Safeguarded Progress in Reinforcement Learning: Safe Bayesian Exploration for Control Policy Synthesis

Rohan Mitta[1], **Hosein Hasanbeig**[2*], **Jun Wang**[3], **Daniel Kroening**[4*], **Yiannis Kantaros**[3], **Alessandro Abate**[1]
[1]University of Oxford,
[2]Microsoft Research,
[3]Washington University,
[4]Amazon
alessandro.abate@cs.ox.ac.uk

Abstract

This paper addresses the problem of maintaining safety during training in Reinforcement Learning (RL), such that the safety constraint violations are bounded at any point during learning. As enforcing safety during training might severely limit the agent's exploration, we propose here a new architecture that handles the trade-off between efficient progress and safety during exploration. As the exploration progresses, we update via Bayesian inference Dirichlet-Categorical models of the transition probabilities of the Markov decision process that describes the environment dynamics. We then propose a way to approximate moments of belief about the risk associated to the action selection policy. We demonstrate that this approach can be easily interleaved with RL and we present experimental results to showcase the performance of the overall architecture.

1 Introduction

Traditionally, RL is principally concerned with the policy that the agent generates by the end of the learning process. In other words, the quality of agent's policy *during* learning is overlooked at the benefit of learning how to behave optimally, eventually. Accordingly, many standard RL methods rely on the assumption that the agent selects each available action at every state infinitely often during exploration (Sutton, Bach, and Barto 2018; Puterman 2014). A related technical assumption that is often made is that the MDP is *ergodic*, meaning that every state is reachable from every other state under proper action selection (Moldovan and Abbeel 2012). These assumptions might be reasonable, e.g., in virtual environments where restarting is always an option. However, in safety-critical scenarios these assumptions might be unreasonable, as we may explicitly require the agent to never visit certain unsafe states. Indeed, in a variety of RL applications the safety of the agent is particularly important, e.g., when using expensive autonomous platforms or robots that work in the proximity of humans. Thus, researchers are paying increasing attention not only to maximising a long-term task-driven reward, but also to enforcing safety during training.

Related Work The general problem of *Safe* RL has been an active area of research in which numerous approaches and definitions of safety have been proposed (Brunke et al. 2021; Garcia and Fernandez 2015; Pecka and Svoboda 2014). Moldovan and Abbeel (2012) define safety in terms of "actions availability", namely ensuring that an agent is always able to return to its current state. Chow et al. (2018a) pursue safety by minimising a cost associated with worst-case scenarios, when cost is associated with a lack of safety. Similarly, Miryoosefi et al. (2019) define the safety constraint in terms of the expected sum of a vector of measurements to be in a target set. Other approaches (Li and Belta 2019; Hasanbeig, Abate, and Kroening 2019a,b; Hasanbeig, Kroening, and Abate 2020; Cai et al. 2021) define safety by the satisfaction of temporal logical formulae of the learnt policy, but do not provide safety *while* training such a policy. Many existing approaches are concerned with guarantees on the safety of the learned policy, often under the assumption that a backup policy is available (Coraluppi and Marcus 1999; Perkins and Barto 2002; Geibel and Wysotzki 2005; Mannucci et al. 2017; Chow et al. 2018b; Mao et al. 2019). These methods are applicable to systems if they can be trained on accurate simulations, but for many other real-world systems we instead require safety *during* training.

There has also been research on maintaining safety during training. For instance, (Alshiekh et al. 2017; Jansen et al. 2019; Giacobbe et al. 2021) leverage the concept of a *shield* that stops the agent from choosing any unsafe actions. The shield assumes the agent observes the entire MDP (and any opponents) to construct a safety (game) model, which is unavailable for many partially-known MDP tasks. The approach by Garcia and Fernandez (2012) assumes a predefined safe baseline policy that is most likely sub-optimal, and attempts to slowly improve it with a slightly noisy action-selection policy, while defaulting to the baseline policy whenever a measure of safety is exceeded. However, this measure of safety assumes that nearby states have similar safety levels, which may not be the case. Another common approach is to use expert demonstrations to learn how to behave safely (Abbeel, Coates, and Ng 2010), or even to default to an expert when the risk is too high (Torrey and Taylor 2012). Obviously, such approaches rely heavily on the expert. Other approaches (Wen

*The work in this paper was done at the University of Oxford.

and Topcu 2018; Cheng et al. 2019; Turchetta, Berkenkamp, and Krause 2016) are either computationally expensive or require strong assumptions on agent-environment interactions. Crucially, maintaining safety in RL by efficiently leveraging available data is an open problem (Taylor et al. 2021).

