Safe Reinforcement Learning with Linear Function Approximation

Sanae Amani [1] **Christos Thrampoulidis** [2] **Lin F. Yang** [1]

Abstract

Safety in reinforcement learning has become increasingly important in recent years. Yet, existing solutions either fail to strictly avoid choosing unsafe actions, which may lead to catastrophic results in safety-critical systems, or fail to provide regret guarantees for settings where safety constraints need to be learned. In this paper, we address both problems by first modeling safety as an unknown linear cost function of states and actions, which must always fall below a certain threshold. We then present algorithms, termed SLUCB-QVI and RSLUCB-QVI, for finite-horizon Markov decision processes (MDPs) with linear function approximation. We show that SLUCB-QVI and RSLUCB-QVI, while with *no safety violation*, achieve a $\tilde{\mathcal{O}}\left(\kappa\sqrt{d^3H^3T}\right)$ regret, nearly matching that of state-of-the-art unsafe algorithms, where H is the duration of each episode, d is the dimension of the feature mapping, κ is a constant characterizing the safety constraints, and T is the total number of action played. We further present numerical simulations that corroborate our theoretical findings.

1. Introduction

Reinforcement Learning (RL) is the study of an agent trying to maximize its expected cumulative reward by interacting with an unknown environment over time (Sutton and Barto, 2018). In most classical RL algorithms, agents aim to maximize a long term gain by exploring all possible actions. However, freely exploring all actions may be harmful in many real-world systems where playing even one unsafe action may lead to catastrophic results. Thus, safety in RL has become a serious issue that restricts the applicability of RL algorithms to many real-world systems. For example, in a self-driving car, it is critical to explore those policies that avoid crash and damage to the car, people and property. Switching cost limitations in medical applications (Bai et al., 2019) and legal restrictions in financial managements (Abe et al., 2010) are other examples of safety-critical applications. All the aforementioned safety-critical environments introduce the new challenge of balancing the goal of reward maximization with the restriction of playing safe actions.

To address this major concern, the learning algorithm needs to guarantee that it does not violate certain safety constraints. From a bandit optimization point of view, (Amani et al., 2019; Pacchiano et al., 2020; Amani and Thrampoulidis, 2021; Moradipari et al., 2019) study a linear bandit problem, in which, at each round, a linear cost constraint needs to be satisfied with high probability. For this problem, they propose no-regret algorithms that with high probability never violate the constraints. There has been a surge of research activity to address the issue of safe exploration in RL when the environment is modeled via the more challenging and complex setting of an unknown MDP. Many of existing algorithms model the safety in RL via Constrained Markov Decision Process (CMDP), that extends the classical MDP to settings with extra constraints on the total expected cost over a horizon. To address the safety requirements in CMDPs, different approaches such as Primal-Dual Policy Optimization (Paternain et al., 2019b;a; Stooke et al., 2020), Constrained Policy Optimization (Achiam et al., 2017; Yang et al., 2020), and Reward Constrained Policy Optimization (Tessler et al., 2018) have been proposed. These algorithms come with either no theoretical guarantees or asymptotic convergence guarantee in the batch offline setting. In another line of work studying CMDP in online settings, (Efroni et al., 2020; Turchetta et al., 2020; Garcelon et al., 2020; Zheng and Ratliff, 2020; Ding et al., 2020a; Qiu et al., 2020; Ding et al., 2020b; Xu et al., 2020; Kalagarla et al., 2020) propose algorithms coming with sub-linear bounds on the number of constraint violation. Additionally, the safety constraint considered in the aforementioned papers is defined by the cumulative expected cost over a horizon falling below a certain threshold.

In this paper, we propose an upper confidence bound (UCB)-

[1]Department of Electrical and Computer Engineering, University of California, Los Angeles. [2]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver. Correspondence to: Sanae Amani <samani@ucla.edu>, Christos Thrampoulidis <cthrampo@ece.ubc.ca>, Lin F. Yang <linyang@ee.ucla.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

based algorithm – termed Safe Linear UCB Q/V Iteration (SLUCB-QVI) – with the focus on deterministic policy selection respecting a more restrictive notion of safety requirements that must be satisfied at each time-step an action is played with high probability. We also present Randomized SLUCB-QVI (RSLUCB-QVI), a safe algorithm focusing on randomized policy selection without any constraint violation. For both algorithms, we assume the underlying MDP has linear structure and prove a regret bound that is order-wise comparable to those of its unsafe counter-parts.

Our main technical contributions allowing us to guarantee sub-linear regret bound while the safety constraints are never violated, include: 1) conservatively selecting actions from properly defined subsets of the unknown safe sets; and 2) exploiting careful algorithmic designs to ensure *optimism in the face of safety constraints*, i.e., the value function of our proposed algorithms are greater than the optimal value functions. See Sections 2,3, and 4 for details.

Notation. We start by introducing a set of notations that are used throughout the paper. We use lower-case letters for scalars, lower-case bold letters for vectors, and upper-case bold letters for matrices. The Euclidean-norm of $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_2$. We denote the transpose of any column vector $\mathbf{x}$ by $\mathbf{x}^\top$. For any vectors $\mathbf{x}$ and $\mathbf{y}$, we use $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote their inner product. Let $\mathbf{A}$ be a positive definite $d \times d$ matrix and $\boldsymbol{\nu} \in \mathbb{R}^d$. The weighted 2-norm of $\boldsymbol{\nu}$ with respect to $\mathbf{A}$ is defined by $\|\boldsymbol{\nu}\|_{\mathbf{A}} = \sqrt{\boldsymbol{\nu}^\top \mathbf{A} \boldsymbol{\nu}}$. For positive integer n, $[n]$ denotes the $\{1, 2, \ldots, n\}$. We use $\mathbf{e}_i$ to denote the i-th standard basis vector. Finally, we use standard $\tilde{\mathcal{O}}$ notation for big-O notation that ignores logarithmic factors.

1.1. Problem formulation

Finite-horizon Markov decision process. We consider a finite-horizon Markov decision process (MDP) denoted by $M = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, c)$, where $\mathcal{S}$ is the state set, $\mathcal{A}$ is the action set, H is the length of each episode (horizon), $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ are the transition probabilities, $r = \{r_h\}_{h=1}^H$ are the reward functions, and $c = \{c_h\}_{h=1}^H$ are the safety measures. For each time-step $h \in [H]$, $\mathbb{P}_h(s'|s,a)$ denotes the probability of transitioning to state s' upon playing action a at state s, and $r_h : \mathcal{S} \times \mathcal{A} \to [0,1]$ and $c_h : \mathcal{S} \times \mathcal{A} \to [0,1]$ are reward and constraint functions. We consider the learning problem where $\mathcal{S}$ and $\mathcal{A}$ are known, while the transition probabilities $\mathbb{P}_h$, rewards r_h and safety measures c_h are *unknown* to the agent and must be learned online. The agent interacts with its unknown environment described by M in episodes. In particular, at each episode k and time-step $h \in [H]$, the agent observes the state s_h^k, plays an action $a_h^k \in \mathcal{A}$, and observes a reward $r_h^k := r_h(s_h^k, a_h^k)$ and a noise-perturbed safety measure $z_h^k := c_h(s_h^k, a_h^k) + \epsilon_h^k$, where ϵ_h^k is a random additive noise.

Safety Constraint. We assume that the underlying system is safety-critical and the learning environment is subject to a side constraint that restricts the choice of actions. At each episode k and time-step $h \in [H]$, when being in state s_h^k, the agent must select a *safe* action a_h^k such that

$$c_h(s_h^k, a_h^k) \leq \tau \tag{1}$$

with high probability, where τ is a known constant. We accordingly define the *unknown* safe action sets as

$$\mathcal{A}_h^{\text{safe}}(s) := \{a \in \mathcal{A} : c_h(s,a) \leq \tau\}, \quad \forall (s,h) \in \mathcal{S} \times [H].$$

Thus, after observing state s_h^k at episode k and time-step $h \in [H]$, the agent's choice of action must belong to $\mathcal{A}_h^{\text{safe}}(s_h^k)$ with high probability. As a motivating example, consider a self-driving car. On the one hand, the agent (car) is rewarded for getting from point one to point two as fast as possible. On the other hand, the driving behavior must be constrained to respect traffic safety standards.

