# DUO: Diverse, Uncertain, On-Policy Query Generation and Selection for Reinforcement Learning from Human Feedback

**Xuening Feng[1], Zhaohui Jiang[1], Timo Kaufmann[2, 3], Puchen Xu[1], Eyke Hüllermeier[2, 3], Paul Weng[4*], Yifei Zhu[1]**

[1]UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai, China
[2]Institute for Informatics, LMU Munich, Munich, Germany
[3]Munich Center of Machine Learning, Munich, Germany
[4]Data Science Research Center, Duke Kunshan University, Kunshan, China
{cindyfeng2019,jiangzhaohui,xupuchen,yifei.zhu}@sjtu.edu.cn,
{timo.kaufmann,eyke}@ifi.lmu.de, paul.weng@duke.edu

## Abstract

Defining a reward function is usually a challenging but critical task for the system designer in reinforcement learning, especially when specifying complex behaviors. Reinforcement learning from human feedback (RLHF) emerges as a promising approach to circumvent this. In RLHF, the agent typically learns a reward function by querying a human teacher using pairwise comparisons of trajectory segments. A key question in this domain is how to reduce the number of queries necessary to learn an informative reward function, since asking a human teacher too many queries is impractical and costly. To tackle this question, most existing methods mainly focus on improving exploration, introducing data augmentation or designing sophisticated training objectives for RLHF, while the potential of query generation and selection schemes have not been fully exploited. In this paper, we propose DUO, a novel method for diverse, uncertain, on-policy query generation and selection in RLHF. Our method produces queries that are (1) more relevant for policy training (via an on-policy criterion), (2) more informative (via a principled measure of epistemic uncertainty), and (3) diverse (via a clustering-based filter). Experimental results on a variety of locomotion and robotic manipulation tasks demonstrate that our method can outperform state-of-the-art RLHF methods given the same total budget of queries while being robust to possibly irrational teachers.

## 1 Introduction

In reinforcement learning (RL), the reward function is typically specified to convey the task objective to the agent and provide guidance for learning to accomplish the task. A well-formulated reward function is fundamental for successful task learning. However, how to define a proper reward function remains an open problem (Amodei et al. 2016; Zhu et al. 2020), especially for complex tasks with large or continuous state and action spaces. Specifically, while a sparse reward function is easy to define, it can hardly guide the agent to reach the goal effectively. In contrast, denser reward functions can provide more informative learning signals but take significant effort for the system designer to formulate and may suffer from reward hacking (Skalse et al. 2022), where the agent learns to achieve high return despite undesirable behavior by exploiting flaws in the incorrectly-defined reward function.

To tackle the difficulty of defining a reward function, various directions have been explored in the past, including inverse reinforcement learning (Arora and Doshi 2021), imitation learning (Hussein et al. 2017), reward learning from demonstrations (Ibarz et al. 2018), and reinforcement learning from human feedback (RLHF) (Kaufmann et al. 2023). Compared with other directions, RLHF has recently drawn much attention due to its simplicity, scalability, and promising empirical results in many domains, such as control (Christiano et al. 2017), image generation (Lee et al. 2023), and language model alignment (Ouyang et al. 2022; OpenAI 2022).

In RLHF, instead of specifying a reward function explicitly, it proposes to learn the reward function from human preference feedback while simultaneously updating a policy via RL training using the learned reward function. In this setting, preference feedback is generally acquired by generating and selecting queries to be asked to a human teacher (oracle). The most typical query is a pairwise comparison of trajectories (Christiano et al. 2017; Lee, Smith, and Abbeel 2021; Lee et al. 2021), which we also focus on in this work.

A key question in RLHF is that of *query efficiency*, i.e., how to generate and select informative queries so that a good policy can be learned with fewer queries. Existing RLHF methods (Christiano et al. 2017; Lee, Smith, and Abbeel 2021; Park et al. 2022) directly take inspiration from common active learning approaches by selecting queries based on an estimate of the model uncertainty about the oracle's response. The rationale is that such queries are more informative and would help learn a good reward function faster. However, they still require a lot of queries to solve complex tasks, leaving much room for efficiency improvement.

By carefully examining the existing RLHF studies, we identify three limitations that commonly impede query efficiency. First, many RLHF methods select queries comparing **off-policy trajectory** segments regardless of their probability of being generated by the current policy. Unfortunately, asking queries about off-policy trajectories does not help obtain a better reward function to improve the current pol-

*Corresponding author

icy. Second, many methods rely on a **heuristic uncertainty evaluation**. In the context of RLHF, this uncertainty can include both aleatoric uncertainty (i.e., inherent randomness in the oracle's response) and epistemic uncertainty (i.e., lack of knowledge about the ground-truth rewards in the learned model) (Nguyen, Shaker, and Hüllermeier 2022). The extra capturing of aleatoric uncertainty leads to queries whose true preferences are inherently hard to determine, resulting in a large cognitive burden on the human teacher. Third, many methods ask a batch of queries to the human teacher, regardless of similarities between queries, referred to as **query similarity**. This introduces extra redundancy and hurts the query efficiency.

To overcome these limitations, this paper presents DUO, a novel query generation and selection scheme for RLHF. DUO meticulously selects **D**iverse, **U**ncertain, **O**n-policy queries so that the overall query efficiency can be significantly improved, illustrated in Figure 1. To compare trajectories that are more relevant to the current policy, DUO uses prioritized sampling to favor on-policy trajectories during query generation. To accurately capture the query uncertainty, DUO evaluates the epistemic uncertainty of a query in a principled way (Hüllermeier, Destercke, and Shaker 2022) with an ensemble of reward networks.

