Original Article

The International Journal of Robotics Research
2025, Vol. 0(0) 1–21

DOI: 10.1177/02783649251394742
journals.sagepub.com/home/ijr
Sage

Active reward learning and iterative trajectory improvement from comparative language feedback

Eisuke Hirota[1,2,*], Zhaojing Yang[1,*], Ayano Hiranaka[1], Miru Jun[1], Jeremy Tien[3], Stuart J. Russell[3], Anca Dragan[3] and Erdem Bıyık[1,4]

Abstract

Human-in-the-loop learning has gained traction in fields like robotics and natural language processing in recent years. While prior work mostly relies on human feedback in the form of preference comparisons, this feedback type has multiple limitations. It does not let users explain the reasons for their preferences and provides only a binary signal for learning, resulting in huge data inefficiency. Consequently, training robots require a substantial amount of human feedback, occupying significant time and burdening the user. To overcome these challenges, we take the insight that language is a preferable medium compared to comparisons, providing more information regarding user preferences. Thus, in this work, we aim to incorporate comparative language feedback to iteratively improve robot trajectories and learn reward functions that encode human preferences. We learn a shared latent space that integrates trajectory data and language feedback, and subsequently leverage the learned latent space to improve trajectories and learn human preferences. We finally introduce an active learning method that integrates comparative language feedback to further boost data-efficiency. Our results in simulation experiments and user studies demonstrate the effectiveness of the learned latent space and the success of our learning algorithms. Our reward learning algorithm exhibits a 23.9% improvement in subjective score on average and 11.3% higher time-efficiency compared to the preference comparison method in the user studies. Our active querying method further improves user experience featuring an 8.31% average improvement in subjective scores compared to random querying. Our code is publicly available at https://liralab.usc.edu/comparative-language-feedback/.

Keywords

reward learning, active learning, inverse reinforcement learning, preference-based learning, human–robot interaction, human-in-the-loop learning

Introduction

Learning from human feedback has gained significant popularity in robotics, leading to the study of different forms of human feedback such as demonstrations (Hoque et al., 2021; Kelly et al., 2019; Ross et al., 2011; Sadigh et al., 2016), preference comparisons (Bıyık et al., 2022; Christiano et al., 2017; Sadigh et al., 2017; Wilde et al., 2021), rankings (Myers et al., 2021), physical corrections (Bajcsy et al., 2017), interventions (Korkmaz and Bıyık, 2025), gaze or visual saliency maps (Banayeeanzade et al., 2025; Liang et al., 2024), human language (Campos and Shern, 2022; Sharma et al., 2022), etc. Among these, preference comparisons gained in popularity for its simplicity and ease of use, especially compared to demonstrations (Bıyık et al., 2022).

Preference comparisons often involve users choosing between a pair of choices. Using these selections to learn a reward function and train a policy is known as reinforcement learning from human feedback (RLHF) (Christiano et al., 2017) or more generally preference-based learning

[1]Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA
[2]School of Computing, State University of New York at Binghamton, NY, USA
[3]Department of Electrical Engineering and Computer Sciences, UC Berkeley, CA, USA
[4]Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, USA

*Equal contribution.

Corresponding author:
Eisuke Hirota, School of Computing, State University of New York at Binghamton, 4400 Vestal Parkway East, Vestal, Binghamton, NY 13902-4600, USA.
Email: ehirota13@gmail.com

(Sadigh et al., 2017; Wirth et al., 2017). It has proven applicable to a broad range of fields ranging from robotics (Bıyık et al., 2022; Lee et al., 2021; Wang et al., 2024) to natural language processing (Ouyang et al., 2022; Stiennon et al., 2020), from traffic routing (Bıyık et al., 2021) to human–computer interaction (Dennler et al., 2023).

Despite their successes, preference comparisons suffer from problems (Casper et al., 2024) such as the unreliability of human data and the limited information bandwidth, that is, each pairwise comparison contains at most 1 bit of information: whether one trajectory is preferred over another. To make up for the lack of explicit justification for a human's preference, preference comparisons must overwork humans through copious amounts of querying, overall exhibiting severe data inefficiencies. Various papers work at overcoming these pitfalls through integrating active learning methods to extract information or diversity for efficient query selection to speed up the reward learning process (Biyik et al., 2019; Bıyık et al., 2024a; Ellis et al., 2024); however, these methods still rely on weakly informative preference comparisons without any explicit justifications. There has additionally been research to provide a better interface (Basu et al., 2018), allowing the users to specify their preferences for every feature, but they require features to be hand-designed and interpretable.

As an alternative form of human feedback, we propose to use *comparative language*. This feedback is considerably more informative than preference comparisons, allowing users to prioritize specific aspects. For example, it allows users to naturally indicate their preference about speed by simply stating, "the robot should move faster," making it more intuitive and interpretable.

In this work, we aim to leverage comparative language feedback to learn the human's preferences. Through the usage of comparative language feedback, we can newly perform trajectory improvement and more effectively succeed in reward learning. We then expand this framework through integrating an active learning method that further optimizes data efficiency to speed up reward learning. We introduce and examine our active reward learning method against baselines in both simulation and real-world studies.

In pursuit of this objective of efficient learning via comparative language feedback from humans, we make three main contributions in this work*:

- We learn a shared latent space that aligns trajectories and comparative language feedback. This alignment enables the robot to comprehend the language feedback, leveraging it for adapting the robot's behavior to learn and better align with the human's preferences.
- We highlight the effectiveness of our approach through conducting experiments in two simulation environments and a human subjects study with a real robot. These results suggest that reward learning from comparative language feedback outperforms traditional preference comparisons in performance and time-efficiency, and is favored by most of the users.
- We propose an active query selection method by developing a mutual information based acquisition function for comparative language feedback and empirically compare its performance against baselines in simulation and a human subjects study with a real robot. These results demonstrate that active learning selects queries that better elicit user preferences while requiring less feedback.

Related work

Before formally defining the problem, we will first review existing works in robot learning from human feedback, preference-based learning, and active reward learning.

Learning from human feedback

Several promising approaches leverage human feedback to train robots (Bajcsy et al., 2017; Bıyık, 2025; Bıyık et al., 2022; Campos and Shern, 2022; Christiano et al., 2017; Holk et al., 2024a; Hoque et al., 2021; Kelly et al., 2019; Liang et al., 2024; Myers et al., 2021; Ross et al., 2011; Sadigh et al., 2017; Sharma et al., 2022; Spencer et al., 2022; Wilde et al., 2021). Demonstrations are one modality of feedback that have been extensively studied. Reward learning from demonstrations, also known as inverse reinforcement learning (IRL), features a user enacting an assumed–optimal trajectory (Abbeel and Ng, 2004). From this collected dataset of demonstrations, a reward function that encodes present trajectory features to reward values is learned. Users provide demonstrations often through teleoperating the robot's movement with a joystick controller (Argall et al., 2009) or virtual reality headset (Zhang et al., 2018), as such, while these demonstrations provide high-quality information for reward learning analysis, tasking a user to repeatedly perform trajectories is time-consuming, burdening, and difficult (Akgün et al., 2012; Casper et al., 2024; Christiano et al., 2017; Hong et al., 2025).

Benefiting from recent advancements in natural language processing, works have leveraged language for adjusting robot trajectories (Bucker et al., 2022, 2023; Han et al., 2024; Sharma et al., 2022; Shi et al., 2024; Yow et al., 2024), fine-tuning language models (Campos and Shern, 2022), and reward shaping (Goyal et al., 2019, 2021). For example, Shi et al. (2024) use language-conditioned behavior cloning (LCBC) for corrective language commands and improving policies. Lynch et al. (2023) describe an approach for real-time guidance from humans using natural language in order to achieve a goal. Cui et al. (2023) introduce an approach to use human language feedback to correct robot manipulation in real-time via shared autonomy. Goyal et al. (2019) utilize human language combined with past action sequences to

generate rewards. Similarly, Zhang et al. (2025) and Goyal et al. (2021) leverage natural language to map image observations to rewards. Holk et al. (2024b) extract sentiment features from natural language to gain insight on trajectory segments that users found beneficial or detrimental. However, these works focus on using human language as an instruction or correction for the robot. Prior work has yet to take advantage of the expressive nature of language and develop a method of reward learning from comparative language feedback.

Addressing this gap will provide more scalable methods for training robots, particularly in complex environments where traditional feedback modalities may be insufficient or impractical. Therefore, our work introduces a reward learning method that leverages *comparative language feedback*, with the additional capability of iteratively updating the learned reward function for better preference alignment, providing an advantage over one-shot corrections (Bucker et al., 2023; Sharma et al., 2022).

Preference-based learning

Based on preference comparison feedback, preference-based learning is widely used for its logical intuitiveness and ease of use. Traditionally, preference comparison feedback iteratively queries two trajectories and asks the user to rank them, after which a reward function is trained from this data. Further work has sought to improve preference-based learning in numerous ways. For example, learning from rankings requires users to order a batch, instead of a pair, of trajectories. This feedback benefits from a low hurdle of entry and has proven to work effectively with deep reinforcement learning (Brown et al., 2019). Furthermore, Sikchi et al. (2023) and Brown et al. (2020) integrate preference feedback into imitation learning, demonstrating higher performance in their respective approaches. Unfortunately, these methods continue to exhibit multiple limitations. First, users struggle to consistently choose between two near-equal trajectories, leading to the paradox of choice—the observation that more choices only increase confusion and not the number of good options—and thus conflicting data entries (Bıyık et al., 2022; Casper et al., 2024). More fundamentally, each pairwise comparison provides at most 1 bit of information: whether one trajectory is preferred over another (Biyik et al., 2019).

Our method in this paper overcomes these barriers. First, we only show one trajectory per query to the user, thereby not inducing the paradox of choice. Second, users respond with comparative language feedback, offering more informative and time-efficient input while maintaining intuitiveness and ease-of-use.

Active reward learning

While traditional, non-active reward learning methods select the next query through random chance; queries chosen without any thought behind them lead to data-inefficiency—the reward model may not learn much per feedback—and overall user burden, as users will be prompted for more or possibly confusing queries. On the other hand, active learning methods aim to iteratively select the *greedily optimal* data point after each training epoch, using various statistical models to estimate the optimality of each point (Castro et al., 2008; Cohn et al., 1996; Settles, 2009). This idea naturally applies to preference-based reward learning paradigms where an oracle (usually a human) is queried iteratively. Thus, active reward learning aims to ensure that the most informative queries are shown to the oracle for feedback, following metrics such as volume retrieval (Sadigh et al., 2017) or information gain (Biyik et al., 2019). Further work has sought to increase data-efficiency by querying users with a batch/slate, instead of a pair, of trajectories (Biyik et al., 2023; Bıyık et al., 2024a) or amplify diversity through leveraging ensembles to "vote" on the next query (Burbidge et al., 2007; Houlsby et al., 2011; Krogh and Vedelsby, 1994; Seung et al., 1992). Most of these methods leverage some sort of an approximate Bayesian inference method given its data-efficiency, natural usage for iterative processes, and, crucially, the ability to update a posterior to continuously learn what the user deems optimal (Gal et al., 2017).

In this paper, our introduction to comparative language feedback mandates one trajectory per query. Ergo, we focus on an active learning method that uses this single trajectory in conjunction with the comparative language feedback response to decide the next query. This inclusion enables an extra boost in data efficiency and leads to faster convergence in preference learning.

