO_g18.

Figure 11 (a) shows the results of the easiness surveys. In all environments, users found information gain queries easier: the results are statistically significant ($p < 0.005$, two-sample t-test). Figure 11 (b) shows the average number of times the users stated they cannot distinguish the options presented. The volume removal formulation yields several queries that are indistinguishable to the users while the information gain avoids this issue. The difference is significant for Driver ($p < 0.05$, paired-sample t-test) and Tosser ($p < 0.005$). Taken together, these results support **H6**.

Figure 11 (c) shows the results of the survey the participants completed at the end of experiment. Users significantly preferred the information gain trajectory over that of volume removal in both environments ($p < 0.05$, one-sample t-test), supporting **H7**.

The Order of Information Sources

Having seen information gain provides a significant boost in the learning rate, we checked whether the passively collected demonstrations or the actively queried preferences should be given to the model first. Specifically, we tested:

H8. *If passively collected demonstrations are used before the actively collected preference query responses, the learning becomes faster.*

While Theorem 2 asserts that we should first initialize DemPref via demonstrations, we performed simulation experiments to check this notion in practice. Using LDS, Driver, Tosser and Fetch, we ran three sets of experiments where we adopted weak preference queries: (i) We initialize the belief with a single demonstration and then query the simulated user with 15 preference questions, (ii) We first query the simulated user with 15 preference questions and we add the demonstration to the belief independently after each question, and (iii) We completely ignore the demonstration and use only 15 preference queries. The reason why we chose to have only a single demonstration is because having more more demonstrations tends to increase the alignment value m

**We excluded Fetch for this question to avoid prohibitive trajectory optimization (due to large action space).

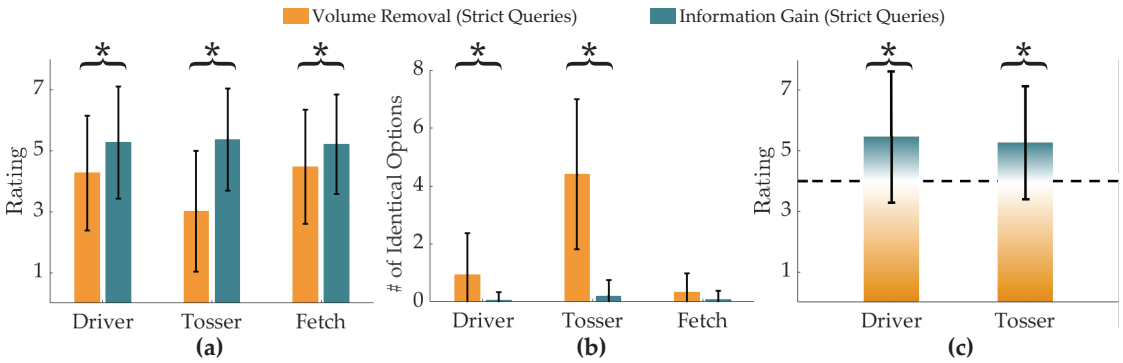

Figure 11. User study results. Error bars show std. Asterisks show statistical significance. **(a)** Easiness survey results averaged over all queries and users. Queries generated using the information gain method are rated significantly easier by the users than the volume removal queries. **(b)** The number of identical options in the experiments averaged over all users. In Driver and Tosser, users indicated significantly less indistinguishable queries with information gain compared to volume removal. **(c)** Final preferences averaged over the users. 7 means the user strongly prefers the optimized trajectory w.r.t. the learned reward by the information gain formulation, and 1 is the volume removal. Dashed line represents indifference between two methods. Users significantly prefer the robot who learned using the information gain method for active query generation.

for both (i) and (ii), thereby making the difference between the methods' performances very small. We ran each set of experiment 100 times with different, randomly sampled, true reward functions. We again used the same data set of $500,000$ queries for query generation. We also used the trajectory that gives the highest reward to the simulated user out of this data set as the demonstration in the first two sets of experiments. Since the demonstration is not subject to noises or biases due to the control of human users, we set $\beta^D = 0.2$.

Figure 12 shows the alignment value against the number of queries. The last set of experiments has significantly lower alignment values than the first two sets especially when the number of preference queries is small. This indicates the demonstration has carried an important amount of information. Comparing the first two sets of experiments, the differences in the alignment values are small. However, the values are consistently higher when the demonstrations are used to initialize the belief distribution. This supports **H8** and numerically validates Theorem 2.

Optimal Stopping

Finally, we experimented our optimal stopping extension for information gain based active querying algorithm in LDS, Driver, Tosser and Fetch environments with simulated users. Again adopting query discretization, we tested:

H9. *Optimal stopping enables cost-efficient reward learning under various costs.*

As the query cost, we employed a cost function to improve interpretability of queries, which may have the associated benefit of making learning more efficient (Bajcsy et al. 2018). We defined a cost function:

$$c(Q) = \epsilon - |\Psi_{i^*}| + \max_{j \in \{1,\dots\} \setminus \{i^*\}} |\Psi_j|,\ i^* = \arg\max_i |\Psi_i|,$$

where $Q = \{\xi_1, \xi_2\}$ and $\Psi = \Phi(\xi_1) - \Phi(\xi_2)$. This cost favors queries in which the difference in one feature is larger than that between all other features. Such a query may prove more interpretable. We first simulate 100 random users and tune ϵ accordingly: For each simulated user, we record the ϵ value that makes the objective zero in the i^{th} query (for smallest i) such that $m_i, m_{i-1}, m_{i-2} \in [x, x + 0.02]$ for some x. We then use the average of these ϵ values for our tests with 100 different random users. Figure 13 shows the results[††]. Optimal stopping rule enables terminating the process with near-optimal cumulative active learning rewards (the cumulative difference between the information gain and the cost as in Equation (14)) in all environments, which supports **H9**.