To further reduce the query redundancy in a batch, DUO proposes a simple yet effective clustering-based filter to select representative and diverse queries from the batch.

Our contributions are threefold: (a) We identify three sources of query inefficiency in previous RLHF methods (Section 2): off-policy trajectory, heuristic uncertainty evaluation, and query similarity. (b) We formulate a novel query generation and selection method to specifically address those three sources of issues (Section 4). (c) We experimentally demonstrate its efficiency on a large range of locomotion and robotic manipulation tasks (Section 5). DUO can outperform baseline methods with the same or even smaller budget of feedback. Each proposed technique individually contributes, with the combination having synergetic effects. In addition, DUO is demonstrated to be quite robust in terms of teachers' potential irrationalities and different algorithm designs.

## 2 Related Work

RLHF is a machine learning paradigm that learns behavior through human feedback, originating from early work on preference-based reinforcement learning (PbRL) and being scaled up to modern deep RL (Wirth et al. 2017; Christiano et al. 2017). We review related work on RLHF, concentrating on query generation and selection. Specifically, we discuss generating relevant candidate queries, selecting informative queries from this pool, and ensuring query diversity.

**Relevant Queries** In RLHF, we first need to generate candidate queries for the human teacher from a set of trajectories. Most existing works randomly sample a subset of collected trajectories and then segment and pair them (Christiano et al. 2017; Lee, Smith, and Abbeel 2021; Park et al. 2022; Liu et al. 2022). Our method, DUO, improves by prioritizing on-policy trajectories, arguing that off-policy trajectories do not lead to actionable reward model improvements. Lindner et al. acknowledge this and propose selecting queries based on their information gain on the optimal policy instead of the reward model, which is promising but computationally demanding. We propose prioritizing trajectories which are more likely under the current policy (on-policiness), a criterion that is easy to compute. Recent work by Hu et al. (2024) also recognizes this issue and proposes sampling trajectories which are more on-policy by only selecting recent experiences. In contrast, DUO determines on-policiness by the likelihood of the trajectory under the current policy, a more principled approach that can select older yet still relevant trajectories.

**Informative Queries** It is crucial to focus on the most informative queries to learn an effective reward model with fewer queries. This active learning problem is typically addressed by selecting queries based on the reward model's prediction uncertainty. Uncertainty estimation methods can be broadly divided into Bayesian and non-Bayesian ones. Bayesian methods, like Gaussian processes (Daniel et al. 2014), are computationally costly and often impractical for deep RL due to scalability issues and restrictive assumptions, such as that reward function is linear in a set of hand-engineered features (Sadigh et al. 2017; Bıyık and Sadigh 2018; Bıyık et al. 2020; Biyik, Talati, and Sadigh 2022). Instead, we focus on non-Bayesian methods using an ensemble of reward networks. Such an ensemble allows for the approximation of epistemic (reducible) uncertainty by quantifying the spread of the ensemble's predictions, e.g., by measuring the variance of the predictions (Christiano et al. 2017; Park et al. 2022). Lee, Smith, and Abbeel propose an entropy-based criterion, focusing on segment pairs with predicted preference probabilities close to $0.5$. This includes irreducible (*aleatoric*) uncertainty (Hüllermeier and Waegeman 2021), present in queries where true preferences are inherently hard to determine. While empirical evaluations with a synthetic human oracle show performance similar to ensemble disagreement, we argue that this is an instance of *heuristic uncertainty evaluation* that is not well-suited in practice since it tends to choose queries that place a large burden on the human teacher. We use the length of the interval of ensemble predictions in DUO, which, similar to the ensemble disagreement, is a measure of epistemic uncertainty with solid theoretical foundations (Hüllermeier, Destercke, and Shaker 2022; Sale, Caprio, and Hüllermeier 2023).

**Diverse Queries** Query diversity is regarded as important because queries are usually presented to the oracle in a batch-based fashion. Even though the current reward model may be highly uncertain about a batch of queries, the information contained in the batch may be redundant. Bıyık and Sadigh recognize this and take inspiration from literature on batch-active learning to evaluate multiple selection schemes focusing on diversity. Though we employ a similar idea to their clustering methods, DUO differs in key aspects: (1) Instead of representing queries as linear feature differences with specifically hand-coded features for each task, we do not make such strong assumption by using reward networks, and instead directly represent queries as sequences of reward differences. Interestingly, representing queries as linear feature differences amounts to aggregating over time, while

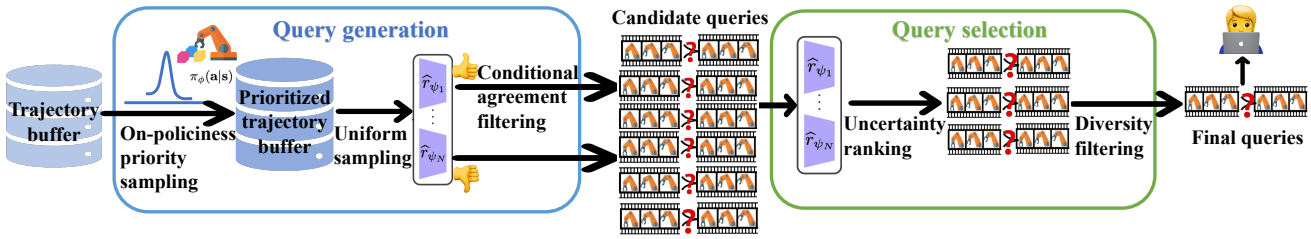



Figure 1: DUO consists of two main phases: query generation and query selection. Generation starts from the buffer of the agent's trajectories. A subset of trajectories is sampled with priority regarding on-policiness under the current policy. We then perform random segmenting and pairing, followed by filtering based on whether or not the decision boundary is included in the ensemble's predictions. The selection from the resulting candidate queries is based on the estimated epistemic uncertainty of the preference predictions and is further filtered by diversity using k-means clustering.