Contributions We tackle the problem of synthesising a policy via RL that optimises a discounted reward, while not violating a safety requirement *during* learning. This paper puts forward a *cautious RL formalism* that (1) assumes the agent has limited observability over states and (2) infers a Dirichlet-Categorical model of the MDP dynamics. We incorporate higher-order information from the Dirichlet distributions, in particular we compute approximations of the (co)variances of the risk terms. This allows the agent to reason about the contribution of epistemic uncertainty to the risk level, and therefore to make better informed decisions about how to stay safe during learning. We show convergence results for these approximations, and propose a novel method to derive an approximate bound on the confidence that the risk is below a certain level. The new method adds a functionality to the agent that prevents it from taking critically risky actions, and instead leads the agent to take safer actions whenever possible, but otherwise leaves the agent to explore. The proposed method is versatile given that it can be added on to any general RL training scheme, in order to maintain safety during learning. Instructions on how to execute all the case studies in this paper are provided on the GitHub page (https://github.com/keeplearning-robot/riskawarerl).

2 Background

2.1 Problem Setup

Definition 2.1 *A finite MDP with rewards (Sutton, Bach, and Barto 2018) is a tuple $M = \langle S, A, s_0, P, R\rangle$ where $S = \{s^1, s^2, s^3, ..., s^N\}$ is a finite set of states, A is a finite set of actions, without loss of generality s_0 is an initial state, $P(s'|s, a)$ is the probability of transitioning from state s to state s' after taking action a, and $R(s, a)$ is a real-valued random variable which represents the reward obtained after taking action a in state s. A realisation of this random variable (namely a sample, obtained for instance during exploration) will be denoted by $r(s, a)$.*

An agent is placed at $s_0 \in S$ at time step $t = 0$. At every time step $t \in \mathbb{N}_0$, the agent selects an action $a_t \in A$, and the environment responds by moving the agent to some new state s_{t+1} according to the transition probability distribution, i.e., $s_{t+1} \sim P(\cdot|s_t, a_t)$. The environment also assigns the agent a reward $r(s_t, a_t)$. The objective of the agent is to learn how to maximise the long term reward. In the following we explain these notions more formally.

Definition 2.2 *A policy π assigns a distribution over A at each state: $\pi(a|s)$ is the probability of selecting action a in state s. Given a policy π, we can then define a state-value function*

$$v_\pi(s) = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s \right],$$

where $\mathbb{E}^\pi[\cdot]$ is the expected value given that actions are selected from π, and $0 < \gamma \leq 1$ is a discount factor.

Specifically, this means that the sequence $s_0, a_0, s_1, a_1, ...$ is such that $a_n \sim \pi(\cdot|s_n)$ and $s_{n+1} \sim P(\cdot|s_n, a_n)$. The discount factor γ is a pre-determined hyper-parameter that causes immediate rewards to be worth more than rewards in the future, as well as ensuring that this sum is well-defined, provided the standard assumption of bounded rewards. The agent's goal is to learn an optimal policy, namely one that maximises the expected discounted return. This is actually equivalent to finding a policy that maximises the state-value function $v_\pi(s)$ at every state (Sutton, Bach, and Barto 2018).

Definition 2.3 *A policy π is optimal if, at every state s, $v_\pi(s) = v_*(s) = \max_{\pi'} v_{\pi'}(s)$.*

Definition 2.4 *Given a policy π, we can define a state-action-value function $v_\pi(s, a) = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \, s_0 = s, a_0 = a\right]$, similarly to the state-value function. This allows us to reinterpret the state-value function as $v_\pi(s) = \sum_a v_\pi(s, a)\pi(a|s)$, and thus we can see that an optimal deterministic policy π must assign zero probability to any action a that doesn't maximise the state-action value function.*

2.2 Dirichlet-Categorical Model

We consider a model for an MDP with unknown transition probabilities (Ghavamzadeh et al. 2015). The transition probabilities for a given state-action pair are assumed to be described by a categorical distribution over the next state. We maintain a Dirichlet distribution over the possible values of those transition probabilities: since the Dirichlet distribution is conjugate, we can employ Bayesian inference to update the Dirichlet distribution, as new observations are made while the agent explores the environment.

