Uncertainty-aware Preference Alignment in Reinforcement Learning from Human Feedback

Sheng Xu [1] **Bo Yue** [1] **Hongyuan Zha** [1] **Guiliang Liu** [1 *]

Abstract

Recent advances in Reinforcement Learning from Human Feedback (RLHF) typically model a reward function by maximizing its likelihood of generating observed human preferences. However, due to the diverse backgrounds of individuals, these preference signals are inherently stochastic. This inherent uncertainty in the preference signals can lead to unstable or unsafe behaviors in the process of reward and policy updates. In this work, we introduce the uncertainty-aware preference alignment in RLHF by learning a distributional reward model and a risk-sensitive policy from the offline preference dataset. Specifically, we propose a Maximum A Posteriori (MAP) objective for updating the reward associated with a trajectory. This updating process incorporates an informative prior to account for the uncertainty in human preferences. Utilizing this updated reward sample, we develop a generative reward model to represent the reward distribution. Driven by the inherent stochasticity in the reward models, we utilize the offline distributional Bellman operator and the Conditional Value-at-Risk (CVaR) metric to learn a risk-sensitive policy from the offline dataset. Experimental results show that the risk-sensitive RLHF agent can effectively identify and avoid states with significant stochasticity, thereby enabling risk-averse control in different tasks.

1. Introduction

In recent years, Reinforcement Learning has achieved remarkable success in addressing a variety of sequential decision problems across different domains, such as electronic games (Mnih et al., 2015; Vinyals et al., 2019; Berner et al., 2019), board games (Silver et al., 2016; 2017), and robotic manipulation (Fang et al., 2019). However, in the process of scaling these successes to real-world applications, a notable challenge is the difficulty in precisely specifying the rewards in the RL objective. Existing manually designed rewards cannot ensure that the learned policy aligns with the actual needs of the industry.

To develop a reliable reward function, Reinforcement Learning from Human Feedback (RLHF), also known as Preference-based Reinforcement Learning (PbRL) (Knox & Stone, 2008), aims to align rewards with human preferences by solving a learning-to-rank problem. Recent advances in RLHF algorithms (Christiano et al., 2017; MacGlashan et al., 2017; Lee et al., 2021; Kim et al., 2023; Wu et al., 2023b; Zhu et al., 2023; Casper et al., 2023; Zhan et al., 2024a) commonly learn reward functions by maximizing the likelihood of the observed pairwise comparisons in the preference dataset. In practice, this preference dataset is collected from a diverse pool of individuals with varying backgrounds, knowledge, and beliefs. Consequently, these preference signals are inherently stochastic. However, driven by the Bradley-Terry model (Bradley & Terry, 1952), the maximum likelihood estimation used in RLHF lacks sensitivity to the inherent uncertainty in human preferences, thus the resulting policy tends to be risk-neutral without considering the safety of sequential decisions.

Striving for uncertainty-aware reward learning, a recent study (Liang et al., 2022) developed an ensemble of deterministic reward functions and employed the variance of model predictions to assess the "novelty" of the learned rewards. These novelty signals are primarily designed to guide exploration in online settings, whereas a more common application involves learning from offline datasets. More importantly, despite its empirical success, the underlying mechanism linking estimated prediction variance to uncertainty in human preferences remains largely unexplained.

In this paper, we introduce an uncertainty-aware RLHF algorithm that accounts for the confidence levels in human preferences within an offline dataset. Specifically, we formulate a Maximum A Posteriori (MAP) objective for preference alignment, thereby incorporating signals from an

[*]Corresponding Author [1]School of Data Science, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China . Correspondence to: Guiliang Liu <liuguiliang@cuhk.edu.cn>.

ICML 2024 Workshop on Models of Human Feedback for AI Alignment, Vienna, Austria. PMLR 235, 2024.

informative prior to the reward updates. By interpreting preference assignment as a voting process, we intentionally select Beta distribution to implement this prior. To ensure computational tractability, we parameterize the Beta distribution with neural functions and train the model via variational inference, guided by an Evidence Lower Bound (ELBo) objective. Intuitively, this probabilistic prior assigns higher probabilities to trajectories that are compared more frequently, reflecting the confidence levels associated with human preferences. For each observed trajectory, we compute its reward using an iterative update rule, which is theoretically motivated by the optimality conditions of our MAP objective. From these point-wise reward estimates, we then construct the reward distribution by training a generative reward model based on sequential state-action pairs.

To justify the effectiveness of the learned reward distribution, we train a risk-averse policy using the offline distributional Bellman operator for policy evaluation, with the Conditional Value-at-Risk (CVaR) metric for policy improvement. Empirical results demonstrate the importance of considering the inherent uncertainty in the preference signals and the risk-sensitive ability of the proposed method, which outperforms other baselines in terms of the worst-case performance.

2. Related Works

In this section, we introduce the previous works that are most related to our approach.

Reinforcement Learning from Human Feedback. Unlike classic RL algorithms (Sutton & Barto, 2018) that rely on pre-defined rewards to guide policy updates, RLHF considers aligning the policy with human preferences, circumventing the requirement for explicit reward signals (Knox & Stone, 2008; MacGlashan et al., 2017; Warnell et al., 2018). Such paradigm is particularly useful in applications where defining precise reward functions is challenging, but human feedback is readily available (e.g., dialogue system (Yang et al., 2023), question answering (Nakano et al., 2021), text summarization (Stiennon et al., 2020), language model training (Bai et al., 2022; Wu et al., 2023b) and virtual game agents (Ibarz et al., 2018)). Previous studies combine RLHF to Deep RL agent (Christiano et al., 2017) and high-dimensional image space (Ibarz et al., 2018). To scale RLHF to more settings, recent advancements extend RLHF to unsupervised pre-learning (Lee et al., 2021), non-Markovian rewards (Kim et al., 2023), offline RL setting (Zhan et al., 2024a), diffusion planner (Dong et al., 2024), k-wise comparison (Zhu et al., 2023) and reward-agnostic setting (Zhan et al., 2024b). Some recent studies (Rafailov et al., 2023; An et al., 2023; Song et al., 2024; Yuan et al., 2023; Liu et al., 2024) consider supervised fine-tuning that directly optimizes generative models with human preferences. The majority of RLHF algorithms utilize the Bradley-Terry model (Bradley & Terry, 1952) to model the likelihood of human preference based on reward signals. However, such a maximum likelihood method is insensitive to the underlying confidence in human preference (Newman, 2023). How to handle the uncertainty in human preference and derive risk-sensitive policies remain a critical challenge (Casper et al., 2023).

Distributional Reinforcement Learning. While the majority of RL research traditionally focuses on maximizing the expected cumulative rewards (Sutton & Barto, 2018), (Bellemare et al., 2017) introduces a distributional perspective on RL, utilizing the distributional Bellman operator for value function updates. Such distributional value functions are sensitive to the aleatoric uncertainty in the environment dynamics (Mavrin et al., 2019), enabling the formulation of risk-sensitive policies (Lim & Malik, 2022; Keramati et al., 2020) and better controlling performance (Bellemare et al., 2023). Some previous studies propose utilizing categorical distribution (Bellemare et al., 2017; Sui et al., 2023), quantile functions (Dabney et al., 2018b;a; Zhang & Yao, 2019; Zhou et al., 2020; 2021; Luo et al., 2022) and diffusion models (Wu et al., 2023a) for representing and updating the distributional value function. In this work, we utilize quantile functions since the statistical benefit of quantile regression is most well-understood (Rowland et al., 2023). Some recent studies extend distributional RL to offline learning (Ma et al., 2021; Wu et al., 2023a), multi-dimensional rewards (Zhang et al., 2021), and multi-agent control (Hu et al., 2022; Sun et al., 2021). However, none of the previous works have considered Preference-based RL (PbRL) from a distributional perspective.

3. Problem Formulation

Markov Decision Process (MDP). The agent optimizes the control policy under a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, p_{\mathcal{T}}, \mu_0, \gamma)$, where 1) $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, 2) $p_{\mathcal{T}} : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ denotes the stochastic transition function, where the simplex over $\mathcal{S}$, $\Delta^{\mathcal{S}} = \{\nu \in [0,1]^{\mathcal{S}} : \sum_{s\in\mathcal{S}} \nu(s) = 1\}$, 3) $R : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$ denotes the reward function, 4) $\mu_0 \in \Delta^{\mathcal{S}}$ denotes the initial state distribution, and 5) $\gamma \in (0, 1]$ denotes the discounting factor. For brevity, we denote $\mathcal{M}_{/R}$ to denote the MDP without knowing the reward. In this work, we mainly study the episodic MDPs where the planning stops at a terminating state $\tilde{s}$, and the corresponding terminating time is denoted as $T \in (0, \infty)$.

Risk-Sensitive Reinforcement Learning. Under an MDP, the objective is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$, which optimizes the following objective:

$$\pi = \arg\max_{\pi} \rho^{\alpha}_{\mu_0, p_{\mathcal{T}}, \pi}[\sum_{t=0}^{T} R(s_t, a_t)].$$

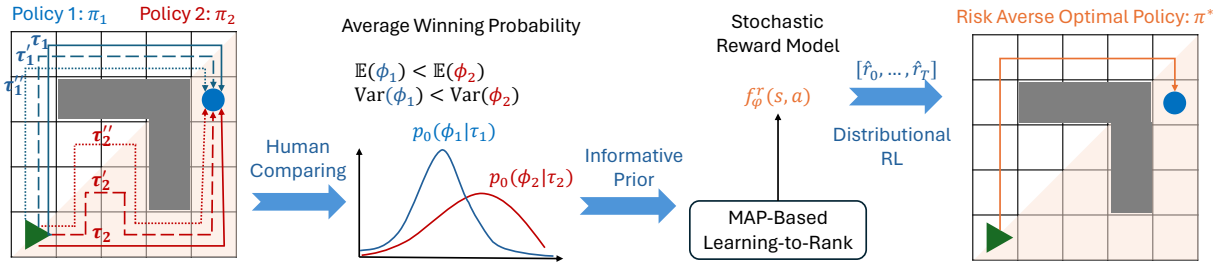

Figure 1. An illustrative example of D-RLHF. In this maze environment, states exhibit greater transition stochasticity in the bottom-right map (shaded in light orange color). Humans uniformly sample and compare pairs of trajectories generated by deterministic policies π_1 and π_2. The trajectory τ_2 is shorter in expectation, so τ_2 is more likely to outperform the average trajectory compared to the longer τ_1. However, its estimation exhibits a higher variance because the greater stochasticity induces π_2 to generate more diverse trajectories, leading to fewer comparisons from humans and thus greater uncertainty for each trajectory. Such uncertainty is captured by the distributional reward model, which steers the risk-averse policy to navigate through a less stochastic but longer path on the top-left map.