representing them as reward differences amounts to aggregating over linear features. Note that without linear features, working in the space of feature differences may not be well-justified, which motivates us to resort to the sequence of reward differences instead. (2) This representation allows us to use $K$-means clustering, where the query closest to each found cluster is selected, without needing a more complex clustering algorithm. (3) We automatically determine the $K$ value for $K$-means using the elbow method, leading to a more adaptive query selection scheme.

## 3 Preliminaries

**Reinforcement Learning** In a standard reinforcement learning (RL) problem, at every time step $t$, an agent performs an action $a_t \in \mathcal{A}$ given current state $s_t \in \mathcal{S}$, moves to a new state $s_{t+1} \in \mathcal{S}$, and receives an immediate reward $r(s_t, a_t)$. The RL agent's objective is to learn how to select actions (i.e., policy $\pi(a_t \mid s_t)$) such that it maximizes its expected accumulated rewards.

**Reinforcement Learning from Human Feedback** In RLHF (Christiano et al. 2017; Lee, Smith, and Abbeel 2021), immediate rewards are not known. Instead, it is assumed that an oracle can provide preference feedback to comparison queries between trajectory segments, which is used to guide the agent to finish the task.

Formally, a segment $\sigma = (s_k, a_k, \ldots, s_{k+h-1}, a_{k+h-1})$ is a sequence of state-action pairs. Given a query $(\sigma^0, \sigma^1)$, the oracle can declare preference $y$ for the former ($\sigma^0 \succ \sigma^1$, $y(0) = 1$) or the latter ($\sigma^1 \succ \sigma^0$, $y(1) = 1$). Here we ignore the case where the two segments are considered equivalent for a perfect oracle. The query of pairwise comparison and corresponding preference feedback is denoted as $(\sigma^0, \sigma^1, y)$ and stored in a dataset $\mathcal{D}$.

Since rewards are unknown, the usual approach in RLHF is to simultaneously learn a reward model $\hat{r}(s, a)$ fitting the oracle preferences and train a policy $\pi$ using the learned rewards. Following previous work (Lee, Smith, and Abbeel 2021), we assume that the reward model $\hat{r}_\psi$ is an ensemble of $N$ reward networks ($\hat{r}_{\psi_i}$ parametrized by $\psi_i$ for $i \in \{1, \ldots, N\}$) with $\psi = (\psi_1, \ldots, \psi_N)$ and the policy $\pi_\phi$ is a neural network parametrized by $\phi$. Generally, parameters of both the policy and reward model are updated by interleaving the following two steps:

- *Step 1 (agent learning)*: The agent interacts with the environment using policy $\pi_\phi$ to collect trajectories. Policy $\pi_\phi$ is then updated with such trajectories via existing RL algorithms to maximize the expected return given by the reward model $\hat{r}_\psi$.
- *Step 2 (reward learning)*: Queries in the form of pairwise segments are generated and selected from the collected trajectories. Each reward network $\hat{r}_{\psi_i}$ is then optimized to fit oracle's feedback on these queries.

In principle, any RL algorithm could be employed in Step 1. In this paper, we specifically use the sample-efficient off-policy algorithm SAC (Haarnoja et al. 2018), as in PEBBLE (Lee, Smith, and Abbeel 2021), due to its entropy-regularized objective function, making it more robust to reward approximation.

In Step 2, reward learning is formulated as a supervised classification problem (Christiano et al. 2017) where the oracle's feedback is assumed to follow the Bradly-Terry model (Bradley and Terry 1952). In this model, with reward model $\hat{r}_\psi$, the preference feedback $\sigma^1 \succ \sigma^0$ to a query $(\sigma^0, \sigma^1)$ has the following probability:

$$\mathbb{P}_\psi[\sigma^1 \succ \sigma^0] = \frac{e^{\sum_t \hat{r}_\psi(s_t^1, a_t^1)}}{e^{\sum_t \hat{r}_\psi(s_t^0, a_t^0)} + e^{\sum_t \hat{r}_\psi(s_t^1, a_t^1)}}, \quad (1)$$

where segment $\sigma^j$ for $j \in \{0, 1\}$ is composed of pairs $(s_t^j, a_t^j)$ and $\hat{r}_\psi(s_t^j, a_t^j)$ is the average output of the $N$ reward networks $\hat{r}_{\psi_i}$ for $i \in \{1, \ldots, N\}$. Given the dataset $\mathcal{D}$ of preference feedback, the task of learning the reward model $\hat{r}_\psi$ can be expressed as minimizing the following cross-entropy loss:

$$\mathcal{L}^{\text{Reward}} = -\mathop{\mathbb{E}}_{(\sigma^0, \sigma^1, y) \sim \mathcal{D}} \big[ y(0) \log P_\psi[\sigma^0 \succ \sigma^1] + y(1) \log P_\psi[\sigma^1 \succ \sigma^0] \big] \quad (2)$$

## 4 Method

To improve batch query efficiency, we specify desired queries as *on-policy*, *uncertain*, and *diverse* queries, defined and explained next.

