Human Feedback Attack on Online RLHF: Attack and Robust Defense

Chenye Yang, Mo Lyu, Guanlin Liu and Lifeng Lai

***Abstract*—In this paper, we investigate human feedback attacks on online Reinforcement Learning from Human Feedback (RLHF) algorithms. The attacker's goal is to force the victim RLHF algorithm to eventually learn a suboptimal policy while inducing a small attack cost. We propose an adversarial attack strategy, and prove that it is successful in terms of misleading the online RLHF algorithm to learn the suboptimal target policy. We also propose a robust defense online RLHF algorithm. We show that the proposed algorithm is robust to any attacker whose attack cost is bounded by a budget. The simulation results validate our theoretical analysis.**

***Index Terms*—reinforcement learning, human feedback, adversarial attack, robust defense**

I. Introduction

Reinforcement Learning from Human Feedback (RLHF) has become a prominent technique to adapt reinforcement learning models to problems of which the reward functions are not clear or difficult for human to demonstrate [1], [2]. In particular, RLHF is an important component of training foundational large models [1], e.g., the large language models (LLMs) [3]. RLHF allows foundational large models to align their output distribution with human preferences once fine-tuned, which improves usefulness and reduces harmful contents. Moreover, RLHF can continuously improve foundational large models through evaluation and data collection.

However, it has been shown that RLHF faces substantial security issues related to the reliability of the human preference data, i.e., when there are adversarial attacks on preferences or random noises in human preferences [2]. The human preference data can be manipulated by adversaries to mislead the RLHF algorithms [4]–[8], which can lead to catastrophic consequences in the real world. For example, when attacked by commercial competitors or individuals with an axe to grind, LLM chatbots could generate misinformation or harmful content, such as fake news, hate speech, or dangerous health advice.

Therefore, it is crucial to study the security issues faced by RLHF algorithms, especially to understand the impact and limitations of the adversarial attacks on RLHF, due to the potential catastrophic real-world consequences. More importantly, the existence of adversarial attacks should be considered in the design of RLHF algorithms, to ensure the robustness of RLHF algorithms against potential attackers, which could maintain the helpfulness and harmlessness of foundational large models.

C. Yang, M. Lyu, G. Liu and L. Lai are with the Department of Electrical and Computer Engineering, University of California, Davis, CA. This work was supported by National Science Foundation under grant CCF-2232907. Email:{cyyyang, molyu, glnliu, lflai}@ucdavis.edu.

In this paper, we first analyze the vulnerability of existing online RLHF algorithms. We propose an adversarial human feedback attack on online RLHF algorithms and prove its success through both theoretical analysis and experimental results. Our attack method manipulates the agent's learning process by providing specifically designed human preferences. The manipulations on human preferences are based on the Bradley-Terry model [9], and has the effect of hiding the true underlying reward function and misleading online RLHF algorithms to a wrong estimation of the reward function. In this way, the attacker can force the online RLHF algorithm to learn an unwanted target policy. Meanwhile, the attacker will induce a small attack cost because of the exploitation nature of the online RLHF algorithms, i.e., the victim algorithm wants to learn the optimal policy in its view eventually.

The success of our human feedback attack demonstrates the vulnerability of RLHF systems to adversarial interventions. To address this issue, we further design a robust online RLHF defense algorithm against attackers with bounded attack budget, ensuring that the agent can learn the original optimal policy effectively even under adversarial attacks. With the existence of the attacker, the defense mechanism encourages the online RLHF algorithm's exploration by raising the reward estimation, of which the amount is decided by the knowledge of attacker's ability to manipulate the human preferences. When there is no adversarial attack, the defense mechanism will not affect the learning process of the online RLHF algorithm, and the robust defense algorithm will behave the same as the standard online RLHF algorithm. We provide theoretical guarantees on the robustness of our defense method against attackers with bounded attack budget and validate its effectiveness with simulation results.

Our contributions are summarized as follows:

- We propose an adversarial human feedback attack on online RLHF algorithms, and prove its success through theoretical analysis and experimental results.
- We design a robust online RLHF algorithm defending against adversarial attacks, and provide theoretical guarantees on the robustness of our defense method, which is then validated by simulations.
- We provide a systematical understanding of the security issues faced by online RLHF algorithms from adversarial human feedback attacks, and valuable insights of designing robust defense algorithms resolving these issues.

The rest of the paper is organized as follows. In Section II, we review the background and related work. In Section III,

we introduce the problem formulation of online RLHF and the human feedback attack. In Section IV, we present the attack strategy design and theoretical analysis. In Section V, we propose the defense algorithm and provide theoretical guarantees. In Section VI, we conduct simulations to validate the attack and defense strategies. Finally, we conclude the paper in Section VII.

II. Background and Related Work

RLHF: RLHF frameworks usually combine three interconnected processes: feedback collection, reward modeling, and policy optimization [1], [2]. Take the example of training LLMs, the RLHF process usually starts with a language model pretrained on high-quality data, and then fine-tunes the model using human feedbacks to align the model's outputs with human preferences [3], [10], [11]. Depend on whether the human feedbacks are collected and whether the reward is modeled online or offline, the theoretical studies of RLHF algorithm can be categorized into online RLHF [12], [13] and offline RLHF [11], [14]. In the online RLHF setting, the agent learns from real-time interactions with human feedbacks, while in the offline RLHF setting, the agent has access to a pre-collected dataset of human preferences.

Adversarial Attack and Defense: Adversarial attacks have been widely studied in the context of machine learning [15], [16], from the traditional multi-armed bandit (MAB) problems [17] to the modern reinforcement learning (RL) algorithms [18]. For example, existing works have identified potential security issues of MAB algorithms in different settings [19]–[24], and different adversarial attacks methods against RL algorithms [25]–[29]. In particular, these works show that an attacker can force the machine learning algorithms to take unwanted actions, and may lead to severe real-world consequences (unfair business competition, health threats, misinformation spreading etc.).

Attack and defense always go hand in hand. With the existence of the attack, it is important to design defense strategies to protect the machine learning algorithms from taking unwanted actions [30]–[32]. For example, in the context of MAB problems, many robust algorithms have been proposed to defend against adversarial attacks in different settings [33]–[41]. Existing works have also proposed defense strategies in RL algorithms [42]–[44]. These works provide valuable insights on how to design defense algorithms against adversarial attacks.

Attack on RLHF and Robust Defense: Despite the rich literature on adversarial attacks and defenses in MAB and RL algorithms, designing adversarial attack methods and robust defense algorithms against adversarial attacks for RLHF is still an open research problem.

In the field of offline RLHF with a pre-collected dataset of human preferences, [4]–[7] propose data poisoning attack to modify the original preference dataset, or append poisoned preferences to the original dataset. [4]–[6] provide valuable experimental results on the impact of adversarial attacks, while [7] conduct a comprehensive theoretical analysis of the effectiveness of the attack. As for the robust algorithms in offline RLHF, [45]–[49] propose robust algorithms against the noise in human preferences. They assume that a small amount of the dataset is corrupted or noisy, and provide theoretical or experimental results on the robustness of offline RLHF algorithms. However, the noisy preferences in their works are not generated by adversarial attacks, i.e., not specifically designed to mislead the algorithm.

In the field of online RLHF learning from real-time interactions with human feedbacks, adversarial attacks and defenses have not been systematically studied. This paper aims to fill this gap by proposing an adversarial attack strategy on online RLHF algorithms, and designing a robust online RLHF algorithm against the adversarial attacks.

III. Problem Formulation

A. Reinforcement Learning with Human Feedback

We consider the episodic Markov Decision Process (MDP) with a state space $\mathcal{S}$, an action space $\mathcal{A}$, and a horizon H for each episode. The transition probability is denoted as $\mathbb{P}_h(s_{h+1}|s_h, a_h)$. A policy π is a mapping from the state space to the action space, i.e., $\pi : \mathcal{S} \to \mathcal{A}$, which specifies an action distribution based on the current state. We assume that there is an unobservable underlying reward function $r^* : (\mathcal{S} \times \mathcal{A}) \to [0, 1]$. The optimal policy is the one that maximizes the expected cumulative reward over the horizon $\mathbb{E}_\pi \left[\sum_{h=1}^H r^*(s_h, a_h)\right]$. One policy π is ϵ-optimal if its expected cumulative reward is within ϵ of the optimal policy: $\mathbb{E}_\pi \left[\sum_{h=1}^H r^*(s_h, a_h)\right] \geq \max_{\pi'} \mathbb{E}_{\pi'} \left[\sum_{h=1}^H r^*(s_h, a_h)\right] - \epsilon$.

A trajectory $\tau \in (\mathcal{S}, \mathcal{A})^H$ is defined as $(s_1, a_1, \ldots, s_H, a_H)$ which is a sequence of interactions with the MDP. In RLHF, the agent interacts with the reward-less environment and receives human preferences in the form of pairwise comparisons between trajectories. In this case, we assume that there is an unobservable underlying reward function r^* for trajectory: $r^* : (\mathcal{S} \times \mathcal{A})^H \to [0, H]$. Then, the policies can be evaluated with their values: $\mathbb{E}_{\tau\sim\pi}[r^*(\tau)]$. Note that the trajectory τ can be reduced to the state-action pair (s, a) by $H = 1$, which leads to (s, a) preferences as a special case of human comparison.

A popular model describing human preference distribution for pairwise comparisons is the Bradley-Terry model [9]: $P(\tau_1 \succ \tau_2) = \frac{\exp(r^*(\tau_1))}{\exp(r^*(\tau_1))+\exp(r^*(\tau_2))}$. Thus, the human comparison results are modeled as sampled from a Bernoulli distribution: $o(\tau_1, \tau_2) \sim \text{Ber}\left(\sigma\left(r^*(\tau_1) - r^*(\tau_2)\right)\right)$, where $\sigma(x)$ is the link function [13], e.g., the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ for Bradley-Terry model [1], [11], [50]–[52]. The outcome of the comparison is $o(\tau_1, \tau_2) = 1$ if τ_1 is preferred to τ_2, and $o(\tau_1, \tau_2) = 0$ otherwise.

Assumption III.1 (Link Function)**.** *$\sigma(0) = \frac{1}{2}$; for $x \in [-H, H]$, $\sigma'(x) \geq \alpha > 0$.*

This assumption is commonly used in RLHF literature [13], [53], and it makes the link function well-defined, makes the optimal policy be possible to identify [13], and ensures our

following analysis. It is also satisfied by the popular Bradley-Terry model.

We introduce the Eluder dimension to measure the difficulty of function approximation, and the realizability assumption of learning the underlying reward function r^*.

Definition III.1 (Eluder Dimension [13])**.** *For any function class $\mathcal{F} \subseteq \mathcal{X} \to \mathbb{R}$, its Eluder dimension $\dim_{\mathrm{E}}(\mathcal{F}, \epsilon)$ is defined as the length of the longest sequence $\{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{X}$ such that there exists $\epsilon' > \epsilon$ so that for all $i \in [n]$, x_i is ϵ'-independent of its prefix sequence $\{x_1, \ldots, x_{i-1}\}$, in the sense that there exists some $f_i, g_i \in \mathcal{F}$ such that*

$$\sqrt{\sum_{j=1}^{i-1} ((f_i - g_i)(x_j))^2} \leq \epsilon' \quad \textit{and} \quad |(f_i - g_i)(x_i)| \geq \epsilon'.$$

Eluder dimension is a measure of the complexity of a function class. It quantifies the longest sequence of points one must observe to distinguish between two functions in the class at any other point.

