MAKING RL WITH PREFERENCE-BASED FEEDBACK EFFICIENT VIA RANDOMIZATION

Runzhe Wu
Department of Computer Science
Cornell University
rw646@cornell.edu

Wen Sun
Department of Computer Science
Cornell University
ws455@cornell.edu

ABSTRACT

Reinforcement Learning algorithms that learn from human feedback (RLHF) need to be efficient in terms of *statistical complexity, computational complexity, and query complexity*. In this work, we consider the RLHF setting where the feedback is given in the format of preferences over pairs of trajectories. In the linear MDP model, using randomization in algorithm design, we present an algorithm that is sample efficient (i.e., has near-optimal worst-case regret bounds) and has polynomial running time (i.e., computational complexity is polynomial with respect to relevant parameters). Our algorithm further minimizes the query complexity through a novel randomized active learning procedure. In particular, our algorithm demonstrates a near-optimal tradeoff between the regret bound and the query complexity. To extend the results to more general nonlinear function approximation, we design a model-based randomized algorithm inspired by the idea of Thompson sampling. Our algorithm minimizes Bayesian regret bound and query complexity, again achieving a near-optimal tradeoff between these two quantities. Computation-wise, similar to the prior Thompson sampling algorithms under the regular RL setting, the main computation primitives of our algorithm are Bayesian supervised learning oracles which have been heavily investigated on the empirical side when applying Thompson sampling algorithms to RL benchmark problems.

1 INTRODUCTION

Reinforcement learning from human feedback (RLHF) has been widely used across various domains, including robotics (Jain et al., 2013; 2015) and natural language processing (Stiennon et al., 2020; Ouyang et al., 2022). Unlike standard RL, RLHF requires the agent to learn from feedback in the format of preferences between pairs of trajectories instead of per-step reward since assigning a dense reward function for each state is challenging in many tasks. For instance, in natural language generation, rating each generated token individually is challenging. Hence, it is more realistic to ask humans to compare two pieces of text and indicate their preference. Recent works have shown that, by integrating preference-based feedback into the training process, we can align models with human intention and enable high-quality human-machine interaction.

Despite the existing empirical applications of RLHF, its theoretical foundation remains far from satisfactory. Empirically, researchers first learn reward models from preference-based feedback and then optimize the reward models via policy gradient-based algorithms such as PPO (Schulman et al., 2017). Questions such as whether or not the learned reward model is accurate, whether PPO is sufficient for deep exploration, and how to strategically collect more feedback on the fly are often ignored. Theoretically, prior works study the regret bound for RL with preference-based feedback (Saha et al., 2023; Chen et al., 2022). Despite achieving sublinear worst-case regret, these algorithms are computationally intractable even for simplified models such as tabular Markov Decision Processes (MDPs). This means that we cannot easily leverage the algorithmic ideas in prior work to guide or improve how we perform RLHF in practice.

In addition to maximizing reward, another important metric in RLHF is the query complexity since human feedback is expensive to collect. To illustrate, we note that InstructGPT's training data comprises a mere 30K instances of human feedback (Ouyang et al., 2022), which is significantly fewer than the internet-scale dataset for training the GPT-3 base model. This underscores the challenge of scaling up the size of human feedback datasets. Ross et al. (2013); Laskey et al. (2016) also pointed out that extensively querying for feedback puts too much burden on human experts. Empirically, Lightman et al. (2023) observes that active learning reduces query complexity and improves the learned reward model. In theory, query complexity is mostly studied in the settings of active learning, online learning, and bandits (Cesa-Bianchi et al., 2005; Dekel et al., 2012; Agarwal, 2013; Hanneke & Yang, 2021; Zhu & Nowak, 2022; Sekhari et al., 2023a;b), but overlooked in RL.

In this work, *we aim to design new RL algorithms that can learn from preference-based feedback and can be efficient in statistical complexity (i.e., regret), computational complexity, and query complexity*. In particular, we strike a near-optimal balance between regret minimization and query complexity minimization. To achieve this goal, our key idea is to use *randomization* in algorithm design. We summarize our new algorithmic ideas and key contributions as follows.

1. For MDPs with linear structure (i.e., linear MDP (Jin et al., 2020)), we propose the first RL algorithm that achieves sublinear worst-case regret and computational efficiency simultaneously with preference-based feedback. Even when reduced to tabular MDPs, it is still the first to achieve a no-regret guarantee and computational efficiency. Moreover, it has an active learning procedure and attains a near-optimal tradeoff between the regret and the query complexity. Our algorithm adds *random Gaussian noises* to the learned state-action-wise reward model and the least-squares value iteration (LSVI) procedure. Using random noise instead of the UCB-style technique (Azar et al., 2017) preserves the Markovian property in the reward model and allows one to use dynamic programming to achieve computation efficiency.
2. For function approximation beyond linear, we present a model-based Thompson-sampling (TS) algorithm that forms posterior distributions over the transitions and reward models. Assuming the transition and the reward model class both have small ℓ_1*-norm eluder dimension* – a structural condition introduced in Liu et al. (2022a) that is more general than the common ℓ_2-norm eluder dimension (Russo & Van Roy, 2013), we show that our algorithm again achieves a near-optimal tradeoff between the Bayesian regret and the Bayesian query complexity. Computation-wise, similar to previous TS algorithms for regular RL (e.g., Osband et al. (2013)), the primary computation primitives are Bayesian supervised learning oracles for transition and reward learning.
3. Our query conditions for both algorithms are based on variance-style uncertainty quantification of the preference induced by the randomness of the reward model. We query for preference feedback only when the uncertainty of the preference on a pair of trajectories is large. Approximately computing the uncertainty can be easily done using i.i.d. random reward models drawn from the reward model distribution, which makes the active query procedure computationally tractable.

Overall, while our main contribution is on the theoretical side, our theoretical investigation provides several new practical insights. For instance, for regret minimization, our algorithms propose to draw a pair of trajectories with one from the latest policy and the other from an older policy instead of drawing two trajectories from the same policy (e.g., Christiano et al. (2017)), avoiding the situation of drawing two similar trajectories when the policy becomes more and more deterministic. Our theory shows that drawing two trajectories from a combination of new and older policies balances exploration and exploitation better. Another practical insight is the variance-style uncertainty measure for designing the query condition. Compared to more standard active learning procedure that relies on constructing version space and confidence intervals (Dekel et al., 2012; Puchkin & Zhivotovskiy, 2021; Zhu & Nowak, 2022; Sekhari et al., 2023a;b), our new approach comes with strong theoretical guarantees and is more computationally tractable. It is also amenable to existing implementations of Thompson sampling RL algorithms (e.g., using bootstrapping to approximate the posterior sampling (Osband et al., 2016a; 2023)).

