Nearly Optimal Reward-Free Reinforcement Learning

Zihan Zhang [1] **Simon S. Du** [2] **Xiangyang Ji** [1]

Abstract

We study the reward-free reinforcement learning framework, which is particularly suitable for batch reinforcement learning and scenarios where one needs policies for multiple reward functions. This framework has two phases: in the exploration phase, the agent collects trajectories by interacting with the environment without using any reward signal; in the planning phase, the agent needs to return a near-optimal policy for arbitrary reward functions. We give a new efficient algorithm, **S**taged **S**ampling + **T**runcated **P**lanning (SSTP), which interacts with the environment at most $O\left(\frac{S^2A}{\epsilon^2}\text{poly}\log\left(\frac{SAH}{\epsilon}\right)\right)$ episodes in the exploration phase, and guarantees to output a near-optimal policy for arbitrary reward functions in the planning phase, where S is the size of state space, A is the size of action space, H is the planning horizon, and ϵ is the target accuracy relative to the total reward. Notably, our sample complexity scales only *logarithmically* with H, in contrast to all existing results which scale *polynomially* with H. Furthermore, this bound matches the minimax lower bound $\Omega\left(\frac{S^2A}{\epsilon^2}\right)$ up to logarithmic factors. Our results rely on three new techniques : 1) A new sufficient condition for the dataset to plan for an ϵ-suboptimal policy ; 2) A new way to plan efficiently under the proposed condition using soft-truncated planning; 3) Constructing extended MDP to maximize the truncated accumulative rewards efficiently.

1. Introduction

Reinforcement learning (RL) studies the problem in which an agent aims to maximize its accumulative rewards by interaction with an unknown environment. A major challenge in RL is *exploration* for which the agent needs to strategically visit new states to learn transition and reward information therein. To execute efficient exploration, the agent must follow a well-designed adaptive strategy by which the agent is properly guided by the reward and transition information, other than the trivial random exploration. Provably algorithms have been proposed to help the agent visit new states efficiently with a fixed reward and transition model. See Section 2 for a review.

However, in various applications, it is necessary to re-design the reward function to incentivize the agent to learn new desired behavior (Altman, 1999; Achiam et al., 2017; Tessler et al., 2018; Miryoosefi et al., 2019). To avoid repeatedly invoking the learning algorithm and interacting with the environment, it is desired to let the agent efficiently explore the environment *without the reward signal* and collect data based on which the agent can compute a near-optimal policy for *any* reward function.

The main challenge of this problem is that the agent needs to collect data that sufficiently covers the state space. This problem was previous studied in (Brafman & Tennenholtz, 2003; Hazan et al., 2019; Du et al., 2019). Recently, Jin et al. (2020) formalized the setting, and named it reward-free RL. In this setting, the agent first collects a dataset by interacting with the environment, and then is required to compute an ϵ-optimal policy given any proper reward function.

Jin et al. (2020) gave a formal theoretical treatment of this setting. They designed a method which guarantees that by collecting $O\left(\left(\frac{S^2AH^3}{\epsilon^2}+\frac{S^4AH^5}{\epsilon}\right)\text{poly}\log\left(SAH/\epsilon\right)\right)$ episodes, the agent is able to output an ϵ-optimal policy, where S is the number of states, A is the number of actions, and H is the planning horizon. [1] They also provided an $\Omega\left(\frac{S^2A}{\epsilon^2}\right)$ lower bound. Recently, Kaufmann et al. (2020); Ménard et al. (2020) gave tighter sample complexity bound. We refer the readers to table 1 for more details.

Unfortunately, it remains open what is the fundamental limit of the sample complexity of reward-free RL. In particular, compared to the $\Omega(\frac{S^2A}{\epsilon^2})$ lower bound, all existing upper bounds have a *polynomial* dependence on H. The gap be-

[1]Tsinghua University [2]University of Washington. Correspondence to: Zihan Zhang <zihan-zh17@mails.tsinghua.edu.cn>, Simon S. Du <ssdu@cs.washington.edu>, Xiangyang Ji <xyji@tsinghua.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] Because we consider reward function satisfying the total reward bounded by 1 setting (instead of H in their paper) this paper, we rescale the error ϵ to ϵH in the bound .

tween upper and lower bound can be huge for environments with a long horizon. Conceptually, this gap represents that we still lack understanding on whether long horizon imposes significant hardness on reward-free RL.

1.1. Our Contribution

In this work, we break the $\text{poly}(H)$-dependency barrier. We design a new algorithm, **S**taged **S**ampling + **T**runcated **P**lanning (SSTP), which enjoys the following sample complexity guarantee.

Theorem 1. *For any $\epsilon, \delta \in (0,1)$, there exists an algorithm (SSTP, Algorithm 1) which can compute an ϵ-optimal policy for any reward function that is non-negative and totally bounded by 1, after $O\left(\left(\frac{SA}{\epsilon^2}\left(S + \log\left(\frac{1}{\delta}\right)\right)\right) \text{poly}\log\left(SAH/\epsilon\right)\right)$ episodes of exploration with probability $1-\delta$.*

The significance of our theorem is that we match the lower bound of $\Omega\left(\frac{SA(S+\log 1/\delta)}{\epsilon^2}\right)$ up to logarithmic factors on $S, A, H, 1/\epsilon$. [2]

Importantly, our bound only depends logarithmically on the planning horizon H. This is an exponential improvement over existing results, and demonstrates that long horizon poses little additional difficulty for reward-free RL. Furthermore, our bound only requires the reward to be totally bounded (cf. Assumption 2), in contrast to uniformly bounded (cf. Assumption 1), which is assumed in previous works. See Section 2 for more discussions.

Remark 1. *Jin et al. (2020),Kaufmann et al. (2020) and Ménard et al. (2020) studied reward-free exploration on the non-stationary episodic MDP (i.e., the transition model depends on the level $h \in [H]$), where the lower bound of sample complexity is at least linear in H because the complexity of MDP is larger (Jin et al., 2018; Zhang et al., 2020c). In this paper, we consider the episodic MDP with a stationary transition, that is, the transition model is independent of the level. This is often considered to be a more realistic model than the non-stationary transition model. Furthermore, for non-stationary episodic MDP, our algorithm could also provide a reward-free sample complexity of $\tilde{O}\left(\frac{HSA}{\epsilon^2}(\log(\frac{1}{\delta}) + S)\right)$, which matches current best result in (Ménard et al., 2020) up to logarithmic terms (see Section 6 for more discussion).*

2. Related Work

We review relevant works in this section. Comparisons between our algorithm and existing ones on reward-free RL are provided in Table 1.

Algorithm	Sample Complexity	Non-unif. Reward	Log H
RF-RL-EXPLORE (Jin et al., 2020)	$\tilde{O}\left(\frac{H^5S^2A}{\epsilon}\log^3(\frac{1}{\delta}) + \frac{H^3S^2A}{\epsilon^2}\log(\frac{1}{\delta})\right)$	No	No
RF-UCRL (Kaufmann et al., 2020)	$\tilde{O}\left(\frac{H^2SA}{\epsilon^2}(\log(\frac{1}{\delta}) + S)\right)$	No	No
RF-EXPRESS (Ménard et al., 2020)	$\tilde{O}\left(\frac{HSA}{\epsilon^2}(\log(\frac{1}{\delta}) + S)\right)$	No	No
SSTP This Work	$\tilde{O}\left(\frac{SA}{\epsilon^2}(\log(\frac{1}{\delta}) + S)\right)$	Yes	Yes
Lower Bound (Jin et al., 2020)	$\Omega\left(\frac{SA}{\epsilon^2}(\log(\frac{1}{\delta}) + S)\right)$	-	-

Table 1. Sample complexity comparisons for state-of-the-art episodic RL algorithms. See Section 2 for discussions on this table. $\widetilde{O}$ omits logarithmic factors on $S, A, H, 1/\epsilon$ but not $1/\delta$. **Sample Complexity**: number of episodes to find an ϵ-suboptimal policy. **Non-unif. Reward**: Yes means the bound holds under Assumption 2 (allows non-uniformly bounded reward), and No means the bound only holds under Assumption 1. **Log H**: Whether the sample complexity bound depends logarithmically on H instead of polynomially on H.