Assumption III.2 (Realizability [13])**.** *$r^* \in \mathcal{R}$, where $\mathcal{R}$ is the set of all possible reward functions $\mathcal{R} = \{r : (\mathcal{S} \times \mathcal{A})^H \to [0, H]\}$.*

Further, denote $\overline{\mathcal{R}} := \{r + c | c \in [-H, 0], r \in \mathcal{R}\}$ as the reward function class augmented with a bias term, due to the fact that human preference is based on the difference between two reward functions.

Since the agent does not have access to r^* in RLHF, it needs to estimate and approximate the underlying reward function. In the offline RLHF setting, one has access to a dataset of human comparison results $\mathcal{D} = \{(\tau_1^i, \tau_2^i, o_{1,2}^i)\}_{i=1}^N$. Maximum likelihood estimation [11], cross-entropy loss method [1], or Bayesian loss method [54] can be used to estimate the underlying reward function from the human comparison dataset. However, in the online RLHF setting, one will not have access to the entire dataset of comparisons. One popular approach to estimate the reward function in the online RLHF setting is the Preference-to-Reward (*P2R*) interface [13]. With the help of *P2R*, any reward-based RL algorithm with sample complexity guarantees (see Definition III.2) can be adapted to reward-less RLHF setting and learn an approximately optimal policy from preference feedback. *P2R* does not induce sample complexity overhead, and the human comparison complexity does not scale with the RL algorithm's sample complexity.

The *P2R* interface is presented in Algorithm 1. It maintains a confidence set of reward functions $\mathcal{B}_r$ and history sets of human comparisons $\mathcal{D}$ and $\mathcal{D}_{\text{hist}}$. When a new trajectory τ is queried, *P2R* first checks the history set $\mathcal{D}_{\text{hist}}$ to see if the reward for τ has been previously estimated. If not, it checks if the confidence set $\mathcal{B}_r$ agrees on the reward for τ. Only when neither of these conditions is satisfied, *P2R* queries the human comparison for the preference between τ and a reference trajectory τ_0. The interactions between the reward-based RL algorithm, the *P2R* interface, the MDP, and the human comparison oracle are shown in Figure 1.

Definition III.2 (Sample Complexity [13])**.** *An RL algorithm $\mathcal{A}$ is $g(\epsilon)$-robust and has sample complexity $C(\epsilon, \delta)$ if it can output an ϵ-optimal policy using $C(\epsilon, \delta)$ samples with probability at least $1 - \delta$, even if the reward of each trajectory τ is perturbed by $\epsilon(\tau)$ with $\|\epsilon(\tau)\|_\infty \leq g(\epsilon)$.*

Algorithm 1 Preference-to-Reward (*P2R*) Interface [13]

$\mathcal{B}_r \leftarrow \mathcal{R}$, $\mathcal{D} \leftarrow \{\}$, $\mathcal{D}_{\text{hist}} \leftarrow \{\}$
Execute the random policy to collect τ_0
Upon query of trajectory τ:
if $(\hat{r}, \tau) \in \mathcal{D}_{\text{hist}}$ **then**
    return $\hat{r}$
end if
if $\max_{r, r' \in \mathcal{B}_r} (r(\tau) - r(\tau_0)) - (r'(\tau) - r'(\tau_0)) < 2\epsilon$ **then**
    $\hat{r} \leftarrow r(\tau) - r(\tau_0)$ for an arbitrary $r \in \mathcal{B}_r$
    $\mathcal{D}_{\text{hist}} \leftarrow \mathcal{D}_{\text{hist}} \cup (\hat{r}, \tau)$
else
    Query comparison oracle m times on τ and τ_0; compute average comparison result $\bar{o}$
    $\hat{r} \leftarrow \arg\min_{x \in [-H, H]} |\sigma(x) - \bar{o}|$, $\mathcal{D} \leftarrow \mathcal{D} \cup (\hat{r}, \tau)$, $\mathcal{D}_{\text{hist}} \leftarrow \mathcal{D}_{\text{hist}} \cup (\hat{r}, \tau)$
    $\mathcal{B}_r \leftarrow \{r \in \mathcal{B}_r : \sum_{(\hat{r}, \tau) \in \mathcal{D}} (r(\tau) - r(\tau_0) - \hat{r})^2 \leq \beta\}$
end if
return $\hat{r}$

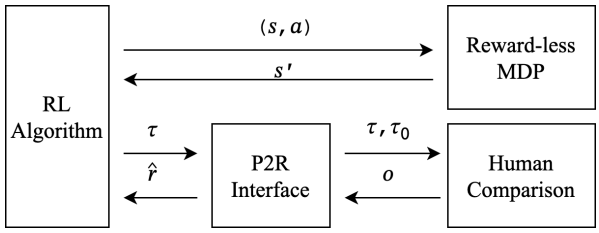


Fig. 1: Interactions protocol of the *P2R* interface [13].

B. Human Feedback Attack

In this paper, we consider a setup where there is an attacker that can intercept the human comparison results $o(\tau_1, \tau_2)$ and manipulate them to $\tilde{o}(\tau_1, \tau_2)$ to mislead the online RLHF algorithm. The attacker's goal is to force the online RLHF algorithm to learn a suboptimal target policy $\pi^\dagger$ while inducing a small attack cost. Specifically, we attack the *P2R* interface described in Section III-A, as shown in Figure 2.

One goal of the attacker is to force the online RLHF algorithm to learn a target policy $\pi^\dagger$. For example, it could be a suboptimal policy that is able to generate useless or even harmful content in LLMs. Here we consider a deterministic $\pi^\dagger$, i.e.,

$$\begin{cases} \pi^\dagger(a_h^\dagger | s_h) & = 1 \\ \pi^\dagger(a_h' | s_h) & = 0 \end{cases} \quad \forall h \in [H],$$

where $a_h^\dagger$ is the action taken by the target policy $\pi^\dagger$ for state s_h at time step h, and a_h' is other possible action.

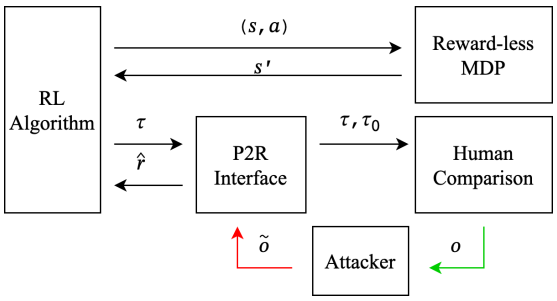


Fig. 2: Human feedback attack on online RLHF with *P2R*.

We assume that our target policy $\pi^\dagger$ is deterministic in the following analysis, which is reasonable because usually the attacker has one specific deterministic action to manipulate the agent to learn at each state [29], [55]–[57]. Note that stochastic target policy is also studied in [28], [58], which is not the focus of this paper.

We now define the total variation distance between two policies π and $\pi^{'}$ as $\mathrm{TV}\left(\pi \| \pi'\right) = \frac{1}{2}\sum_{\tau \in T}\left|P(\tau|\pi) - P(\tau|\pi')\right|$. The total variation distance [59] is used to determine if the learned policy $\hat{\pi}$ is close to the target policy $\pi^\dagger$. Usually, when talking about the policies, the total variation distance is defined with respect to the state-action pair (s,a): $\mathrm{TV}\left(\pi \| \pi'\right) = \max_s \sum_a |\pi(a|s) - \pi'(a|s)|$ [60], [61]. However, in our case, the total variation is defined with respect to the trajectory τ for the following reasons: 1) The human feedback is collected based on the comparison of two trajectories; 2) A different policy means a different $\pi(a|s)$ for each state-action pair (s,a), which will further lead to a different probability distribution of the trajectory τ; 3) Even two trajectories are identical, the probability of being sampled by π and π' could be different, leading to a different total variation distance.

Another goal of the attacker is to induce a small attack cost. For example, the attacker would want to reduce the cost of computational power of deciphering the queries to human and manipulating the comparison results in LLMs.

Definition III.3 (Global Attack Cost)**.**

$$\textit{Cost} = \sum_k \sum_m \mathbb{1}\left\{\tilde{o}_m(\tau_k, \tau_0) \neq o_m(\tau_k, \tau_0)\right\},$$

where k is the k-th trajectory, m represents the m-th comparison oracle query, $\tilde{o}_m(\tau_k, \tau_0)$ and $o_m(\tau_k, \tau_0)$ are the manipulated and original human comparison result respectively.

In the case of human feedback attack, we first propose the global attack cost, which is defined as the total number of attacker generating the manipulated human comparison results when it tries to mislead the online RLHF algorithm during entire learning process. It is a metric to evaluate the success of the attack, where a smaller attack cost means that the attacker could mislead the RLHF algorithm with less effort.

Definition III.4 (Local Attack Cost)**.**

$$\textit{Cost} = \sum_m \mathbb{1}\left\{\tilde{o}_m(\tau, \tau_0) \neq o_m(\tau, \tau_0)\right\},$$

where m represents the m-th comparison oracle query, $\tilde{o}_m(\tau, \tau_0)$ and $o_m(\tau, \tau_0)$ are the manipulated and original human comparison result respectively.

We further propose the local attack cost, which is refined to represent the number of attacker's manipulations on the comparison results for one trajectory τ sent for human comparison. Compared with the global attack cost, the local attack cost is more accurate to evaluate the attacker's effort in manipulating the human feedback.

IV. Attack

In this paper, the fundamental logic of attacking online RLHF algorithm is to force the standard RL algorithm to learn a target policy $\pi^\dagger$ by manipulating the reward observations in *P2R* interface and misleading the reward estimation. With the existence of the attack, the human comparison results $\tilde{o}$ are viewed as sampled w.r.t. a new underlying reward $r^\dagger$, instead of the r^*, and the estimation of underlying reward function of the environment will be misled. As a result, the agent will also be misled to learn a target policy $\pi^\dagger$ w.r.t. the manipulated reward function $r^\dagger$ instead of the original reward function r^*. In the case of Bradley-Terry model, our attack strategy aim to make the learning process as follows:

$$\hat{r}_{\text{att}} \leftarrow \tilde{o}\left(\tau_1, \tau_2\right) \sim \text{Ber}\left(\sigma\left(r^\dagger(\tau_1) - r^\dagger(\tau_2)\right)\right),$$

where $\tilde{o}\left(\tau_1, \tau_2\right)$ is the human comparison result between τ and τ_0 under attack, and $\hat{r}_{\text{att}}$ is the estimation of the underlying reward function under attack, i.e., $r^\dagger$.

Denote $\mathcal{R}^\dagger$ as the set of all possible reward functions after attack $\mathcal{R}^\dagger = \{r : (\mathcal{S}\times\mathcal{A})^H \to [0, H]\}$. Further, denote $\overline{\mathcal{R}}^\dagger := \{r + c | c \in [-H, 0], r \in \mathcal{R}^\dagger\}$ as the reward function class after attack augmented with a bias term. We assume $\mathcal{R}^\dagger \subseteq \mathcal{R}$ and $\overline{\mathcal{R}}^\dagger \subseteq \overline{\mathcal{R}}$. $r^\dagger \in \mathcal{R}^\dagger$. This assumption ensures that we can theoretically find an exact representation of the reward function after attack, and the attacker cannot create a reward function that is not in the original space.

