Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Robots That Know What to Ask: Recovering Misaligned Rewards through Targeted Explanations

Helena Merker, Nick Walker, Andreea Bobu
Massachusetts Institute of Technology
{hmerker, nswalker, abobu}@mit.edu

***Abstract*—Learning reward functions from demonstrations assumes that demonstrations provide adequate supervision over all *features*—or task-relevant aspects of behavior. In practice, demonstrations are often imperfect: humans may underemphasize certain features due to cognitive load or physical difficulty, or the training regime may fail to sufficiently cover all relevant situations. In either case, important features may be underspecified, leading to ambiguity in the learned reward function and misaligned behavior at deployment. We propose a framework that detects such underspecified features and actively solicits targeted corrective demonstrations. Our key insight is that demonstrations implicitly reveal which features are well specified: features that are consistently optimized show little variation across demonstrations, while features that are underspecified vary widely. We leverage this statistical signal to infer which features may have been insufficiently demonstrated. The robot then explains which features it is uncertain about in natural language and queries for demonstrations that explicitly address the identified gaps. We evaluate our approach in a simulated tabletop manipulation domain and in a user study with a real Franka robot. Targeted, explanation-guided queries significantly improve reward recovery compared to random querying and passive data collection, reducing ambiguity that would otherwise persist in learning from imperfect demonstrations.**

I. Introduction

Imagine teaching a robot how to carry a coffee cup across a cluttered table. As you demonstrate the task, you focus on keeping the cup upright and avoiding collisions, but your trajectory passes at varying distances from a nearby laptop. Although maintaining a safe distance from the laptop *is* important, consistently controlling cup orientation, collision avoidance, and object proximity all at once is difficult. When the robot is later deployed, it reliably keeps the cup upright but passes uncomfortably close to the laptop. From the demonstrations alone, the robot cannot tell whether proximity to the laptop was unimportant, or whether the demonstrations simply failed to convey how that feature should be handled.

This illustrates a fundamental challenge in learning from demonstrations: observed behavior alone cannot distinguish whether a feature varies because the user is genuinely indifferent, or simply because they didn't emphasize it sufficiently. Such underspecification can happen for many reasons. Demonstrating a task is cognitively demanding [2], and humans may not always attend to every relevant feature with equal care [4]. Even when they try, the training regime itself may lack sufficient coverage [31]: certain features may remain constant, rare, or difficult to exercise (e.g., obstacles positioned at the edges of the workspace), making it impossible for demonstrators to convey how they should be handled. In both cases, demonstrations underspecify parts of the reward function, creating ambiguity that cannot be resolved from the data alone.

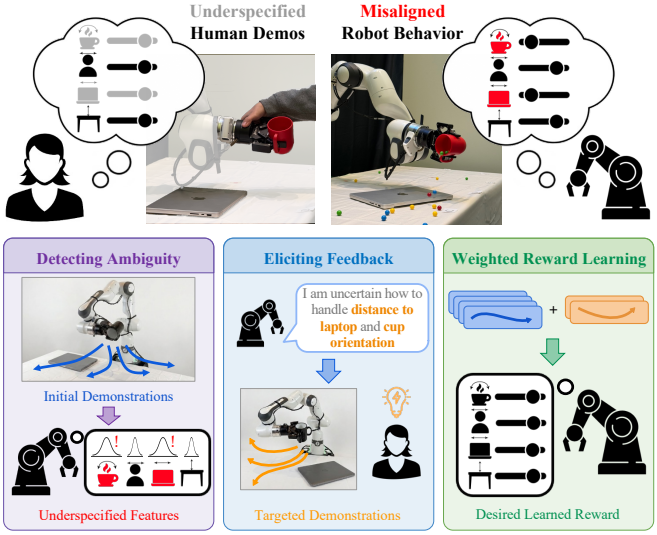


Fig. 1. Humans don't or can't always attend to all important task features when providing demonstrations (top left), which can lead to a learner inferring a misaligned reward (top right). Our method resolves this challenge by identifying when features are likely underspecified and asking for targeted, explanation-guided demonstrations (bottom).

Despite this ambiguity, current approaches implicitly assume that demonstrations provide complete supervision over all task-relevant features, and that any behavior which is inconsistent across demonstrations is evidence of low reward weight [41, 18]. When that assumption fails, the learned objective may not capture all the considerations that a human expects the robot to account for, leading to suboptimal or unsafe behavior at deployment (e.g., a robot that optimizes for efficiency while ignoring proximity to fragile objects or people) [10].

In this work, we introduce Ambiguity-Sensitive Querying for Reward Learning (ASQ), a framework that resolves this reward ambiguity by enabling the robot to diagnose which features are underspecified and to solicit targeted corrective demonstrations. Our key observation is that optimization leaves a statistical footprint in the variability of feature values across demonstrations. Features that demonstrators actively

optimize tend to show low variability across demonstrations, whereas underspecified features tend to vary widely. By modeling the distribution of feature variability under attended and unattended conditions, we can identify which features were likely underspecified during demonstration.

Since these features may have been deliberately ignored, our method queries users to distinguish underspecification from genuine indifference. The robot provides a natural language explanation that identifies the features that it is uncertain about, directing the demonstrator's attention to the specific dimensions where further supervision may be needed. We then instruct the user to provide additional demonstrations while explicitly considering how the robot should handle the identified features. By directing the user's attention to specific features, the resulting demonstrations vary in ways that reflect how those features should influence behavior, resolving ambiguity in the learned reward. We empirically validate our framework across a simulated continuous 7DoF robotic manipulation task, as well as in a user study with a physical Franka robot. We show that targeted queries recover accurate reward functions, and that real users provide more informative corrective demonstrations when given feature-based explanations compared to no guidance or simply observing robot behavior.

II. Related Work

A. Learning from Human Input

Inferring a policy or reward function from human inputs, such as demonstrations [41, 30, 18], comparisons [17], corrections [3, 9], and trajectory rankings [13], has proven effective for learning for robot tasks [35]. Inverse reinforcement learning methods typically rely upon a set of predefined features that capture task-relevant aspects of the environment [1]. Prior work has explored various forms of corrective human feedback, including natural language [38], physical interaction [4], and teleoperation [34]. Even when task-relevant features are available, users may not cleanly separate the intended signal for each feature, making it difficult to determine how each feature should influence behavior [4]. ASQ detects and resolves such ambiguity by identifying which features received inadequate supervision and eliciting targeted feedback.

B. Learning from Incomplete or Imperfect Demonstrations

Inverse reinforcement learning methods often assume that the demonstrator is at least noisy-rational with respect to some underlying reward [41]. In practice, demonstrations can be inconsistent or lack coverage of the state space [15, 37, 8]. Prior work attempts to address this issue by collecting additional feedback modalities, such as pairwise preferences [12] or natural language instructions [24], modeling demonstrator suboptimality [7], or explicitly quantifying objective misspecification [10]. These approaches improve learning from fixed data but do not actively diagnose which features are underspecified. Peng et al. [31] come closest, proposing a human-in-the-loop framework that diagnoses policy failures and solicits corrections, but at the level of state-space coverage rather than feature supervision. ASQ detects sub-optimality and actively queries for additional demonstrations targeting specific features, creating heterogeneous demonstrations that we can weight distinctly to achieve more efficient learning.