2 COMPARISON TO PRIOR WORK

RL with preference-based feedback. Many recent works have obtained statistically efficient algorithms but are computationally inefficient even for tabular MDPs due to intractable policy search and version space construction (Chen et al., 2022; Zhan et al., 2023a;b; Saha et al., 2023). For example, Zhan et al. (2023b); Saha et al. (2023) use the idea from optimal design and rely on the computation oracle: $\arg\max_{\pi,\pi'\in\Pi} \|\mathbb{E}_{s,a\sim\pi}\phi(s,a) - \mathbb{E}_{s,a\sim\pi'}\phi(s,a)\|_A$ with some positive definite matrix A.

Here $\|x\|_A^2 := x^\top Ax$, and ϕ is some state-action feature.[1] It is unclear how to implement this oracle since standard planning approaches based on dynamic programming cannot be applied here. In addition, these methods also actively maintain a policy space by eliminating potentially sub-optimal policies. The policy class can be exponentially large even in tabular settings, so how to maintain it computationally efficiently is unclear. We provide a more detailed discussion on the challenges in achieving computational efficiency in RLHF in Appendix A.

While the work mentioned above is intractable even for tabular MDPs, there are some other works that could be computationally efficient but have weaker statistical results. For instance, very recently, Wang et al. (2023) proposed a reduction framework that can be computationally efficient (depending on the base algorithm used in the reduction). However, their algorithms have PAC bounds while we focus on regret minimization. Moreover, we achieve a near-optimal balance between regret and query complexity. Novoseller et al. (2020) proposed a posterior sampling algorithm for tabular MDP but their analysis is asymptotic (i.e., they do not address exploration, exploitation, and query complexity tradeoff). Xu et al. (2020) proposed efficient algorithms that do reward-free exploration. However, it is limited to tabular MDPs and PAC bounds.

In contrast to the above works, our algorithms aim to achieve efficiency in statistical, computational, and query complexities simultaneously. Our algorithms leverage *randomization* to balance exploration, exploitation, and feedback query. Randomization allows us to avoid non-standard computational oracles and only use standard Dynamic Programming (DP) based oracles (e.g., value iteration), which makes our algorithm computationally more tractable. Prior works that simultaneously achieve efficiency in all three aspects are often restricted in the bandit and imitation learning settings where the exploration problem is much easier (Sekhari et al., 2023a).

RL via randomization. There are two lines of work that study RL via randomization. The first injects random noise into the learning object to encourage exploration. A typical example is the randomized least-squares value iteration (RLSVI) (Osband et al., 2016b), which adds Gaussian noise into the least-squares estimation and achieves near-optimal worst-case regret (Zanette et al., 2020; Agrawal et al., 2021) for linear MDPs. The other line of work is Bayesian RL and uses Thompson sampling (TS) (Osband et al., 2013; Osband & Van Roy, 2014b;a; Gopalan & Mannor, 2015; Agrawal & Jia, 2017; Efroni et al., 2021; Zhong et al., 2022; Agarwal & Zhang, 2022). They achieve provable Bayesian regret upper bound by maintaining posterior distributions over models.

Active learning. Numerous studies have studied active learning across various settings (Cesa-Bianchi et al., 2005; Dekel et al., 2012; Agarwal, 2013; Hanneke & Yang, 2015; 2021; Zhu & Nowak, 2022; Sekhari et al., 2023b;a). However, most of them focus on the bandits and online learning settings, and their active learning procedures are usually computationally intractable due to computing version spaces or upper and lower confidence bounds. In contrast, we design a variance-style uncertainty quantification for our query condition, which can be easily estimated by random samples of reward model. This makes our active learning procedure more computationally tractable.

3 PRELIMINARY

Notations. For two real numbers a and b, we denote $[a, b] := \{x : a \le x \le b\}$. For an integer N, we denote $[N] := \{1, 2, \ldots, N\}$. For a set $\mathcal{S}$, we denote $\Delta(\mathcal{S})$ as the set of distributions over $\mathcal{S}$. Let $d_{\mathrm{TV}}(\cdot,\cdot)$ denote the total variation distance.

We consider a finite-horizon Markov decision process (MDP), which is a tuple $M(\mathcal{S}, \mathcal{A}, r^\star, P^\star, H)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P^\star : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, $r^\star : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, and H is the length of the episode. The interaction proceeds for T rounds. At each round $t \in [T]$, we need to select two policies π_t^0 and π_t^1 and execute them separately, which generates two trajectories τ_t^0 and τ_t^1 where $\tau_t^i = (s_{t,1}^i, a_{t,1}^i, \ldots, s_{t,H}^i, a_{t,H}^i)$ for $i \in \{0, 1\}$. For the ease of notation, we assume a fixed initial state s_1. Then, we need to decide whether to make a query for the preference between τ_t^0 and τ_t^1. If making a query, we obtain a preference feedback $o_t \in \{0, 1\}$ that is sampled from the Bernoulli distribution:

$$\Pr(o_t = 1 \mid \tau_t^1, \tau_t^0, r^\star) = \Pr(\tau_t^1 \text{ is preferred to } \tau_t^0 \mid r^\star) = \Phi\big(r^\star(\tau_t^1) - r^\star(\tau_t^0)\big)$$

---

[1] These works typically assume trajectory-wise feature $\phi(\tau)$ for a trajectory τ. However, even when specified to state-action-wise features, these algorithms are still computationally intractable, even in tabular MDPs.

where $r^\star(\tau_t^i) = \sum_{h=1}^H r^\star(s_{t,h}^i, a_{t,h}^i)$ for $i \in \{0,1\}$ is the trajectory reward, and $\Phi : \mathbb{R} \to [0,1]$ is a monotonically increasing link function. We note that, by symmetry, we have $\Phi(r^\star(\tau_t^0) - r^\star(\tau_t^1)) + \Phi(r^\star(\tau_t^1) - r^\star(\tau_t^0)) = 1$. If not making a query, we receive no feedback.

This feedback model is weaker than the standard RL where the per-step reward signal is revealed. We impose the following assumption on the link function Φ, which has appeared in many existing works of RLHF (Saha et al., 2023; Zhu et al., 2023; Zhan et al., 2023a).

Assumption 3.1. *We assume Φ is differentiable and there exists constants $\kappa, \overline{\kappa} > 0$ such that $\kappa^{-1} \le \Phi'(x) \le \overline{\kappa}^{-1}$ for any $x \in [-H, H]$.*

The constants κ and $\overline{\kappa}$ characterize the non-linearity of Φ and determine the difficulty of estimating the reward from preference feedback. It is noteworthy that, in the theoretical results of our algorithms, the bounds depend polynomially on κ but logarithmically on $\overline{\kappa}$. Some typical examples of the link functions are provided below.

