Learning Multimodal Rewards from Rankings

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning from human feedback has shown to be a useful approach in acquiring robot reward functions. However, expert feedback is often assumed to be drawn from an underlying *unimodal* reward function. This assumption does not always hold including in settings where multiple experts provide data or when a single expert provides data for different tasks—we thus go beyond learning a unimodal reward and focus on learning a *multimodal* reward function. We formulate the multimodal reward learning as a mixture learning problem and develop a novel *ranking*-based learning approach, where the experts are only required to rank a given set of trajectories. Furthermore, as access to interaction data is often expensive in robotics, we develop an *active* querying approach to accelerate the learning process. We conduct experiments and user studies using a multi-task variant of OpenAI's LunarLander and a real Fetch robot, where we collect data from multiple users with different preferences. The results suggest that our approach can efficiently learn multimodal reward functions, and improve data-efficiency over benchmark methods that we adapt to our learning problem.

Keywords: HRI, reward learning, multi-modality, rankings, active learning

1 Introduction

Learning a reward function from different sources of human data is a fundamental problem in robot learning. In recent years, there has been a large body of work that learns reward functions using various forms of human input, such as expert demonstrations [1], suboptimal demonstrations [2, 3, 4], pairwise comparisons [5, 6], physical corrections [7, 8], rankings [9], and trajectory assessments [10]. These works focus on learning a *unimodal* reward function that models human preferences on a target task. However, this unimodality assumption does not always hold: human preferences are usually more complex and need to be captured via a multimodal representation. Further, even if the preferences of a human are truly unimodal, we often use a mixture of data from multiple humans, which can be difficult to disentangle, leading to multimodality.

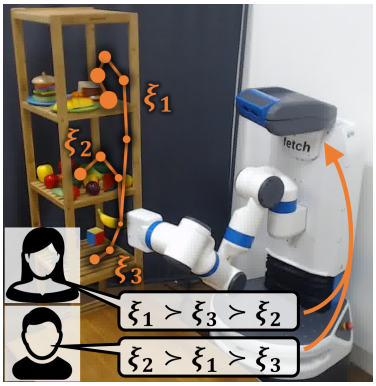


Figure 1: Fetch robot putting a banana on one of the three shelves. The two users have different preferences, and so they provide different rankings to the robot. The robot needs to be able to model multimodal reward functions for successfully achieving the task.

As an example, consider a robot placing a banana on one of the three shelves (see Fig. 1). The middle shelf is often used for fruits, but it has no room left and if the robot tries to put the banana there, it may cause other fruits to fall. The top shelf has some space but it has been used for cooked meals. The bottom shelf has a lot of free space, but is usually used only for toys. In such a scenario, people may have very different preferences about what the robot should do. If we try to learn a unimodal reward using data collected from multiple people, the robot is likely to fail in the task, because the data will include inconsistent preferences.

One solution is of course to label the different modes in the data. For example, one could separate the data based on the preferred shelf, and learn different reward functions for each shelf. However, this separation is not always straightforward. For example in a driving dataset, it is unclear what should

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

be labeled as aggressive or timid driving. Clustering the data based on the human who provided the data is also not viable as it will introduce data-inefficiency issues, and perhaps more importantly, humans are not always unimodal: a usually timid driver can drive more aggressively when in a hurry.

These examples motivate us to develop methods that can learn *multimodal* reward functions using datasets that are not specifically labeled with the modes. To this end, previous work proposed learning from demonstrations to learn multimodal policies [11, 12]. However, learning from expert demonstrations is often extremely challenging in robotics as providing demonstrations on a robot with high degrees of freedom is non-trivial [5, 13], and humans have difficulty giving demonstrations that align with their preferences due to their cognitive biases [14, 15]. Thus, it is desirable to have methods that learn from other more reliable sources of data. For instance, humans can reliably compare two different trajectories, enabling a robot to learn from pairwise comparisons [5, 9, 16].

While learning from pairwise comparisons provides a rich source of data for learning reward functions, the theoretical results by Zhao et al. [17] imply that extending the existing comparison-based reward learning techniques to multimodal reward functions is not possible, i.e. failure cases can be constructed, where pairwise comparisons are not sufficient for identifying different modes of a multimodal function. Our insight is that it is possible to learn a multimodal reward function by going beyond pairwise comparisons and instead using *rankings*.

To achieve this, we formulate multimodal reward learning as a mixture learning problem. As data is a very expensive resource in robotics, we further develop an *active querying* method that aims to ask the human users the most informative queries. Our contributions are three-fold:

- We develop a method that uses rankings from humans to learn multimodal reward functions.
- We develop an active querying method to improve data-efficiency by asking the most informative ranking queries.
- We conduct extensive experiments and user studies with OpenAI's LunarLander and a real Fetch robot to test our learning and querying methods in comparison to baselines.

2 Related Work

Reward Learning in Robotics. Learning reward functions from human feedback is a fundamental problem in robot learning. Ng and Russell [18] and Abbeel and Ng [19] introduce the problem of learning from demonstrations in the space of robotics. Later works focus on improving *inverse reinforcement learning* algorithms by reducing the ambiguity in learned rewards [1, 20].