C. Queries and Explanations in Human-Robot Teaching

Prior work has characterized the types of queries a learner can pose, including label, demonstration, and feature queries, along with how naturally humans answer each [16]. Algorithmic approaches actively select queries that reduce reward uncertainty, such as pairwise trajectory comparisons [36], or choose informative behaviors that help users build a mental model of the robot's objective [23]. A complementary line of work focuses on communicating what the robot has already learned to the user: counterfactual demonstrations help users isolate task-relevant state concepts [31], sampled policy rollouts explain the inferred reward [20], and expressive motion can convey both the robot's intent and its inability to succeed [27], across visual, haptic, and auditory channels [21]. ASQ identifies features that need additional supervision, which may be used as a basis to generate queries expressed in any modality.

III. Problem Definition

We address the challenge of inferring a human's latent preferences from a limited set of demonstrations. We formalize the environment as a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \langle S, A, T, R \rangle$, where S is the state space, A is the action space, T is the transition dynamics, and R is the reward function. The MDP is solved by computing a policy $\pi : S \to A$, a mapping that prescribes which action the robot should execute in each state. Under the inverse reinforcement learning (IRL) formulation, we assume the robot does not know the reward function and must instead attempt to infer it from the human's input about the task (e.g., demonstrations). A key challenge is that demonstrations (especially when given by non-experts) may be imperfect: important aspects of the task may receive insufficient attention due to cognitive constraints, physical difficulty, or limited coverage in the training regime.

Reward Representation. We adopt a linear reward parameterization over state features. Let $\phi : S \to \mathbb{R}^k$ denote a feature mapping from states to a k-dimensional feature vector, with $\theta \in \mathbb{R}^k$ representing the reward weights. The reward at a given state is $R(s) = \theta^\mathsf{T}\phi(s)$. A trajectory $\tau = (s^0, ..., s^T)$ belongs to the space of all trajectories Ξ. The cumulative reward for a trajectory is obtained by aggregating over states:

$$R(\tau) = \sum_{s^t \in \tau} \theta^\mathsf{T}\phi(s^t) = \theta^\mathsf{T}\phi(\tau) \tag{1}$$

where $\phi(\tau) = \sum_{s^t \in \tau} \phi(s^t)$ accumulates features along the trajectory. We denote the human's true preferences as $\theta^* \in \mathbb{R}^k$.

The choice of features ϕ is critical to the expressiveness of learnable reward functions. In practice, these features can be hand-crafted [41], learned [11], or even LLM-generated [33]. Recent progress in LLMs in particular has made it possible

to easily extract task-relevant features from high-level descriptions of the task. Throughout this work, we assume access to such a candidate set $\phi = \{\phi_i\}_{i=1}^k$, and we study how uneven supervision over these features affects reward inference.

Modeling Demonstrations. We model the demonstrations as arising from a Boltzmann noisily-rational process [25, 40, 5], where the human tends to choose trajectories in proportion to their exponentiated reward. In classic formulations [19, 9, 10], a single inverse-temperature parameter β captures how rationally the human optimizes their intended reward function: $P(\tau \mid \theta^*, \beta) \propto \exp\left(\beta \cdot \theta^{*T} \phi(\tau)\right)$. This assumes uniform rationality across all features, which may not hold when demonstrators have limited attention, face varying task difficulty, or when the training regime provides insufficient coverage to adequately exercise certain features. We generalize this to a per-feature rationality vector $\beta \in \mathbb{R}^k_{\geq 0}$, where each component β_i modulates how consistently the demonstrations attend to feature ϕ_i. The probability of generating a trajectory τ is given by:

$$P(\tau \mid \theta^*, \beta) \propto \exp\left(\sum_{i=1}^{k} \beta_i \theta_i^* \phi_i(\tau)\right) \tag{2}$$

Here, β_i represents the human's attention or "care" for feature ϕ_i: higher values indicate the human more reliably produces behavior consistent with that feature, while lower values indicate greater variability–the demonstrations provide weaker signal about that feature's role in the reward.

The Underspecification Problem. Demonstrations may provide uneven supervision across different aspects of a task. Formally, this occurs when the rationality vector β has $\beta_i \approx 0$ for one or more features, meaning that certain features receive insufficient attention during demonstration. Such underspecification can arise when the demonstrated behavior involves several objectives, when the user is demonstrating under increased cognitive load, or when the configuration of the training environment makes it difficult to exercise a particular feature. This uneven supervision creates a fundamental identifiability problem. When $\beta_i \approx 0$, the demonstrations are uninformative about θ_i^*: any value of the true weight produces the same likelihood for the observed data. The demonstrations alone cannot disambiguate whether a feature was deliberately deprioritized by the human ($\theta_i^* \approx 0$) or simply not attended to during the demonstration process ($\beta_i \approx 0$).

Given initial demonstrations $D_{\text{init}} \sim P(\tau \mid \theta^*, \beta)$, our goal is to identify which features received insufficient supervision, then elicit targeted corrective demonstrations that specifically address these features. We generate natural language explanations that make the robot's inferred preferences explicit. These explanations prompt demonstrators to provide corrective demonstrations D_{extra} that address the underspecified features, enabling recovery of the true weights θ^* through targeted human feedback.

Algorithm 1: Ambiguity-Sensitive Querying (ASQ)

Input: Initial demonstrations D_{init}, task-relevant features ϕ, pre-trained reference distributions $P(\sigma_i^2 \mid o_i, \phi_i)$ for $\phi_i \in \phi$, $o_i \in \{0, 1\}$

// Detect underspecified features as in Sec. 4A

for $\phi_i \in \phi$ **do**

  $\sigma_i^2 \leftarrow \text{Var}(\{\phi_i(\tau) : \tau \in D_{\text{init}}\})$

for $\phi_{\text{hyp}} \subseteq \phi$ **do**

  $P(\phi_{\text{hyp}} \mid \{\sigma_i^2\}) \leftarrow$ ComputePosterior$(\phi_{\text{hyp}}, \{\sigma_i^2\}, \{P(\sigma_i^2 \mid o_i, \phi_i)\})$

$\phi_{\text{under}} \leftarrow \arg\max_{\phi_{\text{hyp}}} P(\phi_{\text{hyp}} \mid \{\sigma_i^2\})$

// Elicit targeted feedback as in Sec. 4B

$\texttt{query} \leftarrow \text{GenerateExplanation}(\phi_{\text{under}})$

$D_{\text{extra}} \leftarrow \text{QueryDemonstrator}(\texttt{query})$

// Learn human's preferences as in Sec. 4C

$D_{\text{total}} \leftarrow D_{\text{init}} \cup D_{\text{extra}}$

$\beta^{\text{init}}, \beta^{\text{extra}} \leftarrow \text{AssignRationalityWeights}(\phi_{\text{under}}, \phi)$

$\theta^* \leftarrow \text{WeightedMaxEntIRL}(D_{\text{total}}, \beta^{\text{init}}, \beta^{\text{extra}})$

Output: Learned reward weights θ^*

IV. Method

Our approach detects features for which the demonstrations provide weak supervision, then queries the human to clarify their preferences over these features. We leverage a key insight: features that are actively optimized exhibit low variance across demonstrations, while underspecified features vary widely. We formalize this intuition through Bayesian model selection given observed variances, calibrated against reference distributions of what variance looks like under different optimization conditions. Having identified likely underspecified features, we query the demonstrator about them using natural language explanations. This targeted intervention enables us to distinguish whether high variance reflected genuine indifference to a feature or merely incidental underspecification during the initial demonstrations. We then combine the original and targeted demonstrations, weighting each set according to which features received attention, to recover reward weights that reflect the human's complete preferences. We present the full procedure in Algorithm 1.