Reward-dependent exploration In reward-dependent exploration, the agent aims to learn an ϵ-optimal policy under a fixed reward. Some papers assumed there is a generative model which can be queried to provide a sample for any state-action pair (s, a) (Kearns & Singh, 1999; Azar et al., 2013; Sidford et al., 2018; Agarwal et al., 2019; Li et al., 2020), and the sample complexity is defined as the number of queries needed to compute an ϵ-optimal policy. In the online setting (Brafman & Tennenholtz, 2003; Kakade, 2003; Dann & Brunskill, 2015; Dann et al., 2017; 2019; Azar et al., 2017; Jin et al., 2018; Zanette & Brunskill, 2019; Kaufmann et al., 2020; Zhang et al., 2020c; Wang et al., 2020a; Zhang et al., 2020b), the agent starts from a fixed initial distribution in each episode, and collects a trajectory by interacting with the environment. Then the sample complexity is given by the number of episodes that are necessary to learn an ϵ-optimal policy. The state-of-the-art result by Zhang et al. (2020b) requires $\widetilde{O}\left(\frac{SA}{\epsilon^2} + \frac{S^2A}{\epsilon}\right)$ number of episodes.

Reward Assumption and Dependency on H For the reward, the widely adopted assumption is $r_h \in [0,1]$ for all $h \in [H]$, which implies the total reward $\sum_{h=1}^{H} r_h \in [0, H]$. However, as argued in (Kakade, 2003; Jiang & Agarwal, 2018), the characterization of sample complexity should be independent of the scaling, i.e., the target suboptimality $\epsilon \in (0,1)$ should be a *relative* quantity to measure the performance of an algorithm. To this end, we need to scale the total reward within $[0, 1]$. Then the assumption becomes:

Assumption 1 (Uniformly Bounded Reward)**.** *The reward satisfies that $r_h \in [0, 1/H]$ for all $h \in [H]$.*

Compared to Assumption 1, the totally-bounded reward assumption (Assumption 2) is more general. Therefore, any

[2] The original bound is for the case the total reward is bounded by H, and here we scale down the total reward by a factor of H.

upper bound under Assumption 2 implies an upper bound under Assumption 1. In the view of practice, because environments under Assumption 2 can have high one-step reward, it is more natural to consider Assumption 2 in environments with sparse rewards, such as the Go game, which are often considered to be puzzling. In the view of theoretical basis, it is more complicated to design efficient algorithms under Assumption 2 due to the global structure of the reward. [3]

Recent work (Zanette & Brunskill, 2019; Wang et al., 2020a; Zhang et al., 2020b) made essential progress in reward-dependent exploration under Assumption 2, and obtained sample complexity bounds that only scale *logarithmically* with H. Zanette & Brunskill (2019) showed the main term (with respect to $1/\epsilon^2$) does not depend on H. Wang et al. (2020a) proved a sample complexity bound of $\tilde{O}\left(\frac{S^5A^4}{\epsilon^3}\right)$ despite suffering an exponential computational cost, and later (Zhang et al., 2020b) achieved a nearly sharp sample complexity bound of $\tilde{O}\left(\frac{SA}{\epsilon^2}+\frac{S^2A}{\epsilon}\right)$ with a computationally efficient algorithm. We use some technical ideas from (Zhang et al., 2020b) (cf. Section 4.2). However, because the problem settings are different, we need new techniques to establish a nearly tight sample complexity bound for reward-free exploration.

Reward-Free RL The main algorithm in (Jin et al., 2020) assigns only non-zero reward for each state at every turn, and utilizes a regret minimization algorithm EULER (Zanette & Brunskill, 2019) to visit each state as much as possible. Since their algorithm only learns one state each time, their sample complexity bound is not tight with respect to H. Kaufmann et al. (2020) proposed RF-UCRL to achieve sample complexity of $\tilde{O}\left(\frac{SAH^2}{\epsilon^2}(S+\log(\frac{1}{\delta}))\right)$ by building upper confidence bounds for any reward function and any policy, and then taking the greedy policy accordingly. The later work by Ménard et al. (2020) constructed an exploration bonus of $\frac{1}{n(s,a)}$ instead of the classical exploration bonus of $\frac{1}{\sqrt{n(s,a)}}$, where $n(s,a)$ is the visit count of (s,a). Based on the novel bonus, they achieve sample complexity of $\tilde{O}(\frac{SAH}{\epsilon^2}(S+\log(\frac{1}{\delta}))$. Recently, these results have been extended to linear function approximation settings (Wang et al., 2020b; Zanette et al., 2020). Reward-free exploration is also related to another setting, reward-agnostic RL, in which N reward functions are considered in the planning phase. Zhang et al. (2020a) provided an algorithm which achieves $\tilde{O}\left(\frac{H^3SA\log(N)\log(1/\delta)}{\epsilon^2}\right)$ sample complexity. See Table 1 for a summary.

3. Preliminaries

Notations. Throughout this paper, we define $[N]$ to be the set $\{1,2,\ldots,N\}$ for $N\in\mathbb{Z}_+$. We use $\mathbb{I}[\mathcal{E}]$ to denote the indicator function for an event $\mathcal{E}$, i.e., $\mathbb{I}[\mathcal{E}]=1$ if $\mathcal{E}$ holds and $\mathbb{I}[\mathcal{E}]=0$ otherwise. For notational convenience, we set $\iota=\ln(2/\delta)$ throughout the paper. For two n-dimensional vectors x and y, we use xy to denote $x^\top y$, use $\mathbb{V}(x,y)=\sum_i x_iy_i^2-(\sum_i x_iy_i)^2$, and use x^2 to denote the vector $[x_1^2,x_2^2,...,x_n^2]^\top$ for $x=[x_1,x_2,...,x_n]^\top$. For two vectors x,y, $x\geq y$ denotes $x_i\geq y_i$ for all $i\in[n]$ and $x\leq y$ denotes $x_i\leq y_i$ for all $i\in[n]$. We use $\mathbf{1}$ to denote the S-dimensional vector $[1,...,1]^\top$ and $\mathbf{1}_s$ to denote the S-dimensional vector $[0,...,1,...,0]^\top$ where the only non-zero element is in the s-th dimension.

Episodic Reinforcement Learning. We first describe the setting for standard episodic RL. A finite-horizon Markov Decision Process (MDP) is a tuple $M=(\mathcal{S},\mathcal{A},P,R,H,\mu)$. $\mathcal{S}$ is the finite state space with cardinality S. $\mathcal{A}$ is the finite action space with cardinality A. $P:\mathcal{S}\times\mathcal{A}\rightarrow\Delta(\mathcal{S})$ is the transition operator which takes a state-action pair and returns a distribution over states. For $h=1,2,...,H$, $R_h:\mathcal{S}\times\mathcal{A}\rightarrow\Delta(\mathbb{R})$ is the reward distribution with a mean function $r_h:\mathcal{S}\times\mathcal{A}\rightarrow\mathbb{R}$. $H\in\mathbb{Z}_+$ is the planning horizon (episode length). $\mu\in\Delta(\mathcal{S})$ is the initial state distribution. P, R and μ are unknown. For notational convenience, we use $P_{s,a}$ and $P_{s,a,s'}$ to denote $P(\cdot|s,a)$ and $P(s'|s,a)$ respectively.