We hope that the attack could have the following effects:

- The manipulated human feedback $\tilde{o}$ is seen to be sampled w.r.t. a new underlying reward $r^\dagger$, instead of the r^*.
- The best policy under the manipulated reward $r^\dagger$ is the target policy $\pi^\dagger$.
- The RL algorithm will actually learn the target policy $\pi^\dagger$.
- The attack will only induce a small cost.

Define T as the set of all possible trajectories, and $T^\dagger$ as the set of all trajectories that the agent will experience under the deterministic target policy $\pi^\dagger$:

$$T = \left\{\tau : \tau \in (\mathcal{S}\times\mathcal{A})^H\right\},$$
$$T^\dagger = \left\{\tau \in T : \tau\left(\pi^\dagger, \mathbb{P}_h(\cdot|s,a)\right)\right\}.$$

The new underlying reward function $r^\dagger$ after attack is defined as:

$$r^\dagger(\tau) = \begin{cases} 0 & \tau \in T \setminus T^\dagger \\ r^*(\tau) & \tau \in T^\dagger. \end{cases}$$

Thus, the attack strategy is to manipulate the human comparison results to be sampled w.r.t. the new underlying reward function $r^\dagger$. For the trajectories $\tau \in T^\dagger$, the manipulated human comparison results $\tilde{o}(\tau, \tau_0) \sim \text{Ber}(\sigma(r^*(\tau) - r^*(\tau_0)))$, which means that $\tilde{o}(\tau, \tau_0) = o(\tau, \tau_0)$. For the trajectories $\tau \in T \backslash T^\dagger$, the manipulated human comparison results $\tilde{o}(\tau, \tau_0)$ will be sampled from a new distribution $\text{Ber}(\sigma(-r^*(\tau_0)))$.

Note that when the reference trajectory $\tau_0 \in T \backslash T^\dagger$, we will notice that $\tilde{o}(\tau_0, \tau_0) \sim \text{Ber}(\sigma(-r^*(\tau_0)))$, which only holds when $r^*(\tau_0) = 0$ so that the Bradley-Terry model is satisfied. Thus, we have $r^*(\tau_0) = r^\dagger(\tau_0)$ no matter $\tau_0 \in T^\dagger$ or not. In summary, we have the following assumption on the problem sets on which this attack is effective, to ensure the human comparison model is not violated:

Assumption IV.1 (Baseline Trajectory)**.** *If $\tau_0 \in T \setminus T^\dagger$, then $r^*(\tau_0) = 0$. Otherwise, there is no restriction on $r^*(\tau_0)$.*

In short, our attack strategy is:

$$\begin{cases} \tilde{o}(\tau, \tau_0) = o(\tau, \tau_0) & \tau \in T^\dagger \\ \tilde{o}(\tau, \tau_0) \sim \text{Ber}(\sigma(-r^*(\tau_0))) & \tau \in T \setminus T^\dagger. \end{cases}$$

Here, we present our attack strategy in Algorithm 2.

Algorithm 2 Human Feedback Attack for online RLHF

Upon human comparison is needed for τ
if $\tau \in T^\dagger$ **then**
    $\tilde{o}(\tau, \tau_0) = o(\tau, \tau_0)$
else
    $\tilde{o}(\tau, \tau_0) \sim \text{Ber}(\sigma(-r^*(\tau_0)))$
end if
Return $\tilde{o}(\tau, \tau_0)$ for human comparison result

The following theorems provide the theoretical analysis of the attack strategy.

Theorem IV.1. *Under human feedback attack as Algorithm 2, the deterministic target policy $\pi^\dagger$ is the optimal policy w.r.t. the manipulated reward function $r^\dagger$:*

$$V_{r^\dagger}^{\pi^\dagger} = \max_\pi V_{r^\dagger}^{\pi} = \max_\pi \mathbb{E}_{\tau \sim \pi}\left[r^\dagger(\tau)\right].$$

Theorem IV.1 reveals that with the new underlying reward function $r^\dagger$, the target policy $\pi^\dagger$ is the optimal policy.

Theorem IV.2. *Suppose Assumption III.1 and III.2 hold. Let $\epsilon_0 = g(\epsilon)/2$, $d_{\overline{\mathcal{R}}} = \dim_{\text{E}}(\overline{\mathcal{R}}, \epsilon_0)$ and $m = \Theta\left(\frac{d_{\overline{\mathcal{R}}} \ln\left(d_{\overline{\mathcal{R}}}/\delta\right)}{\epsilon_0^2 \alpha^2}\right)$ in Algorithm 1. Suppose that $\mathscr{A}$ is a $g(\epsilon)$-robust RL algorithm with sample complexity $\mathcal{C}(\epsilon, \delta)$, and is running on the estimated reward function $\hat{r}$ returned by Algorithm 1. Under human feedback attack as Algorithm 2, with probability $1 - \delta$, $\mathscr{A}$ will return a policy $\hat{\pi}$ that is ϵ-optimal w.r.t. the manipulated reward function $r^\dagger$: $V_{r^\dagger}^{\hat{\pi}} \geq V_{r^\dagger}^{\pi^\dagger} - \epsilon$.*

Theorem IV.2 shows that the RL algorithm will return an ϵ-optimal policy. However, it is not guaranteed that the RL algorithm learns the target policy $\pi^\dagger$, meaning that the returned policy $\hat{\pi}$ is not necessarily close to $\pi^\dagger$ in distribution. Because ϵ-optimal is only a property from the perspective of value function, it doesn't describe the action distribution based on the current state.

Theorem IV.3. *The ϵ-optimal policy $\hat{\pi}$ in Theorem IV.2 is $\frac{\epsilon}{H}$-approximate to the target policy $\pi^\dagger$:*

$$TV\left(\hat{\pi} \| \pi^\dagger\right) \leq \frac{\epsilon}{H},$$

where ϵ is from ϵ-optimal in terms of value function, and H is the horizon of each trajectory.

Theorem IV.3 shows that the RL algorithm will return a policy that is close to the target policy $\pi^\dagger$ in distribution, which means that the attack is successful to manipulate the agent to learn the target policy $\pi^\dagger$.

Theorem IV.4. *With probability $1 - \delta$, during the forced learning process as of Theorem IV.2, when the $\frac{\epsilon}{H}$-approximate policy $\hat{\pi}$ is returned, the attacker induces global attack cost:*

$$\textit{Cost} \leq m \cdot \mathcal{C}(\epsilon, \delta).$$

Corollary IV.4.1. *With probability $1 - 2\delta$, during the forced learning process as of Theorem IV.2, when the $\frac{\epsilon}{H}$-approximate policy $\hat{\pi}$ is returned, the attacker induces global attack cost:*

$$\textit{Cost} \leq \tilde{\mathcal{O}}\left(\frac{d_{\overline{\mathcal{R}}}^2}{\epsilon^2 \alpha^2}\right).$$

Theorem IV.4 provides an upper-bound on the global attack cost, which is highly dependent on the RLHF algorithm design. In this case, it depends on how the reward is estimated and the robustness of the RL algorithm $\mathscr{A}$, and particularly, we have Corollary IV.4.1 that provides a more detailed upper-bound on the attack cost.

V. Defense

In practical applications, the attacker may have a limited budget on the attack cost. In this section, we consider one particular kind of attackers, of which the local attack cost (Definition III.4) is limited. For example, in LLMs, even when one answer is queried multiple times for training, the adversary may only manipulate a small portion of the comparison results due to the cost of deciphering the query and perform the manipulation. Denote C as the maximum local attack cost allowed by the attacker for each trajectory. We have that $\sum_m \mathbb{1}\{\tilde{o}_m(\tau, \tau_0) \neq o_m(\tau, \tau_0)\} \leq C$ for each trajectory τ. These attackers with bounded local attack cost could be defended by particularly designed defense algorithms. On the contrary, if an attacker has unbounded local attack cost, there's no algorithm can defend it, since the human feedbacks can be arbitrarily changed. We suppose that the defense algorithm has the knowledge of the upper-bound C of the attacker's local attack cost, but not the specific attack strategy.

Here, we propose a defense online RLHF algorithm, named Robust Preference-to-Reward (*R-P2R*) interface, which is robust to human feedback attacks. The *R-P2R* interface is presented in Algorithm 3. The interactions between the reward-

based RL algorithm, the *R-P2R* interface, the MDP, and the human comparison oracle are shown in Figure 3.

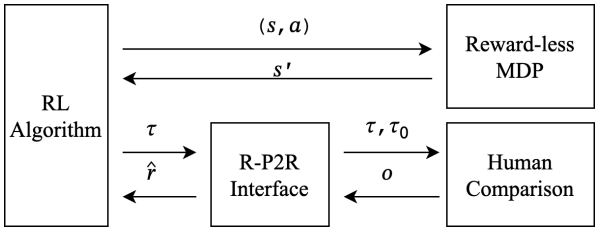


Fig. 3: Interactions protocol of the *R-P2R* interface.

In the *R-P2R* interface, when a trajectory τ is queried by the RL algorithm, *R-P2R* first checks the history set $\mathcal{D}_{\text{hist}}$ to see if the trajectory has been queried before. If the trajectory has been queried before, *R-P2R* will return the estimated reward $\hat{r}$ directly. Otherwise, *R-P2R* will check if the confidence set $\mathcal{B}_r$ agrees on the reward estimation, i.e., the difference between two reward functions on τ and τ_0 is less than 2ϵ. If $\mathcal{B}_r$ agrees on the reward estimation, *R-P2R* will return an arbitrary reward estimation $\hat{r}$ from $\mathcal{B}_r$. Otherwise, *R-P2R* will query the human comparison oracle m times on τ and τ_0 to get the average comparison result $\bar{o}$, and then estimate the reward function $\hat{r}$ based on $\bar{o}$. Meanwhile, *R-P2R* will update the confidence set $\mathcal{B}_r$ and history sets $\mathcal{D}$ and $\mathcal{D}_{\text{hist}}$.

The intuition of the defense algorithm is to encourage the RL algorithm's exploration by raising the reward estimation. In the *R-P2R* interface, we introduce a parameter γ, which raises the average comparison result $\bar{o}$ to 1 for an amount of $\gamma(1-\bar{o})$. When the $\bar{o}$ is small, which could probably be caused by the attacker's manipulation, *R-P2R* will raise $\bar{o}$ more, leading to larger reward estimation. Otherwise, *R-P2R* will raise $\bar{o}$ less. Note that, this will not change the relative order of comparison results for different trajectories. When the attacker has a small local attack cost budget C, which means that for each trajectory not too many comparisons are manipulated, *R-P2R* will be more conservative on the reward estimation, otherwise, more aggressive. For the special case when there's no attack, *R-P2R* will reduce to *P2R* by setting $C = 0$.

The following theorems provide the theoretical analysis of the defense strategy.