Instead of optimizing the risk-neutral expected cumulative rewards, we consider a risk-sensitive measure $\rho^{\alpha}_{\mu_0,p_\mathcal{T},\pi}$ where the confidence level $\alpha < 1$. Specifically, by implementing π under the MDP $\mathcal{M}$, we generate a trajectory $\tau \in (\mathcal{S} \times \mathcal{A})^T$. The corresponding trajectory-generating probability can be defined as $p^\pi(\tau) = \mu_0(s_0)\prod_{t=0}^{T-1}\pi(a_t|s_t)p_\mathcal{T}(s_{t+1}|s_t,a_t)$. We define the corresponding risk envelope $\mathcal{U}^\pi_\alpha = \{\zeta_\alpha : \Gamma \to \left[0, \frac{1}{\alpha}\right] | \sum_{\tau\in\Gamma}\zeta(\tau)p^\pi(\tau) = 1\}$ to be a compact, convex, and bounded set, based on which the risk measure can be induced by the distorted probability distribution $p^\pi_\zeta = \zeta \cdot p^\pi$. In this work, we study the CVaR such that $\rho^\pi_\alpha[\sum_{t=0}^T \gamma^t R_t] = \sup_{\zeta_\alpha\in\mathcal{U}^\pi_\alpha}\mathbb{E}_{\tau\sim p^\pi}[\zeta_\alpha(\tau)\sum_{t=0}^T\gamma^t R_t]$ due to its time consistency and convexity (Rockafellar et al., 2000).

Distributional Reinforcement Learning from Human Feedback (D-RLHF). In many applications, the rewards are not readily available, and the RLHF system learns the reward function from human feedback. Previous research in RLHF typically learns a deterministic reward function (Christiano et al., 2017; Lee et al., 2021; Kim et al., 2023; Casper et al., 2023) under a maximum likelihood objective. However, in real-world applications, human preferences are commonly collected from a heterogeneous pool of individuals with varying backgrounds, knowledge, and beliefs. As a result, these preference signals are inherently stochastic (Swamy et al., 2024). Intuitively, if a trajectory and its counterparts have only been assessed a few times, the corresponding preference signals should have large uncertainty. The more frequently humans compare these trajectories with their counterparts, the more confidence we have about the optimality of these trajectories.

To better accommodate the underlying uncertainty in human preference, we study an uncertainty-aware objective for achieving D-RLHF. Specifically, we capture the uncertainty of human preferences by learning a distributional reward model $f^r_\varphi : \mathcal{S} \times \mathcal{A} \to \Delta^{[R_{\min}, R_{\max}]}$ (Section 4), and incorporate these uncertainty signals into policy learning by utilizing the offline distributional policy evaluation and the risk-averse policy improvement (Section 5).

To better align the uncertainty in preferences with environmental stochasticity, we assume that the preference dataset is generated as follows:

Procedure 3.1. *(Preference Dataset). For each candidate policy $\pi_l \in [\pi_1, \ldots, \pi_L]$, we generate N_l trajectories in the environment $\mathcal{M}_{/R}$. We uniformly sample a pair of them (τ, τ') from a total number of $N_l \cdot L$ generated trajectories. Humans express preferences by mapping (τ, τ') to (τ^i, τ^j) where τ^i ranks higher than τ^j. We repeat the sampling and mapping process until we generate the dataset $\mathcal{D}$.*

This sampling procedure is essential to align the confidence of human preference to the underlying aleatoric uncertainty in environment dynamics. For example, in Figure 1, policy π_2 visits states with stochastic transitions, resulting in the generation of more diverse trajectories $\tau^2_1, \ldots, \tau^2_{N_2}$, compared to the deterministic trajectories $\tau^1_1, \ldots, \tau^1_{N_1}$ from π_1. Owing to this diversity, a specific trajectory such as $\tau^2_{n_2}$ and its similar counterparts are less likely to be chosen during uniform sampling. Consequently, these trajectories receive fewer human evaluations, resulting in lower confidence in assessing their preferences.

Based on the dataset $\mathcal{D}$, we consider the **Offline D-RLHF** problem, where the agent has access solely to an offline dataset that records labeled trajectories instead of interacting directly with environments.

4. Learning Generative Reward Model from Human Feedback

In this section, we introduce our approach to estimating the stochastic reward model by proposing 1) an MAP objective for inferring rewards from human preference, 2) an informative Beta-prior for modeling uncertainty, and 3) the method of learning generative rewards.

4.1. Maximum A Posteriori Objective for Reward Inference

Previous RLHF algorithms commonly utilize the Bradley-Terry model (Bradley & Terry, 1952) to represent the log-likelihood of generating human preferences with the reward function:

$$\mathcal{L}(\varphi, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\tau^i, \tau^j) \in \mathcal{D}} \omega^{\tau^i, \tau^j} \log \frac{e^{[r_\varphi(\tau^i)]}}{e^{[r_\varphi(\tau^i)]} + e^{[r_\varphi(\tau^j)]}}, \tag{1}$$

where $r_\varphi(\tau) = \sum_{t=0}^{T} \gamma^t r_\varphi(s_t, a_t)$ denotes the trajectory segment rewards (φ represents the reward model parameter), (τ^i, τ^j) denotes a pair of trajectories where τ^i ranks higher than τ^j, and ω^{τ^i, τ^j} denotes the frequency of such pairs appearing in the dataset $\mathcal{D}$. The maximum likelihood objective implicitly imposes a uniform prior for $r_\varphi(\tau)$ such that $p_0(r_\varphi(\tau)) = 1/(\frac{R_{\max}}{1-\gamma} - \frac{R_{\min}}{1-\gamma})$. It places a vanishing fraction of its weight on arbitrarily large values (intuitively, a large $r_\varphi(\tau)$ increases p_0 to 1 in an exponential rate, which is not likely to be observed in practice), which causes divergence in the reward function's parameters (Newman, 2023).

To derive a more useful prior for the reward function, we enforce the geometric mean strength to be one ($\prod_n e^{r_\varphi(\tau^n)} = 1$, i.e., $\sum_n r_\varphi(\tau^n) = 0$) and thus the probability of a player with strength $e^{r_\varphi(\tau)}$ winning against the average player (whose strength $e^{r_\varphi(\tau')} = 1$) is $\phi(\tau) = e^{r_\varphi(\tau)}/(e^{r_\varphi(\tau)} + 1)$. For brevity, we use ϕ as a shorthand of $\phi(\tau)$ to represent the meaning of prior. Then the prior on the reward function can be defined as:

$$\begin{aligned} p_0(r_\varphi(\tau)) &= p_0(\phi) \frac{d\phi}{dr_\varphi(\tau)} = p_0(\phi) \frac{d\phi}{de^{r_\varphi(\tau)}} \frac{de^{r_\varphi(\tau)}}{dr_\varphi(\tau)} \\ &= p_0(\phi) \frac{e^{r_\varphi(\tau)}}{(e^{r_\varphi(\tau)} + 1)^2}, \end{aligned} \tag{2}$$

where $p_0(\phi)$ is the prior which has different representations. This update enables the definition of an MAP objective:

$$\begin{aligned} &p(r_\varphi(\tau)|\mathcal{D}) \propto p(\mathcal{D}|r_\varphi(\tau)) p_0(r_\varphi(\tau)) \\ &= \prod_{(\tau^i, \tau^j) \in \mathcal{D}} \left[\frac{e^{[r_\varphi(\tau^i)]}}{e^{[r_\varphi(\tau^i)]} + e^{[r_\varphi(\tau^j)]}} \right]^{\omega^{\tau^i, \tau^j}} \prod_{\tau^i} p_0(\phi) \frac{e^{[r_\varphi(\tau^i)]}}{(e^{[r_\varphi(\tau^i)]} + 1)^2}. \end{aligned} \tag{3}$$

Instead of maximizing the likelihood, maximizing this posterior probability can integrate prior knowledge and regularize the reward values, preventing them from diverging.

An essential prerequisite for implementing D-RLHF with this MAP objective is the construction of an informative prior, $p_0(\phi)$. This prior incorporates the inherent uncertainty of human preferences into the reward learning process. We introduce the estimation of $p_0(\phi)$ in the following section.

4.2. Learning Informative Beta Priors from Human Preference

In this study, we employ the Beta distribution as an informative prior, i.e., $p_0(\phi|\mathcal{D}) = \text{Beta}(\alpha, \beta)$, since 1) the Beta distribution is the conjugate prior for the Bernoulli distribution, facilitating the update of our beliefs with new evidence; 2) the parameters α and β of the Beta distribution can be effectively interpreted as representing the count of positive and negative human feedback, respectively, for a trajectory τ. As the number of such 'votes' increases, our confidence in the inferred probability improves, resulting in a more precise (or 'sharper') distribution. This approach enables quantitatively incorporating the confidence level of the Bernoulli probability estimation into our model.

