

Probabilistic Shielding for Safe Reinforcement Learning

Edwin Hamel-De le Court, Francesco Belardinelli, Alexander W. Goodall

Imperial College London

{e.hamel-de-le-court, francesco.belardinelli, a.goodall22}@ic.ac.uk

Abstract

In real-life scenarios, a Reinforcement Learning (RL) agent aiming to maximise their reward, must often also behave in a safe manner, including at training time. Thus, much attention in recent years has been given to Safe RL, where an agent aims to learn an optimal policy among all policies that satisfy a given safety constraint. However, strict safety guarantees are often provided through approaches based on linear programming, and thus have limited scaling. In this paper we present a new, scalable method, which enjoys strict formal guarantees for Safe RL, in the case where the safety dynamics of the Markov Decision Process (MDP) are known, and safety is defined as an undiscounted probabilistic avoidance property. Our approach is based on state-augmentation of the MDP, and on the design of a shield that restricts the actions available to the agent. We show that our approach provides a strict formal safety guarantee that the agent stays safe at training and test time. Furthermore, we demonstrate that our approach is viable in practice through experimental evaluation.

Introduction

Reinforcement Learning (RL) aims to optimise the behaviour of an agent in an unknown environment. Much attention has been given to RL in recent years, because of its many practical applications, which include for example playing games (Mnih et al. 2013) or robotics (Kober, Bagnell, and Peters 2013). In many of these applications however, safety, either for the agent or for other humans, is critical. Consequently, Safe RL has been developed, where an agent has to maximise its cumulative expected reward subject to a constraint. Even though much progress has been made to provide safety guarantees during training and at test time, approaches providing strict safety guarantees still rely on Linear Programming (Karmarkar 1984), which is known to lack scalability.

Contributions. We design a new shielding approach for finding a policy that maximizes cumulative reward in a finite MDP with known safety dynamics while guaranteeing safety throughout the whole learning phase. We consider MDPs where some states are labelled unsafe, and the safety we consider consists in avoiding those unsafe states with

at least some probability p . This framework comes from a probabilistic version of what is usually called the “safety” fragment of Linear Temporal Logic (LTL) (Alshiekh et al. 2018) (Jansen et al. 2020). Although, for simplicity’s sake, we consider only a subset of that safety fragment, *i.e.* safety definable by state-avoidance, it is possible to reduce the whole fragment to that subset with the usual trick of making a product between the automaton representing the LTL property and the MDP (Alshiekh et al. 2018) (Jansen et al. 2020). Thus, the model we use can capture many real-life scenarios, like a robot’s task of reaching a goal position while avoiding objects on his path.

Our approach is, to our knowledge, in the framework we consider, the first approach not based on Linear Programming (Karmarkar 1984) that gives *strict formal guarantees* for safety throughout learning and at test time. Instead of using Linear Programming, that has limited scalability (Sutton and Barto 2018), we leverage sound Value Iteration algorithms (see e.g. (Haddad and Monmege 2018; Quatmann and Katoen 2018; Hartmanns and Kaminski 2020)), which are algorithms that improve on Value Iteration in order to give formal approximation guarantees. We use the values obtained by Value Iteration over safety costs to construct a shield that makes the agent’s exploration of the MDP safe by constraining its actions. Constructing a shield is a well-known approach to solve constrained RL (Alshiekh et al. 2018; Jansen et al. 2020; Elsayed-Aly et al. 2021; Yang et al. 2023). However, in contrast to shields defined in previous papers, the shield we construct does not directly constrain the actions of the agent in the original MDP, but constrains the actions of the agent in a safety-aware state-augmented MDP. This approach allows for preserving optimality while ensuring safety in a probabilistic context. Any RL algorithm can then be used to solve the shielded MDP, such as PPO (Schulman et al. 2017), A2C (Mnih et al. 2016), etc. Once the shield is constructed, it is used to train the agent to maximize its cumulative reward, with no constraint violations incurred. To summarize, the contributions of the paper are as follows.

1. We design a shield for a finite MDP as a safety-aware state-augmentation of that MDP using only its safety dynamics.
2. We show that the shield makes the agent’s exploration safe.

3. We show that finding an optimal policy among all safe policies reduces to finding an optimal policy in the shield.
4. We provide a practical way of implementing a shield as a gym environment.

The paper is organized as follows. The preliminaries introduce the mathematical background and notations needed for the paper. Then, we introduce the problem considered, give an overview of our approach, formally define the shield and show that it is safe and optimality-preserving. Finally, we discuss a practical way to implement the shield.

Related Work

Safe Reinforcement Learning has gathered much attention in recent years, and several approaches have been proposed. A comprehensive survey can be found in (Gu et al. 2024). Policy-based approaches are arguably the most popular approaches in Safe RL. They usually consist in extending a known RL algorithm like PPO (Schulman et al. 2017), TRPO (Schulman et al. 2015), or SAC (Haarnoja et al. 2018) to Safe RL using a lagrangian (Ray, Achiam, and Amodei 2019), or in enforcing the constraint by changing the objective function (Liu, Ding, and Liu 2020) or modifying the update process (Zhang, Vuong, and Ross 2020), without introducing a dual variable. Some of these algorithms have become widely used for benchmarking against other new Safe RL algorithms, and are implemented in several state-of-the-art Safe RL frameworks (Ray, Achiam, and Amodei 2019; Ji et al. 2023a,b). We introduce in the following other, more specific, approaches that relate to ours.

Shielding. A *shield* is a system that restricts the action of the agent during learning and at test time, to ensure its safety (Alshiekh et al. 2018; Jansen et al. 2020; Elsayed-Aly et al. 2021; Yang et al. 2023). Shielding was introduced in (Alshiekh et al. 2018), where safety was defined as a formula of the "safety" fragment of LTL. This paper can be considered an extension of (Alshiekh et al. 2018) to the probabilistic setting, since in the case where the agent must be safe with probability 1, the shield we introduce is almost the same as the one defined in (Alshiekh et al. 2018). In (Jansen et al. 2020), the authors already also extend (Alshiekh et al. 2018) to the probabilistic case, but no formal guarantees are provided for the safety of the policy, and our respective methods are significantly different. In particular, their approach may indeed violate safety in practice. A comprehensive survey focused on shielding can be found in (Odriozola-Olalde, Zamalloa, and Arana-Arexolaleiba 2023).

