IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Mixed-Type Query Selection for Robotic Scientific Data Collection

Ian C. Rankin[1], Thane Somers[2], Alivia M. Eng[3], and Geoffrey A. Hollinger[1]

***Abstract*—We propose combining preference and rating query types into a mixed-type query selection to learn reward functions for robotic decision making to improve scientific data collection. Mixed-type query selection allows the scientist operating a robot to specify the robot's tradeoffs and goals in terms of both rating, giving a score to one robot plan, and preferences, selecting a preferred plan to another plan. While previous methods have used active learning to allow the user to specify tradeoffs between objectives using rating and preferences individually, our proposed method considers using multiple query types. We assume a user responds to these queries with some noise on their true preferences. Online estimation of error model parameters is difficult; therefore, we show results with both a tuned known error model and a heuristic mixed-type query selection method. When the error model is known, we show performance increases using our mixed-type query selection versus using only ratings or only preferences. In the more realistic case with an unknown error model, we show our heuristic performs better than the worst case single query type in all cases we tested.**

***Index Terms*—Human-Robot Collaboration, Motion and Path Planning, Field Robots**

I. INTRODUCTION

WHEN scientists seek to understand physical processes, robots have many advantages over human sampling approaches to collect the necessary data. However, determining where the robot should sample requires considering many features. For example, the robot may want to consider sampling patterns that are easier for scientists to use the measurements, sample locations that minimize uncertainty in a scientist's hypothesis, maximize sample diversity, and focus sampling on particular regions of interest. Explicitly specifying these tradeoffs is difficult and requires domain knowledge of both the robotic system and the scientific domain being studied [1]. To this end, we propose learning the tradeoff between rewards by querying the scientist operating the robotic system using both ratings, where the user specifies an explicit score to a plan, and preferences, where the user specifies a preferred plan over another possible plan. While rating queries can ground the estimated reward function to an absolute value, the scientist may have challenges scaling ratings and comparing similar plans indirectly. Preference queries can help the scientist compare between similar plans directly, but they only provide relative information between plans. Instead, we propose using mixed-type query selection that combines the two types of queries to select which query type will most likely improve the robots estimate of the best decision aligned with the user, see Fig. 1.

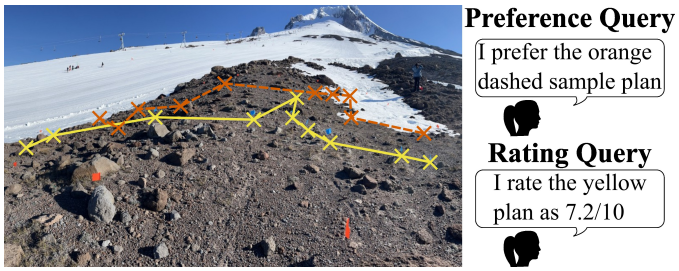


Fig. 1: Preference and rating query types are shown as illustrative sample plans on our motivating domain of in-situ sample measurements at Mt. Hood, Oregon. Mixed-type query selection enables us to choose between the two types of queries.

Our proposed method performs mixed-type query selection to select between query types when one is more applicable than the other. For example, using a rating query to eliminate large parts of the reward space with low rewards and using a preference query to directly compare similar plans. To make the selection of query types, we assume the human user has some error model on their response to a query. For rating queries, this is some error on the scalar value they would specify for a plan. For a preference query, this is some probability of selecting the wrong plan instead of their true preference.

When these error models are known, an extension of the acquisition function for preference query selection used in Ellis et al. [2] allows for a continuous rating response from a user. This extension allows both continuous rating and discrete preference responses to be directly compared to determine which query type to show to the user. However, knowing the user's error model a priori is typically infeasible and is difficult to learn online with the few responses a user can provide. Knowledge of the preferred single query type is difficult in our motivating domain with scientist users because it is unlikely we would be able to perform enough testing prior to deployment to determine an individual's error model. Therefore, we also propose using a set of heuristic mixed strategies that balances query types without knowledge of the user error model. When the user error model is unknown, the heuristics provide consistent performance across different error models compared to a single query type.

Our motivating domain for the mixed-type query selection method is robotic data collection for planetary scientists. We worked with planetary scientists performing in-situ experi-

Manuscript received: February, 28, 2025; Revised June, 19, 2025; Accepted July, 17, 2025.

This paper was recommended for publication by Editor Giuseppe Loianno upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by NSF grants IIS-1845227 and IIS-2103817. NASA Planetary Science and Technology Through Analog Research (PSTAR) program, Award No. 80NSSC22K1313.

[1] Ian C. Rankin and Geoffrey A. Hollinger are with Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, 1500 SW Jefferson Way, Corvallis OR, 97331, United States; {rankini, geoff.hollinger}@oregonstate.edu.

[2] Thane Somers is with Transnetyx Inc., 8110 Cordova Rd #119, Cordova, TN, United States; tsomers@transnetyx.com.

[3] Alivia M. Eng is with Georgia Institute of Technology, 311 Ferst Dr NW, Atlanta, GA, 30318, United States; aeng60@gatech.edu

Digital Object Identifier (DOI): 10.1109/LRA.2025.3597488

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ments to understand the terrain in an icy martian or lunar analogue environment on Mt. Hood. Our method selected sample locations for an in-situ robotic rheometer [3], which could be used in conjunction with a planetary rover. For prior data used by the in-situ sample selection, we generated a drone-orthomosiac of the nearby environment. We selected reward features that fit the planetary scientist's desired sampling behaviors: maximizing spatial distribution, encouraging sample clustering, increasing distribution of sample distances from a feature, sampling ease, and a preference towards samples being in straight lines. In order to generate statistical results, we use simulated users to test our methods and show (1) our proposed mixed-type query selection with a known user error model performs as well or better than using only rating [4, 5], or only preference query types [2], across a variety of simulated user types, and (2) our proposed heuristic performs better than incorrectly selecting the best single query type when the user error model is unknown.