A policy π chooses an action a based on the current state $s\in\mathcal{S}$ and the time step $h\in[H]$. Formally, we define $\pi=\{\pi_h\}_{h=1}^H$ where for each $h\in[H]$, $\pi_h:\mathcal{S}\rightarrow\mathcal{A}$ maps a given state to an action. The policy π induces a (random) trajectory $\{s_1,a_1,r_1,s_2,a_2,r_2,\ldots,s_H,a_H,r_H\}$, where $s_1\sim\mu$, $a_1=\pi_1(s_1)$, $r_1\sim R(s_1,a_1)$, $s_2\sim P(\cdot|s_1,a_1)$, $a_2=\pi_2(s_2)$, etc. We use $\mathbb{E}_{\pi,M}[\cdot]$ and $\mathbb{P}_{\pi,M}[\cdot]$ to denote respectively the expectation and probability under policy π with respect to the MDP M, and omit M when M is clear in the context,

The goal of RL is to find a policy π that maximizes the expected total reward, i.e. $\max_\pi\mathbb{E}_\pi\left[\sum_{h=1}^H r_h\right]$ where the expectation is over the initial distribution state μ, the transition operator P and the reward distribution R.

As for scaling, we make the following assumption about the reward. As we discussed in Section 2, this is a more general assumption than the assumption made in most previous works.

[3] Under Assumption 2, the reward still satisfies $r_h\in[0,1]$, so if an algorithms enjoys an sample complexity bound under Assumption 1, scaling up this bound by an H^2 for PAC bound, one can also obtain a bound under Assumption 2. However, this reduction is highly suboptimal in terms of H, so when comparing with existing results, we display their original results and add a column indicating whether the bound is under Assumption 2 or Assumption 1.

Assumption 2 (Bounded Total Reward)**.** *The reward satisfies that $r_h \geq 0$ for all $h \in [H]$. Besides, $\sum_{h=1}^{H} r_h \leq 1$ almost surely.*

Given a policy π, a level $h \in [H]$ and a state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, the Q-function is defined as:

$$Q_h^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, a_h = a \right].$$

Similarly, given a policy π, a level $h \in [H]$, the value function of a given state $s \in \mathcal{S}$ is defined as:

$$V_h^\pi(s) = \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, \right].$$

Then Bellman equation establishes the following identities for policy π and $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$

$$Q_h^\pi(s,a) = r_h(s,a) + P_{s,a}^\top V_{h+1}^\pi \quad V_h^\pi(s) = \sum_a \pi(a|s) Q_h^\pi(s,a)$$

Throughout the paper, we let $V_{H+1}(s) = 0$ and $Q_{H+1}(s,a) = 0$ for notational simplicity. We use Q_h^* and V_h^* to denote the optimal Q-function and V-function at level $h \in [H]$, which satisfies for any state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q_h^*(s,a) = \max_\pi Q_h^\pi(s,a)$ and $V_h^*(s) = \max_\pi V_h^\pi(s)$.

Dataset A dataset $\mathcal{D} = \{(s_h^k, a_h^k, s_{h+1}^k)\}_{(h,k) \in [H] \times [K]}$ consists of trajectories of K episodes. We also define $\{N_{s,a,s'}(\mathcal{D})\}_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}}$ to be the visitation count and $\{P_{s,a,s'}(\mathcal{D}) = \frac{N_{s,a,s'}(\mathcal{D})}{\sum_{\tilde{s}} N_{s,a,\tilde{s}}(\mathcal{D})}\}_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}}$ to be the empirical transition probability computed by $\mathcal{D}$, where $P_{s,a,s'}(\mathcal{D})$ is defined as $\frac{1}{S}$ if $\sum_{\tilde{s}} N_{s,a,\tilde{s}}(\mathcal{D}) = 0$. We further define $N_{s,a}(\mathcal{D}) = \sum_{\tilde{s}} N_{s,a,\tilde{s}}(\mathcal{D})$ and $P_{s,a}(\mathcal{D})$ be the vector such that the value of its s'-th dimension is $P_{s,a,s'}(\mathcal{D})$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. With the notation defined above, we let $N(\mathcal{D})$ and $P(\mathcal{D})$ be respectively the shorthands of $\{N_{s,a}(\mathcal{D})\}_{(s,a) \in \mathcal{S} \times \mathcal{A}}$ and $\{P_{s,a}(\mathcal{D})\}_{(s,a) \in \mathcal{S} \times \mathcal{A}}$.

Reward-Free Reinforcement Learning Now we formally describe reward-free RL. Let $\epsilon, \delta \in (0,1)$ be the thresholds of sub-optimality and failure probability. Reward-free RL consists of two phases. In the exploration phase, the algorithm collects a dataset $\mathcal{D}$ by interacting with the environment without reward information, and in the planning phase, given any reward function r satisfying Assumption 2, the agent is asked to output an ϵ-optimal policy with probability at least $1 - \delta$. The performance of an algorithm is measured by how many episodes K used in the exploration phase to make sure the planning phase succeeds.

4. Technique Overview

The proposed algorithm has two main components: the sampling phase and the planning phase. At a high level, we first propose a sufficient condition (see Condition 2) for the agent to use the collect samples to learn an ϵ-optimal policy for any reward function satisfying Assumption 2. Then we apply a modified version of Rmax (Brafman & Tennenholtz, 2003) to obtain samples to satisfy Condition 2 in the sampling phase.

4.1. Planning Phase

4.1.1. A Tight Sufficient Condition

To obtain a near-optimal policy for any given reward function, a sufficient condition is to collect $\overline{N}$ samples for each (s,a) pair, where $\overline{N}$ is some polynomial function of S, A and $1/\epsilon$. However, some (s,a) pairs might be rarely visited with any policy so it is hard to get enough samples for such pairs. To address this problem, we observe that such state-action pairs contribute little to the accumulative reward. As mentioned in (Jin et al., 2020), if the maximal expected visit count of (s,a) is $\lambda(s,a)$, then $\overline{N}\lambda(s,a)$ samples of (s,a) is sufficient for us to compute a good policy. Instead of considering each (s,a) pair one by one, we hope to divide the state-action space into a group of disjoint subsets, such that the maximal expect visit count of each subset is proportionally to minimal visit count in this subset. This poses a sufficient condition for the dataset in the plan phase.

Condition 1. *Let $K = \lfloor \log_2(2H/\epsilon) \rfloor$. Given the dataset $\mathcal{D}$, the state-action space $\mathcal{S} \times \mathcal{A}$ could be divided into $K+1$ subsets $\mathcal{S} \times \mathcal{A} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_{K+1}$, such that,*
(1) For any $1 \leq i \leq K$, $N_{s,a}(\mathcal{D}) \geq N_i := 4 \cdot \frac{SH\iota}{2^i \epsilon^2}$ for any $(s,a) \in \mathcal{X}_i$;
(2) For each $1 \leq i \leq K+1$, it holds that $\sup_\pi \mathbb{E}_\pi \left[\sum_{h=1}^{H} \mathbb{I}\left[(s_h, a_h) \in \mathcal{X}_i\right] \right] \leq \frac{H}{2^i}$.

The following proposition shows this condition is sufficient. The proof of Proposition 1 is postpone to Appendix D.

Proposition 1. *Suppose Condition 1 holds for the dataset $\mathcal{D}$. Given any reward function r satisfying Assumption 2, with probability $1 - 4S^2A(\log_2(T_0 H) + 2)\delta$,* Q-COMPUTING$(P(\mathcal{D}), N(\mathcal{D}), r)$ *(see Algorithm* **??***) returns an ϵ-optimal policy.*

In previous work on reward-free exploration (Jin et al., 2020; Kaufmann et al., 2020; Ménard et al., 2020), sufficient conditions similar to Condition 1 have been proposed to prove efficient reward-free exploration. However, to obtain a dataset satisfying Condition 1, the sample complexity bound is at least polynomial in H in the worst case, which is the main barrier of previous work. We give a simple counter-example to explain why Condition 1 is hard to be satisfied without a poly (H) number of episodes. Suppose there is a state

$\tilde{s}$, such that for any other $(s,a) \in \mathcal{S} \times \mathcal{A}$ $P_{s,a,\tilde{s}} = \epsilon_1$, and for any action a $P_{\tilde{s},a,\tilde{s}} = 1$. Direct computation gives that $\lambda(s) := \sum_a \lambda(s,a) = \Theta(H^2\epsilon_1)$. However, the probability that the agent never visit $\tilde{s}$ in N episodes is at least $(1 - H\epsilon_1)^N \approx e^{-NH\epsilon_1} \approx e^{-\frac{N\lambda(s)}{H}}$. In the case $N \ll H$, the expected visit count in N episodes is $N\lambda(s)$, while the empirical visit count could be 0 with constant probability, which implies the expected visited number and the empirical visit can be very different in the $N = o(H)$ regime.