A. Detecting Underspecified Features

We begin by identifying features for which the demonstrations provide weak supervision. Under our demonstration model, feature-level optimization leaves a statistical footprint in the variability of observed feature values across demonstrations. Features that are consistently optimized tend to exhibit low variance, while features that receive little attention vary more widely. Concretely, if a demonstrator reliably attends to keeping a robot far from an obstacle, the corresponding feature values will cluster tightly across trajectories. Conversely, if obstacle distance was not a focus—either because the human deemed it unimportant (θ_i^* low) or simply did not think to vary it informatively (β_i low)—we expect to see greater spread in the observed feature values. We treat high variance as a signal

that a feature may have been underspecified and therefore requires clarification from the user.

Inferring Optimization Status. Our goal is to identify which features were not consistently optimized in the demonstrations and are therefore potentially underspecified. We treat this as a latent inference problem: given observed feature variability, we infer which features the demonstrator did not reliably attend to. To formalize this notion, we introduce a binary indicator $o_i \in \{0, 1\}$ for each feature ϕ_i, where $o_i = 1$ denotes that feature ϕ_i was consistently optimized in the demonstrations and $o_i = 0$ denotes otherwise. Throughout this section, we use the terms "non-optimized" and "underspecified" interchangeably to refer to features with $o_i = 0$.

Given initial demonstrations $D_{\text{init}} = \{\tau_1, \ldots, \tau_n\}$ and the feature set $\phi = \{\phi_i\}_{i=1}^k$, we compute the empirical variance for each feature across demonstrations: $\sigma_i^2 = \frac{1}{n-1}\sum_{j=1}^n (\phi_i(\tau_j) - \bar{\phi}_i)^2$, where $\bar{\phi}_i = \frac{1}{n}\sum_{j=1}^n \phi_i(\tau_j)$ is the sample mean. As discussed above, high variance indicates that a feature may not have been consistently optimized, while low variance suggests reliable attention during demonstration.

We frame detection as a Bayesian model selection problem. Each hypothesis $\phi_{\text{hyp}} \subseteq \phi$ represents a candidate set of underspecified features. Given the observed variances, we compute the posterior probability of each hypothesis:

$$P(\phi_{\text{hyp}} \mid \{\sigma_i^2\}) \propto P(\{\sigma_i^2\} \mid \phi_{\text{hyp}}) \cdot P(\phi_{\text{hyp}}) \ . \tag{3}$$

The likelihood $P(\{\sigma_i^2\} \mid \phi_{\text{hyp}})$ captures how well the observed variance pattern matches what we would expect if exactly the features in ϕ_{hyp} were non-optimized. Assuming conditional independence of feature variances given the optimization status, the likelihood factorizes as:

$$P(\{\sigma_i^2\} \mid \phi_{\text{hyp}}) = \prod_{i \in \phi} P(\sigma_i^2 \mid o_i, \phi_i) \ , \tag{4}$$

where $o_i = 0$ if $\phi_i \in \phi_{\text{hyp}}$ and $o_i = 1$ otherwise. The individual likelihoods $P(\sigma_i^2 \mid o_i, \phi_i)$ are obtained from reference distributions that characterize feature variance under optimized and non-optimized conditions; we describe how we construct these distributions below. We select the set of likely underspecified features ϕ_{under} via maximum a posteriori inference: $\phi_{\text{under}} = \arg\max_{\phi_{\text{hyp}}} P(\phi_{\text{hyp}} \mid \{\sigma_i^2\})$.

Filtering to Task-Relevant Features. Before analyzing feature variability, we restrict the robot's attention to a candidate set of task-relevant features. Irrelevant features—those with $\theta_i^* \approx 0$ regardless of demonstrator attention—will naturally exhibit high variance, but querying about them wastes the human's limited attention. Presenting a user with many candidate features when only a few actually matter imposes unnecessary cognitive load and degrades feedback quality [2, 39]. By filtering to task-relevant features, we focus the clarification interaction on dimensions where human input will be most valuable. Moreover, Bayesian model selection over feature subsets scales exponentially in $|\phi|$, making it essential to limit the hypothesis space to a small, task-relevant set.

In practice, the task-relevant feature set can be identified using domain knowledge or inferred from an LLM that rates feature relevance given a task description [32]. In this work, we use an LLM to provide a prior over feature subsets, effectively approximating $P(\phi_{\text{hyp}})$. This leverages the fact that LLMs encode rich common-sense priors about task structure and human-relevant considerations [14]. By concentrating probability mass on hypotheses involving features a human is likely to care about, this prior reduces false positives and focuses clarification on meaningful dimensions.

Constructing Reference Distributions. The raw variance σ_i^2 is difficult to interpret in isolation: what constitutes "high" variance for one feature may be typical for another, depending on the feature's natural scale, environment dynamics, and inherent task variability. To calibrate our expectations, we construct reference distributions that characterize variance under different optimization conditions.

Specifically, for each feature $\phi_i \in \phi$, we seek to estimate $P(\sigma_i^2 \mid o_i = 1, \phi_i)$ and $P(\sigma_i^2 \mid o_i = 0, \phi_i)$. We build these distributions by simulating trajectories under policies optimizing different rewards R_θ to ensure diverse human intents. For each policy, some features are assigned positive weight (and thus optimized) while others are assigned zero weight (and thus ignored). By aggregating variance observations across policies that include ϕ_i in their objective versus those that exclude it, we obtain empirical samples from each reference distribution. Although individual feature values are not normally distributed, we empirically find that their sample variances are approximately Chi-squared distributed, which facilitates subsequent likelihood computation.

B. Eliciting Targeted Feedback

Having identified features for which the demonstrations provide weak supervision, we query the demonstrator to resolve the ambiguity. This query constitutes a causal intervention: by explicitly directing the user's attention to specific features, we can distinguish whether low optimization reflected genuine indifference or incidental underspecification. Specifically, we generate a natural language explanation that communicates which features the robot is uncertain about. The explanation follows a fixed template that describes the underspecified features in terms the demonstrator can act upon (e.g., "I am uncertain about how to handle distance to the laptop"). After presenting this explanation, we request additional demonstrations with the instruction to perform the task while considering how the robot should handle the identified features. We then collect M additional demonstrations, each of which attends to the previously underspecified features ϕ_{under}.

C. Reward Learning

Standard IRL typically assumes that all demonstrations are generated under a single observation model, treating them as equally informative when inferring the reward weights [41, 18]. Under this formulation, we would naively pool initial and targeted demonstrations into a single dataset $D_{\text{total}} = D_{\text{init}} \cup D_{\text{extra}}$ and recover θ^* by maximizing their log-likelihood:

$$\theta^* = \arg\max_\theta \sum_{\tau \in D_{\text{total}}} \log P(\tau \mid \theta, \beta) \ , \tag{5}$$

where $P(\tau \mid \theta, \beta)$ could be instantiated as the Boltzmann noisily-rational model in Eq. 2, with either a fixed rationality parameter [29] or one jointly estimated with θ [10].