Linear Programming-based approaches. It is well-known that Safe RL can be solved by Linear Programming under certain assumptions (Altman 1999). Several recent works have leveraged Linear Programming to provide statistical guarantees in the model-based case, when the dynamics of the MDP are learned in various contexts. For example, (Efroni, Mannor, and Pirodda 2020) and (Bura et al. 2022) provide algorithms for discounted cumulative rewards and costs, while (Mazumdar, Wisniewski, and Bujorianu 2024) provides an algorithm in the case where safety is defined by a reach-avoid undiscounted property.

State augmentation techniques. State-augmentation techniques with a number representing how far the agent is from being unsafe have also been studied. In (Calvo-Fullana et al. 2024), every state of the MDP is augmented with a Lagrange multiplier. In (Sootla et al. 2022), the states are augmented with the accumulated safety cost up to that state, and are used to reshape the objective. In (Yang et al. 2024), states are also augmented with the accumulated safety cost up to that state, and non-stationary policies depending on that cost are considered. However, in contrast to the aforementioned papers, the agent in our approach is *able to choose* the maximal accumulated safety cost he is able to use in the future depending on the state it goes to. Thus, we do not need to use the augmentation to change the objective function, and are able to provide stricter optimality-preserving and safety guarantees.

Preliminaries

We introduce in this section the mathematical prerequisites necessary for the paper.

Constrained Reinforcement Learning

Markov Decision Processes. A *Markov Decision Process (MDP)* is a tuple $\mathcal{M} = \langle S, A, P, s_{init}, AP, L, R \rangle$, where S is a set of *states*; A is a mapping that associates every state $s \in S$ to a nonempty finite set of *actions* $A(s)$; P is a *transition probability function* that maps every state-action pair (s, a) to a probability measure over S ; $s_{init} \in S$ is the *initial state*¹; AP is a set of *atomic propositions* (or atoms); $L : S \mapsto 2^{AP}$ is a *labeling function*; and $R : S \mapsto \mathbb{R}$ is the *reward function*. For the sake of simplicity, we may write $P(s, a, s')$ instead of $P(s, a)(s')$. An MDP is finite if the sets of states and actions are finite.

A finite (resp. infinite) *path* in \mathcal{M} is a finite (resp. infinite) word $\zeta = s_0 a_0 \cdots s_{n-1} a_{n-1} s_n$ (resp. $\zeta = s_0 a_0 \cdots s_n a_n \cdots$) such that $s_0 = s_{init}$, and such that for any positive integer $i \leq n$ (resp. for any positive integer i), s_i is a state of \mathcal{M} , a_{i-1} is an action in $A(s_{i-1})$, and s_i is in the support of $P(s_{i-1}, a_{i-1})$. In addition, for any finite path $\zeta = s_0 a_0 \cdots s_{n-1} a_{n-1} s_n$ of \mathcal{M} , we let $\text{first}(\zeta) = s_0$, we let $\text{last}(\zeta) = s_n$, and we let $\text{Paths}(\mathcal{M})$ denote the set of *infinite* paths of \mathcal{M} . A *policy* π of \mathcal{M} is a mapping that associates any finite path ζ of \mathcal{M} to an element of $\mathbb{D}(A(\text{last}(\zeta)))$, where $\mathbb{D}(E)$ is the set of all probability measures over E . It is memoryless if $\pi(\zeta)$ only depends on $\text{last}(\zeta)$. It is deterministic if for any finite path ζ of \mathcal{M} , $\pi(\zeta)$ is a Dirac measure. For any policy π of \mathcal{M} , for any state $s \in S$, we let \mathcal{M}_π^s denote the *Markov chain* induced by π in \mathcal{M} starting from state s , we let \mathcal{M}_π denote $\mathcal{M}_\pi^{s_{init}}$, and we let P_π denote the transition function of \mathcal{M}_π . We denote the usual probability measure induced by the Markov chain \mathcal{M}_π^s on $\text{Paths}(\mathcal{M})$ by $\text{prob}_{\mathcal{M}, \pi}^s$. For more details on MDPs and induced Markov

¹This can be assumed wlog compared to a model with an *initial probability distribution* since it is always possible to add a new initial state to such a model with an action from this initial state whose associated probability distribution is the aforementioned initial probability distribution.

chains, see (Baier and Katoen 2008; Bertsekas and Shreve 2007).

Reinforcement Learning. For any random variable $X : \text{Paths}(\mathcal{M}) \mapsto \mathbb{R}$, we let $\mathbb{E}_{\mathcal{M},\pi}^s(X)$ denote the expectation of X with respect to the probability measure $\text{prob}_{\mathcal{M},\pi}^s$, and we let $\mathbb{E}_{\mathcal{M},\pi}(X)$ denote $\mathbb{E}_{\mathcal{M},\pi}^{s_{\text{init}}}(X)$. In addition, when there is no ambiguity, we usually write $\mathbb{E}_{\mathcal{M},\pi}^s(\bullet)$ for $\mathbb{E}_{\mathcal{M},\pi}^s(s_0 a_0 \cdots s_n a_n \cdots \mapsto \bullet)$.

In Reinforcement Learning, we usually solve the following problem: given an MDP \mathcal{M} , and a discount factor $0 < \gamma < 1$, find a policy π^* such that $J(\pi^*) = \max_{\pi} J(\pi)$, where

$$J(\pi) = \mathbb{E}_{\mathcal{M},\pi}^s \left(\sum_{t \in \mathbb{N}} \gamma^t R(s_t) \right).$$

In recent years, many algorithms have been proposed to solve the above problem. Proximal Policy Optimization (PPO) (Schulman et al. 2017), Asynchronous Advantage Actor Critic (A3C) (Mnih et al. 2016), or Soft Actor-Critic (SAC) (Haarnoja et al. 2018), are among the most popular, and we leverage these algorithms in our experiments.

Probabilistic Reachability Goals. In contrast to discounted objectives, reachability goals are a simple form of undiscounted and infinite horizon objectives, that we use as a constraint for MDPs. For any MDP \mathcal{M} , any state s and policy π of \mathcal{M} , any finite path $\zeta = s_0 a_0 \cdots s_{n-1} a_{n-1} s_n$ (resp. infinite path $\zeta = s_0 a_0 \cdots s_n a_n \cdots$) in \mathcal{M} , we write $\zeta \models \text{Reach}(\mathbf{c})$ if there exists $i \in \leq n$ (resp. $i \in \mathbb{N}$) such that $L(s_i) = \mathbf{c}$. Then, $\mathcal{M}_{\pi} \models \mathbb{P}_{\leq p}(\text{Reach}(\mathbf{c}))$ when the property $\text{prob}_{\mathcal{M},\pi}^s \{ \zeta \in \text{Paths}(\mathcal{M}_{\pi}^s) \mid \zeta \models \text{Reach}(\mathbf{c}) \} \leq p$ is true.