We present four main contributions in this paper:

1) Mixed-type queries to allow both preference and rating queries to be used to estimate a robotic reward function.
2) A hybrid strategy for mixed-type query selection for known user error models. Our method uses a reformulation of acquisition selection [2] that enables direct comparison of a proposed rating query using Upper Confidence Bound (UCB) and a proposed preference query using the estimated acquisition function.
3) A heuristic method that uses a mixed strategy to select query types when the user error model is unknown.
4) Validation of mixed-type query selection using simulated users on real-world data from our motivating domain of in-situ rheometer measurements taken for planetary scientists.

II. RELATED WORKS

Scientific data collection, such as robots observing geological features, is of utmost importance for understanding physical processes in environments where collecting data is time-consuming, dangerous, or hard to reach. There are many methods that can help scientists optimize robot missions to collect data [6, 7]. These decision making algorithms require optimizing a reward function within a set of constraints. However, there are typically many features the robot should consider when making decisions. Trading off between the different objectives is challenging and typically not directly considered in most information gathering methods.

Learning an estimated reward function from humans has been achieved using different methods. One common method is learning from demonstrations [8] or inverse reinforcement learning [9, 10]. This broad set of methods allows the user to demonstrate a task to a robot in order to learn the reward function from the observed behavior. However, this assumes that the user can easily generate near-optimal solutions, which is often time-consuming or impossible. Instead, we propose using rating a path or providing a preference between a set of generated full-plans to estimate the reward function. Coactive learning is a method that, in contrast to learning from demonstration, uses modifications of a full demonstration to update the model [1, 11, 12]. However, modifying a full robot demonstration is also a costly process for users and, in some cases, is as time consuming as learning from demonstration [1]. Instead, providing a preference of different demonstrations or a rating of a single demonstration can be easier for the user and does not rely on them being able to generate near-optimal demonstrations.

To select preference queries to estimate the reward function with minimal polling of the user in a robotics domain, Sadigh et al. [13] uses a Bayesian model to estimate a linear function approximation of the reward function with a volume reduction query selection method. Several improvements have been proposed on the volume reduction method, including an improved Mutual Information approach [14] and a query regret maximization to minimize the regret of the estimated reward function [15]. However, the method most similar to ours is an acquisition function approach that selects preference queries that maximize the alignment function of estimated reward functions [2]. This method improves upon previous query selection methods by providing a generalized way to estimate the acquisition over different alignment functions. This enables the designer to select an alignment function that best fulfills their goals, such as minimizing entropy of the reward space, or maximizing the correct ranking of solutions. While Mutual information [14] searches the entire reward space the acquisition function approach [2] searches the reward space, that leads to changed behavior in the robot. However, all of these query selection methods only consider preferences-type queries and do not consider any mixed-type query selection. In contrast, our method generalizes acquisition function selection for preferences, to include continuous rating queries to allow a direct comparison of the relative worth of preference and rating queries.

Although there have been a variety of methods that select queries and estimate the reward function from linear functions [2, 13–16], weighted linear sums do not represent the entire Pareto front of solutions when the front is concave [17]. For example, a weighted min function generates a concave Pareto front and cannot be represented solely by linear combination. An approach proposed to scalarize a Multi-Objective Optimization (MOO) is to represent the estimated reward as a weighted maximization [18, 19]. Another method is to use an expressive nonlinear function such as a Gaussian Process (GP), as has been done with preference learning for robotics [20]. This allows any known knowledge of the reward function to be encoded in the covariance function of the GP while being nonlinear and flexible. To represent our reward function, we use the Bayesian framework described in [21, 22] for GPs similar to [20]. Unlike Bıyık et al. [20], we use an absolute bounded function to model rating queries for mixed-type query selection. Finally, using a MOO allows our method to generate feasible solutions in the concave sections of the Pareto-front, whereas methods, such as [23], that use only linear combinations of functions and vary the parameters are unable to generate solutions in the concave sections.

Rating a particular solution is another method for a user to provide feedback to a robot. This can be considered as

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

a traditional regression problem for machine learning. UCB is a common method for performing Bayesian optimization on regression problems [4, 5]. However, we cannot directly compare the output of the UCB [4, 5] for ratings with the estimated alignment of preferences. This motivates our mixed-type query selection method described below to select between preference and rating queries to estimate the reward function for robotic decision making.

III. PROBLEM DEFINITION

Our goal is to generate the optimal robot plan relative to the user's true preferences. The user's reward function $f(\cdot)$ is assumed static, but is unknown to the robot. Thus, we optimize over an estimate of the user's reward function $\hat{f}(\cdot)$ that is learned from rating and preference queries.

We define the set of tradeoffs to consider as a function $g(p) = \mathbf{x}$ that returns a vector of reward values $\mathbf{x} \in \mathbb{R}^n$, where p is a set of sequential decisions. We then assume the human user has an unknown true reward function $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$. This makes the assumption the user only uses the features in the reward vector, $\mathbf{x}$, to evaluate robot plans. This allows us to formulate the goal objective function similar to the informative path planning problem with an unknown objective [24], where we optimize plans over the uncertain reward function subject to some budget. The goal objective is shown below where p^* is the optimal plan, p, from the set of all feasible plans, S, such that $p \in S$, $C(p)$ is the cost of plan p, and B is the budget constraint for the plan:

$$p^* = \underset{p \in S}{\operatorname{argmax}} \ \hat{f}(g(p)) \text{ s.t. } C(p) \leq B. \tag{1}$$

To learn the estimated reward function, $\hat{f}(\cdot)$, the user is asked to answer preference and rating queries. We define a query, Q, as a set of paths to show to the user and q to be their particular response to the query. For preference queries, Q is a set of two paths shown to the user $Q_p = \{p_u, p_v\}$, where the response provided by the user is a single path $q_p \in Q_p$. We define a preference relationship where an input point p_u is said to be preferred over p_v, if $p_u \succ p_v \leftrightarrow f(g(p_u)) > f(g(p_v))$. For rating queries, a single path is selected to be shown to the user $Q_r = \{p\}$, and the response is the user's evaluation of the path p on a bounded scale in $q_r \in (0, 1)$.

