EFFICIENT REWARD-FREE REINFORCEMENT LEARNING WITH LINEAR FUNCTION APPROXIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study reward-free reinforcement learning with linear function approximation for episodic Markov decision processes (MDPs). In this setting, an agent first interacts with the environment without accessing the reward function in the exploration phase. Then, in the subsequent planning phase, it is given a reward function and supposed to output an ϵ-optimal policy. We propose a novel algorithm LSVI-RFE under the linear MDP setting, where the transition probability and reward functions are linear in a feature mapping $\boldsymbol{\phi}(s, a)$. We prove an $\widetilde{O}(H^4 d^2/\epsilon^2)$ sample complexity upper bound for LSVI-RFE, where H is the episode length and d is the feature dimension. We also establish a sample complexity lower bound of $\Omega(H^3 d^2/\epsilon^2)$. To the best of our knowledge, this is the first computationally efficient algorithm that achieves the minimax optimal sample complexity in linear MDP settings up to an H and logarithmic factors. Our LSVI-RFE algorithm is based on a novel *variance-aware* exploration mechanism, which dynamically adjusts the weight factors according to the episodic exploration error of the trajectory, and weighted linear regression, to avoid overly-conservative exploration in prior works. Our sharper bound relies on a novel analysis concerning the conservatism of elliptical potentials to remove the extra dependency of sample complexity on feature dimension d.

1 INTRODUCTION

In reinforcement learning (RL), an agent tries to learn an optimal policy that maximizes the cumulative long-term rewards by interacting with an unknown environment. Designing efficient exploration mechanisms, being a critical task in RL algorithm design, is of great significance in improving the sample efficiency of RL, both theoretically Azar et al. (2017); Ménard et al. (2021) and empirically Schwarzer et al. (2020); Ye et al. (2021). In particular, for scenarios where reward signals are sparse and require manually-designed reward functions, e.g., Nair et al. (2018); Riedmiller et al. (2018), or multi-task settings where RL agents are required to accomplish different goals in different stages, e.g., Hessel et al. (2019); Yang et al. (2020), efficient exploration of the environments is crucial, as it can save the RL models from being repeatedly trained under different reward functions, resulting in inefficiency and even intractability of sample complexity. Despite the importance of exploration in RL, its theoretical understanding is still limited, especially for MDPs with large (or infinite) states or action spaces.

To understand the exploration mechanism in RL, *reward-free exploration* (RFE) has been proposed in Jin et al. (2020a) to explore the environment without reward signals. RFE contains two phases, exploration and planning. In the exploration phase, the agent first interacts with the environment without accessing the reward function. Then, in the subsequent planning phase, the agent is given a reward function and asked to output an ϵ-optimal policy. RFE has great significance in a host of reinforcement learning applications, e.g., multi-task RL Hessel et al. (2019); Yang et al. (2020), RL with sparse rewards Nair et al. (2018); Riedmiller et al. (2018), and systematic generalization of RL Jiang et al. (2019); Mutti et al. (2022). The minimax optimal sample complexity $O(H^3 S^2 A/\epsilon^2)$ of RFE is obtained in Ménard et al. (2021) for tabular settings where S and A are sizes of state and action space, respectively.

In this paper, we consider the RFE problem in the linear MDP setting, where the transition probability and reward functions are linear in a feature mapping $\boldsymbol{\phi}(s, a)$. The linear MDP model is an

important formulation that provides a linear function approximation for general MDP problems with large (or infinite) state and action space, and has received much recent attention in RL study, e.g., Jin et al. (2020b); Zhou et al. (2021). Existing works in Wang et al. (2020a); Zanette et al. (2020c); Chen et al. (2021) also study RFE in linear MDPs, and best known sample complexity upper bound in this setting is $\widetilde{O}(H^4d^3/\epsilon^2)$ obtained in Chen et al. (2021), and the best known lower bound is $\Omega(\max\{H^3d, d^2\}/\epsilon^2)$ by combining results obtained in Zhang et al. (2021) and Wagenmaker et al. (2022). This means that the following fundamental question still remains open:

Does there exist a computation-efficient and minimax optimal algorithm for RFE in linear MDPs?

We make constructive contributions to the problem by proposing a computationally efficient algorithm LSVI-RFE, achieving a sample complexity upper bound of $\widetilde{O}(H^4d^2/\epsilon^2)$, which is minimax optimal up to an H and logarithmic factors. The LSVI-RFE algorithm is based on a novel *trajectory-aware* exploration mechanism with weighted linear regression to avoid overly-conservative exploration in prior works. In particular, we design a trajectory-aware exploration reward function that dynamically adjusts the weight factors according to the episodic exploration error of the trajectory. Accordingly, our sharper bound relies on a novel analysis concerning the conservatism of elliptical potentials to remove the extra dependency of sample complexity on feature dimension d. We summarize the main contributions of the paper below.

- We propose the LSVI-RFE algorithm for RFE in linear MDPs based on a novel trajectory-aware exploration bonus to avoid overly-conservative exploration in prior works. LSVI-RFE achieves an $\widetilde{O}(H^4d^2/\epsilon^2)$ sample complexity upper bound, which, to our best knowledge, is the sharpest upper bound of a computationally efficient algorithm for RFE in linear MDPs.
- We prove a tight sample complexity lower bound of $\Omega(H^3d^2/\epsilon^2)$ by constructing a hard MDP instance. This result nearly matches LSVI-RFE's complexity bound and shows that LSVI-RFE is the first computationally efficient algorithm to achieve the minimax optimal sample complexity in linear MDPs up to an H and logarithmic factors.
- Our sharper sample complexity bound comes from the conservatism of elliptical potentials to remove the extra dependency of sample complexity on d. This technique is general and has the potential to improve the extra dependency on d in the statistical complexity of algorithms for linear MDPs.

Notations Scalars are denoted in lower case letters, and vectors/matrices are denoted in boldface letters. Denote $\|\boldsymbol{x}\|_{\boldsymbol{\Lambda}}^2 = \boldsymbol{x}^\top \boldsymbol{\Lambda} \boldsymbol{x}$ for vector $\boldsymbol{x}$ and positive definite matrix $\boldsymbol{\Lambda}$. Denote $\{1, ..., n\}$ as $[n]$. $a_n = O(b_n)$ if there exists an absolute constant $c > 0$ such that $a_n \leqslant cb_n$ holds for all $n \geqslant 1$ and $a_n = \Omega(b_n)$ for inverse direction. Besides, $\widetilde{O}(\cdot)$ further suppresses the polylogarithmic factors in $O(\cdot)$ and $\lesssim$ denotes approximately less than up to constant factors.