Goal. A *safe* deterministic policy is a function $\pi : \mathcal{S} \times [H] \to \mathcal{A}$, such that $\pi(s,h) \in \mathcal{A}_h^{\text{safe}}(s)$ is the *safe* action the policy π suggests the agent to play at time-step $h \in [H]$ and state $s \in \mathcal{S}$. Thus, we define the set of safe policies by

$$\Pi^{\text{safe}} := \left\{ \pi : \pi(s,h) \in \mathcal{A}_h^{\text{safe}}(s),\ \forall (s,h) \in \mathcal{S} \times [H] \right\}.$$

For each $h \in [H]$, the cumulative expected reward obtained under a safe policy $\pi \in \Pi^{\text{safe}}$ during and after time-step h, known as the value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$, is defined by

$$V_h^\pi(s) := \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}\left(s_{h'}, \pi(s_{h'}, h')\right) \middle| s_h = s \right], \tag{2}$$

where the expectation is over the environment. We also define the state-action value action $Q_h^\pi : \mathcal{S} \times \mathcal{A}_h^{\text{safe}}(.) \to \mathbb{R}$ for a safe policy $\pi \in \Pi^{\text{safe}}$ at time-step $h \in [H]$ by

$$Q_h^\pi(s,a) := \mathbb{E}\left[\sum_{h'=h+1}^{H} r_{h'}\left(s_{h'}, \pi(s_{h'}, h')\right) \middle| s_h = s, a_h = a \right]. \tag{3}$$

To simplify the notation, for any function f, we denote $[\mathbb{P}_h f](s,a) := \mathbb{E}_{s' \sim \mathbb{P}_h(.|s,a)} f(s')$. Let π_* be the optimal *safe* policy such that $V_h^{\pi*}(s) := V_h^*(s) = \sup_{\pi \in \Pi^{\text{safe}}} V_h^\pi(s)$ for all $(s,h) \in \mathcal{S} \times [H]$. Thus, for all $(s,h) \in \mathcal{S} \times [H]$ and $a \in \mathcal{A}_h^{\text{safe}}(s)$, the Bellman equations for an arbitrary safe policy $\pi \in \Pi^{\text{safe}}$ and the optimal safe policy are:

$$\begin{aligned} Q_h^\pi(s,a) &= r_h(s,a) + [\mathbb{P}_h V_{h+1}^\pi](s,a), \\ V_h^\pi(s) &= Q_h^\pi(s, \pi(s,h)), \end{aligned} \tag{4}$$

$$\begin{aligned} Q_h^*(s,a) &= r_h(s,a) + [\mathbb{P}_h V_{h+1}^*](s,a), \\ V_h^*(s) &= \max_{a \in \mathcal{A}_h^{\text{safe}}(s)} Q_h^*(s,a), \end{aligned} \tag{5}$$

where $V^{\pi}_{H+1}(s) = V^{*}_{H+1}(s) = 0$. Note that in classical RL without safety constraints, the Bellman optimality equation implies that there exists at least one optimal policy that is deterministic (see (Bertsekas et al., 2000; Szepesvári, 2010; Sutton and Barto, 2018)). When considering solving the Bellman equation for the optimal policy, the presence of safety constraints is equivalent to solving it for an MDP without constraints but with different action sets for each $(s,h) \in \mathcal{S} \times [H]$, i.e., $\mathcal{A}^{\text{safe}}_h(s)$.

Let K be the total number of episodes, s^k_1 be the initial state at the beginning of episode $k \in [K]$ and π_k be the high probability *safe* policy chosen by the agent during episode $k \in [K]$. Then the *cumulative pseudo-regret* is defined by

$$R_K := \sum_{k=1}^{K} V^{*}_1(s^k_1) - V^{\pi_k}_1(s^k_1). \tag{6}$$

The agent's goal is to keep R_K as small as possible ($R_K/K \to 0$ as K grows large) *without violating the safety constraint in the process*, i.e., $\pi_k \in \Pi^{\text{safe}}$ for all $k \in [K]$ with high probability.

Linear Function Approximation. We focus on MDPs with linear transition kernels, reward, and cost functions that are encapsulated in the following assumption.

Assumption 1 (Linear MDP (Bradtke and Barto, 1996; Yang and Wang, 2019; Jin et al., 2020))**.** *$M = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, c)$ is a linear MDP with feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist d unknown measures $\boldsymbol{\mu}^*_h := [\mu^{*(1)}_h, \ldots, \mu^{*(d)}_h]^\top$ over $\mathcal{S}$, and unknown vectors $\boldsymbol{\theta}^*_h, \boldsymbol{\gamma}^*_h \in \mathbb{R}^d$ such that $\mathbb{P}_h(.|s,a) = \left\langle \boldsymbol{\mu}^*_h(.), \boldsymbol{\phi}(s,a) \right\rangle$, $r_h(s,a) = \left\langle \boldsymbol{\theta}^*_h, \boldsymbol{\phi}(s,a) \right\rangle$, and $c_h(s,a) = \left\langle \boldsymbol{\gamma}^*_h, \boldsymbol{\phi}(s,a) \right\rangle$.*

This assumption highlights the definition of linear MDP, in which the Markov transition model, the reward functions, and the cost functions are linear in a feature mapping $\boldsymbol{\phi}$.

1.2. Related works

Safe RL with randomized policies: The problem of Safe RL formulated with Constrained Markov Decision Process (CMDP) with a focus on unknown dynamics and *randomized* policies is studied in (Efroni et al., 2020; Turchetta et al., 2020; Garcelon et al., 2020; Zheng and Ratliff, 2020; Ding et al., 2020a; Qiu et al., 2020; Ding et al., 2020b; Xu et al., 2020; Kalagarla et al., 2020). In the above-mentioned papers, the goal is to find the optimal randomized policy that maximizes the reward value function $V^{\pi}_r(s)$ (expected total reward) while ensuring the cost value function $V^{\pi}_c(s)$ (expected total cost) does not exceed a certain threshold. This safety requirement is defined over a *horizon*, in expectation with respect to the environment and the randomization of the policy, and consequently is less strict than the safety requirement considered in this paper, which must be satisfied at each time-step an action is played. In addition to their different problem formulations, the theoretical guarantees of these works fundamentally differ from the ones provided in our paper. The recent closely-related work of (Ding et al., 2020a) studies constrained finite-horizon MDPs with a linear structure as considered in our paper via a primal-dual-type policy optimization algorithm that achieves a $\mathcal{O}(dH^{2.5}\sqrt{T})$ regret and constraint violation and can only be applied to settings with finite action set $\mathcal{A}$. The algorithm of (Efroni et al., 2020) obtains a $\mathcal{O}(|\mathcal{S}|H^2\sqrt{|\mathcal{S}||\mathcal{A}|T})$ regret and constraint violation in the episodic finite-horizon tabular setting via linear program and primal-dual policy optimization. In (Qiu et al., 2020), the authors study an adversarial stochastic shortest path problem under constraints with $\mathcal{O}(|\mathcal{S}|H\sqrt{|\mathcal{A}|T})$ regret and constraint violation. (Ding et al., 2020b) proposes a primal-dual algorithm for solving discounted infinite horizon CMDPs that achieves a global convergence with rate $\mathcal{O}(1/\sqrt{T})$ regarding both the optimality gap and the constraint violation. In contrast to the aforementioned works which can only guarantee bounds on the number of constraint violation, our algorithms *never* violate the safety constraint during the learning process.

Besides primal-dual methods, in (Chow et al., 2018) Lyapunov functions are leveraged to handle the constraints. (Yu et al., 2019) proposes a constrained policy gradient algorithm with convergence guarantee. Both above-stated works focus on solving CMDPs with known transition model and constraint function without providing regret guarantees.

Safe RL with GPs and deterministic transition model and policies: In another line of work, (Turchetta et al., 2016; Berkenkamp et al., 2017; Wachi et al., 2018; Wachi and Sui, 2020) use Gaussian processes to model the dynamics with deterministic transitions and/or the value function in order to be able to estimate the constraints and guarantee safe learning. Despite the fact that some of these algorithms are approximately safe, analysing the convergence is challenging and the regret analysis is lacking.

2. Safe Linear UCB Q/V Iteration

In this section, we present *Safe Linear Upper Confidence Bound Q/V Iteration* (SLUCB-QVI) summarized in Algorithm 1, which is followed by a high-level description of its performance in Section 2. First, we introduce the following necessary assumption and set of notations used in describing Algorithm 1 and its analysis in the next sections.