Formally, for each state-action pair (s^i, a), we have a Dirichlet distribution $p_a^{i1}, p_a^{i2}, ..., p_a^{iN} \sim Dir(\alpha_a^{i1}, \alpha_a^{i2}, ...\alpha_a^{iN})$, where $\mathbf{p_a^i} := (p_a^{i1}, p_a^{i2}, ..., p_a^{iN})$, and the random variable p_a^{ij} represents the agent's belief about the transition probability $P(s^j|s^i, a)$. At the start of learning, the agent will be assigned a prior Dirichlet distribution for each state-action pair, according to its initial belief about the transition probabilities. At every time step, as the agent moves from some state s^i to some state s^k by taking action a, it will generate a transition $s^i \xrightarrow{a} s^k$, which constitutes a new data point for the Bayesian inference. From Bayes' rule:

$$\begin{aligned} &Pr(\mathbf{p_a^i} = \mathbf{q_a^i} | s^i \xrightarrow{a} s^k) \\ &\propto Pr(s^i \xrightarrow{a} s^k | \mathbf{p_a^i} = \mathbf{q_a^i}) Pr(\mathbf{p_a^i} = \mathbf{q_a^i}) \\ &\quad = q_a^{ik} \prod_j (q_a^{ij})^{\alpha_a^{ij}-1} = [\prod_{j \neq k} (q_a^{ij})^{\alpha_a^{ij}-1}](q_a^{ik})^{(\alpha_a^{ik}+1)-1}, \end{aligned}$$

where $\{q_a^{ij}\}_{j=1}^N$ belong to the standard $N-1$ simplex. This immediately yields

$$Pr(\mathbf{p_a^i} = \mathbf{q_a^i} | s^i \xrightarrow{a} s^k) = Dir(\alpha_a^{i1}, \alpha_a^{i2}, ..., \alpha_a^{ik} + 1, ..., \alpha_a^{iN}).$$

Thus, the posterior distribution is also a Dirichlet distribution. This update is repeated at each time step: the relevant

information to the agent's posterior belief about the transition probabilities is the starting prior $Dir(\alpha_a^{i1}, \alpha_a^{i2}, ...\alpha_a^{iN})$ and the transition counts, keeping track of the number of times that $s^i \xrightarrow{a} s^j$ has occurred. The agent's posterior is then $(p_a^{i1}, p_a^{i2}, ..., p_a^{iN}) \sim Dir(\alpha_a^{i1}, \alpha_a^{i2}, ...\alpha_a^{iN})$: from this distribution, we can distill the expected value $\bar{p}_a^{ij}$ of each random variable p_a^{ij}, as well as the covariance of any two p_a^{ij} and p_a^{ik} (therefore also the variance of a single p_a^{ij}):

$$\bar{p}_a^{ij} = \mathbb{E}[p_a^{ij}] = \frac{\alpha_a^{ij}}{\alpha_a^{i0}},\ Cov[p_a^{ij}, p_a^{ik}] = \frac{\alpha_a^{ij}(\delta^{jk}\alpha_a^{i0} - \alpha_a^{ik})}{(\alpha_a^{i0})^2(\alpha_a^{i0}+1)},$$

where $\alpha_a^{i0} = \sum_{k=1}^{N} \alpha_a^{ik}$, and δ^{jk} is the Kronecker delta.

3 Risk-aware Bayesian RL for Cautious Exploration

In this section we propose a new approach to Safe RL, which will specifically address the problem of how to learn an optimal policy in an MDP with rewards while avoiding certain states classified as unsafe during training. The agent is assumed to know which states of the MDP are safe and which are unsafe, but instead of assuming that the agent has this information globally, namely across all states of the MDP, we postulate that the agent observes states within an area around itself. This closely resembles real-world situations, where systems may have sensors that allow them to detect close-by dangerous areas, but not necessarily know about danger zones that are far away from them. In particular, we assume that there is an observation "boundary" O, such that the agent can observe all states that are reachable from the current state within O steps and distinguish which of those states are safe or unsafe. The rest of this section is structured as follows:

In Section 3.1, we define the risk $\rho^m(s,a)$ over m steps of taking an action a at the current state s. We then introduce a random variable $\varrho^m(s,a)$ representing the agent's belief about the risk; In Section 3.2, we leverage a method from Casella and Berger (2021) to approximate the expected value and variance of the random variable $\varrho^m(s,a)$. We provide convergence results on the approximations of the expectation and variance of $\varrho^m(s,a)$; In Section 3.3, we show how the Cantelli Inequality (Cantelli 1929) allows us to estimate a confidence bound on the risk $\rho^m(s,a)$; In Section 3.4, we prescribe a methodology for incorporating the expectation and variance of the risk into the local action selection during the training of the RL agent.