Problem formulation

Reward model

We model the robot as a decision-making agent in a finite-horizon Markov decision process (MDP). Each robot trajectory consists of a sequence of state-action pairs for T time steps: $\tau = \{(s_0, a_0), (s_1, a_1), \ldots, (s_{T-1}, a_{T-1})\}$. A reward function $R(s_t, a_t)$, which is only known by the human user, encodes human preferences regarding the task. The reward of a trajectory is defined as: $R(\tau) = \sum_{t=0}^{T-1} R(s_t, a_t)$. For each trajectory, the human may provide language feedback l that attempts to improve the trajectory in one aspect, for example, speed, distance to objects, height of the robot's arm, etc. For instance, if the robot is currently not distant enough from the stove, the human might respond "move farther from the stove" (see Figure 1). Our goal is to develop a framework where we utilize such feedback to learn a reward function r_ξ, parameterized by ξ, which best mimics the human's underlying reward function R.

Unfortunately, this task is doomed without any information about how language feedback relates to the different

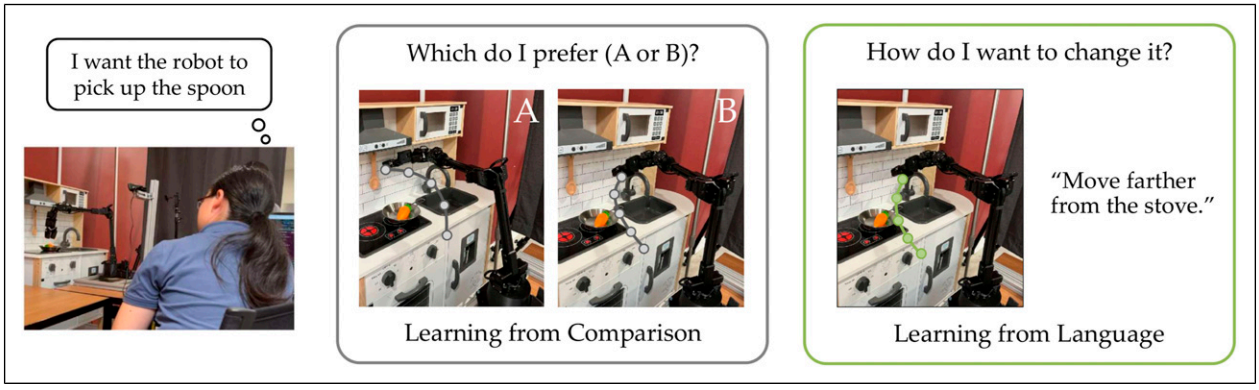

Figure 1. An image from our human subjects studies where the human user wants the robot to pick up the spoon. Preference learning from comparisons requires the user to watch multiple trajectories and rank them to learn a reward function. On the other hand, our introduced method leverages comparative language feedback instead of comparisons, enabling users to provide more informative feedback after observing only one trajectory, guiding the robot to capture human preferences more efficiently.

aspects of the task. In this work, we learn this relation with an offline pretraining phase where we collect a dataset that consists of pairs of robot trajectories and a language label describing how the two trajectories differ. We use this dataset to learn encoders that map trajectories and language feedback onto a shared latent space. We leverage these encoders to learn the preferences (reward functions) of different users based on their language feedback.

Queries

While the goal of iterative trajectory improvement and reward learning is to elicit user preferences to search for an optimal trajectory and create a reward function, respectively, active learning aims to reduce data inefficiency and user burden.

We generally follow the notation of Biyik et al. (2019) who studied active learning for pairwise comparisons. In the case of active learning with comparative language feedback, each query Q consists of a single trajectory τ, that is, $Q = \{\tau\}$. Since reward learning requires providing a sequence of queries to the human, the goal is to minimize the sequence length while maintaining or improving convergence speed.

Query selection is thus performed greedily using approximate Bayesian inference techniques. Starting with some initial prior regarding the distribution over reward function parameters ξ, we query the human with a query Q and receive feedback l in every iteration. Using this collected information, we calculate the posterior:

$$P(\xi \mid Q, l) \propto P(l \mid Q, \xi)P(\xi), \tag{1}$$

where $P(l|Q, \xi)$ represents the probability that the human with reward parameterization ξ responds with language feedback l when shown query Q. This posterior is then leveraged for selecting the next greedily optimal query, conclusively aiming to reduce the total number of queries necessary to learn the human's preferences.

In the next section, we detail our approach to learning the latent space and explain how this learned latent space is utilized for iterative trajectory improvements and reward learning. We then further explain our active learning method that integrates comparative language feedback.

Methods

Our approach is composed of three stages: (i) we learn a shared latent space where robot trajectories and comparative language feedback are aligned, (ii) we leverage this learned latent space for trajectory improvement and reward learning, and finally (iii) we further enhance preference learning through active query selection to achieve higher data efficiency. An overview is shown in Figure 2.

Learning the shared latent space

To learn a shared latent space for trajectories and language feedback, we collect a dataset of (τ^A, τ^B, l) tuples, where τ^A and τ^B are a pair of trajectories and l is a comparative language utterance that describes the difference between the two trajectories. Note that this language utterance does not suggest anything about a human's preference about the robot: it simply describes a difference between the pair of trajectories. It is valid to have l = "move faster" if the robot moves faster in τ^B than in τ^A even if a user would prefer the robot to move slower.

After collecting such a dataset, we propose the model visualized in Figure 2 to learn the shared latent space between trajectories and language utterances. Similarly to Katz et al. (2021), we use a neural network to encode each state-action pair (s_t, a_t) from the pair of trajectories, τ^A and τ^B, to

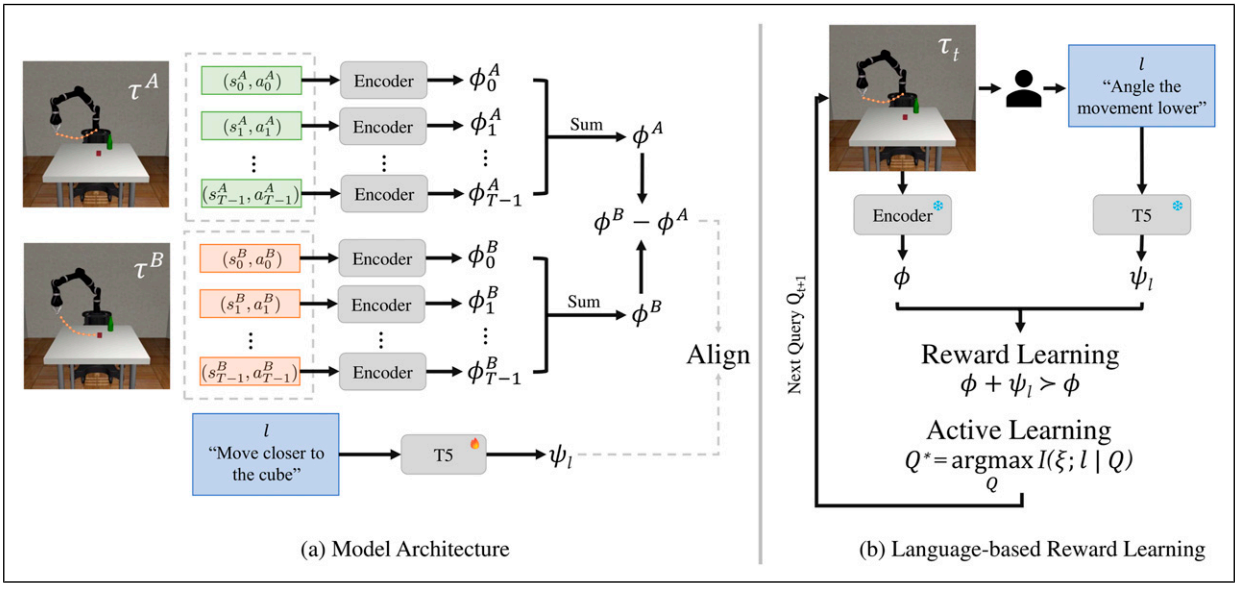

Figure 2. Overview of our approach. (a) Architecture of the model that learns a shared latent space between trajectories and comparative language feedback. Two trajectories and a comparative language feedback are mapped to trajectory embeddings and a language embedding, respectively. The encoder and T5 model are subsequently trained to align the difference between the trajectory embeddings and the language embedding. (b) Comparative language-based active reward learning. The trajectory embedding and language embedding are leveraged for reward learning to learn a reward function, as well as for active learning to choose the next best query.

embeddings ϕ_t^A and ϕ_t^B, respectively. We want these embeddings to contain information about aspects of the trajectories that the human may care and give feedback about.

To achieve this, we align the difference between the trajectory embeddings ϕ^A and ϕ^B with the language embedding ψ_l of the utterance l by using the following loss function:

$$L_{\text{align}}(\tau^A, \tau^B, l) = -\log\left(\text{sigmoid}\left(\psi_l^\top (\phi^B - \phi^A)\right)\right), \quad (2)$$

where the sigmoid is considered as the probability that the language feedback l is uttered for the pair of trajectories τ^A and τ^B. Intuitively, when $\phi^B - \phi^A$ and ψ_l align, that is, have the same direction for fixed magnitudes, the loss is minimized. With this alignment in hand, we acquire the embedding of an improved trajectory $\phi + \psi_l$ for preference learning, given initial trajectory τ and comparative language feedback l.

To realize a language embedding ψ_l, we need a language encoder. Instead of training one from scratch, which would require a prohibitively large and diverse dataset of (τ^A, τ^B, l) tuples, we use a pretrained T5 model (Raffel et al., 2020). To align the trajectory embeddings with the language embeddings, we first freeze the T5 model and solely train the trajectory encoder. Subsequently, we perform co-finetuning of both components.

Additionally, we incorporate a normalization term into the loss function to balance the magnitude of the embeddings. This term constrains the norms of the trajectory embeddings to remain below 1 and the norm of the language embedding to be close to 1:

$$\begin{aligned} L_{\text{norm}}(\tau^A, \tau^B, l) \quad &= \alpha \cdot \big(\max\{\|\phi^A\| - 1, 0\} \\ &\quad + \max\{\|\phi^B\| - 1, 0\}\big) \\ &\quad + \beta \cdot \|\psi_l - 1\|^2, \end{aligned} \quad (3)$$

where α and β are two hyperparameters.

Overall, the objective consists of two terms:

$$L(\tau^A, \tau^B, l) = L_{\text{align}}(\tau^A, \tau^B, l) + L_{\text{norm}}(\tau^A, \tau^B, l), \quad (4)$$

which we use to train the trajectory encoder and finetune T5. Training this architecture enables us to encode any trajectory and language utterance into the same latent space. In the next subsection, we demonstrate how this latent space is useful for iteratively improving robot trajectories and for learning human preferences based on their language feedback.

Utilizing the learned latent space

The shared latent space aligns robot trajectories with human language feedback. This alignment enables an intuitive understanding of user preferences. We will now explore two primary ways to leverage this learned latent space: first, to iteratively improve the robot's trajectory, and second, to accurately learn user preferences.

Iterative trajectory improvement. First, we leverage the latent space to iteratively improve an initial suboptimal robot trajectory. We start by showing the initial trajectory τ^0 to the user and receive language feedback l^0. We then use our trained encoders to compute the trajectory embedding ϕ_{τ^0} and language embedding ψ_{l^0}. We then find the *improved trajectory* τ^1 such that its difference with τ^0 best aligns with the human's language feedback based on cosine similarity:

$$\tau^1 = \underset{\tau'}{\text{argmax}} \frac{\psi_{l^0}^\top (\phi' - \phi^0)}{\|\phi'\|_2 \cdot \|\phi^0\|_2}. \tag{5}$$

We iteratively continue this process for N iterations to obtain τ^0, τ^1, ..., τ^N. In this work, we search over a discrete, predefined set of trajectories to solve the optimization in equation (5) for computational efficiency. It is, however, possible to use reinforcement learning or model-predictive control algorithms to solve this optimization at the expense of increased computational cost.