Theorem V.1. *Assume that the attacker has local attack cost budget C, i.e., $\sum_m \mathbb{1}\{\tilde{o}_m(\tau,\tau_0) \neq o_m(\tau,\tau_0)\} \leq C$ for each trajectory τ. Suppose Assumption III.1 and III.2 hold. Let $\epsilon_0 = g(\epsilon)/2$, $d_{\overline{\mathcal{R}}} = \dim_{\mathbb{E}}(\overline{\mathcal{R}}, \epsilon_0)$ and $m = \max\left\{\frac{256 d_{\overline{\mathcal{R}}} \ln(4d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0^2 \alpha^2}, \frac{32C\sqrt{d_{\overline{\mathcal{R}}}}}{\alpha\epsilon_0}\right\}$. Suppose that $\mathcal{A}$ is an $g(\epsilon)$-robust RL algorithm with sample complexity $C(\epsilon,\delta)$. By running $\mathcal{A}$ with R-P2R as in Algorithm 3, we can learn an ϵ-optimal policy w.r.t. the original underlying reward function, using $C(\epsilon,\delta)$ samples and $\max\left\{\frac{1024 d_{\overline{\mathcal{R}}}^2 \ln(4d_{\overline{\mathcal{R}}}/\delta)}{g^2(\epsilon)\alpha^2}, \frac{64C d_{\overline{\mathcal{R}}}^{3/2}}{\alpha g(\epsilon)}\right\}$ queries to the comparison oracle with probability $1-2\delta$.*

Theorem V.1 reveals that with a limited attack cost budget C on the attacker side, or in other words, a properly chosen and relatively large enough m on *R-P2R* side, the RL algorithm will return an ϵ-optimal policy with high probability.

Particularly, when the local attack cost budget is small, i.e., $C \leq \frac{8\sqrt{d_{\overline{\mathcal{R}}}}\ln(4d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0\alpha}$, the human comparisons needed for each trajectory will be $m = \frac{256 d_{\overline{\mathcal{R}}}\ln(4d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0^2\alpha^2} = \Theta\left(\frac{d_{\overline{\mathcal{R}}}\ln(d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0^2\alpha^2}\right)$. Moreover, an ϵ-optimal policy can be learned by $\mathcal{A}$ with sample complexity $C(\epsilon,\delta)$ and query complexity $\tilde{\mathcal{O}}\left(\frac{d_{\overline{\mathcal{R}}}^2}{\alpha^2 g(\epsilon)^2}\right)$.

Specifically, when there's no attack, i.e., $C = 0$, the *R-P2R* will reduce to the *P2R*, and Theorem V.1 will reduce to Theorem 4 in [13].

Algorithm 3 Robust Preference-to-Reward (*R-P2R*) Interface

```
B_r ← R, D and D_hist ← {}, local attack cost budget C
Execute the random policy to collect τ_0
Upon query of trajectory τ:
γ ← C/(C + m)
if (r̂, τ) ∈ D_hist then
    return r̂
end if
if max_{r,r'∈B_r}(r(τ) − r(τ_0)) − (r'(τ) − r'(τ_0)) < 2ε then
    r̂ ← r(τ) − r(τ_0) for an arbitrary r ∈ B_r
    D_hist ← D_hist ∪ (r̂, τ)
else
    r̂ ← Query-Human-Comparison(τ, τ_0, γ, B_r, D, D_hist)
end if
return r̂
```

Algorithm 4 Query-Human-Comparison (in *R-P2R*)

```
Passing τ, τ_0, γ, B_r, D and D_hist as input
Query comparison oracle m times on τ and τ_0; compute
    average comparison result ō
o' ← ō + γ(1 − ō)
r̂ ← arg min_{x∈[−H,H]} |σ(x) − o'|
D ← D ∪ (r̂, τ), D_hist ← D_hist ∪ (r̂, τ)
B_r ← {r ∈ B_r : Σ_{(r̂,τ)∈D}(r(τ) − r(τ_0) − r̂)^2 ≤ β}
return r̂
```

VI. Experimental data and results

A. Attack

We first evaluate the attack strategy in Algorithm 2 on the online RLHF algorithms. We consider a simple grid world environment with a 11×11 grid, where the agent starts from the center of the grid, i.e., $(5,5)$. The agent has five possible actions: *up*, *right*, *down*, *left*, and *stay*, with mean reward $0.7, 0.8, 0.2, 0.3, 0.5$ respectively. The environment dynamics are defined as: when the agent takes an action, with probability $p = 0.9$, it will move to the desired direction, and with probability $1 - p = 0.1$, it will move *up*, *right*, *down*, *left*, or *stay* with equal probability. The environment is illustrated in Figure 4. We use tuple (x, y, a) to represent the state-action pair, where x and y are the coordinates of the agent, and a is the action taken by the agent. The human comparison is based

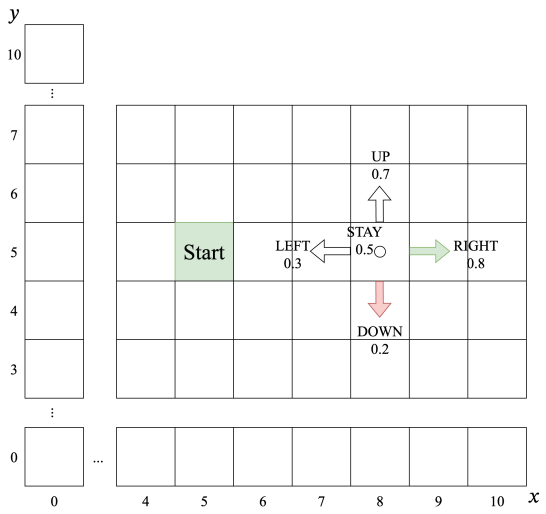


Fig. 4: Grid world environment

on the state-action pair to give (x, y, a) preference, i.e., the trajectory τ is one state-action pair (x, y, a) and has horizon 1. In this environment, the original optimal policy is to move *right* for each state (x, y). The baseline trajectory τ_0 as the reference for human comparison is $(5, 5, down)$.

The agent will interact with the environment for K episodes, starting from the center of the grid for each episode, and within one episode, the agent will take $H = 5$ steps. This setting belongs to tabular MDP of which the state and action space are finite. The Eluder dimension of the reward function class is $d_{\overline{\mathcal{R}}} = \tilde{\mathcal{O}}(SA)$.

We attack the popular online RLHF algorithm *P2R* interface in Algorithm 1 connected with *UCBVI-BF* [62] as the RL algorithm $\mathscr{A}$. *UCBVI-BF* is a model-based tabular value iteration algorithm which guarantees that the value function is an upper confidence bound on the optimal value function. *UCBVI-BF* uses Bernstein-Freedman's concentration inequality to build the confidence set. It has sample complexity $\mathcal{C}(\epsilon, \delta) = \mathcal{O}\left(SAH^3/\epsilon^2\right)$ [13], [63], and is $\mathcal{O}(\epsilon/H)$-robust [13]. Here we set the parameters $\epsilon = 0.1$, $\delta = 0.01$, $g(\epsilon) = \epsilon/4H = 5 \times 10^{-3}$, $\epsilon_0 = g(\epsilon)/2 = 2.5 \times 10^{-3}$, $d_{\overline{\mathcal{R}}} = 605$, and $\beta = \epsilon_0^2/4 = 1.5625 \times 10^{-6}$. The constant α is calculated to be $\alpha = 0.196$ for sigmoid link function $\sigma(x) = \frac{1}{1+e^{-x}}$ and $x \in [-H, H]$. Data is collected from 5 independent runs, and the results are averaged over these runs.

The target policy $\pi^\dagger$ is deterministic, which is to move *down* for each state (x, y).

While [13] proposed a novel approach and solid proofs for online RLHF, i.e., *P2R*, they did not provide simulation results to validate its effectiveness. To address this, we will first demonstrate the functionality of *P2R* in Figure 5, by presenting the results of *UCBVI-BF* running with *P2R* on human preferences, compared to running without *P2R* directly on rewards. It is shown that the *P2R* interface can effectively ensure the learning of optimal policy with human preferences, and convert a reward-based RL algorithm to a preference-based RL algorithm.

Then, we evaluate the attack strategy in Algorithm 2 on the online RLHF *P2R* interface. Figure 6 shows the results of *UCBVI-BF* running with *P2R* under attack. It is shown that the agent will learn the target policy $\pi^\dagger$ with manipulated human preferences. The policy of agent is significantly manipulated from move *right* to move *down* for each state (x, y).

In the simulation setting, the target action under target policy is *down* for each state (x, y), while the original optimal action is *right*. Figure 7 shows the success of the attack in the following aspects:

- Without attack, the agent will learn the optimal policy with human preferences, and the number of times taking original optimal actions is significantly higher than that of target actions, as shown in Figure 7a.
- With attack, the agent will instead learn the target policy, leading to a significantly larger number of times taking target actions among all trajectories, as shown in Figure 7b.
- The number of attacks performed, i.e., the number of trajectories need to be attacked, is sublinear to the total number of trajectories, as shown in Figure 7c.

Note that, in Figure 7c, we use attack count for y-axis, because with our attack strategy, the global attack cost is equal to the attack count multiplied by a constant, which will not affect the sublinear property.

B. Defense

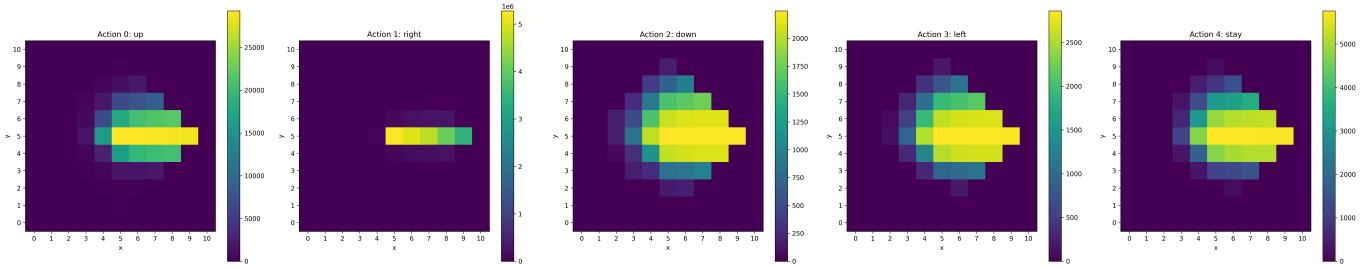
In this section, we evaluate the robust online RLHF strategy as in Algorithm 3. We consider the same tabular grid world environment as in Section VI-A. The agent has the same horizon and runs the same RL algorithm *UCBVI-BF*.

The attacker will flip C out of m human comparisons from 1 to 0 for each trajectory τ which does not move *down*. Meanwhile, the attacker will not manipulate the comparison results for those trajectories that move *down*. Here we set the attack cost budget $C = \frac{8\sqrt{d_{\overline{\mathcal{R}}}}\ln(4d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0\alpha}$, and the comparison number $m = \frac{256 d_{\overline{\mathcal{R}}}\ln(4d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0^2\alpha^2}$.

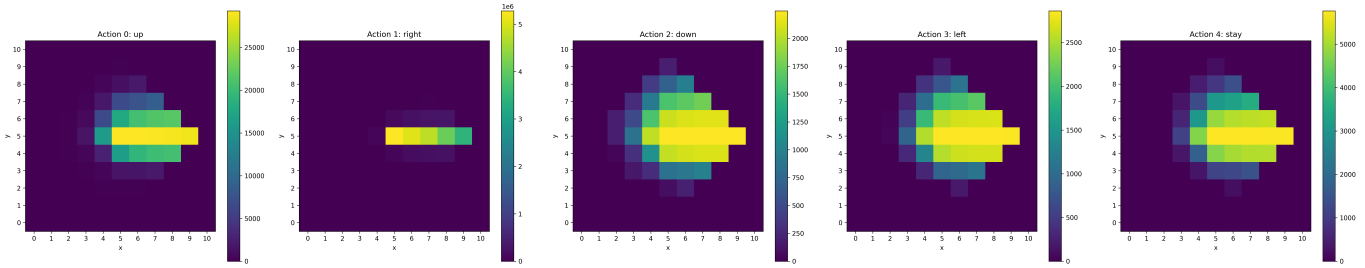
Figure 8 shows the results of *UCBVI-BF* running with *R-P2R* under attack. Compared with the heatmaps in Figure 5, it is shown that the agent can defend against the attack and learn the optimal policy with human preferences by using *R-P2R*. Figure 9 shows the action count for *R-P2R* under attack, which can be compared with Figure 7a, the result of *P2R* without attack. Table I further provides the numerical results of the optimal actions taken for 2.5×10^7 trajectories when running *R-P2R* under attack and *P2R* without attack. The simulation shows that the agent can learn the optimal policy with human preferences by using *R-P2R* even under attack. The robust online RLHF strategy *R-P2R* is successful in defending against the attack.