Due to the difficulty in providing expert demonstrations in robotics, recent works attempt to learn reward functions using other forms of human feedback, such as physical corrections [7, 8] and pairwise comparisons where a human user compares the quality of two robot trajectories [21, 22, 23, 24]. Later works extend these algorithms for better time and data-efficiency [25, 26, 27]. Though there have been works to extend this framework to reward functions modeled with Gaussian processes to capture nonlinearities [6, 28, 29], the underlying reward function has always been unimodal.

Preference-based Learning. Outside of robotics, preference-based learning, where data are in the form of comparisons, selections out of a set, or rankings, has attracted attention due to the ease of collecting data and its reliability. Prior works have studied this in the classification [30], bandits [31], and reinforcement learning settings [32].

In our problem, we have a continuous hypothesis space of reward functions. In such cases, it is common to model human comparisons or rankings with a computational model. Bradley-Terry is one such model for pairwise comparisons [33], which is easily extended to the queries where the human chooses the best of multiple items. Known as multinomial logits (MNL) [34], this model has been widely used for human preferences in many fields [35, 36, 37] including robotics [23, 25, 16].

To extend these models to rankings, Plackett-Luce [38, 39] and Mallows models [40, 41, 42, 43] are commonly employed. In this paper, we use the Plackett-Luce model as it is a natural extension of MNL, which is widely used in robotics with great success. We formalize this model in Section 3.

Even though there has been much research in this domain, all works we mentioned here focus on the unimodal case, and do not work with the multimodal preferences.

Learning Mixture Models from Rankings. One way to model multimodal reward functions is through mixture models, where the data is assumed to come from different individual models with some unknown probabilities. To this end, previous works consider mixtures of MNLs [44, 45],

Plackett-Luce models [17], and Mallows models [46]. Other works adopt different methods to model multimodality, such as by assuming latent state dynamics that transition between different modes [47, 48] or by learning the different modes from labeled datasets [49, 50]. To avoid these modeling assumptions, we focus on directly learning the mixture model.

While Zhao et al. [17] have theoretically studied the mixture of Plackett-Luce choice models, which also informs our algorithm in terms of the query sizes, they only focus on learning the rewards of a discrete set of items. To the best of our knowledge, our paper is the first work that deals with a continuous hypothesis space under a mixture of Plackett-Luce models. Furthermore, we propose an active querying strategy for this mixture model to improve data-efficiency for human-in-the-loop learning, which is crucial in data-hungry applications such as robotics.

3 Problem Formulation

Setup. We consider a fully-observable deterministic dynamical system. A trajectory ξ in this system is a series of states and actions, i.e., $\xi = (s_0, a_0, \ldots, s_T, a_T)$. The set of feasible trajectories is Ξ.

We assume there is a set of M individual reward functions that are possibly different, each of which encodes some preference between the trajectories in Ξ. For the rest of the formalism, we refer to each individual reward function as an *expert* for the clarity of the presentation.

Following the common linearity assumption in reward learning [1, 5, 26], we assume each preference can be modeled as a linear reward function over a known fixed feature space Φ, so the reward associated with a trajectory ξ with respect to the mth expert is $R_m(\xi) = \omega_m^\top \Phi(\xi)$, where ω_m is the unknown vector of weights. Across the expert population, there exists some unknown distribution over the reward parameters, corresponding to the ratio of the data provided by the experts. We represent this distribution with mixing coefficients α_m such that $\sum_{m=1}^{M} \alpha_m = 1$. We will then learn both the unknown reward functions $\{\omega_m\}_{m=1}^{M}$ and the mixing coefficients $\{\alpha_m\}_{m=1}^{M}$, using ranking queries made to the M experts.

Ranking Model. We define a *ranking query* to be a set of the form $Q = \{\xi_1, \ldots, \xi_K\}$ for a fixed query size K. The response to a ranking query is a ranking over the items contained therein, of the form $x = (\xi_{a_1}, \ldots, \xi_{a_K})$, where a_1 is the index of the expert's top choice, a_2 is the second top choice, and so on. While it is not known which expert provided the response to the query, we know the prior that a response comes from expert m with some unknown probability α_m, i.e., $\Pr(R = R_m) = \alpha_m$. Going back to our banana placing example, a ranking query of K robot trajectories is generated by the algorithm, and a user—whose identity is unknown to the algorithm—responds to this query.