*On-policy queries* involve segments of trajectories generated by a current policy $\pi$. Asking such queries can accelerate RLHF training and reduce the total number of queries asked to the oracle. Indeed, while improving the reward approximation, especially in the state-action region visited by optimal policies, seems to be intuitively reasonable, improving it on state-action regions that cannot be visited by the current policy $\pi$ may be wasteful in terms of query efficiency, since better reward approximation in those regions does not help estimate a better policy gradient to train $\pi$. The policy gradient is defined on the state-action pairs visited by the current policy, so a more reliable reward model trained on these pairs would help the most the RL training.

*(Epistemically) uncertain queries* are queries for which the current learned reward model is uncertain about the predicted preferences. Asking queries on which the current reward model is already certain could be superfluous. This is arguably one of the most relevant properties for efficient querying and thus extensively studied in prior work. However, as we show in our work, it is not the only important aspect.

*Diverse queries* are key when asking queries in a batch setting. Asking two queries containing similar information, even if they are both highly uncertain, would be redundant, since the feedback to one query would provide a good hint for the other. A good notion of diversity is crucial in this context.

Conceptually, our approach to query generation and selection can be written as a composition of three functions applied to a replay buffer $\mathcal{B}$ of trajectories:

$$\xi_D(\xi_U(\xi_O(\mathcal{B}))), \tag{3}$$

where $\xi_D$, $\xi_U$, and $\xi_O$ return a subset of queries that are diverse, uncertain, and on-policy, respectively. Next, we provide a high-level description of $\xi_O$, $\xi_U$, and $\xi_D$. Their more detailed implementations can be found in Algorithm 1 in the supplementary material.

### 4.1 Implementation of Function $\xi_O$ for On-Policy Queries

Recall that $\mathcal{B}$ is the replay buffer which stores all trajectories collected during the interaction between the agent and the environment. From a high-level point of view, function $\xi_O(\mathcal{B})$ should generate the subset in which queries are formed by segments generated by the current policy $\pi$. While conceptually simple, this has an important drawback: It requires costly sampling of many trajectories using the current policy $\pi$, neglecting the wealth of trajectories from similar policies already stored in the replay buffer.

Instead, we propose to implement $\xi_O(\mathcal{B})$ via priority sampling over all trajectories (generated by $\pi$ and past policies) in the replay buffer $\mathcal{B}$, favoring trajectories that have a higher probability of being generated by $\pi$. Formally, the priority is derived from the following on-policiness measure

$$O(\tau) = \sum_{t=0}^{T-1} \log \pi\left(a_t \mid s_t\right), \tag{4}$$

where $\tau = (s_0, a_0, s_1, \ldots, s_T)$ is a $T$-length trajectory. This measure computes the log probability of a trajectory being generated by $\pi$, ignoring the unknown transition probabilities. Higher values indicate a trajectory has a higher chance to be generated by $\pi$. Trajectories are then sampled using a probability $\mathbb{P}(\tau)$ proportional to its rectified Z-score $O'(\tau)$:

$$\mathbb{P}(\tau) \propto O'(\tau) = \max\left(0, \frac{O(\tau) - \mu_{O(\mathcal{B})}}{\sigma_{O(\mathcal{B})}}\right), \tag{5}$$

where $\mu_{O(\mathcal{B})}$ and $\sigma_{O(\mathcal{B})}$ are the empirical mean and standard deviation of the "on-policy" measure over the trajectories in $\mathcal{B}$. $\xi_O(\mathcal{B})$ generates on-policy queries by pairing randomly chosen segments from those sampled trajectories.

### 4.2 Implementation of Function $\xi_U$ for Uncertain Queries

Let $\mathcal{Q}_O = \xi_O(\mathcal{B})$ be the set of on-policy queries generated in the previous step. From those queries, we seek to select the presumably most informative ones as a sample. To this end, we rely on the well established principle of uncertainty in active learning, that is, the learner's current uncertainty in predicting the outcome of a query is a good indicator of the (expected) informativeness of that query. As a first filter, we therefore remove from $\mathcal{Q}_O$ all queries with *consensual* ensemble predictions, meaning that all ensemble members favor $\sigma^1$ over $\sigma^0$ (i.e. $\mathbb{P}_{\psi_i}[\sigma^1 \succ \sigma^0] > 1/2$ for all $i \in \{1, \ldots, N\}$), or the opposite (see Appendix B.2 for details). To prioritize the remaining candidate queries, we quantify their uncertainty in terms of a suitable measure. Instead of looking at the total uncertainty, we focus on the epistemic part of the predictive uncertainty. As this is the reducible part of the uncertainty, it is arguably more relevant in the context of active learning (Nguyen, Shaker, and Hüllermeier 2022).

Recall that the preference feedback to a query $(\sigma^0, \sigma^1)$ is supposed to follow a Bradly-Terry distribution as shown in Equation (1), which is a Bernoulli distribution $\text{Ber}(\theta)$ with parameter $\theta = \mathbb{P}_\psi[\sigma^1 \succ \sigma^0]$. Thus, even if the learner has perfect knowledge of the reward model (parameter $\psi$ and hence parameter $\theta$), it cannot deterministically predict the feedback due to remaining *aleatoric* uncertainty (stochasticity of the oracle). This uncertainty is further increased by *epistemic* uncertainty, namely, uncertainty about the true model $\theta$.