Assumption 2 (Non-empty safe sets)**.** *For all $s \in \mathcal{S}$, there exists a known safe action $a_0(s)$ such that $a_0(s) \in \mathcal{A}^{\text{safe}}_h(s)$ with known safety measure $\tau_h(s) := \left\langle \boldsymbol{\phi}\left(s, a_0(s)\right), \boldsymbol{\gamma}^*_h \right\rangle < \tau$ for all $h \in [H]$.*

Knowing safe actions $a_0(s)$ is necessary for solving the safe

linear MDP setting studied in this paper, which requires the constraint (1) to be satisfied from the very first round. This assumption is also realistic in many practical examples, where the known safe action could be the one suggested by the current strategy of the company or a very cost-neutral action that does not necessarily have high reward but its cost is far from the threshold. It is possible to relax the assumption of knowing the cost of the safe actions $\tau_h(s)$. In this case, the agent starts by playing $a_0(s)$ for $T_h(s)$ rounds at time-steps h in order to construct a conservative estimator for the gap $\tau - \tau_h(s)$. $T_h(s)$ is selected in an adaptive way and in Appendix A.4, we show that $\frac{16\log(K)}{(\tau-\tau_h(s))^2} \leq T_h(s) \leq \frac{64\log(K)}{(\tau-\tau_h(s))^2}$. After $T_h(s)$ rounds, the agent relies on these estimates of $\tau_h(s)$ in the computation of estimated safe set of policies (discussed shortly).

Notations. For any vector $\mathbf{x} \in \mathbb{R}^d$, define the normalized vector $\tilde{\mathbf{x}} := \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. We define the span of the safe feature $\boldsymbol{\phi}\left(s, a_0(s)\right)$ as $\mathcal{V}_s = \text{span}\left(\boldsymbol{\phi}\left(s, a_0(s)\right)\right) := \left\{\alpha\boldsymbol{\phi}\left(s, a_0(s)\right) : \alpha \in \mathbb{R}\right\}$ and the orthogonal complement of $\mathcal{V}_s$ as $\mathcal{V}_s^{\perp} := \{\mathbf{y} \in \mathbb{R}^d : \langle \mathbf{y}, \mathbf{x}\rangle = 0, \forall \mathbf{x} \in \mathcal{V}_s\}$. For any $\mathbf{x} \in \mathbb{R}^d$, denote by $\Phi_0(s, \mathbf{x}) := \left\langle \mathbf{x}, \tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)\right\rangle \tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)$ its projection on $\mathcal{V}_s$, and, by $\Phi_0^{\perp}(s, \mathbf{x}) := \mathbf{x} - \Phi_0(s, \mathbf{x})$ its projection onto the orthogonal subspace $\mathcal{V}_s^{\perp}$. Moreover, for ease of notation, let $\boldsymbol{\phi}_h^k := \boldsymbol{\phi}(s_h^k, a_h^k)$.

Algorithm 1 SLUCB-QVI

1: **Input:** $\mathcal{A}$, λ, δ, H, K, τ, $\kappa_h(s)$
2: $\mathbf{A}_h^1 = \lambda I$, $\mathbf{A}_{h,s}^1 = \lambda\left(I - \tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)\tilde{\boldsymbol{\phi}}^{\top}\left(s, a_0(s)\right)\right)$ $\mathbf{b}_h^1 = \mathbf{r}_{h,s}^1 = \mathbf{0}$, $\forall (s,h) \in \mathcal{S} \times [H], Q_{H+1}^k(.,.) = 0$, $\forall k \in [K]$
3: **for** episodes $k = 1$ **to** K **do**
4: Observe the initial state s_1^k.
5: **for** time-steps $h = H$ **to** 1 **do**
6: Compute $\mathcal{A}_h^k(s)$ as in (9) $\forall s \in \mathcal{S}$.
7: Compute $Q_h^k(s,a)$ as in (10) $\forall (s,a) \in \mathcal{S} \times \mathcal{A}_h^k(.)$.
8: **end for**
9: **for** time-steps $h = 1$ **to** H **do**
10: Play $a_h^k = \arg\max_{a \in \mathcal{A}_h^k(s_h^k)} Q_h^k(s_h^k, a)$ and observe s_{h+1}^k, r_h^k and z_h^k.
11: **end for**
12: **end for**

2.1. Overview

From a high-level point of view, our algorithm is the safe version of LSVI-UCB proposed by (Jin et al., 2020). In particular, each episode consists of two loops over all time-steps. The first loop (Lines 5-8) updates the quantities $\mathcal{A}_h^k$, estimated safe sets, and Q_h^k, action-value function, that are used to execute the *upper confidence bound* policy $a_h^k = \arg\max_{a \in \mathcal{A}_h^k(s_h^k)} Q_h^k(s_h^k, a)$ in the second loop (Lines 9-11). The key difference between SLUCB-QVI and LSVI-UCB is the requirement that chosen actions a_h^k must always belong to unknown safe sets $\mathcal{A}_h^{\text{safe}}(s_h^k)$. To this end, at each episode $k \in [K]$, in an extra step in the first loop (Line 6), the agent computes a set $\mathcal{A}_h^k(s)$ for all $s \in \mathcal{S}$, which we will show is guaranteed to be a subset of the unknown safe set $\mathcal{A}_h^{\text{safe}}(s)$, and therefore, is a good candidate to select action a_h^k from in the second loop (Line 10). Construction of $\mathcal{A}_h^k(s)$ depends on an appropriate confidence set around the unknown parameter $\boldsymbol{\gamma}_h^*$ used in the definition of safety constraints (see Assumption 1). Since the agent has knowledge of $\tau_h(s) := \left\langle \boldsymbol{\phi}\left(s, a_0(s)\right), \boldsymbol{\gamma}_h^*\right\rangle$ (see Assumption 2), it can compute $z_{h,s}^k := \left\langle \Phi_0^{\perp}\left(s, \boldsymbol{\phi}_h^k\right), \Phi_0^{\perp}\left(s, \boldsymbol{\gamma}_h^*\right)\right\rangle + \epsilon_h^k = z_h^k - \frac{\left\langle \boldsymbol{\phi}_h^k, \tilde{\boldsymbol{\phi}}(s, a_0(s))\right\rangle}{\left\|\boldsymbol{\phi}(s, a_0(s))\right\|_2}\tau_h(s)$, i.e., the cost incurred by a_h^k along the subspace $\mathcal{V}_s^{\perp}$, which is orthogonal to $\boldsymbol{\phi}\left(s, a_0(s)\right)$. Thus, the agent does not need to build confidence sets around $\boldsymbol{\gamma}_h^*$ along the normalized safe feature vector, $\tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)$. Instead, it only builds the following confidence sets around $\Phi_0^{\perp}\left(s, \boldsymbol{\gamma}_h^*\right)$ which is along the orthogonal direction of $\tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)$:

$$\mathcal{C}_h^k(s) := \left\{\boldsymbol{\nu} \in \mathbb{R}^d : \left\|\boldsymbol{\nu} - \boldsymbol{\gamma}_{h,s}^k\right\|_{\mathbf{A}_{h,s}^k} \leq \beta\right\}, \quad (7)$$

where $\boldsymbol{\gamma}_{h,s}^k := \left(\mathbf{A}_{h,s}^k\right)^{-1}\mathbf{r}_{h,s}^k$ is the regularized least-squares estimator of $\Phi_0^{\perp}\left(s, \boldsymbol{\gamma}_h^*\right)$ computed by the inverse of Gram matrix $\mathbf{A}_{h,s}^k := \lambda\left(I - \tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)\tilde{\boldsymbol{\phi}}^{\top}\left(s, a_0(s)\right)\right) + \sum_{j=1}^{k-1}\Phi_0^{\perp}\left(s, \boldsymbol{\phi}_h^j\right)\Phi_0^{\perp,\top}\left(s, \boldsymbol{\phi}_h^j\right)$ and $\mathbf{r}_{h,s}^k := \sum_{j=1}^{k-1} z_{h,s}^j \Phi_0^{\perp}\left(s, \boldsymbol{\phi}_h^j\right)$. The exploration factor β will be defined shortly in Theorem 1 such that it guarantees that the event

$$\mathcal{E}_1 := \left\{\Phi_0^{\perp}\left(s, \boldsymbol{\gamma}_h^*\right) \in \mathcal{C}_h^k(s),\ \forall (s,h,k) \in \mathcal{S} \times [H] \times [K]\right\} \quad (8)$$

i.e., $\Phi_0^{\perp}\left(s, \boldsymbol{\gamma}_h^*\right)$ belongs to the confidence sets $\mathcal{C}_h^k(s)$, holds with high probability. In the implementations, we treat β as a tuning parameter. Conditioned on event $\mathcal{E}_1$, the agent is ready to compute the following inner approximations of the true unknown safe sets $\mathcal{A}_h^{\text{safe}}$ for all $s \in \mathcal{S}$:

$$\mathcal{A}_h^k(s) = \Bigg\{ a \in \mathcal{A} : \frac{\left\langle \Phi_0\left(s, \boldsymbol{\phi}(s,a)\right), \tilde{\boldsymbol{\phi}}\left(s, a_0\left(s\right)\right)\right\rangle}{\left\|\boldsymbol{\phi}\left(s, a_0\left(s\right)\right)\right\|_2}\tau_h(s) + \left\langle \boldsymbol{\gamma}_{h,s}^k, \Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right)\right\rangle + \beta\left\|\Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right)\right\|_{\left(\mathbf{A}_{h,s}^k\right)^{-1}} \leq \tau \Bigg\} \tag{9}$$

