Comparing Comparisons: Informative and Easy Human Feedback with Distinguishability Queries

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

Abstract

Learning human objectives from preference feedback has significantly advanced reinforcement learning (RL) in domains with hard-to-formalize objectives. Traditional methods with pairwise trajectory comparisons face challenges: trajectories with subtle differences are hard to compare, and comparisons are ordinal, limiting direct inference of preference strength. In this paper, we introduce the *distinguishability query*, where humans compare two pairs of trajectories, indicate which pair is easier to compare, and then give preference feedback on the easier pair. This type of query directly infers preference strength and is expected to reduce cognitive load on the labeler. We also connect this query to cardinal utility and difference relations, and develop an efficient query selection scheme to achieve better trade-off between query informativeness and easiness. Experimental results empirically demonstrates the potential of our method for faster, data-efficient learning and improved user-friendliness on RLHF benchmarks.

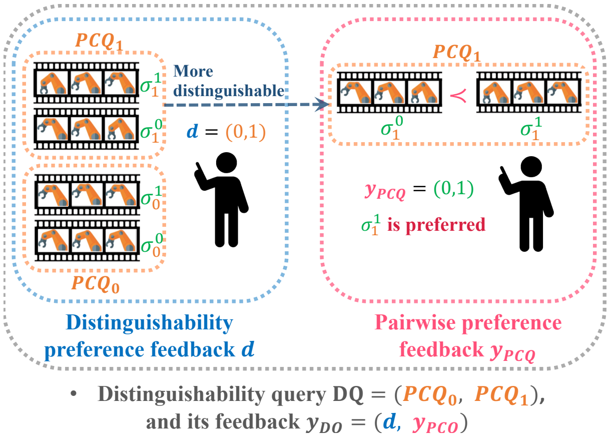

PCQ_1
σ_1^1
σ_1^0
More distinguishable
d = (0,1)
σ_0^1
σ_0^0
PCQ_0
Distinguishability preference feedback d
PCQ_1
$\prec$
σ_1^0
σ_1^1
y_{PCQ} = (0,1)
σ_1^1 is preferred
Pairwise preference feedback y_{PCQ}
• Distinguishability query DQ = (PCQ_0, PCQ_1), and its feedback y_{DQ} = (d, y_{PCQ})

Figure 1. An illustration of the distinguishability query. PCQ refers to the usual Pairwise Comparison Query for *a pair* of segments (σ^0, σ^1), and its feedback is only y_{PCQ} indicating the preferred segment. DQ refers to our proposed Distinguishability Query for *two pairs* (i.e., two PCQs), and its feedback (d, y_{PCQ}) includes an extra d to indicate which PCQ is easier to answer.

1. Introduction

Learning human objectives from preference feedback has been key to the success of reinforcement learning (RL) in domains where objectives are hard to formalize, such as fine-tuning large language models like ChatGPT (OpenAI, 2022; Ouyang et al., 2022) or training simulated robots to perform hard-to-define tasks such as backflips (Christiano et al., 2017). The standard method in this domain is to ask the human to compare pairs of trajectories, and then use these comparisons to learn a reward function that can be used to train an RL agent (Christiano et al., 2017; Lee et al., 2021b; Liang et al., 2022; Park et al., 2022; Hu et al., 2024; Verma & Metcalf, 2024). However, this method has two key limitations: (1) It can be hard for humans to compare trajectories, especially when the differences are subtle, and (2) the comparisons are ordinal, not cardinal, so preference strength can only be learned implicitly through inductive biases in the learning algorithm and utility-dependent noise in the human responses.

In this paper, we propose a new type of query, the *distinguishability query* (illustrated in Figure 1), that addresses above limitations. For a distinguishability query, the human first indicates which pair is easier to compare given two pairs of trajectories, then provides preference feedback on the selected pair as usual in preference-based RLHF methods. While the usual preference choice for a single pair of segments only assures the ordinal information, the extra choice of which comparison to provide feedback on is then used to infer the human's preference strength, assuming the human prefers to provide feedback on the more distinguishable pair. This allows us to learn preference strength

[1] UM-SUTU Joint institlute, 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. Correspondence to: Paul Weng <paul.weng@duke.edu>.

ICML 2024 Workshop on Models of Human Feedback for AI Alignment, Vienna, Austria. 2024. Copyright 2024 by the author(s).

directly, and allows the labeler to provide selective feedback on those comparisons that are easy to make. Our proposed method, named DistQ, is a cooperative approach that integrates distinguishability queries with an effective query selection scheme and a specialized learning objective.

Concretely, our contributions are as follows:

1. We propose a new type of query, the distinguishability query, in the field of RLHF (see Section 4.1);
2. We establish the relation to the concept of cardinal utility and distance relations studied in related disciplines (see Section 2);
3. We design a query selection scheme for distinguishability queries that aims to achieve a better trade-off between informativeness and ease of answering (see Section 4.2);
4. We propose a specific learning objective to improve reward learning by coupling cardinal and ordinal information (see Section 4.3);
5. We empirically demonstrate on classic control tasks with a synthetic oracle that our method can achieve competitive performance in a more user-friendly manner than standard pairwise comparison methods (see Section 5).

2. Related Work

In this work, we propose a new type of query for reinforcement learning from human feedback (RLHF). This new type of query is closely related to decision theory, RLHF approaches using pairwise comparisons, and approaches that aim to reduce the burden on the human labeler. We briefly review these in the following.

Preference Strength Observed choices, which can be seen as the basis of RLHF (Jeon et al., 2020), convey only *ordinal* preferences, i.e., information about rank but not about distance. Reinforcement learning is based on expected utility maximization and thus requires cardinal utilities—a fundamental concept in the von Neumann-Morgenstern expected utility theorem (Von Neumann & Morgenstern, 1947). Cardinal utilities additionally represent *preference strength* and have been extensively studied in related fields such as economics, psychology and decision theory (Suppes & Winet, 1955; Krantz et al., 2006; Jansen et al., 2018). Prior work has shown that the preferences elicited in RLHF, when noisy with certain assumptions, can be used to infer cardinal utilities (Chan et al., 2021; Xu et al., 2020). This aligns with the empirical success of RLHF methods relying on pairwise comparisons (Lee et al., 2021b; Liang et al., 2022; Park et al., 2022; Hu et al., 2024). Nonetheless, these assumptions about the utility-dependent noise are strong and may not hold in practice, leading us to explore more direct ways to elicit cardinal utilities.