Conclusion

Summary. In this paper, we developed a framework that utilizes multiple sources of information to learn the preference reward functions of the human users, demonstrations and preference queries in particular. In addition to proving that active information sources must be used after the passive sources for more efficient learning, we highlighted the important problems associated with the state-of-the-art active preference query generation method. The alternative technique we proposed, information gain maximization, has proven to not suffer from those problems while being at least tractable. Moreover, our proposed technique allows the designers to incorporate an optimal stopping condition, which improves the usefulness of the entire framework. We performed a large number of simulation experiments and user studies on a physical robot whose results have demonstrated the superiority of our proposed method over various state-of-the-art alternatives both in terms of performance and user preferences, as well as the validity of the optimal stopping criterion and the order of information sources.

Limitations and Future Work. In this work, we focused on two common sources of information: demonstrations and preference questions. While we showed that active information sources must be used after the passive ones, our work is limited in terms of the number of information sources considered: once multiple active information sources are available, their optimal order may depend on various factors, such as how informative or how costly they are. Besides, the information sources we adopt in this paper can be further extended. For example, "About Equal"

[††]We found similar results with query-independent costs minimizing the number of queries. See Appendix.

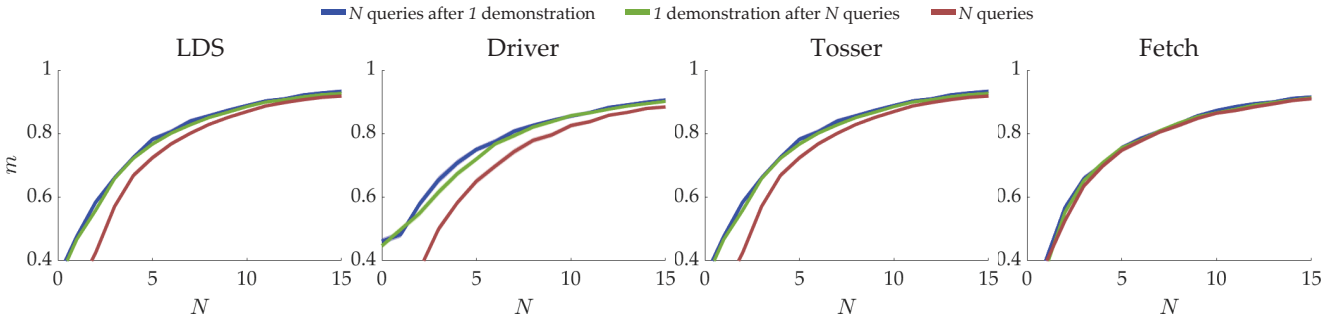

Figure 12. Simulation results for the order of demonstrations and preference queries. Alignment values are plotted (mean±s.e.). It is consistently better to first utilize the passively collected demonstrations rather than actively generated preference queries. The differences in the alignment value is especially small in the Fetch simulations, which might be due to the fact that it is a simpler environment in terms of the number of trajectory features.

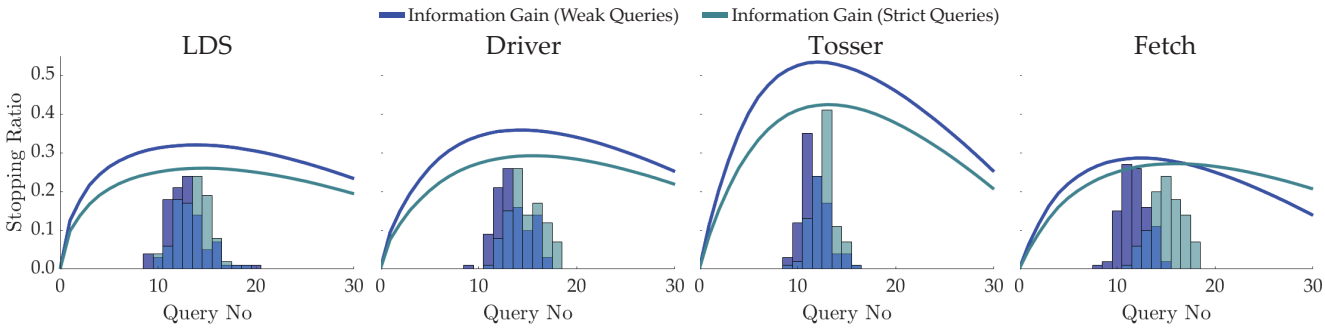

Figure 13. Simulation results for optimal stopping. Line plots show cumulative active learning rewards (cumulative difference between the information gain values and the query costs), averaged over 100 test runs and scaled for visualization. Histograms show when optimal stopping condition is satisfied, which aligns with the desired cumulative rewards.