Reward learning from comparative language feedback. In addition to improving trajectories, we also utilize the learned latent space to learn the user's preference, that is, their reward function. Previous approaches ask users to label their preference from a pair of trajectories. This method requires users to watch two trajectories in order to receive up to 1 bit of information per query—overall, heavily tedious to users and very data-inefficient. In our language-based reward learning approach, we only show users one trajectory τ_i to collect language feedback l_i per query i. Given this queried trajectory and collected language feedback, we construct an improved trajectory $\widehat{\tau}_i$ with trajectory embedding $\phi_{\widehat{\tau}_i} = \phi_{\tau_i} + \psi_{l_i}$. We highlight that $\widehat{\tau}_i$ is an imaginary trajectory that maps to $\phi_{\tau_i} + \psi_{l_i}$ in the learned latent space. Then, following the Bradley-Terry model (Bradley and Terry, 1952), a standard approach in preference-based learning, we model a preference predictor with the reward function r_ξ as:

$$P_\xi(\widehat{\tau}_i \succ \tau_i) = \frac{\exp r_\xi(\phi_{\widehat{\tau}_i})}{\exp r_\xi(\phi_{\widehat{\tau}_i}) + \exp r_\xi(\phi_{\tau_i})}, \tag{6}$$

where r_ξ is a neural network parameterized with ξ. The reward function r_ξ is trained by minimizing the following negative log likelihood loss:

$$L_{\texttt{Explicit}} = -\frac{1}{n} \sum_{i=0}^{n-1} \log P_\xi(\widehat{\tau}_i \succ \tau_i). \tag{7}$$

We now make an important observation about comparative language feedback. Say in a kitchen task that a user tells the robot to move farther from the stove. This feedback primarily indicates a preference for the improved trajectory that moves farther from the stove over the original trajectory. However, this feedback contains much more secondary information than solely this comparison. The user could use any comparative language utterance to teach the robot, but ultimately selected one about the distance to the stove. This implies with high probability that the improvement the robot may get from this feedback is higher than that of any other comparative language feedback.

Mathematically, this indicates a preference for $\widehat{\tau}_i$ over $\widetilde{\tau}_i$ with trajectory embedding $\phi_{\widetilde{\tau}_i} = \phi_{\tau_i} + \psi_{\widetilde{l}_i}$ where $\widetilde{l}_i$ is any language utterance other than l_i. Again we apply the Bradley-Terry model to utilize this implicit preference:

$$P_\xi(\widehat{\tau}_i \succ \widetilde{\tau}_i) = \frac{\exp r_\xi(\phi_{\widehat{\tau}_i})}{\exp r_\xi(\phi_{\widehat{\tau}_i}) + \exp r_\xi(\phi_{\widetilde{\tau}_i})}. \tag{8}$$

Based on this, we sample k language feedback from a set of pre-collected language utterances other than l_i and minimize the following loss regarding the chosen language feedback:

$$L_{\texttt{Implicit}} = -\frac{1}{k} \sum_{j=0}^{k-1} \log P_\xi(\widehat{\tau}_i \succ \widetilde{\tau}_{i,j}). \tag{9}$$

Henceforth, we train the reward model through leveraging the presently-cumulative sequence of provided language feedback in conjunction with the k sampled language feedback at every iteration. Overall, the loss for reward learning from comparative language feedback is as follows:

$$\begin{aligned} L_{\texttt{Reward}} &= L_{\texttt{Explicit}} + \delta L_{\texttt{Implicit}} \\ &= -\frac{1}{n} \sum_{i=0}^{n-1} \log P_\xi(\widehat{\tau}_i \succ \tau_i) \\ &\quad -\frac{\delta}{n} \sum_{i=0}^{n-1} \frac{1}{k} \sum_{j=0}^{k-1} \log P_\xi(\widehat{\tau}_i \succ \widetilde{\tau}_{i,j}), \end{aligned} \tag{10}$$

where δ is a hyperparameter. Training the reward function with this loss enables us to efficiently capture human preferences through comparative language feedback.

Active querying for reward learning

Through leveraging comparative language feedback, we have integrated more bits of information than the pairwise comparisons to accelerate learning the reward function. To further boost the data-efficiency of reward learning, we exploit the sequential nature of this paradigm—after every feedback from the human, we optimize what query to show next. Inspired by previous work, we use the mutual information objective, an acquisition function that aims to greedily maximize the information gained after each query with the side benefit of asking easier questions to the human (Biyik et al., 2019). However, these methods have been developed only for comparison feedback, not language.

Thus, our objective is to accomplish this information gain maximization using comparative language feedback.

Problem 1. *We aim to solve the optimization problem*:

$$Q^* = \underset{Q}{\operatorname{argmax}}\ I(\xi; l \mid Q), \tag{11}$$

where I is information gain and l is the language response to the query Q. This is equivalent to:

$$Q^* = \underset{Q}{\operatorname{argmax}}[H(l \mid Q) - \mathbb{E}_{l|Q}[H(l \mid Q, \xi)]], \tag{12}$$

where H is information entropy (Cover and Thomas, 2012).

To interpret this optimization, we look at the two terms being subtracted. The first entropy term $H(l \mid Q)$ depicts the robot's uncertainty regarding the human's language feedback for the trajectory $\tau \in Q$, and the second entropy term $H(l \mid Q, \xi)$ denotes the human's own uncertainty in the feedback. Conclusively, information gain maximization aims to balance both entropy terms, selecting the trajectory that simultaneously enables the robot to learn the most about the reward function and the human to confidently respond with comparative language feedback.

While Biyik et al. (2019) studied this optimization for pairwise preference comparisons, the probability terms defined for comparative language feedback is not equivalent to those of preference comparisons. The comparative language interface provides the user with the freedom to give language feedback from an open vocabulary, while preference comparison feedback is always from a known binary trajectory set. Deriving these probability terms are not straightforward, ergo, by estimating the expectations via sampling, we rewrite the optimization problem. While the full derivation is provided in the appendix, we present a simplified, implementation-friendly formulation here:

$$Q^* = \operatorname{argmax}_Q \sum_{l \in L} \sum_{\overline{\xi} \in \Xi} \log\left(\frac{P(l \mid Q, \overline{\xi})}{\sum_{\xi' \in \Xi} P(l \mid Q, \xi')}\right), \tag{13}$$

where Ξ and L denote the sampled set of reward parameters and language embeddings, respectively (e.g., using Laplace Approximation (Bernardo et al., 2003; MacKay, 1995)).

To model user response $P(l|Q, \xi)$ computationally, we make two assumptions: (i) the reward is an affine function of the trajectory features, which may still be a nonlinear function of states and actions, and (ii) both the linear reward weights and trajectory features have approximately unit length. Under these assumptions, we adopt the following computational model for user responses:

$$P(l \mid Q = \{\tau\}, \xi) = \frac{l \cdot (\xi - \phi)}{\sum_{\bar{l} \in L} \bar{l} \cdot (\xi - \phi)} \approx \frac{l \cdot (\phi^* - \phi)}{\sum_{\bar{l} \in L} \bar{l} \cdot (\phi^* - \phi)}, \tag{14}$$

where τ^* is the optimal trajectory under the linear reward function r_ξ; ϕ and ϕ^* correspond to the embeddings of trajectories τ and τ^*, respectively. This probability corresponds to the degree of alignment between a user response l and the difference between their personalized optimal trajectory τ^* and the queried trajectory τ. Crucially, we observe that, under the assumptions above, the embeddings of the optimal trajectory τ^* and the user's reward model parameters ξ are equivalent, considering that the maximum dot product between unit vectors is attained only when the vectors are equal.

Using this computational human response model, synthesizing queries that maximize the objective presented in equation (13) yields highly informative data points for comparative language integrated reward learning, improving data-efficiency of reward learning accordingly.

Simulation experiments

In this section, we present our experiments in two simulation domains, demonstrating (i) the superiority of comparative language feedback over traditional pairwise comparisons, and (ii) how language-integrated active querying improves data-efficiency over random querying.

Simulation environments

We used two simulation environments: *Robosuite* (Zhu et al., 2020) and *Meta-World* (Yu et al., 2020). *Robosuite* has a Jaco robot arm at a table with a cube and a bottle, and the task is to pick up the cube. The state consists of 640 × 480 RGB images, resized to 224 × 224, and 25-dimensional proprioception, and the action space is 4-dimensional (the end-effector always points down). *Meta-World* has a Sawyer robot arm at a table, and the task is to push a button on the side of the table. The state consists of 480 × 320 RGB images, resized to 224 × 224, and 20-dimensional proprioception, and the action space is 4-dimensional (the end-effector always points down). Figure 3 depicts our environments.

Simulated humans. For simulation experiments, we synthesize the comparative language feedback from a simulated human with a simplified reward function R:

$$R(s_t, a_t) = w^\top \theta(s_t, a_t), \tag{15}$$

where θ is a function that maps a state-action pair to a vector of high-level features (e.g., speed, distance to the stove). We similarly define $\theta(\tau)$ to denote the sum of features over a trajectory τ. w is a vector of weights that maps the features to a scalar reward value. Both θ and w are unknown to our algorithm.

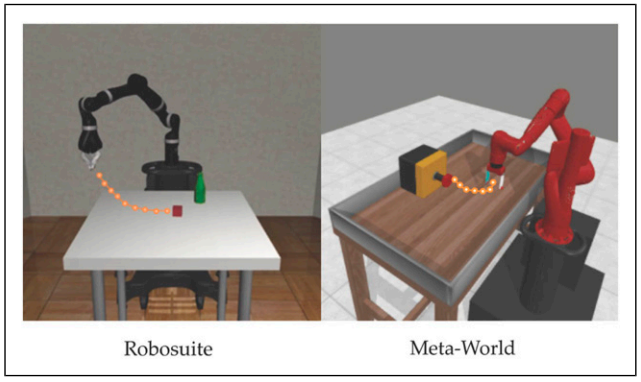

Figure 3. The simulation environments with example trajectories depicted through the orange dots. (Left) *Robosuite* environment: a robot arm moves its end-effector toward the red cube to lift it. (Right) *Meta-World* environment: a robot arm moves its end-effector toward the button to press it.

Given a true reward function r_w, let τ^* be the optimal trajectory under reward r_w. Then, the synthetic human gives the noisy language feedback l^0 when shown trajectory τ^0:

$$l^0 \sim \ell\left(\text{softmax}(w \odot (\theta_l^* - \theta_l^0))\right), \tag{16}$$

where θ^* and θ^0 denote the true cumulative features of τ^* and τ^0, respectively, and $\odot$ denotes element-wise multiplication. The function ℓ takes a sample from the output of the softmax, which is computed across different features we designed for the simulated humans, and outputs a language feedback that corresponds to the sampled feature. The language feedback to output is chosen randomly from all language utterances of that feature in the GPT-augmented dataset, which we discuss next.

Learning the latent space

Lists of language feedback were created for hand-crafted features θ to indicate a change (e.g., {'Move higher', 'Move lower'} for height). They were then augmented with GPT 3.5 (OpenAI, 2023) to 629 sentences for *Robosuite* and 660 for *Meta-World*. The splits are 480, 74, 75 and 492, 84, 84 for the training, validation, and test sets. Note these features are only for synthetic dataset creation—training our architecture does not require hand-designed features.