Imagine the learner quantifies this uncertainty by a (second-order) distribution $\mathbb{P}_\theta$, using the expectation of that distribution as its current best guess $\hat{\theta}$. In the literature, different proposals for quantifying the (total) uncertainty associated with $\hat{\theta}$ and for (additively) decomposing this uncertainty into an aleatoric and an epistemic part have been proposed; e.g., the Shannon entropy (of $\text{Ber}(\hat{\theta})$) decomposes into conditional entropy (of $\mathbb{P}_\theta$) and mutual information. In this regard, different measures of epistemic uncertainty have been justified in terms of meaningful properties. Broadly speaking, all these measures quantify the dispersion of $\mathbb{P}_\theta$ in one way or the other, e.g., in terms of mutual information or variance.

Ensemble learning, which is also used in our approach, is commonly viewed as a practical means to produce a (discrete) approximation of the distribution $\mathbb{P}_\theta$. More specifically, from a Bayesian perspective, the ensemble predictions can be seen as an approximation of the posterior predictive distribution.

Correspondingly, continuous measures are replaced by their discrete counterparts, e.g., mutual information by Jensen-Shannon divergence or variance by sample variance.

These measures may require a large number of ensemble elements to produce an accurate estimation. Instead, in our approach, we adopt the length of the predicted preference interval to quantify the epistemic uncertainty associated with a query $(\sigma^0, \sigma^1)$. Formally, it is defined as $\max_i \hat{\theta}_i - \min_i \hat{\theta}_i$ where $\hat{\theta}_i = \mathbb{P}_{\psi_i}(\sigma^1 \succ \sigma^0)$ is the prediction produced by the $i^{th}$ ensemble member. While simple, this measure has been theoretically axiomatized (Hüllermeier, Destercke, and Shaker 2022) and is well-suited when the ensemble size is small. The queries are then prioritized in terms of this measure, and $\mathcal{Q}_U = \xi_U(\mathcal{Q}_O)$ is obtained by selecting the queries with the highest priority.

### 4.3 Implementation of Function $\xi_D$ for Diverse Queries

Let $\mathcal{Q}_U = \xi_U(\mathcal{Q}_O)$ be the set of uncertain and on-policy queries. Given $\mathcal{Q}_U$, containing redundant queries, we select the most representative ones via a clustering-based filter.

Redundancy or similarity among queries can be evaluated in many different spaces, e.g., state or observation space, or space of linear feature differences (see also Section 2 for related work). Indeed, one may consider defining query similarity via trajectory similarity in a state or observation space (possibly including actions). However, such representation may not be effective for our purpose since many seemingly dissimilar trajectories have similar values. As a sequence, query diversity based on such similarity definition may lead to generating diverse queries about the many diverse bad trajectories, which may not promote efficient RL training. Measuring query diversity in the space of linear feature differences (Bıyık and Sadigh 2018) is another possibility, however it cannot be readily implemented in our setting, since we do not assume a linear reward approximation scheme.

Therefore, we instead resort to representing queries in the space of reward differences. While it is possible that some particular features of trajectories may be ignored in this representation, we conjecture that this value space may capture important aspects relevant for reward learning. More specifically, we design function $\xi_D$ by representing queries in the space of predicted reward differences which imply information about preferences. Formally, a query $(\sigma^0, \sigma^1)$ corresponds to a vector $\Delta\hat{\boldsymbol{r}} = \hat{\boldsymbol{r}}^1 - \hat{\boldsymbol{r}}^0$ where $\hat{\boldsymbol{r}}^i = \big(\hat{r}_\psi(s^i_k, a^i_k), \ldots, \hat{r}_\psi(s^i_{k+h-1}, a^i_{k+h-1})\big)$ for $i \in \{0, 1\}$ denotes the predicted reward sequence given the current reward model for segment $\sigma^i = (s^i_k, a^i_k, \ldots, s^i_{k+h-1}, a^i_{k+h-1})$. If the predicted differences $\Delta\hat{\boldsymbol{r}}$ of different queries are similar, asking all such queries to the oracle may be redundant. Alternatively, one may understand this reward difference space as a value embedding space to describe queries, where the embedding is obtained from the trained reward model.

In this predicted reward difference space, we apply a clustering-based filter (i.e., $K$-means clustering) to select representative queries whose reward difference sequences are the closest to the clustering centers in terms of Mean Squared Error distance. Since the number of representative queries may depend on the particular set $\mathcal{Q}_U$, the value of $K$ is adaptively determined by the elbow method. Empirical results in Section 5 demonstrate the effectiveness of our idea of selecting representative queries in the predicted reward difference space via a straightforward clustering-based filter. It is worth noting that more advanced clustering approaches could also be applied in our framework when necessary, considering the trade-off of performance gain and computational complexity. The diverse queries returned by $\xi_D(\mathcal{Q}_U)$ are finally asked to the oracle. See Appendix B for a detailed algorithmic description of DUO.

## 5 Experiments

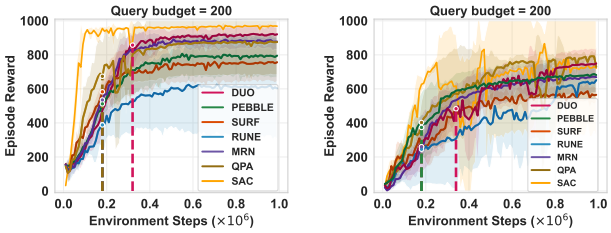
In this section, we design experiments to investigate the following questions: (1) How does DUO improve SOTA methods for RLHF in terms of performance and query efficiency? (2) How robust is DUO to the potential irrationalities of the simulated teacher? (3) How does each component of DUO contribute to the performance? (4) How sensitive is DUO to different hyperparameter settings? (5) How does DUO perform with a real human teacher involved?