This formulation implicitly assumes that all demonstrations reflect the same pattern of attention across features—i.e., that they are generated using the same rationality vector β. In our setting, however, this assumption is violated by construction. The initial demonstrations D_{init} are collected without explicit guidance, and may therefore reflect uneven attention across features, while the targeted demonstrations D_{extra} are collected after *explicitly* directing the demonstrator's attention to a subset of previously underspecified features. As a result, the two demonstration sets arise from different attention patterns and should not be modeled using the same rationality vector.

To capture this distinction, we model the initial and targeted demonstrations using separate rationality vectors, β^{init} and β^{extra}, while assuming a shared underlying reward function parameterization θ^*. Intuitively, β^{init} reflects the features the demonstrator naturally attended to during unguided demonstration, whereas β^{extra} reflects the features they were explicitly instructed to attend to during targeted feedback.

Since we explicitly asked the demonstrator to attend to the features in ϕ_{under}, we assign $\beta_i^{\text{extra}} = \beta_{\text{high}}$ for $\phi_i \in \phi_{\text{under}}$, and low rationality β_{low} for the remaining features. Conversely, for the initial demonstrations, we reflect their inferred lack of attention by setting $\beta_i^{\text{init}} = \beta_{\text{low}}$ for $\phi_i \in \phi_{\text{under}}$ and $\beta_i^{\text{init}} = \beta_{\text{high}}$ for all other features. In our implementation, we estimate β_{low} and β_{high} via hyperparameter tuning on a held-out validation set with simulated human feedback.

To solve Eq. 5, one can minimize the negative log-likelihood with an off-the-shelf optimizer. Under standard IRL, the negative log-likelihood loss for a single demonstration set D is

$$\mathcal{L}(D;\theta,\beta) = \frac{1}{|D|}\sum_{\tau\in D}\left[-\sum_{i=1}^{k}\beta_i\theta_i\phi_i(\tau)\right] + \log Z(\theta,\beta) \ , \tag{6}$$

where Z is the partition function that normalizes the observation model in Eq. 2.

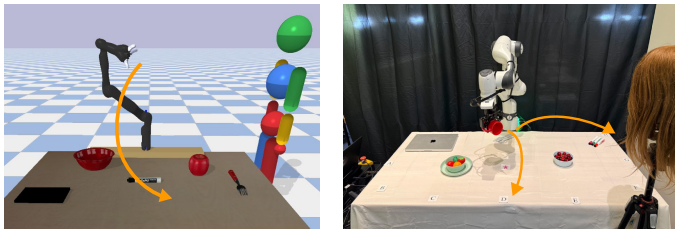
In our setting, the initial and targeted demonstrations are generated under different attention regimes and are therefore modeled with different rationality vectors. We therefore define the loss as the sum of the negative log-likelihoods of the two demonstration sets:

$$\mathcal{L}_{\text{total}} = \frac{|D_{\text{init}}|\cdot\mathcal{L}_{\text{init}} + |D_{\text{extra}}|\cdot\mathcal{L}_{\text{extra}}}{|D_{\text{init}}| + |D_{\text{extra}}|} \tag{7}$$

$$\mathcal{L}_{\text{init}} = \frac{1}{|D_{\text{init}}|}\sum_{\tau\in D_{\text{init}}} -\sum_{i=1}^{k}\beta_i^{\text{init}}\theta_i\phi_i(\tau) + \log Z(\theta,\beta^{\text{init}}) \tag{8}$$

$$\mathcal{L}_{\text{extra}} = \frac{1}{|D_{\text{extra}}|}\sum_{\tau\in D_{\text{extra}}} -\sum_{i=1}^{k}\beta_i^{\text{extra}}\theta_i\phi_i(\tau) + \log Z(\theta,\beta^{\text{extra}}) \tag{9}$$

where $Z(\theta,\beta) = \int \exp\left(\sum_{i=1}^{k}\beta_i\theta_i\phi_i(\tau')\right)d\tau'$ is approximated via importance sampling as in prior work [18]. The combined demonstration set D_{total} contains informative signal about all task-relevant features. The initial demonstrations D_{init} provide supervision for features the user naturally attended to during demonstration, while the targeted demonstrations D_{extra} fill in the gaps for the features that were initially underspecified. This enables recovery of reward weights that reflect the human's complete preferences, rather than only those objectives that happened to receive adequate supervision in the original demonstration set.



(a) (b)

Fig. 2. Experimental environments. (a) JacoRobot simulated environment, where a 7-DoF Jaco arm manipulates a coffee cup across a cluttered tabletop. (b) User study setup with a real Franka Emika FR3 robot arm in a corresponding tabletop manipulation task.

V. Simulated Experiments

We first evaluate our method with simulated human feedback. In simulation, we demonstrate that targeted querying for additional demonstrations on predicted underspecified features improves reward learning. We also show that our framework scales to settings with extraneous, task-irrelevant features.

A. Experimental Setup

We evaluate our approach in a continuous robotic manipulation task where a 7-DoF Jaco arm carries a coffee cup across a cluttered tabletop. This *JacoRobot* environment consists of a robot arm manipulating a coffee cup using the PyBullet simulator (Figure 2a). The environment includes several objects positioned on or near the table: a laptop, human, apple, bowl, fork, and marker. We randomly sample start-goal pairs and smoothly perturb the shortest-path trajectories. Trajectories consist of 21 waypoints, each with 109 dimensions: the robot's joint configuration, xyz positions of all joints and objects, and their rotation matrices. We define four primary features: 1) *laptop*: xy-distance of the end-effector to the laptop, 2) *table*: z-distance of the end-effector above the table, 3) *human*: xy-distance of the end-effector to the human, and 4) *coffee*: length of the end effector's up axis projected onto the world up axis, a measure of how upright the coffee cup is. The robot should avoid the laptop and human, remain close to the table, and ensure that the coffee cup does not spill. For experiments with extraneous objects, we additionally define distractor features: xy-distances to the 5) apple, 6) bowl, 7) fork and 8) marker.

We generate 50000 trajectories across 5000 random start-goal pairs (10 randomly perturbed trajectories per pair). The trajectory pool is split into 60% training, 20% validation, and 20% test sets. Simulated human demonstrators are modeled as Boltzmann-rational agents with ground-truth preference