For the *Robosuite* environment, these hand-crafted features consist of:

- The height of the robot's end-effector
- The speed of the end-effector
- The distance between the end-effector and the bottle
- The distance between the end-effector and the cube
- How well the robot lifts the cube (i.e., the level of success of in the cube-lifting task)

were the level of success is quantified by: (1) whether or not the cube is lifted above the height of a success threshold or (2) a weighted sum of the distance to the cube and whether or not the end-effector is grasping the cube. For the *Meta-World* environment, the hand-crafted features consist of:

- The height of the robot's end-effector.
- The velocity of the robot's end-effector.
- The distance between the end-effector and the button.

Note that the simulated humans leverage the same hand-crafted features θ for their reward function, though the human's reward weights w was randomly initialized for reward learning. Thus, the reward learning objective is for the robot to align with the simulated human's reward function which randomly prioritizes different hand-crafted features.

We trained RL policies with randomized weights w using stratified sampling (Kochenderfer and Wheeler, 2019) over the features, then generated rollout trajectories with a horizon of $T = 500$ for each environment. For *Robosuite*, we generated 448 unique rollouts and 324 for *Meta-World*. The splits were 359, 44, 45 and 260, 32, 32 for training, validation, and test sets. Trajectories were paired within each set and matched with a language feedback that describes the change in each feature from τ^A to τ^B (see Figure 4). For the hyperparameters used in equation (3), we set $\alpha = 1$, $\beta = 0.1$ for both *Robosuite* and *Meta-World* environments.

We trained the encoders with the loss in equation (4), and used accuracy as the metric to evaluate:

$$\text{Acc} = \frac{\left|\{(\tau^A, \tau^B, l) \mid \psi_l^\top (\phi^B - \phi^A) > 0\}\right|}{\text{Total number of samples}}. \tag{17}$$

Training the encoders with the training set of trajectories and language utterances, we observed a test accuracy of 84.9% in *Robosuite* and 82.9% in *Meta-World*. This

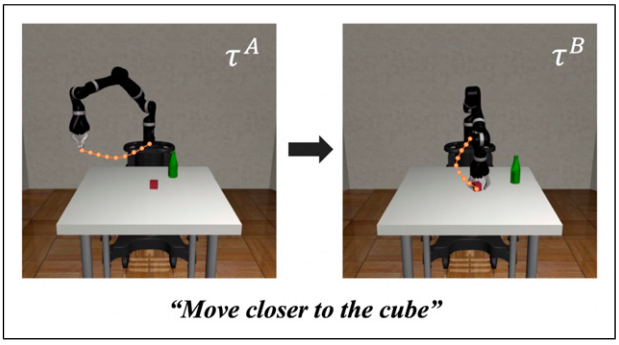

Figure 4. A sample within the dataset used for learning the shared latent space between trajectories and comparative language feedback. Each dataset sample contains a pair of trajectories and a comparative language feedback that describes a difference between the two.

Table 1. Test accuracy of training with co-finetuned encoders versus a frozen language model across both simulation environments.

Method	*Robosuite*	*Meta-World*
Co-finetune	0.8489	0.8288
Freeze T5	0.7625	0.7344

indicates the trajectory and language embeddings are well aligned. Table 1 details our results comparing co-finetuning both the trajectory encoder and language model versus utilizing a frozen language model and only training the trajectory encoder. We found that co-finetuning increases test accuracy performance by more than 8%, indicating that the encoders better learn a shared latent space between trajectories and comparative language feedback.

Iterative trajectory improvement

Next, we conducted iterative trajectory improvement experiments. We set the number of iteration steps $N = 15$ and repeat the full experiment over 100 random seeds. The true rewards of these trajectories are shown in Figure 5. In both environments, we consistently improve the trajectories, which showcases the effectiveness of the learned latent space and the improvement algorithm. However, it can be noted that our algorithm could not reach the performance of the optimal trajectory. This is because every improvement iteration is completely independent from the previous iterations (i.e., the improvements are memoryless) and the robot may get stuck in a loop between good, but non-optimal, trajectories.

Our reward learning algorithm presented in the reward learning section, on the other hand, utilizes the entire history of human feedback, so it does not suffer from this problem. We will now demonstrate this.

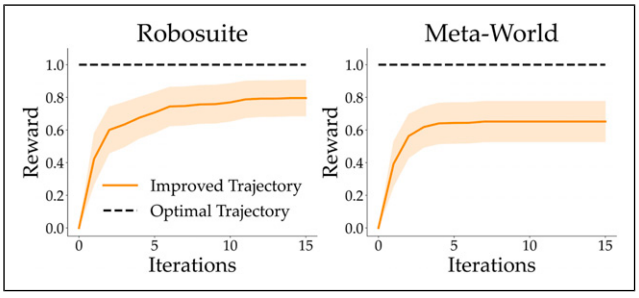

Figure 5. Results of experiments where we use simulated human language feedback to iteratively improve a robot trajectory (mean ± std over 100 runs). The dashed line represents *average* reward of optimal trajectories.

Reward learning

We again randomly initialize the reward weights w to simulate human feedback. We simulate five synthetic humans for *Robosuite* and three for *Meta-World*, and run the experiments with three random seeds for each simulated human. We use the loss in equation (10) to learn the reward function. We adopt cross-entropy $CE\ (P_w,\ P_\xi)$ as the evaluation metric, where

$$P_w(\tau^A \succ \tau^B) = \frac{\exp w^\top \theta(\tau^A)}{\exp w^\top \theta(\tau^A) + \exp w^\top \theta(\tau^B)},$$
$$P_\xi(\tau^A \succ \tau^B) = \frac{\exp r_\xi(\phi_{\tau^A})}{\exp r_\xi(\phi_{\tau^A}) + \exp r_\xi(\phi_{\tau^B})}. \tag{18}$$

Another metric is the true reward value of the optimal trajectory selected from the test set based on the learned reward, reflecting how close the learned reward is to the true reward. All reward values are normalized between 0 and 1. Intuitively, the cross-entropy evaluation metric tests how well a method captures a user's average preference across a multitude of trajectories, while the true reward value evaluation metric engages how well a method accurately identifies an edge case, that is, the optimal trajectory.

We compare our method against reward learning from comparisons, where for each query, the simulated user chooses a preferred trajectory from pair $(\tau^A,\ \tau^B)$ with probability shown in equation (18). We then evaluate the impact of active querying for reward learning with comparative language feedback and compare the results.

Baselines. For our analyses, we compare six methods:

COMPARISON: The reward model is parameterized as a neural network and learns using comparison feedback between two uniformly sampled trajectories at every iteration.

Language: The reward model is parameterized as a neural network and learns using comparative language feedback to a uniformly sampled trajectory at every iteration.

LANGUAGE: The comparative language queries actively selected in every iteration by the following procedure. We assume a linear reward function and a prior distribution over its weights $P(\xi)$. After every feedback, we approximate the posterior formulated in equation (1) with Laplace approximation (Nickisch and Rasmussen, 2008; Bıyık et al., 2024b), which allows us to sample the set of reward weights Ξ. Using these samples $\xi \in \Xi$, we also generate the language embedding samples L from $P\ (l|Q,\ \xi)$ (see equation (14)), again using Laplace approximation for tractability. Having these two sets Ξ and L, we compute the best query by solving equation (13). At the end, even though active learning optimization assumed a linear reward function, we train a neural network based reward model using the collected human data.

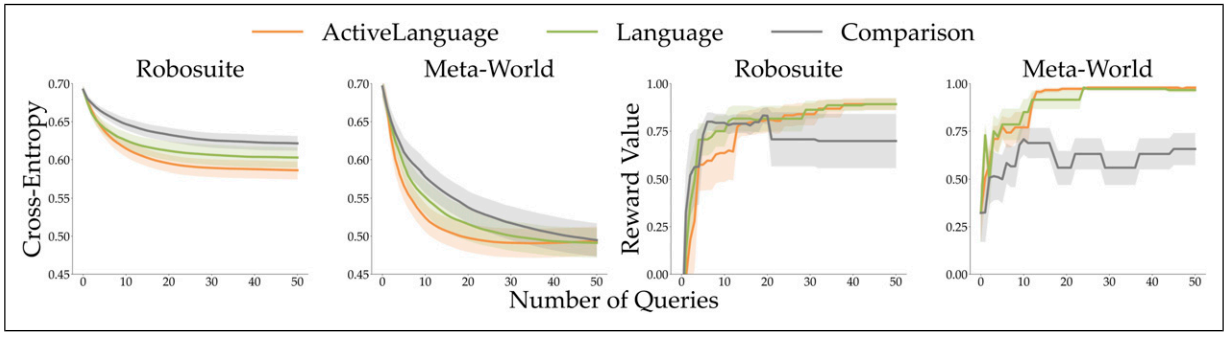

Figure 6. Reward learning evaluation in cross-entropy and reward value results for test set queries are shown for the ACTIVELANGUAGE, LANGUAGE, and COMPARISON methods in the *Robosuite* and *Meta-World* environments (mean ± s.e. over 5 seeds). The usage of comparative language feedback and an active query generation method triumphs over preference comparisons, as seen in both cross-entropy and reward value metrics.

PURELYBAYESIAN: The active querying methodology is identical to that of ACTIVELANGUAGE; however, the reward model at the end is not parameterized as a neural network. Instead, the mean of the set of sampled reward weights, Ξ, calculated during the active learning procedure, is used directly as the weights of the linear reward model.

BALD: Following suit the Bayesian Active Learning by Disagreement (BALD) algorithm by Houlsby et al. (2011), a deep ensemble of reward models is trained for reward learning. The sample mean deep ensemble output is used as the reward model output. For active learning optimization: the parameterizations of the deep ensemble are leveraged for Ξ. The set of sampled language embeddings L is obtained similarly to ACTIVELANGUAGE, but now using the deep ensemble's reward model weights Ξ. Lastly, these Ξ and L are used to maximize information gain as in equation (13).

QUERYBYCOMMITTEE: Following suit the Query By Committee algorithm by Krogh and Vedelsby (1994), a deep ensemble of reward models is learned for reward learning, and the trajectory with the highest ensemble uncertainty is chosen as the next query. We model trajectory uncertainty as the reward value variance outputted by the deep ensemble. This does not leverage our information gain formulation.

At every iteration: COMPARISON and LANGUAGE choose the next query randomly; ACTIVELANGUAGE, PURELYBAYESIAN, and BALD select the next query through computing the expected information gain; and QUERYBYCOMMITTEE selects the next query through estimating uncertainty. Overall, we make two comparisons. First, we analyze the impact of feedback modality through evaluating the results of COMPARISON, LANGUAGE, and ACTIVELANGUAGE. Second, to evaluate the effectiveness of our information gain method, we test ACTIVELANGUAGE against a Bayesian method, PURELYBAYESIAN, and two ensemble methods, BALD and QUERYBYCOMMITTEE.