To address this problem, we observe that in the example above, the probability the agent reaches $\tilde{s}$ is relatively small. If we simply ignore $\tilde{s}$, the regret due to this ignorance is at most $O(H\epsilon_1) = O(\lambda(s)/H)$ instead of original regret bound of $O(\lambda(s))$. This poses our main novel condition to plan for a near-optimal policy given any reward function satisfying Assumption 2. This is one of our key technical contributions.

Condition 2. *Recall* $K = \lfloor \log_2(2H/\epsilon) \rfloor$. *The state-action space* $\mathcal{S} \times \mathcal{A}$ *could be divided into* $K+1$ *subsets* $\mathcal{S} \times \mathcal{A} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_{K+1}$*, such that,*
(1) $N(s,a) \geq N_i = 4 \cdot \frac{H(\iota + 6S\ln(SAH/\epsilon))}{2^i\epsilon^2}$ *for any* $(s,a) \in \mathcal{X}_i$ *for* $1 \leq i \leq K$*;*
(2) Recall $Z_i = \max\{\min\{\frac{H}{2^i\epsilon}, H\}, 1\}$ *for each* $1 \leq i \leq K+1$*. For each* $1 \leq i \leq K+1$*, it holds that* $\sup_\pi \mathbb{P}_\pi[\sum_{h=1}^H \mathbb{I}[(s_h,a_h) \in \mathcal{X}_i] > Z_i] \leq \epsilon$ *and* $\sup_\pi \mathbb{E}_\pi\left[\min\{\sum_{h=1}^H \mathbb{I}[(s_h,a_h) \in \mathcal{X}_i], Z_i\}\right] \leq \frac{H}{2^i}$.

Under Condition 2, the state-action space are divided into $K+1$ subsets according to their visit counts. For the state-action pairs with visit counts in $[N_i, N_{i-1})$, different with the second requirement in Condition 1 we require that the maximal *truncated* expected visit count is strictly bounded proportionally to their visit counts. Let $\mathcal{E}_i$ be the set of trajectories satisfying that $\sum_{h=1}^H \mathbb{I}[(s_h,a_h) \in \mathcal{X}_i] > Z_i$. We also requires that the probability of $\mathcal{E}_i$ is no larger than ϵ for any policy. In fact, we directly pay loss of $\sup_\pi \mathbb{P}_\pi[\mathcal{E}_i]$ due to ignoring $\mathcal{E}_i$ when computing the value function. On the other hand, Z_i is far less than H when i is relatively large, which enables us to collect samples to satisfy Condition 2.

The selection of Z_i is quite non-trivial. On one hand, we need Z_i large enough so that it is possible to ensure $\sup_\pi \mathbb{P}_\pi\left[\sum_{h=1}^H \mathbb{I}[(s_h,a_h) \in \mathcal{X}_i]\right]$ no larger than ϵ (for example, by choosing $Z_i = H+1$, we can easily make this probability 0), and on the other hand, we need Z_i small enough to get rid of polynomial dependence on H. One possible solution is to set Z_i to scale linear as the maximal expected visit count of $\mathcal{X}_i$, which plays a crucial role in the analysis.

4.1.2. Planning using an Auxiliary MDP

Suppose Condition 2 holds for some dataset $\mathcal{D}$ with the partition $\{\mathcal{X}_i\}_{i=1}^{K+1}$. Because we only require the truncated maximal expected visit is properly bounded in Condition 2, standard planning method cannot work trivially. The main difficulty here is that, to apply the bounds of $\sup_\pi \mathbb{E}_\pi\left[\max\{\sum_{h=1}^H \mathbb{I}[(s_h,a_h) \in \mathcal{X}_i], Z_i\}\right]$ and $\sup_\pi \mathbb{P}_\pi[\mathcal{E}_i]$, we should set the reward 0 if $\mathcal{X}_i$ has been visited for more than Z_i times in an episode. A naive solution is to encode the visit counts of $\{\mathcal{X}_i\}_{i=1}^{K+1}$ into the state space. However, in this approach, the size of the new state space is exponential in S, which leads to exponential computational cost. Due to the reason above, to our best of knowledge, no existing algorithms can direct learn such a *truncated* MDP.

To address this problem, we consider an auxiliary MDP $\mathcal{M}^\dagger = \left\langle \mathcal{S} \cup s_{\text{end}}, \mathcal{A}, r^\dagger, \hat{P}^\dagger, \mu^\dagger \right\rangle$. Here s_{end} is an additional absorbing state. The reward function $r^\dagger$ is the same as r except for an additional column 0 for s_{end}, and the transition probability $\hat{P}^\dagger$ is given by $P^\dagger_{s,a} = (1 - \frac{1}{Z_i})\hat{P}_{s,a} + \frac{1}{Z_i}\mathbf{1}_{s_{\text{end}}}$ for any $(s,a) \in \mathcal{X}_i$ and $\hat{P}_{s_{\text{end}},a} = \mathbf{1}_{s_{\text{end}}}$ for any a. In words, we add an absorbing state to the original MDP, such that the agent would fall into s_{end} if it visit $\mathcal{X}_i$ for Z_i times in expectation for some $1 \leq i \leq K+1$. Instead of learning the *truncated* MDP, we consider a *soft-truncated* MDP, which exponentially reduces computational cost. For more details, we refer the readers to Section 5.2.

4.2. Sampling Phase

Having identified the sufficient condition, we need to design an algorithm to collect a set of samples that satisfy this condition.

We make the partition $\mathcal{S} \times \mathcal{A} = \cup_{i=1}^{K+1} \mathcal{X}_i$ by specifying $\mathcal{X}_i$ for $i = 1, 2, ..., K+1$ one by one. We divide the learning process into K stages. Take the first stage as an example. At the beginning of the first stage, we assign reward 1 to all $(s,a) \in \mathcal{S} \times \mathcal{A}$, and proceeds to learn with this reward. Like RMAX, whenever the visit count of some (s,a) pair is equal to or larger than N_1, we say this (s,a) is *known* and set $r(s,a) = 0$. We will discuss the problem of regret minimization for this MDP with time-varying reward function later and simply assume the regret is properly bounded. Defining $\mathcal{X}_1$ be the set of *known* state-action pairs after the first stage, the statements in Condition 2 holds trivially. Beside, the length of each stage is properly designed. Combining this with the bound of regret, we show that the maximal expected visit count of the *unknown* state-action pairs is properly bounded. Because $\mathcal{X}_2 \subset (\mathcal{X}_1)^C$, we learn that the second part in the second statement in Condition 2 holds for $\mathcal{X}_2$. We then continue to learn the second subset

$\mathcal{X}_2$ and so on.

Note that in arguments above, we do not introduce Z_i because $Z_i = H$ for the beginning stages by definition. In the case $Z_i < H$, there are two major problems.