To learn the distribution of ϕ, we propose the variational inference approach to approximate $p_0(\phi|\mathcal{D})$ by estimating the approximate posterior $q_\psi(\phi|\mathcal{D})$ (i.e., $p_0(\phi|\mathcal{D}) \simeq q_\psi(\phi|\mathcal{D})$). The goal of our variational inference approach is to learn an approximate posterior distribution $q_\psi(\phi|\mathcal{D})$ by minimizing the Kullback–Leibler (KL) divergence $D_{kl}(q_\psi(\phi|\mathcal{D}) \| p(\phi|\mathcal{D}))$:

$$\begin{aligned} &D_{kl}\Big(q_\psi(\phi|\mathcal{D}) \| p(\phi|\mathcal{D})\Big) \\ &= -\mathbb{E}_{r \sim q}\Big[\log p(\mathcal{D}|\phi)\Big] + D_{kl}\Big[q_\psi(\phi|\mathcal{D}) \| p(\phi)\Big] + \log\Big[p(\mathcal{D})\Big]. \end{aligned} \tag{4}$$

Minimizing the above objective is equivalent to maximizing the Evidence Lower Bound (ELBo) $\log[p(\mathcal{D})] - D_{kl}(q_\psi(\phi|\mathcal{D}) \| p(\phi|\mathcal{D}))$. By following Equation (5), ELBo can be represented as:

$$\mathbb{E}_{r \sim q}\Big[\log p(\mathcal{D}|\phi)\Big] - D_{kl}\Big[q_\psi(\phi|\mathcal{D}) \| p(\phi)\Big]. \tag{5}$$

The corresponding trajectory-wise objective can be reinterpreted as follows:

$$\begin{aligned} \max_\psi \mathbb{E}_\tau \Big[&\mathbb{E}_{q_\psi, (\tau, \tau') \in \mathcal{D}}[\log \phi(\tau)] - \mathbb{E}_{q_\psi, (\tau', \tau) \in \mathcal{D}}[\log \phi(\tau)] \\ &- D_{kl}[q_\psi(\phi|\tau) \| p(\phi)] \Big], \end{aligned} \tag{6}$$

where 1) $q_\psi(\phi|\tau) = \text{Beta}(\alpha_\tau, \beta_\tau)$, where $[\alpha_\tau, \beta_\tau] = f_\psi^{\text{Beta}}(\tau)$ and f_ψ^{Beta} denotes a neural network parameterized by ψ, 2) $p(\phi) = \text{Beta}(\alpha_0, \beta_0)$ where α_0, β_0 defines our initial belief (hyper-parameters), and 3) $\phi(\tau)$ denotes the Bernoulli probability that τ ranks higher than τ'. Since both the posterior distribution $q_\psi(\phi|\tau)$ and the prior distribution $p(\phi)$ are beta-distributed, we represent the KL divergence term by following the Dirichlet VAE (Joo et al., 2020):

$$\begin{aligned} D_{kl}[q_\psi(\phi|\tau) \| p(\phi)] = &\log\left(\frac{\Gamma(\alpha_\tau + \beta_\tau)}{\Gamma(\alpha_0 + \beta_0)}\right) + \log\left(\frac{\Gamma(\alpha_0)\Gamma(\beta_0)}{\Gamma(\alpha_\tau)\Gamma(\beta_\tau)}\right) \\ &+ (\alpha_\tau - \alpha_0)\Big[\Psi(\alpha_\tau) - \Psi(\alpha_\tau + \beta_\tau)\Big] \\ &+ (\beta_\tau - \beta_0)\Big[\Psi(\beta_\tau) - \Psi(\alpha_\tau + \beta_\tau)\Big], \end{aligned} \tag{7}$$

where 1) $[\alpha_0, \beta_0]$ and $[\alpha_\tau, \beta_\tau]$ are parameters from the prior and the posterior functions, and 2) Γ and Ψ denote the gamma and the digamma functions.

4.3. Learning Generative Reward Model

In this work, we leverage a conditional generative model f_φ^r to represent the joint distribution of the step-wise rewards in a trajectory, i.e., $\hat{r}(\tau) \sim p(r|\tau) = f_\varphi^r(\tau)$, where $\hat{r}(\tau) = \sum_{t=0}^{T} \gamma^t \hat{r}_t$ denotes the trajectory reward. To enable efficient estimation, we derive an iterative update rule based on the MAP objective and estimated Beta prior. In specific, each time we sample rewards from the joint distribution, we update these rewards based on the following iterative update rule:

$$\hat{r}^{k+1}(\tau) = \log \frac{\alpha_\tau / [e^{\hat{r}^k(\tau)} + 1] + \sum_j \omega^{\tau,\tau^j} e^{\hat{r}^k(\tau^j)} / [e^{\hat{r}^k(\tau)} + e^{\hat{r}^k(\tau^j)}]}{\beta_\tau / [e^{\hat{r}^k(\tau)} + 1] + \sum_i \omega^{\tau^i,\tau} / [e^{\hat{r}^k(\tau)} + e^{\hat{r}^k(\tau^i)}]}, \tag{8}$$

where ω^{τ,τ^j} and $\omega^{\tau^j,\tau}$ are calculated based on the dataset $\mathcal{D}$. The design of this iterative update rule is based on the following theorem:

Theorem 4.1. *Let the informative prior $p_0(\phi)$ be a beta distribution Beta(α, β) and $e^{r(\tau)}$ be the strength of a trajectory segment τ. Assuming the geometric mean strength to be 1, i.e., $\prod e^{r(\tau)} = 1$, the iteration of Equation (8) will converge to the maximum of its MAP objective (i.e., Equation (3) with $p_0(\phi)$ the Beta prior), from any starting point, whenever a maximum exists.*

The proof can be found in Appendix A. To train the generative reward model, we 1) sample rewards from this model, 2) refine these rewards based on the specified update rule (Equation (8)) under the guidance of human preference, and 3) update the reward model by fitting it to the updated rewards (see Algorithm 1).

Model Implementation. In this work, we implement f_φ^r by a *distributional reward transformer* parameterized by φ. For each trajectory τ, we sample the initial step-wise rewards $[r_0^0, \ldots, r_T^0]$ from this reward model $f_\varphi^r(\tau)$ and calculate $\hat{r}^0(\tau) = \sum_{t=0}^{T} \gamma^t \hat{r}_t^0$. By utilizing the reward update process (Equation (8)), we compute the updated segment rewards $\hat{r}^K(\tau)$ after K iterations, which are the actual MAP values due to Theorem 4.1. The corresponding loss function can be modeled as:

$$\min_\varphi \mathbb{E}_\mathcal{D} \left[\text{dist}(\hat{r}^K(\tau), \hat{r}^0(\tau)) \right] \text{ where } \hat{r}_t^0 \sim \mathcal{N}(\mu_t, \sigma_t), \tag{9}$$

where $\hat{r}_t^0$ is sampled from a Gaussian distribution parameterized by mean μ_t and variance σ_t^2. To derive tractable gradients, we apply the reparameterization trick to generate $\hat{r}_t^0 = \mu_t + \sigma_t \cdot \epsilon$ where ϵ denotes samples from standard Gaussian distribution. Both μ_t and σ_t are the predictions of a causal transformer such that:

$$[(\mu_t, \sigma_t)_{t=0}^T] = \text{CausalTransformer}(s_0, a_0, \ldots, s_T, a_T). \tag{10}$$

5. Risk-Sensitive Policy Optimization

In this section, we introduce the approach to learning risk-sensitive policy that aligns with the inherent uncertainty in human preferences. Specifically, we employ the distributional Bellman operator to model the distribution of discounted cumulative rewards from the offline dataset. Given the estimated value distribution, we carry out policy improvement by maximizing the CVaR.

5.1. Offline Distributional Policy Evaluation

To enable distributional policy evaluation, we incorporate the learned reward generator f_φ^r to the original MDP $\mathcal{M}_{/R}$ without knowing the ground-truth reward. The resulting running environment is denoted as $\mathcal{M}_{/R} \cup f_\varphi^r$. For brevity, we denote it as $\widehat{\mathcal{M}}$.

Given a policy π, our goal is to learn a distributional action-value function $Z_{\widehat{\mathcal{M}}}^\pi(s, a)$ to estimate the distribution of discounted cumulative reward $\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$ where the initial state-action pair (s_0, a_0) is based on an offline dataset $\mathcal{D}$. We represent the distribution of $Z_{\widehat{\mathcal{M}}}^\pi$ by a uniform mixture of supporting quantiles such that $Z_{\widehat{\mathcal{M}}}^\pi(s_t, a_t) = \mathbb{E}_{\xi \sim U(0,1)}[\delta_{\theta_\xi(s_t, a_t)}]$, where θ_ξ estimates the quantile at the quantile level ξ and δ_{θ_ξ} denotes a Dirac distribution at θ_ξ.

To implement offline update for the model parameters θ, we utilize the following Conservative Distribution Evaluation (CDE) objective (Ma et al., 2021):

$$\min_\theta \mathcal{L}_{TD}(\theta) + \lambda \mathbb{E}_{\xi \sim U[0,1]} \Big[\mathbb{E}_{s \sim \mathcal{D}} (\log \sum_a \exp \theta_\xi(s, a)) - \mathbb{E}_{(s,a) \sim \mathcal{D}} (\theta_\xi(s, a)) \Big], \tag{11}$$

$$\mathcal{L}_{TD}(\theta) = \mathbb{E}_\mathcal{D} \Big[\mathbb{E}_{(\xi, \xi') \sim U(0,1)} \left[\rho_\kappa^\xi \left(\hat{r}_t + \gamma \theta_{\xi'}(s_{t+1}, a_{t+1}) - \theta_\xi(s_t, a_t) \right) \right] \Big], \tag{12}$$

where 1) λ is the penalty weight, 2) ρ_κ^ξ is the ξ-Huber quantile regression loss at threshold κ (Huber, 1964), and 3) $(s_t, a_t, s_{t+1}, a_{t+1})$ is uniformly sampled from the trajectories in the dataset $\mathcal{D}$ while $\hat{r}_t$ is sampled from our reward generator f_φ^r.