Example 3.2 (Link functions)**.** *It is common to have $\Phi(x) = 1/(1+\exp(-x))$, which recovers the Bradley-Terry-Luce (BTL) model (Bradley & Terry, 1952), and we have $\kappa = 2+\exp(-H)+\exp(H)$ and $\overline{\kappa} = 4$. Additionally, if the trajectory-wise reward is scaled within the interval of $[0,1]$, then the difference in reward will be within the range of $[-1,1]$. In this case, another common choice of the link function is $\Phi(x) = (x+1)/2$, which results in $\kappa = \overline{\kappa} = 2$.*

The goal is to minimize the worst-case regret and the query complexity simultaneously:

$$\mathrm{Regret}_T \coloneqq \sum_{t=1}^T \Big(2V^\star(s_1) - V^{\pi_t^0}(s_1) - V^{\pi_t^1}(s_1)\Big), \quad \mathrm{Queries}_T \coloneqq \sum_{t=1}^T Z_t.$$

Here $V^\pi(s_1) \coloneqq \mathbb{E}_\pi[\sum_{h=1}^H r^\star(s_h, a_h)]$ denotes the state-value function of policy π, and we define $V^\star(s_1) \coloneqq V^{\pi^\star}(s_1)$ where $\pi^\star$ is the optimal policy that maximizes the state-value function. The variable $Z_t \in \{0,1\}$ indicates whether a query is made at round t. Note that the regret looks at the sum of the performance gaps between two pairs of policies: $(\pi^\star, \pi_t^0)$ and $(\pi^\star, \pi_t^1)$. This is standard in dueling bandits (Yue & Joachims, 2011; Yue et al., 2012; Dudík et al., 2015; Bengs et al., 2022; Wu et al., 2023b) and RL with preference-based feedback (Saha et al., 2023; Chen et al., 2022).

Bayesian RL. We also consider Bayesian RL in this work when learning with general function approximation. In the Bayesian setting, $P^\star$ and $r^\star$ are sampled from some known prior distributions ρ_{P} and ρ_{r}. The goal is to minimize the Bayesian regret and the Bayesian query complexity:

$$\mathrm{BayesRegret}_T \coloneqq \mathbb{E}\left[\sum_{t=1}^T \Big(2V^\star(s_1) - V^{\pi_t^0}(s_1) - V^{\pi_t^1}(s_1)\Big)\right], \ \mathrm{BayesQueries}_T \coloneqq \mathbb{E}\left[\sum_{t=1}^T Z_t\right].$$

Here the expectation is taken with respect to the prior distribution over $P^\star$ and $r^\star$. We will use Bayesian supervised learning oracles to compute posteriors over the transition and reward model.

4 A MODEL-FREE RANDOMIZED ALGORITHM FOR LINEAR MDPS

In this section, we present a model-free algorithm for linear MDPs which is defined as follows.

Assumption 4.1 (Linear MDP (Jin et al., 2020))**.** *We assume a known feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, an unknown (signed) measure $\mu : \mathcal{S} \to \mathbb{R}^d$, and an unknown vector $\theta_{\mathrm{r}}^\star$ such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have $P^\star(s' \mid s,a) = \phi^\top(s,a) \cdot \mu(s')$ and $r^\star(s,a) = \phi^\top(s,a) \cdot \theta_{\mathrm{r}}^\star$. We assume $\|\phi(s,a)\|_2 \le 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $\int_{\mathcal{S}} \|\mu(s)\|_2 \,\mathrm{d}s \le \sqrt{d}$, and $\|\theta_{\mathrm{r}}^\star\|_2 \le B$ for some $B > 0$. For a trajectory $\tau = (s_1, a_1, \ldots, s_H, a_H)$, we define $\phi(\tau) = \sum_{h=1}^H \phi(s_h, a_h)$ and assume $\|\phi(\tau)\|_2 \le 1$.*

Linear MDPs can capture tabular MDPs by setting $d = |\mathcal{S}||\mathcal{A}|$ and $\phi(s,a)$ to be the one-hot encoding of (s,a). In this case, we have $\|\phi(\tau)\|_2 \le H$. However, we can scale it down to get $\|\phi(\tau)\|_2 \le 1$ at the expense of scaling B up by H. We define $\Theta_B = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \le B\}$, which contains $\theta_{\mathrm{r}}^\star$.

4.1 ALGORITHM

The algorithm, called PR-LSVI, is presented in Algorithm 1. At the beginning of episode k, it first computes the maximum likelihood estimate $\widehat{\theta}_{\mathrm{r},t}$ (Line 3). Computation-wise, while the likelihood objective is not guaranteed to be concave due to the generality of Φ, efficient algorithms exist in certain common scenarios. For example, if $\Phi(x) = 1/(1 + \exp(-x))$, it recovers the BTL model (Example 3.2). In this case, the MLE objective is concave in θ and thus can be solved in polynomial running time. Moreover, we emphasize that the reward is learned under trajectory-wise features, which is different from the standard RL setting where it is learned under state-action features.

Given the MLE $\widehat{\theta}_{\mathrm{r},t}$, it next samples $\overline{\theta}_{\mathrm{r},t}$ from a Gaussian distribution centered at $\widehat{\theta}_{\mathrm{r},t}$ (Line 4). Note that the covariance matrix Σ_{t-1}^{-1} uses trajectory-wise features (Line 16) which allows the randomized Gaussian vector to capture trajectory-wise uncertainty of the learned reward. The noise aims to encourage exploration. Then, it computes the least-squares estimate of the state-action value function $\widehat{\theta}_{\mathrm{P},t,h}$ for each $h \in [H]$ and samples $\overline{\theta}_{\mathrm{P},t,h}$ from a Gaussian distribution centered at $\widehat{\theta}_{\mathrm{P},t,h}$ (Lines 7-8). Similar to the reward model, the noise is added to the state-value function to encourage exploration. We then define the value function $\overline{Q}_{t,h}$ and $\overline{V}_{t,h}$ as

$$\overline{Q}_{t,h}(s,a) := \phi(s,a)^\top \overline{\theta}_{\mathrm{r},t} + \omega_{t,h}(s,a), \qquad \overline{V}_{t,h}(s) := \max_a \overline{Q}_{t,h}(s,a) \tag{1}$$

and the function $\omega : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$\omega_{t,h}(s,a) = \begin{cases} \phi(s,a)^\top \overline{\theta}_{\mathrm{P},t,h} & \text{if } \|\phi(s,a)\|_{\Sigma_{t-1,h}^{-1}} \le \alpha_{\mathrm{L}} \\ \rho(s,a)\Big(\phi(s,a)^\top \overline{\theta}_{\mathrm{P},t,h}\Big) + (1-\rho(s,a))(H-h) & \text{if } \alpha_{\mathrm{L}} < \|\phi(s,a)\|_{\Sigma_{t-1,h}^{-1}} \le \alpha_{\mathrm{U}} \\ H-h & \text{if } \|\phi(s,a)\|_{\Sigma_{t-1,h}^{-1}} > \alpha_{\mathrm{U}} \end{cases}$$