Results and discussion. We first compare between COMPARISON, LANGUAGE, and ACTIVELANGUAGE—the results are shown in Figure 6. We observe that the cross-entropy of the LANGUAGE method decreases faster than that of the COMPARISON preference learning in both environments. Furthermore, ACTIVELANGUAGE then additionally outperforms both methods, demonstrating that our approaches converge quicker than preference learning with comparisons. Meanwhile, we also notice that the true reward value of optimal trajectories reaches higher values in a smaller amount of queries with our LANGUAGE method, and even

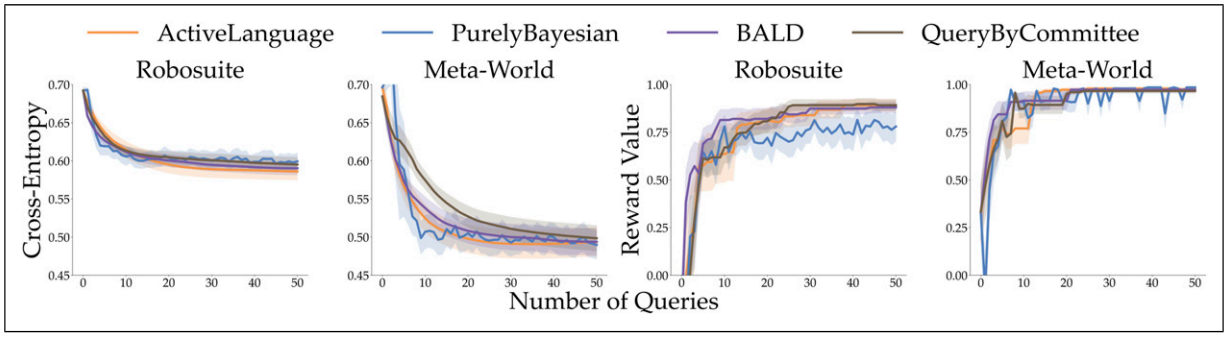

Figure 7. Reward learning evaluation in cross-entropy and reward value results for test set queries are shown for the ACTIVELANGUAGE, PURELYBAYESIAN, QUERYBYCOMMITTEE, and BALD methods in the *Robosuite* and *Meta-World* environments (mean ± s.e. over 5 seeds). The ACTIVELANGUAGE method performs competitively to baselines without requiring an ensemble of networks, as seen in both cross-entropy and reward value metrics.

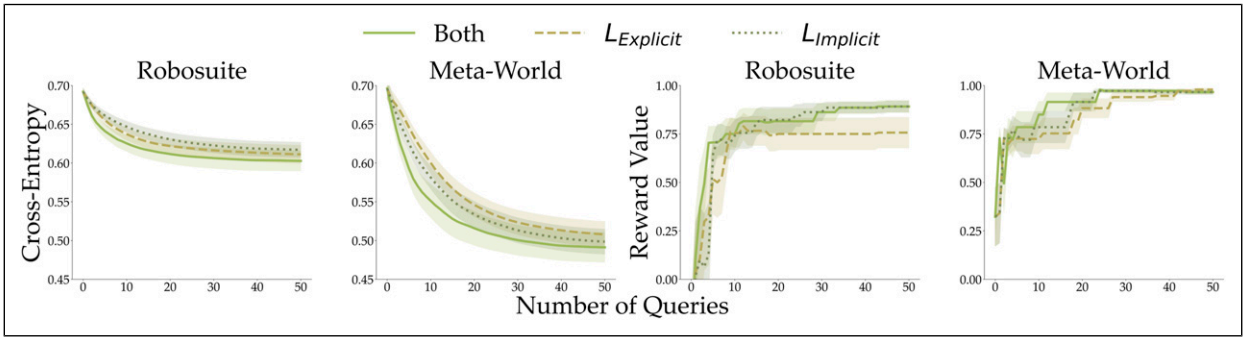

Figure 8. Reward learning evaluation in cross-entropy and reward value results for test set queries are shown for the LANGUAGE (denoted as "Both"), L_{Explicit}, and L_{Implicit} methods in the *Robosuite* and *Meta-World* environment (mean ± s.e. over 5 seeds). The usage of both loss components outperforms learning with only one.

more so using ACTIVELANGUAGE. Interestingly, in both *Robosuite* and *Meta-World* environments, the COMPARISON method degrades in reward value performance, despite continuing to improve in the cross-entropy evaluation. This implies that reward learning with preference comparisons fails to generalize to both the user's average preference and optimal reward value edge case; this method attempts to generalize to the average user trajectory preference through sacrificing its performance for dealing with edge cases.

Next, we compare all the active baselines in Figure 7. We observe that all active methods have competitive cross-entropy and reward value results. In terms of cross-entropy, the ACTIVELANGUAGE method performs the best in both environments. For reward value, ACTIVELANGUAGE is slightly faster in the *Meta-World* environment and performs the comparably equally in *Robosuite*. These trends suggest that leveraging an ensemble does not necessarily boost performance. We hypothesize that this occurs because the ensembles may not consist of a significantly heterogeneous mix of networks, given that the reward model we assume for synthetic humans yield a convex optimization problem even though we still learn deep neural networks. As such, ensemble-based active querying methods will suffer from a lack of diversity and overall select inferior queries. Meanwhile, ACTIVELANGUAGE need not train over an ensemble of networks to find optimal queries and efficiently learn a reward model. We also observe that the curve for PURELYBAYESIAN is not smooth because it samples new reward parameters every iteration, while the other methods update parameters through gradient descent methods. Consequently, its evaluation results often jump radically, leading to worse results in both environments. On the other hand, while ACTIVELANGUAGE leverages the same querying method as PURELYBAYESIAN, the usage of a neural network for the reward model instead of sampling from a distribution enables a smoother learning process and better convergence.

Ablation study. Lastly, we conducted an ablation study to evaluate the different components of the loss function (see equation (10)). Figure 8 depicts our results. For cross-entropy, the combination of both the explicit and implicit components outperforms using each component individually, demonstrating the validity of our loss function design. In terms of reward value, the combination of both methods again performs the best in both environments.

Figure 9. (Left) Closeup view of the kitchen set with the spoon hanging on the wall, above the pan on the stove. (Right) For the experiment: WidowX 250 6DOF robot arm, user, and desktop to input user feedback.

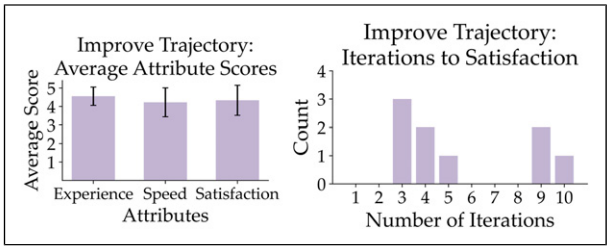

Figure 10. User studies results for the *Trajectory Improvement* method. (Left) Average attribute scores reported in the post-study survey. (Right) Count of users with particular number of iterations to satisfaction.

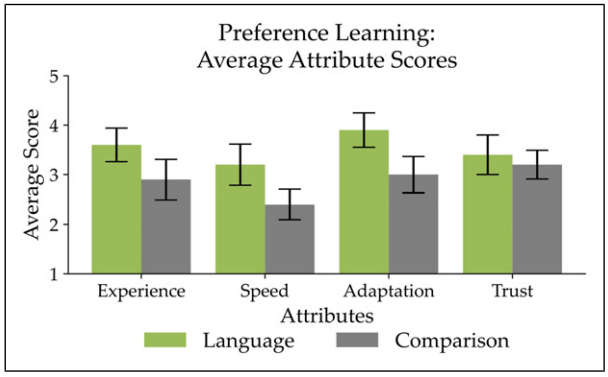

Figure 11. Average attribute scores reported in the post-study survey for the *Reward Learning* method comparing LANGUAGE and COMPARISON. LANGUAGE outperforms COMPARISON across all attributes (mean ± s.e. over 10 users).

User studies

To verify the effectiveness of our approach, we conducted three human subject studies, recruiting 10 subjects for each (4 female and 6 male for the first two studies and 3 female and 7 male for the third study) from varying backgrounds and observing them to interact with our real robot setup. We note that the first two studies were conducted with the same human subjects. While the results of the studies were not shared with the subjects before the completion of study 2, it is still possible that subjects' participation in the first study introduced some bias that may affect the results of study 2. All studies were approved by an Institutional Review Board (IRB) of USC Human Research Protection Program (approval no. UP-24-00340) on April 30, 2024. Participants gave written consent for review and signature before starting user studies.

Experiment setup

We follow the setup of Ebert et al. (2022) where the robot sits in front of a kitchen set (Figure 9). The task is for the robot to prioritize picking up the spoon while avoiding any obstructions such as the pan on the stove. The robot state consists of 480 × 320 images, resized to 224 × 224, and 22-dimensional proprioception, and the actions are 7-dimensional (6DOF + gripper).

To train the latent space, we collected a dataset of 321 trajectories with varying levels of success at picking up the spoon, avoiding the pan, and speed. These trajectories were divided into 192 for training, 64 for validation, and 65 for testing. 496 GPT-augmented sentences were split into 297 for training, 99 for validation, and 100 for testing. The following lists the hand-crafted features for designing the language feedback necessary to learn the shared latent space:

- The velocity of the end-effector
- The distance between the end-effector and the pan
- How well the robot picks up the spoon (i.e., level of success)

were the level of success is quantified by: (1) whether the spoon is grasped from the hook (2) a weighted sum of the distance to the spoon and whether the end-effector is grasping the spoon. The sentences in the GPT-augment dataset were paired with the trajectories in the same manner as in the simulation experiments. Finally, another set of 32 trajectories was collected for the user studies following the pretraining phase, which will be used for trajectory improvement or (active) reward learning.

Study 1. In the first study, we investigate the performance of iterative trajectory improvement application we presented. Users are given a suboptimal trajectory from the dataset and tasked to provide language feedback. Afterward, an improved trajectory is shown. This process is repeated up to 10 iterations or until the user is satisfied, whichever is earlier.

Table 2. User satisfaction between LANGUAGE and COMPARISON reported in the post-study survey for various aspects. The satisfaction is not mutually exclusive between the methods.

	Satisfactory	
Method	Language	Comparison
Time efficient	9/10	1/10
Convenient	6/10	8/10
Adaptable	8/10	3/10

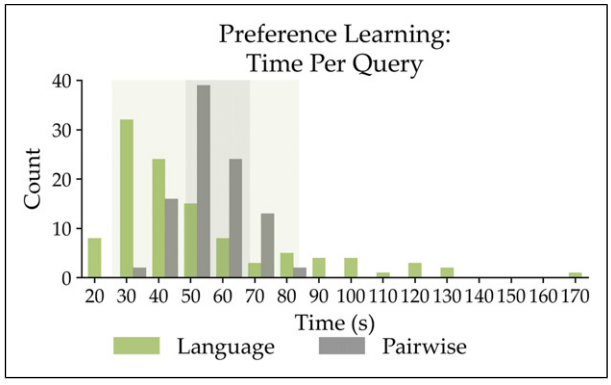

Figure 12. Average time per query for 10 distinct users in the *Reward Learning* method comparing LANGUAGE and COMPARISON. LANGUAGE feedback takes on-average less time per query but has higher variance due to the variance in the lengths of user feedback.

Study 2. In the second user study, we investigate the effectiveness of reward learning. For 20 iterations, users provide feedback (either preference comparison, for COMPARISON, or comparative language, for LANGUAGE) on randomly chosen trajectories that train a reward model. After every 5 iterations, the user must rate (on a scale from 1 to 5, 5 being the best) the trajectory that the learned-so-far reward model believes is optimal. Upon completion, the user performs this experiment with the other feedback type. To avoid bias, 5 users began with the comparison-based method and the other 5 with the comparative language based method.

Study 3. Finally, in the third study, we investigate the active learning extension to reward learning. For 20 iterations, users provide comparative language feedback on (either randomly chosen, LANGUAGE, or actively chosen, ACTIVELANGUAGE) trajectories that train a reward model. After every 5 iterations, the user must rate (on a scale from 1 to 5, 5 being the best) the optimal trajectory under the reward model learned so far. Upon completion, the user performs this experiment with the other querying method. To avoid bias, 5 users began with the random querying method and the other 5 with the active querying method.

Before and after each study, users were asked to complete pre-study and post-study surveys (see Appendix).