5.2. Risk-averse Policy Improvement

To better handle the underlying uncertainty in human preference, we adopt risk-averse policy updates by maximizing the estimate of CVaR within the distribution of cumulative

rewards. However, (Lim & Malik, 2022) indicates that directly integrating CVaR-based policy improvement with distributional policy evaluation does not necessarily guarantee convergence to the optimal policy. To overcome this issue, we utilize the following distributional policy improvement objective for static CVaR (Lim & Malik, 2022):

$$\pi(a_{t+1}|s_{t+1}) = \operatorname*{argmax}_{a_{t+1}} \mathbb{E}_{\xi\sim U[0,1]}\left[-(q(s_{t+1})-\theta_\xi(s_{t+1},a_{t+1}))^+\right], \tag{13}$$

where $q(s_{t+1}) = (q(s_t) - r_t)/\gamma$ keeps track of the reward history with the initial value of $q^\alpha = \mathbb{E}_{a\sim\pi}\left[F^{-1}_{Z^\pi(s,a)}(\alpha)\right]$, where $F^{-1}_{Z^\pi(s,a)}$ denotes the inverse cumulative density function of distribution $Z^\pi(s,a)$. As is shown in (Bäuerle & Ott, 2011), π converges to the optimal static CVaR policy by iteratively calculating q and updating π under an MDP whose state space is augmented by q (i.e., $\tilde{s} = (s,q) \in \mathcal{S}\times\mathcal{R}$). Intuitively, q is a moving threshold keeping track of the accumulated rewards so far.

The complete D-RLHF algorithm is shown in Algorithm 1.

Algorithm 1 Distributional Reinforcement Learning from Human Feedback (D-RLHF)

Input: The preference dataset $\mathcal{D}$, reward learning epochs N, maximum iterations K.
Initialize the reward model f^r_φ, the action-value model $Z^\pi_{\widehat{\mathcal{M}}}(s,a)$ and the policy $\pi(a|s)$.
Build a buffer $\mathcal{B}_\tau$ that records all the trajectories in $\mathcal{D}$.
Update the informative Beta prior with Objective (6).
for $n = 1, 2, \cdots, N$ **do**
  for $\tau \in \mathcal{B}_\tau$ **do**
    Sample rewards $[\hat{r}_0, \ldots, \hat{r}_T] \sim f^r_\varphi(\tau)$ and calculate $\hat{r}(\tau) = \sum_{t=0}^T \gamma^t \hat{r}_t$.
    Estimate the beta prior $[\hat{\alpha}(\tau), \hat{\beta}(\tau)] = f^{\text{Beta}}_\psi(\tau)$.
    Calculate the updated $\hat{r}^K(\tau)$ with Objective (8) based on the preference dataset $\mathcal{D}$.
    Update the reward model f^r_φ with Objective (9).
  end for
end for
for $\tau \in \mathcal{B}_\tau$ **do**
  Sample step-wise rewards $[\hat{r}_0, \ldots, \hat{r}_T] \sim f^r_\varphi(\tau)$ for the trajectory τ.
  Update the distributional action-value function $Z^\pi_{\widehat{\mathcal{M}}}(s,a)$ with Objective (11).
  Update the policy model $\pi(a|s)$ with Objective (13).
end for

6. Empirical Evaluation

In the empirical study, we start by illustrating the learned reward model in a discrete Gridworld environment (Section 6.1). Next, we construct three Risky PointMaze environments and evaluate the effectiveness of the proposed D-RLHF algorithm with the trajectory visualization (Section 6.2). Lastly, to demonstrate the model performance in the complex environment, we study robot navigation tasks, including a Risky Swimmer and a high-dimensional Risky Ant environment for evaluation (Section 6.3).

Experiment Settings. Our experiments primarily utilize the public platform Uni-RLHF (Yuan et al., 2024), which is tailored for offline RLHF. Additionally, to accommodate the underlying risks during the preference learning processes, we introduce risky regions by incorporating noise into transitions within them. The risky regions induce larger uncertainties in environments and preferences. We create the offline dataset by uniformly sampling from the expert policies trained online and then generating preferences based on their true rewards and risky steps. Please check Appendix B.2 for more details.

By following (Ma et al., 2021), we evaluate each approach using 100 test episodes by reporting both the mean and $\text{CVaR}_{0.1}$ (i.e., the average over the worst 10 episodes) for studied metrics, including: 1) *episodic rewards*, which calculate the cumulative rewards within an episode, and 2) *episodic violations*, which aggregate the total number of time steps spent inside the risky region. Each experiment is repeated with four random seeds, and the results are presented with mean $\pm$ standard deviation (std).

6.1. Reward Visualization in Gridworld

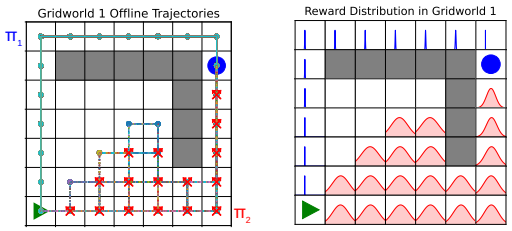

Figure 2. (Left) The Gridworld environment and offline trajectories. (Right) The learned risk-sensitive reward distributions represented by the probability density function. Please refer to Figure 6 in Appendix D.1 for the mean and standard deviation values of each distribution and the results in the remaining two settings.

In this experiment, we construct a Gridworld environment to better illustrate the case previously described in Figure 1. As shown in the left plot of Figure 2, the objective for the agent is to navigate from an initial position (green arrow) to a specified target (blue circle) while avoiding the walls (grey blocks). Within the bottom-right area (red markers), the environment demonstrates a degree of stochasticity, where, with specific probabilities (p = 0.1), it receives a random action instead of the agent's intended action (refer to Appendix C.1 for more details). Intuitively, the trajectories τ_2 generated by π_2 (passing through right-bottom) exhibit

higher rewards in expectation. To accommodate this situation, we assign greater preference to τ_2 by setting the expected chance of observing the preference that τ_2 ranks higher than τ_1 to be 0.6.

The right plot of Figure 2 illustrates the learned reward distributions at each state, where we utilize the blue and red colors to represent the rewards at risk-averse and risky regions, respectively. We find that the distributional reward model successfully captures the underlying uncertainty within the offline dataset in the sense that the rewards in risky regions exhibit a larger expectation but a higher variance. This leads to the result that the generated risk-averse policy avoids the risky area and navigates through the top-left map. Additionally, we also construct two distinct Gridworlds and illustrate the corresponding rewards. Please check Figure 6 in Appendix D.1 for complete results.

6.2. Model Performance in Risky PointMaze

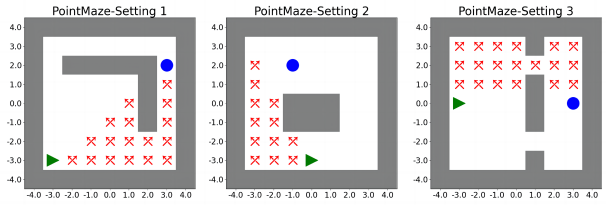

Figure 3. Three Risky PointMaze settings with green, blue, and red markers indicating the starting, target, and risky locations, respectively.

Task Description. In this experiment, we extend Gridworld to the continuous domain by constructing three PointMaze environments, as shown in Figure 3. In the risky regions denoted by red markers, the environmental transitions are influenced by additional Gaussian noise calculated such that $p_{\mathcal{T}}(s_{t+1}|s_t, a_t) = f(s_t, a_t) + \mathcal{N}(\mu_1, \sigma_1)$, where $f(\cdot)$ denotes the original transition function. Please check Appendix C.2 for more environmental details.

Comparison Methods. Besides our **D-RLHF** algorithm that learns a distributional reward model with a risk-averse policy, the following baselines are compared: 1) *regular RLHF* (**RLHF**) (Christiano et al., 2017) that learns a reward model through the Maximum Likelihood Estimation (MLE) objective, 2) *Ensemble RLHF* (**E-RLHF**) (Liang et al., 2022) that learns an ensemble of reward functions (we use five ensembles) and constructs the final reward as a combination of mean and standard deviation of these ensembles. Both of them utilize the Conservative Q-learning (CQL) (Kumar et al., 2020) for offline policy optimization.

Results Analysis. Table 1 shows the evaluation performance, with the best results in each setting (highest rewards or lowest violations) highlighted in bold. Please check Figure 7 in Appendix D.2 for evaluation results with the complete training phase. The results show that D-RLHF consistently outperforms other methods with higher $\text{CVaR}_{0.1}$ rewards and fewer violations in both mean and $\text{CVaR}_{0.1}$ metrics. This underscores the risk-averse policy in D-RLHF avoids passing through the regions with high uncertainty. When it comes to the mean rewards, D-RLHF still achieves compatible performance due to its superior $\text{CVaR}_{0.1}$ performance. We also find that RLHF and E-RLHF sometimes achieve higher mean rewards than D-RLHF. This is because the two methods do not acknowledge the risky regions and solely pursue expected cumulative rewards, resulting in traversing through risky areas.

Table 1. Evaluation results in three Risky PointMaze settings. Each value is reported as the mean $\pm$ standard deviation calculated from 100 episodes and 4 seeds.

	Method		RLHF	E-RLHF	D-RLHF (ours)
PointMaze Setting 1	Rewards	Mean	30.8 ± 27.4	$\mathbf{40.7 \pm 18.6}$	32.6 ± 33.5
		$\text{CVaR}_{0.1}$	-60.0 ± 0.0	-60.0 ± 0.0	$\mathbf{-16.6 \pm 58.9}$
	Violations	Mean	272.7 ± 20.2	265.8 ± 12.2	$\mathbf{86.1 \pm 86.4}$
		$\text{CVaR}_{0.1}$	450.7 ± 23.6	452.4 ± 46.0	$\mathbf{187.3 \pm 134.4}$
PointMaze Setting 2	Rewards	Mean	63.2 ± 5.0	64.0 ± 6.4	$\mathbf{65.4 \pm 11.2}$
		$\text{CVaR}_{0.1}$	42.3 ± 16.7	43.0 ± 17.3	$\mathbf{53.1 \pm 10.5}$
	Violations	Mean	121.2 ± 5.9	125.9 ± 15.3	$\mathbf{5.2 \pm 7.3}$
		$\text{CVaR}_{0.1}$	150.3 ± 23.4	163.9 ± 31.9	$\mathbf{50.8 \pm 71.7}$
PointMaze Setting 3	Rewards	Mean	64.2 ± 11.9	$\mathbf{68.6 \pm 7.5}$	63.8 ± 19.2
		$\text{CVaR}_{0.1}$	22.7 ± 51.7	41.8 ± 16.7	$\mathbf{48.9 \pm 55.1}$
	Violations	Mean	71.1 ± 71.7	107.1 ± 71.2	$\mathbf{27.7 \pm 33.1}$
		$\text{CVaR}_{0.1}$	167.5 ± 120.3	185.0 ± 61.5	$\mathbf{117.7 \pm 96.4}$

Results Visualization. Figure 4 illustrates 10 evaluation rollouts from RLHF and D-RLHF in the first setting of Risky PointMaze (check Figure 8 in Appendix D.2 for complete results). We find that D-RLHF drives a risk-averse policy that navigates to the longer but less stochastic path. By contrast, the traditional RLHF method struggles to perceive such uncertainties and tends to navigate through the risky region directly, where the noisy transition (i.e., aleatoric uncertainty) occasionally induces unsafe movements, leading to its poor $\text{CVaR}_{0.1}$ performance.