Probabilistic Shielding

In this section, we introduce a new theoretical framework for probabilistic shielding, and show that it gives safety and optimality guarantees.

Problem Statement

We assume in the rest of the paper that all the labelling functions of the MDPs considered take values in $\{\mathbf{s}, \mathbf{u}\}$, where *safe* states are labelled by \mathbf{s} and *unsafe* states are labelled by \mathbf{u} .

Definition 1 (Reachability-Constrained Optimization Problem (RCOP)). *Given a finite MDP \mathcal{M} , a safety threshold $0 \leq p \leq 1$, and a discount factor $0 < \gamma < 1$, find a policy π^* such that $\mathcal{M}_{\pi^*} \models \mathbb{P}_{\leq p}(\text{Reach}(\mathbf{u}))$ and such that π^* is optimal among all policies π satisfying $\mathcal{M}_{\pi} \models \mathbb{P}_{\leq p}(\text{Reach}(\mathbf{u}))$, i.e., such that*

$$J(\pi^*) = \max_{\{\pi \mid \mathcal{M}_{\pi} \models \mathbb{P}_{\leq p}(\text{Reach}(\mathbf{u}))\}} (J(\pi)).$$

The above problem is a form of generalisation of the problem considered in (Alshiekh et al. 2018). Furthermore, variants of the above problem, i.e RL with infinite-horizon undiscounted safety properties, are considered in

many books and papers (see for example (Bertsekas and Shreve 2007; Altman 1999; Jansen et al. 2020; Mazumdar, Wisniewski, and Bujorianu 2024; Mqirmi, Belardinelli, and León 2021)). However, a significant difference between the problem we consider and most Safe RL approaches is that, similarly to (Yang et al. 2024), we make the choice of including non-stationary policies in the problem. We make this choice because in our context, where the discount factor of the reward (which is less than 1) and the discount factor of the constraint cost (which is equal to 1) are not equal, optimal memoryless policies are not guaranteed to exist (Altman 1999).

Method Overview

In order to tackle RCOP, we compute, for all states of the MDP, an approximation of the minimal probability of reaching, from that state, an unsafe state. More precisely, we define $\beta_{\mathcal{M}}$ as the mapping such that for every state s of the MDP \mathcal{M} , $\beta_{\mathcal{M}}(s)$ is equal to

$$\min_{\pi} \text{prob}_{\mathcal{M},\pi}^s \{ \zeta \in \text{Paths}(\mathcal{M}_{\pi}^s) \mid \zeta \models \text{Reach}(\mathbf{u}) \}.$$

The mapping $\beta_{\mathcal{M}}$ is the smallest fixed point of the following equation (Baier and Katoen 2008),

$$\beta(s) = \begin{cases} 1 & \text{if } L(s) = \mathbf{u} \\ (\mathcal{B}_{\mathcal{M}}(\beta))(s) & \text{otherwise,} \end{cases}$$

where $(\mathcal{B}_{\mathcal{M}}(\beta))(s) = \min_{a \in A} \sum_{s' \in S} P(s, a, s') \beta(s')$.

This fixed point can be computed with linear programming (Forejt et al. 2011) in polynomial time (Karmarkar 1984). In practice, this approach is inefficient and state-of-the-art methods rely on *value iteration*, i.e., iterating the operator $\mathcal{B}_{\mathcal{M}}$ from β_0 such that $\beta_0(s) = 1$ if $L(s) = \mathbf{u}$, and $\beta_0(s) = 0$ otherwise (Sutton and Barto 2018) to compute an approximation of $\beta_{\mathcal{M}}$. However, value iteration as previously described, if stopped before convergence, only gives a lower bound on $\beta_{\mathcal{M}}$, whereas an upper bound is needed to provide safety guarantees in our approach.

Definition 2. *For any MDP \mathcal{M} , and any $\epsilon \geq 0$, an inductive ϵ -upper bound of $\beta_{\mathcal{M}}$ is a mapping β that associates to any state s of \mathcal{M} a number in $[0; 1]$ such that for all states s , $0 \leq \beta(s) - \beta_{\mathcal{M}}(s) \leq \epsilon$, and $(\mathcal{B}_{\mathcal{M}}(\beta))(s) \leq \beta(s)$.*

The first step of our approach thus consists in computing an inductive ϵ -upper bound β of $\beta_{\mathcal{M}}$, with a small ϵ . To our knowledge, the fastest algorithms for that purpose in the general case are Interval Iteration (Haddad and Monmege 2018), Sound Value Iteration (Quatmann and Katoen 2018) and Optimistic Value Iteration (Hartmanns and Kaminski 2020), with no clear overall faster one (Hartmanns and Kaminski 2020).

Once β is computed, we construct a shield $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ by augmenting every state of the MDP \mathcal{M} with a real number in $[0; 1]$, that is, a “safety level” representing intuitively a maximal probability of reaching an unsafe state from the current state while following *any* actions. Thus, any action (a, α) taken in $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ is composed of an action a of \mathcal{M} , together with predictions $\alpha \in [0; 1]$, that may depend on the current “safety level”, as to what the next “safety levels” will be. Furthermore, $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ is defined so that:

1. The predictions must be coherent, *i.e.* that the sum of the next “safety levels” as predicted by α , pondered with the probabilities given by action a , is less than or equal to the current “safety level”.
2. The predicted “safety levels” cannot be less than the inductive ϵ -upper bound β of the minimal probability $\beta_{\mathcal{M}}(s)$ of reaching in \mathcal{M} an unsafe state from the current state s .

Finally, we learn a policy in the constructed shield.