Distance Relations Cardinal utilities can either be modelled as a real-valued function (unique up to positive affine transformations) or as a relation on pairs of outcomes. The latter formalism aligns closely with our proposed query type, asking the human labeler to distinguish between two pairs of outcomes. Formally, we can model the human labeler's preferences using two relations R and D (Suppes & Winet, 1955; Jansen et al., 2018), with R being a preference relation and D a difference relation. If a pair of pairs satisfies $((a, b), (c, d)) \in D$, then exchanging b by a is at least as desirable as exchanging d by c, that is, a is more strongly preferred over b than c is over d. Several prior works establish a set of axioms that determine a utility function from such a relation unique up to positive affine transformations (Alt, 1936; Suppes & Winet, 1955; Köbberling, 2006). Notable among these axioms are completeness and transitivity. If completeness is not satisfied, the relation merely determines a set of compatible utility functions (Pivato, 2013).

Eliciting Preference Strength Explicitly eliciting distinguishability (cardinal utilities) has not been studied within the context of RLHF to our knowledge, though related concepts have been explored in decision theory. These approaches are not directly applicable to RLHF, since they generally assume preferences over a limited set of outcomes that can be elicited (near) exhaustively, in contrast with our setting that requires generalization to unseen items. Jansen et al. (2022) propose inferring absolute preference strength through direct (label elicitation) or indirect (time elicitation) methods. Time elicitation infers preference strength from consideration time, relying on assumptions about the human labeler's behavior that may not hold in practice. Label elicitation collects explicit ordinal preference strength labels, placing an additional burden on the human labeler and suffering from limitations inherent to explicit ratings such as annotation biases and inconsistencies (Yannakakis & Martínez, 2015). In contrast to these approaches, distinguishability queries require only relative labels, making them less burdensome since the labeler only chooses between two queries instead of providing a label for each.

Reducing the Burden on the Human Labeler Another limitation of pairwise comparisons is that they can place a high burden on the human labeler when the behaviors to compare are similar or neither is preferred. Prior work has addressed this issue through multiple strategies: (1) pre-training, either in an unsupervised manner (Lee et al., 2021b) or using demonstrations (Ibarz et al., 2018; Palan et al., 2019; Bıyık et al., 2022), (2) allowing labelers to abstain from answering queries (Lee et al., 2021a), and (3) query selection strategies that aim to select queries that are easier for the human labeler to answer (Bıyık et al., 2019). The first two strategies are entirely orthogonal to our work, and can be combined with our approach. In this third strategy, which our approach falls into, Bıyık et al. (2019)

propose to use information gain to select queries that are informative and easy to answer, implicitly prioritizing queries the human will be able to answer and thus lead to the largest gain of information. Distinguishability queries could easily be constructed according to information gain, however we opted for our proposed query selection scheme (based on ensemble disagreement and prediction entropy), since an approach based on information gain is computationally challenging and difficult to scale. More importantly, in contrast to using this strategy in isolation, our distinguishability queries additionally let the human labeler choose the easier queries themselves, which has several additional benefits: (1) effectively reducing the burden on the human labeler who only has to answer easier-to-answer pairwise comparisons; (2) allowing them to compensate for the limitations of the measure of easiness; (3) gaining additional insight into utility differences, possibly even improving the easiness estimates (based on entropy, and thus indirectly utility differences) for future queries; (4) possibly accounting for both reward and response model error.

3. Preliminaries

Reinforcement learning We consider a reinforcement learning (RL) setting where an agent interacts with the environment to maximize its expected cumulative reward. This can be modelled with a discrete-time Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$. Here $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space, $\mathcal{P}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$, $r(\boldsymbol{s}, \boldsymbol{a})$, and $\gamma \in (0, 1]$ represent the transition function, reward function, and the discount factor, respectively.

At each timestep t, the agent receives a state $\boldsymbol{s}_t \in \mathcal{S}$ from the environment and chooses an action $\boldsymbol{a}_t \in \mathcal{A}$ according to its policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$. Conventionally, the environment also provides a reward signal $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and the agent transitions to the next state $s_{t+1} \sim \mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. The return $\mathcal{R}_t = \sum_{k=0}^{\infty} \gamma^k r(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k})$ is defined as the discounted cumulative sum of rewards from timestep t. The agent's goal is then to learn an optimal policy that maximizes the expected return from each state $\boldsymbol{s}_t$.

Reinforcement Learning from Human Feedback Reinforcement learning from human feedback (RLHF) (Kaufmann et al., 2023) is a framework that aims to learn optimal agent behavior from human feedback. In this paper, we focus on an RLHF framework that aims to infer a reward function $r(\boldsymbol{s}, \boldsymbol{a})$ unknown to the agent, which is then used to train a policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ (Lee et al., 2021b;a). The agent infers the reward function from qualitative human preference feedback, learning an approximate reward function $\hat{r}_\psi(\boldsymbol{s}, \boldsymbol{a})$. This approximation is modeled as an ensemble of N neural networks $\hat{r}_{\psi_i}$ that is parameterized by ψ_i for $i \in \{1, \ldots, N\}$) with $\psi = (\psi_1, \ldots, \psi_N)$. The reward model is then used in place of the true reward function to train the agent's policy π_ϕ. Policy π_ϕ and reward function $\hat{r}_\psi$ are updated by interleaving the following two steps:

- *Step 1 (agent learning)*: The agent interacts with the environment using policy π_ϕ to collect trajectories. The policy is updated via a conventional RL algorithm to maximize the expected return of the reward model $\hat{r}_\psi$.
- *Step 2 (reward learning)*: Preference queries are generated and selected from the collected trajectories. The responses are used to update the reward model $\hat{r}_\psi$ to better fit the human feedback.

In principle, any RL algorithm could be employed in Step 1. Following PEBBLE (Lee et al., 2021b), we use the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018) in our experiments.

Pairwise Comparison Queries in RLHF Human feedback is commonly collected in the form of pairwise comparison queries (PCQ) in RLHF (Christiano et al., 2017; Lee et al., 2021b; Liang et al., 2022; Park et al., 2022; Hu et al., 2024). Given trajectory segments σ^0 and σ^1 represented by a sequence of states and actions, a pairwise comparison query is asked to the oracle (e.g., human labeler) to indicate which segment is preferred. The oracle expresses preference by giving feedback $y_{\text{PCQ}} \in \{(1, 0), (0, 1)\}$, where $(1, 0)$ means segment σ^0 is preferred over σ^1 and $(0, 1)$ means the opposite. Here we ignore the case where the two segments are considered equivalent. This query and the corresponding preference feedback is denoted as a triple $(\sigma^0, \sigma^1, y_{\text{PCQ}})$ and is stored in a dataset $\mathcal{D}_{\text{PCQ}}$.