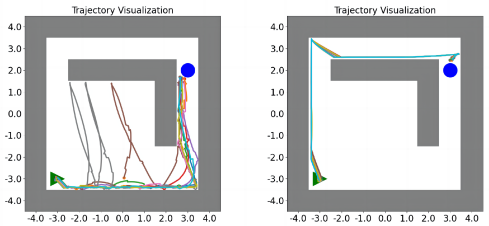

Figure 4. The trajectories generated by RLHF (left) and D-RLHF (right) during evaluation.

6.3. Model Performance in Risky Robot Navigation

Task Description. To better evaluate the performance of the proposed D-RLHF and baseline methods, we follow Ma et al. (2021) and construct more complicated robot naviga-

Table 2. Evaluation results in the Risky Ant and the Risky Swimmer environments. Each value is reported as the mean ± standard deviation calculated from 100 episodes and 4 random seeds.

	Method		CQL-HF	CODAC-HF	D-RLHF-Uniform	D-RLHF-Neutral	D-RLHF(ours)
Risky Ant	Reward	Mean	-1814.6 ± 96.4	**-1582.9 ± 75.4**	-2060.5 ± 100.3	-1768.3 ± 44.1	-1896.2 ± 69.7
		$\text{CVaR}_{0.1}$	-3092.0 ± 63.5	-2922.2 ± 99.0	-2577.2 ± 114.3	-2635.1 ± 104.3	**-2215.3 ± 98.6**
	Violation	Mean	58.4 ± 4.9	68.5 ± 6.5	44.4 ± 7.9	53.2 ± 6.6	**21.8 ± 3.9**
		$\text{CVaR}_{0.1}$	218.5 ± 34.9	289.0 ± 43.7	172.8 ± 11.7	209.5 ± 37.4	**124.5 ± 22.9**
Risky Swimmer	Reward	Mean	-2821.7 ± 265.3	-2698.4 ± 192.3	-2711.2 ± 276.9	**-2575.1 ± 188.0**	-2912.8 ± 183.6
		$\text{CVaR}_{0.1}$	-4512.8 ± 432.1	-4316.1 ± 310.4	-3856.2 ± 299.8	-4070.4 ± 246.5	**-3498.2 ± 230.9**
	Violation	Mean	332.9 ± 22.6	316.7 ± 31.9	252.4 ± 19.8	230.8 ± 29.7	**113.6 ± 11.5**
		$\text{CVaR}_{0.1}$	563.4 ± 47.0	512.3 ± 56.1	426.3 ± 32.0	381.9 ± 44.1	**175.5 ± 17.8**

tion tasks, including a high-dimensional Risky Ant environment and a Risky Swimmer environment.

For example, under the Ant environment shown in Figure 5 (Appendix C.3 covers more details), the goal is to travel from a starting point to a destination (green ball), where there exists a risky region (red plane) in the middle of the route. The environmental transition within the red circle is subject to a Gaussian noise $\mathcal{N}(\mu_2, \sigma_2)$, which introduces the risk. While a risk-neutral agent might pass through the risky region regardless of the underlying risk, a risk-aware agent should avoid it.

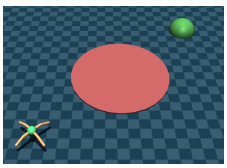

Figure 5. Risky Ant.

Comparison Methods. We use the following baselines alongside the proposed D-RLHF for comparison: 1) *Conservative Q-Learning with Human Feedback* (**CQL-HF**) (Kumar et al., 2020) and 2) *Conservative Offline Distributional Actor Critic with Human Feedback* **(CODAC-HF)** (Ma et al., 2021) that learns a non-distributional conservative Q-function and a conservative return distribution. Both of them utilize the typical MLE reward model. In addition, we perform ablation studies where 3) **D-RLHF-Uniform** replaces the informative Beta prior with a uniform one (i.e., $\alpha = \beta = 1$), and 4) **D-RLHF-Neutral** replaces the CVaR objective with a risk-neutral one (i.e., expectation).

Results Analysis. The empirical results in the Risky Ant and Risky Swimmer environments are shown in Table 2. We find that all the methods will inevitably encounter the risky region due to the intention of reward maximization. However, compared to other methods, D-RLHF exhibits better performance with higher $\text{CVaR}_{0.1}$ rewards and fewer violations (both mean and $\text{CVaR}_{0.1}$), which demonstrates the performance of its risk-averse policy. Note that although CODAC-HF with the distributional critic obtains the highest mean rewards in Risky Ant, it struggles to optimize the worst-case (i.e., $\text{CVaR}_{0.1}$) rewards and commits the highest number of violations. Interestingly, we find that the ablation methods D-RLHF-Uniform and D-RLHF-Neutral exhibit relatively better performance than CQL-HF and CODAC-HF in terms of the violations, which indicates the effectiveness of the distributional reward model and the risk-averse policy optimization.

7. Limitation

Offline Setting. This paper mainly focuses on the offline RLHF setting, where the agent can not interact with the environment and update human preferences. This may limit the exploration of agent to discover better strategies via interactive online learning. However, the proposed method can also be generalized to the online RLHF setting to learn a risk-aware policy from diverse human preferences.

Comparison with Direct Preference Optimization. Our study primarily adheres to the traditional RLHF framework, which involves initially learning a reward model from human preferences, followed by policy optimization. Under this consideration, we do not compare with the Direct Preference Optimization (DPO) (Rafailov et al., 2023) methods which directly optimizes the policy with human preferences without explicit reward modeling. This limitation can be potentially resolved by our future research.

8. Conclusion

In this paper, we introduce an uncertainty-aware preference alignment framework for Reinforcement Learning from Human Feedback (RLHF). We propose a Maximum A Posteriori (MAP) objective for learning a distributional reward model with an informative Beta prior and then utilize the distributional Bellman operator with the Conditional Value-at-Risk (CVaR) metric to develop a risk-sensitive policy, which is aware of the inherent uncertainty in human preferences. Empirical results demonstrate the effectiveness of the risk-sensitive ability of our approach. Future directions involve incorporating uncertainty awareness into direct preference optimization (DPO)-based methods with diverse human preferences.

References

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K., and Song, H. O. Direct preference-based policy optimization without reward modeling. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., Showk, S. E., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022.

Bäuerle, N. and Ott, J. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, 2011.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 70, pp. 449–458, 2017.

Bellemare, M. G., Dabney, W., and Rowland, M. *Distributional Reinforcement Learning*. MIT Press, 2023.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., de Oliveira Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019.

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.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., Wang, T., Marks, S., Ségerie, C., Carroll, M., Peng, A., Christoffersen, P. J. K., Damani, M., Slocum, S., Anwar, U., Siththaranjan, A., Nadeau, M., Michaud, E. J., Pfau, J., Krasheninnikov, D., Chen, X., Langosco, L., Hase, P., Biyik, E., Dragan, A. D., Krueger, D., Sadigh, D., and Hadfield-Menell, D. Open problems and fundamental limitations of reinforcement learning from human feedback. *CoRR*, abs/2307.15217, 2023.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 4299–4307, 2017.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. In *International Conference on Machine Learning, ICML*, volume 80, pp. 1104–1113, 2018a.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence*, pp. 2892–2901, 2018b.

Dong, Z., Yuan, Y., HAO, J., Ni, F., Mu, Y., ZHENG, Y., Hu, Y., Lv, T., Fan, C., and Hu, Z. Aligndiff: Aligning diverse human preferences via behavior-customisable diffusion model. In *International Conference on Learning Representations, ICLR*, 2024.

Duan, J., Guan, Y., Li, S. E., Ren, Y., Sun, Q., and Cheng, B. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE transactions on neural networks and learning systems*, 33:6584–6598, 2021.

Fang, B., Jia, S., Guo, D., Xu, M., Wen, S., and Sun, F. Survey of imitation learning for robotic manipulation. *Int. J. Intell. Robotics Appl.*, 3(4):362–369, 2019.

Hu, J., Sun, Y., Chen, H., Huang, S., Piao, H., Chang, Y., and Sun, L. Distributional reward estimation for effective multi-agent deep reinforcement learning. In *Advances in Neural Information Processing Systems, NeurIPS*, 2022.

Huber, P. J. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

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, NeurIPS*, pp. 8022–8034, 2018.

Joo, W., Lee, W., Park, S., and Moon, I. Dirichlet variational autoencoder. *Pattern Recognition*, 107:107514, 2020.

Keramati, R., Dann, C., Tamkin, A., and Brunskill, E. Being optimistic to be conservative: Quickly learning a cvar policy. In *AAAI Conference on Artificial Intelligence*, pp. 4436–4443, 2020.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., and Lee, K. Preference transformer: Modeling human preferences using transformers for RL. In *International Conference on Learning Representations, ICLR*, 2023.

Knox, W. B. and Stone, P. Tamer: Training an agent manually via evaluative reinforcement. In *IEEE International Conference on Development and Learning*, pp. 292–297, 2008.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems, NeurIPS*, 33:1179–1191, 2020.