The Shield: Safety and Optimality Guarantees

We now give a formal definition of the shield used in our approach, and justify that our approach is theoretically sound. In the following, we let \mathcal{M} be an MDP, $\gamma \in [0; 1]$ be a discount factor, $p \in [0; 1]$, $\epsilon \in \mathbb{R}^+$, and we let β be an inductive ϵ -upper bound of $\beta_{\mathcal{M}}$ such that $\beta(s_{init}) \leq p$. Notice that if such a β does not exist, RCOP is unfeasible. Moreover, for any $s \in S$, we let χ^s denote the polytope in $\mathbb{R}^{A(s)}$ representing probability distributions over $A(s)$, *i.e.* the set of all $x \in \mathbb{R}^{A(s)}$ such that $\sum_{a \in A(s)} x_a = 1$ and $x_a \geq 0$ for any

$a \in A(s)$, and for any $s \in S$, for any mapping α from S to $[0; 1]$, any $q \in [\beta(s); 1]$, we let $A_{\alpha}^{s,q}$ denote the half-space representing Condition 2 above with s being the current state and q being the current “safety level”, *i.e.* we let $A_{\alpha}^{s,q}$ denote the set of all $x \in \mathbb{R}^{A(s)}$ such that

$$\sum_{a \in A(s)} x_a \left(q - \sum_{s' \in S} P(s, a, s') \alpha(s') \right) \geq 0.$$

Finally, we let $C_{\alpha}^{s,q}$ be the polytope of probability distributions satisfying Condition 2, *i.e.* the polytope defined by $\chi^s \cap A_{\alpha}^{s,q}$, we let $V_{\alpha}^{s,q}$ be the (finite) set of vertices of $C_{\alpha}^{s,q}$, and we let $X = \prod_{s \in S} [\beta(s); 1]$.

Definition 3 (The Shield). *We let $Sh_{\beta}^{\leq p}(\mathcal{M})$ be the MDP \mathcal{M}' with*

- set of states $S' = \{(s, q) \mid s \in S, q \in [\beta(s); 1]\}$;
- sets of actions $A'(s, q) = \bigcup_{\alpha \in X} \bigcup_{v \in V_{\alpha}^{s,q}} (\alpha, v)$;
- initial state $s'_{init} = (s_{init}, p)$;
- labelling $L'(s, x) = L(s)$;
- reward $R'(s, x) = R(s)$;
- transition probability function P' such that for any $(s, q) \in S'$, any $\alpha \in X$, and any $v \in V_{\alpha}^{s,q}$, $P'((s, q), (\alpha, v))$ is equal to

$$\sum_{s' \in S} \delta_{(s', \alpha(s'))} \left(\sum_{a \in A(s)} v_a P(s, a, s') \right)$$

where $\delta_{(s', \alpha(s'))}$ is the Dirac measure on S' with support $\{(s', \alpha(s'))\}$.

In the above definition, α corresponds to the “safety levels” of the next states chosen by the agent, the definition of X guarantees that Condition 2 is satisfied, and the polytope $C_{\alpha}^{s,q}$ corresponds to all combinations of actions that satisfy Condition 1. Notice that $Sh_{\beta}^{\leq p}(\mathcal{M})$ is indeed an MDP since

for every $(s, q) \in S'$, $V_{\beta}^{s,q}$ is nonempty because β is inductive, *i.e.* because $\beta_{\mathcal{M}}(\beta) \leq \beta$.

We now show that any memoryless policy in the shield is safe. The proof of the following theorem is inspired by the proof of Theorem 10.15 in (Baier and Katoen 2008).

Theorem 1 (Safety guarantee in any shield). *For any memoryless policy π in $Sh_{\beta}^{\leq p}(\mathcal{M})$, we have*

$$Sh_{\beta}^{\leq p}(\mathcal{M})_{\pi} \models \mathbb{P}_{\leq p}(\text{Reach}(\mathbf{u})).$$

We now justify that we can use an optimal policy of the shield to find a policy of the original MDP that is safe, and whose expected cumulative reward is close to a solution of RCOP. The closer β is to $\beta_{\mathcal{M}}$, the closer the expected cumulative reward of the policy obtained from our approach will be to the expected cumulative reward of a solution of RCOP.

For any memoryless policy π of $Sh_{\beta}^{\leq p}(\mathcal{M})$, we let $\hat{\pi}$ denote the policy of \mathcal{M} such that $\hat{\pi}(s_0 \cdots s_n) = \mu_n$ where $s_0 = s_{init}$, $q_0 = p$, $(\alpha^{i+1}, v^{i+1}) = \pi(s_i, q_i)$, $q_{i+1} = \alpha_{s_{i+1}}^{i+1}$, and $\mu_n(a) = v_a^{n+1}$. It is easy to see that $J(\hat{\pi}) = J(\pi)$ and that

$$\begin{aligned} \text{prob}_{\mathcal{M}, \hat{\pi}}^{s_{init}} (\zeta \in \text{Paths}(\mathcal{M}) \mid \zeta \models \text{Reach}(\mathbf{u})) = \\ \text{prob}_{Sh_{\beta}^{\leq p}(\mathcal{M}), \pi}^{(s_{init}, p)} (\zeta \in \text{Paths}(Sh_{\beta}^{\leq p}(\mathcal{M})) \\ \mid \zeta \models \text{Reach}(\mathbf{u})). \end{aligned}$$

Thus, as a corollary of Theorem 1, if π is a memoryless policy of $Sh_{\beta}^{\leq p}(\mathcal{M})$, $\hat{\pi}$ is safe.

Corollary 1 (Safety guarantee in the original MDP). *If π is a memoryless policy of $Sh_{\beta}^{\leq p}(\mathcal{M})$, then*

$$\mathcal{M}_{\hat{\pi}} \models \mathbb{P}_{\leq p}(\text{Reach}(\mathbf{u})).$$

We let $\mathcal{B}(\mathcal{M})$ be the set of inductive upper bounds of $\beta_{\mathcal{M}}$, that we equip with the norm $\|\cdot\|_{\infty}$ such that $\|\beta_1 - \beta_2\|_{\infty}$ is the maximum of $|\beta_1(s) - \beta_2(s)|$ for all states s of \mathcal{M} .