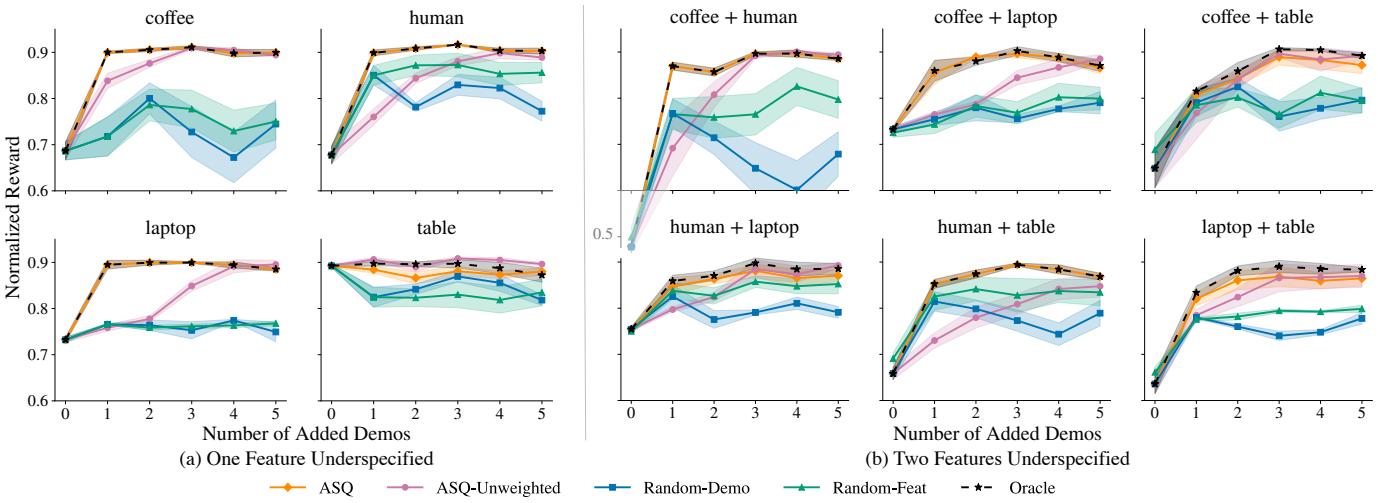


Fig. 3. Reward recovery in *JacoRobot* when the initial demonstrations underspecify (a) a single feature or (b) two features. ASQ consistently outperforms the random baselines and approaches Oracle performance, recovering the true reward with fewer targeted demonstrations. Lines show mean normalized reward across 5 seeds and shaded regions denote standard error.

weights θ^* and per-feature rationality coefficients β. Setting β_i to a high value produces demonstrations that reliably optimize ϕ_i, while setting β_i to zero simulates underspecification. We note that this is a conservative model of human behavior; real demonstrators may de-prioritize rather than entirely ignore non-targeted features.

Methods. We compare five approaches. 1) **ASQ** uses variance-based detection to identify the most likely underspecified features, requests demonstrations targeting those features, and applies demonstration-specific weights during reward learning. 2) **ASQ-Unweighted** ablates the rationality weighting, applying uniform weights across all demonstrations, isolating the contribution of the weighting scheme from the querying strategy. 3) **Random-Demo** requests each additional demonstration for N randomly selected features. 4) **Random-Feat** selects N random features at the start, and then all additional demonstrations target those same features. 5) **Oracle** uses the ground-truth underspecified features instead of the variance-based predictions. In all conditions, the simulated human provides 10 initial demonstrations with N features underspecified, followed by zero to five additional targeted demonstrations.

For evaluation, we select the top-k trajectories from the test set with the highest learned reward, compute their average ground-truth reward, and normalize by the range of ground-truth rewards across all test trajectories. We use $k = 20$ for all experiments. All experiments are repeated across 5 seeds, and we report mean normalized reward with standard error.

LLM-Based Filtering. Real-world robotic environments often contain features that are irrelevant to the current task. To this end, we expand the JacoRobot feature set to include four additional objects. The four original features ϕ_{task} (laptop, table, human, coffee) remain task-relevant, while the four new features ϕ_{extra} (apple, bowl, fork, and marker) serve as distractors. In this 8-feature setting, we query a large language model (GPT-5.2) with the task description and the full feature list, asking it to identify which features a typical human would consider necessary for safe and successful task completion. The LLM returns a binary relevance judgment for each feature, providing a filtered feature set $\phi \subseteq \phi_{\text{task}} \cup \phi_{\text{extra}}$. Features judged as irrelevant are excluded from subsequent detection, reducing the hypothesis space and preventing false positives on distractor features. To strengthen the random baselines, we add the following constraint. If all of the randomly chosen features are from ϕ_{extra}, then the additional demonstrations reflect the suboptimal demonstrator's initial behavior. We assume that, if a human was asked to provide demonstrations about only irrelevant features, that they would repeat their original behavior rather than optimize for features for which they have no preferences.

B. Results

Task-Relevant Features. We first evaluate settings where all features in the environment are task-relevant, eliminating the need for LLM-based filtering. The first condition that we consider is where exactly one feature is underspecified during initial demonstrations ($\beta_i = 0$ for underspecified ϕ_i). We simulate a human providing targeted corrective demonstrations for the requested features ϕ_{under} by setting $\beta_j = 0$ for all $\phi_j \notin \phi_{\text{under}}$. In the real world, it is possible that a human demonstrator may actually provide more informative targeted demonstrations. Specifically, a real-life demonstrator may instead just de-prioritize, rather than ignore, the features optimized for in the initial demonstrations. As shown in Figure 3a, ASQ outperforms the random baselines in all four conditions. When coffee, human, or laptop is underspecified, ASQ recovers reward more quickly and achieves higher final performance. The table feature, however, shows that underspecification is not synonymous with suboptimality. When a demonstrator attends to three features, the resulting trajectories could happen to also perform well on the fourth feature, leading to a high normalized reward at the start. Secondly, we

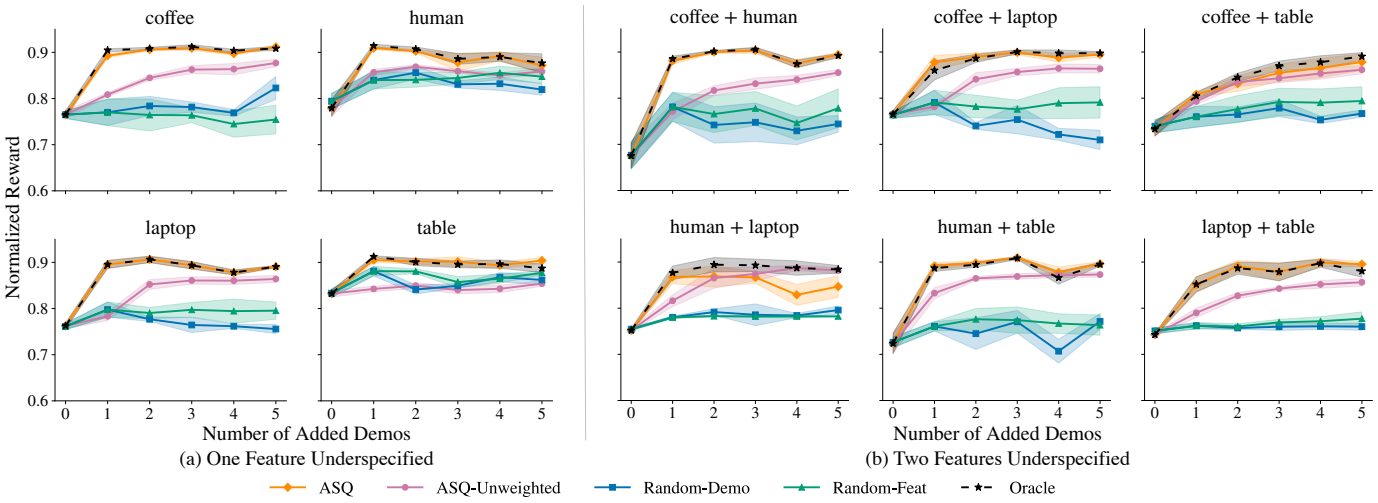


Fig. 4. Reward recovery in *JacoRobot* in the 8-feature setting that augments the feature set of 4 task-relevant features with 4 irrelevant distractor objects, with (a) one task-relevant feature underspecified and (b) two task-relevant features underspecified. LLM-based filtering prunes the distractors before variance-based detection, allowing ASQ to focus queries on task-relevant features. ASQ matches or exceeds the random baselines across all single-feature and pairwise comparison conditions, demonstrating robustness to extraneous environmental features. Lines show mean normalized reward across 5 seeds and shaded regions denote standard error.

evaluate all six pairwise combinations of two underspecified features. We show in Figure 3b that ASQ's advantage persists when two features are simultaneously underspecified, although the gain over the baselines varies depending upon the feature combination.