These observed preferences are linked to the reward function by means of a Bradley-Terry model (Bradley & Terry, 1952) which assumes pairwise preferences are governed with a pair of latent utilities (p^0, p^1) by

$$P[\sigma^1 \succ \sigma^0] = \frac{\exp(p^1)}{\exp(p^0) + \exp(p^1)} \,.$$

In the context of RLHF, where the utility of a trajectory (segment) is defined as its return, the predicted probability of segment σ^1 being preferred over σ^0 is

$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp \sum_t \gamma^t \hat{r}_\psi(\boldsymbol{s}_t^1, \boldsymbol{a}_t^1)}{\sum_{i \in \{0,1\}} \exp \sum_t \gamma^t \hat{r}_\psi(\boldsymbol{s}_t^i, \boldsymbol{a}_t^i)} \,, \tag{1}$$

where $\hat{r}_\psi(s_t^j, a_t^j)$ for $j \in \{0, 1\}$ is the average output of the N reward networks $\hat{r}_{\psi_i}$ for $i \in \{1, \ldots, N\}$. Unless stated otherwise, $\gamma = 1$ in this paper.

Given dataset $\mathcal{D}_{\text{PCQ}}$ and the corresponding predictions from Equation (1), reward learning in Step 2 is formulated as a supervised classification problem (Christiano et al., 2017). The reward model $\hat{r}_\psi$ can be learned by minimizing the

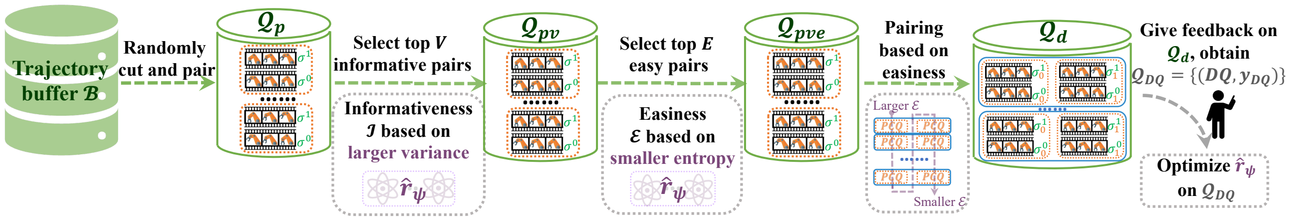

Trajectory buffer $\mathcal{B}$
Randomly cut and pair
$\mathcal{Q}_p$
σ^1
σ^0
Select top V informative pairs
Informativeness $\mathcal{I}$ based on larger variance
$\hat{r}_\psi$
$\mathcal{Q}_{pv}$
Select top E easy pairs
Easiness $\mathcal{E}$ based on smaller entropy
$\hat{r}_\psi$
$\mathcal{Q}_{pve}$
Pairing based on easiness
Larger $\mathcal{E}$
PCQ
Smaller $\mathcal{E}$
$\mathcal{Q}_d$
σ_0^1
σ_0^0
σ_1^1
σ_1^0
Give feedback on $\mathcal{Q}_d$, obtain $\mathcal{Q}_{DQ} = \{(DQ, y_{DQ})\}$
Optimize $\hat{r}_\psi$ on $\mathcal{Q}_{DQ}$

Figure 2. Our proposed query selection scheme for DQ. We first filter a subset $\mathcal{Q}_{pv}$ with higher informativeness $\mathcal{I}$ from the set of randomly generated segment pairs $\mathcal{Q}_p$, then filter a subset $\mathcal{Q}_{pve}$ with higher easiness $\mathcal{E}$ from $\mathcal{Q}_{pv}$. Both $\mathcal{I}$ and $\mathcal{E}$ are calculated from the current reward model $\hat{r}_\psi$. Finally, we pair the PCQs in $\mathcal{Q}_{pve}$ by matching the e^{th} easiest one to the $(e+\frac{|\mathcal{Q}_{pve}|}{2})^{\text{th}}$ to construct $\mathcal{Q}_d$. We obtain $\mathcal{Q}_{DQ}$ with labelers' feedback on $\mathcal{Q}_d$, and then our special training objective for $\hat{r}_\psi$ on $\mathcal{Q}_{DQ}$ can utilize not only y_{PCQ} but also d.

cross-entropy loss

$$\mathcal{L}^{\text{Rew}}_{y_{\text{PCQ}}} = - \mathop{\mathbb{E}}_{\substack{(\sigma^0,\sigma^1,y_{\text{PCQ}}) \\ \sim \mathcal{D}_{\text{PCQ}}}} \Big[y_{\text{PCQ}}(0) \log P_\psi[\sigma^0 \succ \sigma^1] \qquad (2)$$
$$+ y_{\text{PCQ}}(1) \log P_\psi[\sigma^1 \succ \sigma^0] \Big].$$

4. Method

Although pairwise comparisons are widely used and have shown impressive performance in simulated environments (Christiano et al., 2017; Lee et al., 2021b; Liang et al., 2022; Park et al., 2022; Hu et al., 2024; Verma & Metcalf, 2024), this query type is challenging in practical applications with real humans involved, especially when the humans face hard-to-answer queries. The case of equally preferable choice alternatives results in a waste of queries and bad user experience, thereby limiting application of RLHF methods to real-world scenarios. Our work aims to address this issue by proposing a new type of query, the distinguishability query, which is designed to be both informative and easier to answer for the human labeler.

We first introduce this novel query type in Section 4.1. We further design an efficient query selection scheme tailored to this type of query and explicate the procedure in Section 4.2. Finally, a special training objective is proposed to fully utilize information from such queries in Section 4.3. With these three components, we hypothesize that the queries posed to the labeler will be informative and easy to answer, and the reward learning process will be complement this new feedback type, ensuring good user experience, query efficiency, and high final performance. See Figure 2 for the overall procedure.

4.1. Distinguishability Query

Intuitively, in order to avoid posing unanswerable pairwise comparison queries and to learn about preference strength, we propose to provide the oracle with two such queries together as one distinguishability query. We let the oracle select the more distinguishable one which is easier to answer, and then provide preference feedback to the chosen pairwise query. This effectively combines a query about ordinal preferences with one about preference strength, while simultaneously reducing the burden on the human labeler.

Recall that in Section 3, the pairwise comparison query is denoted as $\text{PCQ} = (\sigma^0, \sigma^1)$ and corresponding preference feedback is $y_{\text{PCQ}} \in \{(1,0),(0,1)\}$. Formally, we represent the **distinguishability query** as $\text{DQ} = (\text{PCQ}_0, \text{PCQ}_1) = \big((\sigma_0^0, \sigma_0^1), (\sigma_1^0, \sigma_1^1)\big)$. The feedback to a distinguishability query $y_{\text{DQ}} = (d, y_{\text{PCQ}})$ consists of two components: the **distinguishability preference feedback** $d \in \{(1,0),(0,1)\}$ indicating which pairwise comparison query is more distinguishable, and the **pairwise preference feedback** y_{PCQ} to the selected more distinguishable pairwise comparison query. Such query and corresponding feedback is represented by $(\text{DQ}, y_{\text{DQ}})$ and is stored in a dataset $\mathcal{D}_{\text{DQ}}$. See Figure 1 for an illustration.