2 Related Works

RL with Linear Function Approximation There are a number of ways to parameterize an MDP linearly. The first sample efficient algorithm is introduced by Jiang et al. (2017), where low Bellman rank MDPs are considered. Subsequent works include Dann et al. (2018); Sun et al. (2019). Yang & Wang (2019) develops the first statistically and computationally efficient algorithm for linear MDPs with generative models, while Jin et al. (2020b) considers RL settings and proposes the LSVI-UCB algorithm. Concurrently, Zanette et al. (2020a) provides a Thompson sampling based algorithm. He et al. (2021) provides first logarithmic gap dependent regret bounds for linear MDPs, and Wagenmaker et al. (2021) provides a first-order regret bound for linear MDPs with a Bernstein-type bonus. More works on RL with linear function approximation include Zanette et al. (2020b) for low inherent bellman error case, Wang et al. (2020c) for linear Q function case, and Wang et al. (2020b) for bounded Eluder dimension case. Another popular linearly parameterized MDP is the linear mixture MDP, studied in Modi et al. (2020); Yang & Wang (2020); Jia et al. (2020); Ayoub et al. (2020); Cai et al. (2020); Zhou et al. (2021).

Reward-Free Exploration in Tabular MDPs Reward-free exploration is studied in Jin et al. (2020a); Kaufmann et al. (2021); Ménard et al. (2021); Zhang et al. (2020) Their algorithms achieve an $\widetilde{O}(H^5S^2A/\epsilon^2)$ sample complexity in the tabular setting, where S is the number of states and A is the number of actions. Subsequently, the RF-UCRL algorithm in Kaufmann et al. (2021) improves

the result in Jin et al. (2020a) by an H factor. Ménard et al. (2021) modifies the Upper Confidence Bound (UCB)-bonus of UCRL and achieves a sample complexity of $\widetilde{O}(H^3S^2A/\epsilon^2)$, matching the minimax lower bound provided in Jin et al. (2020a) up to logarithmic factors. Concurrently, the SSTP algorithm Zhang et al. (2020) also achieves minimax optimality in the time-homogeneous setting. Wu et al. (2022) proposes the first gap dependent bound for this setting.

RFE with Linear Function Approximation There are recent works Wang et al. (2020a); Zanette et al. (2020c); Zhang et al. (2021); Chen et al. (2021); Wagenmaker et al. (2022) focusing on RFE in RL with linear function approximation, where the state-of-the-art upper bound of this setting is $\widetilde{O}(H^4d^3/\epsilon^2)$ obtained in Chen et al. (2021), and the best known lower bound is $\Omega(\max\{H^3d, d^2\}/\epsilon^2)$ by combining results obtained in Zhang et al. (2021) and Wagenmaker et al. (2022). There is an independent work of our paper that gives a sample complexity of $\widetilde{O}(H^5d^2/\epsilon^2)$ in Wagenmaker et al. (2022), which has an additional H factor compared to our our results. Another line of works Zhang et al. (2021); Chen et al. (2021) focus on RFE in linear mixture MDPs, where the minimax optimal sample complexity is obtained in Chen et al. (2021) when $d > H$ but the algorithm is not computationally-efficient. The detailed comparison of some most related works is shown in Table 1.[1]

Table 1: Comparison of reward-free exploration in episodic RL with linear function approximation.

Algorithm	Setting	Computation-Efficient	Sample Complexity
UCRL-RFE$^+$ Zhang et al. (2021)	Linear Mixture MDP	No	$\widetilde{O}(H^5d(H+d)/\epsilon^2)$
Chen et al. Chen et al. (2021)	Linear Mixture MDP	No	$\widetilde{O}(H^3d(H+d)/\epsilon^2)$
Lower bound Chen et al. (2021)	Linear Mixture MDP		$\Omega(H^3d^2/\epsilon^2)$
Wang et al. Wang et al. (2020a)	Linear MDP	Yes	$\widetilde{O}(H^6d^3/\epsilon^2)$
FRANCIS Zanette et al. (2020c)	Linear MDP	Yes	$\widetilde{O}(H^5d^3/\epsilon^2)$
Chen et al. Chen et al. (2021)	Linear MDP	Yes	$\widetilde{O}(H^4d^3/\epsilon^2)$
RFLIN Wagenmaker et al. (2022)	Linear MDP	Yes	$\widetilde{O}(H^5d^2/\epsilon^2)$
LSVI-RFE (Alg. 1)	Linear MDP	Yes	$\widetilde{O}(H^4d^2/\epsilon^2)$
Lower bound (Th. 6.1)	Linear MDP		$\Omega(H^3d^2/\epsilon^2)$

3 PRELIMINARIES

We consider episodic finite-horizon MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, H, \{\mathbb{P}_h\}_h, \{r_h\}_h\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $H \in \mathbb{Z}^+$ is the length of each episode, $\mathbb{P}_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ and $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ are time-dependent transition probability and deterministic reward function. We assume that $\mathcal{S}$ is a measurable space with a possibly infinite number of elements and $\mathcal{A}$ is a finite set.

For a time-inhomogeneous MDP, the policy is time-dependent, which is denoted as $\pi = \{\pi_1, ..., \pi_H\}$, where $\pi_h(s)$ is the action that agent takes at state s at the h-th step. We define the state-action function (i.e., Q-function) and value function as

$$Q_h^\pi(s, a; r) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \mid s_h = s, a_h = a, \pi\right], \tag{1}$$

$$V_h^\pi(s; r) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \mid s_h = s, \pi\right], \tag{2}$$

respectively for a specific set of the reward function $r = \{r_h\}_{h=1}^H$. For any function $V(\cdot; r) : \mathcal{S} \to \mathbb{R}$, we further denote $\mathbb{P}_h V(s, a; r) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} V(s'; r)$ and value function variance $[\mathbb{V}_h V](s, a; r) = \mathbb{P}_h V^2(s, a; r) - [\mathbb{P}_h V(s, a; r)]^2$, where V^2 stands for the function whose value at s is $V^2(s; r)$. The Bellman equation associated with a policy π for reward function r is

$$Q_h^\pi(s, a; r) = r_h(s, a) + \mathbb{P}_h V_{h+1}^\pi(s, a; r), \quad V_h^\pi(s; r) = Q_h^\pi(s, \pi_h(s); r), \tag{3}$$

[1] We consider time-inhomogenous settings and the bound of UCRL-RFE$^+$ Zhang et al. (2021) is degraded by an H factor.

for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$. Since the action space and the episode length are both finite, there always exists an optimal policy π^* for the reward function $r = \{r_h\}_{h=1}^H$, such that the associated optimal state-action function and value function are $Q_h^*(s, a; r) = \sup_\pi Q_h^\pi(s, a; r)$ and $V_h^*(s; r) = \sup_\pi V_h^\pi(s; r)$, respectively. For any $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$, the Bellman optimality equation for the reward function $r = \{r_h\}_{h=1}^H$ is

$$Q_h^*(s, a; r) = r_h(s, a) + \mathbb{P}_h V_{h+1}^*(s, a; r), \quad V_h^*(s; r) = \max_{a \in \mathcal{A}} Q_h^*(s, a; r), \tag{4}$$

The structural assumption we make in this paper is a linear structure in both transition and reward, which has been considered in prior works, e.g., Yang & Wang (2019); Jin et al. (2020b). The formal definition is given as follows.