Scaling to Extraneous Features. In the 8-feature setting, we evaluate whether LLM-based filtering enables our method to maintain robust performance despite the presence of task-irrelevant distractors. For these experiments, we simulate the demonstrator to have $\theta_i = 0$ for $\phi_i \in \phi_{\text{extra}}$, reflecting that the user's preferences assign no weight to task-irrelevant features. We show in Figure 4a that ASQ outperforms the baselines in the single-feature setting. Compared to the baselines, ASQ improves reward learning more rapidly in the early demonstrations, reducing the number of corrective demonstrations required from the demonstrator for reward recovery. When two features are underspecified, ASQ also shows an advantage for all pairwise combinations (Figure 4b). These results demonstrate the effectiveness of the method in settings where robots must distinguish between meaningful task features and irrelevant environmental distractors.

VI. User Study

To evaluate whether feature-based explanations help humans provide more informative corrective demonstrations, we conduct a user study comparing our explanation-guided approach against natural alternatives.

A. Experiment Design

Our primary investigation concerns whether explanations that identify underspecified features in interpretable terms lead to better corrective demonstrations than simply observing the robot's behavior. While simulation experiments can evaluate whether targeted demonstrations improve reward learning, they cannot capture whether real users can translate different forms of feedback into actionable guidance. A robot might convey its limitations by rolling out its current policy, allowing the user to observe failures directly. Alternatively, it might provide an explicit explanation naming the features it misunderstands. We hypothesize that the latter produces demonstrations more useful for reward recovery. Rather than leaving users to diagnose the problem themselves, explanations direct attention to the specific features requiring supervision.

We conducted a user study under an IRB-approved protocol using a real Franka Emika FR3 robot arm in a tabletop setting. The setup, shown in Figure 2b, preserved the manipulation task from our simulation experiments while grounding the interaction in a realistic physical context. The study followed a within-subjects design that varied feedback requests at three levels. 1) **Unguided**: the robot requested demonstrations with no additional context. 2) **Rollout**: the robot first executed a trajectory according to its currently learned reward function, then requested demonstrations. 3) **Explanation**: the robot provided a natural language explanation identifying the feature it was uncertain about, then requested demonstrations. The explanation followed a fixed template that named the underspecified feature directly.

Participants were told all four task objectives upfront (keeping the mug upright, close to the table, away from the laptop, and away from people) and reminded of them before each condition (Appendix D). Ensuring participants cared about all task-relevant features isolates the effect of attentional guidance on demonstration quality, letting us investigate which querying method best directs attention to the underspecified dimension and yields more informative corrective data. Each participant completed all three conditions. We counterbalanced condition order by assigning participants evenly across the six possible orderings. Within each condition, participants first completed

a familiarization phase to become comfortable with the interaction modality, followed by two experimental tasks. The familiarization phase used *coffee* as the underspecified feature: participants stood in the demonstrator position while the robot queried according to the assigned condition, after which they provided three practice demonstrations. The first experimental task targeted *human* as the underspecified feature, while the second task used *laptop* as the underspecified feature. For each task, participants provided three demonstrations. After completing both tasks within a condition, participants filled out a survey about their experience with that method before proceeding to the next condition. A final comparative survey collected overall preferences across methods.

Manipulated Variables. We compare the type of feedback provided before demonstration collection: no feedback (*Unguided*), behavioral rollout (*Rollout*), or feature-based explanation (*Explanation*). The explanation always correctly identified the underspecified feature, enabling us to evaluate whether this form of communication helps users provide better demonstrations independent of prediction accuracy.

Hypotheses. We test three hypotheses regarding the benefits of feature-based explanations for eliciting corrective demonstrations: **H1.** Providing an explanation when querying for demonstrations leads humans to provide corrective demonstrations that yield better aligned rewards. **H2.** Explanations reduce the cognitive load of providing a corrective demonstration relative to conditions where users must base their correction on observed behavior. **H3.** Participants will subjectively prefer the *Explanation* condition over the alternative methods.

Dependent Measures. We evaluate each condition using both objective and subjective metrics. For objective evaluation, we measure normalized reward recovery: for each participant, condition, and task, we train a reward function using MaxEnt IRL on their collected demonstrations combined with a fixed set of initial demonstrations, then compute the normalized ground truth reward of trajectories selected by the learned reward function. For subjective evaluation, we administer the NASA Task Load Index (NASA-TLX) [22] questionnaire measuring cognitive workload after each condition, as well as a forced-choice preference question in the final comparative survey, asking participants to select which method they preferred overall.

Participants. We recruited $N = 12$ participants (6 female, 6 male; age $M = 28.5$, $SD = 12.5$) with moderate prior experience controlling or operating robots ($M = 4.2$, $SD = 1.5$ on a 1–7 scale).

B. Analysis

All analyses use linear mixed-effects models with participant as a random effect and condition order as a covariate to account for the within-subjects design.

H1: Reward Improvement was supported. We found a significant main effect of condition on normalized reward improvement, $F(2, 22) = 14.94$, $p < .001$. Estimated marginal means revealed that the *Explanation* condition ($M = 0.021$, $SE = 0.005$) yielded significantly greater reward improvement than both *Unguided* ($M = -0.018$, $SE = 0.005$; $t = -6.31$, $p < .001$) and *Rollout* ($M = -0.016$, $SE = 0.005$; $t = -6.06$, $p < .001$). No difference was observed between *Unguided* and *Rollout* ($p = .966$). Demonstrations elicited via feature-based explanations thus produce better reward recovery when combined with demonstration-specific rationality weighting. This weighting presumes that the human attended to the named features, an assumption that is only justified in *Explanation*, where we directed the demonstrator's attention to those features. In contrast, in *Unguided* and *Rollout*, the human may have chosen to focus on any of the four task-relevant features, so applying the same weighting scheme would attribute attention we cannot verify.

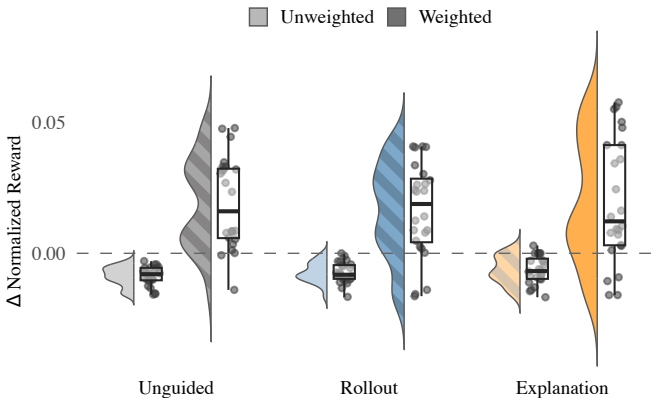


Fig. 5. Change in normalized reward by condition. Hashed indicates an ablative configuration; applying weighting to demonstrations that couldn't have been known to be targeted, or not weighting demonstrations that can be presumed to be targeted. Only the *Explanation* condition yielded positive improvement on average. To account for stochasticity in the process, reward learning was repeated 100 times and the median performance used. Since participants demonstrated two tasks, the average of the change in normalized reward is used.