We define the **distinguishability measurement** $\mathcal{M}$ for a pairwise comparison query (σ^0, σ^1) as

$$\mathcal{M}(\sigma^0, \sigma^1) = \left| \sum_t \gamma^t r(\boldsymbol{s}_t^1, \boldsymbol{a}_t^1) - \sum_t \gamma^t r(\boldsymbol{s}_t^0, \boldsymbol{a}_t^0) \right|. \quad (3)$$

Larger $\mathcal{M}$ indicates stronger distinguishability. Then in the context of RLHF, given the reward model $\hat{r}_\psi$, the corresponding predicted distinguishability is

$$\hat{\mathcal{M}}_\psi(\sigma^0, \sigma^1) = \left| \sum_t \gamma^t \hat{r}_\psi(\boldsymbol{s}_t^1, \boldsymbol{a}_t^1) - \sum_t \gamma^t \hat{r}_\psi(\boldsymbol{s}_t^0, \boldsymbol{a}_t^0) \right|. \quad (4)$$

We assume the predicted distinguishability preference also follows the Bradley-Terry model as below:

$$\widetilde{P}_\psi[(\sigma_1^0, \sigma_1^1) \succ (\sigma_0^0, \sigma_0^1)] = \frac{\exp \hat{\mathcal{M}}_\psi(\sigma_1^0, \sigma_1^1)}{\sum_{h \in \{0,1\}} \exp \hat{\mathcal{M}}_\psi(\sigma_h^0, \sigma_h^1)}, \quad (5)$$

where $(\sigma_1^0, \sigma_1^1) \succ (\sigma_0^0, \sigma_0^1)$ represents that pairwise comparison query (σ_1^0, σ_1^1) is predicted to be more distinguishable than the other.

4.2. Selecting Informative and Easy-to-Answer Distinguishability Queries

Given the newly proposed type of query, we specifically design a method to select distinguishability queries which are informative and also easy to address for the oracle as shown in Figure 2. Overall, we begin with selecting desirable pairwise comparison queries, and pair the selected ones into distinguishability queries.

Broadly speaking, we aim to select queries on which our current predictive uncertainty could be reduced by feedback from the oracle, while also ensuring that the oracle can easily provide feedback. This aligns well with the concepts of epistemic and aleatoric uncertainty (Hüllermeier & Waegeman, 2021): While the latter refers to inherent uncertainty due to randomness in the data-generating process (in our case the labeler's responses), the former is caused by the learner's limited knowledge of this process. Thus, while aleatoric uncertainty is irreducible, epistemic uncertainty can in principle be reduced through additional (training) information, and hence is a natural target for active learning and query construction (Nguyen et al., 2022). Since we assume the labeler gives stochastic feedback according to a Bradley-Terry model, aleatoric uncertainty is high when the utility difference is small, and low when the utility difference is large.

Our selection scheme therefore favors queries that are informative and hence epistemically uncertain, and meanwhile easy to answer by the oracle. Given the ensemble reward model, the *variance* of predictions from the reward ensemble can be taken as a measure of epistemic uncertainty of the model, and hence as a measure of the informativeness of queries (Depeweg et al., 2018). The difficulty (or easiness) of a query depends on the learner's *total* uncertainty (epistemic + aleatoric), which is commonly measured in terms of the *entropy*[1] of the aggregated ensemble prediction (Depeweg et al., 2018). Note that the quest for high epistemic but low total uncertainty implies low aleatoric uncertainty.

We next explain our method step by step in a more detailed way.

Informativeness based on Variance As Figure 2 shows, given a trajectory buffer $\mathcal{B}$, we first randomly sample segments from trajectories and pair them to obtain a set $\mathcal{Q}_p$ of candidate comparison queries. Then with the current reward model $\hat{r}_\psi$, we can compute the predicted pairwise preference probability in Equation (1) for each query $(\sigma^0, \sigma^1) \in \mathcal{Q}_p$.

We define the informativeness $\mathcal{I}$ of a query (σ^0, σ^1) as the variance of reward model prediction, which is

$$\mathcal{I}(\sigma^0, \sigma^1) = \mathcal{V}(\sigma^0, \sigma^1) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P^1_{\psi_i} - P^1_{\psi})^2}, \quad (6)$$

where $P^1_{\psi_i} = P_{\psi_i}[\sigma^1 \succ \sigma^0]$ is the prediction solely from neural network $\hat{r}_{\psi_i}$ and $P^1_{\psi} = P_{\psi}[\sigma^1 \succ \sigma^0]$ is the average prediction from the ensemble reward model. Queries with higher variance indicate higher epistemic uncertainty of the current reward model, thus providing more information for reward learning. In this step, pairwise comparison queries with top V informativeness are finally selected from set $\mathcal{Q}_p$ for later steps.

Easiness based on Entropy Let $\mathcal{Q}_{pv}$ be the set of the V informative queries obtained in the last step. Here we define the easiness $\mathcal{E}$ of a query (σ^0, σ^1) as the negative entropy of reward model prediction, which is

$$\mathcal{E}(\sigma^0, \sigma^1) = -\mathcal{H}(\hat{r}_\psi) = \sum_{j=0}^{1} P^j_\psi \log P^j_\psi, \quad (7)$$

where $P^j_\psi = P_\psi[\sigma^j \succ \sigma^{1-j}]$ for $j \in \{0, 1\}$ represents the average prediction from the ensemble reward model.

It is worthwhile to mention that although higher entropy values may also correspond to more uncertain predictions, and the entropy criterion has been used for pairwise comparison query selection (Lee et al., 2021b), relying only on the highest entropy can result in queries that are nearest to the decision boundary and thus really hard for the oracle to answer. By selecting queries with the smallest entropy among the ones selected with the largest variance, we guarantee that the queries eventually selected are as easy to answer as possible, despite the epistemic uncertainty involved.

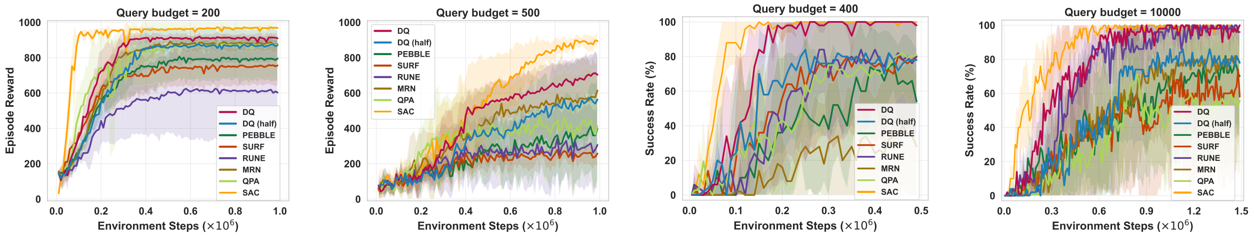
In this step, we compute easiness $\mathcal{E}$ for each $(\sigma^0, \sigma^1) \in \mathcal{Q}_{pv}$ and further select a subset of queries with top E easiness for the last step.