Definition 3.1 (Linear MDP)**.** A MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, H, \{\mathbb{P}_h\}_h, \{r_h\}_h\}$ is a linear MDP with a known feature mapping $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist unknown d-dimensional measures $\boldsymbol{\mu}_h = (\mu_h(s)_{s \in \mathcal{S}}) \in \mathbb{R}^{d \times |\mathcal{S}|}$ over and an unknown vector $\boldsymbol{\theta}_h \in \mathbb{R}^d$, such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\mathbb{P}_h(s' \mid s, a) = \langle \boldsymbol{\phi}(s, a), \boldsymbol{\mu}_h(s') \rangle, \; r_h(s, a) = \langle \boldsymbol{\phi}(s, a), \boldsymbol{\theta}_h \rangle.$$

We make the following norm assumptions: for any $h \in [H]$, (i) $\sup_{s,a} \|\boldsymbol{\phi}(s, a)\|_2 \leqslant 1$, (ii) $\|\boldsymbol{\mu}_h \boldsymbol{v}\|_2 \leqslant \sqrt{d}$ for any vector $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|}$ such that $\|\boldsymbol{v}\|_\infty \leqslant 1$, (iii) $\|\boldsymbol{\theta}_h\|_2 \leqslant W$, where W is a constant. These assumptions are mild and are common in the existing literature Jin et al. (2020b); Agarwal et al. (2019).

Reward-Free Exploration (RFE) We consider the following RFE model, which has been considered in previous literature Jin et al. (2020a); Kaufmann et al. (2021); Ménard et al. (2021); Zhang et al. (2020). Specifically, there are two phases in the RFE paradigm. (i) *Exploration Phase*: The agent interacts with the environment for exploration up to K episodes without accessing the reward function. (ii) *Planning Phase*: The agent is given a reward function $r = \{r_h\}_{h=1}^H$ with the goal of outputting an ϵ-optimal policy π based on learned information from the exploration phase.

We define the *sample complexity* to be the number of episodes K required in the exploration phase to output an ϵ-optimal policy π in the planning phase, i.e.,

$$\mathbb{E}_{s \sim \mu} \left[V_1^*(s; r) \right] - \mathbb{E}_{s \sim \mu} \left[V_1^\pi(s; r) \right] \leqslant \epsilon$$

where $\mu \in \Delta(\mathcal{S})$ denotes the initial state distribution.

4 Algorithm and Main Results

This section presents our algorithm, LSVI-RFE, for reward-free reinforcement learning in linear MDPs. LSVI-RFE makes use of a carefully-designed exploration-driven reward function in the exploration phase (Algorithm 1), with an aggressive trajectory-aware mechanism for exploration (Line 7 of Algorithm 1). The exploration phase also utilizes weighted linear regression to estimate the parameter $\{\boldsymbol{\mu}_h\}_{h \in [H]}$. Then, the agent follows the greedy policy to the constructed optimistic value function. In the planning phase (Algorithm 2), an ϵ-optimal policy π is constructed by value iteration based on the estimated transition matrix from the exploration phase.

The key novelty of LSVI-RFE is the aggressive *trajectory-aware* exploration reward function in Eq. (5). Initially, this trajectory-aware reward function dynamically adjusts the weight according to the trajectory exploration measure, which controls the magnitude of the corresponding martingale difference sequence. This mechanism removes the additional cost of sample complexity on feature dimension d in building a uniform convergence argument by the covering net in existing works, e.g., Jin et al. (2020b); Wang et al. (2020a). In addition, the reward function is more aggressive than those in prior works Wang et al. (2020a); Zanette et al. (2020c) by a $2H$ factor to avoid overly-conservative exploration. This aggressiveness in the reward function removes the extra dependency of sample complexity on episode length H in prior works Wang et al. (2020a); Zanette et al. (2020c).

4.1 Exploration Phase

In the exploration phase, LSVI-RFE in Algorithm 1 follows the standard procedure of optimistic learning framework, yet is equipped with a dedicated exploration-driven reward function to explore

the environment. The observed state-action pairs are collected in each episode to estimate the parameter $\{\boldsymbol{\mu}_h\}_{h\in[H]}$ by weighted linear regression. Then the optimistic state-action function is constructed (Line 8), and the agent executes the greedy policy with respect to the optimistic state-action function (Line 10). Two critical steps of our LSVI-RFE algorithm in the exploration are trajectory-aware reward function and weighted linear regression, which are illustrated below. Specifically, Lines 13-20 are responsible for determining $\widehat{\omega}_k$ in the trajectory-aware reward function, and Line 23 presents the solution to the weighted ridge regression.

Aggressive Trajectory-Aware Reward Function If given the precise transition matrix, the optimal policy can be determined by value iteration for any reward function. Thus, it suffices to explore each state-action pair enough to estimate the transition matrix accurately. LSVI-RFE designs a dedicated exploration-driven reward function in Line 7 of Algorithm 1, i.e.,

$$r_{k,h}(s,a) = 2\widehat{\beta}^E \|\boldsymbol{\phi}(s,a)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}, \text{ for } \forall (s,a) \in \mathcal{S}\times\mathcal{A}, \tag{5}$$

where $\widehat{\boldsymbol{\Lambda}}_{k,h}$ is the covariance matrix of the feature mapping $\boldsymbol{\phi}$. Intuitively, the reward function in Eq. (5) represents the uncertainty level of the state-action pair (s,a) caused by the randomness of the transition matrix. Take the tabular setting for an example. If $\boldsymbol{\phi}(s,a) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ is a one-hot vector, i.e., zeros everywhere except that the entry corresponding to (s,a) is one, then $\widehat{\boldsymbol{\Lambda}}_{k,h}$ is a diagonal matrix and the diagonal entry contains $N_{k,h}(s,a)$, i.e., the number of times (s,a) has been visited at episode k and stage h. Thus, exploration-driven reward function for linear MDPs in Eq. (1) is consistent with that for tabular RFE in Jin et al. (2020a); Kaufmann et al. (2021), where the reward function takes the form of $\widehat{\beta}^E/\sqrt{N_{k,h}(s,a)}$. In other words, the reward function in Eq. (5) inherits the basic principle of exploration driven, i.e., it stimulates the agent to visit state-action pairs with low visitation times.