The regret minimization algorithm Most regret minimization algorithms such (Azar et al., 2017; Zanette & Brunskill, 2019) works in the regime $Z_i = H$, where no truncation occurs. However, in the case $Z_i \ll H$, the regret bounds by these algorithms depends on H polynomially. To address this problem, we constructed an expanded MDP with truncated cumulative reward (see definition in Section 5.1), where the Q-function is strictly bounded by Z_i. In this way, we obtain desired regret bounds. We would like to mention that our algorithm is somewhat similar to recent work (Zhang et al., 2020b) which addresses the regret minimization problem with a total-bounded reward function. More precisely, after re-scaling, the reward function in our regret minimization problem is also total bounded by 1 and each single reward is bounded by $1/Z_i$. Although the reward function might vary in different episodes, we can provide efficient regret bounds in a similar way to the analysis in (Zhang et al., 2020b).

Bound of $\mathbb{P}\left[\mathcal{E}_{i+1}\right]$ Recall that $\mathcal{E}_i$ is the set of trajectories satisfying that $\sum_{h=1}^{H}\mathbb{I}[(s_h,a_h)\in\mathcal{X}_i] > Z_i$. Define $\mathcal{Y}_{i+1} = (\mathcal{S}\times\mathcal{A})/(\mathcal{X}_1\cup\ldots\cup\mathcal{X}_i)$ for $i \geq 1$ and $\mathcal{Y}_1 = \mathcal{S}\times\mathcal{A}$. It is clear that $\mathcal{W}_i$ is decreasing in i. By the upper bound of regret (see Lemma 3), we show that the maximal truncated expected visit count $\sup_\pi \mathbb{E}_\pi\left[\min\{\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h)\in\mathcal{Y}_i\right], Z_i\}\right]$ is properly bounded. Noting that $Z_{i+1}\leq Z_i$, we have that

$$\begin{aligned}
\mathbb{P}\left[\mathcal{E}_{i+1}\right] &\leq \sup_\pi \mathbb{P}_\pi\left[\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h)\in\mathcal{Y}_{i+1}\right] > Z_{i+1}\right] \\
&\leq \sup_\pi \mathbb{P}_\pi\left[\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h)\in\mathcal{Y}_{i+1}\right] > Z_i\right] \\
&\quad + \frac{1}{Z_{i+1}}\sup_\pi \mathbb{E}_\pi\left[\min\left\{\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h)\in\mathcal{Y}_{i+1}\right], Z_i\right\}\right] \\
&\leq \sup_\pi \mathbb{P}_\pi\left[\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h)\in\mathcal{Y}_{i}\right] > Z_i\right] \\
&\quad + \frac{1}{Z_{i+1}}\sup_\pi \mathbb{E}_\pi\left[\min\left\{\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h)\in\mathcal{Y}_{i+1}\right], Z_i\right\}\right].
\end{aligned} \tag{1}$$

By properly choosing the value of Z_i, we show that the second term in **RHS** of (1) could be bounded by $O(\epsilon)$. Then by induction, we show that $\mathbb{P}\left[\mathcal{E}_{i+1}\right] \leq K\epsilon$. Noting that $K = \lfloor\log_2(2H/\epsilon)\rfloor$ is a logarithmic term, we can bound the probability of $\mathbb{P}\left[\mathcal{E}_{i+1}\right]$ properly.

Algorithm 1 MAIN ALGORITHM: STAGED SAMPLING + TRUNCATED PLANNING

```
1: $(\mathcal{D}, \{\mathcal{X}_i\}_{i=1}^{K+1}) \leftarrow$ STAGED SAMPLING (Algorithm 2);
2: Given any reward function $r$ satisfying Assumption 2, return $\pi \leftarrow$TRUNCATED PLANNING$(\mathcal{D}, \{\mathcal{X}_i\}_{i=1}^{K+1}, r)$ (Algorithm 3).
```

Following the arguments above, we set the number of episodes in each stage to be $T_0 := C_1\frac{SA(\iota+S)l}{\epsilon^2}$ where C_1 is some large enough constant and l is a poly-logarithmic term in $(S, A, H, 1/\epsilon)$. At the beginning of an episode in the i-th stage, we assign reward 1 to a state-action pair if its visit number is less than N_i and otherwise 0. We then apply Algorithm 4 to minimize regret in each stage, and finally obtain $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_{K+1}$.

For more technical details, we refer the reader to Section 5.1 and 5.2.

5. Proof of Theorem 1

Similar as in Section 4, our proof consists of two parts, one for the sampling phase and another for the planning phase. We propose the main lemmas for these two parts respectively.

Lemma 1. *By running Algorithm 2, with probability* $1 - K\left(2(\log_2(T_0H)+1)\log_2(T_0H) + 4S^2A(\log_2(H)+2)\right)\delta$, *we can collect a dataset* $\mathcal{D}$ *and obtain the partition* $\{X_i\}_{i=1}^{K+1}$ *such that Condition 2 holds for the collected dataset* $\mathcal{D}$ *with the partition* $\{X_i\}_{i=1}^{K+1}$. *Besides, we consumes at most* $KT_0 = \tilde{O}(\frac{SA(\iota+S)}{\epsilon^2})$ *episodes to run Algorithm 2.*

Lemma 2. *Assuming Condition 2 holds for the collected dataset* $\mathcal{D}$ *with partition* $\{\mathcal{X}_i\}_{i=1}^{K+1}$ *, with probability* $1 - 4S^2AKT_0H(\log_2(T_0H)+2)\delta$, *Algorithm 3 can compute an* ϵ*-optimal policy using these samples for any reward function* r *satisfying Assumption 2.*

Theorem 1 follows by combining Lemma 1 with Lemma 2 and replacing δ by $\mathrm{poly}(S, A, 1/\epsilon, \log(H))\delta$. The rest part of this section is devoted to the proofs of Lemma 1 and Lemma 2.

5.1. Sampling Phase: Proof of Lemma 1

As mentioned in Section 4, we aim to collect samples such that Condition 2 holds. Our algorithm proceeds in $K+1$ stages, where each stage consists of T_0 episodes. Therefore, at most $KT_0 = \tilde{O}(\frac{SA(\iota+S)}{\epsilon^2})$ episodes are needed to run Algorithm 2. In an episode, saying the k-th episode in the i-th stage, we define $\mathcal{Y}^k = \{(s,a)|N^k(s,a) < N_i\}$ to be the set of unknown state-action pairs. In particular, we define $\mathcal{Y}_i := \mathcal{Y}^{\bar{k}(i)}$ where $\bar{k}(i)$ is the first episode in

Algorithm 2 STAGED SAMPLING

```
Initialize: $\mathcal{D} \leftarrow \emptyset$, $\mathcal{Y}_1 \leftarrow \mathcal{S} \times \mathcal{A}$
for $i = 1, 2, ..., K$ do
    $(D_i, \mathcal{Y}_{i+1}) \leftarrow$ TRVRL$(i, \mathcal{Y}_i)$;
    $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$;
    $\mathcal{X}_i \leftarrow \mathcal{Y}_i / \mathcal{Y}_{i+1}$
end for
$\mathcal{X}_{K+1} \leftarrow \mathcal{Y}_{K+1}$;
Return $(\mathcal{D}, \{\mathcal{X}_i\}_{i=1}^{K+1})$.
```

the i-th stage. Note that $\mathcal{X}_i$ is defined as $\mathcal{Y}_i/\mathcal{Y}_{i+1}$ and $\mathcal{Y}_1 = \mathcal{S} \times \mathcal{A}$, which means $\mathcal{Y}_{i+1} = \mathcal{Y}_1/(\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_i) = (\mathcal{S} \times \mathcal{A})/(\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_i)$. So we have that $\mathcal{W}_i = \mathcal{Y}_i$ for $i \geq 1$.