option in preference queries allowed users to give more expressive feedbacks. One could also think of other forms of responses, such as "I (dis)like both" or "I like this more than the other by this much", in which cases a better human choice model would be needed. Secondly, we considered a standard reward model in IRL which is linear in some trajectory features. While this might be expressive enough with well-designed features, it might be necessary in some cases to account for nonlinearities. Even though we recently developed an active preference query generation method for rewards modeled using Gaussian processes to tackle this issue (Biyik et al. 2020), incorporating demonstrations to this framework to attain a nonlinear version of DemPref might be nontrivial. Third, our work is limited in terms of the number of baselines used for comparison. While, to the best of our knowledge, prior works do not incorporate demonstrations and preference queries, each component could be individually tested against other baselines (e.g., Brown and Niekum (2019); Michini and How (2012); Park et al. (2020)), which might inform a better DemPref algorithm. Recent research which builds on our approach highlights that preference questions generated with information gain — although easy to answer — may mislead the human into thinking the robot does not understand what it has actually learned (Habibian et al. 2021). Finally, our results showed having too many demonstrations, which are imprecise sources of information, might harm the performance of DemPref. An interesting future research direction is then to investigate the optimal number of demonstrations, and to decide when and which demonstrations are helpful.

Acknowledgments

This work is supported by FLI grant RFP2-000, and NSF Awards #1849952 and #1941722.

References

Abbeel P and Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM, p. 1.

Abbeel P and Ng AY (2005) Exploration and apprenticeship learning in reinforcement learning. In: *Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 1–8.

Ailon N (2012) An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research* 13(Jan): 137–164.

Akgun B, Cakmak M, Jiang K and Thomaz AL (2012) Keyframe-based learning from demonstration. *International Journal of Social Robotics* 4(4): 343–355.

Akrour R, Schoenauer M and Sebag M (2012) April: Active preference learning-based reinforcement learning. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 116–131.

Bajcsy A, Losey DP, O'Malley MK and Dragan AD (2018) Learning from physical human corrections, one feature at a time. In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, pp. 141–149.

Bajcsy A, Losey DP, O'Malley MK and Dragan AD (2017) Learning robot objectives from physical human interaction. *Proceedings of Machine Learning Research* 78: 217–226.

Basu C, Biyik E, He Z, Singhal M and Sadigh D (2019) Active learning of reward dynamics from hierarchical queries. In:

Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Basu C, Yang Q, Hungerman D, Sinahal M and Draqan AD (2017) Do you want your autonomous car to drive like you? In: *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, pp. 417–425.

Ben-Akiva ME, Lerman SR and Lerman SR (1985) *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press.

Biyik E, Huynh N, Kochenderfer MJ and Sadigh D (2020) Active preference-based gaussian process regression for reward learning. In: *Proceedings of Robotics: Science and Systems (RSS)*.

Biyik E, Lazar DA, Sadigh D and Pedarsani R (2019a) The green choice: Learning and influencing human decisions on shared roads. In: *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*.

Biyik E, Palan M, Landolfi NC, Losey DP and Sadigh D (2019b) Asking easy questions: A user-friendly approach to active reward learning. In: *Proceedings of the 3rd Conference on Robot Learning (CoRL)*.

Biyik E and Sadigh D (2018) Batch active preference-based learning of reward functions. In: *Conference on Robot Learning (CoRL)*.

Bıyık E, Wang K, Anari N and Sadigh D (2019) Batch active learning using determinantal point processes. *arXiv preprint arXiv:1906.07975* .

Bobu A, Bajcsy A, Fisac JF and Dragan AD (2018) Learning under misspecified objective spaces. In: *Conference on Robot Learning*. pp. 796–805.

Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J and Zaremba W (2016) Openai gym. *arXiv preprint arXiv:1606.01540* .

Brown D, Goo W, Nagarajan P and Niekum S (2019) Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In: *International Conference on Machine Learning*. pp. 783–792.

Brown DS, Goo W and Niekum S (2020) Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In: *Conference on robot learning*. PMLR, pp. 330–359.

Brown DS and Niekum S (2019) Deep bayesian reward learning from preferences. In: *Workshop on Safety and Robustness in Decision Making at the 33rd Conference on Neural Information Processing Systems (NeurIPS) 2019*.

Cakmak M, Srinivasa SS, Lee MK, Forlizzi J and Kiesler S (2011) Human preferences for robot-human hand-over configurations. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1986–1993.

Chen L, Paleja R and Gombolay M (2020) Learning from suboptimal demonstration via self-supervised reward regression. In: *Conference on robot learning*. PMLR.

Choudhury R, Swamy G, Hadfield-Menell D and Dragan AD (2019) On the utility of model learning in hri. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, pp. 317–325.

Christiano PF, Leike J, Brown T, Martic M, Legg S and Amodei D (2017) Deep reinforcement learning from human preferences. In: *Advances in Neural Information Processing Systems*. pp. 4299–4307.

Chu W and Ghahramani Z (2005) Gaussian processes for ordinal regression. *Journal of machine learning research* 6(Jul): 1019–1041.

Cover TM and Thomas JA (2012) *Elements of information theory*. John Wiley & Sons.

Daw ND, O'doherty JP, Dayan P, Seymour B and Dolan RJ (2006) Cortical substrates for exploratory decisions in humans. *Nature* 441(7095): 876.