We then capture how human experts respond to these ranking queries by modeling a ranking distribution through an iterative process using Luce's choice axiom [51]. In this process, the experts repeatedly select their top choice a_m with a probability distribution generated with the softmax rule to generate a ranking from the order items were selected:

$$\Pr\left(x_1 = \xi_{a_1} \mid R = R_m\right) = \frac{e^{R_m(\xi_{a_1})}}{\sum_{j=1}^{K} e^{R_m(\xi_{a_j})}} \,.$$

In the following iterations, the experts select their top choice among the remaining trajectories:

$$\Pr\left(x_i = \xi_{a_i} \mid x_1, \ldots, x_{i-1}, R = R_m\right) = \frac{e^{R_m(\xi_{a_i})}}{\sum_{j=i}^{K} e^{R_m(\xi_{a_j})}} \,. \tag{1}$$

This is known as the Plackett-Luce ranking model [38, 39]. Together with the prior over experts α_m, the resulting distribution over rankings $x \sim X$ is a mixture of Plackett-Luce models with mixing coefficients α_m and weights proportional to $e^{R_m(\xi)}$.

Hence, the ranking distribution first selects the reward function R_m with probability α_m, and then selects trajectories from Q sequentially with probability proportional to the exponent of their reward, i.e., e^{R_m}, among the remaining trajectories until none is left, generating a ranking of the trajectories.

So given knowledge of the true reward function weights ω_m and mixing coefficients α_m, we have the following joint mass over observations x from a query Q:

$$\Pr(x \mid Q) = \sum_{m=1}^{M} \alpha_m \prod_{i=1}^{K} \frac{e^{\omega_m^\top \Phi(\xi_{a_i})}}{\sum_{j=i}^{K} e^{\omega_m^\top \Phi(\xi_{a_j})}} \,. \tag{2}$$

Objective. Our goal is to design a series of adaptive queries $Q^{(t)}$ to optimally learn the reward weights ω_m and corresponding mixing coefficients α_m upon observing the query responses $x^{(t)}$. We constrain all queries to consist of a fixed number of elements K.

4 Active Learning of Multimodal Rewards from Rankings

In this section, we first start with presenting our learning framework. We then discuss how we can improve data-efficiency, and propose an active querying approach.

4.1 Learning from Rankings

To learn the reward weights ω_m and mixing coefficients α_m, we adopt a Bayesian learning approach. For this, we maintain a posterior over the parameters ω_m and α_m. Denoting the distribution over the parameters α_i and ω_i as Θ, this posterior can be written as

$$\begin{aligned}\Pr(\Theta \mid Q^{(1)}, x^{(1)}, Q^{(2)}, x^{(2)}, \dots) &\propto \Pr\Big(\Theta \mid Q^{(1)}, Q^{(2)}, \dots\Big) \Pr\Big(x^{(1)}, x^{(2)}, \cdots \mid \Theta, Q^{(1)}, Q^{(2)}, \dots\Big) \\ &= \Pr(\Theta) \Pr\Big(x^{(1)}, x^{(2)}, \cdots \mid \Theta, Q^{(1)}, Q^{(2)}, \dots\Big) = \Pr(\Theta) \prod_t \Pr\Big(x^{(t)} \mid \Theta, Q^{(t)}\Big), \end{aligned} \tag{3}$$

where we use the conditional independence of ranking queries in the last equation and the likelihood terms come from Eq. (2). To be able to compute this posterior, we assume some prior distribution over the reward weights and the mixing coefficients, which is system-dependent and may come from domain knowledge. For example, in our simulations and user studies, we adopted a Gaussian prior $\omega_i \sim \mathcal{N}(0, I)$ and a uniform prior $\alpha \sim \mathrm{Unif}(\Delta_{M-1})$ where Δ_{M-1} is the unit $M-1$ simplex. Learning this posterior distribution in Eq. (3), one can compute a maximum likelihood estimate (MLE) or expectation as the predicted reward weights and mixing coefficients.

Equation (3) implies the queries made to the experts, $Q^{(t)}$'s, affect how well the posterior will be learned. Assuming a limited budget of queries, which is often the case in many real-world applications, including robotics, one would ideally find an optimal adaptive sequence of queries such that the responses would give the highest amount of information about the reward weights and the mixing coefficients. However, this is NP-hard, even in the unimodal case with pairwise comparisons [52]. We therefore resort to greedy optimization techniques to develop our active learning approach.

4.2 Active Querying via Information Gain

A query Q is desirable if observing its value x yields high information about the underlying model parameters, α_m and ω_m. Therefore, we propose using an information gain objective to adaptively select the most informative query at each querying step.

Assume at a fixed timestep t we have made past query observations $\mathcal{D} = \big\{Q^{(t')}, x^{(t')}\big\}_{t'=1}^{t-1}$. The desired query is then $Q^* = \arg\max_Q I(Q; \Theta \mid \mathcal{D})$ where $I(\cdot;\cdot)$ denotes mutual information. Equivalently, denoting conditional entropy with $\mathcal{H}[\cdot \mid \cdot]$ and the joint distribution over x and $\theta = \{\alpha_m, \omega_m\}_{m=1}^M$ conditioned on Q and $\mathcal{D}$ as $P(X, \Theta \mid Q, \mathcal{D})$, we see

$$Q^* = \arg\min_Q \mathop{\mathbb{E}}_{P(X,\Theta \mid Q,\mathcal{D})} \log \frac{\mathbb{E}_{\theta' \sim \Theta \mid \mathcal{D}} \Pr\left[X = x \mid Q, \theta'\right]}{\Pr[X = x \mid Q, \theta]}. \tag{4}$$

The details of this derivation are presented in Appendix A.