Note that $\frac{\left\langle \Phi_0\left(s, \boldsymbol{\phi}(s,a)\right), \tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)\right\rangle}{\left\|\boldsymbol{\phi}\left(s, a_0(s)\right)\right\|_2}\tau_h(s)$ is the known cost of action a at state s along direction $\tilde{\boldsymbol{\phi}}\left(s, a_0\left(s\right)\right)$ and $\max_{\boldsymbol{\nu} \in \mathcal{C}_h^k(s)} \left\langle \Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right), \boldsymbol{\nu}\right\rangle = \left\langle \boldsymbol{\gamma}_{h,s}^k, \Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right)\right\rangle + \beta\left\|\Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right)\right\|_{\left(\mathbf{A}_{h,s}^k\right)^{-1}}$ is its maximum possible cost in the orthogonal space $\mathcal{V}_s^{\perp}$. Thus, $\frac{\left\langle \Phi_0\left(s, \boldsymbol{\phi}(s,a)\right), \tilde{\boldsymbol{\phi}}\left(s, a_0(s)\right)\right\rangle}{\left\|\boldsymbol{\phi}\left(s, a_0(s)\right)\right\|_2}\tau_h(s) + \left\langle \boldsymbol{\gamma}_{h,s}^k, \Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right)\right\rangle + \beta\left\|\Phi_0^{\perp}\left(s, \boldsymbol{\phi}(s,a)\right)\right\|_{\left(\mathbf{A}_{h,s}^k\right)^{-1}}$ is a high probability upper bound on the true unknown cost $\langle \boldsymbol{\phi}(s,a), \boldsymbol{\gamma}_h^* \rangle$, which implies that $\mathcal{A}_h^k(s) \subset \mathcal{A}_h^{\text{safe}}(s)$.

Proposition 1. Conditioned on $\mathcal{E}_1$ in (8), for all $(s,h,k) \in \mathcal{S} \times [H] \times [K]$, it holds that $\left\langle \boldsymbol{\phi}(s,a), \boldsymbol{\gamma}_h^* \right\rangle \leq \tau,\ \forall a \in \mathcal{A}_h^k(s)$.

Thus, conditioned on $\mathcal{E}_1$, the decision rule $a_h^k := \arg\max_{a \in \mathcal{A}_h^k(s_h^k)} Q_h^k(s_h^k, a)$ in Line 10 of Algorithm 1 suggests that a_h^k does not violate the safety constraint. Note that $\mathcal{A}_h^k(s)$ is always non-empty, since as a consequence of Assumption 2, the safe action $a_0(s)$ is always in $\mathcal{A}_h^k(s)$.

Now that the estimated safe sets $\mathcal{A}_h^k(s)$ are constructed, we describe how the action-value functions Q_h^k are computed to be used in the UCB decision rule, selecting the action a_h^k in the second loop of the algorithm. The linear structure of the MDP allows us to parametrize $Q_h^*(s,a)$ by a linear form $\langle \mathbf{w}_h^*, \boldsymbol{\phi}(s,a) \rangle$, where $\mathbf{w}_h^* := \boldsymbol{\theta}_h^* + \int_{\mathcal{S}} V_{h+1}^*(s') d\boldsymbol{\mu}(s')$. Thus, a natural idea to estimate $Q_h^*(s,a)$ is to solve least-squares problem for $\mathbf{w}_h^*$. In fact, for all $(s,a) \in \mathcal{S} \times \mathcal{A}_h^k(.)$, the agent computes $Q_h^k(s,a)$ defined as

$$Q_h^k(s,a) = \min\Bigg\{ \left\langle \mathbf{w}_h^k, \boldsymbol{\phi}(s,a)\right\rangle + \kappa_h(s)\beta\left\|\boldsymbol{\phi}(s,a)\right\|_{\left(\mathbf{A}_h^k\right)^{-1}}, H \Bigg\}, \tag{10}$$

where $\mathbf{w}_h^k := \left(\mathbf{A}_h^k\right)^{-1}\mathbf{b}_h^k$ is the regularized least-squares estimator of $\mathbf{w}_h^*$ computed by the inverse of Gram matrix $\mathbf{A}_h^k := \lambda I + \sum_{j=1}^{k-1} \boldsymbol{\phi}_h^j \boldsymbol{\phi}_h^{j\top}$ and $\mathbf{b}_h^k := \sum_{j=1}^{k-1} \boldsymbol{\phi}_h^j \left[r_h^j + \max_{a \in \mathcal{A}_{h+1}^k(s_{h+1}^j)} Q_{h+1}^k(s_{h+1}^j, a)\right]$. Here, $\kappa_h(s)\beta\left\|\boldsymbol{\phi}(s,a)\right\|_{\left(\mathbf{A}_h^k\right)^{-1}}$ is an exploration bonus that is characterized by: 1) β that encourages enough exploration regarding the uncertainty about r and $\mathbb{P}$; and 2) $\kappa_h(s) > 1$ that encourages enough exploration regarding the uncertainty about c. While we make use of standard analysis of unsafe bandits and MDPs (Abbasi-Yadkori et al., 2011) and (Jin et al., 2020) to define β, appropriately quantifying $\kappa_h(s)$ is the main challenge the presence of safety constraints brings to the analysis of SLUCB-QVI compared to the unsafe LSVI-UCB and it is stated in Lemma 1.

3. Theoretical guarantees of SLUCB-QVI

In this section, we discuss the technical challenges the presence of safety constraints brings to our analysis and provide a regret bound for SLUCB-QVI. Before these, we make the remaining necessary assumptions under which our proposed algorithm operates and achieves good regret bound.

Assumption 3 (Subgaussian Noise)**.** *For all $(h,k) \in [H] \times [K]$, ϵ_h^k is a zero-mean σ-subGaussian random variable.*

Assumption 4 (Boundedness)**.** *Without loss of generality, $\left\|\boldsymbol{\phi}(s,a)\right\|_2 \leq 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, and $\max\left(\left\|\boldsymbol{\mu}_h^*(\mathcal{S})\right\|_2, \left\|\boldsymbol{\theta}_h^*\right\|_2, \left\|\boldsymbol{\gamma}_h^*\right\|_2\right) \leq \sqrt{d}$ for all $h \in [H]$.*

Assumption 5 (Star convex sets)**.** *For all $s \in \mathcal{S}$, the set $\mathcal{D}(s) := \left\{\boldsymbol{\phi}(s,a) : a \in \mathcal{A}\right\}$ is a star convex set around the safe feature $\boldsymbol{\phi}\left(s, a_0\left(s\right)\right)$, i.e., for all $\mathbf{x} \in \mathcal{D}(s)$ and $\alpha \in [0,1]$, $\alpha\mathbf{x} + (1-\alpha)\boldsymbol{\phi}\left(s, a_0\left(s\right)\right) \in \mathcal{D}(s)$.*

Assumptions 3 and 4 are standard in linear MDP and bandit literature (Jin et al., 2020; Pacchiano et al., 2020; Amani et al., 2019). Assumption 5 is necessary to ensure that the agent has the opportunity to explore the feature space around the given safe feature vector $\boldsymbol{\phi}\left(s, a_0\left(s\right)\right)$. For example, consider a simple setting where $\mathcal{S} = \{s_1\}, \mathcal{A} = \{a_1, a_2\}, H = 1, \boldsymbol{\mu}^*(s_1) = (1,1), \boldsymbol{\theta}^* = (0,1), \boldsymbol{\gamma}^* = (0,1), \tau = 2$, $a_0(s_1) = a_2$, and $\mathcal{D}(s_1) = \{\boldsymbol{\phi}(s_1,a_1), \boldsymbol{\phi}(s_1,a_2)\} = \{(0,1),(1,0)\}$, which is not a star convex set. Here, both actions a_1 and a_2 are safe. The optimal safe policy always plays a_1, which gives the highest reward. However, if $\mathcal{D}(s_1)$ does not contain the whole line connecting $(1,0)$ and $(0,1)$, the agent keeps playing a_2 and will not be able to explore other safe action and identify that the optimal policy would always select a_1. Also, it is worth mentioning that the star convexity of the sets $\mathcal{D}(s)$ is a milder assumption than convexity assumption considered in existing safe algorithms of (Amani et al., 2019; Moradipari et al., 2019).