3.1 Definition and Characterisation of the Risk

Given the observation boundary O, we reason about the risk incurred over the next m steps after taking a particular action a in the current state s, for any $m \leq O$. However, note that there is a dependence between the agent's estimate of such a risk and the use of that estimate to inform its action selection policy. In order to solve this dilemma we fix a policy over the m-step horizon, and calculate the corresponding risk, given that policy. Similar to temporal-difference learning schemes, this is done by assuming best-case action selection, namely, the m-step risk $\rho^m(s,a)$ at state s after taking action a is defined assuming that after selecting action a, the agent will select subsequent actions to minimize the expected risk. Assuming that the agent is at state s, we define the agent's approximation of the m-step risk $\bar{\varrho}^{\,m}(s,a)$ by back-propagating the risk given the "expected safest policy" over m steps, as follows:

$$\bar{\varrho}^{\,n+1}(s^k,a) = \begin{cases} 1 & s^k \text{ observed and unsafe} \\ \sum_{j=1}^{N} \bar{p}_a^{kj}\, \bar{\varrho}^{\,n}(s^j) & \text{otherwise;} \end{cases} \tag{1}$$

$$\bar{\varrho}^{\,n}(s^k) := \begin{cases} 1 & s^k \text{ observed and unsafe} \\ \min_{a\in A} \bar{\varrho}^{\,n}(s^k,a) & \text{otherwise;} \end{cases} \tag{2}$$

$$\bar{\varrho}^{\,0}(s^k) := \mathbb{1}(s^k \text{ is observed and unsafe}). \tag{3}$$

We terminate this iterative process at $n+1 = m$ and once we have calculated $\bar{\varrho}^{\,m}(s,a)$, for actions $a \in A$. Note that, despite the use of progressing indices n, this is an iterative back-propagation that leverages the expected values of agent's belief about the transition probabilities, i.e., $\bar{p}_a^{kj}$. Thus, $\bar{\varrho}^{\,m}(s,a)$ is the agent's approximation of the expectation of the probability of entering an unsafe state within m steps by selecting action a at state s, and thereafter by selecting actions that it currently believes will minimize the probability of entering unsafe states over the given time horizon.

Remark 3.1 *We note that, in practice, an autonomous agent can determine, with some certainty, whether a subset of its observation are is safe to visit or not. Consider a mobile robot that moves in an office environment and can deem certain states as obstacles-to-avoid based on the received signals from onboard sensors. It is straightforward to extend the indicator function in* (3) *to a probability distribution, to reflect agent uncertainty over such signals.*

The term $\bar{p}_a^{kj} = \mathbb{E}[p_a^{kj}]$ is used as a point estimate of the true transition probability $t_a^{kj} = P(s^j|s^k,a)$. The value of $\bar{\varrho}^{\,m}(s,a)$ only relies on states which the agent believes are reachable from s within m steps. In particular so long as the horizon m is less than the observation boundary O, the agent is able to observe all states which are relevant to the calculation of $\bar{\varrho}^{\,m}(s,a)$, so specifically, $\mathbb{1}(s^j \text{ is unsafe}) = \mathbb{1}(s^j \text{ is observed and unsafe})$ for all relevant states s^j. Please refer to the extended paper (Mitta et al. 2023) for details.

3.2 Approximation of Expected Value and Covariance of the Risk

In the previous section, we presented the underlying mechanism for calculating an m-step *expected* risk. However, relying only on this expected value disregards the agent's confidence placed over this expectation: as a shortcoming of this, the agent might be willing to take actions that have lower expected risk, but which come with lower confidence as well. Evidently this behavior can be unsafe, and we would prefer the agent to employ its confidence in the decision-making process. In the following, we formalize the underpinnings of

how to incorporate a confidence approximation into the agent action selection policy.

Let $\mathbf{x}$ denote the vector of variables x_a^{ij} where i, j range from 1 to N and a ranges over A, i.e., $\mathbf{x} = \left((x_a^{ij})_{i,j=1,\ldots,N \text{ and } \forall a \in A}\right)$. We assume that these indices are ordered lexicographically by (i, a, j). This is because i and a will be used to signify a state-action pair (s^i, a), and j will be used to signify a potential next state s^j. Introduce a set of functions $g_k^n[\mathbf{x}]$ (we shall see they take the shape of polynomials), defined as follows for each state s^k:

$$g^{n+1}(s^k, a)[\mathbf{x}] := \begin{cases} 1 & \text{if } s^k \text{ is observed and unsafe} \\ \sum_{j=1}^{N} x_a^{kj} g^n(s^j)[\mathbf{x}] & \text{otherwise;} \end{cases}$$

$$g^n(s^k)[\mathbf{x}] := \begin{cases} 1 & \text{if } s^k \text{ is observed and unsafe} \\ g^n\left(s^k, \arg\min_a \bar{\varrho}_k^{\,n}(a)\right)[\mathbf{x}] & \text{otherwise;} \end{cases}$$

$$g^0(s^k)[\mathbf{x}] := \mathbb{1}(s^k \text{ is observed and unsafe}).$$

Then we can write the risk (of selecting action a in state s, over m steps) defined above as $\rho^m(s, a) = g^m(s, a)[\mathbf{t}]$, where $\mathbf{t} = \left((t_a^{ij})_{i,j=1,\ldots,N \text{ and } \forall a \in A}\right)$ is a vector of all "true" transition probabilities, namely $t_a^{ij} = P(s^j | s^i, a)$. We can similarly write the agent's approximation of the expected risk, as described in Section 3.1, as $\bar{\varrho}^{\,m}(s, a) = g^m(s, a)[\bar{\mathbf{p}}]$, where similarly $\bar{\mathbf{p}} = \left((\bar{p}_a^{ij})_{i,j=1,\ldots,N \text{ and } a \in A}\right)$, and $\bar{p}_a^{ij}$ is the expected value of each random variable p_a^{ij}. We refer to the actions specified by the $\arg\min$ operators as the *agent's expected safest action* in each state over the next m steps.