where $\rho(s,a) = (\alpha_{\mathrm{U}} - \|\phi(s,a)\|_{\Sigma_{t-1,h}^{-1}})/(\alpha_{\mathrm{U}} - \alpha_{\mathrm{L}})$ interpolates between the two regimes to ensure continuity. This truncation trick is from Zanette et al. (2020) and is crucial. It controls the abnormally high value estimates. Specifically, when $\|\phi(s,a)\|_{\Sigma_{t-1,h}^{-1}}$ is large, the uncertainty in the direction of $\phi(s,a)$ is large, which makes the estimate $\phi(s,a)^\top \overline{\theta}_{\mathrm{P},t,h}$ abnormally large. In this case, we have to truncate it to $H - h$. Moreover, we note that the usual "value clipping" trick (i.e., simply constraining the value function within the range of $[0, H - h + 1]$ by clipping) cannot easily work here since it introduces bias to the random walk analysis, also pointed out by Zanette et al. (2020).

Then, the algorithm computes the greedy policy π_t^0 with respect to $\overline{Q}_{t,h}$. The comparator policy π_t^1 is simply set to the greedy policy from the previous episode, π_{t-1}^0 . In other words, we are comparing the two most recent greedy policies. This is different from previous work, which compares the current greedy policy with a fixed comparator (Wang et al., 2023). Analytically, for our algorithm, the cumulative regret incurred by π_t^1 for all $t \in [T]$ is equivalent to that incurred by π_t^0 for all $t \in [T]$. Hence, it suffices to compute the regret for one of them and multiply it by two to get the total regret.

Given the trajectories τ_t^0 and τ_t^1 generated by π_t^0 and π_t^1, we compute the *expected absolute reward difference* between the trajectories under the same noisy distribution of the reward parameter:

$$\mathop{\mathbb{E}}_{\theta_0,\theta_1 \sim \mathcal{N}(\widehat{\theta}_{\mathrm{r},t}, \sigma_{\mathrm{r}}^2 \Sigma_{t-1}^{-1})} \Big[\big| (\phi(\tau_t^0) - \phi(\tau_t^1))^\top (\theta_0 - \theta_1) \big| \Big]. \tag{2}$$

This represents the uncertainty of the preference between the two trajectories, and we make a query only when it is larger than a threshold ϵ (Line 13). Intuitively, we only make a query on two trajectories when we are uncertain about the preference (e.g., the expected disagreement between two randomly sampled reward models is large). Computationally, we can estimate this expectation by drawing polynomially many reward models from the distribution $\mathcal{N}(\widehat{\theta}_{\mathrm{r},t}, \sigma_{\mathrm{r}}^2 \Sigma_{t-1}^{-1})$ and computing the empirical average. The deviation of the empirical average to the true mean can be easily bounded by standard concentration inequalities. We simply use expectation here for analytical simplicity. If the query condition is triggered, we make a query for feedback on τ_t^0, τ_t^1, and update the trajectory-wise feature covariance matrix accordingly.

4.2 ANALYSIS

The theoretical results of Algorithm 1 are stated in Theorem 4.2. The detailed assignment of hyperparameters can be found in Table 1, and the proof is provided in Appendix B.

Theorem 4.2. *Define* $\gamma = \sqrt{\kappa + B^2}$*, which characterizes the difficulty of estimating the reward model. Set* $\sigma_{\mathrm{r}} = \widetilde{\Theta}(\gamma\sqrt{d})$, $\sigma_{\mathrm{P}} = \widetilde{\Theta}(H^{3/2}d^2\gamma)$, $\alpha_{\mathrm{U}} = (d^{5/2}H^{3/2}\gamma)^{-1}$, $\alpha_{\mathrm{L}} = \alpha_{\mathrm{U}}/2$*, and* $\lambda = 1$.

Algorithm 1 Preference-based and Randomized Least-Squares Value Iteration (PR-LSVI)

```
Require: STD $\sigma_\mathrm{r}, \sigma_\mathrm{P}$, threshold $\epsilon$, value cutoff parameters $\alpha_\mathrm{L}, \alpha_\mathrm{U}$, and regularization parameter $\lambda$.
Let $\pi_0^0$ be an arbitrary policy, $\Sigma_0 \leftarrow \lambda I$, $\Sigma_{0,h} \leftarrow \lambda I$ $(\forall h \in [H])$.
for $t = 1, \ldots, T$ do
    $\widehat{\theta}_{\mathrm{r},t} \leftarrow \arg\max_{\theta\in\Theta_B} \sum_{s=1}^{t-1} Z_s \ln(o_s\Phi((\phi(\tau_s^1)-\phi(\tau_s^0))^\top\theta)+(1-o_s)\Phi((\phi(\tau_s^0)-\phi(\tau_s^1))^\top\theta))$
    $\overline{\theta}_{\mathrm{r},t} \sim \mathcal{N}(\widehat{\theta}_{\mathrm{r},t}, \sigma_\mathrm{r}^2\Sigma_{t-1}^{-1})$
    $\widehat{\theta}_{\mathrm{P},t,H} \leftarrow 0$, $\overline{\theta}_{\mathrm{P},t,H} \leftarrow 0$
    for $h = H-1, \ldots, 1$ do
        $\widehat{\theta}_{\mathrm{P},t,h} \leftarrow \Sigma_{t-1,h}^{-1}(\sum_{i=1}^{t-1}\phi(s_{i,h}^0, a_{i,h}^0)\overline{V}_{t,h+1}(s_{i,h+1}^0))$
        $\overline{\theta}_{\mathrm{P},t,h} \sim \mathcal{N}(\widehat{\theta}_{\mathrm{P},t,h}, \sigma_\mathrm{P}^2\Sigma_{t-1,h}^{-1})$
        Define $\overline{Q}_{t,h}$ and $\overline{V}_{t,h}$ as in (1).
    end for
    Set $\pi_t^0 \leftarrow \{\pi_{t,h}^0 : \pi_{t,h}^0(s) = \arg\max_a \overline{Q}_{t,h}(s,a), \forall s\in\mathcal{S}, h\in[H]\}$ and $\pi_t^1 \leftarrow \pi_{t-1}^0$.
    Sample $\tau_t^0 \sim \pi_t^0$ and $\tau_t^1 \sim \pi_t^1$.
    $Z_t \leftarrow \mathbb{1}\{\mathbb{E}_{\theta_0,\theta_1\sim\mathcal{N}(\widehat{\theta}_{\mathrm{r},t},\sigma_\mathrm{r}^2\Sigma_{t-1}^{-1})}[|(\phi(\tau_t^0)-\phi(\tau_t^1))^\top(\theta_0-\theta_1)|] > \epsilon\}$
    if $Z_t = 1$ then
        Query preference feedback $o_t$ on $\{\tau_t^0, \tau_t^1\}$
        $\Sigma_t \leftarrow \Sigma_{t-1} + (\phi(\tau_t^0)-\phi(\tau_t^1))(\phi(\tau_t^0)-\phi(\tau_t^1))^\top$
    else
        $\Sigma_t \leftarrow \Sigma_{t-1}$
    end if
    $\Sigma_{t,h} \leftarrow \Sigma_{t-1,h} + \phi(s_{t,h}^0, a_{t,h}^0)\phi^\top(s_{t,h}^0, a_{t,h}^0)$ $(\forall h\in[H])$.
end for
```