### 5.1 Experimental Setup

**Tasks** We evaluate DUO on continuous control tasks, including 3 locomotion tasks from DMControl (Tassa et al. 2018) and 4 robotic manipulation tasks from Meta-World (Yu et al. 2020), similar to prior works (Lee, Smith, and Abbeel 2021; Lee et al. 2021; Park et al. 2022; Liang et al. 2022; Liu et al. 2022; Hu et al. 2024). We follow the setting where agents receive synthetic feedback from a scripted teacher, who provides preferences based on the ground truth reward, which is not directly observed by agents. With such feedback, agents learn to solve corresponding tasks guided by the underlying reward function. Performance is measured as the true average episode return for locomotion tasks and success rate for manipulation tasks, reporting the mean and standard deviation across five runs. Evaluation under different types of irrational scripted teachers as proposed by Lee et al. is also considered, detailed in Appendix C.1.

**Baselines** We implement DUO on top of the widely adopted method PEBBLE (Lee, Smith, and Abbeel 2021) in this paper. Meanwhile, many other methods based on PEBBLE have also shown good performance on control tasks mentioned above, including SURF (Park et al. 2022), RUNE (Liang et al. 2022), MRN (Liu et al. 2022), and QPA (Hu et al. 2024). Therefore, we adopt all these SOTA methods as baselines to demonstrate the effectiveness of our method. All baselines are evaluated with the original settings listed in their paper. More details are provided in Appendix A. Besides, considering all methods employ SAC for policy learning, we also measure the performance of SAC using the ground truth reward function as an upper bound.

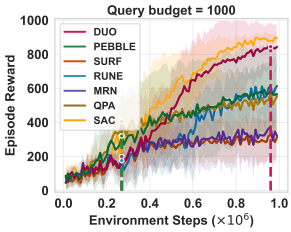
**Implementation of DUO** Apart from query generation and selection schemes, DUO follows the general architecture of PEBBLE. Unless stated otherwise, we also model the target reward function as an ensemble of three neural networks. See Appendix C.2 for more implementation details.



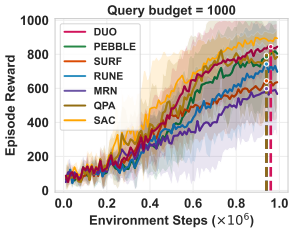

(a) Walker walker




(b) Cheetah run

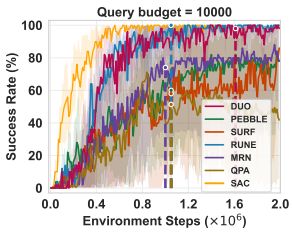



(c) Quadruped walk

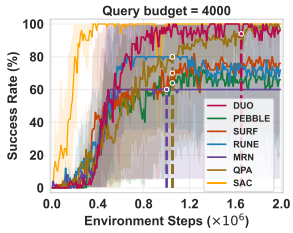



(d) Quadruped walk

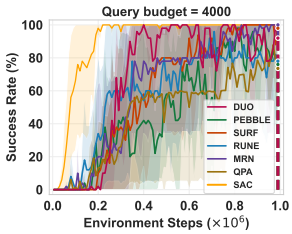
Figure 2: Learning curves on locomotion tasks as measured on the ground truth reward. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs. The vertical dashed lines indicate the end of the querying process.



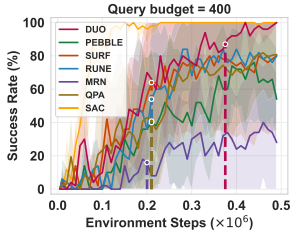

(a) Sweep into




(b) Drawer open




(c) Door open




(d) Window open

Figure 3: Learning curves on manipulation tasks as measured on the success rate. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs. The vertical dashed lines indicate the end of the querying process.

## 5.2 Benchmark Tasks with Unobserved Rewards

**Locomotion Tasks from DMControl** We compare DUO with the other 5 baselines on 3 typical DMControl tasks: *Walker walk, Cheetah run*, and *Quadruped walk*, respectively. All approaches share the same query budget under the same task for a fair comparison. Figures 2(a) to 2(c) show the learning curves of all methods with given budgets on selected tasks. See Appendix D.2 also for numerically presented results. Note that the vertical dashed lines in different colors indicate the end of the query-feedback process for each corresponding method. Here all methods run out of the given budgets, though DUO experiences a longer querying process resulting from its query selection scheme. That is, with the diversity filter, DUO finally selects an unfixed number of representative queries in each query-feedback session, leading to a possibly extended querying process.

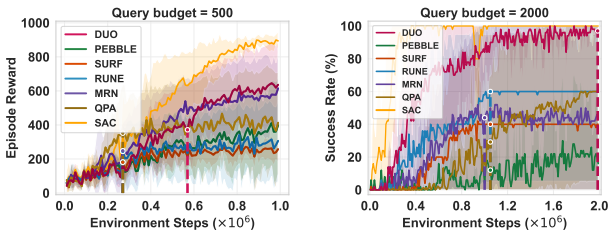
We find that DUO converges to higher performance than almost all the other baselines on the three tasks, except for falling a bit behind QPA on Cheetah run. To be mentioned, QPA also employs the idea of on-policiness (though with a different implementation), which the authors demonstrate has a positive impact on performance improvement. Particularly, on Quadruped walk, the most complex task in terms of the dimension of state and action space, DUO outperforms the baselines significantly and even reaches similar level to SAC.