4.3 Overall Algorithm

To efficiently solve the optimization in Eq. (4), we first note that we should use a Monte Carlo approximation since the expectations are taken over a continuous variable Θ and a discrete variable X over an intractably large set of $K!$ alternatives. To perform this Monte Carlo integration, we require samples from the posterior $\Pr(X, \Theta \mid Q, \mathcal{D})$.

Our key insight is that we can obtain joint samples from both posteriors by first sampling from $\bar{\Theta} \sim \Pr(\Theta \mid \mathcal{D})$ and then sampling $x \sim \Pr\big(X \mid Q, \Theta = \bar{\Theta}\big)$ since $\Theta \perp Q \mid \mathcal{D}$ and $X \perp \mathcal{D} \mid Q, \Theta$. We perform the sampling $x \sim \Pr\big(X \mid Q, \Theta = \bar{\Theta}\big)$ efficiently using Eq. (2). In general, exact sampling from the posterior $\Pr(\Theta \mid \mathcal{D})$ is intractable. However, we note Eq. (3) can be directly evaluated (using Eq. (2)) and gives $\Pr(\Theta \mid \mathcal{D})$ up to a proportionality constant factor.

With this unnormalized posterior of Eq. (3), we use the Metropolis-Hastings algorithm as described in Appendix B to generate samples from the posterior $\Pr(\Theta \mid \mathcal{D})$.

We see our optimization problem simplifies to finding, for N fixed samples $\bar{\theta}_i \sim \Pr(\Theta \mid \mathcal{D})$ and corresponding samples: $x_i \sim \Pr\big(X \mid Q, \bar{\theta}_i\big)$

$$\mathcal{L}(Q; x, \bar{\theta}) = \sum_{i=1}^{N} \Bigg[\log\Big(\sum_{j=1}^{N} \Pr\big[x_i \mid Q, \bar{\theta}_j\big] \Big) - \log \Pr\big[x_i \mid Q, \bar{\theta}_i\big] \Bigg], \quad Q^* = \arg\min_{Q} \mathcal{L}(Q; x, \bar{\theta}) \,. \quad (5)$$

We solve this optimization using simulated annealing [53] (see Appendix C). Algorithm 1 goes over the pseudocode of our approach, and we discuss the hyperparameters in our experiments in Appendix D.

Algorithm 1 Active Querying via Information Gain

Require: Observations $\mathcal{D}$

$\{\bar{\theta}_i\}_{i=1}^{N} \sim \Pr(\Theta \mid \mathcal{D})$ w/ Eq. (3) via Metropolis-Hastings
procedure EVALQUERY(Q)
  $\forall i, x_i \sim \Pr\big(x_i \mid Q, \bar{\theta}_i\big)$
  return $\mathcal{L}(Q; x, \bar{\theta})$ ▷ Eq. (5)
end procedure
$Q \leftarrow$ MINIMIZE(EVALQUERY) ▷ Simulated annealing
select query Q

4.4 Analysis

We start the analysis by stating the bounds on the required number of trajectories in each ranking query to achieve *generic identifiability*. A Plackett-Luce model over Ξ is generically identifiable if for any sets of parameters Θ_1 and Θ_2 inducing the same distribution over the responses to all queries of size K on Ξ, the mixing coefficients of Θ_1 and Θ_2 are the same and the induced rewards $R_m(\xi)$ are identical across Ξ up to a constant additive scaling factor.

Theorem 1 (Zhao et al. [17])**.** *A mixture of M Plackett-Luce models with query size K and $|\Xi| = K$ is generically identifiable if $M \leq \lfloor \frac{K-2}{2} \rfloor!$.*

This statement follows directly from [17], which proves the above bound assuming that each query to the Plackett-Luce mixture is a full ranking over the set of items (i.e. $|\Xi| = K$). However, the assumption $|\Xi| = K$ is untenable in the active learning context, as it prevents any active query selection. To apply this result for our active learning algorithms, we relax the condition to $|\Xi| \geq K$.

Corollary 1.1. *A mixture of M Plackett-Luce models with query size K is* generically identifiable *if $M \leq \lfloor \frac{K-2}{2} \rfloor!$.*

We prove Corollary 1.1 in Appendix E.1. In our context, generic identifiability implies if the human response is modelled by a Plackett-Luce mixture, our Algorithm 1 will be able to recover its true parameters (up to a constant additive factor for the rewards) in the limit of infinite queries.