Dragan AD and Srinivasa SS (2012) *Formalizing assistive teleoperation*. MIT Press, July.

Guo S and Sanner S (2010) Real-time multiattribute bayesian preference elicitation with pairwise comparison queries. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. pp. 289–296.

Habibian S, Jonnavittula A and Losey DP (2021) Here's what I've learned: Asking questions that reveal reward learning. *arXiv preprint arXiv:2107.01995* .

Holladay R, Javdani S, Dragan A and Srinivasa S (2016) Active comparison based learning incorporating user uncertainty and noise. In: *RSS Workshop on Model Learning for Human-Robot Communication*.

Ibarz B, Leike J, Pohlen T, Irving G, Legg S and Amodei D (2018) Reward learning from human preferences and demonstrations in atari. In: *Advances in neural information processing systems*. pp. 8011–8023.

Javdani S, Srinivasa SS and Bagnell JA (2015) Shared autonomy via hindsight optimization. *Robotics science and systems: online proceedings* 2015.

Katz S, Maleki A, Biyik E and Kochenderfer MJ (2021) Preference-based learning of reward function features. *arXiv preprint arXiv:2103.02727* .

Katz SM, Bihan ACL and Kochenderfer MJ (2019) Learning an urban air mobility encounter model from expert preferences. In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*.

Khurshid RP and Kuchenbecker KJ (2015) Data-driven motion mappings improve transparency in teleoperation. *Presence: Teleoperators and Virtual Environments* 24(2): 132–154.

Krishnan K (1977) Incorporating thresholds of indifference in probabilistic choice models. *Management science* 23(11): 1224–1233.

Kulesza T, Amershi S, Caruana R, Fisher D and Charles D (2014) Structured labeling for facilitating concept evolution in machine learning. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 3075–3084.

Kwon M, Biyik E, Talati A, Bhasin K, Losey DP and Sadigh D (2020) When humans aren't optimal: Robots that collaborate with risk-aware humans. In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.

Lepird JR, Owen MP and Kochenderfer MJ (2015) Bayesian preference elicitation for multiobjective engineering design optimization. *Journal of Aerospace Information Systems* 12(10): 634–645.

Li K, Tucker M, Biyik E, Novoseller E, Burdick JW, Sui Y, Sadigh D, Yue Y and Ames AD (2021a) Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. In: *International Conference on Robotics and Automation (ICRA)*.

Li M, Canberk A, Losey DP and Sadigh D (2021b) Learning human objectives from sequences of physical corrections. In: *International Conference on Robotics and Automation (ICRA)*.

Lucas CG, Griffiths TL, Xu F and Fawcett C (2009) A rational model of preference learning and choice prediction by children. In: *Advances in neural information processing systems*. pp. 985–992.

Luce RD (2012) *Individual choice behavior: A theoretical analysis*. Courier Corporation.

Michini B and How JP (2012) Bayesian nonparametric inverse reinforcement learning. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, pp. 148–163.

Ng AY, Russell SJ et al. (2000) Algorithms for inverse reinforcement learning. In: *Icml*, volume 1. p. 2.

Nikolaidis S, Ramakrishnan R, Gu K and Shah J (2015) Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In: *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, pp. 189–196.

Palan M, Shevchuk G, Landolfi NC and Sadigh D (2019) Learning reward functions by integrating human demonstrations and preferences. In: *Proceedings of Robotics: Science and Systems (RSS)*.

Park D, Noseworthy M, Paul R, Roy S and Roy N (2020) Inferring task goals and constraints using bayesian nonparametric inverse reinforcement learning. In: *Proceedings of the Conference on Robot Learning*, *Proceedings of Machine Learning Research*, volume 100. PMLR, pp. 1005–1014.

Ramachandran D and Amir E (2007) Bayesian inverse reinforcement learning. In: *IJCAI*, volume 7. pp. 2586–2591.

Sadigh D, Dragan AD, Sastry SS and Seshia SA (2017) Active preference-based learning of reward functions. In: *Proceedings of Robotics: Science and Systems (RSS)*.

Sadigh D, Sastry S, Seshia SA and Dragan AD (2016) Planning for autonomous cars that leverage effects on human actions. In: *Robotics: Science and Systems*, volume 2. Ann Arbor, MI, USA.

Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O (2017) Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Shah A and Shah J (2020) Interactive robot training for non-markov tasks. *arXiv preprint arXiv:2003.02232* .

Todorov E, Erez T and Tassa Y (2012) Mujoco: A physics engine for model-based control. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5026–5033.

Tucker M, Novoseller E, Kann C, Sui Y, Yue Y, Burdick J and Ames AD (2020) Preference-based learning for exoskeleton gait optimization. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Viappiani P and Boutilier C (2010) Optimal bayesian recommendation sets and myopically optimal choice query sets. In: *Advances in neural information processing systems*. pp. 2352–2360.

Wilde N, Kulić D and Smith SL (2019) Bayesian active learning for collaborative task specification using equivalence regions. *IEEE Robotics and Automation Letters* 4(2): 1691–1698.

Wise M, Ferguson M, King D, Diehr E and Dymesich D (2016) Fetch and freight: Standard platforms for service robot applications. In: *Workshop on Autonomous Mobile Service Robots*.