Algorithm 1 Least-Squares Value Iteration - RFE (LSVI-RFE): Exploration Phase

Require: Regularization parameter λ, exploration radius $\widehat{\beta}^E$

1: **for** step $h = H, ..., 1$ **do**
2: $\quad\widehat{\boldsymbol{\Lambda}}_{1,h}, \widetilde{\boldsymbol{\Lambda}}_{1,h} \leftarrow \lambda\mathbf{I}$; $\widehat{\boldsymbol{\mu}}_{1,h} \leftarrow \mathbf{0}$; $\widehat{V}_{0,h}(\cdot) \leftarrow H$;
3: **end for**
4: $\widehat{V}_{1,H+1}(\cdot) \leftarrow 0$
5: **for** episode $k = 1, ..., K$ **do**
6: $\quad$**for** step $h = H, ..., 1$ // Value iteration **do**
7: $\quad\quad r_{k,h}(\cdot,\cdot) = 2\widehat{\beta}^E\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}$ // Exploration driven reward
8: $\quad\quad \widehat{Q}_{k,h}(\cdot,\cdot) = r_{k,h}(\cdot,\cdot) + \langle\widehat{\boldsymbol{\mu}}_{k,h}\widehat{\boldsymbol{V}}_{k,h+1}, \boldsymbol{\phi}(\cdot,\cdot)\rangle + \widehat{\beta}^E\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}$ // Optimistic Q function
9: $\quad\quad \widehat{V}_{k,h}(\cdot) \leftarrow \min\left\{\max_{a\in\mathcal{A}}\widehat{Q}_{k,h}(\cdot,a), H\right\}$
10: $\quad\quad \pi_h^k(\cdot) \leftarrow \arg\max_{a\in\mathcal{A}}\widehat{Q}_{k,h}(\cdot,a)$
11: $\quad$**end for**
12: $\quad$Receive the initial state s_1^k
13: $\quad\widetilde{\omega}_k, \widehat{\omega}_k \leftarrow \sqrt{H}$
14: $\quad$**for** step $h = 1, ..., H$ **do**
15: $\quad\quad a_h^k \leftarrow \pi_h^k(s_h^k)$, and observe $s_{h+1}^k \sim \mathbb{P}_h(\cdot|s_h^k, a_h^k)$
16: $\quad\quad \widetilde{\boldsymbol{\Lambda}}_{k+1,h} \leftarrow \widetilde{\boldsymbol{\Lambda}}_{k,h} + \widetilde{\omega}_k^{-2}\boldsymbol{\phi}(s_h^k, a_h^k)\boldsymbol{\phi}(s_h^k, a_h^k)^\top$
17: $\quad\quad$**if** $\|\widetilde{\omega}_k^{-1}\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Lambda}}_{k,h}^{-1}} > 1/(8\widehat{\beta}^E d)$ **then**
18: $\quad\quad\quad \widehat{\omega}_k \leftarrow \sqrt{Hd}$ // Enlarge weight $\widehat{\omega}_k$ if necessary
19: $\quad\quad$**end if**
20: $\quad$**end for**
21: $\quad$**for** $h = 1, ...H$ **do**
22: $\quad\quad \widehat{\boldsymbol{\Lambda}}_{k+1,h} \leftarrow \widehat{\boldsymbol{\Lambda}}_{k,h} + \widehat{\omega}_k^{-2}\boldsymbol{\phi}(s_h^k, a_h^k)\boldsymbol{\phi}(s_h^k, a_h^k)^\top$
23: $\quad\quad \widehat{\boldsymbol{\mu}}_{k+1,h} \leftarrow \widehat{\boldsymbol{\Lambda}}_{k+1,h}^{-1}\sum_{i=1}^k \widehat{\omega}_i^{-2}\boldsymbol{\phi}(s_h^i, a_h^i)\boldsymbol{\delta}(s_{h+1}^i)^\top$ // Solution to the weighted linear regression
24: $\quad$**end for**
25: **end for**
26: Return $\{\widehat{\boldsymbol{\Lambda}}_{K+1,h}, \widehat{\boldsymbol{\mu}}_{K+1,h}\}_{h\in[H]}$.

However, we emphasize our reward function is trajectory-aware and aggressive with the following critical difference to prior algorithms Wang et al. (2020a); Zanette et al. (2020b); Chen et al. (2021) for RFE:

(i) **Trajectory-aware weights:** We introduce a weighted reward function, where the covariance matrix $\widehat{\boldsymbol{\Lambda}}_{k,h}$ contains a weight term $\widehat{\omega}_k$ for different episodes k in Line 22 of Algorithm 1. $\widehat{\omega}_k$ is dynamically controlled according to the trajectory property in Lines 13-20 to keep the sum of elliptical potentials small, i.e., $\sum_{h=1}^{H} \widehat{\beta}^E \|\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}/\widehat{\omega}_k$ is small. Specifically, we have the following two cases: (a) If $\|\widetilde{\omega}_k^{-1}\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Lambda}}_{k,h}^{-1}} \leqslant 1/(8\widehat{\beta}^E d)$ for any $h \in [H]$, the claim holds trivially; (b) Otherwise, the claim still holds by enlarging $\widehat{\omega}_k$ from $\sqrt{H}$ to $\sqrt{Hd}$.

Remark 4.1. The motivation to dynamically adjust the weight is due to the conservatism of elliptical potentials (Lemma C.9 in Appendix) such that we can keep $\sum_{h=h'}^{H} \widehat{\beta}^E \|\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}/\widehat{\omega}_k$ small with neglectable cost. Considering $\sum_{h=h'}^{H} \widehat{\beta}^E \|\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}/\widehat{\omega}_k$ is the upper bound of the magnitude of the considered martingale sequence, this fine-grained control of the magnitude is critical for removing the extra dependency on d, which is detailed in step 2 in Section 5.

(ii) **Aggressive $2H$ enlargement:** Our reward function is more aggressive by a factor of $2H$ enlargement. Existing works for RFE in linear MDPs, e.g., Wang et al. (2020a), sets the reward function as $\min\{\beta\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}/H, 1\}$ so that it belongs to $[0, 1]$, where $\beta\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}$ is the exploration bonus. Our reward function in Eq. (5) instead takes the same order as the exploration bonus, i.e., no $1/H$ factor but with a $2H$ enlargement. Initially, we introduce a factor of H enlargement to achieve faster learning rates, which is firstly proposed in Chen et al. (2021) for linear mixture MDPs, removing the extra H^2 factor in sample complexity. We prove that H enlargement also works for our trajectory-aware reward function and saves an H^2 factor in the sample complexity. Secondly, a factor of 2 enlargement is required to ensure the monotonicity of the estimated value function between the exploration phase and planning phase, i.e., $V_h^*(\cdot; r) + \widehat{V}_{k,h}(\cdot) \geqslant \widehat{V}_h(\cdot)$ (Lemma A.14 in Appendix), which is also necessary for removing the extra dependency on d.