Remark 1. *Greedy selection of queries maximizing information gain in Eq. (4) is not necessarily within a constant factor of optimality.*

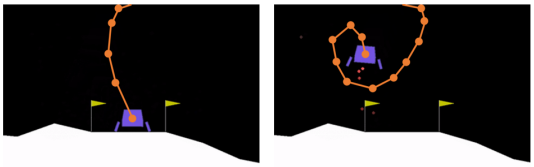
Appendix E.2 justifies Remark 1. In fact, greedy optimization of information gain for adaptive active learning can be significantly worse than a constant factor of optimality in pathological settings [54]. Despite its lack of theoretical guarantees, information gain is a commonly used effective approach in adaptive active learning [27, 55, 56]. Although other approaches like volume removal satisfy adaptive submodularity [23], they fail in settings with noisy observations by selecting high-noise low-information queries, and in practice achieve far worse performance than information gain.

5 Experiments

Having presented our learning and active querying algorithms, we now evaluate their performance in comparison with other alternatives. We start with describing the two tasks we experimented with:

LunarLander. We used 1000 trajectories in OpenAI Gym's LunarLander environment [57] shown in Fig. 2 (see Appendix G.1 for details on how they were generated).

Fetch Robot. We generated 351 distinct trajectories of the Fetch robot [58] putting the banana on the shelves as shown in Fig. 1 (see Appendix G.2 for details).



(a) Landing softly on the landing pad *(b) Staying in the air as long as possible*

Figure 2: The LunarLander environment is visualized with the two tasks. Sample trajectories associated with these tasks are shown.

5.1 Methods

We compare our active querying via information gain (IG) discussed in Algorithm 1 with two baselines: A simple benchmark for active learning is *random* query selection without replacement. We

also benchmark against *volume removal* (VR), a common objective for active learning of robot reward functions [23]. See Appendix F for the details of these two baselines.

5.2 Metrics

We want to evaluate both the active querying and the learning performance. The former requires metrics that assess the quality of the algorithm's selected queries $\mathcal{D} = \left\{ (Q^{(t)}, x^{(t)}) \right\}_t$ in terms of the information they provide on the model parameters Θ. We use two such metrics: mean squared error (MSE) and log-likelihood. Since both active and non-active methods are expected to reach the same performance with a large number of queries, we look at the area under the curve (AUC) of these two metrics over number of queries. To evaluate the learning performance, we quantify the success of a robot, which learned a multimodal reward, via the *learned policy rewards* on the actual task.

MSE. Suppose we know the human is truly modeled by Θ^* adhering to the assumed model class of Section 3. Given a set of observations $\mathcal{D}$, we can compute an MLE estimate $\widehat{\Theta}$ of the model parameters using Eq. (2). The MSE is then the squared error between $\widehat{\Theta}$ and Θ^* (see Appendix H.1).

While this metric cannot be evaluated with real humans, we can use this metric with synthetic human models (model with known parameters Θ^*) in simulation. A lower MSE score means the selected queries $\mathcal{D}$ allow us to better learn a multimodal Plackett-Luce model close to the true model Θ^*.

Log-Likelihood. The log-likelihood metric measures the log-likelihood of the response to a random query given the past observations $\mathcal{D}$. If the past observations $\mathcal{D}$ are informative, the true response to a random query Q will in expectation be more likely, meaning the log-likelihood metric will be greater. See Appendix H.2 for details on how we compute this metric.

Learned Policy Reward. We take the MLE estimate of each reward weights vector and train a DQN policy using them [59].[1] We then run these learned policies on the actual environment with the corresponding true reward functions (see Appendix H.3) to obtain the learned policy rewards.

5.3 Results

Multimodal Learning is Necessary. We first compare unimodal and multimodal models to show the insufficiency of unimodal rewards when the data come from a mixture. To leave out any possible bias due to active querying, we make this comparison using random querying.

We let the true reward function have $M = 2$ modes and set a query size of $K = 6$ items for identifiability as Section 4.4 suggests, and for acquiring high information from each query. We simulate 100 pairs of experts whose reward weights ω_m and the mixing coefficients α_m are sampled from the prior $\Pr(\Theta)$. Having these simulated experts respond to 15 queries, we report the MSE in Fig. 3.

The unimodal reward model causes an unstably increasing MSE. This is mostly due to the outliers where the reward weights ω_1 and ω_2 are far away from each other and the unimodal reward fails to learn any of them. We therefore also plot the median values and quartiles in Fig. 3. While the bimodal reward model learned using our proposed approach decreases the MSE over time, the unimodal model has a roughly constant MSE, which suggests it is unable to learn when the data come from a mixture.

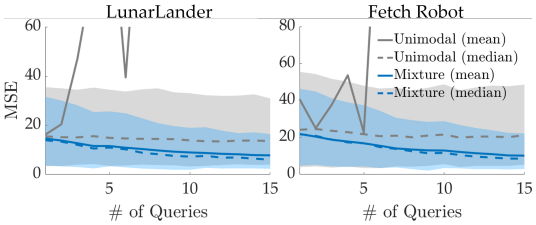


Figure 3: Unimodal and bimodal reward learning models are compared under MSE. Both mean and median values (over 100 runs) are shown. Shaded regions show the first and the third quartiles.