Ziebart BD, Maas AL, Bagnell JA and Dey AK (2008) Maximum entropy inverse reinforcement learning. In: *Aaai*, volume 8. Chicago, IL, USA, pp. 1433–1438.

Appendix

In the appendix, we present

1. The derivation of the information gain formulation,
2. The proof of Theorem 3,
3. The extension of the information gain formulation to the case with user-specific and unknown δ under weak preference queries,
4. The comparison of information gain and volume removal formulations without query space discretization, i.e., with continuous trajectory optimization,
5. The experimental analysis of the effect of the information coming from "About Equal" responses in weak preference queries, and
6. Optimal stopping under query-independent costs.

Derivation of Information Gain Solution

We first present the full derivation of Equation (12),

$$Q_i^* = \underset{Q_i=\{\Lambda_1,\dots,\Lambda_K\}}{\arg\max} \; I(q_i;\omega \mid Q_i, b^i) \; .$$

We first write the mutual information as the difference between two entropy terms:

$$\begin{aligned} &I(q_i;\omega \mid Q_i, b^i) \\ &\quad = H(\omega \mid Q_i, b^i) - \mathbb{E}_{q_i|Q_i,b^i}\left[H(\omega \mid q_i, Q_i, b^i)\right] \; . \end{aligned}$$

Next, we expand the entropy expressions and use $P(q_i \mid Q_i, b^i)P(\omega \mid q_i, Q_i, b^i) = P(\omega, q_i \mid Q_i, b^i)$ to combine the expectations for the second term to get:

$$\begin{aligned} &H(\omega \mid Q_i, b^i) - \mathbb{E}_{q_i|Q_i,b^i}\left[H(\omega \mid q_i, Q_i, b^i)\right] \\ &\quad = -\mathbb{E}_{\omega|Q_i,b^i}\left[\log_2 P(\omega \mid Q_i, b^i)\right] + \\ &\qquad \mathbb{E}_{\omega,q_i|Q_i,b^i}\left[\log_2\left(P(\omega \mid q_i, Q_i, b^i)\right)\right] \; . \end{aligned}$$

Since the first term is independent from q_i, we can write this expression as

$$\begin{aligned} &\mathbb{E}_{\omega,q_i|Q_i,b^i}\left[\log_2 P(\omega \mid q_i, Q_i, b^i) - \log_2 P(\omega \mid Q_i, b^i)\right] \\ &= \mathbb{E}_{\omega,q_i|Q_i,b^i}\left[\log_2 P(q_i \mid Q_i, b^i, \omega) - \log_2 P(q_i \mid Q_i, b^i)\right] \\ &= \mathbb{E}_{\omega,q_i|Q_i,b^i}\Bigg[\log_2 P(q_i \mid Q_i, \omega) - \\ &\qquad \log_2\left(\int P(q_i \mid Q_i, \omega')P(\omega' \mid Q_i, b^i)d\omega'\right)\Bigg] \; , \end{aligned}$$

where the integral is taken over all possible values of ω.

Having Ω as a set of M samples drawn from the prior b^i,

$$
\begin{aligned}
&I(q_i;\omega \mid Q_i)\\
&\doteq \mathbb{E}_{\omega,q_i|Q_i}\Bigg[\log_2 P(q_i \mid Q_i,\omega) - \\
&\qquad \log_2\left(\frac{1}{M}\sum_{\omega'\in\Omega} P(q_i \mid Q_i,\omega')\right)\Bigg]\\
&= \mathbb{E}_{\omega,q_i|Q_i}\left[\log_2 \frac{M\cdot P(q_i \mid Q_i,\omega)}{\sum_{\omega'\in\Omega} P(q_i \mid Q_i,\omega')}\right]\\
&= \mathbb{E}_{\omega|Q_i}\left[\mathbb{E}_{q_i|Q_i,\omega}\left[\log_2 \frac{M\cdot P(q_i \mid Q_i,\omega)}{\sum_{\omega'\in\Omega} P(q_i \mid Q_i,\omega')}\right]\right]\\
&= \mathbb{E}_{\omega|Q_i}\left[\sum_{q_i\in Q_i} P(q_i \mid Q_i,\omega)\log_2 \frac{M\cdot P(q_i \mid Q_i,\omega)}{\sum_{\omega'\in\Omega} P(q_i \mid Q_i,\omega')}\right]\\
&\doteq \frac{1}{M}\sum_{q_i\in Q_i}\sum_{\bar{\omega}\in\Omega} P(q_i \mid Q_i,\bar{\omega})\log_2 \frac{M\cdot P(q_i \mid Q_i,\bar{\omega})}{\sum_{\omega'\in\Omega} P(q_i \mid Q_i,\omega')},
\end{aligned}
$$

where, in the last step, we use the sampled ω's to compute the expectation over $\omega \mid Q_i$. This completes the derivation.