For preference queries, we model the probability of the user selecting a particular path with a choice model [21]. The choice model represents the probability of a particular preference between a query's path given the latent function. We use the following probit function over a logit for analytical reasons [22]. Given the the query $Q_p = \{p_u, p_v\}$ using the estimate of the reward function for each path as:

$$P(q = p_u | \hat{f}, Q_p = \{p_u, p_v\}) = \Phi\left(\frac{\hat{f}(\mathbf{x}_u) - \hat{f}(\mathbf{x}_v)}{\sigma_R \sqrt{2}}\right), \tag{2}$$

where $\Phi(\cdot) \in (0, 1)$, is the Cumulative Distribution Function (CDF) of the Gaussian Distribution. A single hyperparameter σ_R defines the probit function, with large values defining a larger reward difference required to get the same probability of a query being selected.

To incorporate rating queries with a user error model, we employ a parameterized beta distribution that enables a flexible, but analytically simple, model of the likelihood of a query response given the estimated reward function [22]. We assume the user is provided a query $Q_r = \{p\}$, for which they provide feedback as a real number $q \in (0, 1)$. The likelihood is defined as:

$$P(q | \hat{f}, Q_r = \{p\}) = \text{Beta}\left(q | \alpha(\hat{f}(\mathbf{x})), \beta(\hat{f}(\mathbf{x}))\right), \tag{3}$$

where α and β are shape parameters of the Beta distribution. A common reformulation of α and β that uses a mean link function to map the estimated reward, $\hat{f}(\cdot)$, to the mean of the query response allows us to match the input query from $(0, 1)$ to the probability distribution. The mean link uses the Gaussian CDF to map estimated reward to the mean of the Beta distribution as shown below:

$$\mu_B(\hat{f}(\mathbf{x})) = \Phi\left(\frac{\hat{f}(\mathbf{x})}{\sigma_B \sqrt{2}}\right). \tag{4}$$

The reformulated shape parameters are defined as:

$$\alpha(\hat{f}(\mathbf{x})) = \nu \mu_B(\hat{f}(\mathbf{x})), \beta(\hat{f}(\mathbf{x})) = \nu\left(1 - \mu_B(\hat{f}(\mathbf{x}))\right), \tag{5}$$

which define $\mu_B(\hat{f}(\mathbf{x}))$ as the mean of the Beta distribution with two hyperparameters, σ_B, and ν. The parameter σ_B defines the scale of the estimated reward to the query. The parameter ν is a representation of the sample size and roughly defines the "peakiness" of the Beta distribution, with larger values of ν centering the distribution on the mean. The combination of the preference and rating likelihood functions allows us to estimate the reward function using a GP.

Our goal therefore is to select query, and query types, $Q \in \{Q_p, Q_r\}$, that allow us to optimize the resulting robot path, Eq. 1. To specify the selection of different query types in a generalized way, we find queries that maximize the alignment of the estimated reward function, $\hat{f}$ to the user's reward function, f, [2]. Different alignment functions can be used to encourage different behavior from the query selection. Here the alignment function increases as the two input reward functions for particular examples become similar. This is desired because we only want changes in actions of the robot to affect the alignment. Therefore, our goal is to maximize the below equation, where $a(\cdot, \cdot)$ is the alignment between evaluations of reward functions, $R_f = \{f(g(p)) \ \forall \ p \in P_{ref}\}$ and $R_{\hat{f}} = \{\hat{f}(g(p)) \ \forall \ p \in P_{ref}\}$ are the evaluations of their respective reward functions for paths in the reference set, P_{ref}, and $\mathcal{Y} = \{(q_1, Q_1), (q_2, Q_2), ..\}$ is a set of responses and queries provided by the user:

$$\mathbb{E}_{\hat{f} \sim P(\hat{f}|\mathcal{Y})}\left[a(R_{\hat{f}}, R_f)\right]. \tag{6}$$

The above equation ensures that the expected alignment between the estimated and user reward functions is maximized. The reference P_{ref} is a set of feasible solutions to stabilize the alignment functions. For our method, we used a random sampling of prior data and feasible solutions, P, to fill P_{ref} with 30 solutions. While Ellis et al. [2] used Eq 6 to select preference queries, we use the estimated alignment to select

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

between query types. Our preference query types are selected using this method, but rating query types are selected via a UCB method.

IV. GAUSSIAN PROCESS FORMULATION

To estimate the reward function with both preference and rating queries using a nonparametric nonlinear function, a variation of a GP is used. The GP approximates the posterior of the Bayesian updates from Eq. 2 and Eq. 3 using a Laplace approximation [22]. The Laplace approximation is used instead of more precise expectation propagation methods because it is more computationally feasible.

A standard GP is a continuous Gaussian random function with a mean and covariance function, $k(\cdot,\cdot)$. Given a GP prior for a function $\hat{f}(\mathbf{x}) \sim \mathcal{GP}(0, k(\cdot,\cdot))$ then the prior log-likelihood on a set of input points $X = \{\mathbf{x}_1, \mathbf{x}_2, ...\}$ is defined as [25]:

$$\log P(\mathbf{f}|X) = -\frac{1}{2}\mathbf{f}^\intercal K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi, \quad (7)$$

where $K = k(X,X)$ is the covariance matrix of the prior input, $\mathbf{f}$ is the evaluation of the GP at test points $\mathbf{f} \sim \hat{f}(X)$, and n is the number of points in X. We selected a standard Radial Basis Function (RBF), or squared exponential, for the covariance function as a flexible but common kernel that only depends on the distance between two inputs to calculate.