Now, crucially, we can also define a new random variable $\varrho^m(s, a) = g^m(s, a)[\mathbf{p}]$, where $\mathbf{p} = \left((p_a^{ij})_{i,j=1,\ldots,N \text{ and } \forall a \in A}\right)$. Since the p_a^{ij}s are random variables representing the agent's beliefs about the true transition probabilities t_a^{ij}, we in fact have that this random variable $\varrho^m(s, a)$ represents the agent's beliefs about the true risk $\rho^m(s, a)$. In the following, we show that $\bar{\varrho}^m(s, a)$ can be viewed as an approximation of $\mathbb{E}[\varrho^m(s, a)]$, and we provide and justify an approximation of $Var[\varrho^m(s, a)]$ that is directly correlated to agent's confidence on $\mathbb{E}[\varrho^m(s, a)]$. These approximations can be used by the agent to reason more accurately about the true risk of selecting an action a in a state s, over m steps, i.e., $r^m(s, a)$.

In order to construct approximations of the expectation and the variance of $R^m(s, a)$, we make use of the first-order Taylor expansion of $g^m(s, a)[\mathbf{x}]$ around $\mathbf{x} = \bar{\mathbf{p}}$, following a method by Casella and Berger (2021). The first-order Taylor expansion is

$$g^m(s, a)[\mathbf{x}] = g^m(s, a)[\bar{\mathbf{p}}] + \textstyle\sum_{i,j=1}^{N} \sum_{b \in A} \frac{\partial g^m(s,a)}{\partial x_b^{ij}} (x_b^{ij} - \bar{p}_b^{ij}),$$

where the partial derivatives are also evaluated at $\bar{\mathbf{p}}$ and we have disregarded the remainder term. Reasoning over the random variables $\mathbf{p}$ for $\mathbf{x}$:

$$g^m(s, a)[\mathbf{p}] \approx g^m(s, a)[\bar{\mathbf{p}}] + \textstyle\sum_{i,j=1}^{N} \sum_{b \in A} \frac{\partial g^m(s,a)}{\partial x_b^{ij}} (p_b^{ij} - \bar{p}_b^{ij}). \quad (4)$$

We can then take the expectation of both sides, obtaining

$$\begin{aligned} &\mathbb{E}[g^m(s, a)[\mathbf{p}]] && (5) \\ &\approx \mathbb{E}[g^m(s, a)[\bar{\mathbf{p}}]] + \mathbb{E}[\sum_{i,j=1}^{N} \sum_{b \in A} \frac{\partial g^m(s, a)}{\partial x_b^{ij}} (p_b^{ij} - \bar{p}_b^{ij})] \\ &= g^m(s, a)[\bar{\mathbf{p}}] + \sum_{i,j=1}^{N} \sum_{b \in A} \frac{\partial g^m(s, a)}{\partial x_b^{ij}} \mathbb{E}[(p_b^{ij} - \bar{p}_b^{ij})] \\ &= g^m(s, a)[\bar{\mathbf{p}}], && (6) \end{aligned}$$

where the above steps follow since the only random term in the right-hand side is p_b^{ij}, for which $\mathbb{E}(p_b^{ij}) = \bar{p}_b^{ij}$. Also, recall that $g^m(s, a)[\mathbf{p}] = \varrho^m(s, a)$ and $g^m(s, a)[\bar{\mathbf{p}}] = \bar{\varrho}^{\,m}(s, a)$. Thus, we have $\bar{\varrho}^m(s, a)$ as an approximation of the expectation of $\varrho^m(s, a)$. For the approximation of the variance of the agent's believed risk, which is again a random variable, we can write:

$$\begin{aligned} &\mathit{Var}(g^m(s, a)[\mathbf{p}]) && (7) \\ &\approx \mathbb{E}[(g^m(s, a)[\mathbf{p}] - g^m(s, a)[\bar{\mathbf{p}}])^2] \\ &\approx \mathbb{E}\left[\left(\sum_{i,j=1}^{N} \sum_{b \in A} \frac{\partial g^m(s, a)}{\partial x_b^{ij}} (p_b^{ij} - \bar{p}_b^{ij})\right)^2\right] \quad \text{(from (4))} \\ &= \sum_{i,j,s,t=1}^{N} \sum_{b_1, b_2 \in A} \frac{\partial g^m(s, a)}{\partial x_{b_1}^{ij}} \frac{\partial g^m(s, a)}{\partial x_{b_2}^{st}} \mathit{Cov}(p_{b_1}^{ij}, p_{b_2}^{st}) \\ &= \sum_{i=1}^{N} \sum_{b \in A} \sum_{j,t=1}^{N} \frac{\partial g^m(s, a)}{\partial x_b^{ij}} \frac{\partial g^m(s, a)}{\partial x_b^{it}} \mathit{Cov}(p_b^{ij}, p_b^{it}) && (8) \\ &:= \bar{V}^m(s, a), && (9) \end{aligned}$$

where $\bar{V}^m(s, a)$ is the approximation for the variance of $\varrho^m(s, a)$, i.e., $\bar{V}^m(s, a) \approx \mathit{Var}(\varrho^m(s, a))$, and the last line follows from the fact that the covariance between two transition probability beliefs $p_{b_1}^{ij}$ and $p_{b_2}^{st}$ is always 0, unless they correspond to the same starting state-action pair (s^i, b). In other words, $\mathit{Cov}(p_{b_1}^{ij}, p_{b_2}^{st}) = 0$ unless $i = j$ and $b_1 = b_2$. Next, we show consistency of the estimate in the limit, and the proof is available in the extended paper (Mitta et al. 2023).

Theorem 3.1 *Under standard Q-learning convergence assumptions (Watkins 1989), namely that reachable state-action pairs are visited infinitely often, the estimate of the mean of the believed risk distribution $\bar{\varrho}^m(s, a)$ converges to the true risk $\rho^m(s, a)$, and it does so with the variance of the believed risk distribution $Var\left(g^m(s, a)[\mathbf{p}]\right)$ approaching the estimate of that variance $\bar{V}^m(s, a)$. Specifically,*

$$\frac{\bar{\varrho}^m(s, a) - \rho^m(s, a)}{\sqrt{\bar{V}^m(s, a)}} \to \mathcal{N}(0, 1) \textit{ in distribution.}$$

3.3 Estimating a Confidence on the Approximation of the Risk

So far we have shown that when the agent is in the state s, for each possible action a, approximations of the expectation

and variance of its belief $\varrho^m(s,a)$ about the risk $\rho^m(s,a)$ can be formally obtained: we have denoted these two approximations by $\bar{\varrho}^m(s,a)$ and $\bar{V}^m(s,a)$, respectively. We now describe a method for combining these approximations to obtain a bound on the level of confidence that the risk $\rho^m(s,a)$ is below a certain threshold.

We appeal to the Cantelli Inequality, which is a one-sided Chebychev bound (Cantelli 1929). Having computed $\bar{\varrho}^m(s,a)$ and $\bar{V}^m(s,a)$, for a particular confidence value $0 < C < 1$ we can define $\Phi := \bar{\varrho}^m(s,a) + \sqrt{\frac{\bar{V}^m(s,a)C}{1-C}}$. From the Cantelli Inequality we then have

$$\Pr(\varrho^m(s,a) \leq \Phi) \geq C.$$

Specifically, Φ is the lowest risk level such that, according to its approximations, the agent can be at least $100 \times C$ % confident that the true risk is below level Φ. The exploration mechanism can therefore leverage Φ to ensure that the required safety level is met. Please refer to the extended paper (Mitta et al. 2023) for more details.

3.4 RCRL: Risk-aware Bayesian RL for Cautious Exploration

In this section we propose an overall approach for safe RL, which leverages the expectation and variance of the defined risk measure to allow an agent to explore the environment safely, while attempting to learn an optimal policy. In order to select an optimal-yet-safe action at each state, we propose a *double-learner* architecture, referred to as *Risk-aware Cautious RL (RCRL)* and explained next.

The first learner is an optimistic agent whose objective is to maximize the expected cumulative return. The second learner is a pessimistic agent that maintains a Dirichlet-Categorical model of the transition probabilities of the MDP. In particular, this agent is initialized with a prior that encodes any information the agent might have about the transition probabilities. For each state-action pair (s^i, a) we have a Dirichlet distribution $p_a^{i1}, p_a^{i2}, ..., p_a^{iN} \sim Dir(\alpha_a^{i1}, \alpha_a^{i2}, ...\alpha_a^{iN})$. As the agent explores the environment, the Dirichlet distributions are updated using Bayesian inference.