Proof of Theorem 3

Theorem 3. *Terminating the algorithm is optimal if and only if global solution to* (14) *is negative.*

Proof. We need to show if the global optimum is negative, then any longer-horizon optimization will also give negative reward (difference between information gain and the cost) in expectation. Let Q_i^* denote the global optimizer. For any $k \geq 0$,

$$
\begin{aligned}
&I(q_i,\ldots,q_{i+k};\omega \mid Q_i,\ldots,Q_{i+k}) - \sum_{j=0}^{k} c(Q_{i+j})\\
&\quad = I(q_i;\omega \mid Q_i) + \cdots +\\
&\qquad I(q_{i+k};\omega \mid q_i,\ldots,q_{n+k-1},Q_i,\ldots,Q_{i+k}) - \sum_{j=0}^{k} c(Q_{i+j})\\
&\quad \leq I(q_i;\omega \mid Q_i) + \ldots + I(q_{i+k};\omega \mid Q_{i+k}) - \sum_{j=0}^{k} c(Q_{i+j})\\
&\quad \leq (k+1)\left[I(q_i;\omega \mid Q_i^*) - c(Q_i^*)\right] < 0
\end{aligned}
$$

where the first inequality is due to the submodularity of the mutual information, and the second inequality is because Q_n^* is the global maximizer of the greedy objective. The other direction of the proof is very clear: If the global optimizer is nonnegative, then querying Q_n^* will not decrease the cumulative active learning reward in expectation, so stopping is not optimal.

Extension to User-Specific and Unknown δ

We now derive the information gain solution when the minimum perceivable difference parameter δ of the extended human model (for weak preference queries) we introduced in the Experiments section is unknown. One can also introduce a temperature parameter β to both standard and extended models such that $R(\xi_k)$ values will be replaced with $\beta R(\xi_k)$ in Eqs. (15) and (16). This temperature parameter is useful for setting how noisy the user choices are, and learning it can ease the feature design.

Therefore, for generality, we denote all human model parameters that will be learned as a vector ν. Furthermore, we denote the belief over (ω,ν) at iteration i as b_+^i. Since our true goal is to learn ω, the optimization now becomes:

$$
Q_i^* = \underset{Q_i=\{\Lambda_1,\ldots,\Lambda_K\}}{\arg\max}\ \mathbb{E}_{\nu|Q_i,b_+^i}\left[I(q_i;\omega \mid Q_i,b_+^i)\right]
$$

We now work on this objective as follows:

$$
\begin{aligned}
&\mathbb{E}_{\nu|Q_i,b_+^i}\left[I(q_i;\omega \mid Q_i,b_+^i)\right]\\
&\quad = \mathbb{E}_{\nu|Q_i,b_+^i}\Big[H(\omega \mid \nu,Q_i,b_+^i) - \\
&\qquad \mathbb{E}_{q_i|\nu,Q_i,b_+^i}\left[H(\omega \mid q_i,\nu,Q_i,b_+^i)\right]\Big]\\
&\quad = \mathbb{E}_{\nu|Q_i,b_+^i}\left[H(\omega \mid \nu,Q_i,b_+^i)\right] - \\
&\qquad \mathbb{E}_{\nu,q_i|Q_i,b_+^i}\left[H(\omega \mid q_i,\nu,Q_i,b_+^i)\right]\\
&\quad = -\mathbb{E}_{\nu,\omega|Q_i,b_+^i}\left[\log_2 P(\omega \mid \nu,Q_i,b_+^i)\right] + \\
&\qquad \mathbb{E}_{\nu,q_i,\omega|Q_i,b_+^i}\left[\log_2 P(\omega \mid q_i,\nu,Q_i,b_+^i)\right]\\
&\quad = \mathbb{E}_{\nu,q_i,\omega|Q_i,b_+^i}\left[\log_2 P(\omega \mid q_i,\nu,Q_i,b_+^i) - \right.\\
&\qquad \left.\log_2 P(\omega \mid \nu,Q_i,b_+^i)\right]\\
&\quad = \mathbb{E}_{\nu,q_i,\omega|Q_i,b_+^i}\left[\log_2 P(q_i \mid \omega,\nu,Q_i,b_+^i) - \right.\\
&\qquad \left.\log_2 P(q_i \mid \nu,Q_i,b_+^i)\right]\\
&\quad = \mathbb{E}_{\nu,q_i,\omega|Q_i,b_+^i}\left[\log_2 P(q_i \mid \omega,\nu,Q_i,b_+^i) - \right.\\
&\qquad \left.\log_2 P(\nu,q_i \mid Q_i,b_+^i) + \log_2 P(\nu \mid Q_i,b_+^i)\right]
\end{aligned}
$$

Noting that $P(\nu \mid Q_i,b_+^i) = P(\nu \mid b_+^i)$, we drop the last term because it does not involve the optimization variable Q_i. Also noting $P(q_i \mid \omega,\nu,Q_i,b_+^i) = P(q_i \mid \omega,\nu,Q_i)$, the new objective is:

$$
\begin{aligned}
&\mathbb{E}_{\nu,q_i,\omega|Q_i,b_+^i}\left[\log_2 P(q_i \mid \omega,\nu,Q_i) - \log_2 P(\nu,q_i \mid Q_i,b_+^i)\right]\\
&\doteq \frac{1}{M}\sum_{(\bar{\omega},\bar{\nu})\in\Omega^+}\sum_{q_i\in Q_i} P(q_i \mid \bar{\omega},\bar{\nu},Q_i)\\
&\qquad \left[\log_2 P(q_i \mid \bar{\omega},\bar{\nu},Q_i) - \log_2 P(\bar{\nu},q_i \mid Q_i,b_+^i)\right]
\end{aligned}
$$

where Ω^+ is a set containing M samples from b_+^i. Since $P(\bar{\nu},q_i \mid Q_i,b_+^i) = \int P(q_i \mid \bar{\nu},\omega',Q_i)P(\bar{\nu},\omega' \mid Q_i,b_+^i)d\omega'$ where the integration is over all possible values of ω, we can write the second logarithm term as:

$$
\log_2\left(\frac{1}{M}\sum_{\omega'\in\Omega(\bar{\nu})} P(q_i \mid \bar{\nu},\omega',Q_i)\right)
$$

with asymptotic equality, where $\Omega(\bar{\nu})$ is the set that contains M samples from b_+^i with fixed $\bar{\nu}$. Note that while we can actually compute this objective, it is computationally much heavier than the case without ν, because we need to take M samples of ω for each $\bar{\nu}$ sample.