Weighted Linear Regression To enable trajectory-aware exploration, we need to employ weighted linear regression to assemble trajectory-aware weights in linear regression. Note that the weighted ridge regression estimator was initially built for linear bandits with heteroscedastic noises, e.g., Lattimore et al. (2015); Kirschner & Krause (2018). Specifically, denote $\boldsymbol{\delta}(s) \in \mathbb{R}^{|\mathcal{S}|}$ as a one-hot vector that is zero everywhere except the entry corresponding to state s is one, and define $\boldsymbol{\epsilon}_h^k := \mathbb{P}_h(\cdot \mid s_h^k, a_h^k) - \boldsymbol{\delta}(s_{h+1}^k)$. We see that $\mathbb{E}[\boldsymbol{\epsilon}_h^k \mid s_h^k, a_h^k] = 0$, which implies $\boldsymbol{\delta}(s_{h+1}^k)$ is an unbiased estimate of $\mathbb{P}_h(\cdot \mid s_h^k, a_h^k) = \boldsymbol{\mu}_h^\top \boldsymbol{\phi}(s_h^k, a_h^k)$. Thus, $\boldsymbol{\mu}_h$ can be learned via regression from $\boldsymbol{\phi}(s_h^k, a_h^k)$ to $\boldsymbol{\delta}(s_{h+1}^k)$. Intuitively, we enlarge the weight $\widehat{\omega}_k$ if $\sum_{h=1}^{H} \widehat{\beta}^E \|\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}$ is large. In particular, $\sum_{h=1}^{H} \widehat{\beta}^E \|\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}$ upper bounds the episodic exploration error (detailed in Section 5), i.e., $\widehat{V}_{k,h+1}(s_h^k)$, by Lemma A.17 in Appendix. Thus, enlarging the weight $\widehat{w}_k$ when the policy incurs large episodic exploration error ensures that trajectory data generated by this poor policy have smaller weights in the regression problem in Eq. (6). The estimated parameter $\widehat{\boldsymbol{\mu}}_{k,h}$ in Line 23 of Algorithm 1 is the solution to the following weighted linear regression:

$$\min_{\boldsymbol{\mu} \in \mathbb{R}^{d \times |\mathcal{S}|}} \sum_{i=1}^{k-1} \left\| \left[\boldsymbol{\mu}_h^\top \boldsymbol{\phi}(s_h^k, a_h^k) - \boldsymbol{\delta}(s_{h+1}^i) \right] \widehat{\omega}_i^{-1} \right\|_2^2 + \lambda \|\boldsymbol{\mu}\|_F^2, \tag{6}$$

where $\|\cdot\|_F$ denotes Frobenius norm, with solution

$$\widehat{\boldsymbol{\mu}}_{k,h} = \widehat{\boldsymbol{\Lambda}}_{k,h}^{-1} \sum_{i=1}^{k-1} \widehat{\omega}_i^{-2} \boldsymbol{\phi}(s_h^i, a_h^i) \boldsymbol{\delta}(s_{h+1}^i)^\top, \tag{7}$$

where $\widehat{\boldsymbol{\Lambda}}_{k,h} = \sum_{i=1}^{k-1} \widehat{\omega}_{i,h}^{-2} \boldsymbol{\phi}(s_h^i, a_h^i) \boldsymbol{\phi}(s_h^i, a_h^i)^\top + \lambda \mathbf{I}$, and the estimated transition probability is denoted as $\widehat{\mathbb{P}}_{k,h}(\cdot \mid s_h^k, a_h^k) = \widehat{\boldsymbol{\mu}}_{k,h}^\top \boldsymbol{\phi}(s_h^k, a_h^k)$.

Algorithm 2 Least-Squares Value Iteration - RFE (LSVI-RFE): Planning Phase

Require: Planning radius $\widehat{\beta}^P$, parameter $\{\widehat{\boldsymbol{\Lambda}}_{K+1,h}, \widehat{\boldsymbol{\mu}}_{K+1,h}\}_{h\in[H]}$, reward function $r = \{r_h\}_{h\in[H]}$

1: $\widehat{V}_{H+1}(\cdot) \leftarrow 0$
2: **for** step $h = H, ..., 1$ **do**
3: // Value iteration
4: $b_h(\cdot,\cdot) = \min\{\widehat{\beta}^P \|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}^{-1}_{K+1,h}}, H\}$
5: $\widehat{Q}_h(\cdot,\cdot) = r_h(\cdot,\cdot) + \langle \widehat{\boldsymbol{\mu}}_{K+1,h}\widehat{\boldsymbol{V}}_{h+1}, \boldsymbol{\phi}(\cdot,\cdot)\rangle + b_h(\cdot,\cdot)$ // Optimistic Q function
6: $\widehat{V}_h(\cdot) \leftarrow \min\left\{\max_{a\in\mathcal{A}} \widehat{Q}_h(\cdot,a), H\right\}$
7: $\pi_h(\cdot) \leftarrow \arg\max_{a\in\mathcal{A}} \widehat{Q}_h(\cdot,a)$
8: **end for**
9: Return $\boldsymbol{\pi} = \{\pi_h\}_{h\in[H]}$

4.2 Planning Phase

During the planning phase, the ϵ-optimal policy π is the greedy policy concerning the optimistic value iteration with respect to the estimated transition matrix from the exploration phase, i.e., parameters $\{\widehat{\boldsymbol{\mu}}_{K+1,h}\}_{h\in[H]}$. Here we also introduce a UCB bonus term $b_h(\cdot,\cdot)$ to ensure optimism. As revealed in Step 4 in Section 5, the summation of UCB bonus term $b_h(\cdot,\cdot)$ is small on average, which ensures the near optimality of the returned greedy policy π.

4.3 Sample Complexity

The sample complexity of the proposed LSVI-RFE algorithm is given below in Theorem 4.2 with a proof sketch in Section 5, and the detailed proof is in Appendix A.

Theorem 4.2 (Upper bound)**.** *Set the confidence radius as $\widehat{\beta}^E = \widetilde{O}(d\sqrt{H})$ and $\widehat{\beta}^P = \widehat{\beta}^{P(1)} + \widehat{\beta}^{P(2)} = \widetilde{O}(\sqrt{Hd})$*[2]*. After collecting $\widetilde{O}\left(H^4d^2/\epsilon^2\right)$ trajectories during the exploration phase, with probability $1-4\delta$, LSVI-RFE outputs an ϵ-optimal policy for an arbitrary reward function satisfying Definition 3.1 during the planning phase.*

Remark 4.3. Theorem 4.2 shows that LSVI-RFE achieves the state-of-the-art sample complexity for RFE in linear MDPs. Compared to the RFE algorithms in Wang et al. (2020a); Zanette et al. (2020c); Wagenmaker et al. (2022) for linear MDPs, our sample complexity is sharper in H, which comes from the more aggressive reward function, having the same order as the exploration bonus. Moreover, the optimal dependency on d is achieved by our trajectory-aware reward function such that the extra overhead due to building the uniform convergence argument in Wang et al. (2020a); Chen et al. (2021) is removed.