VII. Conclusion

In this paper, we have proposed a novel adversarial human feedback attack strategy on online RLHF algorithms. We have



(a) *UCBVI-BF* with *P2R* on human preferences



(b) *UCBVI-BF* on rewards

Fig. 5: *UCBVI-BF* with and without *P2R* (no attack)

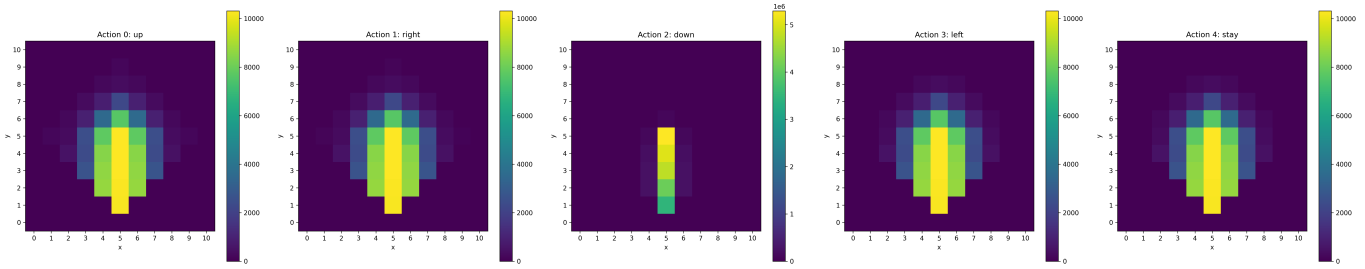


Fig. 6: *UCBVI-BF* with *P2R* under attack

Case	Optimal Actions Count	Percentage (%)
Attack *R-P2R*	24,429,701	97.72
No Attack *P2R*	24,429,805	97.72

TABLE I: Optimal actions taken for 2.5×10^7 trajectories when running *R-P2R* under attack and *P2R* without attack.

proved that our attack method is successful in manipulating the agent's learning process by providing misleading human preferences. We have also proposed a robust online RLHF strategy, named *R-P2R* interface, which is able to defend against any human feedback attacks whose local attack cost is bounded. The experimental results have verified our theoretical analysis and demonstrated the effectiveness of the proposed attack and defense strategy.

REFERENCES

[1] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30:4302–4310, Dec. 2017.

[2] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[3] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[4] J. Wu, J. Wang, C. Xiao, C. Wang, N. Zhang, and Y. Vorobeychik. Preference poisoning attacks on reward model learning. *arXiv preprint arXiv:2402.01920*, 2024.

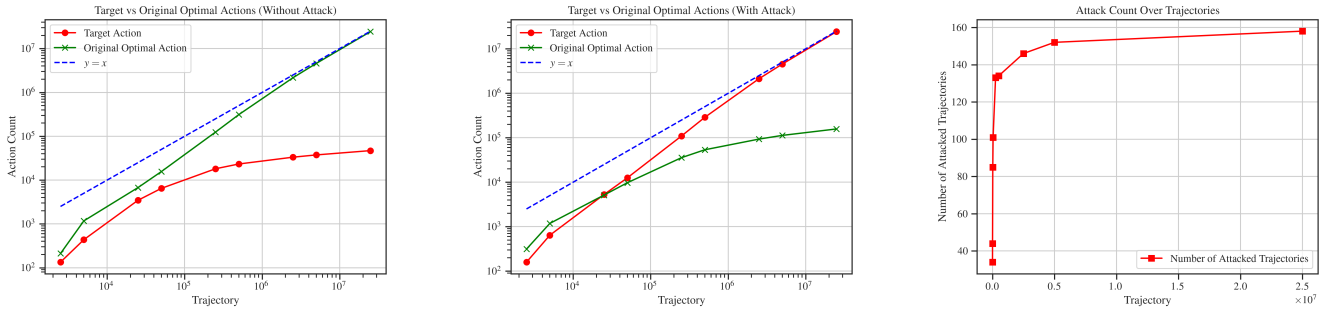
[5] J. Rando and F. Tramèr. Universal jailbreak backdoors from poisoned human feedback. *arXiv preprint arXiv:2311.14455*, 2023.

[6] T. Baumgärtner, Y. Gao, D. Alon, and D. Metzler. Best-of-venom: Attacking RLHF by injecting poisoned preference data. *arXiv preprint arXiv:2404.05530*, 2024.

[7] A. Nika, J. Nöther, D. Mandal, P. Kamalaruban, A. Singla, and G. Radanovic. Policy teaching via data poisoning in learning from human preferences. In *Proc. International Conference on Artificial Intelligence and Statistics*, Mai Khao, Thailand, May. 2025.

[8] P. Pathmanathan, S. Chakraborty, X. Liu, Y. Liang, and F. Huang. Is poisoning a real threat to LLM alignment? maybe more so than you think. *arXiv preprint arXiv:2406.12091*, 2024.

[9] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.



(a) Action count without attack (log scale) (b) Action count under attack (log scale) (c) Number of attacks performed

Fig. 7: The success of the attack on *P2R*

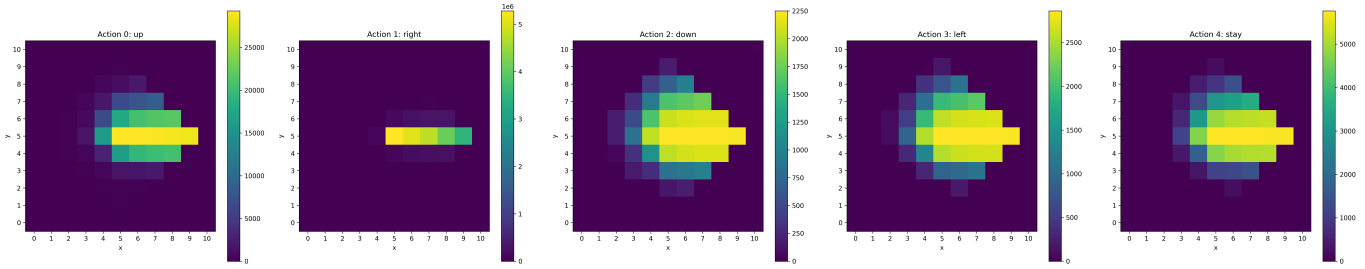


Fig. 8: *UCBVI-BF* with *R-P2R* under attack

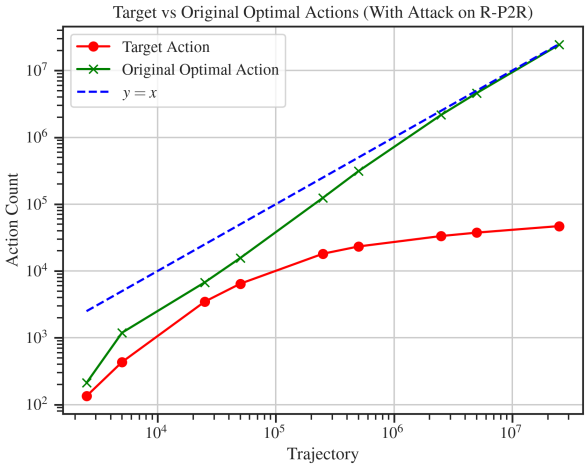


Fig. 9: Action count for *R-P2R* under attack (log scale)

[10] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[11] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, Dec. 2023.

[12] A. Saha, A. Pacchiano, and J. Lee. Dueling RL: Reinforcement learning with trajectory preferences. In *Proc. International Conference on Artificial Intelligence and Statistics*, volume 206, pages 6263–6289, Valencia, Spain, Apr. 2023.

[13] Y. Wang, Q. Liu, and C. Jin. Is RLHF more difficult than standard RL? a theoretical perspective. In *Advances in Neural Information Processing Systems*, volume 36, pages 76006–76032, New Orleans, LA, Dec. 2023.

[14] B. Zhu, M. Jordan, and J. Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proc. International Conference on Machine Learning*, volume 202, pages 43037–43067, Honolulu, HI, Jul. 2023.

[15] E. R. Balda, A. Behboodi, and R. Mathar. Perturbation analysis of learning algorithms: Generation of adversarial examples from classification to regression. *IEEE Transactions on Signal Processing*, 67(23):6078–6091, 2019.

[16] Y. Jin and L. Lai. Fairness-aware regression robust to adversarial attacks. *IEEE Transactions on Signal Processing*, 71:4092–4105, 2023.

[17] D. A. Berry and B. Fristedt. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5(71-87):7–7, 1985.

[18] R. S. Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

[19] K-S Jun, L. Li, Y. Ma, and X. Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 31, pages 3640–3649, Montréal, Canada, Dec. 2018.

[20] F. Liu and N. Shroff. Data poisoning attacks on stochastic bandits. In *Proc. International Conference on Machine Learning*, volume 97, Long Beach, CA, Jun. 2019.

[21] G. Liu and L. Lai. Action-manipulation attacks against stochastic bandits: Attacks and defense. *IEEE Transactions on Signal Processing*, 68:5152–5165, 2020.

[22] Y. Ma and Z. Zhou. Adversarial attacks on adversarial bandits. *arXiv preprint arXiv:2301.12595*, 2023.

[23] C. Yang, G. Liu, and L. Lai. Reward attack on stochastic bandits with non-stationary rewards. In *Proc. Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2023.

[24] G. Liu and L. Lai. Efficient action poisoning attacks on linear contextual bandits. *arXiv preprint arXiv:2112.05367*, 2021.

[25] V. Behzadan and A. Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In *Proc. International Conference on Machine Learning and Data Mining in Pattern Recognition*, volume 13, pages 262–275, New York, NY, Jul. 2017.

[26] Y. Huang and Q. Zhu. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *Proc. International Conference*

on Decision and Game Theory for Security, volume 10, pages 217–237, Stockholm, Sweden, Oct. 2019.

[27] Y. Ma, X. Zhang, W. Sun, and X. Zhu. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32:14570–14580, Dec. 2019.

[28] X. Zhang, Y. Ma, A. Singla, and X. Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *Proc. International Conference on Machine Learning*, volume 119, pages 11225–11234, Online, Jul. 2020.

[29] G. Liu and L. Lai. Provably efficient black-box action poisoning attacks against reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12400–12410, Dec. 2021.

[30] X. Dong, Z. Wu, Q. Ling, and Z. Tian. Byzantine-robust distributed online learning: Taming adversarial participants in an adversarial environment. *IEEE Transactions on Signal Processing*, 72:235–248, 2024.