However, Eq. 7 only specifies the prior likelihood on the prior input, but does not calculate the posterior with respect to observations. To perform the Laplace approximation we need to find the mode of the posterior likelihood, also known as the maximum a posteriori (MAP) as shown below:

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmax}} \ \log P(\mathcal{Y}|\mathbf{f}) + \log P(\mathbf{f}|X). \quad (8)$$

The log likelihood of the observations given the latent input points can be found by the summation of the log of the relative and absolute probit functions for the user error model, Eq. 2 and Eq. 3, as shown below:

$$\log P(\mathcal{Y}|\mathbf{f}) = \sum_{(q,Q)\in\mathcal{Y}} \log P(q|\mathbf{f}_Q, Q). \quad (9)$$

The Laplace approximation approximates the posterior as a GP with the mean as the mode of the likelihood and a second-order Taylor approximation as the posterior covariance. To calculate the Taylor approximation, the negative Hessian, W, of the log-likelihood of the data given the latent function is required, as shown below:

$$W_{i,j} = -\sum_{k}^{n} \frac{\partial^2}{\partial f(\mathbf{x}_i)\partial f(\mathbf{x}_j)} \log P(y_k|\mathbf{f}). \quad (10)$$

Finally, this allows us to define the posterior GP, where K^* is the posterior covariance with respect to the input data:

$$\hat{f} = P(\mathbf{f}|\mathcal{Y}, X) \approx \mathcal{N}(\hat{\mathbf{f}}, K^*), \text{ s.t. } K^* = (K^{-1} + W)^{-1}. \quad (11)$$

We refer the reader to [22] for more details on predicting the mean and covariance for unseen test points and optimizing for the mode in Eq. 8. We use this GP formulation to estimate the reward function given preference and rating queries.

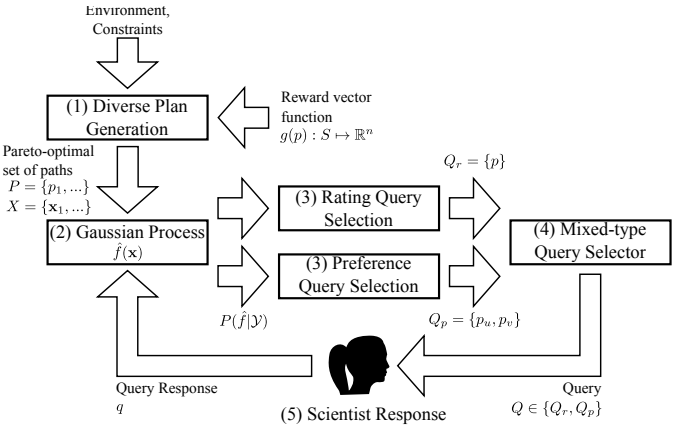


Fig. 2: Overview of our mixed-type query selection method.

V. METHODOLOGY

Our proposed methodology is a multi-step process to generate possible sampling plans and query the scientist user as described in Figure 2.

1) Using the reward vector, $g(p)$, we first generate a diverse set of feasible plans, P.
2) The set of feasible plans and associated reward vectors is scored using the underlying GP.
3) The absolute rating and pairwise preference-based query selectors are run to generate the best query and estimated alignment for each type of query.
4) The estimated alignment from each query selector is compared and the one with the highest estimated alignment is selected to show to the user.
5) The selected query is shown to the user and feedback is provided to the GP to estimate the reward function.

This process repeats for each query selection.

A. Alignment Maximization for Query Selection

To select queries, we wish to maximize the expected alignment of the estimated reward function to the user's reward function, Eq. 6. However, this expectation cannot be directly computed since we do not have access to the user's reward function. Instead, we compute Eq. 6 by comparing the estimated reward function to itself as opposed to the user's hidden reward function, from [2], as shown below:

$$\begin{aligned} Q^* &= \underset{Q}{\operatorname{argmax}}\ \phi(Q) \\ \phi(Q) &= \mathbb{E}_{q\sim P(q|Q,\mathcal{Y})}\left[\mathbb{E}_{\hat{f},\hat{f}'\sim P(\hat{f}|\mathcal{Y})}\left[a(R_{\hat{f}}, R_{\hat{f}'})\right]\right]. \end{aligned} \quad (12)$$

Solving for Eq. 12 selects the query estimated to maximize alignment of the estimated reward, to the true reward.

To make calculating the expectation in Eq. 12 feasible, we sample from the probability distribution $P(\hat{f}|\mathcal{Y})$ to get possible reward values for $R_{\hat{f}}$ and $R_{\hat{f}'}$. If we are using a GP as the latent function, then this is performed by multi-dimensional Gaussian sampling from the posterior of the GP.

To calculate Eq. 12 for both preference and rating queries, an alignment function is required. Since our domain assumes a full plan solution is provided for each reward, we do not use alignment functions that require action level rewards, such as

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Equivalent-Policy Invariant Comparison (EPIC) [2]. Instead, we compare between ρ-projection, log-likelihood alignments, and a new Spearman correlation method.

The log-likelihood metrics align if the probabilities of queries with one reward function are similar to the probabilities of queries of the second reward function:

$$a^{LL}(R_f, R_{f'}) = b^{LL}(R_{f'}, R_f) + b^{LL}(R_f, R_{f'})$$
$$b^{LL}(R_f, R_{f'}) = \sum_{Q \in \mathcal{Q}} \log P(q = \underset{p \in Q}{\operatorname{argmax}}\, R_{\hat{f}'}(p) | Q, R_{\hat{f}}), \tag{13}$$

for some set of queries $\mathcal{Q}$ that should be related to the reference set P_{ref}. The ρ-projection method, [26], projects the reward space such that L2-distance measures the misalignment between the functions:

$$a^{\rho}(R_f, R_{f'}) = -||\rho(R_f), \rho(R_{f'})||_{L2}$$
$$\rho(R_f) = \frac{\left[e^{f(g(p_1))}, e^{f(g(p_2))}, ..., e^{f(g(p_N))}\right]}{\sum_i^N e^{f(g((p_i))}}. \tag{14}$$

Finally, since our goal is to optimize plans that match the user's reward function, we only care if the relative ranks of the estimated and true reward function align. Therefore, we introduce the Spearman correlation alignment as a rank only correlation to align two reward functions.

$$a^{s}(R_f, R_{f'}) = r_s(R_f, R_{f'}), \tag{15}$$

where r_s is the Spearman correlation of the reward functions.