Given that DUO enjoys a longer querying process, one may question whether the performance gain of DUO only comes from the extended querying process and whether other methods could also achieve the same level with such a querying process. To address this concern, we specifically design hyperparameters for the baselines to make sure they share a querying process similar to DUO. Figure 2(d) shows the learning curves of all methods in this setting on Quadruped walk. We see that by extending the querying process, all baselines do achieve performance improvement, but DUO still outperforms others. What's more, the improvement further proves the effectiveness of DUO, which achieves automatic and adaptive querying process extension via diversity filtering without manually designing hyperparameters. See Appendix D.1 for detailed discussion. All these results demonstrate that DUO can solve complex locomotion tasks without an explicit reward function while improving the query efficiency of RLHF.

**Robotic Manipulation Tasks from Meta-World** We also evaluate all methods on four Meta-World tasks: *Sweep into, Drawer open, Door open*, and *Window open*. Figure 3 shows the learning curves of all methods on each manipulation task given the query budget. DUO still outperforms the baselines significantly in all considered tasks except for Door open, where all methods converge to a similar final performance. However, even there, DUO learns faster than other baselines, implying that DUO is more query-efficient.Besides, here in Door open, the query budget is actually not used up for DUO. DUO only needs about 2760 queries to achieve such performance, while other baselines require almost the whole budget of 5000 queries. These results further demonstrate that DUO can achieve better query efficiency on a variety of complex tasks.

## 5.3 Robustness to Irrationalities

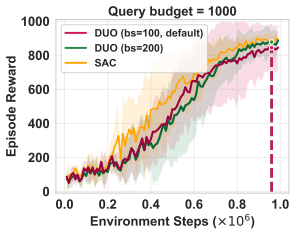
Previous experiments are conducted with a perfectly rational teacher that provides preference feedback strictly based on



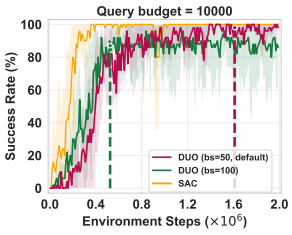

(a) Quadruped walk




(b) Drawer open




(c) Quadruped walk




(d) Sweep into

Figure 4: Sensitivity study on different tasks. Figure 4(a) and Figure 4(b) show results of different query budget, and Figure 4(c) and Figure 4(d) show results of different predefined query batch size.

the difference in the ground truth returns. In practice, real humans are likely to have difficulty answering queries with perfect rationality. To evaluate robustness of DUO to potential irrationalities, we adopt feedback from five types of scripted irrational teachers (Lee et al. 2021) on several tasks. Details about various irrationalities are provided in Appendix C.1.

We provide results of all methods under these irrationalities on different tasks in Appendix D.5. Results on Quadruped walk in Figure 7 show that DUO performs consistently better than almost all the baselines except that with the mistake teacher, who might provide wrong feedback with a certain probability. DUO works a bit worse than QPA but still competitively. Similar phenomenon can also be observed in Figure 8 on Walker walk. These results imply that DUO is quite robust to different kinds of irrationalities, which is of crucial importance for real-world applications.

### 5.4 Ablation Study

To figure out how the proposed components contribute to the final performance of DUO, we evaluate the performance in the absence of each component on various tasks (see Appendix D.3.). Figure 5 shows that each component has a positive influence on the final performance, which means both query generation (on-policiness) and query selection (uncertianty and diversity) play an important role in improving query efficiency, and their proper combination makes it possible for DUO to outperform other baselines.

### 5.5 Sensitivity Study

Basically we basically follow the general setting in previous work. To investigate the robustness of DUO, we evaluate DUO's performance on various tasks under different hyperparameter settings, that is, different query budget and different predefined query batch size (i.e., number of queries per feedback session).

Figure 4 shows partial results of DUO with different settings on several complex locomotion and robotic manipulation tasks. We see that with smaller query budget, DUO still outperforms significantly most baselines consistently as shown in Figures 4(a) and 4(b), which implies DUO is not only robust to different hyperparameters but also very query-efficient especially for complex tasks with limited queries. Besides, as Figures 4(c) and 4(d) show, given the same budget, even with different predefined query batch size, DUO can ask necessary number of queries per feedback session, which is reflected on the spread of querying process, and converge to similar final performance. See Appendix D.4 for results about different query budgets and query batch sizes on more tasks. We also provide empirical results under other different settings, including query frequency and segment size, which further demonstrate the robustness and effectiveness of DUO.

### 5.6 User Study

To illustrate that DUO is also effective in realistic settings, we perform a user study where a human user (already familiar with the task) guides the agent. This is done in a Quadruped environment, which is also considered by Lee, Smith, and Abbeel (2021). Here, the human hopes the agent could learn to stand and wave its right hind leg, and thus provides corresponding preferences over the presented pairs of video segments (i.e., queries). The human are asked 150 queries by the agent trained with PEBBLE and DUO, respectively. We evaluate 10 times the learned behavior of the agent at the end of training for both methods. Results show that with DUO, the agent always successfully performs the desired behavior but with PEBBLE, it hardly even stands up. Videos of selected queries and evaluation of trained agents for both methods are provided in supplementary materials.