Then, PR-LSVI (Algorithm 1) guarantees the following with probability at least $1-\delta$*:*

$$\mathrm{Regret}_T = \widetilde{O}\left(\epsilon T d^{1/2} + \sqrt{T}\cdot d^3 H^{5/2}\gamma + d^{17/2}H^{11/2}\gamma^3\right), \quad \mathrm{Queries}_T = \widetilde{O}\left(d^4\gamma^4/\epsilon^2\right).$$

To further study the balance between the regret and the query complexity, we let $\epsilon = T^{-\beta}$ for some $\beta \le 1/2$. Then, the upper bounds in Theorem 4.2 can be rewritten as

$$\mathrm{Regret}_T = \widetilde{O}(T^{1-\beta}), \quad \mathrm{Queries}_T = \widetilde{O}(T^{2\beta})$$

where we only focus on the dependence on T and omit any other factors for simplicity. We see that there is a tradeoff in T between the regret and the query complexity — the smaller regret we want, the more queries we need to make. For example, when $\beta = 0$, the regret is $\widetilde{O}(T)$, and the query complexity is $\widetilde{O}(1)$, meaning that we will incur linear regret if we don't make any query. If we increase β to $1/2$, the regret decreases to $\widetilde{O}(\sqrt{T})$ while the query complexity increases to $\widetilde{O}(T)$, meaning that the regret bound is optimal in T but we make queries every episode.

We emphasize that this tradeoff in T is *optimal*, as evidenced by a lower bound result established by Sekhari et al. (2023a). Their lower bound was originally proposed for contextual dueling bandits, which is a special case of our setting. Their results are stated below.

Theorem 4.3. *(Sekhari et al., 2023a, Theorem 5) The following two claims hold: (1) For any algorithm, there exists an instance that leads to* $\mathrm{Regret}_T = \Omega(\sqrt{T})$*; (2) For any algorithm achieving an expected regret upper bound in the form of* $\mathbb{E}[\mathrm{Regret}_T] = O(T^{1-\beta})$ *for some* $\beta > 0$*, there exists an instance that results in* $\mathbb{E}[\mathrm{Queries}_T] = \Omega(T^{2\beta})$*.*

However, the dependence on other parameters (e.g., d and H) can be loose, and further improvement may be possible. We leave further investigation of these factors as future work.

Although injecting random noise is inspired by RLSVI (Zanette et al., 2020), we highlight five key differences between ours and theirs: (1) Since the feedback is trajectory-wise, we need to design random noise that preserves the state-action-wise format (so that it can be used in DP) but captures the trajectory-wise uncertainty. We do this by maintaining Σ_t, which uses trajectory-wise feature differences; (2) Since the preference feedback is generated from some probabilistic model, we learn the reward model via MLE and use MLE generalization bound (Geer, 2000) to capture the uncertainty in learning. This allows us to use a more general link function Φ; (3) We design a new regret decomposition technique to accommodate preference-based feedback. Particularly, we decompose regret to characterizes the *reward difference* between π_t^0 and π_t^1: $\mathrm{Regret}_T \lesssim \sum_{t=1}^T (\overline{V}_t - \widetilde{V}_t) - (V^{\pi_t^0} - V^{\pi_t^1})$ where $\overline{V}_t$ is an estimate of $V^{\pi_t^0}$, and $\widetilde{V}_t := \mathbb{E}_{\tau\sim\pi_t^1}[\sum_{h=1}^H \phi(s_h, a_h)^\top \overline{\theta}_{\mathrm{r},t}]$ is an estimate of $V^{\pi_t^1}$ under the real transition and the learned reward model. This is different from standard RL (Zanette et al., 2020), and is necessary since we cannot guarantee the learned reward model will be accurate in a state-action-wise manner under the preference-based feedback. (4) Our algorithms have a new randomized active learning procedure for reducing the number of queries, and our analysis achieves a near-optimal tradeoff between regret and query complexity; (5) In every round t, we propose to draw a pair of trajectories where one is from the current greedy policy π_t^0 and the other is from the greedy policy of the previous round, π_{t-1}^0. This ensures π_t^1 is conditionally independent of the Gaussian noises at round t, which is the key to optimism (with a constant probability).

Running time. To assess the time complexity of Algorithm 1, assuming finite number of actions[2], all steps can be computed in polynomial running time (i.e., polynomial in d, H, A) except the MLE of the reward model (Line 3), which depends on the link function Φ. For the popular BTL model where $\Phi(x) = 1/(1 + \exp(-x))$, the MLE objective is concave with respect to θ and θ belongs to a convex set Θ_B. In this case, we can use any convex programming algorithms for the MLE procedure (e.g., projected gradient ascent).

5 A MODEL-BASED THOMPSON SAMPLING ALGORITHM

In this section, we aim to extend to nonlinear function approximation. We do so in a model-based framework with Thompson sampling (TS). The motivation is that TS is often considered a computationally more tractable alternative to UCB-style algorithms.

5.1 ALGORITHM

The algorithm, called PbTS, is presented in Algorithm 2. At the beginning of episode k, it computes the reward model posterior $\rho_{\mathrm{r},t}$ and the transition model posterior $\rho_{\mathrm{P},t}$ (Line 3). Then, it samples P_t and r_t from the posteriors and computes the optimal policy π_t^0 assuming the true reward function is r_t and the true model is P_t (Line 5). Here we denote $V_{r,P}^{\pi}$ as the state-value function of π under reward function r and model P. Note that this oracle is a standard planning oracle. The comparator policy π_t^1 is simply set to be the policy from the previous episode, π_{t-1}^0, as we did in Algorithm 1. The two policies then generate respective trajectories τ_t^0 and τ_t^1. To decide whether we should make a query, we compute the uncertainty quantity under the posterior distribution of the reward: $\mathbb{E}_{r,r'\sim\rho_{\mathrm{r},t}}[|r(\tau_t^0) - r(\tau_t^1) - (r'(\tau_t^0) - r'(\tau_t^1))|]$, which is analogous to (2) in Algorithm 1. We make a query only when it is larger than a threshold ϵ. Similar to Algorithm 1, we can approximate this expectation by sampling polynomial many pairs of r and r' and then compute the empirical average.