Given these assumptions, we are now ready to present the formal guarantees of SLUCB-QVI in the following theorem.

Theorem 1 (Regret of SLUCB-QVI)**.** *Under Assumptions 1, 2, 3, 4, and 5, there exists an absolute constant $c_\beta > 0$ such that for any fixed $\delta \in (0, 0.5)$, if we set $\beta :=$*

$$\max\left(\sigma\sqrt{d\log\left(\frac{2+\frac{2T}{\lambda}}{\delta}\right)}+\sqrt{\lambda d}, c_\beta dH\sqrt{\log(\tfrac{dT}{\delta})}\right),$$

and $\kappa_h(s) := \frac{2H}{\tau-\tau_h(s)} + 1$, *then with probability at least* $1-2\delta$, *it holds that* $R_K \leq 2H\sqrt{T\log(\frac{dT}{\delta})} + (1+\kappa)\beta\sqrt{2dHT\log\left(1+\frac{K}{d\lambda}\right)}$, *where* $\kappa := \max_{(s,h)\in\mathcal{S}\times[H]}\kappa_h(s)$

Here, $T = KH$ is the total number of action plays. We observe that the regret bound is of the same order as that of state-of-the-art unsafe algorithms, such as LSVI-UCB (Jin et al., 2020), with only an additional factor κ in its second term. The complete proof is reported in the Appendix A.3. In the following section, we give a sketch of the proof.

3.1. Proof sketch of Theorem 1

First, we state the following theorem borrowed from (Abbasi-Yadkori et al., 2011; Jin et al., 2020).

Theorem 2 (Thm. 2 in (Abbasi-Yadkori et al., 2011) and Lemma B.4 in (Jin et al., 2020))**.** *For any fixed policy* π, *define* $V_h^k(s) := \max_{a\in\mathcal{A}_h^k(s,a)} Q_h^k(s,a)$, *and the event*

$$\mathcal{E}_2 := \Big\{\Big|\langle \mathbf{w}_h^k, \boldsymbol{\phi}(s,a)\rangle - Q_h^\pi(s,a) + [\mathbb{P}_h(V_{h+1}^\pi - V_{h+1}^k)](s,a)\Big| \leq \beta\big\|\boldsymbol{\phi}(s,a)\big\|_{(\mathbf{A}_h^k)^{-1}}, \forall(a,s,h,k)\in\mathcal{A}\times\mathcal{S}\times[H]\times[K]\Big\},$$

and recall the definition of $\mathcal{E}_1$ *in* (8). *Then, under Assumptions 1, 2, 3, 4, and the definition of* β *in Theorem 1, there exists an absolute constant* $c_\beta > 0$, *such that for any fixed* $\delta \in (0, 0.5)$, *with probability at least* $1-\delta$, *the event* $\mathcal{E} := \mathcal{E}_2 \cap \mathcal{E}_1$ *holds.*

As our main technical contribution, in Lemma 1, we prove that when $\kappa_h(s) := \frac{2H}{\tau-\tau_h(s)} + 1$, then *optimism in the face of safety constraint*, i.e., $Q_h^*(s,a) \leq Q_h^k(s,a)$ is guaranteed. Intuitively, this is required because the maximization in Line 10 of Algorithm 1 is not over the entire $\mathcal{A}_h^{\text{safe}}(s_h^k)$, but only a subset of it. Thus, larger values of $\kappa_h(s)$ (compared to $\kappa_h(s) = 1$ in unsafe algorithm LSVI-UCB) are needed to provide enough exploration to the algorithm so that the selected actions in $\mathcal{A}_h^k(s_h^k)$ are -often enough- *optimistic*, i.e., $Q_h^*(s,a) \leq Q_h^k(s,a)$.

Lemma 1 (Optimism in the face of safety constraint in SLUCB-QVI)**.** *Let* $\kappa_h(s) := \frac{2H}{\tau-\tau_h(s)} + 1$ *and Assumptions 1,2,3,4,5 hold. Then, conditioned on* $\mathcal{E}$, *it holds that* $V_h^*(s) \leq V_h^k(s), \forall(s,h,k)\in\mathcal{S}\times[H]\times[K]$.

We report the proof in Appendix A.2. As a direct conclusion of Lemma 1 and on event $\mathcal{E}_2$ defined in Theorem 2, we have

$$\begin{aligned} Q_h^*(s,a) &\leq \Big\langle \mathbf{w}_h^k, \boldsymbol{\phi}(s,a)\Big\rangle \\ &\quad + \beta\big\|\boldsymbol{\phi}(s,a)\big\|_{(\mathbf{A}_h^k)^{-1}} + [\mathbb{P}_h V_{h+1}^* - V_{h+1}^k](s,a) && \text{(Event } \mathcal{E}_2\text{)} \\ &\leq Q_h^k(s,a). && \text{(Lemma 1)} \end{aligned}$$

This is encapsulated in the following corollary.

Corollary 1 (UCB)**.** *Let* $\kappa_h(s) := \frac{2H}{\tau-\tau_h(s)} + 1$ *and Let Assumptions 1,2,3,4,5 hold. Then, conditioned on* $\mathcal{E}$, *it holds that* $Q_h^*(s,a) \leq Q_h^k(s,a), \forall(a,s,h,k)\in\mathcal{A}\times\mathcal{S}\times[H]\times[K]$.

After proving UCB nature of SLUCB-QVI using Lemma 1, we are ready to exploit the standard analysis of classical unsafe LSVI-UCB (Jin et al., 2020) to complete the analysis and establish the final regret bound of SLUCB-QVI.

4. Extension to randomized policy selection

SLUCB-QVI presented in Section 2 can only output a deterministic policy. In this section, we show that our results can be extended to the setting of randomized policy selection, which might be desirable in practice. A randomized policy $\pi : \mathcal{S}\times[H] \to \Delta_{\mathcal{A}}$ maps states and time-steps to distributions over actions such that $a \sim \pi(s,h)$ is the action the policy π suggests the agent to play at time-step $h\in[H]$ when being at state $s\in\mathcal{S}$. At each episode k and time-step $h\in[H]$, when being in state s_h^k, the agent must draw its action a_h^k from a *safe* policy $\pi_k(s_h^k, h)$ such that

$$\mathbb{E}_{a_h^k\sim\pi_k(s_h^k,h)} c_h(s_h^k, a_h^k) \leq \tau \tag{11}$$

with high probability. We accordingly define the *unknown* set of safe policies by

$$\tilde{\Pi}^{\text{safe}} := \Big\{\pi : \pi(s,h)\in\Gamma_h^{\text{safe}}(s),\ \forall(s,h)\in\mathcal{S}\times[H]\Big\},$$

where $\Gamma_h^{\text{safe}}(s) := \big\{\theta\in\Delta_{\mathcal{A}} : \mathbb{E}_{a\sim\theta} c_h(s,a) \leq \tau\big\}$. Thus, after observing state s_h^k at time-step $h\in[H]$ in episode k, the agent's choice of policy must belong to $\Gamma_h^{\text{safe}}(s_h^k)$ with high probability. In this formulation, the expectation in the definition of (action-) value functions for a policy π is over both the environment and the randomness of policy π. We denote them by $\tilde{V}_h^\pi$ and $\tilde{Q}_h^\pi$ to distinguish them from V_h^π and Q_h^π defined in (2) and (3) for a deterministic policy π. Let π_* be the optimal safe policy such that $\tilde{V}_h^{\pi_*}(s) := \tilde{V}_h^*(s) = \sup_{\pi\in\tilde{\Pi}^{\text{safe}}} \tilde{V}_h^\pi(s)$ for all $(s,h)\in\mathcal{S}\times[H]$. Thus, for all $(a,s,h)\in\mathcal{A}\times\mathcal{S}\times[H]$, the Bellman equations for

a safe policy $\pi \in \tilde{\Pi}^{\text{safe}}$ and the optimal safe policy are

$$\tilde{Q}_h^\pi(s,a) = r_h(s,a) + [\mathbb{P}_h \tilde{V}_{h+1}^\pi](s,a),$$
$$\tilde{V}_h^\pi(s) = \mathbb{E}_{a\sim\pi(s,h)}\left[\tilde{Q}_h^\pi(s,a)\right], \quad (12)$$
$$\tilde{Q}_h^*(s,a) = r_h(s,a) + [\mathbb{P}_h \tilde{V}_{h+1}^*](s,a),$$
$$\tilde{V}_h^*(s) = \max_{\theta\in\Gamma_h^{\text{safe}}(s)} \mathbb{E}_{a\in\theta}\left[\tilde{Q}_h^*(s,a)\right], \quad (13)$$

where $\tilde{V}_{H+1}^\pi(s) = \tilde{V}_{H+1}^*(s) = 0$, and the cumulative regret is defined as $R_K := \sum_{k=1}^K \tilde{V}_1^*(s_1^k) - \tilde{V}_1^{\pi_k}(s_1^k)$. This definition of safety constraint in (11) frees us from star-convexity assumption on the sets $\mathcal{D}(s) := \{\boldsymbol{\phi}(s,a) : a \in \mathcal{A}\}$ (Assumption 5), which is necessary for the deterministic policy selection approach. We propose a modification of SLUCB-QVI which is tailored to this new formulation and termed Randomized SLUCB-QVI (RSLUCB-QVI). This new algorithm also achieves a sub-linear regret with the same order as that of SLUCB-QVI, i.e., $\tilde{\mathcal{O}}\left(\kappa\sqrt{d^3H^3T}\right)$.