B. Preference Type Query Selection

For preference type query selection, we use a sampling method that is computationally feasible to solve Eq. 12. By sampling from possible reward functions over the prior distribution for each dummy variable, $\hat{f}, \hat{f}', \hat{f}'' \sim P(\hat{f}|\mathcal{Y})$, summing the expectations and iterating through responses, the estimated alignment in Eq. 12 can be calculated via:

$$\phi(Q)_p = \sum_{q \in Q} \frac{\mathbb{E}_{\hat{f},\hat{f}'}\left[P(q|Q, R_{\hat{f}}) P(q|Q, R_{\hat{f}'}) a(R_{\hat{f}}, R_{\hat{f}'})\right]}{\mathbb{E}_{\hat{f}''}\left[P(q|R_{\hat{f}''})\right]}. \tag{16}$$

C. Rating-based Query Selection

We calculate the estimated alignment for rating type queries, by reformulating the discrete summation as an integral over possible query values, as shown below:

$$\phi(Q)_r = \int \mathbb{E}_{\hat{f},\hat{f}'}\left[\frac{P(q|Q, R_{\hat{f}}) P(q|Q, R_{\hat{f}'}) f(R_{\hat{f}}, R_{\hat{f}'})}{\mathbb{E}_{\hat{f}''}\left[P(q|Q, R_{\hat{f}''})\right]}\right] dq. \tag{17}$$

Since each query is only a single path, performing the integration numerically over the sum of the sampled estimations is computationally expensive but feasible. However, as shown in Fig. 8, a UCB was found to outperform the estimated alignment method for rating only queries. The UCB uses the posterior prediction of the GP of the estimated reward function to determine the best query to use as shown below:

$$Q^* = \underset{Q}{\operatorname{argmax}}\, \hat{f}(Q) + \gamma \sqrt{K^*(Q)}. \tag{18}$$

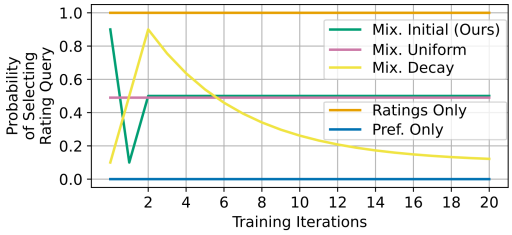


Fig. 3: Mixed strategy heuristic methods shown as the probability of selecting a rating query over the number of queries shown to the user.

For each of our experiments, we set $\gamma = 1$. In order to use this outperforming method for rating queries, we use a UCB to select rating queries but the estimated alignment of the selected query to perform query selection.

D. Mixed-Type Query Selection

Since preference and rating query types use different methods to select their respective queries, they cannot be directly compared with each other. To add a direct comparison, we calculate the estimated alignment for the proposed preference query, $\phi_p(Q_p^*)$, and the proposed rating query, $\phi_r(Q_r^*)$. This enables us to select the query that will maximize the estimated alignment between the user's reward function and the estimated reward function. We found using the same sampled reward function for ϕ_r and ϕ_p helped match the estimated alignments to each other. However, both estimated alignments for each query type rely on the parameters of the user error model specified in Eq. 2 and Eq. 3. If we know the parameters of the error model we can perform the mixed-type query selection as shown in Fig. 6, 5a, and 8.

E. Heuristic Mixed-Type Query Selection

For most scientific data collection robots, the ability to completely characterize the user error model before deployment is difficult. Instead, we propose using a set of heuristics to select query types when the user error model is unknown. The estimated alignment for preferences, $\phi_p(Q_p^*)$, and ratings queries, $\phi_r(Q_r^*)$, do not match each other when the user error model is unknown. This makes the query selection using the estimated alignments unstable. We instead propose a heuristic that avoids the worst-case scenario of selecting the least-performant single query type across different user error models. We developed several mixed strategies based on the the current number of queries shown to the user. Empirically, we selected initial setup and uniform mixed strategies as the two with the most consistent performance compared to a ramped decaying mixed strategy and a deterministic alternating strategy. The probability of selecting a rating query given the number of queries for every tested mixed strategy heuristic is shown in Fig. 3.

VI. EXPERIMENT DESIGN

We present results that demonstrate our methods performance against pure preference queries, using the state-of-the-art method proposed by Ellis et al. [2], and against pure rating queries, using a UCB method for selecting queries. Additionally, we test with simulated users with: (1) a known

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

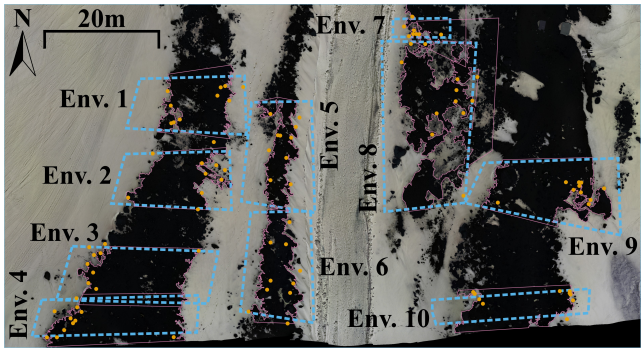


Fig. 4: Mt. Hood, OR, potential rheometer measurement locations generated by our algorithm. Environment 1 was scientist selected, others selected with similar properties. Orange dots represent potential sample locations, pink lines represents hand-defined ice-soil boundary. The drone-orthomosaic shown is used as prior data to the in-situ sample selection.

user error model, (2) our heuristic method with unknown user error models, and (3) a comparison of mixed-type query selection with different alignment functions. Validation using simulated users was performed so we could test with a wide range of error models and different underlying user reward functions. The required experiment trials to test the proposed approach with real users would be infeasible to perform with the same variety of user types. The tradeoffs considered were defined in collaboration with real users.