5.2 ANALYSIS

The theoretical results of Algorithm 2 should rely on the complexity of the reward and the transition model. In our analysis, we employ two complexity measures — eluder dimension and bracketing number. We start by introducing a generic notion of ℓ_p-eluder dimension (Russo & Van Roy, 2013).

Definition 5.1 (ℓ_p-norm ϵ-dependence). *Let $p > 0$. Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets and $d(\cdot,\cdot)$ be a distance function on $\mathcal{Y}$. Let $\mathcal{F} \subseteq \mathcal{X} \to \mathcal{Y}$ be a function class. We say an element $x \in \mathcal{X}$ is ℓ_p-norm ϵ-dependent on $\{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{X}$ with respect to $\mathcal{F}$ and d if any pair of functions $f, f' \in \mathcal{F}$ satisfying $\sum_{i=1}^n d^p(f(x_i), f'(x_i)) \leq \epsilon^p$ also satisfies $d(f(x), f'(x)) \leq \epsilon$. Otherwise, we say x is ℓ_p-norm ϵ-independent of $\{x_1, x_2, \ldots, x_n\}$.*

Definition 5.2 (ℓ_p-norm eluder dimension). *The ℓ_p-norm ϵ-eluder dimension of function class $\mathcal{F} \subseteq \mathcal{X} \to \mathcal{Y}$, denoted by $\dim_p(\mathcal{F}, \epsilon, d)$, is the length of the longest sequence of elements in $\mathcal{X}$ satisfying that there exists $\epsilon' \geq \epsilon$ such that every element in the sequence is ℓ_p-norm ϵ'-independent of its predecessors.*

---

[2] This is to ensure that $\arg\max_a Q(s, a)$ can be computed efficiently.

Algorithm 2 Preference-based Thompson Sampling (PbTS)

```
Require: priors $\rho_{\mathrm{P}}$ and $\rho_{\mathrm{r}}$, threshold $\epsilon$.
Let $\pi_0^0$ be an arbitrary policy.
for $t = 1, \ldots, T$ do
    Compute posteriors:
        $$\rho_{\mathrm{P},t}(P) \propto \rho_{\mathrm{P}}(P) \prod_{i=1}^{t-1} \prod_{h=1}^{H} P(s_{i,h+1}^0 \mid s_{i,h}^0, a_{i,h}^0),$$
        $$\rho_{\mathrm{r},t}(r) \propto \rho_{\mathrm{r}}(r) \prod_{i=1}^{t-1} \Big( o_i \Phi\big(r(\tau_i^1) - r(\tau_i^0)\big) + (1 - o_i) \Phi\big(r(\tau_i^0) - r(\tau_i^1)\big) \Big)^{Z_i}.$$
    Sample $P_t \sim \rho_{\mathrm{P},t}$ and $r_t \sim \rho_{\mathrm{r},t}$.
    Compute $\pi_t^0 \leftarrow \arg\max_\pi V_{r_t,P_t}^{\pi}(s_1)$ and $\pi_t^1 \leftarrow \pi_{t-1}^0$.
    Sample $\tau_t^0 \sim \pi_t^0$ and $\tau_t^1 \sim \pi_t^1$.
    $Z_t \leftarrow \mathbb{1}\{\mathbb{E}_{r,r' \sim \rho_{\mathrm{r},t}}[|r(\tau_t^0) - r(\tau_t^1) - (r'(\tau_t^0) - r'(\tau_t^1))|] > \epsilon\}$
    if $Z_t = 1$ then
        Query preference feedback $o_t$ on $\{\tau_t^0, \tau_t^1\}$
    end if
end for
```

The eluder dimension is non-decreasing in p, i.e., $\dim_p(\mathcal{F}, \epsilon, d) \le \dim_q(\mathcal{F}, \epsilon, d)$ for any $p \le q$. In the analysis, we will focus on ℓ_1- and ℓ_2-norm eluder dimension, which have been used in nonlinear bandits and RL extensively (Wen & Van Roy, 2013; Osband & Van Roy, 2014a; Jain et al., 2015; Wang et al., 2020; Ayoub et al., 2020; Foster et al., 2021; Ishfaq et al., 2021; Chen et al., 2022; Liu et al., 2022a; Sekhari et al., 2023a;b). Examples where eluder dimension is small include linear functions, generalized linear models, and functions in Reproducing Kernel Hilbert Space (RKHS).

The other complexity measure we use is the bracketing number (Van de Geer, 2000).

Definition 5.3 (Bracketing number)**.** *Consider a function class $\mathcal{F} \subseteq \mathcal{X} \to \mathbb{R}$. Given two functions $l, u : \mathcal{X} \to \mathbb{R}$, the bracket $[l, u]$ is defined as the set of functions $f \in \mathcal{F}$ with $l(x) \le f(x) \le u(x)$ for all $x \in \mathcal{X}$. It is called an ω-bracket if $\|l - u\| \le \omega$. The bracketing number of $\mathcal{F}$ w.r.t. the metric $\|\cdot\|$, denoted by $N_{[]}(\omega, \mathcal{F}, \|\cdot\|)$, is the minimum number of ω-brackets needed to cover $\mathcal{F}$.*

The logarithm of the bracketing number is small in many common scenarios, which has been extensively examined by previous studies (e.g., Van de Geer (2000)) for deriving MLE generalization bound (Agarwal et al., 2020; Uehara & Sun, 2021; Liu et al., 2022b; 2023). For example, when $\mathcal{F}$ is finite, the bracketing number is bounded by its size. When $\mathcal{F}$ is a d-dimensional linear function class, the logarithm of the bracketing number is upper bounded by d up to logarithmic factors.

It is worth noting that while we will employ both measures to the model class $\mathcal{P}$, we can not similarly apply them to the reward class $\mathcal{R}$. Instead, we have to rely on the complexity of the following function class, which comprises functions mapping pairs of trajectories to reward differences:

$$\widetilde{\mathcal{R}} := \left\{ \widetilde{r} \,:\, \widetilde{r}(\tau^0, \tau^1) = \sum_{h=1}^{H} r(s_h^0, a_h^0) - r(s_h^1, a_h^1),\ \forall \tau^i = \{s_h^i, a_h^i\}_h, i \in \{0, 1\}, r \in \mathcal{R} \right\}. \tag{3}$$

We have to use $\widetilde{\mathcal{R}}$ instead of $\mathcal{R}$ because we only receive preference feedback, and thus we cannot guarantee that the learned reward model is accurate state-action-wise. Now we are ready to state our main results. The proofs are provided in Appendix C.