Active Querying with Information Gain is Data-Efficient. We next compare our information-gain-based active querying approach with the other baselines. For this, we use the same experiment setup as above with $M = 2$ reward function modes and ranking queries of size $K = 6$, and simulate 75 pairs of human experts. We present the results in terms of MSE in Fig. 4. In LunarLander, the information gain objective significantly outperforms both random querying and volume removal in terms of the AUC MSE ($p < 0.005$, paired-sample t-test). Notably, volume removal performs even worse than the random querying method, which might be due to the known issues of volume removal optimization as briefly discussed in Appendix F.2. On the other hand, the difference is not

[1] As we are using a real Fetch robot for our experiments and it would be infeasible and unsafe to train DQN on Fetch, so this metric is limited to our simulations, i.e., LunarLander in our experiments.

statistically significant in the Fetch Robot experiment, which might be due to the small trajectory dataset for this experiment, or because almost all trajectories in the dataset minimize or maximize some of the trajectory features, accelerating and simplifying learning under the linear reward assumption. See Appendix G.2 for details about the trajectory features and how we generated the trajectory dataset.

We further analyze the querying methods in this multimodal setting under the log-likelihood metric in Fig. 4. Information gain significantly outperforms random querying and volume removal in both experiments with respect to the AUC log-likelihood ($p < 0.005$). With respect to the final log-likelihood, information gain reduces the amount of required data in LunarLander by about 35% compared to random querying and about 60% to volume removal. Similarly in the Fetch Robot, the improvement is approximately 25% over both baselines.

Appendix J presents an additional experiment which clearly shows the effectiveness of our approach for learning a mixture of more than two reward functions (specifically, $M = 5$).

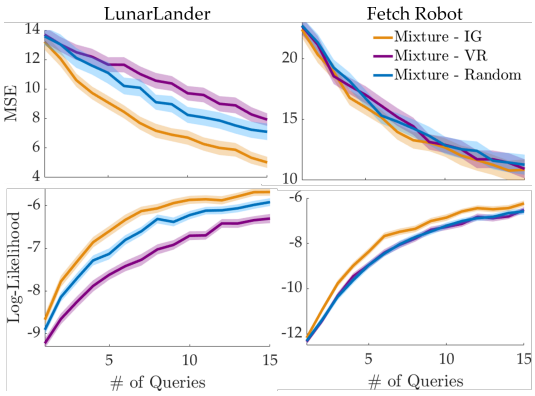


Figure 4: Different querying methods are compared with the (top) MSE and (bottom) log-likelihood metrics (mean±se over 75 runs).

Information Gain Leads to Better Learning. Having seen the superior predictive performance of the reward learned via information gain optimization, we next assess its performance in the actual environment. As random querying outperforms volume removal in terms of log likelihood and MSE as in Fig. 4, we compare the information gain with random querying.

For this, we run the multimodal reward learning with 75 pairs of randomly generated reward weights ($M = 2$ and $K = 6$). For each of the 150 individual reward functions, we compute the learned policy rewards. Fig. 5 shows the results. While the standard errors in the plots seem high, this is mostly because optimal trajectories for different reward weights differ substantially in terms of rewards, which causes an irreducible variance. However, since the underlying true rewards are the same between the information gain and random querying methods, we ran the paired sample t-test between the results and observed statistical significance ($p < 0.05$). This means although the learned policy rewards between different runs differ substantially, the reward function learned via the information gain method leads to better task performance compared to random querying.

6 User Studies

We now empirically analyze the performance of our algorithm with two online user studies.[2] We again used the LunarLander and Fetch Robot environments. We provide a summary and a video of the user studies and their results at `https://sites.google.com/view/multimodal-reward-learning/`.

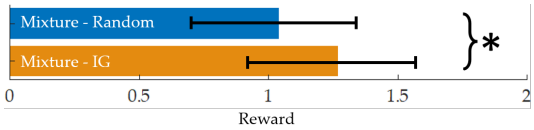


Figure 5: Information gain and random querying methods are compared with the learned policy rewards (mean±se over 75 runs which correspond to 150 randomly generated reward weights) in LunarLander.

Experimental Setup. For LunarLander, subjects were presented with either of the following instructions at every ranking query: "Land softly on the landing pad" or "Stay in the air as long as possible". We randomized these instructions such that users get one of them with 0.6 and the other with 0.4 probability. We kept the presented instructions hidden from the learning algorithms so that they need to learn a multimodal reward without knowing which mode each ranking belongs to.

For the Fetch robot environment, we recorded the 351 trajectories on the real robot as short video clips so that the experiment can be conducted online under the pandemic regulations. Human subjects participated in the experiment as groups of two to test learning from multiple users. Each participant was instructed that the robot needs to put the banana in one of the shelves and different shelves have different conditions (the same as in our running example, see Fig. 1, Appendix I.1).

[2] We have IRB approval from a research compliance office under the protocol number IRB-52441.

After emphasizing there is no one correct choice and it only depends on their preferences, we asked each participant to indicate their preferences between the shelves on an online form. Afterwards, each group of two subjects responded to 30 ranking queries in total where each query consisted of 6 trajectories. We selected who responds to each query randomly, with probabilities 0.6 and 0.4.