Assumption 1 (Slater’s condition). *There exists a policy π in \mathcal{M} and a number $q < p$ such that*

$$\mathcal{M}_{\pi} \models \mathbb{P}_{\leq q}(\text{Reach}(\mathbf{u})).$$

Theorem 2 (Optimality-preserving guarantees). *We have the three following properties.*

1. For any $\epsilon > 0$, for any inductive ϵ -upper bound β of $\beta_{\mathcal{M}}$, there exists an optimal, memoryless, and deterministic policy π_{β}^* of $Sh_{\beta}^{\leq p}(\mathcal{M})$.
2. The policy $\widehat{\pi_{\beta}^*}$ is a solution to RCOP.
3. If Assumption 1 holds, then

$$\lim_{\beta \in \mathcal{B}(\mathcal{M}), \beta \rightarrow \beta_{\mathcal{M}}} J(\widehat{\pi_{\beta}^*}) = J(\widehat{\pi_{\beta_{\mathcal{M}}}^*}).$$

Discussion. Definition 3 allows us to construct a shield from any MDP \mathcal{M} with known safety dynamics via an algorithm that computes an inductive upper bound of $\beta_{\mathcal{M}}$. Theorem 1 and Corollary 1 show that if we train an agent using the shield, the agent will be safe. Furthermore, Theorem 2

Algorithm 1: Probabilistic Shielding

- 1: **Input:** An MDP \mathcal{M} , a discount factor γ , an uncertainty threshold ϵ , a safety threshold p .
- 2: Compute an inductive ϵ -upper bound β of

$$\beta_{\mathcal{M}}(s) = \max_{\pi} \text{prob}_{\mathcal{M}, \pi}^s \{ \zeta \mid \zeta \models \text{Reach}(\mathbf{u}) \}$$

- 3: Construct the shield $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$
 - 4: Learn a memoryless policy π^* in $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ with an RL algorithm.
 - 5: **Return** $\widehat{\pi}^*$.
-

justifies that training an agent with the shield yields a cumulative reward close to optimal.

For the sake of simplicity, we made the choice of presenting our shielding approach in the case where the full dynamics of the MDP is known. However, every definition can be straightforwardly adapted to an MDP where only the safety dynamics, *i.e.* a quotient of the MDP containing all of the safety-relevant information, is known. We make use of that adaptation in our experiments. The assumption of knowing the safety dynamics is strong, but is adopted in several papers, and in particular in the majority of shielding methods (see (Alshiekh et al. 2018; Elsayed-Aly et al. 2021; He, León, and Belardinelli 2022) for example), and could be alleviated in the future by introducing a three-step algorithm that at each iteration, learns a better conservative estimation of the safety dynamics, changes $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ according to that estimation, and does a step of policy iteration in $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$. The size of the state and action space of the shield is bigger than the state and action space of the original MDP, and thus may lead to slower convergence than state-of-the-art Safe RL algorithms. However, the safety of the agent after computing β is guaranteed, and the only constraint violations that may thus occur in a real-life scenario occur when computing β . This is one of the strictest guarantees possible for constraint violations in Safe RL as β only depends on the safety dynamics of the MDP, and could be theoretically be computed with any ϵ -greedy safe policy. Thus, if an ϵ -greedy safe policy is known in advance and used to compute β , the algorithm incurs *exactly zero* constraints violations. This strict guarantee is, to our knowledge, offered in a more scalable way compared to previous Safe RL algorithms that usually use to that end Linear Programming (as in (Liu et al. 2021) for example).

Implementation

We suppose in the following, without any loss of generality, that for any $s \in S$, there exists an integer d such that $\#A(s) = d$, and we let $\{a_1^s, \dots, a_d^s\}$ denote $A(s)$. In a gym environment, the policy that the agent follows is output by a neural network. However, even if the sets $A(s)$ all have the same size, this does not guarantee that a probability distribution over a set $A'(s)$ (a set of actions of $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$), can be directly output by a neural network, since the sets $V_{\alpha}^{s,q}$ do not necessarily all have the same size, even if s and q

are fixed. Therefore, to implement the shield as a gym environment, we change the MDP $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ into an encoded MDP $\text{En}_{\beta}^{\leq p}(\mathcal{M})$ that is equivalent, *i.e.* such that every policy of $\text{En}_{\beta}^{\leq p}(\mathcal{M})$ can be transformed into a policy of $\text{Sh}_{\beta}^{\leq p}(\mathcal{M})$ and the converse. To avoid instability, the MDP $\text{En}_{\beta}^{\leq p}(\mathcal{M})$ is constructed so that the dependency of the probabilistic transition function on the state-action pair is as continuous as possible. The results obtained show that this approach scales well. For the sake of simplicity, we do not define $\text{En}_{\beta}^{\leq p}(\mathcal{M})$ entirely, but we give in the following the main technical idea of $\text{En}_{\beta}^{\leq p}(\mathcal{M})$, which is a way of mapping the set $V_{\alpha}^{s,q}$ to a larger set of fixed size, so that the dependency of the probabilistic transition function on the state-action pair is roughly continuous. We give such a mapping g below.

Formally, g associates to any (s, q, α, i, j) such that $s \in S$, $q \in [\beta(s); 1]$, $\alpha \in X$, $i, j \in \{1, \dots, d\}$, and $V_{\alpha}^{s,q}$ is nonempty, an element of $V_{\alpha}^{s,q}$. Intuitively, if we let χ_i^s denote the element of $[0; 1]^{A(s)}$ such that $\chi_i^s(a) = 1$ if $a = a_i^s$ and 0 otherwise, $g(s, q, \alpha, i, j)$ corresponds to the intersection between the border of the half-space $A_{\alpha}^{s,q}$ and the line between χ_i^s and χ_j^s if there is one, to χ_i^s if χ_i^s is in $A_{\alpha}^{s,q}$, or to a means of the points in $V_{\alpha}^{s,q}$ weighted by the minimum of their distances to χ_i^s and χ_j^s otherwise. A formal definition is given below.

- If $i = j$,
 - if $\chi_i^s \in A_{\alpha}^{s,q}$, $g(s, q, \alpha, i, j) = \chi_i^s$,
 - otherwise

$$g(s, q, \alpha, i, j) = \frac{\sum_{v \in V_{\alpha}^{s,q}} \frac{1}{\|\chi_i^s - v\|} v}{\sum_{v \in V_{\alpha}^{s,q}} \frac{1}{\|\chi_i^s - v\|}}$$

- Otherwise, if $i \neq j$,
 - if $\chi_i^s \in A_{\alpha}^{s,q}$, $g(s, q, \alpha, i, j) = \chi_i^s$,
 - otherwise if $\chi_j^s \in A_{\alpha}^{s,q}$, $g(s, q, \alpha, i, j)$ is defined as $\lambda_{max} \chi_i^s + (1 - \lambda_{max}) \chi_j^s$ where λ_{max} is the maximal $\lambda \in [0; 1]$ such that $\lambda \chi_i^s + (1 - \lambda) \chi_j^s \in A_{\alpha}^{s,q}$ (notice that $g(s, q, \alpha, i, j) \in V_{\alpha}^{s,q}$ in that case),
 - and otherwise

$$g(s, q, \alpha, i, j) = \frac{\sum_{v \in V_{\alpha}^{s,q}} \frac{1}{\min(\|\chi_i^s - v\|, \|\chi_j^s - v\|)} v}{\sum_{v \in V_{\alpha}^{s,q}} \frac{1}{\min(\|\chi_i^s - v\|, \|\chi_j^s - v\|)}}.$$