The motivating domain we selected for our work is one where a robot performs sampling for planetary scientists to understand the physical properties of the soil, in particular, "ice's influence on the strength of regolith-ice mixtures" [3]. The icy martian or lunar analogue of Mt. Hood was used as a test environment. We provide an autonomous method to propose sample locations for an in-situ robotic rheometer [3]. We interviewed several of the planetary scientists on the project to better understand their objectives. Through these conversations, we collaboratively defined a set of feasible, but competing, objectives for the planner. These objectives are: (1) maximizing the spatial distribution of sample locations, (2) minimizing the distance between samples to encourage clusters, (3) maximizing the distribution of samples with respect to their distance to the ice-soil boundary, (4) low elevation gradient to make the robotic rheometer sample easier to take, and (5) a preference towards samples being in straight lines. We also defined a set of constraints on how many points could be sampled on ice or soil. For our experiments we used 6 ice and 4 soil samples.

The environments are selected from a drone-orthomosaic with 2.4 cm resolution, shown in Fig. 4. The mosaic was acquired using the MicaSense RedEdge P-dual system mounted on a DJI Matrice 350 real-time kinematic positioning drone. The drone was flown within an hour of local solar noon and without cloud coverage to limit photometric effects. The offset nature of the MicaSense cameras allows for the creation of digital elevation models using photogrammetry, which we utilized for slope and elevation measurements. The drone-orthomosaic was taken on the first day of field work and acts as reconnaissance data, mimicking larger-scale, lower-resolution mosaics acquired by Mars rovers for scientific planning. One of the scientists selected one particular environment to perform in-situ sampling. We additionally selected another 9 similar but different set of environments from Mt. Hood for our user-preference tests. We use each of these 10 environments in conjunction with the simulated users discussed below for our results.

A. Multi-Objective Sample Planner

As discussed in Sec. V we need to generate a diverse set of sample plans. We use Pareto Monte-Carlo Tree Search (MCTS) [27] to generate a diverse set of Pareto-optimal sample plans. Pareto-MCTS is a modification of MCTS [4] that performs the selection step along the Pareto-optimal set of the UCB used in the standard MCTS formulation. To discretize the sample optimization we generate 31 random points along the ice-soil boundary, which is specified by hand from the drone orthomosiac taken prior to the in-situ sampling. Then for each of the random points we specify a perpendicular line to the ice-soil boundary and add possible sample points every 20 cm to a distance of 2 m on both sides of the ice-soil boundary. This is selected to provide a diverse set of points along the ice-soil boundary for MCTS to use and also allows straight line samples as desired by the scientists. Once these possible sample locations were generated, we run the Pareto-MCTS to generate a diverse set of Pareto-optimal sample plans, such that $C(p)$ is the number of points selected and $B = 10$. Each sample plan in the Pareto-optimal set was then scored by the estimated reward function.

B. Simulated Users

The simulated users have a random, monotonically-increasing function generated from one of three classes for their hidden reward functions: weighted min, weighted logistic, and weighted linear function. After calculating the underlying reward, these are mapped to the query types, and noise is injected. For preferences, we use the Luce-Shepard rule, a rule for choices from psychology and previous reward learning works [2, 14, 28]. This says the probability of a human selecting between options is the softmax of their rewards. Similar to Ellis et al. [2] we use a tuned parameter κ such that on average the preference is correct about 95% of the time for the experiment domain,

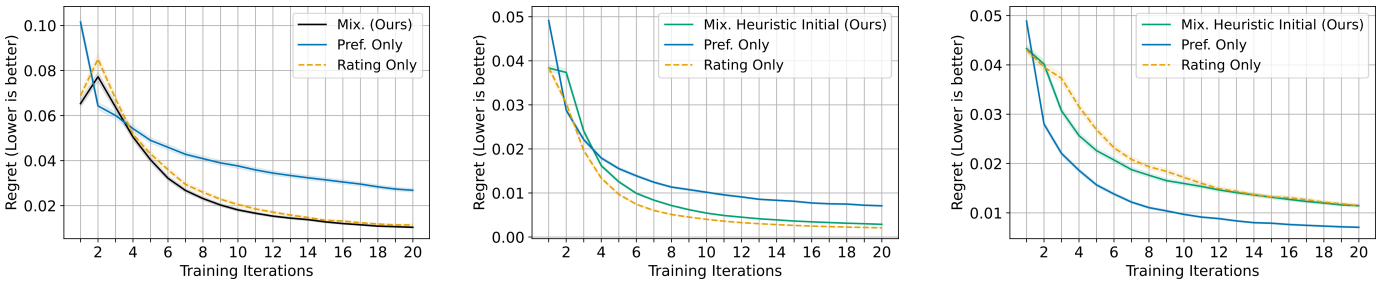
$$P(q|Q) = \frac{e^{\kappa f(\mathbf{x}_q)}}{\sum_{q_i \in Q} e^{\kappa f(\mathbf{x}_{q_i})}}. \tag{19}$$

Rating queries assume an absolute input between $(0, 1)$. Since the underlying function of the simulated user may not be between $(0, 1)$, we squash the input with a sigmoid function, which maps so that $r : \mathbb{R} \mapsto (0, 1)$. Before the sigmoid, normal random noise is added as shown below:

$$r(\mathbf{x}) = \text{sigmoid}(f(\mathbf{x}) + e) \text{ s.t. } e \sim N(0, \sigma_{rate}). \tag{20}$$

We tune the random noise, σ_{rate}, so that $r(\mathbf{x}_1) > r(\mathbf{x}_2)$ if $f(\mathbf{x}_1) > f(\mathbf{x}_2)$ with some probability, p_{abs}. Since a correct rating adds more information than a correct preference, we use different p_{abs} to model different error models of users. We select $p_{abs} = 0.95$ to model a user who is better at

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Training on min function with a simulated user better at responding to rating queries.

(b) Training of the heuristic method. Logistic function with a user better at responding to ratings.

(c) Training of the heuristic method. Logistic function with a user better at responding to preferences.