To separate the effect of the explanation from the weighting it enables, we compared all-weighted and all-unweighted reward variants. Within *Explanation*, all-weighted rewards ($M = 0.021$, $SE = 0.007$) were higher than all-unweighted rewards ($M = -0.017$, $SE = 0.003$), $t(23) = 4.56$, $p < .001$, mean difference $= 0.038$. As shown by the hashed entries of Figure 5, assuming correct weighting substantially narrows the gap between *Explanation* and the other conditions. Moreover, when only all-weighted rewards were analyzed, condition differences were not significant (all $p \geq .183$), leaving only a directional trend favoring *Explanation*. However, the demonstrations themselves did change meaningfully across conditions. Per-participant feature analysis (Appendix E) shows that under *Explanation*, variance on the features the robot did not flag widens relative to the other conditions, consistent with participants relaxing their attention on objectives that were not identified as uncertain. Participants' self-reports corroborate this shift: in the *Explanation* condition, participants overwhelmingly reported emphasizing the named feature in their demonstrations, whereas in *Rollout* they frequently misidentified the underspecified feature (Figure 6).

H2: Cognitive Workload was not supported. We found no significant effect of condition on NASA-TLX composite scores, $F(2, 22) = 0.76$, $p = .480$. Workload was comparable across *Unguided* ($M = 28.1$, $SE = 3.7$), *Rollout* ($M = 28.3$, $SE = 3.7$), and *Explanation* ($M = 25.6$, $SE = 3.7$), with no significant pairwise differences (all $p \geq .530$). Although we hypothesized that explanations would reduce cognitive load by narrowing what the demonstrator must consider, the physical demands of the task remained largely invariant across conditions: participants kinesthetically guided the same robot through the same motion for the same number of demonstrations. In this controlled setting, attentional guidance changes what the demonstrator thinks about, but not the physical or temporal effort each demonstration requires. Analysis of individual TLX subscales revealed no significant effects for any dimension (all $p > .19$). Demonstration duration also did not differ significantly across conditions, $F(2, 22) = 1.25$, $p = .307$ (Unguided: $M = 17.3$s; Rollout: $M = 14.6$s; Explanation: $M = 16.6$s). In a practical deployment, we hypothesize that the workload benefit of targeted querying would likely manifest as a reduction in the number of demonstrations required to correct the misaligned reward. Investigating whether explanation-guided querying reduces total teaching burden in less constrained settings, where the human controls when to stop providing demonstrations, is a direction for future work.

H3: Subjective Preference was not supported. Five of twelve participants preferred *Explanation*, five preferred *Rollout*, and the remaining two preferred *Unguided* (exact multinomial vs. uniform: $\chi^2 = 1.50$, $p = .616$). While preferences did not significantly favor any condition, the qualitative reasons participants gave were informative. Participants who preferred *Rollout* often described the rollout as a concrete behavior to repair rather than as an abstract representation of the robot's misunderstanding. P6 explained that it felt easier to teach the robot by providing the "smallest perturbation to an existing trajectory," while P7 noted that "seeing the path the robot thought was best" made it easier to identify and correct flaws. Participants preferring *Explanation* emphasized explicit guidance and reduced cognitive load. P4 noted that "it was nice to have specific direction about where to target the efforts of my demonstration," and P8 explained that being told "exactly what [the robot] was unsure of" made it easier to focus on one aspect of the task, rather than "trying to do several things at once." Those preferring *Unguided* cited varied reasons, including task familiarity and freedom to give natural demonstrations. P11 noted that the condition allowed them to offer the "easiest demonstrations," whereas knowing exactly what the robot was confused about required them to "think a bit more" about how to convey the correction. These responses suggest that no single feedback modality works for all users.

VII. Conclusion

We introduced ASQ, a framework that addresses a fundamental limitation of learning from demonstrations: from demonstrations alone, a robot cannot determine whether feature indifference reflects the demonstrator's true preferences or simply indicates incomplete supervision. Our insight is that optimization leaves a statistical signal in feature variability, enabling identification of underspecified features through Bayesian model selection over observed variance patterns. By generating natural language queries that direct the demonstrator's attention to these specific dimensions, ASQ resolves reward ambiguity with minimal human effort. In simulation, ASQ improves reward learning using only a small number of targeted demonstrations in a continuous manipulation domain. Our user study with a physical robot confirmed that our approach translates to real human demonstrators, showing that feature-based explanations produce more aligned reward functions compared to alternative modalities.

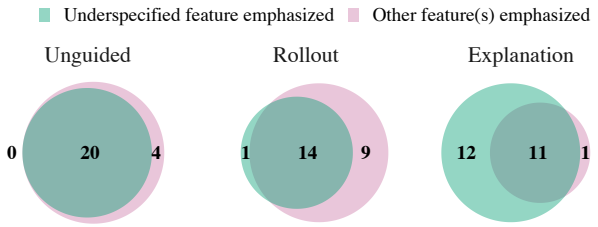


Fig. 6. Whether participants reported emphasizing the underspecified feature in their demonstrations, emphasizing other features, or both. Each of the twelve participants demonstrated two tasks in each condition.

Limitations and Future Work. Although ASQ successfully resolves reward ambiguity, several limitations remain. Our approach treats feature attention as binary and static across a demonstration session, but in practice, attention may fluctuate within a single trajectory. We also assume conditional independence of feature variances given optimization status, yet features are often correlated in practice. Beyond these modeling choices, our framework elicits feedback exclusively through kinesthetic demonstrations, but humans communicate preferences through diverse modalities. Augmenting the framework to incorporate natural language feedback from the user or comparative judgments could resolve ambiguity more efficiently. In our method, features the user is indifferent to are likely task-irrelevant and will not be queried if detected by LLM-based filtering. Nonetheless, a natural extension is a two-stage interaction in which the robot first names uncertain features and asks the user whether they matter before collecting demonstrations. Additionally, our user study involved a modest sample size, and larger-scale studies across diverse populations would strengthen confidence in our findings. Our experiments also focused on a manipulation domain with a fixed set of objects. Evaluating ASQ in more complex or dynamic environments where objects appear, disappear, or move throughout the task would further validate the robustness and broader applicability of our method.

Acknowledgments

This work was funded by a seed grant from the MIT Schwarzman College of Computing's Social and Ethical Responsibilities of Computing program.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. DOI: 10.1145/1015330.1015430.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. DOI: 10.1145/3290605.3300233.

[3] Andrea Bajcsy, Dylan P Losey, Marcia K O'malley, and Anca D Dragan. Learning robot objectives from physical human interaction. In *Conference on robot learning*, pages 217–226. PMLR, 2017. URL http://proceedings.mlr.press/v78/bajcsy17a.html.