While RSLUCB-QVI respects a milder definition of the safety constraint (cf. (11)) compared to that considered in SLUCB-QVI (cf. (1)), it still possesses significant superiorities over other existing algorithms solving CMDP with randomized policy selection (Efroni et al., 2020; Turchetta et al., 2020; Garcelon et al., 2020; Zheng and Ratliff, 2020; Ding et al., 2020a; Qiu et al., 2020; Ding et al., 2020b; Xu et al., 2020; Kalagarla et al., 2020). First, the safety constraint considered in these algorithms is defined by the *cumulative* expected cost over a horizon falling below a certain threshold, while RSLUCB-QVI guarantees that the expected cost incurred at each time-step an action is played (not over a horizon) is less than a threshold. Second, even for this looser definition of safety constraint, the best these algorithms can guarantee in terms of constraint satisfaction is a sub-linear bound on the number of constraint violation, whereas RSLUCB-QVI ensures *no constraint violation*.

4.1. Randomized SLUCB-QVI

We now describe RSLUCB-QVI summarized in Algorithm 2. Let $\boldsymbol{\phi}^\theta(s) := \mathbb{E}_{a\sim\theta}\boldsymbol{\phi}(s,a)$. At each episode $k \in [K]$, in the first loop, the agent computes the estimated set of true unknown set $\Gamma_h^{\text{safe}}(s)$ for all $s \in \mathcal{S}$ as follows:

$$\Gamma_h^k(s) := \left\{\theta \in \Delta_{\mathcal{A}} : \mathbb{E}_{a\sim\theta}\left[\frac{\left\langle \Phi_0\left(s,\boldsymbol{\phi}(s,a)\right), \tilde{\boldsymbol{\phi}}\left(s,a_0(s)\right)\right\rangle}{\left\|\boldsymbol{\phi}\left(s,a_0(s)\right)\right\|_2}\tau_h(s)\right]\right.$$
$$\left. + \max_{\boldsymbol{\nu}\in\mathcal{C}_h^k(s)} \left\langle \Phi_0^\perp\left(s, \mathbb{E}_{a\sim\theta}\left[\boldsymbol{\phi}(s,a)\right]\right), \boldsymbol{\nu}\right\rangle \le \tau\right\}$$
$$= \left\{\theta \in \Delta_{\mathcal{A}} : \frac{\left\langle \Phi_0\left(s,\boldsymbol{\phi}^\theta(s)\right), \tilde{\boldsymbol{\phi}}\left(s,a_0(s)\right)\right\rangle}{\left\|\boldsymbol{\phi}\left(s,a_0(s)\right)\right\|_2}\tau_h(s)\right.$$
$$\left. + \left\langle \boldsymbol{\gamma}_{h,s}^k, \Phi_0^\perp\left(s,\boldsymbol{\phi}^\theta(s)\right)\right\rangle + \beta\left\|\Phi_0^\perp\left(s,\boldsymbol{\phi}^\theta(s)\right)\right\|_{\left(\mathbf{A}_{h,s}^k\right)^{-1}} \le \tau\right\}. \quad (14)$$

Note that due to the linear structure of the MDP, we can again parametrize $\tilde{Q}_h^*(s,a)$ by a linear form $\langle\tilde{\mathbf{w}}_h^*, \boldsymbol{\phi}(s,a)\rangle$, where $\tilde{\mathbf{w}}_h^* := \boldsymbol{\theta}_h^* + \int_{\mathcal{S}} \tilde{V}_{h+1}^*(s')d\boldsymbol{\mu}(s')$. In the next step, for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, the agent computes

$$\tilde{Q}_h^k(s,a) = \left\langle \tilde{\mathbf{w}}_h^k, \boldsymbol{\phi}(s,a)\right\rangle + \kappa_h(s)\beta\left\|\boldsymbol{\phi}(s,a)\right\|_{\left(\mathbf{A}_h^k\right)^{-1}}, \quad (15)$$

where $\tilde{\mathbf{w}}_h^k := \left(\mathbf{A}_h^k\right)^{-1}\tilde{\mathbf{b}}_h^k$ is the regularized least-squares estimator of $\tilde{\mathbf{w}}_h^*$ computed by the Gram matrix $\mathbf{A}_h^k$ and $\tilde{\mathbf{b}}_h^k := \sum_{j=1}^{k-1}\boldsymbol{\phi}_h^j[r_h^j + \min\{\max_{\theta\in\Gamma_{h+1}^k(s_{h+1}^j)}\mathbb{E}_{a\sim\theta}[\tilde{Q}_{h+1}^k(s_{h+1}^j,a)], H\}]$. After these computations in the first loop, the agent draws actions a_h^k from distribution $\Gamma_h^k(s_h^k)$ in the second loop. Define $\tilde{V}_h^k(s) := \min\left\{\max_{\theta\in\Gamma_h^k(s)}\mathbb{E}_{a\sim\theta}\left[\tilde{Q}_h^k(s,a)\right], H\right\}$, and

$$\mathcal{E}_3 := \left\{\left|\langle\tilde{\mathbf{w}}_h^k, \boldsymbol{\phi}(s,a)\rangle - \tilde{Q}_h^\pi(s,a) + [\mathbb{P}_h\tilde{V}_{h+1}^\pi - \tilde{V}_{h+1}^k](s,a)\right|\right.$$
$$\left. \le \beta\left\|\boldsymbol{\phi}(s,a)\right\|_{\left(\mathbf{A}_h^k\right)^{-1}}, \forall(a,s,h,k) \in \mathcal{A}\times\mathcal{S}\times[H]\times[K]\right\}.$$

It can be easily shown that the results stated in Theorem 2 hold for the settings focusing on randomized policies, i.e., under Assumptions 1, 2, 3, and 4, and by the definition of β in Theorem 1, with probability at least $1-2\delta$, the event $\tilde{\mathcal{E}} := \mathcal{E}_1 \cap \mathcal{E}_3$ holds. Therefore, as a direct conclusion of Proposition 1, it is guaranteed that conditioned on $\mathcal{E}_1$, all the policies inside $\Gamma_h^k(s)$ are safe, i.e., $\Gamma_h^k(s) \subset \Gamma_h^{\text{safe}}(s)$. Now, in the following lemma, we quantify $\kappa_h(s)$.

Lemma 2 (Optimism in the face of safety constraint in RSLUCB-QVI). *Let $\kappa_h(s) := \frac{2H}{\tau-\tau_h(s)}+1$ and Assumptions 1,2,3,4 hold. Then, conditioned on event $\tilde{\mathcal{E}}$, it holds that $\tilde{V}_h^*(s) \le \tilde{V}_h^k(s), \forall(s,h,k) \in \mathcal{S}\times[H]\times[K]$.*

The proof is included in Appendix B.1. Using Lemma 2, we show that $\tilde{Q}_h^*(s,a) \le \tilde{Q}_h^k(s,a), \forall(a,s,h,k) \in \mathcal{A}\times\mathcal{S}\times[H]\times[K]$. This highlights the UCB nature of RSLUCB-QVI, allowing us to exploit the standard analysis of unsafe LSVI-UCB (Jin et al., 2020) to establish the regret bound.