Fig. 5: Examples of training on different simulated users. In Fig. 5a we show our mixed-type query selection performs better than either single query types for the particular domain. In Fig. 5b, and 5c we show matched pairs of examples on the heuristic method. This shows that the while the heuristic does not perform better than the optimal single query type, it always performs better than picking the incorrect single query type.

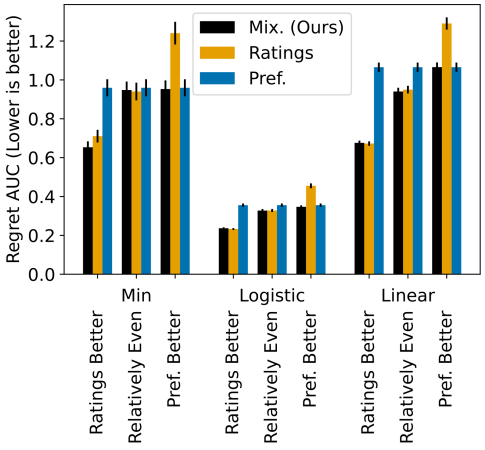


Fig. 6: Performance of Mixed-type query selection with a known user error model. These results show our method performs as well or better than using only ratings or preferences in all of our tested cases. Uncertainty shows 95% confidence interval on mean.

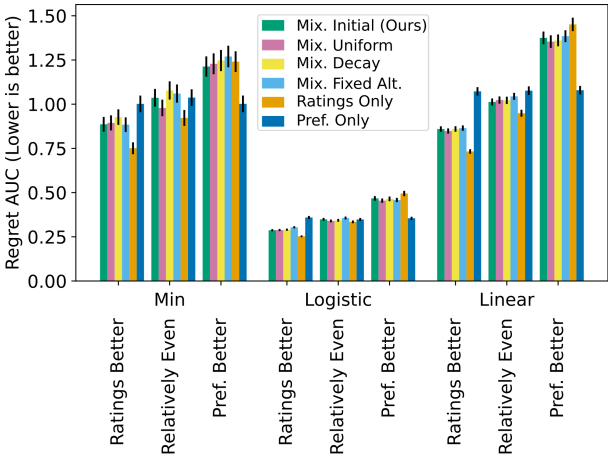


Fig. 7: Performance of Mixed-type query selection with the heuristic method for selecting query types when user error model is unknown. The initial setup and uniform random mixed strategy heuristics perform better than the worse case of using rating or preferences. Uncertainty shows 95% confidence interval on mean.

ratings, $p_{abs} = 0.8$ to model a user that is relatively equal at query types, and $p_{abs} = 0.7$ to model a user who is better at responding to preferences.

VII. RESULTS & CONCLUSIONS

We present the results in three parts. First, the query type selection with known user error model and explicitly tuned parameters. Second, the unknown user error model with a heuristic for selecting query types. Finally, we show results with different alignment functions and demonstrate why we use a hybrid approach that uses UCB for rating type selections. For each of the results we use the Spearman alignment function and UCB for rating only queries since there is little difference in alignment functions. With the exception of the alignment function plot, all results were run over all 10 environments 200 times, for a total of 2000 simulation runs on each case. We perform all experiments out to 20 queries. This number is selected because more queries affect GP scalability and would be increasingly burdensome for the scientist to answer.

We present results comparing our hybrid model, Sec. V-D with empirically tuned parameters to ratings and preferences alone, across each of the simulated user function classes and error models. The regret measures the difference between the simulated users actual best option, using $f(\cdot)$ versus the solution selected using the learned reward function, $\hat{f}(\cdot)$. The Area Under Curve (AUC) is the summation of all regrets measured in the 20 training iterations and evaluates training performance across the training period. This is different from the common AUC used with a Receiver Operator Curve (ROC), but similar to other active learning methods [28]. The results of these AUC tests are given in Fig. 6, showing our proposed mixed-query type method performs as well or better than using ratings only or preferences only in all cases. An example of the training process is shown in Fig. 5a.

In practice, determining the user error model is difficult. For this situation, we present our results in Fig. 7 that shows our proposed initial setup and uniform random heuristic consistently strikes a balance between query types when the optimal single-query type is unknown. I.e., the mixed query type heuristic never performs better than using the best of preferences or ratings but it performs better or the same as the worst case in all scenarios. For example, in Fig. 5b and Fig. 5c we show training examples for the logistic rating and preference-better cases. While our proposed mixed strategy heuristics do not perform as well as the best case single query type, in realistic scenarios, we do not know the optimal single query type. In scientific data collection domains, we typically do not get a second attempt to collect data; therefore, selecting a heuristic that reduces the chance of obtaining poor measurements is desirable. We also test against the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

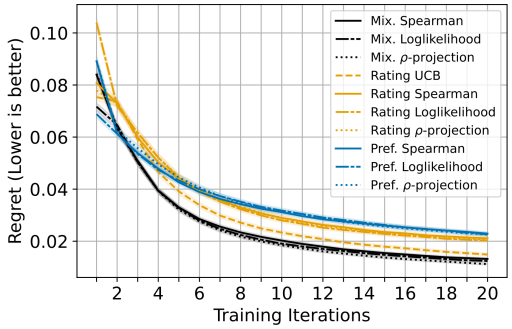


Fig. 8: Performance of query type selection with different alignment functions with known parameters. This is averaged over the: min, logistic, linear functions; and the rating better, preference better, and relatively even user error models. This shows each of the alignment functions perform relatively similar, and shows why we use a hybrid approach that uses UCB to select rating queries rather than directly using the acquisition function. Shaded region shows the 95% confidence interval on the mean.

deterministic alternating strategy and the ramp and decay mixed strategy, neither of these perform as consistently as the initial setup and uniform random heuristic strategies.

In Fig. 8 we show results of the acquisition functions and query type selections over the number of queries to the user. These results are averaged over all 9 cases of linear, logistic, and min classes of synthetic users, as well as the preferences better, rating better, and relatively even cases. Each line represents the mean of the 9 cases, 10 environments and 100 simulations runs on each, for a total of 9000 simulations runs for each line in the plot. These results show that while there are minor differences in the acquisition functions in the preference and mixed query type selections, these differences are dominated by the query type selection. In the rating only query type selection, the different alignment functions provide similar performance, but the UCB method performs significantly better. This is why we use a hybrid approach for the mixed query type method as described in Sec. V-D.