For each action a available in the current state s, the pessimistic learner computes the approximations $\bar{\varrho}^m(s,a)$ and $\bar{V}^m(s,a)$ of its belief $\varrho^m(s,a)$ of the risk, over the next m steps, associated to taking action a in s. The "risk horizon" m is a hyper-parameter that, as discussed, should be set to be at most the observation boundary O. The pessimistic learner is initialized with two extra hyper-parameters Φ_{max} and $C(n)$: Φ_{max} represents the maximum level of risk that the agent should be prepared to take, whereas $C(n)$ is a decreasing function of the number of times n that the current state has been visited, which satisfies $C(0) < 1$ and $\lim_{n\to\infty} C(n) = 0$. From Section 3.3, the agent can then compute, for each action a, the value

$$\Phi = \bar{\varrho}^m(s,a) + \sqrt{\frac{\bar{V}^m(s,a)C(n)}{1-C(n)}}, \tag{10}$$

which can thus define a set of safe actions: these are all the actions that the agent believes have risk less than Φ_{max}, with confidence at least $C(n)$, namely

$$A_{safe} = \{a \in A | \Phi \leq \Phi_{max}\}.$$

In case there are no actions a such that $\Phi \leq \Phi_{max}$, the agent instead allows

$$A_{safe} = \{a \in A | \bar{\varrho}^m(s,a) = \min_{a'} \bar{\varrho}^m(s,a')\}. \tag{11}$$

Finally, the agent selects an action a^*_{safe} from the set of safe actions according to the Q-values of those actions, e.g., using softmax action selection (Sutton, Bach, and Barto 2018) with some *temperature* $\mathcal{T} > 0$:

$$Pr(a^*_{safe} = a) = \frac{e^{Q(s,a)/\mathcal{T}}}{\sum_{a \in A_{safe}} e^{Q(s,a)/\mathcal{T}}}. \tag{12}$$

The pseudo-code for the full algorithm is available in the extended paper (Mitta et al. 2023).

Remark 3.2 *RCRL focuses on ensuring safety in exploration, prioritizing theoretical guarantees over traditional RL's goal of maximizing reward. While pushing exploration boundaries, RCRL's primary objective is to maintain agent safety rather than maximizing expected rewards.*

In summary, we effectively have two agents learning to accomplish two tasks. The first agent performs Q-learning to learn an optimal policy for the reward. The second agent determines the best approximation of the expected value and variance of each action, enabling it to prevent the first agent from selecting actions that it cannot guarantee to be safe enough (with at least a given confidence). When instead the pessimistic agent cannot guarantee that any action is safe enough, it forces the optimistic learner to go into "safety mode", i.e., to forcibly select the actions that minimize the expected value of the risk, as per (11). From an empirical perspective, implementing this concept of a "safety mode" allows for continued progress, and pairs well with the definition of risk: namely, when the agent deems that a state is too risky, it will go into this "safety mode" until it is back in a state with sufficiently safe actions.

Finally, note that $C(n)$ represents the level of confidence that the agent requires in an action being safe enough for it to consider taking that action. When the agent starts exploring and $C(n)$ is at its highest, the agent only explores actions that it is very confident in. However, it may need to take actions that it is less confident in order to find an optimal policy. Thus, as it continues exploring, $C(n)$ is reduced, allowing the agent to select actions upon which it is not as confident. However, in the limit, when $C(n) \to 0$, we have that $\Phi = \bar{\varrho}^m(s,a)$, which means that the agent never takes an action if its approximation of the expected risk $\bar{\varrho}^m(s,a)$ is more than the maximum allowable risk Φ_{max}.

4 Experiments

Details on the experiments are presented in the extended paper Mitta et al. (2023).

BridgeCross - We first evaluated the performance of RCRL on a *Slippery Bridge Crossing* example. The states of the

Experiment	$\|S\|$	$\|A\|$	Safety Setup	Successes	Fails	# Ep.
BridgeCross	400	5	Prior 1, $\Phi_{max} = 0.33$	404.3	54.2	500
	400	5	Prior 1, $\Phi_{max} = 0.01$	506.0	417.9	1500
	400	5	Prior 2, $\Phi_{max} = 0.33$	424.3	32.1	500
	400	5	Prior 2, $\Phi_{max} = 0.01$	384.6	0.5	500
	400	5	Prior 3, $\Phi_{max} = 0.01$	407.4	14.4	500
	400	5	Prior 3, $\Phi_{max} = 0.003$	421.3	1.1	500
	400	5	QL with Penalty	414.6	990.5	1500
	400	9	Prior 1, $\Phi_{max} = 0.33$	299.1	173.4	500
	400	9	Prior 1, $\Phi_{max} = 0.01$	348.9	523.2	1500
	400	9	Prior 2, $\Phi_{max} = 0.33$	444.7	38.9	500
	400	9	Prior 2, $\Phi_{max} = 0.01$	17.6	14.5	500
	400	9	Prior 3, $\Phi_{max} = 0.01$	391.7	15.4	500
	400	9	Prior 3, $\Phi_{max} = 0.003$	430.0	2.2	500
	400	9	QL with Penalty	367.9	1119.2	1500
Pacman	4000	5	Risk Horizon $m = 2$	234	77	311
	4000	5	Risk Horizon $m = 3$	207	68	275
	4000	5	QL with Penalty	0	1500	1500

Table 1: Average number of successes and failures. BridgeCross: different priors and acceptable risks Φ_{max}. Pacman: varying risk horizon m.