Lee, K., Smith, L. M., and Abbeel, P. PEBBLE: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning, ICML*, volume 139, pp. 6152–6163, 2021.

Liang, X., Shu, K., Lee, K., and Abbeel, P. Reward uncertainty for exploration in preference-based reinforcement learning. In *International Conference on Learning Representations, ICLR*, 2022.

Lim, S. H. and Malik, I. Distributional reinforcement learning for risk-sensitive policies. In *Advances in Neural Information Processing Systems, NeurIPS*, 2022.

Liu, H., Sferrazza, C., and Abbeel, P. Chain of hindsight aligns language models with feedback. In *International Conference on Learning Representations, ICLR*, 2024.

Luo, Y., Liu, G., Duan, H., Schulte, O., and Poupart, P. Distributional reinforcement learning with monotonic splines. In *International Conference on Learning Representations, ICLR*, 2022.

Ma, Y. J., Jayaraman, D., and Bastani, O. Conservative offline distributional reinforcement learning. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 19235–19247, 2021.

MacGlashan, J., Ho, M. K., Loftin, R. T., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E., and Littman, M. L. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning, ICML*, volume 70, pp. 2285–2294, 2017.

Mavrin, B., Yao, H., Kong, L., Wu, K., and Yu, Y. Distributional reinforcement learning for efficient exploration. In *International Conference on Machine Learning (ICML)*, volume 97, pp. 4424–4434, 2019.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021.

Newman, M. E. J. Efficient computation of rankings from pairwise comparisons. *Journal of Machine Learning Research*, 24:238:1–238:25, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference, NeurIPS*, 2023.

Rockafellar, R. T., Uryasev, S., et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Rowland, M., Tang, Y., Lyle, C., Munos, R., Bellemare, M. G., and Dabney, W. The statistical benefits of quantile temporal-difference learning for value estimation. In *International Conference on Machine Learning, ICML*, volume 202, pp. 29210–29231, 2023.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. doi: 10.1038/NATURE16961. URL https://doi.org/10.1038/nature16961.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T. P., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359, 2017.

Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. Preference ranking optimization for human alignment. In *AAAI Conference on Artificial Intelligence*, pp. 18990–18998. AAAI Press, 2024.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Sui, Y., Huang, Y., Zhu, H., and Zhou, F. Adversarial learning of distributional reinforcement learning. In *International Conference on Machine Learning, ICML*, volume 202, pp. 32783–32796, 2023.

Sun, W., Lee, C., and Lee, C. DFAC framework: Factorizing the value function via quantile mixture for multi-agent distributional q-learning. In *International Conference on Machine Learning, ICML*, volume 139, pp. 9945–9954, 2021.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Swamy, G., Dann, C., Kidambi, R., Wu, Z. S., and Agarwal, A. A minimaximalist approach to reinforcement learning from human feedback. *CoRR*, abs/2401.04056, 2024.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, Ç., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, 2019.

Warnell, G., Waytowich, N. R., Lawhern, V., and Stone, P. Deep TAMER: interactive agent shaping in high-dimensional state spaces. In *AAAI Conference on Artificial Intelligence*, pp. 1545–1554. AAAI Press, 2018.

Wu, R., Uehara, M., and Sun, W. Distributional offline policy evaluation with predictive error guarantees. In *International Conference on Machine Learning, ICML*, volume 202, pp. 37685–37712, 2023a.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023b.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., and Hu, X. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 2023.

Yuan, H., Yuan, Z., Tan, C., Wang, W., Huang, S., and Huang, F. RRHF: rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Yuan, Y., Hao, J., Ma, Y., Dong, Z., Liang, H., Liu, J., Feng, Z., Zhao, K., and Zheng, Y. Uni-rlhf: Universal platform and benchmark suite for reinforcement learning with diverse human feedback. In *International Conference on Learning Representations, ICLR*, 2024.

Zhan, W., Uehara, M., Kallus, N., Lee, J. D., and Sun, W. Provable offline preference-based reinforcement learning. In *International Conference on Learning Representations, ICLR*, 2024a.

Zhan, W., Uehara, M., Sun, W., and Lee, J. D. Provable reward-agnostic preference-based reinforcement learning. In *International Conference on Learning Representations, ICLR*, 2024b.

Zhang, P., Chen, X., Zhao, L., Xiong, W., Qin, T., and Liu, T. Distributional reinforcement learning for multi-dimensional reward functions. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 1519–1529, 2021.

Zhang, S. and Yao, H. QUOTA: the quantile option architecture for reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pp. 5797–5804, 2019.

Zhou, F., Wang, J., and Feng, X. Non-crossing quantile regression for distributional reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Zhou, F., Zhu, Z., Kuang, Q., and Zhang, L. Non-decreasing quantile function network with efficient exploration for distributional reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3455–3461, 2021.

Zhu, B., Jordan, M. I., and Jiao, J. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning, ICML*, volume 202, pp. 43037–43067, 2023.

Appendix

A. Proof of Theorem 4.1

We prove the theorem in two steps.

The first step. We prove that the MAP objective of Equation (3) with $p_0(\phi) = \text{Beta}(\alpha, \beta)$ reaches its maximum when:

$$\hat{r}^k(\tau) = \log \frac{\alpha_\tau / [e^{\hat{r}^k(\tau)} + 1] + \sum_j \omega^{\tau, \tau^j} e^{\hat{r}^k(\tau^j)} / [e^{\hat{r}^k(\tau)} + e^{\hat{r}^k(\tau^j)}]}{\beta_\tau / [e^{\hat{r}^k(\tau)} + 1] + \sum_i \omega^{\tau^i, \tau} / [e^{\hat{r}^k(\tau)} + e^{\hat{r}^k(\tau^i)}]}. \tag{14}$$

Proof. To incorporate the informative Beta prior $p_0(\phi) = \text{Beta}(\alpha, \beta)$ into the iterative update objective, we start by rewriting the prior on the rewards as follows:

$$p(r(\tau)) = p_0(\phi) \frac{e^{r(\tau)}}{(e^{r(\tau)} + 1)^2} \tag{15}$$

$$= \frac{\phi^{\alpha_\tau - 1}(1 - \phi)^{\beta_\tau - 1}}{B(\alpha_\tau, \beta_\tau)} \cdot \frac{e^{r(\tau)}}{(e^{r(\tau)} + 1)^2} \tag{16}$$

$$= \frac{\left(\frac{e^{r(\tau)}}{e^{r(\tau)}+1}\right)^{\alpha_\tau - 1} \left(\frac{1}{e^{r(\tau)}+1}\right)^{\beta_\tau - 1}}{B(\alpha_\tau, \beta_\tau)} \cdot \frac{e^{r(\tau)}}{(e^{r(\tau)} + 1)^2}. \tag{17}$$

where $\alpha_\tau > 0$, $\beta_\tau > 0$ are the prior parameters for trajectory τ, and $B(\alpha_\tau, \beta_\tau) = \int_0^1 t^{\alpha_\tau - 1}(1 - t)^{\beta_\tau - 1} dt$ is the Beta function serving as a normalization constant. Substitute the above prior into Equation (3), we get:

$$\begin{aligned} &p(r(\tau)|\mathcal{D}) \propto p(\mathcal{D}|r(\tau))p(r(\tau)) \\ &= \prod_{ij} \left[\frac{e^{[r(\tau^i)]}}{e^{[r(\tau^i)]} + e^{[r(\tau^j)]}} \right]^{\omega^{\tau^i, \tau^j}} \prod_i \frac{\left(\frac{e^{[r(\tau^i)]}}{e^{[r(\tau^i)]}+1}\right)^{\alpha_{\tau^i} - 1} \left(\frac{1}{e^{[r(\tau^i)]}+1}\right)^{\beta_{\tau^i} - 1}}{B(\alpha_{\tau^i}, \beta_{\tau^i})} \frac{e^{[r(\tau^i)]}}{(e^{[r(\tau^i)]} + 1)^2}. \end{aligned} \tag{18}$$

For simplicity, we denote $e^{[r(\tau^i)]}$ as s_i, and α_{τ_i}, β_{τ_i} as α_i, β_i in the remaining part of the proof. The log-likelihood can be represented as:

$$\begin{aligned} &\sum_{ij} \omega^{\tau^i, \tau^j} \log s_i - \sum_{ij} \omega^{\tau^i, \tau^j} \log(s_i + s_j) + \sum_i (\alpha_i - 1) \log s_i - \sum_i (\alpha_i - 1) \log(s_i + 1) \\ &- \sum_i (\beta_i - 1) \log(s_i + 1) - N \log B(\alpha_i, \beta_i) + \sum_i \log s_i - \sum_i 2 \log(s_i + 1) \\ &= \sum_{ij} \omega^{\tau^i, \tau^j} \Big(\log(s_i) - \log(s_i + s_j) \Big) + \sum_i \Big(\alpha_i \log(s_i) - (\alpha_i + \beta_i) \log(s_i + 1) \Big) - N \log B(\alpha_i, \beta_i), \end{aligned} \tag{19}$$

where N is the number of i. Differentiating the above equation with respect to s_i for any i and setting the result to zero, we get:

$$\sum_j \frac{\omega^{\tau^i, \tau^j}}{s_i} - \sum_j \frac{\omega^{\tau^i, \tau^j} + \omega^{\tau^j, \tau^i}}{s_i + s_j} + \frac{\alpha_i}{s_i} - \frac{\alpha_i + \beta_i}{s_i + 1} = 0. \tag{20}$$

After rearranging the above equation, we obtain,

$$s_i = \frac{\alpha_i / (s_i + 1) + \sum_j \omega^{\tau^i, \tau^j} s_j / (s_i + s_j)}{\beta_i / (s_i + 1) + \sum_j \omega^{\tau^j, \tau^i} / (s_i + s_j)}. \tag{21}$$

□

The second step. We prove that iteration of Equation (8) will converge to its global maximum (Equation 14), from any starting point, whenever a maximum exists.