Theorem 3 (Regret of RSLUCB-QVI). *Under Assumptions 1, 2, 3, and 4, there exists an absolute constant $c_\beta > 0$ such that for any fixed $\delta \in (0, 1/3)$, and the definition of β in Theorem 1, if we set $\kappa_h(s) := \frac{2H}{\tau-\tau_h(s)} + 1$,*

Algorithm 2 RSLUCB-QVI

1: **Input:** $\mathcal{A}$, λ, δ, H, K, τ, $\kappa_h(s)$
2: $\mathbf{A}_h^1 = \lambda I$, $\mathbf{A}_{h,s}^1 = \lambda\left(I - \tilde{\boldsymbol{\phi}}\left(s, a_0\left(s\right)\right)\tilde{\boldsymbol{\phi}}^\top\left(s, a_0\left(s\right)\right)\right)$ $\tilde{\mathbf{b}}_h^1 = \mathbf{r}_{h,s}^1 = \mathbf{0}$, $\forall (s,h) \in \mathcal{S}\times[H], \tilde{Q}_{H+1}^k(.,.) = 0$, $\forall k \in [K]$
3: **for** episodes $k = 1$ **to** K **do**
4: Observe the initial state s_1^k.
5: **for** time-steps $h = H$ **to** 1 **do**
6: Compute $\Gamma_h^k(s)$ as in (14) $\forall s \in \mathcal{S}$.
7: Compute $\tilde{Q}_h^k(s,a)$ as in (15) $\forall (s,a) \in \mathcal{S}\times\mathcal{A}$.
8: **end for**
9: **for** time-steps $h = 1$ **to** H **do**
10: Play $a_h^k \sim \arg\max_{\theta \in \Gamma_h^k(s_h^k)} \mathbb{E}_{a\sim\theta}\left[\tilde{Q}_h^k(s_h^k, a)\right]$ and observe s_{h+1}^k, r_h^k and z_h^k.
11: **end for**
12: **end for**

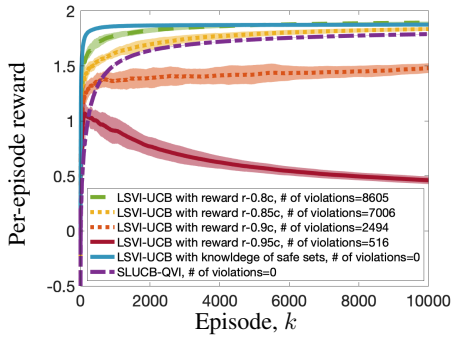


Figure 1. Comparison of SLUCB-QVI to the unsafe state-of-the-art verifying that: 1) when LSVI-UCB (Jin et al., 2020) has knowledge of $\boldsymbol{\gamma}_h^*$, it outperforms SLUCB-QVI (without knowledge of $\boldsymbol{\gamma}_h^*$) as expected; 2) when LSVI-UCB does not know $\boldsymbol{\gamma}_h^*$ (as is the case for SLUCB-QVI) and its goal is to maximize $r - \lambda' c$ instead of r, larger λ' leads to smaller per-episode reward and number of constraint violations while the number of constraint violations for SLUCB-QVI is zero.

then with probability at least $1-3\delta$*, it holds that* $R_K \leq 2H\sqrt{T\log(\frac{dT}{\delta})} + 2(1+\kappa)\beta\sqrt{2dHT\log\left(1+\frac{K}{d\lambda}\right)}$*, where* $\kappa := \max_{(s,h)\in\mathcal{S}\times[H]} \kappa_h(s)$.

See Appendix B.2 for the proof.

5. Experiments

In this section, we present numerical simulations to complement and confirm our theoretical findings. We evaluate the performance of SLUCB-QVI on synthetic environments and implement RSLUCB-QVI on the *Frozen Lake* environment from OpenAI Gym (Brockman et al., 2016).

5.1. SLUCB-QVI on synthetic environments

The results shown in Figure 1 depict averages over 20 realizations, for which we have chosen $\delta = 0.01$, $\sigma = 0.01$, $\lambda = 1$, $d = 5$, $\tau = 0.5$, $H = 3$ and $K = 10000$. The parameters $\{\boldsymbol{\theta}_h^*\}_{h\in[H]}$ and $\{\boldsymbol{\gamma}_h^*\}_{h\in[H]}$ are drawn from $\mathcal{N}(0, I_d)$. In order to tune parameters $\{\boldsymbol{\mu}_h^*(.)\}_{h\in[H]}$ and the feature map $\boldsymbol{\phi}$ such that they are compatible with Assumption 1, we consider that the feature space $\{\boldsymbol{\phi}(s,a) : (s,a) \in \mathcal{S}\times\mathcal{A}\}$ is a subset of the d-dimensional simplex and $\mathbf{e}_i^\top \boldsymbol{\mu}_h^*(.)$ is an arbitrary probability measure over $\mathcal{S}$ for all $i \in [d]$. This guarantees that Assumption 1 holds.

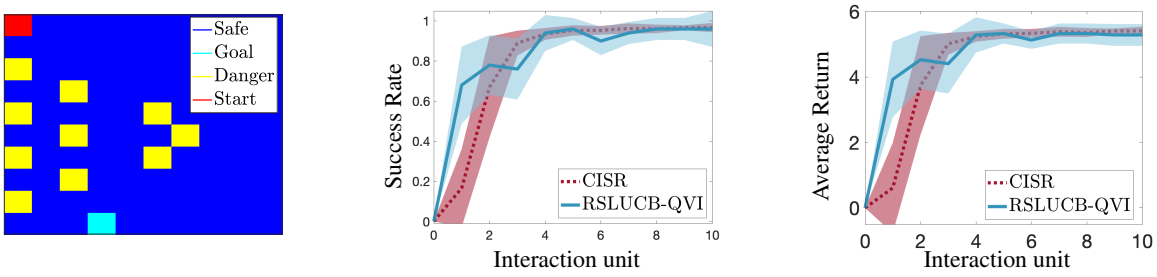
Computing safe sets $\mathcal{A}_h^k(s)$ in the first loop of SLUCB-QVI (Line 6), is followed by selecting an action that maximizes a linear function (in feature map $\boldsymbol{\phi}$) over the feature space $\mathcal{D}_h^k(s_h^k) := \left\{\boldsymbol{\phi}(s_h^k, a) : a \in \mathcal{A}_h^k(s_h^k)\right\}$ in its second loop (Line 10). Unfortunately, even if the feature space $\{\boldsymbol{\phi}(s,a) : (s,a) \in \mathcal{S}\times\mathcal{A}\}$ is convex, the set $\mathcal{D}_h^k(s_h^k)$ can have a form over which maximizing the linear function is intractable. In our experiments, we define map $\boldsymbol{\phi}$ such that the sets $\mathcal{D}(s)$ are star convex and *finite* around $\boldsymbol{\phi}\left(s, a_0\left(s\right)\right)$ with $N = 100$ (see Definition 1) and therefore, we can show that the optimization problem in Line 10 of SLUCB-QVI can be solved efficiently (see Appendix C for a proof).

Definition 1 (Finite star convex set)**.** A star convex set $\mathcal{D}$ around $\mathbf{x}_0 \in \mathbb{R}^d$ is finite, if there exist finitely many vectors $\{\mathbf{x}_i\}_{i=1}^N$ such that $\mathcal{D} = \cup_{i=1}^N [\mathbf{x}_0, \mathbf{x}_i]$, where $[\mathbf{x}_0, \mathbf{x}_i]$ is the line connecting $\mathbf{x}_0$ and $\mathbf{x}_i$.

Figure 1 depicts the average per-episode reward of SLUCB-QVI and compares it to that of baseline and emphasizes the value of SLUCB-QVI in terms of respecting the safety constraints at all time-steps. Specifically, we compare SLUCB-QVI with 1) LSVI-UCB (Jin et al., 2020) when it has knowledge of safety constraints, i.e., $\boldsymbol{\gamma}_h^*$; and 2) LSVI-UCB, when it does not know $\boldsymbol{\gamma}_h^*$ (as is the case for SLUCB-QVI) and its goal is to maximize the function $r - \lambda' c$, with the constraint being pushed into the objective function, for different values of $\lambda' = 0.8, 0.85, 0.9$ and 0.95. Thus, playing costly actions is discouraged via low rewards. The plot verifies that LSVI-UCB with knowledge of $\boldsymbol{\gamma}_h^*$ outperforms SLUCB-QVI without knowledge of $\boldsymbol{\gamma}_h^*$ as expected. Also, larger λ' leads to smaller per-episode reward and number of constraint violations when LSVI-UCB seeks to maximize $r - \lambda' c$ (without knowledge of $\boldsymbol{\gamma}_h^*$) while the number of constraint violations for SLUCB-QVI is zero.