Results and discussion

Study 1. Our post-study surveys reveal that the *Trajectory Improvement* method has consistently positive responses for user experience and speed of adaptation, but has two peaks in iterations to satisfaction (Figure 10). We conjecture that (i) the dataset of the 32 trajectories may not have contained the trajectories that the users desired, (ii) some users started with high expectations but became more lenient over time, or (iii) some users had stricter criteria for satisfaction, leading trajectory improvement to take longer.

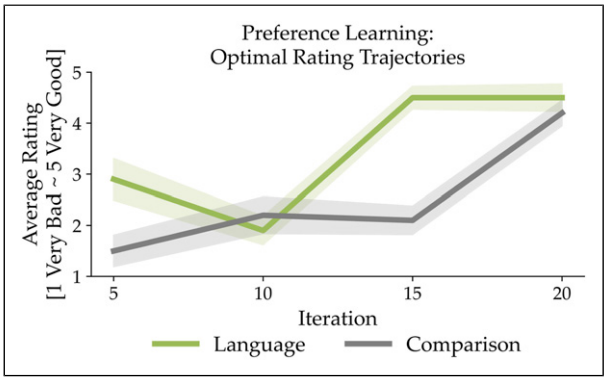

Figure 13. Average ratings of the optimal trajectory shown every 5 iterations in the *Reward Learning* method comparing LANGUAGE and COMPARISON (mean ± s.e. over 10 users). LANGUAGE converges faster and higher than COMPARISON.

Study 2. As for the reward learning study, our language-based method scored better than the comparison-based method for all attributes asked to the users in the post-study survey. The average score of the language-based method over all attributes is 23.9% higher than the comparison-based method, illustrating the superior performance of our approach (see Figure 11).

Shown in Table 2, users found the comparison method to be more convenient than the language method; binary responses are indeed hypothetically easier for users than asking for comparative language feedback.

However, our method triumphs in terms of time-efficiency and adaptability, taking 11.3% less average time for users to answer each query as shown in Figure 12, despite these queries including the time it took for users to type their feedback. Ergo, although providing language feedback may occupy some time, users spend significantly more time in the comparison setting watching multiple trajectories and contemplating during difficult queries consisting of near-equal trajectories. We note that the difference would be even more significant if language feedback was collected through a text-to-speech (TTS) interface rather than a keyboard, as several users chose to give long sentences as feedback.

Figure 13 shows the user rating of optimal trajectories given by the learned reward function. To quantitatively assess the learning efficiency, we follow prior work in active learning (Culver et al., 2006; Fuchsgruber et al., 2024; Myers et al., 2021) by checking the area under curve (AUC) for both lines. We found that the AUC for the comparative language curve is higher than that of the curve for preference comparison with statistical significance ($p < 0.05$, one-sample t-test). This indicates that our language-based approach more efficiently captures user preferences compared to the comparison-based method.

Unexpectedly, Figure 13 also details the decrease in their ratings from query 5 to query 10 while using comparative language feedback. We believe this occurs due to the relatively small subject size: it is possible that the average rating after 5 iterations, where the standard deviation is indeed large, is higher than it should be. Another potential explanation is that users may be more lenient to the robot after 5 iterations than 10 iterations; the comparative language method requires users to only watch 5 trajectories by the 5th iteration, while the preference comparison method requires users to watch 10 trajectories at the same iteration (two trajectories per iteration).

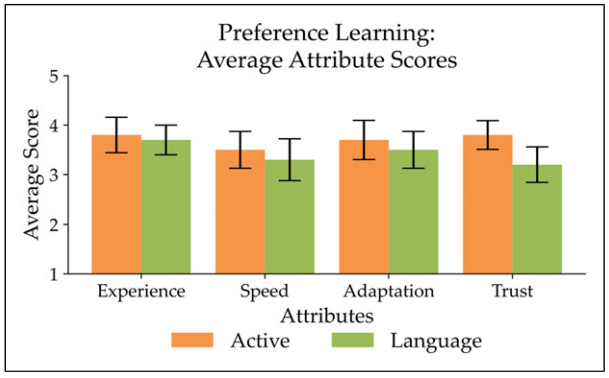

Figure 14. Average attribute scores reported in the post-study survey for the *Reward Learning* method comparing ACTIVELANGUAGE and LANGUAGE (mean ± s.e. over 10 users). ACTIVELANGUAGE slightly but consistently outperforms LANGUAGE across all attributes.

Study 3. Our active querying method scored moderately better than the random querying method for all qualitative metrics. The average score of the active querying method over all attributes is 8.31% higher than the random querying method. Figure 14 shows the user rating of optimal trajectories given by the learned reward function.

Figure 15 shows the user rating of optimal trajectories given by the learned reward function. Like previously, we quantitatively assess the learning efficiency by checking the area under curve (AUC) for both lines. We observe that the AUC for the active method curve is higher than the curve for the non-active method without statistical significance ($p < 0.33$ for all queries and of $p < 0.22$ after the 20th query, one-sample t-test). We believe the lack of statistical significance is due to the small dataset size (32 trajectories) relative to the number of queries (20 trajectories shown for feedback), while our simulation experiments ran 50 queries for datasets containing hundreds of trajectories. Ergo, queries determined through random querying will highly overlap those selected through active querying, implying that the active querying method would only give a moderate boost in data-efficiency. Nevertheless, the addition of active querying generation led to increased performance in all attributes and optimal trajectory ratings, consequently demonstrating its effectiveness in aligning robot behavior with human preferences.

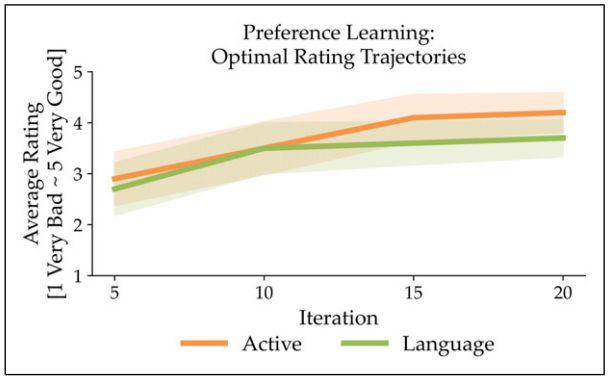

Figure 15. Average ratings of the optimal trajectory shown every 5 iterations in the *Reward Learning* method comparing ACTIVELANGUAGE and LANGUAGE (mean ± s.e. over 10 users). ACTIVELANGUAGE converges faster and higher than LANGUAGE.

Conclusion

Summary

In this paper, we presented a robot learning framework to learn human preferences using comparative language feedback. For this, we aligned trajectories with language feedback in a shared latent space through training and fine-tuning a trajectory encoder and a language encoder. Subsequently, we used this shared latent space for trajectory improvement and to learn preferences through an active reward learning paradigm. Experiments in simulation environments and real-world user studies suggest that our approach consistently outperforms comparison-based preference learning and is favored by most users for aspects such as time-efficiency and adaptability. Likewise, our addition of an active querying method for comparative language further improves data-efficiency and outperforms other active baselines.

Limitations and future work

Even though our user studies show some generalizability in the comparative language feedback, our model is limited by the feedback about the objects seen in the pretraining. Ensuring that our method can work for a wider array of objects in addition to lessening the manual burden of collecting a pretraining dataset are potential next steps. Herein, future work should research into image annotations or foundations models to caption all objects within a scene. Regarding lessening the data collection burden, investigation of leveraging both hand-crafted and learned features to collect synthetic and human language comparisons or using foundation models to generate pretraining data are both notable paths. We generated synthetic language comparisons for pretraining using a set of hand-crafted features in this paper. A real world application would use a combination of hand-crafted and learned features: hand-crafted features would be used to generate synthetic language comparisons for pretraining which are then augmented with comparisons from humans. At the end, this combined dataset would be further augmented with LLM-generated language labels, just like in this paper, for better generalizability to different ways humans may provide comparative language feedback.

Furthermore, the collected comparative language feedback primarily describes a difference between two trajectories across one pre-defined feature for simplicity reasons. Extending these feedback to provide improvements on multiple features may give additional data efficiency benefits and allow users to give feedback in a truly open-vocabulary manner, albeit this method may be computationally complex for both learning the shared latent space as well as reward learning. Such method entails the experimentation of more complex encoders and reward model parameterizations, or adding further components to the reward learning framework such as an LLM to decompose a single multi-feature comparative feedback into multiple single-feature comparative feedback instances, and subsequently using each instance for reward learning. From another lens, real use cases of language feedback are ambiguous and not explicitly comparative, for example, "Be careful around that stove." Future work should study how such ambiguous feedback can be mapped to specific features used in reward learning, potentially through the usage of foundation models that may use task-related context to perform a mapping.

Regarding the active learning portion, we leveraged Laplace approximation for quick sampling through fitting a Gaussian distribution over a linear reward space. This reward function is linear over nonlinear features learned by neural networks during pretraining; hence, the reward space is nonlinear with respect to the environment state and actions. Nevertheless, a user's hidden reward function may be more complex, being multimodal or unable to be well represented as a linear function over interpretable features. Further research should aim to investigate the usage of more complex reward models, perhaps parameterized by a neural network or Gaussian process, and developing comparative language based active querying methods for such.

ORCID iDs

Eisuke Hirota https://orcid.org/0009-0003-3105-7819
Ayano Hiranaka https://orcid.org/0009-0002-3434-9406
Erdem Bıyık https://orcid.org/0000-0002-9516-3130

Ethical considerations

This study was approved by an Institutional Review Board (IRB) of USC Human Research Protection Program (approval no. UP-24-00340) on April 30, 2024.

Consent to participate

Participants gave written consent for review and signature before starting user studies.

Consent for publication

Participants gave written consent for review and signature before starting user studies.

Author contributions

JT created the trajectory datasets for learning the shared latent space and developed the initial code base for iterative trajectory improvement, under the supervision of SJR and AD. ZY and MJ worked on iterative trajectory improvement, and implemented reward learning algorithms. They also conducted the relevant simulation experiments and the first two user studies. EH developed the active querying method and conducted its simulation experiments. EH and AH conducted the third user study. EB supervised all steps throughout the project and obtained the IRB approvals. ZY, EH, and EB led the writing of the manuscript.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by NSF HCC grant #2310757 and a gift in support of the Center for Human-Compatible AI at UC Berkeley from the Open Philanthropy Foundation. Eisuke Hirota was funded by NSF CNS grant #2051117 through the Robotics & Autonomous Systems REU site at the University of Southern California.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Note

1. Parts of this work have been published in the Conference on Robot Learning (CoRL) (Yang et al., 2024). This work newly includes the active learning method to select the most informative queries to the human, and several new results in simulation experiments and human subjects studies.

References

Abbeel P and Ng A (2004) Apprenticeship learning via inverse reinforcement learning. In: Proceedings, twenty-first international conference on machine learning, Banff, AB, Canada, 04 July 2004.

Akgün B, Cakmak M, Jiang K, et al. (2012) Keyframe-based learning from demonstration. *International Journal of Social Robotics* 4: 343–355. https://api.semanticscholar.org/CorpusID:10004846

Argall BD, Chernova S, Veloso M, et al. (2009) A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5): 469–483.

Bajcsy A, Losey DP, O'malley MK, et al. (2017) Learning robot objectives from physical human interaction. In: *Conference on robot learning*. PMLR, pp. 217–226.

Banayeeanzade A, Bahrani F, Zhou Y, et al. (2025) Gabril: Gaze-based regularization for mitigating causal confusion in imitation learning. In: *International Conference on Intelligent Robots and Systems (IROS)*.