Proof. For simplicity, we rewrite the iteration objective (Equation (8)) as follows:

$$s_i' = \frac{\alpha_i/(s_i+1) + \sum_j \omega^{\tau^i,\tau^j} s_j/(s_i+s_j)}{\beta_i/(s_i+1) + \sum_j \omega^{\tau^j,\tau^i}/(s_i+s_j)}. \tag{22}$$

Consider an asynchronous update scheme. It is worth noting that: 1) any fixed point of this iteration corresponds to a stationary point of the posterior probability; 2) the iteration only produces non-negative values of s_i (given non-negative initial values); 3) the MAP objective (3) is bounded. Therefore, if the posterior probability under s_i' increases after applying the iteration, the converged fixed point is indeed the global maximum.

Let's examine the step where a specific s_i is updated. We define a function $f(s_i)$ as the sum of the current term in the log-likelihood of the posterior probability (i.e., Equation 19) that is dependent on s_i.

$$\begin{aligned} f(s_i) &= \sum_j \omega^{\tau^i,\tau^j} \log\left(\frac{s_i}{s_i+s_j}\right) - \sum_j \omega^{\tau^j,\tau^i}\log(s_i+s_j) + \alpha_i \log(s_i) - (\alpha_i+\beta_i)\log(s_i+1) - N\log B(\alpha_i,\beta_i) \\ &= \sum_j \omega^{\tau^i,\tau^j} \log\left(\frac{s_i}{s_i+s_j}\right) - \sum_j \omega^{\tau^j,\tau^i}\log(s_i+s_j) + \alpha_i \log\left(\frac{s_i}{s_i+1}\right) - \beta_i\log(s_i+1) - N\log B(\alpha_i,\beta_i). \end{aligned}$$

Suppose we update s_i into s_i' using Equation (22), we have

$$\begin{aligned} f(s_i') &= \sum_j \omega^{\tau^i,\tau^j} \log\left(\frac{s_i'}{s_i'+s_j}\right) - \sum_j \omega^{\tau^j,\tau^i}\log(s_i'+s_j) + \alpha_i \log\left(\frac{s_i'}{s_i'+1}\right) - \beta_i\log(s_i'+1) - N\log B(\alpha_i,\beta_i) \\ &\overset{(a)}{\geq} \sum_j \omega^{\tau^i,\tau^j} \log\left(\frac{s_i}{s_i+s_j}\right) + \frac{s_i'-s_i}{s_i'}\sum_j \omega^{\tau^i,\tau^j}\frac{s_j}{s_i+s_j} - \sum_j \omega^{\tau^j,\tau^i}\log(s_i+s_j) - (s_i'-s_i)\sum_j \frac{\omega^{\tau^j,\tau^i}}{s_i+s_j} \\ &\quad + \alpha_i \log\left(\frac{s_i}{s_i+1}\right) + \alpha_i \frac{s_i'-s_i}{s_i'(s_i+1)} - \beta_i\log(s_i+1) - \beta_i\frac{s_i'-s_i}{s_i+1} - N\log B(\alpha_i,\beta_i) \\ &\overset{(b)}{=} f(s_i) + (s_i'-s_i)\left[\frac{1}{s_i'}\sum_j \omega^{\tau^i,\tau^j}\frac{s_j}{s_i+s_j} - \sum_j \frac{\omega^{\tau^j,\tau^i}}{s_i+s_j} + \frac{\alpha_i}{s_i'(s_i+1)} - \frac{\beta_i}{s_i+1}\right] \\ &= f(s_i). \end{aligned} \tag{23}$$

- (a) holds due to Equation (16) and (17) in (Newman, 2023) (treat $\pi_i = s_i$, $\pi_i' = s_i'$ and $\pi_j = s_j$), along with two inequalities that $\log(x/(x+1)) \geq \log(y/(y+1)) + (x-y)/(x(y+1))$ and $-\log(x+1) \geq -\log(y+1) - (x-y)/(y+1)$.
- (b) holds due to the iteration given by Equation (22).

Consequently, applying Equation (22) for updates increases $f(s_i)$ and also the posterior probability until a fixed point is reached, where $s_i' = s_i$. Once the global maximum is attained for all s_i, the MAP objective (14) reaches its maximum value. This completes the proof. □

B. Implementation Details

B.1. Experimental Setting

In this paper, we utilized a total of 8 NVIDIA GeForce RTX 4090 GPUs, each equipped with 24 GB of memory. The random seeds in the continuous environments are 0, 123, 321, and 666. We trained the agents for 500 epochs in the offline setting and chose the final epoch for evaluation over 100 episodes. For fairness, we implement the reward model for each method the same, by a causal transformer like Kim et al. (2023). We also utilize a transformer-based architecture for the Beta model utilized in our method for learning informative priors.

B.2. Offline Preference Dataset

Based on Procedure 3.1, we create the offline preference dataset as follows: For Gridworld and PointMaze environments, we train two deterministic policies: one that navigates the shorter path through the risky region (risky), and the other that avoids the risky region by taking a longer route (risk-averse). Then we perform uniform sampling over them. As a result, the risky one's trajectories are more diverse because of the random noise in the risky region (as shown in the left column of Figure 6). To assess the risk-awareness of our methods, we encourage policies to embrace riskier actions by assigning higher preference to trajectories produced by risky policies. Specifically, we establish the expected likelihood of a risky trajectory outranking a risk-averse one to be 0.6. Following Yuan et al. (2024), we compare the trajectory segments with a length of 8 in Gridworld and a length of 100 in PointMaze.

For the risky robot navigation task (i.e., Risky Ant and Risky Swimmer), we train two Distributional Soft Actor Critic (DSAC) (Duan et al., 2021) agents online in each environment over 1000 episodes: one optimizing for expected returns and the other for CVaR returns. These agents are then employed to generate expert trajectories and we uniformly sample from them as the dataset for offline RL training. Consequently, trajectories produced by the former agent tend to be riskier, favoring shorter paths through risky regions for higher expected rewards, while trajectories from the latter aim to avoid risk due to CVaR optimization. Following this, we generate the preference labels as follows: for a pair of trajectories (τ_1, τ_2), if $|r(\tau_1) - r(\tau_2)| > t$, we prioritize the trajectory with higher rewards, otherwise, we select the trajectory with more steps in risky regions. Here t is a threshold and we set $t = 10$. The trajectory segment length is 100 in each environment.

B.3. Hyperparameters

As our approach primarily relies on the Conservative Offline Distributional Actor Critic method (Ma et al., 2021) for offline policy learning, we maintain the CODAC-specific hyperparameters consistent with the original study and only adjust the learning rate and Lagrange threshold. Regarding the reward model, we adhere to the architecture of the preference transformer model (Kim et al., 2023) as implemented in the Uni-RLHF benchmark (Yuan et al., 2024). Additionally, we employ the transformer architecture for the Beta model to learn sequential representations. We summarize the main hyperparameters in Table 3.

Table 3. List of the utilized hyperparameters in the proposed D-RLHF. To ensure equitable comparisons, we maintain consistency in the parameters of the same neural networks across different models.

Parameters	Risky PointMaze	Risky Ant	Risky Swimmer
General			
Max Episode Length	600	400	1000
Discount Factor	0.99	0.99	0.99
Training Epochs	50	500	500
Policy Model			
Actor Network	256, 256	256, 256	256, 256
Critic Network	256, 256	256, 256	256, 256
Actor Learning Rate	3e-6	3e-5	3e-5
Critic Learning Rate	3e-5	3e-5	3e-5
Min Q Weight	5	10	10
Lagrange Threshold	10	10	10
Number of Quantiles	32	32	32
Huber Regression Threshold	1	1	1
Entropy Tuning	True	True	True
Risk Level	0.1	0.1	0.1
Reward Model			
Network	256	256	256
Learning Rate	5e-5	5e-5	5e-5
Number of Attention Heads	4	4	4
Number of Layers	1	1	1
Batch Size	64	64	64
Beta Model			
Network	256	256	256
Learning Rate	3e-5	3e-5	3e-5
Number of Attention Heads	4	4	4
Number of Layers	1	1	1
Batch Size	64	64	64

C. Environmental Setting

C.1. Gridworld

The Gridworld environment consists of a map with several grids for movement. We create three unique scenarios, as shown in the left column of Figure 6. The agent's objective is to navigate from a starting location to a target location while avoiding the specified walls. At each step, the agent can choose from four possible actions, each corresponding to one of the four cardinal directions (up, down, left, right). Starting from the initial position, the agent receives a reward of 1 upon successfully reaching the target location, and a reward of 0 in all other cases. The game continues until a maximum of 50 time steps is reached. Additionally, we introduce risky regions to the environment where the transition exhibits a degree of uncertainty. Specifically, within the risky regions, with a predetermined probability ($p = 0.1$), the environment executes a random action instead of the intended action chosen by the agent.

C.2. Risky PointMaze

The PointMaze environment is a continuous domain that generalizes from the discrete Girdworld. In this scenario, the objective is to control a 2-degree-of-freedom (DoF) ball to reach a designated goal in a closed maze. As shown in Figure 3, we keep the same starting, target, wall, and risky locations as the previous Gridworld environment for the sake of evaluation in the continuous domain. The risky regions are characterized by adding Gaussian noise to the environmental transition functions, introducing stochasticity and risk into the agent's movements. Specifically, the transition in risky regions is $p_{\mathcal{T}}\left(s_{t+1}|s_t, a_t\right) = f\left(s_t, a_t\right) + \mathcal{N}(\mu_1, \sigma_1)$, where $f(\cdot)$ is the original transition function. We fix μ_1 = 0 and σ_1 = 0.05 across the environments. The maximum step is 600.

C.3. Risky Robot Navigation

Risky Swimmer In this environment, the agent controls a robot with two rotors connecting three segments, whose goal is to navigate from a starting state [1, 1] to a target state [5, 5] as quickly as possible. There is a risky region centered at [5, 5] with a radius of 1. The agent's dynamics remain consistent with the MuJoCo Swimmer environment. At each timestep, the agent's reward is calculated as the negative Euclidean distance to the goal plus 0.1 times its velocity. If the agent enters the risky regions, its transition will be influenced by a Gaussian noise $\mathcal{N}(0, 0.05)$. The episode terminates when the Euclidean distance between the agent and the target is less than 1 or reaches the maximum steps of 1000.