Appendix I.2 presents details on the user interface used in our experiments.

Independent Variables. We varied the querying algorithm: active with information gain and random querying. We excluded the volume removal method to reduce the experiment completion time for the subjects, as it already performed worse than random querying in our simulations.

Procedure. We conducted the experiments as a within-subjects study. We recruited 24 participants (ages 19 – 56; 9 female, 15 male) for LunarLander and 26 participants (ages 19 – 56; 11 female, 15 male) for the Fetch robot. Each subject in the LunarLander, and each group of two subjects in the Fetch robot experiment responded to 40 ranking queries; 15 with each algorithm and 10 random queries for evaluation at the end. The order of the first 30 queries was randomized to prevent bias.

Dependent Measures. Learning the multimodal reward functions via the 15 rankings collected by each algorithm, we measured the log-likelihood of the final 10 rankings collected for evaluation.

Hypotheses. With LunarLander and Fetch robot, we test the following hypotheses respectively:
H1. *Querying the participants, who are trying to teach two different tasks, actively with information gain will lead to faster learning than random querying.*
H2. *While learning from two people with different preferences, active querying with information gain will lead to faster learning than random querying.*

Results. Figure 6 visualizes how log-likelihood of the evaluation queries changes over the course of learning by both algorithms. Active querying with information gain leads to significantly faster learning compared to random querying in LunarLander. Indeed, the difference in AUC log-likelihood is statistically significant ($p < 0.05$). Furthermore, the active querying method enabled reaching the final performance of random querying after only 9 or 10 queries, for around a 35% reduction in the amount of data needed, supporting **H1**.

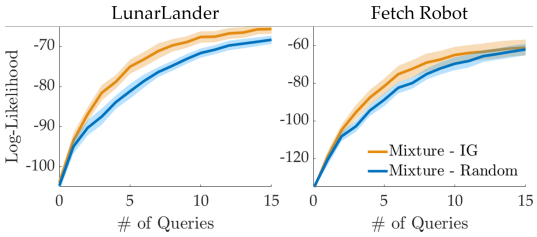


Figure 6: User study results (mean±se over 24 users for LunarLander and 13 groups for Fetch Robot).

As the robot experiments have a an easier task with a small number of variables between the trajectories, both querying methods converge to similar performances by the end of 15 queries. However, active querying accelerates learning in the early stages—the difference in AUC log-likelihood is again statistically significant ($p < 0.05$). Looking at the final performance with random querying, improvement in data efficiency is about 10%, supporting **H2**.

7 Conclusion

Summary. This work presents a novel approach for learning multimodal reward functions. We formulated the problem as a mixture learning problem solved using ranking queries that are answered by experts. We further developed an active querying method that maximizes information gain to improve the quality of ranking queries made to the experts. The results suggest our model learns multimodal reward functions, with data efficiency improved by our new active querying method.

Limitations and Future Work. Our model for learning multimodal rewards requires knowing the number of different modes (experts or tasks) M in advance. This might be difficult in some settings. For example, when several experts belonging to different clusters, e.g., timid and aggressive drivers, provide data, it might be difficult to know the number of clusters in advance. However, a simple approach that fits the multimodal reward under various M could reveal the true number of underlying modes. In addition, theoretical results assert, to guarantee the reliable learning of a multimodal reward with M modes, ranking queries should consist of K queries such that $M \leq \lfloor \frac{K-2}{2} \rfloor!$. While this is manageable by multiple pairwise comparisons for each query or an iterative process where the expert selects the top item, it might consume too much time for large M. Thus, an interesting future direction is to investigate how to incorporate multiple forms of expert feedback, e.g., demonstrations in addition to rankings, to pretrain and reduce the required interaction time with humans.

References

[1] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438, 2008.

[2] Z. Cao and D. Sadigh. Learning from imperfect demonstrations with varying dynamics. *IEEE Robotics and Automation Letters*, 2021.

[3] D. H. Grollman and A. Billard. Donut as i do: Learning from failed demonstrations. In *ICRA*, 2011.

[4] Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama. Imitation learning from imperfect demonstration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6818–6827. PMLR, June 2019.

[5] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *arXiv preprint arXiv:2006.14091*, 2020.

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

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

[8] M. Li, A. Canberk, D. P. Losey, and D. Sadigh. Learning human objectives from sequences of physical corrections. In *International Conference on Robotics and Automation (ICRA)*, 5 2021.

[9] D. Brown, W. Goo, P. Nagarajan, and S. Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pages 783–792. PMLR, 2019.

[10] A. Shah and J. Shah. Interactive robot training for non-markov tasks. *arXiv preprint arXiv:2003.02232*, 2020.

[11] K. Hausman, Y. Chebotar, S. Schaal, G. Sukhatme, and J. Lim. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In *31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, pages 1236–1246, 2018.

[12] C. Fei, B. Wang, Y. Zhuang, Z. Zhang, J. Hao, H. Zhang, X. Ji, and W. Liu. Triple-gail: A multi-modal imitation learning framework with generative adversarial nets. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2929–2935, July 2020.