To learn the *unknown* state-action pairs, we adopt the idea of Rmax by setting reward function to be $r(s, a) = \mathbb{I}\left[(s, a) \in \mathcal{Y}^k\right]$. However, by the definition of Condition 2, it suffices to assign reward 1 to the first Z_i visits to $\mathcal{Y}_i$. So it corresponds to learn a policy to maximize

$$\mathbb{E}_\pi \left[\min\left\{\sum_{h=1}^{H} \mathbb{I}\left[(s_h, a_h) \in \mathcal{Y}^k\right], Z_i\right\}\right].$$

To address this learning problem, we consider an expanded MDP $\mathcal{M}^k = \langle \mathcal{S}^k, \mathcal{A}^k, P^k, r^k, \mu^k \rangle$, where

$$\begin{aligned}
&\mathcal{S}^k = \mathcal{S} \times [Z_i + 1];\\
&\mathcal{A}^k = \mathcal{A};\\
&r^k(s, z, a) = \mathbb{I}[(s, a) \in \mathcal{Y}^k, z \leq Z_i], \forall (s, z, a) \in \mathcal{S}^k \times \mathcal{A}^k;\\
&P^k(s', z'|s, z, a) = P(s'|s, a) \cdot \Big(\mathbb{I}\left[z' = z + 1 \cap (s, a) \in \mathcal{Y}^k\right]\\
&\qquad + \mathbb{I}\left[z' = z \cap (s, a) \notin \mathcal{Y}^k\right]\Big), \forall (s, a) \in \mathcal{S} \times \mathcal{A}, z \in [Z_i];\\
&P^k(s', Z_i + 1|s, Z_i + 1, a) = P(s'|s, a), \forall (s, a) \in \mathcal{S} \times \mathcal{A};\\
&\mu^k(s, z) = \mu(s)\mathbb{I}[z = 1].
\end{aligned}$$

Roughly speaking, a state in $\mathcal{M}^k$ not only represent its position in $\mathcal{S}$, but also records the reward the agent has collected in current episode. We then define the pseudo regret in the i-th stage as:

$$R_i := \sum_{k \text{ in stage } i} \left(\sup_\pi \mathbb{E}_\pi \left[\sum_{h=1}^{H} r_h^k\right] - \sum_{h=1}^{H} r_h^k\right),$$

where r_h^k is a shorthand of $r^k(s_h^k, z_h^k, a_h^k)$. We show that R_i could be bounded properly in a similar way to (Zhang et al., 2020b).

Lemma 3. *For any $1 \leq i \leq K$, by running Algorithm 4 with input i, with probability $1 - (2(\log_2(T_0 H) + 1)\log_2(T_0 H) + 4SA(\log_2(Z_i) + 2))\delta$, R_i is bounded by*

$$\tilde{O}\left(Z_i\sqrt{SAT_0(S + \iota)} + Z_i SA(S + \iota)\right). \tag{2}$$

Algorithm 4 and the proof of Lemma 3 is postponed to Appendix. B due to limitation of space.

Let i be fixed. Recall that $\overline{k}_i$ is the first episode in the i-th stage. We define $u_k = \sup_\pi \mathbb{E}_\pi\left[\sum_{h=1}^{H} r_h^k\right]$, $\overline{u}^i = u_{\overline{k}(i)}$ and $\underline{u}^i = u_{\underline{k}(i)}$ where $\underline{k}(i)$ is the index of the last episode in the i-th stage. Because r^k is non-increasing in k, μ^k is also non-increasing in k. If $\underline{u}^i > \frac{H}{2^i}$, then by Lemma 3 and the definition of T_0 we have that there exists a constant C_2 and a poly-logarithmic factor l_2 in $(S, A, H, 1/\epsilon)$ such that

$$\begin{aligned}
&T_0\underline{u}_i - \sum_{k \text{ in stage } i} \sum_{h=1}^{H} r_h^k\\
&\leq C_2 l_2 \left(Z_i\sqrt{SAT_0(S + \iota)} + Z_i SA(S + \iota)\right).
\end{aligned}$$

By choosing $C_1 l_1 \geq 8C_2 l_2$, we have that

$$\begin{aligned}
&\sum_{k \text{ in stage } i} \sum_{h=1}^{H} r_h^k\\
&\geq T_0\underline{u}_i - C_2 l_2\left(Z_i\sqrt{SAT_0(S + \iota)} + Z_i SA(S + \iota)\right)\\
&\geq \frac{C_1 l_1}{2} \frac{SAH(\iota + 6S\ln(SAH/\epsilon))}{2^i \epsilon^2}\\
&> \frac{C_1}{8} SAN_i.
\end{aligned} \tag{3}$$

By choosing $C_1 \geq 16$, we learn that $\sum_{k \text{ in stage } i}\sum_{h=1}^{H} r_h^k > 2SAN_i$. On the other hand, each (s, a) could provide at most N_i rewards in the i-th stage, which implies that $\sum_{k \text{ in stage } i}\sum_{h=1}^{H} r_h^k \leq 2SAN_i$. This leads to a contradiction. We then have that $\underline{u}_i \leq \frac{H}{2^i}$.

Again because the reward function is non-increasing in k, we have that

$$\overline{u}_{i+1} \leq \underline{u}_i \leq \frac{H}{2^i}. \tag{4}$$

Define $p_i = \sup_\pi \mathbb{P}_\pi\left[\sum_{h=1}^{H} \mathbb{I}\left[(s_h, a_h) \in \mathcal{Y}_i\right] > Z_i\right]$. Then we have that

$$\begin{aligned}
p_{i+1} &\leq \sup_\pi \mathbb{P}_\pi \left[Z_{i+1} < \sum_{h=1}^{H} \mathbb{I}\left[(s_h, a_h) \in \mathcal{Y}_{i+1}\right] \leq Z_i\right]\\
&\quad + \sup_\pi \mathbb{P}_\pi \left[\sum_{h=1}^{H} \mathbb{I}\left[(s_h, a_h) \in \mathcal{Y}_{i+1}\right] > Z_i\right]\\
&\leq \mathbb{I}\left[2^{i+1}\epsilon \geq 1\right] \frac{u_i}{Z_{i+1}} + p_i\\
&\leq \epsilon + p_i.
\end{aligned}$$

By induction, we can obtain that $p_i \leq i\epsilon \leq (K+1)\epsilon$ and $\sum_{i=1}^{K} p_i \leq (K+1)^2\epsilon$. We claim that Condition 2 holds by defining $\mathcal{X}_i = \mathcal{Y}_i/\mathcal{Y}_{i+1}$ for $1 \leq i \leq K$ and $\mathcal{X}_{K+1} = \mathcal{Y}_{K+1}$.
(1) By the definition of $\mathcal{Y}_{i+1}$, we learn that for any $(s,a) \in \mathcal{X}_i$, $N(s,a) \geq 2N_{i+1} \geq N_i$.
(2) By the arguments above, we have that for each $1 \leq i \leq K+1$.

$$\sup_{\pi} \mathbb{P}_{\pi}\left[\sum_{h=1}^{H} \mathbb{I}\left[(s_h,a_h) \in \mathcal{X}_i\right] > Z_i\right]$$
$$\leq \sup_{\pi} \mathbb{P}_{\pi}\left[\sum_{h=1}^{H} \mathbb{I}\left[(s_h,a_h) \in \mathcal{Y}_i\right] > Z_i\right] = p_i \leq (K+1)\epsilon$$

and

$$\sup_{\pi} \mathbb{E}_{\pi}\left[\max\left\{\sum_{h=1}^{H} \mathbb{I}\left[(s_h,a_h) \in \mathcal{X}_i\right], Z_i\right\}\right]$$
$$\leq \overline{u}_i \leq \underline{u}_{i-1} \leq \frac{H}{2^{i-1}}.$$

Noting that there are exactly K stages and each stage consists of $T_0 = C_1 l_1 \frac{SA(\iota+6S\ln(SAH/\epsilon))\log_2(H)}{\epsilon^2}$ episodes, we prove that we can collect a dataset satisfying Condition 2 within

$$KT_0 = \tilde{O}(\frac{SA(S+\ln(1/\delta))}{\epsilon^2})$$

episodes.