Basu C, Singhal M and Dragan AD (2018) Learning from richer human guidance: augmenting comparison-based learning

with feature queries. In: Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction, Chicago, IL, USA, 26 February 2018, pp. 132–140.

Bernardo J, Bayarri M, Berger J, et al. (2003) The Variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics* 7(453-464): 210.

Bıyık E (2025) *Training Robots With Natural and Lightweight Human Feedback*. AI Magazine.

Biyik E, Palan M, Landolfi NC, et al. (2019) Asking easy questions: a user-friendly approach to active reward learning. *Proceedings of the 3rd Conference on Robot Learning (CoRL)*. PMLR, 100, 1177–1190.

Bıyık E, Lazar DA, Pedarsani R, et al. (2021) Incentivizing efficient equilibria in traffic networks with mixed autonomy. *IEEE Transactions on Control of Network Systems* 8(4): 1717–1729.

Bıyık E, Losey DP, Palan M, et al. (2022) Learning reward functions from diverse sources of human feedback: optimally integrating demonstrations and preferences. *The International Journal of Robotics Research* 41(1): 45–67.

Biyik E, Yao F, Chow Y, et al. (2023) Preference elicitation with soft attributes in interactive recommendation. ArXiv Preprint arXiv:2311.02085.

Bıyık E, Anari N and Sadigh D (2024a) Batch active learning of reward functions from human preferences. *ACM Transactions on Human-Robot Interaction* 13(2): 1–27.

Bıyık E, Huynh N, Kochenderfer MJ, et al. (2024b) Active preference-based Gaussian process regression for reward learning and optimization. *The International Journal of Robotics Research* 43(5): 665–684.

Bradley RA and Terry ME (1952) Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* 39(3/4): 324–345.

Brown D, Goo W, Nagarajan P, et al. (2019) Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In: *International conference on machine learning*. PMLR, 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.

Bucker A, Figueredo L, Haddadinl S, et al. (2022) Reshaping robot trajectories using natural language commands: a study of multi-modal data alignment using transformers. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022, pp. 978–984.

Bucker A, Figueredo L, Haddadin S, et al. (2023) Latte: language trajectory transformer. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May 2023–02 June 2023, pp. 7287–7294.

Burbidge R, Rowland JJ and King RD (2007) Active learning for regression based on query by committee. In: Intelligent Data Engineering and Automated Learning-IDEAL 2007: 8th international conference, Birmingham, UK, 16–19 December 2007, pp. 209–218.

Campos JA and Shern J (2022) Training language models with language feedback. In: *ACL workshop on learning with natural language supervision*, Vol. 2022.

Casper S, Davies X, Shi C, et al. (2024) Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research (TMLR)*.

Castro R, Kalish C, Nowak R, et al. (2008) Human active learning. *Advances in Neural Information Processing Systems* 21: 241–248.

Christiano PF, Leike J, Brown T, et al. (2017) Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems* 30: 4299–4307.

Cohn DA, Ghahramani Z and Jordan MI (1996) Active learning with statistical models. *Journal of Artificial Intelligence Research* 4: 129–145.

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

Cui Y, Karamcheti S, Palleti R, et al. (2023) No, to the right: online language corrections for robotic manipulation via shared autonomy. In: Proceedings of the 2023 ACM/IEEE international conference on human-robot interaction, Stockholm, Sweden, 13 March 2023, pp. 93–101.

Culver M, Kun D and Scott S (2006) Active learning to maximize area under the ROC curve. In: Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006, pp. 149–158.

Dennler N, Delgado D, Zeng D, et al. (2023) The Rosid tool: empowering users to design multimodal signals for human-robot collaboration. In: 18th International Symposium on Experimental Robotics (ISER).

Ebert F, Yang Y, Schmeckpeper K, et al(2022) Bridge data: boosting generalization of robotic skills with cross-domain datasets. In: *Proceedings of Robotics: Science and Systems (RSS)*. DOI:10.15607/RSS.2022.XVIII.063

Ellis E, Ghosal GR, Russell SJ, et al. (2024) A generalized acquisition function for preference-based reward learning. In: International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024.

Fuchsgruber D, Wollschläger T, Charpentier B, et al(2024) Uncertainty for active learning on graphs. In: *Proceedings of the 41st International Conference on Machine Learning*. pp. 14275–14307.

Gal Y, Islam R and Ghahramani Z (2017) Deep Bayesian active learning with image data. In: International conference on machine learning, Sydney, NSW, Australia, 06 August 2017, pp. 1183–1192.

Goyal P, Niekum S and Mooney RJ (2019) Using natural language for reward shaping in reinforcement learning. *IJCAI* 2385–2391.

Goyal P, Niekum S and Mooney R (2021) PixL2R: guiding reinforcement learning using natural language by mapping pixels to rewards. In: *Conference on robot learning*. PMLR, pp. 485–497.

Han M, Zhu Y, Zhu SC, et al(2024) Interpret: interactive predicate learning from language feedback for generalizable task planning. In: *Proceedings of Robotics: Science and Systems (RSS)*. DOI:10.15607/RSS.2024.XX.034

Holk S, Marta D and Leite I (2024a) POLITE: preferences combined with highlights in reinforcement learning. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024, pp. 2288–2295.

Holk S, Marta D and Leite I (2024b) PREDILECT: preferences delineated with zero-shot language-based reasoning in reinforcement learning. In: Proceedings of the 2024 ACM/IEEE international conference on human-robot interaction, Boulder, CO, USA, 11–14 March 2024, pp. 259–268.

Hong M, Liang A, Kim K, et al. (2025) Hand me the data: fast robot adaptation via hand path retrieval. ArXiv Preprint arXiv: 2505.20455.

Hoque R, Balakrishna A, Novoseller E, et al(2021) ThriftyDAgger: budget-aware novelty and risk gating for interactive imitation learning. In: *Proceedings of the 5th Conference on Robot Learning (CoRL)*.

Houlsby N, Huszár F, Ghahramani Z, et al. (2011) Bayesian active learning for classification and preference learning. ArXiv Preprint arXiv:1112.5745.

Katz SM, Maleki A, Bıyık E, et al. (2021) Preference-based learning of reward function features. ArXiv Preprint arXiv: 2103.02727.

Kelly M, Sidrane C, Driggs-Campbell K, et al. (2019) HG-DAgger: interactive imitation learning with human experts. In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019, pp. 8077–8083.

Kochenderfer MJ and Wheeler TA (2019) *Algorithms for Optimization*. MIT Press.

Korkmaz Y and Bıyık E (2025) *Mile: model-based intervention learning*. In: *2025 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 15673-15679.

Krogh A and Vedelsby J (1994) Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems* 7: 231–238.

Lee K, Smith L and Abbeel P (2021) Pebble: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR 139: 6152-6163.

Liang A, Thomason J and Bıyık E (2024) ViSaRL: visual reinforcement learning guided by human saliency. In: International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024.

Lynch C, Wahid A, Tompson J, et al. (2023) Interactive language: talking to robots in real time. *IEEE Robotics and Automation Letters*. doi: 10.1109/LRA.2023.3295255

MacKay DJ (1995) Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems* 6(3): 469–505.

Myers V, Biyik E, Anari N, et al(2021) Learning multimodal rewards from rankings. In: *Proceedings of the 5th Conference on Robot Learning (CoRL)*. PMLR, pp. 342-352.

Nickisch H and Rasmussen CE (2008) Approximations for binary Gaussian process classification. *Journal of Machine Learning Research* 9(10): 2035–2078.

OpenAI (2023) *Gpt-3.5*. https://platform.openai.com/docs/models/gpt-3-5

Ouyang L, Wu J, Jiang X, et al. (2022) Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35: 27730–27744.

Raffel C, Shazeer N, Roberts A, et al. (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140): 1–67.

Ross S, Gordon G and Bagnell D (2011) A reduction of imitation learning and structured prediction to no-regret online learning. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, pp. 627–635.

Sadigh D, Dragan AD, Sastry SS, et al. (2017) Active preference-based learning of reward functions. In: *Proceedings of Robotics: Science and Systems (RSS)*. DOI: 10.15607/RSS.2017.XIII.053.

Sadigh D, Sastry SS, Seshia SA, et al. (2016) Planning for autonomous cars that leverage effects on human actions. In: *Proceedings of Robotics: Science and Systems (RSS)*. DOI: 10.15607/RSS.2016.XII.029.

Settles B (2009) Active learning literature survey.

Seung HS, Opper M and Sompolinsky H (1992) Query by committee. *Proceedings of the fifth annual workshop on computational learning theory*. Association for Computing Machinery, 287–294.

Sharma P, Sundaralingam B, Blukis V, et al. (2022) Correcting robot plans with natural language feedback. In: *Proceedings of Robotics: Science and Systems (RSS)*.

Shi LX, Hu Z, Zhao TZ, et al. (2024) Yell at your robot: improving on-the-fly from language corrections. *arXiv preprint arXiv: 2403.12910*.

Sikchi H, Saran A, Goo W, et al. (2023) A ranking game for imitation learning. *Transactions on Machine Learning Research*.

Spencer J, Choudhury S, Barnes M, et al. (2022) Expert intervention learning: an online framework for robot learning from explicit and implicit human feedback. *Autonomous Robots* 46: 1–15.

Stiennon N, Ouyang L, Wu J, et al. (2020) Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33: 3008–3021.

Wang Y, Sun Z, Zhang J, et al(2024) RL-VLM-F: reinforcement learning from vision language foundation model feedback. In: International Conference on Machine Learning (ICML), Vienna, Austria, 21 July 2024.

Wilde N, Biyik E, Sadigh D, et al(2021) Learning reward functions from scale feedback. In: *Proceedings of the 5th Conference on Robot Learning (CoRL)*. PMLR, pp. 353-362.

Wirth C, Akrour R, Neumann G, et al. (2017) A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research* 18(136): 1–46.

Yang Z, Jun M, Tien J, et al. (2024) Trajectory improvement and reward learning from comparative language feedback. In: 8th annual conference on robot learning. https://openreview.net/forum?id=1tCteNSbFH

Yow J, Garg NP, Ramanathan M, et al. (2024) Extract–explainable trajectory corrections from language inputs using textual description of features. *Front. Robot. AI* 11: 1345693. doi: 10.3389/frobt.2024.1345693

Yu T, Quillen D, He Z, et al. (2020) Meta-world: a benchmark and evaluation for multi-task and meta reinforcement learning. In: *Conference on Robot Learning*. PMLR, pp. 1094–1100.

Zhang T, McCarthy Z, Jow O, et al. (2018) Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018, pp. 5628–5635.

Zhang J, Luo Y, Anwar A, et al. (2025) ReWiND: language-guided rewards teach robot policies without new demonstrations. *Proceedings of The 9th Conference on Robot Learning (CoRL)*. PMLR, 460–488.

Zhu Y, Wong J, Mandlekar A, et al. (2020) Robosuite: a modular simulation framework and benchmark for robot learning. ArXiv Preprint arXiv:2009.12293.