Since the convex polytope $C_{\alpha}^{s,q}$ whose set of vertices is $V_{\alpha}^{s,q}$ is the intersection of the polytope χ whose set of vertices is $(\chi_i)_{i \in \{1, \dots, d\}}$, and of the half-space $A_{\alpha}^{s,q}$, it is easy to see that the elements of $V_{\alpha}^{s,q}$ are all on the edges of χ . As a consequence, we have

$$V_{\alpha}^{s,q} \subseteq \{g(s, q, \alpha, i, j) \mid i, j \in \{1, \dots, d\}\}$$

for any $(s, q) \in S'$ and any $\alpha \in X$, such that $V_{\alpha}^{s,q}$ is nonempty.

Experiments

We demonstrate the viability of our approach with four case studies. The algorithm used to compute an inductive ϵ -upper

	Media ...	Colour bomb	Color bomb v2	Bridge	Bridge v2	Pacman
random_action_probability	-	0.1	0.1	0.04	0.04	-
episode_length	40	100	250	600	600	1000
total_timesteps	25k	25k	100k	200k	200k	500k
safety_bound	0.001	0.05	0.05	0.01	0.01	0.01
action_space_size	2	4	4	4	4	5
state_space_size	462	81	900	400	400	100k

Table 1: Environment Parameters

bound of β_M is Interval Iteration (Haddad and Monmege 2018), which is simple in our case as the end components of the MDPs corresponding to the environments are trivial. We use PPO (Schulman et al. 2017) as an RL algorithm to find an optimal policy in the shield. We demonstrate the viability of our approach with five case studies of increasing complexity. For each case study, we compare the safety and the cumulative reward given at each epoch by unshielded PPO (Schulman et al. 2017), PPO-shield (our approach), PPO-Lagrangian (Ray, Achiam, and Amodei 2019), a combination of a lagrangian approach and PPO, and CPO (Achiam et al. 2017). We use Omnisafe (Ji et al. 2023b) for the implementation of PPO-Lagrangian and CPO.

Environment descriptions

We provide descriptions for each of our testing environments below. For the gridworld environments, the agent has access to four actions in every state (except for the terminal one), which are $\{left, right, up, down\}$. Every action carries a probability `random_action_probability` of choosing randomly, in a uniform manner, another direction. For example, the action *left* makes the agent go left with probability $1 - \text{random_action_probability}$, and the agent goes right, up, and down with remaining probability $\text{random_action_probability}/3$. Furthermore, safety in all the environments is defined as avoiding the unsafe states with probability at least $1 - \text{safety_bound}$.

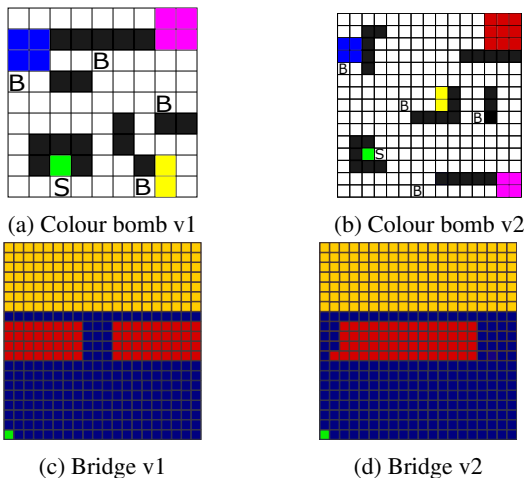


Figure 1: Gridworld Environments

Table 1 details the parameters of each of our environments including this random probability, the maximum episode length, the total number of interactions (or timesteps) and the safety bound. We also provide illustrations of the relevant environments in Figure 1.

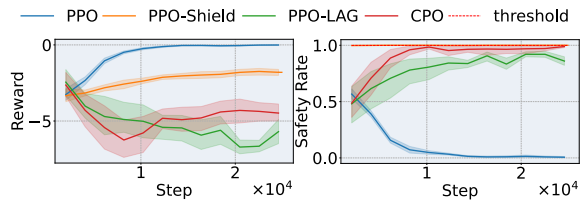
Media streaming The agent is tasked with managing a data buffer. The data buffer has size 20, with packets leaving the data buffer according to a Bernoulli process with rate $\mu_{out} = 0.7$. The agent has two actions $A = \{fast, slow\}$ which fill the data buffer with new packets according to a Bernoulli process with rates $\mu_{fast} = 0.9$ and $\mu_{slow} = 0.1$ respectively. The goal is to minimise the outage time: if the data buffer is empty, the agent receives a reward of -1 and 0 otherwise. The state space is augmented with a cost c which corresponds to the number of times the action *fast* is used. The unsafe states are all the states corresponding to a total number of *fast* actions used above the threshold $C = \lfloor \text{episode_length}/2 \rfloor$. Thus, the agent must avoid using more than C *fast* actions with high probability. A similar environment has been considered in (Bura et al. 2022).

Colour bomb gridworld v1 The agent operates in a 9×9 gridworld (see Fig. 1a). Upon reaching a coloured zone that is yellow, blue or pink, the agent receives a reward of $+1$ and the episode terminates. Alternatively, when reaching the green or red zones, the agent can choose either to stay inside of them, or to go to any white square that borders. All other states provide a reward of 0. The unsafe states are the bombs labelled as B states (S denotes the starting state). A similar environment has been used in (Alshiekh et al. 2018), albeit with a hard safety constraint instead of a probabilistic one.

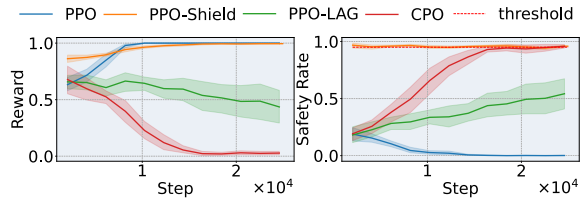
Colour bomb gridworld v2 The agent operates in a 15×15 gridworld (see Fig. 1b), similar to the previous environment. However, in contrast to the previous environment, the non-green coloured zones that give a reward of $+1$ and terminate the episode are randomised, either at the start of an episode or when the agent enters the green zone.