In this letter, we introduced mixed-type query selection to learn a reward function from domain experts for robotic data collection. With a known user error model, we proposed an estimated alignment that enables comparison between rating and preference query types providing improved performance over a single query type. In the more realistic case with an unknown error model, we proposed heuristics that provide consistent performance across users.

REFERENCES

[1] T. Somers and G. A. Hollinger. "Human–robot planning and learning for marine data collection". In: *Autonomous Robots* 40.7 (2016), pp. 1123–1137.

[2] E. Ellis, G. R. Ghosal, S. J. Russell, A. Dragan, and E. Bıyık. "A Generalized Acquisition Function for Preference-based Reward Learning". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan*. 2024.

[3] J. Ruck, J. Bush, J. Caporale, E. Fulcher, N. Jones, S. Thompson, B. McKeeby, K. Fisher, M. Nachon, D. Koditschek, et al. "Unveiling the Mechanics of Frozen Frontiers with a Robotic Rheometer". In: *LPI Contributions* 3040 (2024), p. 2651.

[4] L. Kocsis and C. Szepesvári. "Bandit based monte-carlo planning". In: *Proc. European Conference on Machine Learning (ECML), Berlin, Germany*. Springer. 2006, pp. 282–293.

[5] A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer. "Upper-Confidence-Bound Algorithms for Active Learning in Multi-armed Bandits". In: *Proc. International Conference on Algorithmic Learning Theory (ALT), Espoo, Finland*. 2011, pp. 189–203.

[6] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. "Efficient informative sensing using multiple robots". In: *Journal of Artificial Intelligence Research* 34 (2009), pp. 707–755.

[7] S. McCammon, D. Jones, and G. A. Hollinger. "Topology-Aware Self-Organizing Maps for Robotic Information Gathering". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, United States*. 2020, pp. 1717–1724.

[8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. "A survey of robot learning from demonstration". In: *Robotics and Autonomous Systems* 57.5 (2009), pp. 469–483.

[9] A. Y. Ng and S. Russell. "Algorithms for inverse reinforcement learning." In: *Proc. International Conference on Machine Learning (ICML), Stanford, CA, United States*. Vol. 1. 2000, p. 2.

[10] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. "Cooperative inverse reinforcement learning". In: *Advances in Neural Information Processing Systems* (2016), pp. 3916–3924.

[11] P. Shivaswamy and T. Joachims. "Coactive learning". In: *Journal of Artificial Intelligence Research* 53 (2015), pp. 1–40.

[12] R. Goetschalckx, A. Fern, and P. Tadepalli. "Coactive learning for locally optimal problem solving". In: *Proc. AAAI Conference on Artificial Intelligence (AAAI), Québec, Canada*. Vol. 28. 1. 2014.

[13] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia. "Active preference-based learning of reward functions". In: *Proc. Robotics: Science and Systems (RSS), The Hague, Netherlands*. 2017.

[14] E. Bıyık, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh. "Asking easy questions: A user-friendly approach to active reward learning". In: *Proc. Conference on Robot Learning (CoRL), Osaka, Japan*. 2019.

[15] N. Wilde, D. Kulić, and S. L. Smith. "Active preference learning using maximum regret". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, United States*. 2020, pp. 10952–10959.

[16] E. Bıyık, N. Anari, and D. Sadigh. "Batch Active Learning of Reward Functions from Human Preferences". In: *ACM Transactions on Human-Robot Interaction* 13.2 (2024), pp. 1–27.

[17] I. Das and J. E. Dennis. "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems". In: *Structural optimization* 14 (1997).

[18] A. Botros, N. Wilde, A. Sadeghi, J. Alonso-Mora, and S. L. Smith. "Regret-based Sampling of Pareto Fronts for Multi-Objective Robot Planning Problems". In: *IEEE Transactions on Robotics* 40 (2024), pp. 3778–3794.

[19] N. Wilde, S. L. Smith, and J. Alonso-Mora. "Scalarizing Multi-Objective Robot Planning Problems using Weighted Maximization". In: *IEEE Robotics and Automation Letters* 9 (2024), pp. 2503–2510.

[20] E. Bıyık, N. Huynh, M. J. Kochenderfer, and D. Sadigh. "Active preference-based Gaussian process regression for reward learning and optimization". In: *The International Journal of Robotics Research* 43.5 (2024), pp. 665–684.

[21] W. Chu and Z. Ghahramani. "Preference learning with Gaussian processes". In: *Proc. International Conference on Machine Learning (ICML), Bonn, Germany*. 2005, pp. 137–144.

[22] B. S. Jensen and J. B. Nielsen. "Pairwise judgements and absolute ratings with Gaussian process priors". In: *Technical University of Denmark (DTU), Department of Applied Mathematics and Computer Science, Tech. Rep* (2011).

[23] T. N. Somers. "Efficiently Learning Human Preferences for Robot Autonomy". MA thesis. Oregon State University, 2022.

[24] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin. "Efficient planning of informative paths for multiple robots". In: *Proc. International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India*. Vol. 7. 2007, pp. 2204–2211.

[25] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[26] S. Balakrishnan, Q. P. Nguyen, B. K. H. Low, and H. Soh. "Efficient Exploration of Reward Functions in Inverse Reinforcement Learning via Bayesian Optimization". In: *Advances in Neural Information Processing Systems* (2020), pp. 4187–4198.

[27] W. Chen and L. Liu. "Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning." In: *Proc. Robotics: Science and Systems (RSS), Freiburg im Breisgau, Germany*. 2019.

[28] V. Myers, E. Biyik, N. Anari, and D. Sadigh. "Learning multimodal rewards from rankings". In: *Proc. Conference on Robot Learning (CoRL), Auckland, New Zealand*. 2022, pp. 342–352.