[31] F. Li, L. Lai, and S. Cui. On the adversarial robustness of lasso based feature selection. *IEEE Transactions on Signal Processing*, 69:5555–5567, 2021.

[32] K. Pillutla, S. M. Kakade, and Z. Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[33] T. Lykouris, V. Mirrokni, and R. Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proc. Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122, Los Angeles, CA, Jun. 2018.

[34] A. Gupta, T. Koren, and K. Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Proc. Conference on Learning Theory*, volume 99, pages 1562–1578, Phoenix, AZ, Jun. 2019.

[35] Z. Guan, K. Ji, D. Bucci Jr, T. Hu, J. Palombo, M. Liston, and Y. Liang. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *Proc. AAAI Conference on Artificial Intelligence*, volume 34, pages 4036–4043, New York, NY, Feb. 2020.

[36] I. Bogunovic, A. Losalka, A. Krause, and J. Scarlett. Stochastic linear bandits robust to adversarial attacks. In *Proc. International Conference on Artificial Intelligence and Statistics*, volume 130, Apr. 2021.

[37] L. Yang, M. Hajiesmaili, S. Talebi, J. C. S. Lui, and W. S. Wong. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. In *Advances in Neural Information Processing Systems*, volume 33, Dec. 2020.

[38] Q. Ding, C. J. Hsieh, and J. Sharpnack. Robust stochastic linear contextual bandits under adversarial attacks. In *Proc. International Conference on Artificial Intelligence and Statistics*, Mar. 2022.

[39] Z. Feng, D. Parkes, and H. Xu. The intrinsic robustness of stochastic bandits to strategic manipulation. In *Proc. International Conference on Machine Learning*, volume 119, Jul. 2020.

[40] Y. Kang, C. J. Hsieh, and T. C. M. Lee. Robust lipschitz bandits to adversarial corruptions. In *Advances in Neural Information Processing Systems*, volume 36, New Orleans, LA, Dec. 2023.

[41] C. Yang, G. Liu, and L. Lai. Stochastic bandits with non-stationary rewards: Reward attack and defense. *IEEE Transactions on Signal Processing*, 72:5007–5020, 2024.

[42] C. Tessler, Y. Efroni, and S. Mannor. Action robust reinforcement learning and applications in continuous control. In *Proc. International Conference on Machine Learning*, volume 97, pages 6215–6224, Long Beach, CA, Jun. 2019.

[43] X. Zhang, Y. Chen, X. Zhu, and W. Sun. Corruption-robust offline reinforcement learning. In *Proc. International Conference on Artificial Intelligence and Statistics*, volume 151, pages 5757–5773, Online, Mar. 2022.

[44] C. Ye, R. Yang, Q. Gu, and T. Zhang. Corruption-robust offline reinforcement learning with general function approximation. *Advances in Neural Information Processing Systems*, 36:36208–36221, Dec. 2023.

[45] J. Cheng, G. Xiong, X. Dai, Q. Miao, Y. Lv, and F-Y Wang. Rime: Robust preference-based reinforcement learning with noisy preferences. *arXiv preprint arXiv:2402.17257*, 2024.

[46] D. Mandal, A. Nika, P. Kamalaruban, A. Singla, and G. Radanović. Corruption robust offline reinforcement learning with human feedback. *arXiv preprint arXiv:2402.06734*, 2024.

[47] Y. Yan, X. Lou, J. Li, Y. Zhang, J. Xie, C. Yu, Y. Wang, D. Yan, and Y. Shen. Reward-robust RLHF in LLMs. *arXiv preprint arXiv:2409.15360*, 2024.

[48] S. R. Chowdhury, A. Kini, and N. Natarajan. Provably robust DPO: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*, 2024.

[49] J. Wu, Y. Xie, Z. Yang, J. Wu, J. Chen, J. Gao, B. Ding, X. Wang, and X. He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *arXiv preprint arXiv:2407.07880*, 2024.

[50] W. Xiong, H. Dong, C. Ye, Z. Wang, H. Zhong, H. Ji, N. Jiang, and T. Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. *arXiv preprint arXiv:2312.11456*, 2023.

[51] T. Liu, Y. Zhao, R. Joshi, M. Khalman, M. Saleh, P. J. Liu, and J. Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

[52] S. R. Chowdhury, X. Zhou, and N. Natarajan. Differentially private reward estimation with preference feedback. In *Proc. International Conference on Artificial Intelligence and Statistics*, volume 238, pages 4843–4851, Valencia, Spain, May. 2024.

[53] R. Wu and W. Sun. Making RL with preference-based feedback efficient via randomization. *arXiv preprint arXiv:2310.14554*, 2023.

[54] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[55] J. McMahan, Y. Wu, X. Zhu, and Q. Xie. Optimal attack and defense for reinforcement learning. In *Proc. AAAI Conference on Artificial Intelligence*, volume 38, pages 14332–14340, Vancouver, Canada, Feb. 2024.

[56] A. Rangi, H. Xu, L. Tran-Thanh, and M. Franceschetti. Understanding the limits of poisoning attacks in episodic reinforcement learning. *arXiv preprint arXiv:2208.13663*, 2022.

[57] G. Liu and L. Lai. Efficient adversarial attacks on online multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:24401–24433, Dec. 2024.

[58] X. Y. Lee, Y. Esfandiari, K. L. Tan, and S. Sarkar. Query-based targeted action-space adversarial policies on deep reinforcement learning agents. In *Proc. ACM/IEEE International Conference on Cyber-Physical Systems*, pages 87–97, Online, May 2021.

[59] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

[60] R. Cheng, A. Verma, G. Orosz, S. Chaudhuri, Y. Yue, and J. Burdick. Control regularization for reduced variance reinforcement learning. In *Proc. International Conference on Machine Learning*, volume 97, pages 1141–1150, Long Beach, CA, Jun. 2019.

[61] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proc. International Conference on Machine Learning*, volume 70, pages 22–31, Sydney, Australia, Aug. 2017.

[62] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proc. International Conference on Machine Learning*, volume 70, pages 263–272, Sydney, Australia, Aug. 2017.

[63] Z. Zhang, Y. Chen, J. D. Lee, and S. S. Du. Settling the sample complexity of online reinforcement learning. In *Proc. Conference on Learning Theory*, volume 247, pages 5213–5219, Edmonton, Canada, Jun. 2024.

APPENDIX

A. Proof of Theorem IV.1

Lemma A.1. *Denote $T^\dagger$ as the set of all possible trajectories τ generated by the target policy $\pi^\dagger$, with the given transition of MDP. If the target policy is deterministic, then, for any trajectory $\tau \in T^\dagger$, we have: $P\left(\tau|\pi^\dagger\right) \geq P\left(\tau|\pi\right) \quad \forall\pi$.*

Proof. Denote $\tau : (s_1, a_1, s_2, \ldots, s_H, a_H, s_{H+1})$ as an arbitrary trajectory in $T^\dagger$, for any possible policy π and a deterministic target policy $\pi^\dagger$, we have:

$$
\begin{aligned}
&P\left(\tau|\pi^\dagger\right) \\
&= P(s_1)\pi^\dagger(a_1|s_1)\mathbb{P}(s_2|s_1,a_1)\ldots\pi^\dagger(a_H|s_H)\mathbb{P}(s_{H+1}|s_H,a_H) \\
&= P(s_1)\mathbb{P}(s_2|s_1,a_1)\ldots\mathbb{P}(s_{H+1}|s_H,a_H) \\
&\geq P(s_1)\pi(a_1|s_1)\mathbb{P}(s_2|s_1,a_1)\ldots\pi(a_H|s_H)\mathbb{P}(s_{H+1}|s_H,a_H) \\
&= P\left(\tau|\pi\right),
\end{aligned}
$$

where $\mathbb{P}$ represents the transition probability of MDP, and π represents the policy. □

Proof of Theorem IV.1. When we consider the trajectory reward function $r : (\mathcal{S} \times \mathcal{A})^H \to [0, H]$, the value function of a policy π is defined as $V_r^\pi = \mathbb{E}_{\tau\sim\pi}[r(\tau)]$.

Under attack, for any policy π, we have:

$$\begin{aligned} V_{r^\dagger}^\pi &= \sum_{\tau\in T} P(\tau|\pi) r^\dagger(\tau) \\ &= \sum_{\tau\in T^\dagger} P(\tau|\pi) r^\dagger(\tau) + \sum_{\tau\in T\setminus T^\dagger} P(\tau|\pi) r^\dagger(\tau) \\ &= \sum_{\tau\in T^\dagger} P(\tau|\pi) r^\dagger(\tau) \le \sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger) r^\dagger(\tau) = V_{r^\dagger}^{\pi^\dagger}, \end{aligned}$$

where the inequality is due to Lemma A.1. □

B. Proof of Theorem IV.2

The proof follows the logic of Section B.3 in [13]. The basic idea is that the RLHF algorithm will see the manipulated reward function $r^\dagger$ as the true underlying reward function, and sometimes the trajectories will be split into two parts $T^\dagger$ and $T \setminus T^\dagger$ for discussion.

Lemma A.2. *Under human feedback attack as Algorithm 2, in P2R as Algorithm 1, with $m = \Theta\left(\frac{\ln(1/\delta')}{\alpha^2\epsilon'^2}\right)$, for each τ such that the comparison oracle is queried, with probability $1-\delta'$,*

$$\left|\hat{r}(\tau) - \left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \le \epsilon'.$$

Proof. For those $\tau \in T^\dagger$, we have $\tilde{o}(\tau, \tau_0) = o(\tau, \tau_0)$, and $\tilde{o} \sim \text{Ber}\left(\sigma\left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right) = \text{Ber}\left(\sigma\left(r^*(\tau) - r^*(\tau_0)\right)\right)$.

By Hoeffding's inequality, with probability $1-\delta'$, we have:

$$\left|\bar{o} - \sigma\left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \le \sqrt{\frac{\ln(2/\delta')}{m}}.$$

Since $\hat{r} \leftarrow \arg\min_{x\in[-H,H]} |\sigma(x) - \bar{o}|$, we have:

$$\begin{aligned} &|\sigma(\hat{r}(\tau)) - \bar{o}| \le \left|\sigma\left(r^\dagger(\tau) - r^\dagger(\tau_0)\right) - \bar{o}\right| \le \sqrt{\frac{\ln(2/\delta')}{m}} \\ \implies &\left|\sigma(\hat{r}(\tau)) - \sigma\left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \le 2\sqrt{\frac{\ln(2/\delta')}{m}}. \end{aligned}$$

With Assumption III.1, we have:

$$\begin{aligned} &\left|\hat{r}(\tau) - \left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \\ &\le \frac{1}{\alpha} \cdot \left|\sigma(\hat{r}(\tau)) - \sigma\left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \\ &\le \frac{2}{\alpha}\sqrt{\frac{\ln(2/\delta')}{m}} \le \epsilon'. \end{aligned}$$

Similarly, for those $\tau \in T \setminus T^\dagger$, we have $\tilde{o}(\tau, \tau_0) \sim \text{Ber}\left(\sigma\left(-r^\dagger(\tau_0)\right)\right) = \text{Ber}\left(\sigma\left(-r^*(\tau_0)\right)\right)$. By Hoeffding's inequality, with probability $1-\delta'$, we have:

$$\left|\bar{o} - \sigma\left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| = \left|\bar{o} - \sigma\left(-r^*(\tau_0)\right)\right| \le \sqrt{\frac{\ln(2/\delta')}{m}}.$$

Thus, we have $\left|\hat{r}(\tau) - \left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \le \epsilon'$. □

Lemma A.3. *Under human feedback attack as Algorithm 2, in P2R as Algorithm 1, set $m = \Theta\left(\frac{d\ln(d/\delta)}{\epsilon_0^2\alpha^2}\right)$ and $\beta = \frac{\epsilon_0^2}{4}$. With probability $1-\delta$, the number of samples on which the comparison oracle is queried is at most $\dim_{\text{E}}\left(\overline{\mathcal{R}}^\dagger, \epsilon_0\right)$.*

Proof. The proof of Lemma B.6. in [13] holds with the attacked human feedback, by replacing the original underlying reward function r^* with the manipulated reward function $r^\dagger$, and the original reward function space $\overline{\mathcal{R}}$ with the manipulated reward function space $\overline{\mathcal{R}}^\dagger$. Some clarifications can be found in the proof of Lemma A.8. □

Lemma A.4. *Let $\overline{\mathcal{R}}^\dagger \subseteq \overline{\mathcal{R}}$ be two function classes of $(\mathcal{S}\times\mathcal{A})^H \to [-H, H]$. Then for any $\epsilon > 0$, we have: $\dim_{\text{E}}\left(\overline{\mathcal{R}}^\dagger, \epsilon\right) \le \dim_{\text{E}}\left(\overline{\mathcal{R}}, \epsilon\right)$.*

Proof. Suppose, for contradiction, that $\dim_{\text{E}}\left(\overline{\mathcal{R}}^\dagger, \epsilon\right) > \dim_{\text{E}}\left(\overline{\mathcal{R}}, \epsilon\right)$.