5.2. Planning Phase: Proof of Lemma 2

Suppose we have a dataset $\mathcal{D}$ satisfying Condition 2 with partition $\{\mathcal{X}_i\}_{i=1}^{K+1}$. Let $\hat{P}_{s,a}$ and $N(s,a)$ be the shorthand of $P_{s,a}(\mathcal{D})$ and $N_(s,a)(\mathcal{D})$ respectively. Denote the empirical transition and visit count of (s,a) as $\hat{P}_{s,a}$ and $N(s,a)$ respectively.

As mentioned in Section 4, we consider the reward-free auxiliary MDP $\mathcal{M}^{\dagger} = \langle \mathcal{S} \cup \{s_{\text{end}}\}, \mathcal{A}, P^{\dagger}, \mu \rangle$. The transition function $P^{\dagger}_{s,a} = (1-\frac{1}{Z_i})P_{s,a} + \frac{1}{Z_i}\mathbf{1}_{s_{\text{end}}}$ for all $(s,a) \in \mathcal{X}_i$ and $P^{\dagger}_{s_{\text{end}},a} = \mathbf{1}_{s_{\text{end}}}$ for any a. We first show that for any policy, the value function of $\mathcal{M}^{\dagger}$ is $\widetilde{O}(\epsilon)$-closed to that of $\mathcal{M}$.

Lemma 4. *For any policy π and reward function r satisfying Assumption 2 and $r_{s_{\text{end}}} = 0$, define V_1^{π} and $V_1^{\dagger\pi}$ be the value function under $\mathcal{M}$ and $\mathcal{M}^{\dagger}$ with π respectively. We then have*

$$V_1^{\dagger\pi} \leq V_1^{\pi} \leq V_1^{\dagger\pi} + 4(K+1)^2\epsilon.$$

Algorithm 3 Truncated Planning

```
Input: The partition $\{\mathcal{X}_i\}_{i=1}^{K+1}$; the dataset $\mathcal{D}$; the reward function $r$.
Initialize: $r(s_{\text{end}},a) \leftarrow 0$; $P(\cdot|s_{\text{end}},a) \leftarrow \mathbf{1}_{s_{\text{end}}}$ for all $a \in \mathcal{A}$; $\hat{P} \leftarrow \{P_{s,a}(\mathcal{D})\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$; $N \leftarrow \{N_{s,a}(\mathcal{D})\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$;
for $i = 1,2,...,K+1$ do
    $\hat{P}^{\dagger}_{s,a} \leftarrow (1-\frac{1}{Z_i})\hat{P}_{s,a} + \frac{1}{Z_i}\mathbf{1}_{s_{\text{end}}}$ for any $(s,a) \in \mathcal{X}_i$;
end for
for $(s,a,h) \in \mathcal{S}\times\mathcal{A}\times[H]$ do
    $Q_h(s,a) \leftarrow 1$;
end for
for $(a,h) \in \mathcal{A}\times\mathcal{H}$ do
    $Q_h(s_{\text{end}},a) \leftarrow 0$;
end for
for $h = H, H-1, ..., 1$ do
    for $(s,a) \in \mathcal{S}\times\mathcal{A}$ do
        $b_h(s,a) \leftarrow 2\sqrt{\frac{\mathbb{V}(\hat{P}^{\dagger}_{s,a}, V_{h+1})\iota_1}{N(s,a)}} + \frac{29\iota_1}{3N(s,a)}$;   (5)
        $Q_h(s,a) \leftarrow \min\{r(s,a) + \hat{P}^{\dagger}_{s,a}V_{h+1} + b_h(s,a), 1\}$;   (6)
        $V_h(s) \leftarrow \max_a Q_h(s,a)$;
    end for
end for
$\pi_h(s) \leftarrow \arg\max_a Q_h(s,a), \forall s,h$;
Return $\pi$.
```

Instead of learning $\mathcal{M}$, we aim to learn $\mathcal{M}^{\dagger}$. Let $\hat{P}_{s,a}$ be the empirical transition computed by the collected samples. As described in Algorithm 3, for each $1 \leq i \leq K+1$, we define $\hat{P}^{\dagger}_{s,a} = (1-\frac{1}{Z_i})\hat{P}_{s,a} + \frac{1}{Z_i}\mathbf{1}_{s_{\text{end}}}$ for (s,a) in $\mathcal{X}_i$ and then update backward the Q-function and value function in an optimistic way. The final output policy π is induced by the Q-function above. We first verify the Q-function is optimistic, i.e.,

Lemma 5. *With probability $1-4S^2AKT_0H(\log_2(T_0H)+2)\delta$, $Q_h(s,a) \geq Q_h^{\dagger *}(s,a)$ for any (s,a,h).*

Without loss of generality, we assume $\mu = \mathbf{1}_{s_1}$. Now we bound the gap $V_1^*(s_1) - V_1^{\pi}(s_1)$. Let $\epsilon_1 = \min\{\frac{\iota}{T_0 H}, \frac{\iota^2}{T_0^2 H^3}\} \leq 1$, $\delta_1 = \delta\epsilon_1^S$ and $\iota_1 = \ln(1/\delta_1) \leq \iota S\ln(T_0^3 H^4/\iota^3)$.

Lemma 6. *With probability $1-4S^2AKT_0H(\log_2(T_0H)+2)\delta$, it holds that*

$$V_1^*(s_1) - V_1^{\pi}(s_1) \leq V_1(s_1) - V_1^{\pi}(s_1)$$
$$\leq \sum_{s,a,h} w_h(s,a,\pi)\beta_h(s,a)$$

where $w_h(s,a,\pi) := \mathbb{E}_{\pi,\mathcal{M}^{\dagger}}\left[\mathbb{I}[(s_h,a_h)=(s,a)]\right]$ *and*

$\beta_h(s,a) := \min\{6\sqrt{\frac{\mathbb{V}(P^{\dagger}_{s,a},V_{h+1})\iota_1}{N(s,a)}} + \frac{12\iota_1}{N(s,a)}, 1\}$.

Define $\omega_i^{\dagger}(\pi) = \sum_{(s,a)\in\mathcal{X}_i}\sum_h w_h(s,a,\pi)$ for $1 \leq i \leq K+1$.

Lemma 7. $w_i^{\dagger}(\pi) \leq O(\frac{HK}{2^i})$ *for* $1 \leq i \leq K$.

By Lemma 7, we further have

Lemma 8.

$$\sum_{s,a,h} w_h(s,a,\pi)\beta_h(s,a)$$

$$\leq O\left(K\epsilon\sqrt{2+2\sum_{s,a,h}\sum_{s,a,h} w_h(s,a,\pi)\beta_h(s,a)} + K^2\epsilon^2\right).$$

By Lemma 6 and solving the inequality $x \leq O(K\epsilon\sqrt{2+2x} + K^2\epsilon^2)$, we learn that

$$V_1^{\dagger *}(s_1) - V_1^{\dagger\pi}(s_1)$$
$$\leq \sum_{s,a,h} w_h(s,a,\pi)\beta_h(s,a) \leq O\left(K\epsilon + K^2\epsilon^2\right).$$

Recall that by Lemma 4, we have $|V_1^{\pi}(s_1) - V_1^{\dagger\pi}(s_1)| \leq O\left((K+1)^2\epsilon\right)$ and $|V_1^{\dagger *}(s_1) - V_1^{*}(s_1)| \leq O\left((K+1)^2\epsilon\right)$. We then finally conclude that

$$V_1^{*}(s_1) - V_1^{\pi}(s_1) \leq \sum_{s,a,h} w_h(s,a,\pi)\beta_h(s,a) \leq O\left(K^2\epsilon\right).$$

Since $K \leq \log_2(2H/\epsilon)$, we finish the proof by rescaling ϵ.