Forming Distinguishability Queries Let $\mathcal{Q}_{pve}$ be the set of the E easy while informative pairwise comparison queries obtained in the last step. Ordering these queries based on easiness $\mathcal{E}$ from high to low, we then pair the e^{th} and $(e + \frac{E}{2})^{\text{th}}$ ones into a distinguishability query, where $e \in \{0, \ldots, \frac{E}{2}\}$. Finally, a set $\mathcal{Q}_d$ of $\frac{E}{2}$ distinguishability queries is obtained and asked to the oracle for feedback to obtain $\mathcal{Q}_{DQ}$.

4.3. Training with Distinguishability Query

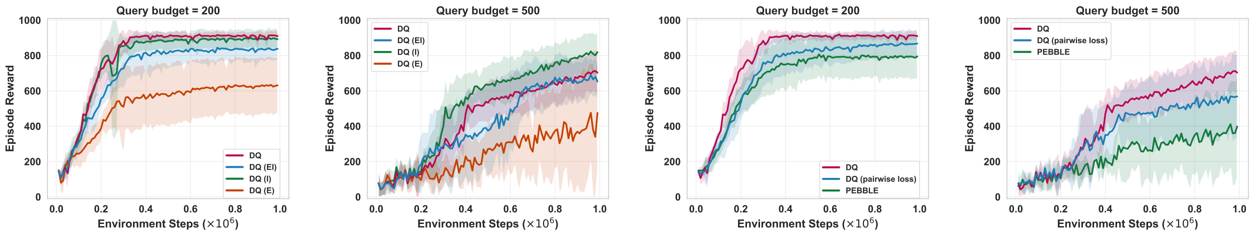
As mentioned in Section 4.1, the distinguishability feedback $y_{\text{DQ}} = (d, y_{\text{PCQ}})$ consists of both the distinguishability preference feedback d and the pairwise preference feedback y_{PCQ}. Given the assumption shown in Equation (5) that the predicted distinguishability preference also follows the

[1] An alternative is the Gini index, which fits theoretically with variance as epistemic uncertainty. Practically, there is basically no difference between these measures.



(a) Walker walk (b) Quadruped walk (c) Window open (d) Sweep into

Figure 3. Learning curves on locomotion and robotic manipulation tasks. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.



(a) Walker walk (b) Quadruped walk (c) Walker walk (d) Quadruped walk

Figure 4. Ablation study on locomotion tasks. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

Bradley-Terry model, we can train the reward model with additional information by formulating the distinguishability preference prediction also as a supervised classification problem.

Analogous to Equation (2), we define the cross-entropy loss for the distinguishability preference feedback d as

$$\mathcal{L}^{\text{Reward}}_{y_\text{d}} = - \mathop{\mathbb{E}}_{((\sigma_0^0,\sigma_0^1),(\sigma_1^0,\sigma_1^1),y_\text{DQ})\sim\mathcal{D}_\text{DQ}} \Big[\sum_{j=0}^{1} d(j)\log \widetilde{P}_\psi[(\sigma_j^0,\sigma_j^1)\succ(\sigma_{1-j}^0,\sigma_{1-j}^1)]\Big]. \tag{8}$$

Without loss of generality, assume (σ_1^0, σ_1^1) is the more distinguishable one. We can define the cross-entropy loss for the pairwise preference feedback y_PCQ in the context of distinguishability query as

$$\mathcal{L}^{\text{Reward}}_{y_\text{PCQ}} = - \mathop{\mathbb{E}}_{((\sigma_0^0,\sigma_0^1),(\sigma_1^0,\sigma_1^1),y_\text{DQ})\sim\mathcal{D}_\text{DQ}} \Big[\sum_{j=0}^{1} y_\text{PCQ}(j)\log P_\psi[\sigma_1^j\succ\sigma_1^{1-j}]\Big]. \tag{9}$$

We finally update the reward model $\hat{r}_\psi$ by minimizing the linear combination of $\mathcal{L}^{\text{Reward}}_{d}$ and $\mathcal{L}^{\text{Reward}}_{y_\text{PCQ}}$ as

$$\mathcal{L}^{\text{Reward}}_{y_\text{DQ}} = \lambda_d \mathcal{L}^{\text{Reward}}_{y_\text{d}} + \lambda_p \mathcal{L}^{\text{Reward}}_{y_\text{PCQ}}, \tag{10}$$

where λ_d and λ_p denote the weight for $\mathcal{L}^{\text{Reward}}_{d}$ and $\mathcal{L}^{\text{Reward}}_{y_\text{PCQ}}$, respectively. The full procedure of DistQ is summarized in Algorithm 1.

5. Experiments

In this section, we conduct experiments to investigate the following questions:

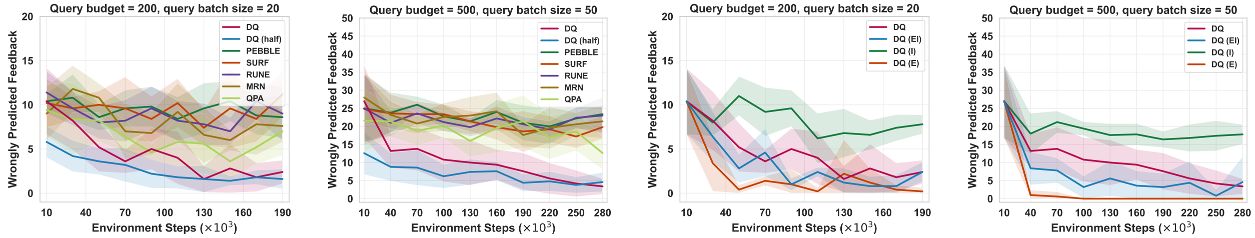
1. How do the proposed distinguishability query and corresponding query selection method help with performance and query efficiency compared with state-of-the-art (SOTA) RLHF methods that only utilize pairwise comparison queries?

2. Are the pairwise comparison queries selected by our method easier to answer compared with the ones selected by baseline methods?

3. How does each proposed technique contribute to the overall design?

5.1. Experimental Setup

Tasks Similar to prior works (Lee et al., 2021b;a; Park et al., 2022; Liang et al., 2022; Liu et al., 2022; Hu et al., 2024), we consider a series of continuous control tasks including locomotion tasks from DeepMind Control Suite



(a) Walker walk (b) Quadruped walk (c) Walker walk (d) Quadruped walk

Figure 5. Wrongly predicted feedback to pairwise comparison queries on locomotion tasks. The solid line and shaded regions represent the mean and standard deviation, respectively, across five runs.

(DMControl) (Tassa et al., 2018) and robotic manipulation tasks from Meta-World benchmark (Yu et al., 2019).