Let $\dim_{\text{E}}\left(\overline{\mathcal{R}}, \epsilon\right) = D$. By the assumption, there is an $\epsilon' \ge \epsilon$ and a sequence $\{\tau_1, \tau_2, \ldots, \tau_{D+1}\}$ that is ϵ'-independent with respect to $\overline{\mathcal{R}}^\dagger$. This means for each $i \in [D+1]$, there exists $f_i^\dagger, g_i^\dagger \in \overline{\mathcal{R}}^\dagger$ such that:

$$\sqrt{\sum_{j=1}^{i-1}\left(f_i^\dagger(\tau_j) - g_i^\dagger(\tau_j)\right)^2} \le \epsilon' \text{ and } \left|f_i^\dagger(\tau_i) - g_i^\dagger(\tau_i)\right| \ge \epsilon'.$$

Since $\overline{\mathcal{R}}^\dagger \subseteq \overline{\mathcal{R}}$, we have $f_i^\dagger, g_i^\dagger \in \overline{\mathcal{R}}$.

Consider the same sequence $\{\tau_1, \tau_2, \ldots, \tau_{D+1}\}$, and the same pairs $\left(f_i^\dagger, g_i^\dagger\right)$ but now regarded as members of $\overline{\mathcal{R}}$.

For each $i \in [D+1]$, because $f_i^\dagger, g_i^\dagger \in \overline{\mathcal{R}}$, we have the same inequalities:

$$\sqrt{\sum_{j=1}^{i-1}\left(f_i^\dagger(\tau_j) - g_i^\dagger(\tau_j)\right)^2} \le \epsilon' \text{ and } \left|f_i^\dagger(\tau_i) - g_i^\dagger(\tau_i)\right| \ge \epsilon'.$$

Thus, the sequence $\{\tau_1, \tau_2, \ldots, \tau_{D+1}\}$ also forms an ϵ'-independent sequence of length $D+1$ with respect to $\overline{\mathcal{R}}$.

By the definition of $\dim_{\text{E}}\left(\overline{\mathcal{R}}, \epsilon\right)$, the longest possible ϵ-independent sequence in $\overline{\mathcal{R}}$ is of length D. However, we found a sequence of length $D+1$, which contradicts the maximality of D. Therefore, we have $\dim_{\text{E}}\left(\overline{\mathcal{R}}^\dagger, \epsilon\right) \le \dim_{\text{E}}\left(\overline{\mathcal{R}}, \epsilon\right)$. □

Lemma A.5. *Under human feedback attack as Algorithm 2, with probability $1-\delta$, $r^\dagger \in \mathcal{B}_r$ throughout the execution of P2R as Algorithm 1.*

Lemma A.6. *Under human feedback attack as Algorithm 2, with probability $1-\delta$, for each τ in Line 3 of P2R as Algorithm 1, the returned reward $\hat{r}$ satisfies*

$$\left|\hat{r} - \left(r^\dagger(\tau) - r^\dagger(\tau_0)\right)\right| \le 2\epsilon_0.$$

Proof. The proof of Lemma B.7 and B.8 in [13] can be applied to Lemma A.5 and A.6, by replacing the original reward function r^* with the manipulated reward function $r^\dagger$. □

Proof of Theorem IV.2. Choose $\epsilon_0 := g(\epsilon)/2, \beta = \frac{\epsilon_0^2}{4}$ and $m = \Theta\left(\frac{d_{\overline{\mathcal{R}}} \ln\left(d_{\overline{\mathcal{R}}}/\delta\right)}{\epsilon_0^2 \alpha^2}\right)$. By Lemma A.6, the reward estimation $\hat{r}$ is $g(\epsilon)$-close to $\hat{r}^\dagger = r^\dagger(\tau) - r^\dagger(\tau_0)$. By definition of sample complexity, with probability $1-\delta$, the policy $\hat{\pi}$ returned by the RL algorithm running on the estimated reward function $\hat{r}$ is ϵ-optimal for $\hat{r}^\dagger$, which implies that it is also ϵ-optimal for the manipulated reward function $r^\dagger$. □

C. Proof of Theorem IV.3

Proof. From Theorem IV.2, we have $V_{r^\dagger}^{\hat{\pi}} \geq V_{r^\dagger}^{\pi^\dagger} - \epsilon$. By the definition of value function of a policy, it is equivalent to:

$$\begin{aligned} &\sum_{\tau \in T} P(\tau|\hat{\pi}) r^\dagger(\tau) \geq \sum_{\tau \in T} P(\tau|\pi^\dagger) r^\dagger(\tau) - \epsilon \\ \implies &\sum_{\tau \in T} \left(P(\tau|\pi^\dagger) - P(\tau|\hat{\pi})\right) r^\dagger(\tau) \leq \epsilon. \end{aligned}$$

Since for all $\tau \in T \setminus T^\dagger$, $r^\dagger(\tau) = 0$, it follows that:

$$\sum_{\tau \in T^\dagger} \left(P(\tau|\pi^\dagger) - P(\tau|\hat{\pi})\right) r^\dagger(\tau) \leq \epsilon.$$

With the assumption of deterministic target policy $\pi^\dagger$, we have for all $\tau : (s_1, a_1, s_2, \ldots, s_H, a_H, s_{H+1}) \in T^\dagger$:

$$\begin{aligned} &P(\tau|\pi^\dagger) = P(s_1)\mathbb{P}(s_2|s_1,a_1)\ldots\mathbb{P}(s_{H+1}|s_H,a_H), \\ &P(\tau|\hat{\pi}) \\ &= P(s_1)\hat{\pi}(a_1|s_1)\mathbb{P}(s_2|s_1,a_1)\ldots\hat{\pi}(a_H|s_H)\mathbb{P}(s_{H+1}|s_H,a_H) \\ &= P(\tau|\pi^\dagger)\cdot\hat{\pi}(\tau), \end{aligned}$$

where $\hat{\pi}(\tau) = \prod_{h=1}^H \hat{\pi}(a_h|s_h)$.

Thus, we have: $\sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger) r^\dagger(\tau)(1-\hat{\pi}(\tau)) \leq \epsilon$, where the $r^\dagger(\tau) = r^*(\tau) \in [0, H]$ for all possible $\tau \sim \pi^\dagger$. Since this relation holds for all kind of original underlying reward function r^*, it must hold for the case when $r^*(\tau) = H$ for all $\tau \in T^\dagger$ and maximizes the left side of the inequality. Therefore, we have: $\sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger)(1-\hat{\pi}(\tau)) \leq \frac{\epsilon}{H}$.

Moreover, since $\sum_{\tau\in T^\dagger} P(\tau|\hat{\pi}) + \sum_{\tau\in T\setminus T^\dagger} P(\tau|\hat{\pi}) = 1 = \sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger)$, thus:

$$\begin{aligned} \sum_{\tau\in T\setminus T^\dagger} P(\tau|\hat{\pi}) &= \sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger) - P(\tau|\hat{\pi}) \\ &= \sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger)(1-\hat{\pi}(\tau)) \leq \frac{\epsilon}{H}. \end{aligned}$$

The total variation distance between policies $\hat{\pi}$ and $\pi^\dagger$ is:

$$\begin{aligned} &\mathrm{TV}\left(\hat{\pi}\|\pi^\dagger\right) = \frac{1}{2}\sum_\tau \left|P(\tau|\hat{\pi}) - P(\tau|\pi^\dagger)\right| \\ &= \frac{1}{2}\left(\sum_{\tau\in T^\dagger}\left|P(\tau|\hat{\pi}) - P(\tau|\pi^\dagger)\right| + \sum_{\tau\in T\setminus T^\dagger}\left|P(\tau|\hat{\pi}) - P(\tau|\pi^\dagger)\right|\right) \\ &= \frac{1}{2}\left(\sum_{\tau\in T^\dagger} P(\tau|\pi^\dagger)(1-\hat{\pi}(\tau)) + \sum_{\tau\in T\setminus T^\dagger} P(\tau|\hat{\pi})\right) \leq \frac{\epsilon}{H}. \end{aligned}$$

□

D. Proof of Theorem IV.4 and Corollary IV.4.1

Proof of Theorem IV.4. Since $\mathscr{A}$ is a $g(\epsilon)$-robust RL algorithm with sample complexity $\mathcal{C}(\epsilon,\delta)$, the number of all samples / episodes / trajectories until it learns the policy $\hat{\pi}$ is bounded by $\mathcal{C}(\epsilon,\delta)$. m represents the number of human comparison queries for each trajectory. Because the attacker will generate the manipulated human comparison results $\tilde{o}$ for each comparison in m queries whenever the episode trajectory $\tau \in T \setminus T^\dagger$, which may not be the case for all $\mathcal{C}(\epsilon,\delta)$ samples. Thus, the attack cost is at most $\mathcal{C}(\epsilon,\delta)\cdot m$. □

Proof of Corollary IV.4. By Lemma A.3 and A.4, with probability $1-\delta$, the number of samples on which the comparison oracle is queried is at most $\dim_{\mathrm{E}}\left(\overline{\mathcal{R}}, \epsilon_0\right)$. With the attack design, we know that not all these samples are in $T \setminus T^\dagger$, meaning the attacker will not generate the manipulated human comparison results $\tilde{o}$ for all these samples. Thus, the attack cost is at most $\dim_{\mathrm{E}}\left(\overline{\mathcal{R}}, \epsilon_0\right)\cdot m \leq \tilde{\mathcal{O}}\left(\frac{d_{\overline{\mathcal{R}}}^2}{\epsilon^2\alpha^2}\right)$. By Theorem IV.2, with probability $1-\delta$, the RL algorithm will return an ϵ-optimal policy $\hat{\pi}$. The probability $1-2\delta$ is from union bound. □