One property of this objective that will ease the computation is the fact that it is parallelizable. An alternative approach is to actively learn (ω,ν) instead of just ω. This will of course cause some performance loss, because we are only interested in ω. However, if we learn them together, the

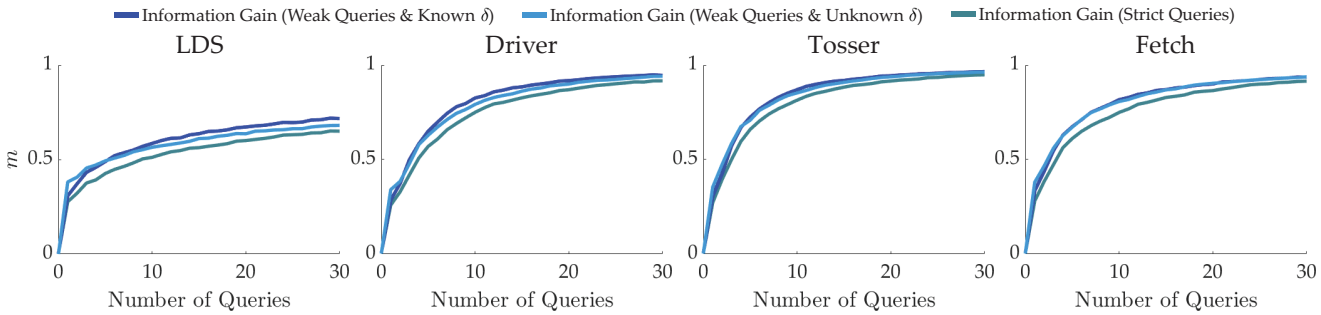

Figure 14. The simulation results with information gain formulation for unknown δ. Plots are mean±s.e.

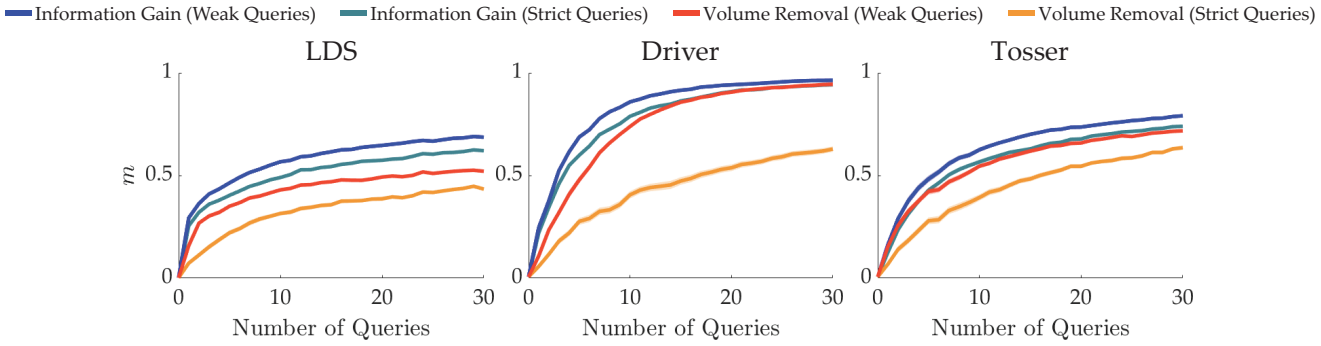

Figure 15. Alignment values are plotted (mean±s.e.) for th experiments without query space discretization, i.e., with continuous trajectory optimization for active query generation.

derivation follows the derivation of Equation (12), which we already presented, by simply replacing ω with (ω, ν), and the final optimization becomes:

$$\underset{Q_i=\{\Lambda_1,\dots,\Lambda_K\}}{\arg\max} \frac{1}{M} \sum_{q_i \in Q_i} \sum_{(\bar{\omega},\bar{\nu})\in\Omega^+} P(q_i \mid Q_i, \bar{\omega}, \bar{\nu}) \log_2 \frac{M \cdot P(q_i \mid Q_i, \bar{\omega}, \bar{\nu})}{\sum_{(\omega',\nu')\in\Omega^+} P(q_i \mid Q_i, \omega', \nu')}$$

Using this approximate, but computationally faster optimization, we performed additional analysis where we compare the performances of strict preference queries, weak preference queries with known δ and weak preference queries without assuming any δ (all with the information gain formulation). As in the previous simulations, we simulated 100 users with different random reward functions. Each user is simulated to have a true δ, uniformly randomly taken from $[0, 2]$. During the sampling of Ω^+, we did not assume any prior knowledge about δ, except the natural condition that $\delta \geq 0$. The comparison results are in Figure 14. While knowing δ increases the performance as expected, weak preference queries are still better than strict queries even when δ is unknown. This supports the advantage of employing weak preference queries.