Appendix

Derivation of active querying

$$
\begin{aligned}
Q^* &= \operatorname{argmax}_Q I(\xi; l \mid Q) \\
I\ &(\xi; l \mid Q) \\
&= I(l; \xi \mid Q) && \text{(Mutual information symmetry)} \\
&= H(\xi \mid Q) - \mathbb{E}_{l|Q}[H(\xi \mid Q, l)] && \text{(Definition of mutual information)} \\
&= -\mathbb{E}_{\xi|Q}[\log_2(P(\xi \mid Q))] + \mathbb{E}_{l|Q}[\log_2(P(\xi \mid Q, l))] && \text{(Definition of Shannon's entropy)} \\
&= \mathbb{E}_{\xi, l|Q}[\log_2(P(\xi \mid Q, l)) - \log_2(P(\xi \mid Q))] && \text{(Combine terms)} \\
&= \mathbb{E}_{\xi, l|Q}[\log_2(P(l \mid Q, \xi)) - \log_2(P(l \mid Q))] && \text{(Conditional probability property)} \\
&= \mathbb{E}_{\xi, l|Q}\left[\log_2(P(l \mid Q, \xi)) - \log_2\left(\int P(l \mid Q, \xi')P(\xi' \mid Q)d\xi'\right)\right] && \text{(Law of Total Probability)} \\
&\doteq \mathbb{E}_{\xi, l|Q}\left[\log_2(P(l \mid Q, \xi)) - \log_2\left(\frac{1}{M}\sum_{\xi' \in R} P(l \mid Q, \xi')\right)\right] && \text{(Monte Carlo integration)} \\
&= \mathbb{E}_{\xi, l|Q}\left[\log_2\left(\frac{M \cdot P(l \mid Q, \xi)}{\sum_{\xi' \in R} P(l \mid Q, \xi')}\right)\right] && \text{(Logarithm subtraction rule)} \\
&= \mathbb{E}_{\xi|Q}\left[\mathbb{E}_{l|Q, \xi}\left[\log_2\left(\frac{M \cdot P(l \mid Q, \xi)}{\sum_{\xi' \in R} P(l \mid Q, \xi')}\right)\right]\right] && \text{(Law of total expectation)} \\
&\doteq \mathbb{E}_{\xi|Q}\left[\frac{1}{K}\sum_{l \in L} \log_2\left(\frac{M \cdot P(l \mid Q, \xi)}{\sum_{\xi' \in R} P(l \mid Q, \xi')}\right)\right] && \text{(Approx. of integral)} \\
&\doteq \frac{1}{MK}\sum_{l \in L}\sum_{\bar{\xi} \in R} \log_2\left(\frac{M \cdot P(l \mid Q, \bar{\xi})}{\sum_{\xi' \in R} P(l \mid Q, \xi')}\right), && \text{(Approx. of integral)}
\end{aligned}
$$

Derivation of active querying

where there are two integrals: one spans all values of ξ and another spans all values of l.

User study surveys

The following are the questions in our user study surveys.

Pre-study survey. To assess the subjects' experience and level of skill with robotics and machine learning, we conducted a short pre-study survey before any of the experiments.

(1) "Age"
(2) "Gender"
(3) "Race"
(4) "Highest level of education"
(5) "Please describe your level of robotics experience, on a scale from 1 to 5."
(6) "Please describe your level of machine learning experience, on a scale from 1 to 5."
(7) "Have you ever interacted with a robot before? Please describe:"

The following consists of the demographic data of the participants of studies 1 and 2 (mean ± std):

(1) 23.8 ± 3.5
(2) 6 male, 4 female
(3) 5 Asian, 2 White, 1 Hispanic, 2 Other
(4) 4 High School, 2 Bachelor's, 4 Master's
(5) 3.1 ± 1.4
(6) 4.1 ± 1.2
(7) 7 Yes, 3 No

The following consists of the demographic data of the participants of study 3 (mean ± std):

(1) 25.8 ± 3.0
(2) 7 male, 3 female
(3) 7 Asian, 2 White, 1 Other
(4) 3 Bachelor's, 7 Master's
(5) 3.5 ± 1.1
(6) 4.1 ± 0.7
(7) 10 Yes

Post-study survey—Improve trajectory. Users filled out this post-study survey after completing the Improve Trajectory experiment.

(1) "Are you satisfied with the final trajectory?" [1 Completely unsatisfied—5 Completely satisfied]
(2) "How many iterations did it take for the robot to adapt to your feedback?"
(3) "How fast does the robot adapt to your feedback?" [1 Very slow—5 Very fast]
(4) "How did the robot's learning capabilities affect your interaction experience?" [1 Negatively—5 Positively]
(5) "Overall, do you think the approach is effective?" [Y/N]
(6) "Do you have any other comments?"

Post-study survey—Preference learning. Users filled out this post-study survey after completing two methods of the preference learning experiment.

(1) "What did you like about these methods? For each aspect below, please click the circle if you found the method satisfactory in that regard:" [Adaptability, Time efficiency, Convenience, None of them]
(2) "Could you explain the reasons?"
(3) "How long did it take for the robot to adapt to your feedback?" [1 Very slowly—5 Very quickly]
(4) "How did the robot's learning capabilities affect your interaction experience?" [1 Negatively—5 Positively]
(5) "Did you notice the robot learning or adapting to your behavior?" [1 Not at all—5 Constantly]
(6) "How did the robot's learning process affect your level of trust in its capabilities?" [1 Negatively—5 Positively]
(7) "Overall, which method do you prefer?" [Pair-wise comparison/Language feedback]
(8) "Why do you prefer this method?"
(9) "Is there anything you wish the system would do but currently does not?"
(10) "Do you have any other comments?"

User study guidelines: Study 1 and study 2

Introduction. Welcome to our user study on human preference learning! In this study, you will interact with a robot and provide feedback based on your preferences. Please note that there will be no physical contact with the robot.

Task description. The robot is situated in a kitchen setup and needs to pick up a spoon from the wall. However, there is a pan on the stove cooking something. For safety reasons, you should aim to have the robot successfully pick up the spoon while avoiding the pan. You can also adjust the robot's speed based on your preference.

Pre-study survey. Before we begin, please complete a pre-study survey about your background and understanding of robotics and AI systems.

Experiment 1: Improve trajectory. In this experiment, the robot will first execute a suboptimal trajectory. Your task is to improve its behavior by providing comparative language feedback, such as "avoid the pan better." After each iteration, you will be asked if you are satisfied with the current trajectory. The experiment ends when you are satisfied with the trajectory or reach the maximum number of iterations. A

post-study survey will be required after completing this experiment.

Experiment 2: Preference learning. In this experiment, you will be shown a series of queries and provide feedback on each one. Through this process, we will gradually learn your preferences and present you with the best trajectories based on the learned preferences. You will compare two approaches: language preference learning and pairwise preference learning.

(1) Language Preference Learning
   - In each query, you will be shown one trajectory. You need to give comparative language feedback based on your preferences, such as "move faster," "avoid the pan better," or "be more adept at picking up the spoon." After every 5 queries, you will be shown the best trajectory based on the currently learned preferences and asked to rate it. You will complete a total of 20 queries.

(2) Pairwise Preference Learning
   - In each query, you will be shown a pair of trajectories. You need to choose your preferred one. After every 5 queries, you will be shown the best trajectory based on the currently learned preferences and asked to rate it. You will complete a total of 20 queries.

After completing the experiment, you will be required to fill in a post-study survey.

User study guidelines: Study 3

Introduction. Welcome to our user study on human preference learning! In this study, you will interact with a WidowX 250s robot and provide comparative language feedback based on your preferences. There will be no physical contact with the robot.

Task description. The robot is situated in a kitchen setup and needs to pick up a spoon from the wall. However, there is a pan on the stove cooking something. For safety reasons, you should aim to have the robot successfully pick up the spoon while avoiding the pan. You can provide feedback according to your preference, that is, suggest changes to the robot's position to the spoon, speed, distance from the pan, etc.

Pre-study survey. Before we begin, please complete a pre-study survey about your background and understanding of robotics and AI systems.

Preference learning. In this experiment, you will be shown a series of queries and provide feedback on each one. Through this process, we will gradually learn your preferences and present you with the best trajectories based on the learned preferences. You will compare two different approaches. In both approaches, the experiment will run as follows:

In each query, you will be shown one robot trajectory. You need to give comparative language feedback based on your preferences, such as "move faster," "avoid the pan better," or "be more adept at picking up the spoon." After every 5 queries, you will be shown the best trajectory based on the currently learned preferences and asked to rate it. You will complete giving feedback for a total of 20 queries and rating 4 trajectories.

Post-study survey. After completing each set (20 queries +4 trajectories), please complete a post-study survey detailing your experience and thoughts for that set.

Original language feedback utterances

All the original language feedback for each of the simulation environments are listed here. These sentences were then augmented with GPT-3.5.

Robosuite:

- Distance to the cube
  - Move farther from the cube.
  - Move further from the cube.
  - Move more distant from the cube.
  - Move less nearby from the cube.
  - Move nearer to the cube.
  - Move closer to the cube.
  - Move more nearby to the cube.
  - Move less distant to the cube.
- Distance to the bottle
  - Move further from the bottle.
  - Move farther from the bottle.
  - Move more distant from the bottle.
  - Move less nearby from the bottle.
  - Move nearer to the bottle.
  - Move closer to the bottle.
  - Move more nearby to the bottle.
  - Move less distant to the bottle.
- Height of the robot arm
  - Move taller.
  - Move at a greater height.
  - Move higher.
  - Move to a greater height.
  - Move lower.
  - Move at a lesser height.
  - Move shorter.
  - Move to a lower height.
- Speed of the robot arm
  - Move quicker.
  - Move swifter.

  - Move at a higher speed.
  - Move faster.
  - Move more quickly.
  - Move at a lower speed.
  - Move more moderate.
  - Move slower.
  - Move more sluggish.
  - Move more slowly.
- Proficiency at cube-lifting
  - Lift the cube better.
  - Lift the cube more successfully.
  - Lift the cube more effectively.
  - Lift the cube worse.
  - Lift the cube not as well.
  - Lift the cube less successfully.

Meta-World:

- Height of the robot arm
  - Move higher.
  - Move more up.
  - Move higher up from the table.
  - Increase the overall height of the trajectory.
  - Go higher up.
  - Move your gripper higher.
  - Don't go down, instead go up.
  - Stay higher and farther from the table.
  - Move your hand up as you perform the task.
  - Make sure to stay higher above the table, rather than lower.
  - Move lower.
  - Move more down.
  - Move lower to the table.
  - Decrease the overall height of the trajectory.
  - Go lower down.
  - Move your gripper lower.
  - Don't go up, instead go down.
  - Stay lower and nearer to the table.
  - Move your hand lower as you perform the task.
  - Make sure to go lower to the table, rather than higher.
- Speed of the robot arm
  - Move faster.
  - Move at a quicker speed.
  - Increase the pace.
  - Press the button faster.
  - Increase your velocity.
  - Move your gripper faster.
  - Move to the button faster.
  - Don't go too slowly, instead go quickly.
  - Move your hand faster as you perform the task.
  - Make sure to go much faster, rather than slower.
  - Move slower.
  - Move at a more sluggish speed.
  - Decrease the pace.
  - Press the button slower.
  - Decrease your velocity.
  - Move your gripper slower.
  - Move to the button slower.
  - Don't go too fast, instead go more slowly.
  - Move your hand slower as you perform the task.
  - Make sure to go much slower, rather than faster.
- Distance to the button
  - Move farther from the button.
  - Increase distance from the button.
  - Stay farther from the button.
  - Give wider berth to the button.
  - Keep a larger distance from the button.
  - Keep your gripper farther away from the button.
  - Move your gripper away from the button.
  - Don't go toward the button, instead move away from it.
  - Move your hand farther from the button on the wall as you perform the task.
  - Make sure to go much farther from the button, rather than closer to the button.
  - Move closer to the button.
  - Decrease distance from the button.
  - Stay closer to the button.
  - Get closer to the button.
  - Keep a smaller distance to the button.
  - Keep your gripper closer to the button.
  - Move your gripper so that it is closer to the button.
  - Don't go away from the button, instead go toward it.
  - Move your hand closer to the button on the wall as you perform the task.
  - Make sure to go much closer to the button, rather than farther from the button.