Risky Ant In this environment, the agent controls a high-dimensional ant robot with four legs, featuring 113 dimensions of observation. The goal is to navigate from the starting state [2, 2] to the target state [8, 8]. A risky region is centered at [5, 5] with a radius of 2. The agent's dynamics are identical to those of the MuJoCo Ant environment. At each timestep, the agent's reward is calculated as the negative Euclidean distance to the goal plus 0.1 times its velocity, encouraging rapid progress toward the target. If the agent enters the risky region, its transitions will be affected by Gaussian noise $\mathcal{N}(0, 0.05)$. The episode terminates when the Euclidean distance between the agent and the target is less than 1, or when the maximum of 400 steps is reached.

D. More Experimental Results

D.1. Gridworld

In Figure 6, the middle column of plots illustrates the learned risk-sensitive reward distributions by our method. It is evident that rewards in high-risk regions exhibit both a higher expectation and greater variance compared to those in risk-averse regions. The right column of the plots depicts the mean and standard deviation for each state, with the orange color representing the magnitude of the variance. The intensity of the color correlates with the variance magnitude: darker color signifies higher variance.

D.2. PointMaze

Figure 7 illustrates the evaluation results in three PointMaze environments over 100 episodes and 4 random seeds along the training procedure.

Figure 8 illustrates the trajectories generated by the traditional RLHF (top row) and the proposed D-RLHF (bottom row) in three PointMaze environments. We find that D-RLHF demonstrates a risk-averse strategy by selecting a longer path with lower variance.

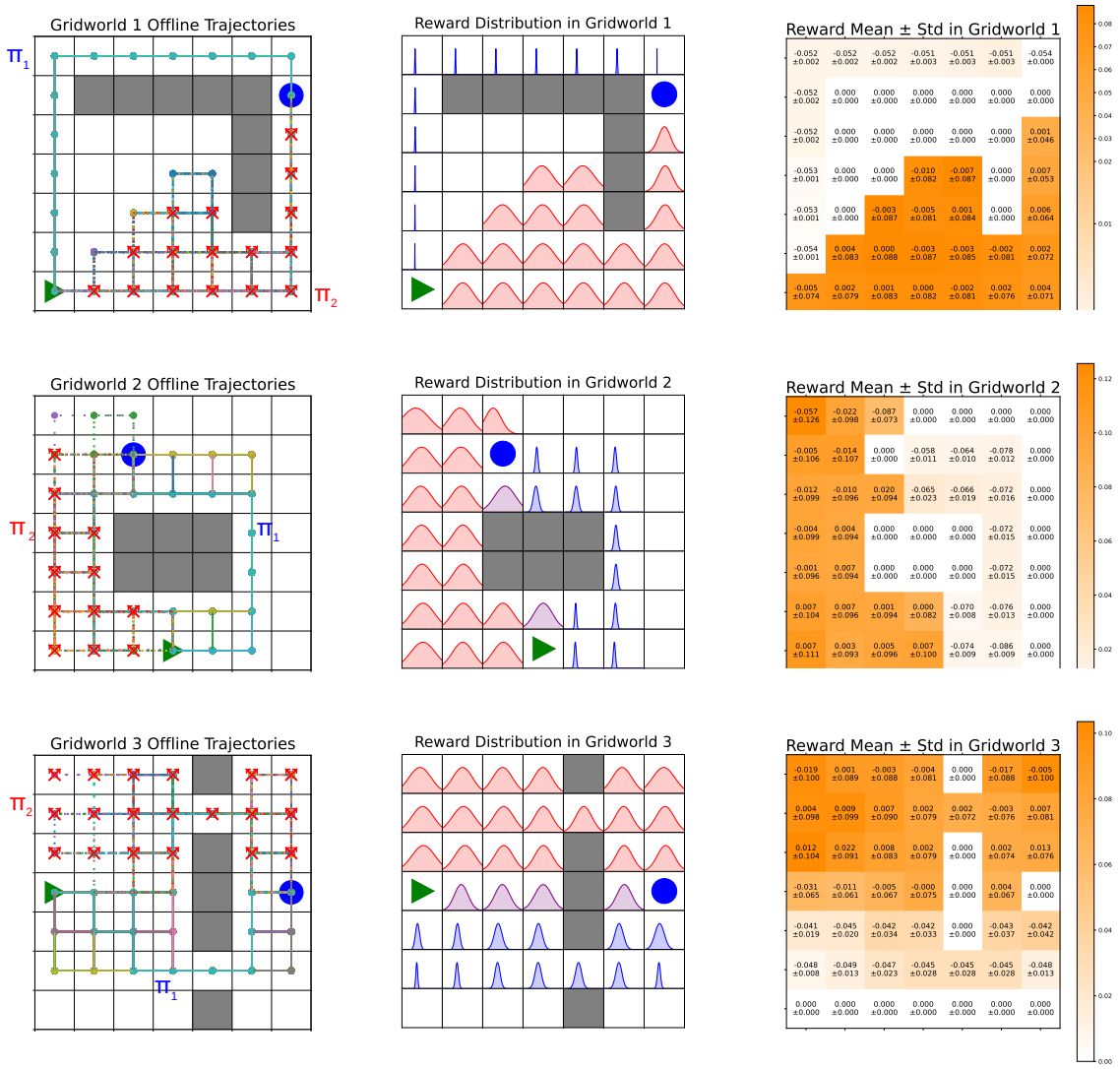

Figure 6. (Left) The Gridworld environment and offline trajectories, where solid trajectories are generated by risk-averse policy π_1 and dotted trajectories are generated by risky policy π_2. (Middle) The learned risk-sensitive reward distributions by our method. (Right) The mean and standard deviations of learned rewards. In terms of three distinct settings, Gridworld 1 is on the top, Gridworld 2 is in the middle, and Gridworld 3 is on the bottom.

Figure 7. The evaluation results along the whole training procedure in three PointMaze settings, where the top row denotes episode rewards and the bottom row denotes the episode violations.

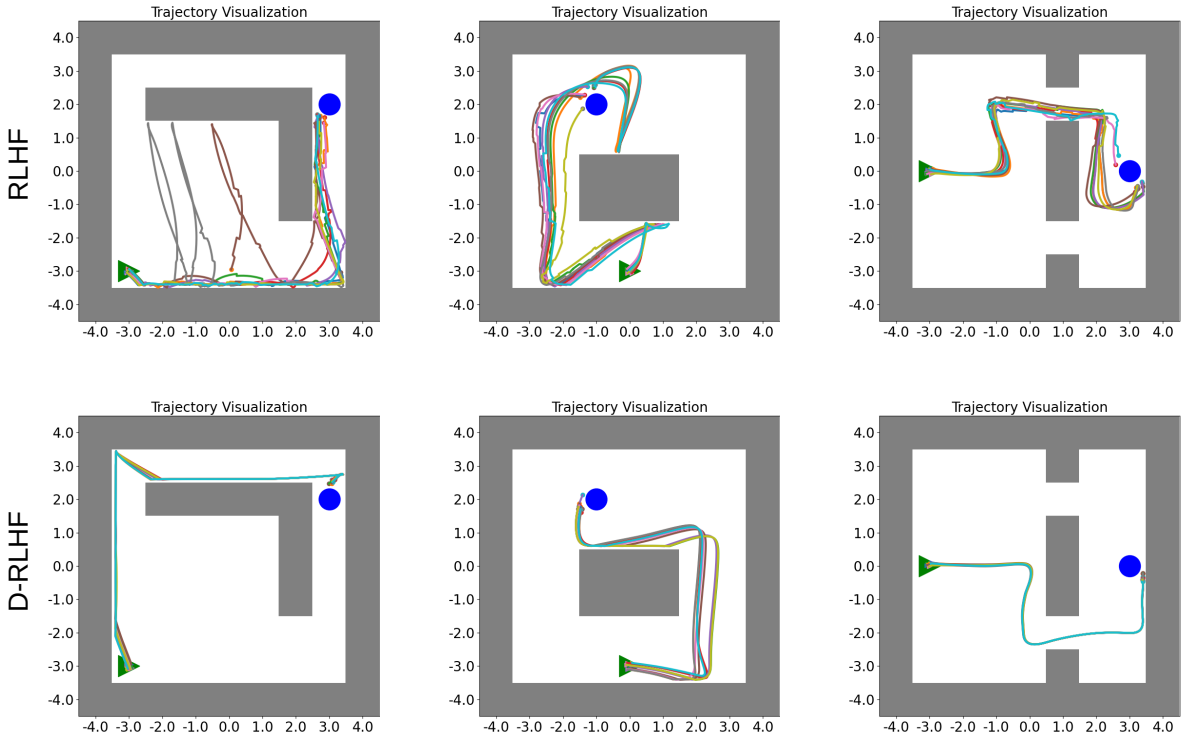

Figure 8. Each column refers to a PointMaze scenario. We illustrate the trajectories generated by traditional RLHF (top row) and D-RLHF (bottom row).

E. Broader Impacts

The development of a risk-aware preference alignment framework for Reinforcement Learning from Human Feedback (RLHF) holds significant potential for both positive and negative societal impacts. On the positive side, incorporating risk awareness into RLHF can enhance the safety and reliability of AI systems, leading to more ethical and trustworthy applications in critical areas such as healthcare, autonomous driving, and finance. This can result in improved decision-making processes that better align with human values and societal norms, ultimately fostering greater public trust in AI technologies. However, the negative impacts must also be considered. A risk-aware framework might inadvertently prioritize conservative approaches, potentially circumventing innovation and reducing the efficiency of AI systems. Additionally, if not properly designed, such systems could reinforce existing biases or inequalities by overly relying on feedback from specific groups, thereby marginalizing underrepresented voices. Balancing these risks and benefits is crucial to ensuring that the deployment of RLHF technologies promotes broad societal welfare while mitigating potential harm.