Theorem 5.4. *PbTS (Algorithm 2) guarantees that*

$$\mathrm{BayesRegret}_T = \widetilde{O}\Big(T\epsilon + H^2 \cdot \dim_1\big(\mathcal{P}, 1/T\big) \cdot \sqrt{T \cdot \iota_{\mathcal{P}}} + \kappa \cdot \dim_1\big(\widetilde{\mathcal{R}}, 1/T\big) \cdot \sqrt{T \cdot \iota_{\mathcal{R}}} \Big),$$

$$\mathrm{BayesQueries}_T = \widetilde{O}\left(\min\left\{ \frac{\kappa\sqrt{T \cdot \iota_{\mathcal{R}}}}{\epsilon} \cdot \dim_1\big(\widetilde{\mathcal{R}}, \epsilon/2\big), \frac{\kappa^2 \cdot \iota_{\mathcal{R}}}{\epsilon^2} \cdot \dim_2\big(\widetilde{\mathcal{R}}, \epsilon/2\big) \right\} \right)$$

where we denote $\iota_{\mathcal{P}} := \log(N_{[]}((HT|\mathcal{S}|)^{-1}, \mathcal{P}, \|\cdot\|_\infty))$ and $\iota_{\mathcal{R}} := \log(N_{[]}(\overline{\kappa}(2T)^{-1}, \widetilde{\mathcal{R}}, \|\cdot\|_\infty))$.

Similar to the analysis of Algorithm 1, we study the balance between the Bayesian regret and the query complexity by setting $\epsilon = T^{-\beta}$ for some $\beta \leq 1/2$. Then, we can simplify the bounds into $\mathrm{BayesRegret}_T = \widetilde{O}(T^{1-\beta})$ and

$$\mathrm{BayesQueries}_T = \widetilde{O}\Bigg(\min\Bigg\{ \underbrace{T^{\beta+\frac{1}{2}} \cdot \dim_1\Big(\widetilde{\mathcal{R}}, \epsilon/2\Big)}_{\text{(i)}},\ \underbrace{T^{2\beta} \cdot \dim_2\Big(\widetilde{\mathcal{R}}, \epsilon/2\Big)}_{\text{(ii)}} \Bigg\}\Bigg)$$

where we have hidden factors except T and the eluder dimension for brevity. We see that there is again a tradeoff in T between the Bayesian regret and the query complexity, similar to the one in Theorem 4.2. Term (ii) demonstrates that the tradeoff in T is again *optimal*, evidenced by the lower bound (Theorem 4.3). Moreover, term (i) further improves the dependence on the eluder dimension (recalling that ℓ_1-norm version is smaller than the ℓ_2-norm version). However, the T-dependence is worse. It is desired to derive a query complexity upper bound that scales as $\widetilde{O}(T^{2\beta} \cdot \dim_1(\widetilde{\mathcal{R}}, \epsilon/2))$, attaining the favorable dependence on both T and the eluder dimension. We leave it as future work.

We emphasize that the Bayesian regret analysis in Theorem 5.4 is not a simple extension of previous TS works. We highlight four main differences: (1) The feedback is preference-based, which necessitates a new Bayesian regret decomposition:

$$\mathrm{BayesRegret}_T = \underbrace{\sum_{t=0}^{T} \mathbb{E}\left[V^{\pi_t^0}_{r_t,P_t} - V^{\pi_t^0}_{r_t,P^\star} \right]}_{\mathtt{T}_{\mathrm{model}}} + \underbrace{\sum_{t=0}^{T} \mathbb{E}\left[\left(V^{\pi_t^0}_{r_t,P^\star} - V^{\pi_t^1}_{r_t,P^\star} \right) - \left(V^{\pi_t^0}_{r^\star,P^\star} - V^{\pi_t^1}_{r^\star,P^\star} \right) \right]}_{\mathtt{T}_{\mathrm{reward}}}.$$

Here $\mathtt{T}_{\mathrm{model}}$ and $\mathtt{T}_{\mathrm{reward}}$ are the respective regret incurred due to model and reward misspecification. We highlight that $\mathtt{T}_{\mathrm{reward}}$ characterizes the misspecification in terms of the *reward difference* between π_t^0 and π_t^1, which is different from the standard Bayesian RL. (2) Unlike prior works (Russo & Van Roy, 2014), we do not rely on upper confidence bounds (UCB) or optimism. Instead, we construct version spaces by classic MLE generalization bound. Taking the reward learning as an example, given the preference data $\{\tau_i^0, \tau_i^1, o_i\}_{i=1}^{t-1}$, we construct the version space at round t as

$$\mathcal{V}_t = \left\{ r \in \mathcal{R} \ :\ \sum_{i=1}^{t-1} d^2_{\mathrm{TV}}\left(\Pr(\cdot \mid \tau_i^1, \tau_i^0, \widehat{r}_t),\, \Pr(\cdot \mid \tau_i^1, \tau_i^0, r) \right) \leq \beta \right\}$$

where $\widehat{r}_t := \arg\max_r \log \sum_{i=1}^{t-1} \Pr(o_i \mid \tau_i^1, \tau_i^0, r)$ is the MLE from the preference data and β is tuned appropriately to ensure $r^\star \in \mathcal{V}_t$ with high probability. We then show the posterior probability of r_t and $r^\star$ not belonging to $\mathcal{V}_t$ is small. (3) Our analysis uses the tighter ℓ_1-norm eluder dimension, which is strictly better than the ℓ_2-norm eluder dimension used in prior work. (4) We also equipped it with a randomized active learning procedure for query complexity minimization.

Computation. The computational bottleneck of Algorithm 2 lies in the computation of the posterior distribution (Line 3). Prior TS works have used Bootstrapping to approximate posterior sampling (Osband et al., 2016a; 2023) and achieved competitive performance in common RL benchmarks.

Non-Markovian reward. Algorithm 2 can also be applied to non-Markovian reward (i.e., reward model is trajectory-wise) without any change. Here we consider Markovian reward for the consistency with Algorithm 1 and for the purpose of using a standard planning oracle for computing an optimal policy from a reward and transition model. While non-Markovian reward is more general, it is unclear how to solve the planning problem efficiently even in tabular MDPs. This computational intractability makes non-Markovian rewards not easily applicable in practice.

Extension to SEC. In Appendix C.4, we extend the eluder dimension in Theorem 5.4 to the Sequential Extrapolation Coefficient (SEC) (Xie et al., 2022), which is more general.