5 Mechanism

In this section, we overview the key techniques and ideas in the analysis of reward-free exploration and the proof of Theorem 4.2. As preliminary steps in optimistic learning, we build the confidence sets $\widehat{\mathcal{C}}^E_{k,h}$ and $\widehat{\mathcal{C}}^P_h$ for both exploration and planning phases, respectively in Steps 1 and 2. Subsequently, we bound the exploration error, i.e., summation of the exploration bonus, during the exploration phase based on the built confidence set $\widehat{\mathcal{C}}^E_{k,h}$ in Step 3. Finally, we can bound the sub-optimality gap of the recovered policy in the planning phase with confidence set $\widehat{\mathcal{C}}^P_h$ and the monotonicity of the exploration bonus in Step 4. The full proof is in Appendix A.

Step 1: Build Confidence Set $\widehat{\mathcal{C}}^E_{k,h}$ in the Exploration Phase LSVI-RFE estimates the parameter $\{\boldsymbol{\mu}_h\}_{h\in[H]}$ of the transition probability matrix motivated by the trajectory-aware reward function in the exploration phase. The confidence set $\widehat{\mathcal{C}}^E_{k,h}$ is built with a Hoeffding-type self-normalized bound, i.e., Lemma C.1 in Appendix and standard covering net argument, e.g., Lemma B.3. in Jin

[2]The exact forms of $\widehat{\beta}^E$, $\widehat{\beta}^{P(1)}$, $\widehat{\beta}^{P(2)}$ are given in Eq. (23), (31) and (33), respectively in Appendix.

et al. (2020b), such that we have for any $k \in [K], h \in [H]$,

$$\boldsymbol{\mu}_h \in \widehat{\mathcal{C}}^E_{k,h} := \left\{ \boldsymbol{\mu} : \left\| (\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}_{k,h}) \widehat{\boldsymbol{V}}_{k,h+1} \right\|_{\widehat{\Lambda}_{k,h}} \leqslant \widehat{\beta}^E \right\} \tag{8}$$

with high probability, which is detailed in Lemma C.2 with $\widehat{\beta}^E = \widetilde{O}(\sqrt{H}d)$. Consequently, we have $(\widehat{\mathbb{P}}_{k,h} - \mathbb{P}_h)\widehat{V}_{k,h+1} \leqslant \widehat{\beta}^E \|\boldsymbol{\phi}(s^k_h, a^k_h)\|_{\widehat{\boldsymbol{\Lambda}}^{-1}_{k,h}}$ by the Cauchy-Schwarz inequality. Under confidence set $\widehat{\mathcal{C}}^E_{k,h}$, $\widehat{V}_{k,h}(\cdot)$ is a upper confidence estimator of the optimal value function with exploration-driven reward, i.e., $V^*_h(\cdot; r^k)$, where $r_k = \{r_{k,h}\}_{h\in[H]}$.

Step 2: Build the Confidence Set $\widehat{\mathcal{C}}^P_h$ in the Planning Phase We prove

$$\boldsymbol{\mu}_h \in \widehat{\mathcal{C}}^P_h := \left\{ \boldsymbol{\mu} : \left\| (\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}_{K+1,h}) \widehat{\boldsymbol{V}}_{h+1} \right\|_{\widehat{\Lambda}_{K+1,h}} \leqslant \widehat{\beta}^P \right\}, \tag{9}$$

with high probability for any $h \in [H]$ in the planning phase, where the bonus radius $\widehat{\beta}^P = \widetilde{O}(\sqrt{dH})$ is sharper than $\widehat{\beta}^E = \widetilde{O}(d\sqrt{H})$ in the exploration phase due to the trajectory-aware reward function in Eq. (5). Specifically, we build the confidence set $\widehat{\mathcal{C}}^P_h$ by the intersection of two confidence sets $\widehat{\mathcal{C}}^{P(1)}_h$ and $\widehat{\mathcal{C}}^{P(2)}_h$, similarly as that in Azar et al. (2017) for tabular MDPs, where

$$\widehat{\mathcal{C}}^{P(1)}_h := \left\{ \boldsymbol{\mu} : \|(\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}_{K+1,h}) \boldsymbol{V}^*_{h+1}\|_{\widehat{\Lambda}_{K+1,h}} \leqslant \widehat{\beta}^{P(1)} \right\},$$
$$\widehat{\mathcal{C}}^{P(2)}_h := \left\{ \boldsymbol{\mu} : \left\|(\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}_{K+1,h}) \left(\widehat{\boldsymbol{V}}_{h+1} - \boldsymbol{V}^*_{h+1}\right)\right\|_{\widehat{\Lambda}_{K+1,h}} \leqslant \widehat{\beta}^{P(2)} \right\},$$

where $V^*_{h+1}(\cdot)$ refers to $V^*_{h+1}(\cdot; r)$. (i) In particular, since $V^*_{h+1}(\cdot)$ is a fixed function, where r is the real reward function given in planning phase, we can build confidence set $\widehat{\mathcal{C}}^{P(1)}_h$ with radius $\widehat{\beta}^{P(1)} = \widetilde{O}(\sqrt{Hd})$ by Hoeffding-type self-normalized bound directly without a uniform convergence argument, as detailed in Lemma A.10. (ii) When building the confidence set $\widehat{\mathcal{C}}^{P(2)}_h$, the aggressiveness in the trajectory-aware reward function in Eq. (5) ensures the monotonicity of the constructed optimistic value function between the exploration phase and planning phase, i.e., $V^*_h(\cdot; r) + \widehat{V}_{k,h}(\cdot) \geqslant \widehat{V}_h(\cdot)$ (Lemma A.14 in Appendix), leading to $[\widehat{V}_{h+1} - V^*_{h+1}]/\widehat{\omega}_k \leqslant \widehat{V}_{k,h+1}/\widehat{\omega}_k$ for any $k \in [K]$, $h \in [H]$. Besides, $r_{k,h}(\cdot,\cdot)$ in Eq. (5) also ensures $\widehat{V}_{k,h}(\cdot)/\widehat{\omega}_k \leqslant \sum_{h=1}^H \widehat{\beta}^E \|\boldsymbol{\phi}(s^k_h, a^k_a)\|_{\widehat{\boldsymbol{\Lambda}}^{-1}_{k,h}}/\widehat{\omega}_k \lesssim \sqrt{H}/\sqrt{d}$, where the first inequality holds due to Lemma A.17 in Appendix, such that the magnitude of the considered martingale difference sequence is small and $\widehat{\beta}^{P(2)} = \widetilde{O}(\sqrt{Hd})$ as detailed in Lemma A.17. Consequently, the overhead due to building a uniform convergence argument by covering net is removed and we have $\widehat{\beta}^P = \widehat{\beta}^{P(1)} + \widehat{\beta}^{P(2)} = \widetilde{O}(\sqrt{Hd})$.