To quantitatively evaluate the performance of involved RLHF methods, we follow a general setting where the agent has no access to the ground truth reward from the environment but can only receive synthetic feedback based on the ground truth reward from a scripted labeler. Unless stated otherwise, we consider a perfectly rational labeler in our experiments. Given the feedback, the agent learns to solve the corresponding task guided by the underlying reward function. The performance is then measured as the true average return for locomotion tasks and success rate for manipulation tasks. We report the mean and standard deviation across five runs for all experiments.

Baselines For comparison, we adopt a variety of SOTA methods in the field of RLHF, including PEBBLE (Lee et al., 2021b), SURF (Park et al., 2022), RUNE (Liang et al., 2022), MRN (Liu et al., 2022), and QPA (Hu et al., 2024). These methods all utilize pairwise comparison queries. As for query selection, PEBBLE adopts an entropy-based method (i.e., smaller $\mathcal{E}$), while the other four, which also take PEBBLE as the backbone algorithm, but adopt an variance-based method (i.e., larger $\mathcal{I}$). All baselines are evaluated with the original settings listed in their paper. More details are provided in Appendix A. What is more, considering all these methods employ SAC for agent learning, we also measure the performance of SAC using the ground truth reward function as an upper bound of performance.

Implementation We implement the distinguishability query and the query selection method on top of the widely-adopted method PEBBLE (Lee et al., 2021b). This implementation is then evaluated and compared with all baselines. We argue that the proposed new query and corresponding query selection method can actually be implemented on top of any RLHF method utilizing pairwise comparison queries. See Appendix B for more implementation details.

5.2. Benchmark Tasks with Unobserved Rewards

Figure 3 shows the learning curves of our method DistQ and the five baselines on two locomotion tasks (i.e., Walker walk and Quadruped walk) and two robotic manipulation tasks (i.e., Window open and Sweep into). All baseline methods utilize a budget of pairwise comparison queries indicated by ”Query budget” in each sub-figure. Note that for our method, although a distinguishability query is composed of two pairwise comparison queries, we only ask **the more distinguishable one** to the oracle. Therefore, we show the results of our method using both full budget (i.e., the DistQ curve) and half budget (i.e., the DistQ (half) curve) for a straightforward and fair comparison, instead of simply considering the latter case. Take Figure 3(a) as an example, curves of all baseline methods are obtained by asking 200 pairwise comparison queries. The curve of DistQ (dark pink) and the curve of DistQ (half) (blue) are obtained with 200 and 100 distinguishability queries, respectively.

Locomotion Tasks from DMControl As shown in Figures 3(a) and 3(b), we find that DistQ with full budget outperforms all baseline methods, which meets our expectation since distinguishability queries naturally provide more information for reward learning, leading to a better reward model to guide the agent learning. Notably, DistQ with half budget also outperforms most baselines only except for MRN. However, we still achieve similar performance to MRN without too much decrease.

Recall that here DistQ is implemented on top of PEBBLE. The other baselines based on PEBBLE mainly focus on different aspects to improve query efficiency and performance, like exploration, unlabeled data augmentation, and new training procedure for the agent, which is orthogonal to DistQ. Therefore, we expect that implementing DistQ specifically on each of the baselines could certainly bring about improvement accordingly.

Robotic Manipulation Tasks from Meta-World Similar phenomena also occurred in Figures 3(c) and 3(d). DistQ with full budget still outperforms all baselines and even converges to the same performance as SAC (yellow). The version with half budget also exhibits better or similar performance compared with baselines with the exception for RUNE on Sweep into.

Above results effectively demonstrate that the proposed DistQ can not only guarantee performance but also query efficiency. The new type of query provides adequate information to reach satisfying performance through the distinguishability preference feedback while reduces oracle's effort on answering too many pairwise comparion queries.

5.3. Query Easiness

During each feedback session in reward learning, a batch of pairwise comparison queries are selected and asked to the oracle for feedback. Meanwhile, we can use the current reward model to predict feedback to the selected queries. If the predicted feedback is not consistent with the ground-truth feedback from the oracle, corresponding pairwise comparison query is expected to be hard to answer.

To investigate whether we ask easier pairwise comparison queries than the baseline methods, we display the curves of wrongly predicted feedback to pairwise comparison queries along the training process of various methods in Figures 5(a) and 5(b). Note that here DistQ with half budget (blue) also adopts half of the query batch size. We see that among all methods, DistQ with full budget (pink) always enjoys the least wrongly predicted feedback given the same number of pairwise comparison queries are asked to all baselines, which effectively support our argument that DistQ can really ask easier-to-answer queries.

5.4. Ablation Study

To figure out how the proposed techniques contribute to the final performance of DistQ, we carry out ablation study on the query selection method and the newly designed loss function, respectively.

For query selection, we individually change the order of informativeness and easiness (EI) selection and keep only informativeness (I) or easiness (E) selection, for which the results on two locomotion tasks are shown in Figures 4(a) and 4(b). To deeply understand how these different ablations affect the final performance, we also measure the easiness of queries generated by each ablation as in Section 5.3. Corresponding results are shown in Figures 5(c) and 5(d). We see that changing the order (EI, blue) actually has different influences on different tasks in terms of episode reward, which may be due to the query distributions with regard to informativeness and easiness are different on different tasks. However, keeping only one of the selection criteria does shed light on the effectiveness of our method. With only informativeness (I, green), though performing (nearly) the best, it generates the most wrongly predicted pairwise feedback, which implies that the selected queries may be hard to answer. The opposite case is with only easiness (E, orange) which generates the least wrong prediction but also suffers from the worst performance at the same time. Therefore, two techniques individually help in one aspect and together they make our method (pink) enjoy both good performance and asking easier queries.

For the new loss function $\mathcal{L}^{\text{Reward}}_{y_{\text{DQ}}}$ in Equation (10), recall that it consists of $\mathcal{L}^{\text{Reward}}_{y_{\text{d}}}$ $\mathcal{L}^{\text{Reward}}_{y_{\text{PCQ}}}$. To figure out whether $\mathcal{L}^{\text{Reward}}_{y_{\text{d}}}$ helps and whether the pairwise comparison queries selected by DistQ are more helpful, we compare the performance of DistQ trained with only $\mathcal{L}^{\text{Reward}}_{y_{\text{PCQ}}}$ in Equation (10), DistQ trained normally, and PEBBLE trained normally in Figures 4(c) and 4(d). On both tasks, training with $\mathcal{L}^{\text{Reward}}_{y_{\text{PCQ}}}$ (blue) hurts for DistQ (pink), which is obvious since we lose the information in $\mathcal{L}^{\text{Reward}}_{y_{\text{d}}}$ for learning a good reward model. However, training with $\mathcal{L}^{\text{Reward}}_{y_{\text{PCQ}}}$ (blue) surprisingly performs better than PEBBLE (green) which is also trained with only pairwise preference loss. This effectively demonstrates that with the same number of pairwise queries, the ones selected by DistQ are more helpful.