6. Discussions on Non-Stationary Episodic MDP

We claim that SSTP could provide a reward-free sample complexity of $\tilde{O}(\frac{SAH}{\epsilon^2}(\log(\frac{1}{\delta}) + S))$ with a slight modification. Because the analysis for the non-stationary episodic MDP is very similar to previous analysis, we only point out the major differences between the proofs and omit the details. We also follow previous notations.

Planning phase Let $N_h(s,a)$ denote the count of (s,a,h) in the dataset. For non-stationary episodic MDP, we propose a sufficient condition for the dataset to plan for a near-optimal policy given any reward function as follows.

Condition 3. *Recall* $K = \lfloor\log_2(2H/\epsilon)\rfloor$. $\mathcal{S}\times\mathcal{A}\times[H]$ *could be divided into* $K+1$ *subsets* $\mathcal{S}\times\mathcal{A}\times[H] = \mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_{K+1}$*, such that,*
(1) $N_h(s,a) \geq N_i = 4\cdot\frac{H(\iota+6S\ln(SAH/\epsilon))}{2^i\epsilon^2}$ *for any* $(s,a,h)\in\mathcal{X}_i$ *for* $1 \leq i \leq K$*;*
(2) Recall $Z_i = \max\{\min\{\frac{H}{2^i\epsilon}, H\}, 1\}$ *for each* $1 \leq i \leq K+1$*. For each* $1 \leq i \leq K+1$*, it holds that* $\sup_{\pi}\mathbb{P}_{\pi}[\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h,h)\in\mathcal{X}_i\right] > Z_i] \leq \epsilon$ *and* $\sup_{\pi}\mathbb{E}_{\pi}\left[\min\{\sum_{h=1}^{H}\mathbb{I}\left[(s_h,a_h,h)\in\mathcal{X}_i\right], Z_i\}\right] \leq \frac{H}{2^i}$.

Because the transition model at different layer could be different, instead of the requirement on $N(s,a)$, we ask $N_h(s,a) \geq N_i$ for $(s,a,h)\in\mathcal{X}_i$. Following the arguments in the proof of Lemma 2, we can show that with Condition 3, we can compute an ϵ-optimal policy for any reward function satisfying Assumption 2.

Sampling phase To learn a dataset satisfying Condition 3, in a similar way as Algorithm 4, we invoke TRVRL to learn $\mathcal{Y}_i$ for $1 \leq i \leq K+1$. The major difference is that HT_0 episodes are required for each stage. In this way, following the proof of Lemma 3, we have that the regret in the i-th stage is bounded by

$$\tilde{O}\left(Z_i\sqrt{SAH^2T_0(S+\iota)} + Z_iSAH(S+\iota)\right). \tag{7}$$

Compared to Lemma 3, we note that the bound in (7) has two additional $\sqrt{H}$ factors. The first $\sqrt{H}$ factor is because the length is multiplied by H and the second is due to the structure of non-stationary episodic MDP. By (7), we can further ensure that $\underline{u}_i \leq \frac{H}{2^i}$ by noting that when $\underline{u}_i > \frac{H}{2^i}$,

$$\sum_{k \text{ in stage } i}\sum_{h=1}^{H} r_h^k \geq \frac{C_1}{8}SAHN_i, \tag{8}$$

which contradicts to the fact that $\sum_{k \text{ in stage } i}\sum_{h=1}^{H} r_h^k \leq 2SAHN_i$ (because each (s,a,h) could be visited for at most N_i times).

Lower bound The current best lower bound is $\Omega(\frac{SA}{\epsilon^2}(H + S + \log(\frac{1}{\delta}))$ by the lower bounds in (Jin et al., 2020) and (Zhang et al., 2020c). It remains an open problem whether the $\tilde{O}(\frac{SAH}{\epsilon^2}(S + \log(\frac{1}{\delta})))$ upper bound is tight.

7. Conclusion

We give a new algorithm, SSTP, which enjoys a near-optimal sample complexity for reward-free RL. Importantly, we show the sample complexity only depends logarithmically on the planning horizon. Our algorithm relies on three new technical ideas: 1) A new sufficient condition for the dataset to plan for an ϵ-suboptimal policy ; 2) A new way to plan efficiently under the proposed condition using soft-truncated planning; 3) Constructing extended MDP to maximize the truncated accumulative rewards efficiently. In this way, we can divide the state-action space into different groups according to their maximal possible frequencies, which is especially suited for RL with growing batches.

Another important future direction is to generalize our algorithm to RL with function approximation. For example, can we obtain a near-optimal sample complexity for reward-free RL with linear function approximation (Wang et al., 2020b; Zanette et al., 2020)?

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. *arXiv preprint arXiv:1705.10528*, 2017.

Agarwal, A., Kakade, S., and Yang, L. F. Model-based reinforcement learning with a generative model is minimax optimal. *arXiv preprint arXiv:1906.03804*, 2019.

Altman, E. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Azar, M. G., Munos, R., and Kappen, H. J. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3): 325–349, 2013.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

Brafman, R. I. and Tennenholtz, M. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3(Oct):213–231, March 2003. ISSN 1532-4435.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2818–2826, 2015.

Dann, C., Lattimore, T., and Brunskill, E. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5717–5727, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Dann, C., Li, L., Wei, W., and Brunskill, E. Policy certificates: Towards accountable reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1507–1516, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Du, S. S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudík, M., and Langford, J. Provably efficient RL with rich observations via latent state decoding. *arXiv preprint arXiv:1901.09018*, 2019.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691, 2019.

Jiang, N. and Agarwal, A. Open problem: The dependence of sample complexity lower bounds on planning horizon. In *Conference On Learning Theory*, pp. 3395–3398, 2018.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. *arXiv preprint arXiv:2002.02794*, 2020.

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

Kaufmann, E., Ménard, P., Domingues, O. D., Jonsson, A., Leurent, E., and Valko, M. Adaptive reward-free exploration. *arXiv preprint arXiv:2006.06294*, 2020.

Kearns, M. J. and Singh, S. P. Finite-sample convergence rates for Q-learning and indirect algorithms. In *Advances in neural information processing systems*, pp. 996–1002, 1999.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. Breaking the sample size barrier in model-based reinforcement learning with a generative model. *arXiv preprint arXiv:2005.12900*, 2020.

Maurer, A. and Pontil, M. Empirical Bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E., and Valko, M. Fast active learning for pure exploration in reinforcement learning. *arXiv preprint arXiv:2007.13442*, 2020.

Miryoosefi, S., Brantley, K., Daume III, H., Dudik, M., and Schapire, R. E. Reinforcement learning with convex constraints. In *Advances in Neural Information Processing Systems*, pp. 14093–14102, 2019.

Sidford, A., Wang, M., Wu, X., Yang, L., and Ye, Y. Near-optimal time and sample complexities for solving Markov decision processes with a generative model. In *Advances in Neural Information Processing Systems*, pp. 5186–5196, 2018.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Wang, R., Du, S. S., Yang, L. F., and Kakade, S. M. Is long horizon reinforcement learning more difficult than short horizon reinforcement learning? *arXiv preprint arXiv:2005.00527*, 2020a.

Wang, R., Du, S. S., Yang, L. F., and Salakhutdinov, R. On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*, 2020b.

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

Zanette, A., Lazaric, A., Kochenderfer, M. J., and Brunskill, E. Provably efficient reward-agnostic navigation with linear value iteration. *arXiv preprint arXiv:2008.07737*, 2020.

Zhang, X., Singla, A., et al. Task-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2006.09497*, 2020a.

Zhang, Z., Ji, X., and Du, S. S. Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. *arXiv preprint arXiv:2009.13503*, 2020b.

Zhang, Z., Zhou, Y., and Ji, X. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *arXiv preprint arXiv:2004.10019*, 2020c.

Zhang, Z., Zhou, Y., and Ji, X. Model-free reinforcement learning: from clipped pseudo-regret to sample complexity. *arXiv preprint arXiv:2006.03864*, 2020d.