Step 3: Bound the Exploration Error The exploration error refers to the summation of the constructed optimistic value function in the exploration phase, i.e., $\widehat{V}_{k,1}$, which is upper bounded by the summation of the bonus term under the confidence sets $\widehat{\mathcal{C}}^E_{k,h}$ for any $h \in [H]$. It is named as the exploration error since the summation of the bonus term upper bounds the estimation error $(\widehat{\mathbb{P}}_{k,h} - \mathbb{P}_h)\widehat{V}_{k,h+1}$. In particular, the exploration error can be upper bounded by Lemma A.19 as

$$\sum_{k=1}^K \widehat{V}_{k,1}(s^k_1) \leqslant 4\widehat{\beta}^E \underbrace{\sqrt{\sum_{k=1}^K \sum_{h=1}^H \widehat{\omega}^2_k}}_{I_1} \underbrace{\sqrt{\sum_{k=1}^K \sum_{h=1}^H \min\left\{ \|\widehat{\omega}^{-1}_k \boldsymbol{\phi}(s,a)\|^2_{\widehat{\boldsymbol{\Lambda}}^{-1}_{k,h}}, 1 \right\}}}_{I_2} \leqslant \widetilde{O}(\sqrt{d^3H^4K}), \tag{10}$$

where the second inequality holds since I_2 can be bounded as $\widetilde{O}(\sqrt{Hd})$ by the elliptical potential lemma (Lemma C.7), and $I_1 = \widetilde{O}(\sqrt{HT})$ since the enlargement operation in Line 18 of Algorithm 1 rarely happens due to the conservatism of elliptical potentials according to Lemma C.9.

Step 4: Bound the Sub-optimality Gap The sub-optimality gap refers to the expected gap between the optimal value function $V^*_1(\cdot; r)$ and the value function $V^\pi_1(\cdot; r)$ associated with the

recovered policy π in the planning phase. Notice that $\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{K+1,h}^{-1}} \leqslant \|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}}$ since $\widehat{\boldsymbol{\Lambda}}_{k,h} \preceq \widehat{\boldsymbol{\Lambda}}_{K+1,h}$ for any $k \in [K]$. We thus have

$$r_{k,h}(\cdot,\cdot) = \widehat{\beta}^E \|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{k,h}^{-1}} \geqslant \sqrt{d} b_h(\cdot,\cdot) = \sqrt{d}\widehat{\beta}^P \|\boldsymbol{\phi}(\cdot,\cdot)\|_{\widehat{\boldsymbol{\Lambda}}_{K+1,h}^{-1}}, \tag{11}$$

where $b_h(\cdot,\cdot)$ is the exploration bonus in the planning phase. Then, we can apply standard analysis as in Wang et al. (2020a) to bound $\mathbb{E}_{s_1\sim\mu}\left[V_1^*\left(s_1;r\right) - V_1^\pi\left(s_1;r\right)\right]$ by

$$\begin{aligned}\mathbb{E}_{s_1\sim\mu}\left[V_1^*\left(s_1;r\right) - V_1^\pi\left(s_1;r\right)\right] &\leqslant \mathbb{E}_{s_1\sim\mu}\left[\widehat{V}_1\left(s_1\right) - V_1^\pi\left(s_1;r\right)\right] \\ &\leqslant \mathbb{E}_{s_1\sim\mu}\left[\widetilde{V}_1^\pi(s_1;b)\right] \leqslant \mathbb{E}_{s_1\sim\mu}\left[\widetilde{V}_1^\pi(s_1;r_k)\right]/\sqrt{d} \leqslant \mathbb{E}_{s_1\sim\mu}\left[\sum_{k=1}^K \widehat{V}_{k,1}(s_1)\right]/(K\sqrt{d}),\end{aligned} \tag{12}$$

where the first inequality holds due to optimism under the confidence sets $\widehat{\mathcal{C}}_h^P$ for any $h \in [H]$ (Lemma A.14), the second inequality is the application of regret composition and $\widetilde{V}_1^\pi$is an auxiliary value function defined in Appendix, the third inequality holds by Eq. (11), and the last inequality holds due to the optimism under the confidence sets $\widehat{\mathcal{C}}_{k,h}^E$ for any $h \in [H]$ (Lemma A.7). Indeed, Eq. (12) builds the connection between the exploration phase and planning phase. Thus, if π is an ϵ-optimal policy, we obtain $K \geqslant \widetilde{O}(d^2H^4/\epsilon^2)$ by combining Eq. (10) and Eq. (12).

6 Lower Bound and Optimality Gap

We provide a sample complexity lower bound for reward-free RL under the linear MDP setting in Theorem 6.1. We show that there exists an instance of linear MDP, such that any reward-free RL algorithm requires $\Omega(d^2H^3/\epsilon^2)$ episodes of interaction during the exploration phase to find a near-optimal policy during the planning phase.

Theorem 6.1 (Lower Bound)**.** *Suppose $H \geqslant 4, d \geqslant 2, 1/32K < \delta < 1/H$. Then, there exists a linear MDP instance $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \{\mathbb{P}_h\}, \{r_h\})$, such that any algorithm $\mathcal{ALG}$ that learns an ϵ-optimal policy with probability at least $1-\delta$ needs to collect at least $K = Cd^2H^3/\epsilon^2$ episodes during the exploration phase, where C is an absolute constant, and δ has no dependence on ϵ, H, d, K.*

Proof Sketch. Notice that if the reward function is given in the exploration phase, the RFE setting degrades to Probably Approximately Correct (PAC) RL setting Dann et al. (2019). Thus, it suffice to prove a lower bound for PAC RL, since an algorithm for RFE also works for PAC RL by neglecting the reward function in the exploration phase. The proof of Theorem 6.1 is inspired by the lower bound of RFE in linear mixture MDPs in Chen et al. (2021). First we construct a hard-to-learn MDP $\mathcal{M}$ such that any algorithm which runs K episodes will obtain the regret at least $\Omega(dH\sqrt{HK})$ as shows in Zhou et al. (2021). Then, for any algorithm $\mathcal{ALG}_1$ running K episodes to learn an ϵ-optimal policy $\pi(K)$ with probability at least $1-\delta$, it suffices to prove that under the instance $\mathcal{M}$, $K \geqslant Cd^2H^3/\epsilon^2$. We prove this by constructing a new algorithm $\mathcal{ALG}_2$ which runs $\mathcal{ALG}_1$ in the first K episodes and executes the generated policy $\pi(K)$ in the rest $(c-1)K$ episodes, where $c > 1$ is a positive constant. In particular, the regret under $\mathcal{M}$ in the last $(c-1)K$ episodes by executing $\pi(K)$ satisfies

$$\Omega(dH\sqrt{HK}) \lesssim \sum_{k=K+1}^{K_2} \mathbb{E}_{x_1\sim\nu}\left[V_1^*(x_1) - V_1^{\widehat{\pi}}(x_1)\right] \lesssim K\epsilon, \tag{13}$$

where ν is the initial state distribution, the first inequality holds due to the hardness of the constructed MDP, and the second inequality holds due to $\pi(K)$ is an ϵ-optimal policy. Thus, Eq. (13) directly gives $K \geqslant \Omega(d^2H^3/\epsilon^2)$. For detailed proof, please refer to Appendix B. □