Bridge crossing (v1 and v2) The agent operates in a 20×20 gridworld (see Fig. 1c). The goal is to cross the bridge to the safe terminal yellow states, which provide a reward of $+1$. The unsafe states are the red states (lava), and the agent must thus avoid falling in lava with high probability. The start state is denoted by the green square. Bridge crossing v1 has been used in (Mitta et al. 2024).

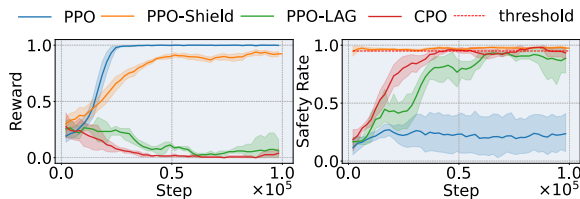
Pacman We also consider a 15×19 pacman environment inspired by (Racanière et al. 2017), with one ghost, and col-



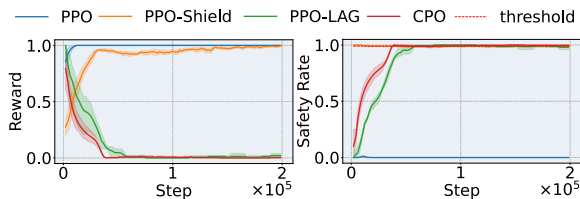
(a) Media streaming



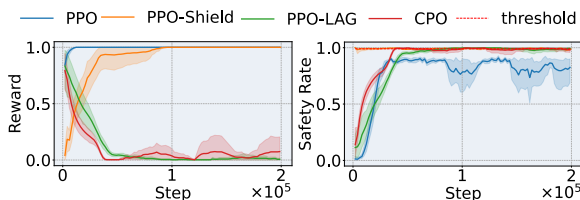
(b) Colour bomb gridworld v1



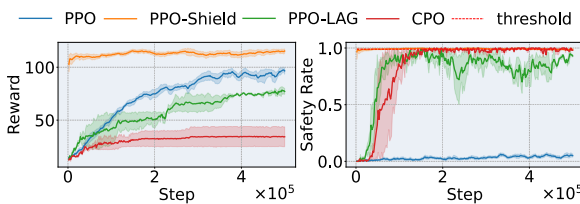
(c) Colour bomb gridworld v2



(d) Bridge crossing v1



(e) Bridge crossing v2



(f) Pacman

Figure 2: Learning curves

lectible coins (+1 reward) in every position (no food). Taking in to consideration all possible locations and directions of the ghost and the agent, and the locations of the coins the state space is combinatorially large, although similar to

(Alshiekh et al. 2018), we can leverage a *safety abstraction* of the environment (ignoring the dynamics of the coins) for efficient interval iteration. We note that even with the safety abstraction the total number of states exceeds 100k, demonstrating that our approach is still feasible for large state spaces. The goal is to collect as many coins while avoiding the (unsafe) ghosts for the duration of the episode.

Results

Figure 2 presents the results of our experiments. In every environment, PPO-shield does indeed guarantee safety throughout training and at test time. In terms of cumulative reward, PPO-shield converges to the expected value of 1 (or almost 1) in the Colour bomb gridworld v1 and v2, and Bridge crossing v1 and v2 environments. Furthermore, in the media streaming environment, where there is a trade-off between safety and reward, PPO-shield still improves to an expected strictly negative value. In terms of rate of convergence, PPO-shield converges slightly slower than PPO in every environment except for the Bridge Crossing environments where PPO-shield converges significantly slower and PPO converges immediately. This can be explained by the fact that, if not considering safety, the optimal path for the agent in the Bridge Crossing environments is to go straight up, whereas if considering safety, to get an optimal reward, the agent has to find the path across the bridge in Bridge Crossing v1, and the path that goes around the lava to the right in Bridge Crossing v2, correctly evaluating that the straight-up path is too risky. Thus, in these cases, safety is very restrictive, which may explain the longer convergence time. Overall, the rate of convergence of PPO-shield remains fast, requiring a maximum of 100 000 steps in all of our case studies.

We can also see that PPO-shield significantly outperforms CPO and PPO-Lagrangian in all of the case studies. Even though CPO and PPO-Lagrangian both seem to learn the constraint correctly, neither of them manages to optimize the reward in every single one of our case studies. This might be due to the fact that these algorithms are slow to converge when the safety requirement is very restrictive.

Conclusion

We have developed a shielding approach for Safe RL with probabilistic state-avoidance constraints. We have shown that this approach is theoretically sound, and offers strict safety guarantees. Furthermore, this approach relies on Value Iteration on the safety dynamics of the MDP, which is known to be scalable, and allows to decouple the safety dynamics and the reward dynamics of the MDP, in contrast to Safe RL approaches based on Linear Programming. In addition, our experiments show that our method is viable in practice and can significantly outperform state-of-the-art Safe RL algorithms.

Acknowledgements

This paper was supported by the EPSRC grant number EP/X015823/1, titled "An abstraction-based technique for Safe Reinforcement Learning".