## 6 Conclusion

We present DUO, a query generation and selection scheme for RLHF that improves query efficiency by focusing on diverse, uncertain, and on-policy queries. Experiments show that DUO significantly outperforms current SOTA algorithms in RLHF in terms of query efficiency and performance on a variety of locomotion and robotic manipulation tasks. We also demonstrate that DUO is robust to different types of irrationalities and hyperparameter settings, and show how each component of DUO contributes to the final performance. A user study further validates the effectiveness of DUO in practical scenarios. Overall, we believe that DUO provides an effective perspective to future research in RLHF.

However, our current evaluation is limited to control tasks. Considering the blossom of Large Language Models (LLMs), we believe that it is worthwhile to migrate our proposition to alignment of LLMs, which would be a promising area for future work.

## Acknowledgements

This work has been supported in part by the program of National Natural Science Foundation of China (No. 62176154), a research project funded by NetEase, and the LMUexcellent project funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder as well as by the Hightech Agenda Bavaria.

## References

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. Preprint, arxiv:1606.06565.

Arora, S.; and Doshi, P. 2021. A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress. *Artificial Intelligence*, 297: 103500.

Bıyık, E.; Palan, M.; Landolfi, N. C.; Losey, D. P.; and Sadigh, D. 2020. Asking Easy Questions: A User-Friendly Approach to Active Reward Learning. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR.

Bıyık, E.; and Sadigh, D. 2018. Batch Active Preference-Based Learning of Reward Functions. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR.

Biyik, E.; Talati, A.; and Sadigh, D. 2022. APReL: A Library for Active Preference-based Reward Learning Algorithms. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 613–617. IEEE Press.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4): 324–345.

Christiano, P.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems ((NeurIPS)*, volume 30. Curran Associates, Inc.

Daniel, C.; Viering, M.; Metz, J.; Kroemer, O.; and Peters, J. 2014. Active Reward Learning. In *Proceedings of Robotics: Science and Systems (RSS)*, volume 10.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR.

Hu, X.; Li, J.; Zhan, X.; Jia, Q.-S.; and Zhang, Y.-Q. 2024. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hüllermeier, E.; Destercke, S.; and Shaker, M. H. 2022. Quantification of Credal Uncertainty in Machine Learning: A Critical Analysis and Empirical Comparison. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. PMLR.

Hüllermeier, E.; and Waegeman, W. 2021. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, 110(3): 457–506.

Hussein, A.; Gaber, M. M.; Elyan, E.; and Jayne, C. 2017. Imitation Learning: A Survey of Learning Methods. *ACM Computing Surveys*, 50(2): 21:1–21:35.

Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; and Amodei, D. 2018. Reward Learning from Human Preferences and Demonstrations in Atari. In *Advances in Neural Information Processing Systems ((NeurIPS)*, volume 31. Curran Associates, Inc.

Kaufmann, T.; Weng, P.; Bengs, V.; and Hüllermeier, E. 2023. A Survey of Reinforcement Learning from Human Feedback. Preprint, arxiv:2312.14925.

Lee, K.; Liu, H.; Ryu, M.; Watkins, O.; Du, Y.; Boutilier, C.; Abbeel, P.; Ghavamzadeh, M.; and Gu, S. S. 2023. Aligning Text-to-Image Models Using Human Feedback. Preprint, arxiv:2302.12192.

Lee, K.; Smith, L.; Dragan, A.; and Abbeel, P. 2021. B-Pref: Benchmarking Preference-Based Reinforcement Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, volume 1.

Lee, K.; Smith, L. M.; and Abbeel, P. 2021. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR.

Liang, X.; Shu, K.; Lee, K.; and Abbeel, P. 2022. Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lindner, D.; Turchetta, M.; Tschiatschek, S.; Ciosek, K.; and Krause, A. 2021. Information Directed Reward Learning for Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34. Curran Associates, Inc.

Liu, R.; Bai, F.; Du, Y.; and Yang, Y. 2022. Meta-Reward-Net: Implicitly Differentiable Reward Learning for Preference-based Reinforcement Learning. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems (NeurIPS)*.

Nguyen, V.-L.; Shaker, M. H.; and Hüllermeier, E. 2022. How to Measure Uncertainty in Uncertainty Sampling for Active Learning. *Machine Learning*, 111(1): 89–122.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue. (accessed 2023-02-02).

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Park, J.; Seo, Y.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2022. SURF: Semi-supervised Reward Learning with Data Augmentation for Feedback-efficient Preference-based Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Sadigh, D.; Dragan, A.; Sastry, S.; and Seshia, S. 2017. Active Preference-Based Learning of Reward Functions. In *Proceedings of Robotics: Science and Systems (RSS)*, volume 13.

Sale, Y.; Caprio, M.; and Hüllermeier, E. 2023. Is the Volume of a Credal Set a Good Measure for Epistemic Uncertainty? In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. PMLR.

Skalse, J. M. V.; Howe, N.; Krasheninnikov, D.; and Krueger, D. 2022. Defining and Characterizing Reward Gaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; Lillicrap, T.; and Riedmiller, M. 2018. DeepMind Control Suite. Preprint, arxiv:1801.00690.

Wirth, C.; Akrour, R.; Neumann, G.; and Fürnkranz, J. 2017. A Survey of Preference-Based Reinforcement Learning Methods. *Journal of Machine Learning Research*, 18(136): 1–46.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2020. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR.

Zhu, H.; Yu, J.; Gupta, A.; Shah, D.; Hartikainen, K.; Singh, A.; Kumar, V.; and Levine, S. 2020. The Ingredients of Real World Robotic Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.