E. Proof of Theorem V.1

Lemma A.7. *With* $m = \max\left\{\frac{16\ln(2/\delta')}{\alpha^2\epsilon'^2}, \frac{8C}{\alpha\epsilon'}\right\}$, *for each* τ *such that the comparison oracle is queried in R-P2R as Algorithm 4, with probability* $1-\delta'$,

$$|\hat{r}(\tau) - (r^*(\tau) - r^*(\tau_0))| \leq \epsilon'.$$

Proof. Denote the average comparison result in *P2R* and *R-P2R* before attack as $\bar{o}$, and the corresponding one after attack as $\bar{o}^\dagger$. Denote the new update rule in *R-P2R* (Algorithm 4 line 3) before attack as $o' = \gamma + (1-\gamma)\bar{o}$, and the corresponding one after attack as $o^\dagger = \gamma + (1-\gamma)\bar{o}^\dagger$. Denote the underlying reward function before attack as r^*, and the manipulated reward function after attack as $r^\dagger$. The notations before and after attack are summarized as follows:

$$\begin{gathered} \bar{o} \to \bar{o}^\dagger \\ o' = \gamma + (1-\gamma)\bar{o} \to o^\dagger = \gamma + (1-\gamma)\bar{o}^\dagger \\ r^* \to r^\dagger. \end{gathered}$$

Suppose that the comparison oracle is queried for trajectory τ. As commonly assumed in the literature, the human feedbacks (o-s) are sampled from a Bernoulli distribution with parameter $\sigma(r^*(\tau) - r^*(\tau_0))$. By Hoeffding's inequality, with probability $1-\delta'$, we have:

$$\begin{aligned} |\bar{o} - \sigma(r^*(\tau) - r^*(\tau_0))| &\leq \sqrt{\frac{\ln(2/\delta')}{m}}, \\ \left|\bar{o}^\dagger - \sigma(r^\dagger(\tau) - r^\dagger(\tau_0))\right| &\leq \sqrt{\frac{\ln(2/\delta')}{m}}. \end{aligned}$$

With the existence of the attack, we have:

$$\begin{aligned} \hat{r}(\tau) &= \arg\min_{x\in[-H,H]} \left|\sigma(x) - o^\dagger\right| \\ &= \arg\min_{x\in[-H,H]} \left|\sigma(x) - \gamma - (1-\gamma)\bar{o}^\dagger\right|. \end{aligned}$$

Thus, we have:

$$\begin{aligned}
&\left|\sigma(\hat{r}(\tau)) - \gamma - (1-\gamma)\bar{o}^\dagger\right| \\
&\le \left|\sigma(r^*(\tau) - r^*(\tau_0)) - \gamma - (1-\gamma)\bar{o}^\dagger\right| \\
&= \left|\sigma(r^*(\tau) - r^*(\tau_0)) - \bar{o}^\dagger - \gamma + \gamma\bar{o}^\dagger\right| \\
&= \left|\sigma(r^*(\tau) - r^*(\tau_0)) - \bar{o} + \bar{o} - \bar{o}^\dagger - \gamma + \gamma\bar{o}^\dagger\right| \\
&\le \left|\sigma(r^*(\tau) - r^*(\tau_0)) - \bar{o}\right| + \left|\bar{o} - \bar{o}^\dagger - \gamma + \gamma\bar{o}^\dagger\right|.
\end{aligned}$$

Also, we have $\left|\gamma\bar{o}^\dagger - \gamma\right| \le \gamma$ and:

$$\begin{aligned}
\left|\bar{o} - \bar{o}^\dagger\right| &= \left|\frac{1}{m}\sum_{i=1}^m o_i - \frac{1}{m}\sum_{i=1}^m \tilde{o}_i\right| \\
&\le \frac{1}{m}\sum_{i=1}^m |o_i - \tilde{o}_i| \le \frac{1}{m}\sum_{i=1}^m \mathbb{1}(o_i \ne \tilde{o}_i) \le \frac{C}{m}.
\end{aligned}$$

Thus, with the update rule of γ, and $k = 1, 2, \ldots$, we have:

$$\begin{aligned}
&\left|\sigma(\hat{r}(\tau)) - \gamma - (1-\gamma)\bar{o}^\dagger\right| \\
&\le \left|\sigma(r^*(\tau) - r^*(\tau_0)) - \bar{o}\right| + \left|\bar{o} - \bar{o}^\dagger - \gamma + \gamma\bar{o}^\dagger\right| \\
&\le \sqrt{\frac{\ln(2/\delta')}{m}} + \frac{C}{m} + \frac{C}{C+km} \\
&\le \sqrt{\frac{\ln(2/\delta')}{m}} + \frac{2C}{m}.
\end{aligned}$$

It follows that:

$$\left|\sigma(\hat{r}(\tau)) - \sigma(r^*(\tau) - r^*(\tau_0))\right| \le 2\sqrt{\frac{\ln(2/\delta')}{m}} + \frac{4C}{m}.$$

By Assumption III.1, we have:

$$\begin{aligned}
\left|\hat{r}(\tau) - (r^*(\tau) - r^*(\tau_0))\right| &\le \frac{1}{\alpha}\left|\sigma(\hat{r}(\tau)) - \sigma(r^*(\tau) - r^*(\tau_0))\right| \\
&\le \frac{2}{\alpha}\sqrt{\frac{\ln(2/\delta')}{m}} + \frac{4C}{\alpha m}.
\end{aligned}$$

To make the right side of the inequality less than ϵ', we could pick m such that:

$$\frac{2}{\alpha}\sqrt{\frac{\ln(2/\delta')}{m}} \le \frac{\epsilon'}{2} \quad \text{and} \quad \frac{4C}{\alpha m} \le \frac{\epsilon'}{2}.$$

In other words, we could pick $m = \max\left\{\frac{16\ln(2/\delta')}{\alpha^2\epsilon'^2}, \frac{8C}{\alpha\epsilon'}\right\}$, and then with probability $1-\delta'$ we have: $\left|\hat{r}(\tau) - (r^*(\tau) - r^*(\tau_0))\right| \le \epsilon'$. □

Lemma A.8. *In R-P2R as Algorithm 3, set* $m = \max\left\{\frac{256d\ln(4d/\delta)}{\epsilon_0^2\alpha^2}, \frac{32C\sqrt{d}}{\alpha\epsilon_0}\right\}$, *and* $\beta = \frac{\epsilon_0^2}{4}$. *With probability* $1-\delta$, *the number of samples on which the comparison oracle is queried is at most* $\dim_{\mathrm{E}}\left(\overline{\mathcal{R}}, \epsilon_0\right)$.

Proof. Note that the *R-P2R* does not know what will be the reward function space after the attack, i.e., $\overline{\mathcal{R}}^\dagger$. It will search within the original reward function space $\overline{\mathcal{R}}$. The proof of Lemma B.6. in [13] can be applied here. However, it needs to be clarified that the use of Lemma A.7 in the proof is through choose $\epsilon' = \frac{\epsilon_0}{4\sqrt{d}}$ and $\delta' = \frac{\delta}{2d}$. Union bound is used to achieve the probability $1-\delta$.

Thus, with probability $1-\delta$, $\forall k \le \min(K, 2d)$, $|\hat{r}_k - \tilde{r}^*(\tau_k)| \le \frac{\epsilon_0}{4\sqrt{d}}$ (by Lemma A.7). Then for any $i \le K$,

$$\begin{aligned}
\sum_{k\le i}\left(\tilde{r}_i(\tau_k) - \tilde{r}^*(\tau_k)\right)^2 &\le \sum_{k\le i}\left(\tilde{r}_i(\tau_k) - \hat{r}_k\right)^2 \\
&\quad + 2\sum_{k\le i}\left|\tilde{r}_i(\tau_k) - \hat{r}_k\right|\left|\hat{r}_k - \tilde{r}^*(\tau_k)\right| \\
&\quad + \sum_{k\le i}\left(\hat{r}_k - \tilde{r}^*(\tau_k)\right)^2 \\
&\le \beta + 2\sqrt{K\beta}\frac{\epsilon_0}{4\sqrt{d}} + K\left(\frac{\epsilon_0^2}{4d}\right)^2 \le \epsilon_0^2,
\end{aligned}$$

where the $\sqrt{K\beta}$ term is from Cauchy-Schwarz inequality:

$$\begin{aligned}
\left(\sum_{k\le i}\left|\tilde{r}_i(\tau_k) - \hat{r}_k\right| \cdot 1\right)^2 &\le \left(\sum_{k\le i} 1^2\right)\cdot\left(\sum_{k\le i}\left|\tilde{r}_i(\tau_k) - \hat{r}_k\right|^2\right) \\
&\le K \cdot \sum_{k\le i}\left(\tilde{r}_i(\tau_k) - \hat{r}_k\right)^2 \le K\beta.
\end{aligned}$$

□

Lemma A.9. *With probability* $1-\delta$, $r^* \in \mathcal{B}_r$ *throughout the execution of R-P2R as Algorithm 3.*

Proof. By Lemma A.7 and Lemma A.8, with probability $1-\delta$, at every step of *R-P2R*, each reward estimation $\hat{r}$ is close to $r^*(\tau) - r^*(\tau_0)$, thus:

$$\sum_{(\hat{r},\tau)\in\mathcal{D}}\left(\hat{r} - (r^*(\tau) - r^*(\tau_0))\right)^2 \le d\left(\frac{\epsilon_0}{4\sqrt{d}}\right)^2 \le \beta.$$

□

Lemma A.10. *With probability* $1-\delta$, *for each* τ *in Line 3 of R-P2R as Algorithm 3, the returned reward* $\hat{r}$ *satisfies:*

$$\left|\hat{r} - (r^*(\tau) - r^*(\tau_0))\right| \le 2\epsilon_0.$$

Proof. The proof of Lemma B.8. in [13] holds with Lemma A.9. □

Proof of Theorem V.1. Set $\epsilon_0 = g(\epsilon)/2$, $\beta = \frac{\epsilon_0^2}{4}$ and $m = \max\left\{\frac{256d_{\overline{\mathcal{R}}}\ln(4d_{\overline{\mathcal{R}}}/\delta)}{\epsilon_0^2\alpha^2}, \frac{32C\sqrt{d_{\overline{\mathcal{R}}}}}{\alpha\epsilon_0}\right\}$. By Lemma A.10, with probability $1-\delta$, the reward estimation $\hat{r}$ is $g(\epsilon)$-close to $r^*(\tau) - r^*(\tau_0)$. By definition of sample complexity, with probability $1-\delta$, the policy $\hat{\pi}$ returned by the RL algorithm running on the estimated reward function $\hat{r}$ is ϵ-optimal for $r^*(\tau) - r^*(\tau_0)$, and thus it is also ϵ-optimal for the original reward function r^*. The probability $1-2\delta$ is from union bound.

Since the RL algorithm has a sample complexity of $\mathcal{C}(\epsilon, \delta)$, the total number of samples is at most $\mathcal{C}(\epsilon, \delta)\cdot m$. By Lemma A.8, the number of human comparison queries is at most:

$$\dim_{\mathrm{E}}\left(\overline{\mathcal{R}}, \epsilon_0\right)\cdot m \le \max\left\{\frac{1024d_{\overline{\mathcal{R}}}^2\ln(4d_{\overline{\mathcal{R}}}/\delta)}{g^2(\epsilon)\alpha^2}, \frac{64Cd_{\overline{\mathcal{R}}}^{3/2}}{\alpha g(\epsilon)}\right\}.$$

□