5.2. RSLUCB-QVI on Frozen Lake environment

We evaluate the performance of RSLUCB-QVI in the Frozen Lake environment. The agent seeks to reach a goal in a 10×10 2D map (Figure 2a) while avoiding dangers.



(a) Map (b) Success rate (c) Average return

Figure 2. Comparison of RSLUCB-QVI and CISR (Turchetta et al., 2020) in Frozen Lake environment.

At each time step, the agent can move in four directions, i.e., $\mathcal{A} = \{a_1 : \text{left}, a_2 : \text{right}, a_3 : \text{down}, a_4 : \text{up}\}$. With probability 0.9 it moves in the desired direction and with probability 0.05 it moves in either of the orthogonal directions. We set $H = 1000$, $K = 10$, $d = |\mathcal{S}| = 100$, and $\boldsymbol{\mu}^*(s) \sim \mathcal{N}(0, I_d)$ for all $s \in \mathcal{S} = \{s_1, \ldots, s_{100}\}$. We then properly specified the feature map $\boldsymbol{\phi}(s, a)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ by solving a set of linear equations such that the transition specifics of the environment explained above are respected. In order to interpret the requirement of avoiding dangers as a constraint of form (11), we tuned $\boldsymbol{\gamma}^*$ and τ as follows: the cost of playing action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$ is the probability of the agent moving to one of the danger states. Therefore a safe policy insures that the expected value of probability of moving to a danger state is a small value. To this end, we set $\boldsymbol{\gamma}^* = \sum_{s \in \text{Danger states}} \boldsymbol{\mu}^*(s)$ and $\tau = 0.1$. Also, for each state $s \in \mathcal{S}$ a safe action, playing which leads to one of the danger states with small probability ($\tau = 0.1$) is given to the agent. We solve a set of linear equations to tune $\boldsymbol{\theta}^*$ such that at each state $s \in \mathcal{S}$, the direction which leads to a state that is closest to the goal state gives the agent a reward 1, while playing other three directions gives it a reward 0.01. This model persuades the agent to move towards to the goal.

After specifying the feature map $\boldsymbol{\phi}$ and tuning all parameters, we implemented RSLUCB-QVI for 10 interaction units (episodes) i.e, $K = 10$) each consisting of 1000 time-steps (horizon), i.e., $H = 1000$). During each interaction unit (episode) and after each move, the agent can end up in one of three kinds of states: 1) goal, resulting in a successful termination of the interaction unit; 2) danger, resulting in a failure and the consequent termination of the interaction unit; 3) safe. The agent receives a return of 6 for reaching the goal and 0.01 otherwise.

In Figure 2, we report the average of success rate and return over 20 agents for each of which we implemented RSLUCB-QVI 10 times and compare our results with that of CISR proposed by (Turchetta et al., 2020) in which a teacher helps the agent in selecting safe actions by making interventions. While the performances of both approaches, RSLUCB-QVI and CISR, are fairly comparable, an important point to consider is that each interaction unit (episode) in CISR consists of 10000 time-steps whereas this number is 1000 in RSLUCB-QVI. Notably, the learning rate of RSLUCB-QVI is faster than that of CISR. Also it is noteworthy that we comparedRSLUCB-QVI with CISR when it uses the *optimized* intervention, which gives the best results compared to other types of intervention.

6. Conclusion

In this paper, we developed SLUCB-QVI and RSLUCB-QVI, two safe RL algorithms in the setting of finite-horizon linear MDP. For these algorithms, we provided sub-linear regret bounds $\tilde{\mathcal{O}}\left(\kappa\sqrt{d^3H^3T}\right)$, where H is the duration of each episode, d is the dimension of the feature mapping, κ is a constant characterizing the safety constraints, and $T = KH$ is the total number of action plays. We proved that with high probability, they never violate the unknown safety constraints. Finally, we implemented SLUCB-QVI and RSLUCB-QVI on synthetic and Frozen Lake environments, respectively, which confirms that our algorithms have performances comparable to that of state-of-the-art that either have knowledge of the safety constraint or take advantage of a teacher's advice helping the agent avoid unsafe actions.

Acknowledgements

We thank Mohammad Ghavamzadeh for discussion and bringing finite star convex sets into our attention.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Abe, N., Melville, P., Pendus, C., Reddy, C. K., Jensen, D. L., Thomas, V. P., Bennett, J. J., Anderson, G. F., Cooley, B. R., Kowalczyk, M., et al. (2010). Optimizing debt collections using constrained reinforcement learning. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 75–84.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. (2017). Constrained policy optimization. In *International Conference on Machine Learning*, pages 22–31. PMLR.

Amani, S., Alizadeh, M., and Thrampoulidis, C. (2019). Linear stochastic bandits under safety constraints. In *Advances in Neural Information Processing Systems*, pages 9252–9262.

Amani, S. and Thrampoulidis, C. (2021). Decentralized multi-agent linear bandits with safety constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6627–6635.

Bai, Y., Xie, T., Jiang, N., and Wang, Y.-X. (2019). Provably efficient q-learning with low switching cost. *arXiv preprint arXiv:1905.12849*.

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pages 908–918.

Bertsekas, D. P. et al. (2000). *Dynamic programming and optimal control: Vol. 1*. Athena scientific Belmont.

Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1-3):33–57.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. (2018). A lyapunov-based approach to safe reinforcement learning. In *Advances in neural information processing systems*, pages 8092–8101.

Ding, D., Wei, X., Yang, Z., Wang, Z., and Jovanović, M. R. (2020a). Provably efficient safe exploration via primal-dual policy optimization. *arXiv preprint arXiv:2003.00534*.

Ding, D., Zhang, K., Basar, T., and Jovanovic, M. (2020b). Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems*, 33.

Efroni, Y., Mannor, S., and Pirotta, M. (2020). Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*.

Garcelon, E., Ghavamzadeh, M., Lazaric, A., and Pirotta, M. (2020). Conservative exploration in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1431–1441. PMLR.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143.

Kalagarla, K. C., Jain, R., and Nuzzo, P. (2020). A sample-efficient algorithm for episodic finite-horizon mdp with constraints. *arXiv preprint arXiv:2009.11348*.

Moradipari, A., Amani, S., Alizadeh, M., and Thrampoulidis, C. (2019). Safe linear thompson sampling. *arXiv preprint arXiv:1911.02156*.

Pacchiano, A., Ghavamzadeh, M., Bartlett, P., and Jiang, H. (2020). Stochastic bandits with linear constraints. *arXiv preprint arXiv:2006.10185*.

Paternain, S., Calvo-Fullana, M., Chamon, L. F., and Ribeiro, A. (2019a). Safe policies for reinforcement learning via primal-dual methods. *arXiv preprint arXiv:1911.09101*.

Paternain, S., Chamon, L. F., Calvo-Fullana, M., and Ribeiro, A. (2019b). Constrained reinforcement learning has zero duality gap. *arXiv preprint arXiv:1910.13393*.

Qiu, S., Wei, X., Yang, Z., Ye, J., and Wang, Z. (2020). Upper confidence primal-dual optimization: Stochastically constrained markov decision processes with adversarial losses and unknown transitions. *arXiv preprint arXiv:2003.00660*.

Stooke, A., Achiam, J., and Abbeel, P. (2020). Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pages 9133–9143. PMLR.

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

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103.

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

Turchetta, M., Berkenkamp, F., and Krause, A. (2016). Safe exploration in finite markov decision processes with gaussian processes. *arXiv preprint arXiv:1606.04753*.

Turchetta, M., Kolobov, A., Shah, S., Krause, A., and Agarwal, A. (2020). Safe reinforcement learning via curriculum induction. *arXiv preprint arXiv:2006.12136*.

Wachi, A. and Sui, Y. (2020). Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pages 9797–9806. PMLR.

Wachi, A., Sui, Y., Yue, Y., and Ono, M. (2018). Safe exploration and optimization of constrained mdps using gaussian processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Xu, T., Liang, Y., and Lan, G. (2020). A primal approach to constrained policy optimization: Global optimality and finite-time analysis. *arXiv preprint arXiv:2011.05869*.

Yang, L. and Wang, M. (2019). Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR.

Yang, T.-Y., Rosca, J., Narasimhan, K., and Ramadge, P. J. (2020). Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*.

Yu, M., Yang, Z., Kolar, M., and Wang, Z. (2019). Convergent policy optimization for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3127–3139.

Zheng, L. and Ratliff, L. J. (2020). Constrained upper confidence reinforcement learning. *arXiv preprint arXiv:2001.09377*.