Comparison of Information Gain and Volume Removal without Query Space Discretization

We repeated the experiment that supports **H5**, and whose results are shown in Figure 9, without query space discretization. By optimizing over the continuous action space of the environments, we tested information gain and volume removal formulations with both strict and weak preference queries in LDS, Driver and Tosser tasks. We excluded Fetch again in order to avoid prohibitive trajectory optimization due to large action space. Figure 15 shows the result. As it is expected, information gain formulation outperforms the volume removal with both preference query types. And, weak preference queries lead to faster learning compared to strict preference queries.

Effect of Information from "About Equal" Responses

We have seen that weak preference queries consistently decrease wrong answers and improve the performance. However, this improvement is not necessarily merely due to the decrease in wrong answers. It can also be credited to the information we acquire thanks to "About Equal" responses.

To investigate the effect of this information, we perform two additional experiments with 100 different simulated human reward functions with weak preference queries: First, we use the information by the "About Equal" responses; and second, we ignore such responses and remove the query from the query set to prevent repetition. Figure 16 shows the results. It can be seen that for both volume removal and information gain formulations, the information from "About Equal" option improves the learning performance in Driver, Tosser and Fetch tasks, whereas its effect is very small in LDS.

Optimal Stopping under Query-Independent Costs

To investigate optimal stopping performance under query-independent costs, we defined the cost function as $c(Q) = \epsilon$, which just balances the trade-off between the number of

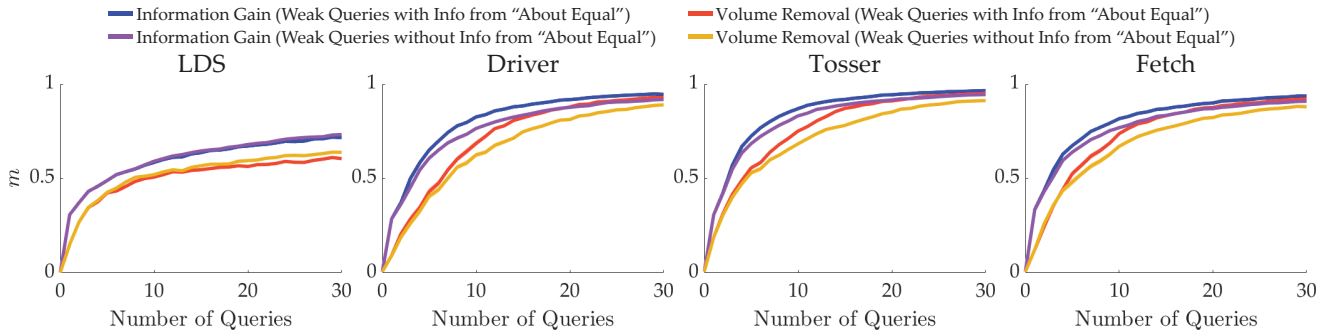

Figure 16. The results (mean±s.e.) of the simulations with weak preference queries where we use the information from "About Equal" responses (blue and red lines) and where we don't use (purple and orange lines).

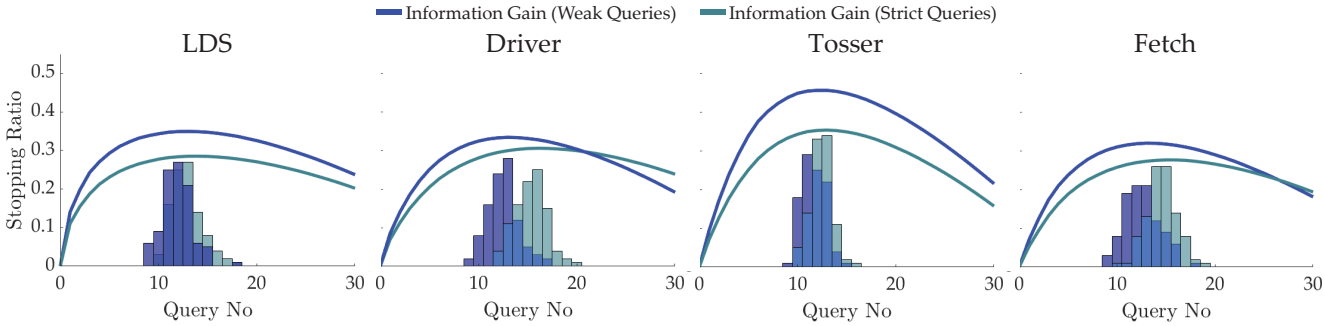

Figure 17. Simulation results for optimal stopping under query-independent costs. Line plots show cumulative active learning rewards (cumulative difference between the information gain values and the query costs), averaged over 100 test runs and scaled for better appearance. Histograms show when optimal stopping condition is satisfied.

questions and learning performance. Similar to the query-dependent costs case we described in the Experiments section, we first simulate 100 random users and tune ϵ accordingly in the same way. We then use this tuned ϵ for our tests with 100 different random users. Figure 17 shows the results. Optimal stopping rule enables terminating the process with near-optimal cumulative active learning rewards in all environments, which again supports **H9**.