[4] Andrea Bajcsy, Dylan P. Losey, Marcia K. O'Malley, and Anca D. Dragan. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2018, Chicago, IL, USA, March 05-08, 2018*, pages 141–149. ACM, 2018. DOI: 10.1145/3171221.3171267.

[5] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29, 2007.

[6] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. DOI: 10.18637/jss.v067.i01.

[7] Mark Beliaev and Ramtin Pedarsani. Inverse reinforcement learning by estimating expertise of demonstrators. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 15532–15540. AAAI Press, 2025. DOI: 10.1609/AAAI.V39I15.33705.

[8] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. URL http://papers.nips.cc/paper_files/paper/2023/hash/fe692980c5d9732cf153ce27947653a7-Abstract-Conference.html.

[9] A. Bobu, A. Bajcsy, J. F. Fisac, and A. D. Dragan. Learning under misspecified objective spaces. In *Conference on Robot Learning (CoRL)*, 2018. URL http://proceedings.mlr.press/v87/bobu18a.html.

[10] A. Bobu, A. Bajcsy, J. F. Fisac, S. Deglurkar, and A. D. Dragan. Quantifying hypothesis space misspecification in learning from human–robot demonstrations and physical corrections. *Transactions on Robotics (T-RO)*, 2020. DOI: 10.1109/TRO.2020.2971415.

[11] Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D Dragan. Inducing structure in reward learning by learning features. *The International Journal of Robotics Research*, 41(5):497–518, 2022. DOI: 10.1177/02783649221078031.

[12] Daniel S. Brown, Wonjoon Goo, and Scott Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 330–359. PMLR, 2019. URL http://proceedings.mlr.press/v100/brown20a.html.

[13] Daniel S. Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. Safe imitation learning via fast bayesian reward inference from preferences. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1165–1177. PMLR, 2020. URL http://proceedings.mlr.press/v119/brown20a.html.

[14] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712, 2023. DOI: 10.48550/ARXIV.2303.12712.

[15] Maya Cakmak and Andrea L. Thomaz. Optimality of human teachers for robot learners. In *2010 IEEE 9th International Conference on Development and Learning*, pages 64–69, 2010. DOI: 10.1109/DEVLRN.2010.5578865.

[16] Maya Cakmak and Andrea L. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '12, page 17–24, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450310635. DOI: 10.1145/2157689.2157693.

[17] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

[18] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, 2016. URL http://proceedings.

mlr.press/v48/finn16.html.

[19] Jaime F. Fisac, Andrea Bajcsy, Sylvia L. Herbert, David Fridovich-Keil, Steven Wang, Claire J. Tomlin, and Anca D. Dragan. Probabilistically safe robot planning with confidence-based human predictions. In *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. DOI: 10.15607/RSS.2018.XIV.069.

[20] Morris Gu, Elizabeth Croft, and Dana Kulić. Demonstration based explainable ai for learning from demonstration methods. *IEEE Robotics and Automation Letters*, 10(7): 6552–6559, 2025. DOI: 10.1109/LRA.2025.3568617.

[21] Soheil Habibian, Antonio Alvarez Valdivia, Laura H. Blumenschein, and Dylan P. Losey. A survey of communicating robot learning during human-robot interaction. *The International Journal of Robotics Research*, 44(4): 665–698, 2025. DOI: 10.1177/02783649241281369.

[22] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in Psychology*, 52:139–183, 1988. DOI: 10.1016/S0166-4115(08)62386-9.

[23] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. Enabling robots to communicate their objectives. *Auton. Robots*, 43(2):309–326, 2019. DOI: 10.1007/S10514-018-9771-0.

[24] Minyoung Hwang, Alexandra Forsey-Smerek, Nathaniel Dennler, and Andreea Bobu. Masked irl: Llm-guided reward disambiguation from demonstrations and language, 2025. URL https://arxiv.org/abs/2511.14565.

[25] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957. DOI: 10.1103/PhysRev.106.620.

[26] Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. lmertest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13): 1–26, 2017. DOI: 10.18637/jss.v082.i13.

[27] Minae Kwon, Sandy H. Huang, and Anca D. Dragan. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, page 87–95, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450349536. DOI: 10.1145/3171221.3171276.

[28] Russell V. Lenth and Julia Piaskowski. *emmeans: Estimated Marginal Means, aka Least-Squares Means*, 2025. URL https://rvlenth.github.io/emmeans/. R package version 2.0.1.

[29] Chang Liu, Jessica B. Hamrick, Jaime F. Fisac, Anca D. Dragan, J. Karl Hedrick, S. Shankar Sastry, and Thomas L. Griffiths. Goal inference improves objective and perceived performance in human-robot collaboration. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 940–948. ACM, 2016. URL http://dl.acm.org/citation.cfm?id=2937062.

[30] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 663–670. Morgan Kaufmann, 2000.

[31] Andi Peng, Aviv Netanyahu, Mark K. Ho, Tianmin Shu, Andreea Bobu, Julie Shah, and Pulkit Agrawal. Diagnosis, feedback, adaptation: A human-in-the-loop framework for test-time policy adaptation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 27630–27641. PMLR, 2023. URL https://proceedings.mlr.press/v202/peng23c.html.

[32] Andi Peng, Andreea Bobu, Belinda Z. Li, Theodore R. Sumers, Ilia Sucholutsky, Nishanth Kumar, Thomas L. Griffiths, and Julie A. Shah. Preference-conditioned language-guided abstraction. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2024, Boulder, CO, USA, March 11-15, 2024*, pages 572–581. ACM, 2024. DOI: 10.1145/3610977.3634930.

[33] Andi Peng, Belinda Z. Li, Ilia Sucholutsky, Nishanth Kumar, Julie Shah, Jacob Andreas, and Andreea Bobu. Adaptive language-guided abstraction from contrastive explanations. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 3425–3438. PMLR, 2024. URL https://proceedings.mlr.press/v270/peng25c.html.

[34] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. An autonomous dynamic camera method for effective remote teleoperation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2018, Chicago, IL, USA, March 05-08, 2018*, pages 325–333. ACM, 2018. DOI: 10.1145/3171221.3171279.

[35] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annu. Rev. Control. Robotics Auton. Syst.*, 3:297–330, 2020. DOI: 10.1146/ANNUREV-CONTROL-100819-063206.

[36] Dorsa Sadigh, Anca D. Dragan, Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. DOI: 10.15607/RSS.2017.XIII.053.

[37] Maram Sakr, Juyan Zhang, H. F. Machiel Van der Loos, Dana Kulić, and Elizabeth Croft. Consistency matters: Defining demonstration data quality metrics in robot learning from demonstration. *J. Hum.-Robot Interact.*, 15(2), December 2025. DOI: 10.1145/3773904.

[38] Pratyusha Sharma, Balakumar Sundaralingam, Valts Blukis, Chris Paxton, Tucker Hermans, Antonio Torralba, Jacob Andreas, and Dieter Fox. Correcting robot plans with natural language feedback. In *Robotics: Science and Systems XVIII, New York City, NY, USA, June 27 - July*

1, 2022, 2022. DOI: 10.15607/RSS.2022.XVIII.065.

[39] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285, 1988. ISSN 0364-0213. DOI: 10.1016/0364-0213(88)90023-7.

[40] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press Princeton, NJ, 1945.

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