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

[14] C. Basu, Q. Yang, D. Hungerman, M. Singhal, and A. D. Draqan. Do you want your autonomous car to drive like you? In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pages 417–425, 2017.

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

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

[17] Z. Zhao, P. Piech, and L. Xia. Learning mixtures of plackett-luce models. In *International Conference on Machine Learning*, pages 2906–2914. PMLR, 2016.

[18] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *in Proc. 17th International Conf. on Machine Learning*, 2000.

[19] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[20] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736, 2006.

[21] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1986–1993, 2011.

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

[23] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2017.

[24] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *NIPS*, 2017.

[25] E. Biyik and D. Sadigh. Batch active preference-based learning of reward functions. In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*, volume 87 of *Proceedings of Machine Learning Research*, pages 519–528. PMLR, October 2018.

[26] N. Wilde, D. Kulić, and S. L. Smith. Active preference learning using maximum regret. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10952–10959, 2020.

[27] E. Biyik, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In *Proceedings of the 3rd Conference on Robot Learning (CoRL)*, October 2019.

[28] M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. W. Burdick, and A. D. Ames. Preference-based learning for exoskeleton gait optimization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2351–2357, 2020.

[29] K. Li, M. Tucker, E. Bıyık, E. Novoseller, J. W. Burdick, Y. Sui, D. Sadigh, Y. Yue, and A. D. Ames. Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. *arXiv preprint arXiv:2011.04812*, 2020.

[30] L. Chen, H. Hassani, and A. Karbasi. Near-optimal active learning of halfspaces via query synthesis in the noisy setting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[31] R. Busa-Fekete and E. Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *International Conference on Algorithmic Learning Theory*, pages 18–39. Springer, 2014.

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

[33] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowd-sourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202, 2013.

[34] X. Chen, Y. Li, and J. Mao. A nearly instance optimal algorithm for top-k ranking under the multinomial logit model. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2504–2522. SIAM, 2018.

[35] M. Ben-Akiva and S. R. Lerman. *Discrete choice analysis: theory and application to travel demand*. Transportation Studies, 2018.

[36] R. C. Wilson and A. G. Collins. Ten simple rules for the computational modeling of behavioral data. *Elife*, 8:e49547, 2019.

[37] E. Bıyık, D. A. Lazar, D. Sadigh, and R. Pedarsani. The green choice: Learning and influencing human decisions on shared roads. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 347–354, 2019.

[38] L. Maystre and M. Grossglauser. Fast and accurate inference of plackett-luce models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 172–180, 2015.

[39] C. Archambeau and F. Caron. Plackett-luce regression: a new bayesian model for polychotomous data. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 84–92, 2012.

[40] T. Lu and C. Boutilier. Learning mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 145–152, 2011.

[41] V. Vitelli, Ø. Sørensen, M. Crispino, A. Frigessi Di Rattalma, and E. Arjas. Probabilistic preference learning with the mallows rank model. *Journal of Machine Learning Research*, 18(158):1–49, 2018.

[42] T. Lu and C. Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *Journal of Machine Learning Research*, 15(117):3963–4009, 2014.

[43] R. Busa-Fekete, E. Hüllermeier, and B. Szörényi. Preference-based rank elicitation using statistical models: The case of mallows. In *International Conference on Machine Learning*, pages 1071–1079. PMLR, 2014.

[44] P. De Blasi, L. F. James, J. W. Lau, et al. Bayesian nonparametric estimation and consistency of mixed multinomial logit choice models. *Bernoulli*, 16(3):679–704, 2010.

[45] F. Chierichetti, R. Kumar, and A. Tomkins. Learning a mixture of two multinomial logits. In *International Conference on Machine Learning*, pages 961–969. PMLR, 2018.

[46] A. Liu and A. Moitra. Efficiently learning mixtures of mallows models. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–638, 2018.

[47] J. Morton and M. J. Kochenderfer. Simultaneous policy learning and latent state inference for imitating driver behavior. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[48] C. Basu, E. Biyik, Z. He, M. Singhal, and D. Sadigh. Active learning of reward dynamics from hierarchical queries. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[49] Z. Cao, E. Biyik, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh. Reinforcement learning based control of imitative policies for near-accident driving. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2020.

[50] A. H. Qureshi, J. J. Johnson, Y. Qin, T. Henderson, B. Boots, and M. C. Yip. Composing task-agnostic policies with deep reinforcement learning. In *International Conference on Learning Representations*, 2019.

[51] R. D. Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[52] N. Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(1), 2012.

[53] D. Bertsimas, J. Tsitsiklis, et al. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[54] D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 1*, pages 766–774, 2010.

[55] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[56] A. X. Zheng, I. Rish, and A. Beygelzimer. Efficient test selection in active diagnosis via entropy approximation. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 675–682, 2005.

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

[58] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. Fetch and freight: Standard platforms for service robot applications. In *Workshop on autonomous mobile service robots*, 2016.

[59] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[60] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[61] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.