6 CONCLUSION

We use randomization to design algorithms for RL with preference-based feedback. Randomization allows us to minimize regret and query complexity while at the same time maintaining computation efficiency. For linear models, our algorithms achieve a near-optimal balance between the worst-case reward regret and query complexity with computational efficiency. For models beyond linear, using eluder dimension, we present a TS-inspired algorithm that balances Bayesian regret and Bayesian query complexity nearly optimally.

ACKNOWLEDGEMENTS

Both RW and WS acknowledge support from NSF IIS-2154711, NSF CAREER 2339395, and Cornell Infosys Collaboration.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Marc Abeille and Alessandro Lazaric. Linear thompson sampling revisited. In *Artificial Intelligence and Statistics*, pp. 176–184. PMLR, 2017.

Alekh Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *International Conference on Machine Learning*, pp. 1220–1228. PMLR, 2013.

Alekh Agarwal and Tong Zhang. Model-based rl with optimistic posterior sampling: Structural conditions and sample complexity. *Advances in Neural Information Processing Systems*, 35: 35284–35297, 2022.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.

Priyank Agrawal, Jinglin Chen, and Nan Jiang. Improved worst-case regret bounds for randomized least-squares value iteration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6566–6573, 2021.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. *Advances in Neural Information Processing Systems*, 30, 2017.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pp. 463–474. PMLR, 2020.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272. PMLR, 2017.

Viktor Bengs, Aadirupa Saha, and Eyke Hüllermeier. Stochastic contextual dueling bandits under linear stochastic transitivity models. In *International Conference on Machine Learning*, pp. 1764–1786. PMLR, 2022.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, 2005.

Niladri Chatterji, Aldo Pacchiano, Peter Bartlett, and Michael Jordan. On the theory of reinforcement learning with once-per-episode feedback. *Advances in Neural Information Processing Systems*, 34:3401–3412, 2021.

Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, pp. 3773–3793. PMLR, 2022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple experts. *The Journal of Machine Learning Research*, 13(1):2655–2697, 2012.

Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pp. 2826–2836. PMLR, 2021.

Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Conference on Learning Theory*, pp. 563–587. PMLR, 2015.

Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7288–7295, 2021.

Dylan Foster, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. In *Conference on Learning Theory*, pp. 2059–2059. PMLR, 2021.

Sara A Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized markov decision processes. In *Conference on Learning Theory*, pp. 861–898. PMLR, 2015.

Steve Hanneke and Liu Yang. Minimax analysis of active learning. *J. Mach. Learn. Res.*, 16(1): 3487–3602, 2015.

Steve Hanneke and Liu Yang. Toward a general theory of online selective sampling: Trading off mistakes and queries. In *International Conference on Artificial Intelligence and Statistics*, pp. 3997–4005. PMLR, 2021.

Haque Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin Yang. Randomized exploration in reinforcement learning with general value function approximation. In *International Conference on Machine Learning*, pp. 4607–4616. PMLR, 2021.

Ashesh Jain, Brian Wojcik, Thorsten Joachims, and Ashutosh Saxena. Learning trajectory preferences for manipulators via iterative improvement. *Advances in neural information processing systems*, 26, 2013.

Ashesh Jain, Shikhar Sharma, Thorsten Joachims, and Ashutosh Saxena. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research*, 34(10):1296–1313, 2015.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

Michael Laskey, Sam Staszak, Wesley Yu-Shu Hsieh, Jeffrey Mahler, Florian T Pokorny, Anca D Dragan, and Ken Goldberg. Shiv: Reducing supervisor burden in dagger using support vectors for efficient learning from demonstrations in high dimensional state spaces. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 462–469. IEEE, 2016.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Qinghua Liu, Alan Chung, Csaba Szepesvári, and Chi Jin. When is partially observable reinforcement learning not scary? In *Conference on Learning Theory*, pp. 5175–5220. PMLR, 2022a.

Qinghua Liu, Csaba Szepesvári, and Chi Jin. Sample-efficient reinforcement learning of partially observable markov games. *Advances in Neural Information Processing Systems*, 35:18296–18308, 2022b.

Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. Optimistic mle: A generic model-based algorithm for partially observable sequential decision making. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 363–376, 2023.

Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1029–1038. PMLR, 2020.

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. *Advances in Neural Information Processing Systems*, 27, 2014a.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. *Advances in Neural Information Processing Systems*, 27, 2014b.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016a.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pp. 2377–2386. PMLR, 2016b.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Approximate thompson sampling via epistemic neural networks. *arXiv preprint arXiv:2302.09205*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

David Pollard. Empirical processes: theory and applications. Ims, 1990.

Nikita Puchkin and Nikita Zhivotovskiy. Exponential savings in agnostic active learning through abstention. In *Conference on Learning Theory*, pp. 3806–3832. PMLR, 2021.

Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert. Learning monocular reactive uav control in cluttered natural environments. In *2013 IEEE international conference on robotics and automation*, pp. 1765–1772. IEEE, 2013.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling rl: Reinforcement learning with trajectory preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 6263–6289. PMLR, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Contextual bandits and imitation learning via preference-based active queries. *arXiv preprint arXiv:2307.12926*, 2023a.

Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Selective sampling and imitation learning via online regression. *arXiv preprint arXiv:2307.04998*, 2023b.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations*, 2021.

Sara Van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is rlhf more difficult than standard rl? *arXiv preprint arXiv:2306.14111*, 2023.

Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. *Advances in Neural Information Processing Systems*, 26, 2013.

Runzhe Wu, Masatoshi Uehara, and Wen Sun. Distributional offline policy evaluation with predictive error guarantees. In *International Conference on Machine Learning*, pp. 37685–37712. PMLR, 2023a.

Yue Wu, Tao Jin, Hao Lou, Farzad Farnoud, and Quanquan Gu. Borda regret minimization for generalized linear dueling bandits. *arXiv preprint arXiv:2303.08816*, 2023b.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. *arXiv preprint arXiv:2210.04157*, 2022.

Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 241–248. Citeseer, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Andrea Zanette, David Brandfonbrener, Emma Brunskill, Matteo Pirotta, and Alessandro Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964. PMLR, 2020.

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D Lee, and Wen Sun. Provable offline reinforcement learning with human feedback. *arXiv preprint arXiv:2305.14816*, 2023a.

Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D Lee. How to query human feedback efficiently in rl? *arXiv preprint arXiv:2305.18505*, 2023b.

Han Zhong, Wei Xiong, Sirui Zheng, Liwei Wang, Zhaoran Wang, Zhuoran Yang, and Tong Zhang. Gec: A unified framework for interactive decision making in mdp, pomdp, and beyond. *CoRR*, 2022.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*, 2023.

Yinglun Zhu and Robert Nowak. Efficient active learning with abstention. *Advances in Neural Information Processing Systems*, 35:35379–35391, 2022.