Remark 6.2. Theorem 6.1 presents an improved sample complexity lower bound for RFE in linear MDPs than the results of $\Omega(H^2d/\epsilon^2)$ in Zhang et al. (2021) and $\Omega(d^2/\epsilon^2)$ in Wagenmaker et al. (2022). The lower bound and Theorem 4.2 together show that the sample complexity of LSVI-RFE, i.e., $\widetilde{O}(H^4d^2/\epsilon^2)$, matches the lower bound $\Omega(H^3d^2/\epsilon^2)$ except for an H and logarithmic factors. To our best knowledge, the upper bound in Theorem 4.2 and lower bound in Theorem 6.1 are both sharper than those in existing works Wang et al. (2020a); Zanette et al. (2020c); Zhang et al. (2021); Chen et al. (2021).

On the optimal dependency on H The sample complexity of LSVI-RFE is $\widetilde{O}(H^4 d^2/\epsilon^2)$, which matches the lower bound up to an H and logarithmic factors. The factor H between the upper and lower bounds is potentially due to utilizing a Hoeffding-type bonus instead of a Bernstein-type one. Intuitively, a Bernstein-type bonus based on the variance of the value function, combined with the Law of Total Variance (LTV) Lattimore & Hutter (2012), can effectively reduce a $\sqrt{H}$ factor in the statistical complexity of RL algorithms. This phenomenon has been observed in existing works Azar et al. (2017); Jin et al. (2018); Zanette & Brunskill (2019) for regret minimization in tabular MDPs. However, under the RFE setting, the agent is unaware of the reward during the exploration phase, which brings obstacles to building the variance estimator.

Degradation to PAC RL Our sample complexity upper bound and lower bound in Theorem 4.2 and 6.1 are also applicable to the PAC RL setting, i.e., the agent is aware of the reward function during the exploration phase, both upper and lower bound are sharp yet there is still an H gap. However, a direct adaption of the Bernstein-type bonus will not improve the sample complexity under PAC RL due to policy inconsistency. Specifically, when we apply the total variance lemma (Lemma C.5 in Jin et al. (2018)), the exploration policy π^k for episode k is inconsistent with the recovered policy π in the planning phase, since π^k is the greedy with respect to Q function by exploration-driven reward function, not the real reward function given in the planning phase. A potential solution is to bound the distance between two policies by policy distance measures like KL divergence by noticing it helps RFE RL in tabular settings to reach minimax optimality in Ménard et al. (2021).

7 Conclusion

This work studies reward-free reinforcement learning with linear function approximation for episodic Markov decision processes (MDPs). We propose a novel algorithm LSVI-RFE with $\widetilde{O}(H^4 d^2/\epsilon^2)$ sample complexity upper bound for linear MDPs. We also establish a sample complexity lower bound of $\Omega(H^3 d^2/\epsilon^2)$, showing that LSVI-RFE's complexity is optimal up to an H and logarithmic factors. LSVI-RFE introduces a novel trajectory-aware exploration bonus with weighted linear regression to avoid overly-aggressive exploration in prior works. Our sharper bound relies on a novel analysis concerning the conservatism of elliptical potentials to remove the extra dependency of sample complexity on d. We leave removing the H gap as future work.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 2019.

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.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pp. 1283–1294. PMLR, 2020.

Xiaoyu Chen, Jiachen Hu, Lin F Yang, and Liwei Wang. Near-optimal reward-free exploration for linear mixture mdps with plug-in solver. *arXiv preprint arXiv:2110.03244*, 2021.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient pac rl with rich observations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 1429–1439, 2018.

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pp. 1507–1516. PMLR, 2019.

Jiafan He, Dongruo Zhou, and Quanquan Gu. Logarithmic regret for reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pp. 4171–4180. PMLR, 2021.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3796–3803, 2019.

Zeyu Jia, Lin Yang, Csaba Szepesvari, and Mengdi Wang. Model-based reinforcement learning with value-targeted regression. In *Learning for Dynamics and Control*, pp. 666–686. PMLR, 2020.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pp. 1704–1713. PMLR, 2017.

Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020a.

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, 2020b.

Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pp. 865–891. PMLR, 2021.

Johannes Kirschner and Andreas Krause. Information directed sampling and bandits with heteroscedastic noise. In *Conference On Learning Theory*, pp. 358–384. PMLR, 2018.

Tor Lattimore and Marcus Hutter. Pac bounds for discounted mdps. In *International Conference on Algorithmic Learning Theory*, pp. 320–334. Springer, 2012.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Linear multi-resource allocation with semi-bandit feedback. In *NIPS*, pp. 964–972, 2015.

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pp. 7599–7608. PMLR, 2021.

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pp. 2010–2020. PMLR, 2020.

Mirco Mutti, Riccardo De Santi, Emanuele Rossi, Juan Felipe Calderon, Michael Bronstein, and Marcello Restelli. Provably efficient causal model-based reinforcement learning for systematic generalization. *arXiv preprint arXiv:2202.06545*, 2022.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299. IEEE, 2018.

Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pp. 4344–4353. PMLR, 2018.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, pp. 2898–2933. PMLR, 2019.

Andrew Wagenmaker, Yifang Chen, Max Simchowitz, Simon S Du, and Kevin Jamieson. First-order regret in reinforcement learning with linear function approximation: A robust estimation approach. *arXiv preprint arXiv:2112.03432*, 2021.

Andrew Wagenmaker, Yifang Chen, Max Simchowitz, Simon S Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. *arXiv preprint arXiv:2201.11206*, 2022.

Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020a.

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, 2020b.

Yining Wang, Ruosong Wang, Simon Shaolei Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. In *International Conference on Learning Representations*, 2020c.

Jingfeng Wu, Vladimir Braverman, and Lin Yang. Gap-dependent unsupervised exploration for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 4109–4131. PMLR, 2022.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pp. 6995–7004. PMLR, 2019.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *Advances in Neural Information Processing Systems*, 33:4767–4777, 2020.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34, 2021.

Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pp. 7304–7312. PMLR, 2019.

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, 2020a.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pp. 10978–10989. PMLR, 2020b.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020c.

Weitong Zhang, Dongruo Zhou, and Quanquan Gu. Reward-free model-based reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems*, 34, 2021.

Zihan Zhang, Simon S Du, and Xiangyang Ji. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*, 2020.

Dongruo Zhou, Quanquan Gu, and Csaba Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pp. 4532–4576. PMLR, 2021.