References

- Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained Policy Optimization. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 22–31. PMLR.
- Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe Reinforcement Learning via Shielding. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2669–2678. AAAI Press.
- Altman, E. 1999. *Constrained Markov Decision Processes*, volume 7. CRC Press.
- Baier, C.; and Katoen, J.-P. 2008. *Principles of model checking*. MIT press.
- Bertsekas, D. P.; and Shreve, S. E. 2007. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific. ISBN 1886529035.
- Bura, A.; HasanzadeZonuzi, A.; Kalathil, D.; Shakkottai, S.; and Chamberland, J. 2022. DOPE: Doubly Optimistic and Pessimistic Exploration for Safe Reinforcement Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Calvo-Fullana, M.; Paternain, S.; Chamon, L. F. O.; and Ribeiro, A. 2024. State Augmented Constrained Reinforcement Learning: Overcoming the Limitations of Learning With Rewards. *IEEE Trans. Autom. Control.*, 69(7): 4275–4290.
- Efroni, Y.; Mannor, S.; and Pirodda, M. 2020. Exploration-Exploitation in Constrained MDPs. *CoRR*, abs/2003.02189.
- Elsayed-Aly, I.; Bharadwaj, S.; Amato, C.; Ehlers, R.; Topcu, U.; and Feng, L. 2021. Safe Multi-Agent Reinforcement Learning via Shielding. In Dignum, F.; Lomuscio, A.; Endriss, U.; and Nowé, A., eds., *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, 483–491. ACM.
- Forejt, V.; Kwiatkowska, M.; Norman, G.; and Parker, D. 2011. *Automated Verification Techniques for Probabilistic Systems*, 53–113. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-21455-4.
- Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; and Knoll, A. 2024. A Review of Safe Reinforcement Learning: Methods, Theories, and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12): 11216–11235.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 1856–1865. PMLR.
- Haddad, S.; and Monmege, B. 2018. Interval iteration algorithm for MDPs and IMDPs. *Theoretical Computer Science*, 735: 111–131. Reachability Problems 2014: Special Issue.
- Hartmanns, A.; and Kaminski, B. L. 2020. Optimistic Value Iteration. In Lahiri, S. K.; and Wang, C., eds., *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II*, volume 12225 of *Lecture Notes in Computer Science*, 488–511. Springer.
- He, C.; León, B. G.; and Belardinelli, F. 2022. Do Androids Dream of Electric Fences? Safety-Aware Reinforcement Learning with Latent Shielding. In Pedroza, G.; Hernández-Orallo, J.; Chen, X. C.; Huang, X.; Espinoza, H.; Castillo-Effen, M.; McDermid, J. A.; Mallah, R.; and hÉigeartaigh, S. Ó., eds., *Proceedings of the Workshop on Artificial Intelligence Safety 2022 (SafeAI 2022) co-located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI2022), Virtual, February, 2022*, volume 3087 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Jansen, N.; Könighofer, B.; Junges, S.; Serban, A.; and Bloem, R. 2020. Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper). In Konnov, I.; and Kovács, L., eds., *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPICs*, 3:1–3:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Ji, J.; Zhang, B.; Zhou, J.; Pan, X.; Huang, W.; Sun, R.; Geng, Y.; Zhong, Y.; Dai, J.; and Yang, Y. 2023a. Safety-Gymnasium: A Unified Safe Reinforcement Learning Benchmark. arXiv:2310.12567.
- Ji, J.; Zhou, J.; Zhang, B.; Dai, J.; Pan, X.; Sun, R.; Huang, W.; Geng, Y.; Liu, M.; and Yang, Y. 2023b. OmniSafe: An Infrastructure for Accelerating Safe Reinforcement Learning Research. arXiv:2305.09304.
- Karmarkar, N. 1984. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, STOC '84*. New York, NY, USA: Association for Computing Machinery. ISBN 0897911334.
- Kober, J.; Bagnell, J.; and Peters, J. 2013. Reinforcement Learning in Robotics: A Survey. *The International Journal of Robotics Research*, 32: 1238–1274.
- Liu, T.; Zhou, R.; Kalathil, D.; Kumar, P. R.; and Tian, C. 2021. Learning Policies with Zero or Bounded Constraint Violation for Constrained MDPs. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 17183–17193.
- Liu, Y.; Ding, J.; and Liu, X. 2020. IPO: Interior-Point Policy Optimization under Constraints. In *The Thirty-Fourth*

- AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, 4940–4947. AAAI Press.
- Mazumdar, A.; Wisniewski, R.; and Bujorianu, M. 2024. Safe Reinforcement Learning for Constrained Markov Decision Processes with Stochastic Stopping Time. *CoRR*, abs/2403.15928.
- Mitta, R.; Hasanbeig, H.; Wang, J.; Kroening, D.; Kantaros, Y.; and Abate, A. 2024. Safeguarded Progress in Reinforcement Learning: Safe Bayesian Exploration for Control Policy Synthesis. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 21412–21419. AAAI Press.
- Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous Methods for Deep Reinforcement Learning. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1928–1937. JMLR.org.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR*, abs/1312.5602.
- Mqirmi, P. E.; Belardinelli, F.; and León, B. G. 2021. An Abstraction-based Method to Check Multi-Agent Deep Reinforcement-Learning Behaviors. In Dignum, F.; Lomuscio, A.; Endriss, U.; and Nowé, A., eds., *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, 474–482. ACM.
- Odrizola-Olalde, H.; Zamalloa, M.; and Arana-Arexolaleiba, N. 2023. Shielded Reinforcement Learning: A review of reactive methods for safe learning. In *2023 IEEE/SICE International Symposium on System Integration (SII)*, 1–8.
- Quatmann, T.; and Katoen, J. 2018. Sound Value Iteration. In Chockler, H.; and Weissenbacher, G., eds., *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I*, volume 10981 of *Lecture Notes in Computer Science*, 643–661. Springer.
- Racanière, S.; Weber, T.; Reichert, D.; Buesing, L.; Guez, A.; Jimenez Rezende, D.; Puigdomènech Badia, A.; Vinyals, O.; Heess, N.; Li, Y.; et al. 2017. Imagination-augmented agents for deep reinforcement learning. *Advances in neural information processing systems*, 30.
- Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking safe exploration in deep reinforcement learning. volume 7, 2.
- Schulman, J.; Levine, S.; Moritz, P.; Jordan, M. I.; and Abbeel, P. 2015. Trust Region Policy Optimization. *CoRR*, abs/1502.05477.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.
- Sootla, A.; Cowen-Rivers, A. I.; Jafferjee, T.; Wang, Z.; Mguni, D. H.; Wang, J.; and Ammar, H. 2022. Saute RL: Almost Surely Safe Reinforcement Learning Using State Augmentation. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 20423–20443. PMLR.
- Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book. ISBN 0262039249.
- Yang, W.; Marra, G.; Rens, G.; and Raedt, L. D. 2023. Safe Reinforcement Learning via Probabilistic Logic Shields. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, 5739–5749. ijcai.org.
- Yang, Z.; Jin, H.; Tang, Y.; and Fan, G. 2024. Risk-Aware Constrained Reinforcement Learning with Non-Stationary Policies. In Dastani, M.; Sichman, J. S.; Alechina, N.; and Dignum, V., eds., *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024, 2029–2037*. International Foundation for Autonomous Agents and Multiagent Systems / ACM.
- Zhang, Y.; Vuong, Q.; and Ross, K. W. 2020. First Order Constrained Optimization in Policy Space. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.