6. Discussion

Conclusion In this paper, we present DistQ which consists of a new type of query, the distinguishability query, and a corresponding efficient query selection method considering both informativeness and easiness of queries. Along with a specifically designed loss function, DistQ is proposed to achieve more informative and user-friendly RLHF. Extensive experiments demonstrate that DistQ outperforms current SOTA baselines in RLHF with regard to query efficiency and performance on a variety of locomotion and robotic manipulation tasks. Besides, DistQ exhibits considerable potential in generating easier queries to answer, which we expect to be critical for applying RLHF in more realistic scenarios.

Limitations While we think our proposed query selection method works well, there may exist more suitable measurements for informativeness and easiness, which deserves deeper investigation. Also, for now DistQ hasn't been evaluated with real humans involved on more practical tasks, which prevents our method from further improvement. We believe that DistQ needs to be tested more thoroughly on real world domains to make its performance better understood and to potentially further improve it.

Overall, we believe that DistQ proposes an effective perspective to more informative and user-friendly RLHF.

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.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alt, F. Über die Meßbarkeit des Nutzens. *Zeitschrift für Nationalökonomie / Journal of Economics*, 7(2):161–169, 1936. ISSN 0044-3158. URL https://www.jstor.org/stable/41792549.

Bıyık, E., Palan, M., Landolfi, N. C., Losey, D. P., and Sadigh, D. Asking Easy Questions: A User-Friendly Approach to Active Reward Learning. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR, 2019. URL https://proceedings.mlr.press/v100/b-iy-ik20a.html.

Bıyık, E., Losey, D. P., Palan, M., Landolfi, N. C., Shevchuk, G., and Sadigh, D. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research*, 41(1):45–67, 2022. doi: 10.1177/02783649211041652.

Bradley, R. A. and Terry, M. E. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.2307/2334029.

Chan, L., Critch, A., and Dragan, A. Human irrationality: Both bad and good for reward inference, 2021. URL http://arxiv.org/abs/2111.06956. preprint.

Christiano, P., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-Sensitive Learning. In *Proc. ICML, 35th International Conference on Machine Learning*, pp. 1184–1193. PMLR, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. 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, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Hu, X., Li, J., Zhan, X., Jia, Q.-S., and Zhang, Y.-Q. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=UoBymIwPJR.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. doi: 10.1007/s10994-021-05946-3.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in Atari. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8cbe9ce23f42628c98f80fa0fac8b19a-Abstract.html.

Jansen, C., Schollmeyer, G., and Augustin, T. Concepts for decision making under severe uncertainty with partial ordinal and partial cardinal preferences. *International Journal of Approximate Reasoning*, 98:112–131, 2018. doi: 10.1016/j.ijar.2018.04.011.

Jansen, C., Blocher, H., Augustin, T., and Schollmeyer, G. Information efficient learning of complexly structured preferences: Elicitation procedures and their application to decision making under uncertainty. *International Journal of Approximate Reasoning*, 144:69–91, 2022. doi: 10.1016/j.ijar.2022.01.016.

Jeon, H. J., Milli, S., and Dragan, A. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2f10c1578a0706e06b6d7db6f0b4a6af-Abstract.html.

Kaufmann, T., Weng, P., Bengs, V., and Hüllermeier, E. A Survey of Reinforcement Learning from Human Feedback, 2023. URL http://arxiv.org/abs/2312.14925. preprint.

Köbberling, V. Strength of Preference and Cardinal Utility. *Economic Theory*, 27(2):375–391, 2006. ISSN 0938-2259. URL https://www.jstor.org/stable/25056023.

Krantz, D. H., Suppes, P., and Luce, R. D. *Foundations of Measurement. Volume 1: Additive and Polynomial Representations*. Courier Corporation, 2006. ISBN 978-0-486-45314-9.

Lee, K., Smith, L., Dragan, A., and Abbeel, P. 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, 2021a. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/d82c8d1619ad8176d665453cfb2e55f0-Abstract-round1.html.

Lee, K., Smith, L. M., and Abbeel, P. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2021b. URL https://proceedings.mlr.press/v139/lee21i.html.

Liang, X., Shu, K., Lee, K., and Abbeel, P. Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=OWZVD-l-ZrC.

Liu, R., Bai, F., Du, Y., and Yang, Y. Meta-Reward-Net: Implicitly Differentiable Reward Learning for Preference-based Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=OZKBReUF-wX.

Nguyen, V., Shaker, M., and Hüllermeier, E. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122, 2022. doi: 10.1007/s10994-021-06003-9.

OpenAI. Introducing ChatGPT, 2022. URL https://openai.com/index/chatgpt/. (accessed 2024-05-28).

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. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Palan, M., Shevchuk, G., Landolfi, N. C., and Sadigh, D. Learning Reward Functions by Integrating Human Demonstrations and Preferences. In *Proceedings of Robotics: Science and Systems (RSS)*, volume 15, 2019. URL http://www.roboticsproceedings.org/rss15/p23.html.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. 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)*, 2022. URL https://openreview.net/forum?id=TfhfZLQ2EJO.

Pivato, M. Multiutility representations for incomplete difference preorders. *Mathematical Social Sciences*, 66(3): 196–220, 2013. doi: 10.1016/j.mathsocsci.2013.05.003.

Suppes, P. and Winet, M. An Axiomatization of Utility Based on the Notion of Utility Differences. *Management Science*, 1(3-4):259–270, 1955. doi: 10.1287/mnsc.1.3-4.259.

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. DeepMind Control Suite, 2018. URL http://arxiv.org/abs/1801.00690. preprint.

Verma, M. and Metcalf, K. Hindsight PRIORs for Reward Learning from Human Preferences. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=NLevOah0CJ.

Von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, 2 edition, 1947.

Xu, Y., Wang, R., Yang, L., Singh, A., and Dubrawski, A. Preference-based Reinforcement Learning with Finite-Time Guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d9d3837ee7981e8c064774da6cdd98bf-Abstract.html.

Yannakakis, G. N. and Martínez, H. P. Ratings are Overrated! *Frontiers in ICT*, 2:13, 2015. doi: 10.3389/fict.2015.00013.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR, 2019. URL https://proceedings.mlr.press/v100/yu20a.html.

A. Baselines

All baseline methods considered in this paper are under the RLHF framework as we explained in Section 3. In this section, we provide more details about these baselines. Specifically, we summarize the query selection methods adopted by DistQ and baselines in Table 1.

Table 1. Query selection methods adopted by different methods.

Methods	DistQ	PEBBLE	SURF	RUNE	MRN	QPA
Query selection	Larger variance + lower entropy	Larger entropy	Larger variance			

A.1. PEBBLE

Based on the framework proposed by Christiano et al. (2017), PEBBLE (Lee et al., 2021b) uses SAC to replace the original on-policy RL implementation. Besides, PEBBLE additionally pre-trains the policy in an unsupervised way before the reward learning part to improve sample efficiency over the original preference-based reward learning framework. Specifically, it improves the basic framework described in Section 3 with three parts:

1. Unsupervised pre-training: Instead of initializing the policy model randomly at the beginning, PEBBLE pre-trains the policy model separately by maximizing the entropy of encountered states, which can obtain a better policy generating more diverse trajectories to ease the beginning of reward learning.

2. Entropy-based query selection: It first randomly samples a large batch of segment pairs, and then filters the group of pairs with larger entropy $\mathcal{H}(P_\psi)$.

3. Relabel relabeling: To stabilise the learning process of SAC, an off-policy RL method that is sensitive to inconsistent reward signals, PEBBLE re-predicts the reward signals for state-action pairs in the replay buffer once the reward model is updated.

A.2. SURF

SURF (Park et al., 2022) takes PEBBLE as its backbone algorithm and argue that it reduces the number of feedback needed from the labeler while maintaining the same or even higher level of performance. However, instead of investigating how the queries are generated or sampled, they focus on making use of more available data. SURF introduces a semi-supervised learning (SSL) approach to generate pseudo labels for unlabeled data and also proposes a specific data augmentation technique for the pairwise segments query. With the two ingredients, SURF is demonstrated to significantly improve the query-efficiency of RLHF algorithms on both locomotion and manipulation tasks mentioned in Section 5.1.

A.3. RUNE

RUNE (Liang et al., 2022) contains an intrinsic reward designed by measuring novelty based on the learned reward, and is also integrated with PEBBLE. Specifically, it utilizes disagreements across an ensemble of learned reward models. It incorporates the uncertainty of the reward function into the reward itself, which encourages the agent to explore uncertain regions of the environment.

A.4. MRN

MRN (Liu et al., 2022) leverages meta-learning to make the reward learning process implicitly differentiable, enabling the use of gradient-based optimization for learning the reward function. This framework aims to improve the use of preference data, which is critical in PbRL, as it leverages human preferences as the reward signal, thus avoiding the need for reward engineering. It also incorporates bi-level optimization, which can be seen as a method to enhance data efficiency.

A.5. QPA

QPA (Hu et al., 2024) involves near-on-policy queries and a specially designed hybrid experience replay, which together enforce the bidirectional query-policy alignment. The near on-policy query selection ensures the selection of segments from recent trajectories which are more aligned with the current policy. The hybrid experience replay involves maintaining a separate buffer that stores the most recent trajectories, effectively emulating a near-on-policy buffer's characteristics. The data augmentation generates multiple instances from a single pair, effectively expanding the preference dataset.

B. Experimental Settings

Our method DistQ is implemented based on the framework of PEBBLE. DistQ and all the baseline methods follow the general hyperparameter configurations described in Table 2. For other specific hyperparameter settings of each baseline, we follow their papers and published codes.

Table 2. Hyperparameters setting.

HYPERPARAMETER	VALUE	HYPERPARAMETER	VALUE
General settings			

Initial temperature	0.1	Hidden units per each layer	1024(DMControl)
			256(Meta-world)

Length of segment	50	# of layers	2(DMControl)
			3(Meta-world)

Learning rate	0.0003 (Meta-world)
	0.0005 (Walker)
		Batch Size	1024(DMControl)
			512(Meta-world)

	0.0001 (Quadruped)	Optimizer	Adam
Critic target update freq	2	Critic EMA τ	0.005
(β_1, β_2)	(0.9,0.999)	Discount $\bar{\gamma}$	0.99
Frequency of feedback	5000 (Meta-world) 20000 (Walker) 30000 (Quadruped)	Maximum budget / # of queries per session	500/50, 200/20 (DMControl) 10000/50, 400/10 (Meta-world)
# of ensemble models N_{en}	3	# of pre-training steps	10000
Other settings for DistQ			
Loss weights $(\lambda_d,\ \lambda_p)$	(1, 1)	Size of $\mathcal{Q}_p$	10×# of queries per session
Size of $\mathcal{Q}_{pv}$ (V)	5×# of queries per session	Size of $\mathcal{Q}_{pve}$ (E)	2×# of queries per session

C. Algorithm

C.1. Pseudo code

Since our method is based on PEBBLE, we use the normal black color to denote steps that are the same as PEBBLE, and use the orange color to highlight the different parts in Algorithm 1.

Algorithm 1 DistQ

```
Randomly initialize policy model $\pi_\phi$ and reward model $\hat{r}_\psi$
Dataset for trajectories $\mathcal{B} \leftarrow \emptyset$
Dataset for distinguishability feedback $\mathcal{Q}_{DQ} \leftarrow \emptyset$
//PRE-TRAIN
Pre-training as PEBBLE's to obtain $\mathcal{B}, \pi_\phi$
for each iteration do
  //REWARD LEARNING
  if Iteration$\% K == 0$ then
    //SAMPLING QUERIES
    Randomly sample segment pairs $\mathcal{Q}_p = \{(\sigma^0, \sigma^1)\}$ from $\mathcal{B}$
    Calculate informativeness $\mathcal{I}$ of queries in $\mathcal{Q}_p$ (Equation (6)), $\mathcal{Q}_{pv}$ are the top $V$ ones with larger $\mathcal{I}$
    Calculate easiness $\mathcal{E}$ of queries in $\mathcal{Q}_{pv}$ (Equation (7)), $\mathcal{Q}_{pve}$ are the top $E$ ones with larger $\mathcal{E}$
    Sort queries in $\mathcal{Q}_{pve}$ according to $\mathcal{E}$, pair the $e^{\text{th}}$ with the $(e + \frac{E}{2})^{\text{th}}$ to form $\mathcal{Q}_d$
    ${\mathcal{Q}_{DQ}}' \leftarrow \mathcal{Q}_d$ with feedback
    $\mathcal{Q}_{DQ} \leftarrow \mathcal{Q}_{DQ} \cup {\mathcal{Q}_{DQ}}'$
    //TRAINING $\hat{r}_\psi$ ON EXTENDED $\mathcal{Q}_{DQ}$
    for each gradient step do
      Randomly sample a minibatch $\{((\sigma_0^0, \sigma_0^1), (\sigma_1^0, \sigma_1^1), y_{\text{DQ}})\}$ from $\mathcal{Q}_{DQ}$
      Optimize $\mathcal{L}^{\text{Reward}}_{y_{\text{DQ}}}$ ( Equation (10)) with respect to $\psi$
    end for
  end if
  //COLLECT TRAJECTORIES
  for each timestep $t$ do
    Collect interaction data by $a_t \sim \pi_\phi(a_t|s_t)$, $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$
    Store $\mathcal{B} \leftarrow \mathcal{B} \cup (s_t, a_t, s_{t+1})$
  end for
  //POLICY LEARNING
  for each gradient step do
    Optimize $\pi_\phi$ with a minibatch $\{(s, a, \hat{r}_\psi(s, a), s')\}$ randomly sampled from $\mathcal{B}$
  end for
end for
```