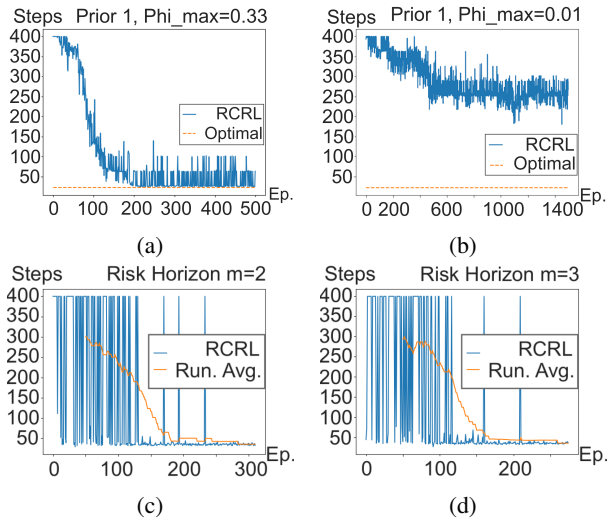

Figure 1: The number of steps it takes the agent to get to the rewarding state. Averaged over 10 experiments. Results for RL and for RCRL across different priors and horizons.

MDP consist of a 20×20-grid. The agent is initialized at q_0 in the bottom-left corner (green). The agent's task is to get to the goal region without ever entering an unsafe state. In particular, upon reaching a goal state, the agent is given a reward of 1 and the learning episode is terminated; at every other state it receives a reward of 0, and upon reaching an unsafe state the learning episode terminates with reward 0.

We consider two cases regarding the action space. Case I: We assume that at each time step the agent might move into one of the 4 neighbouring states, or stay in its current position; thus, the agent has access to 5 actions at each state, $A = \{right, up, left, down, stay\}$. Case II: We consider a larger action space that includes the diagonal actions as well, i.e., $|\mathcal{A}| = 9$. In both cases, if the agent selects action $a \in A$, then it has a 96% chance of moving in direction a, and a 4% chance of "slipping", namely moving into another random direction. If any movement would ever take the agent outside of the map, then the agent will just remain in place. The agent is assumed to have an observation boundary $O = 2$ steps. Note that due to the slipperiness of the movement and the narrow passage to reach the goal state, minimizing the risk is not aligned with maximizing the expected reward.

We tested RCRL with 5 different combinations of a prior and a maximum acceptable risk Φ_{max}. The following additional hyper-parameters of the algorithm were kept constant: the maximum number of steps per episode $max\_steps = 400$, the maximum number of episodes $max\_episodes = 500$ (although this was increased to 1500 in two cases when the agent did not converge to near-optimal policy within the first 500, cf. Table 1); the learning rate $\mu = 0.85$; the discount factor $\gamma = 0.9$; and the risk horizon $m = 2$. Recall that a prior consists of a Dirichlet distribution $p_a^{i1}, ..., p_a^{iN} \sim Dir(\alpha_a^{i1}, ..., \alpha_a^{iN})$ for every state-action pair (s^i, a). We considered three priors:

- Prior 1 – completely uninformative: in this case we assigned a value of 1 to every α. This yields a distribution that is uniform over its support.
- Prior 2 – weakly informative: we assigned a value of 12 to the α corresponding to moving in the correct direction, and a value of 1 to all other α's. This gives a distribution in between Prior 1 and 3 in degree of bias and concentration.
- Prior 3 – highly informative: we assigned a value of 96 to the α corresponding to moving in the correct direction, and a value of 1 to all other α's. This gives a distribution that is highly concentrated, and for which the mean values of the transition probability random variables are the true transition probabilities of the MDP, and hence unbiased.

We tested the algorithm with all three priors and a maximum acceptable risk of $\Phi_{max} = 0.01$ and repeating each experiment 10 times to take averages. We first discuss the results for Case I. On average, the agent with the highly informative prior (Prior 3) entered unsafe states 14.4 times (on average), and always converged to near-optimality within about 200 steps, successfully crossing the bridge 407.4 times. For the other 78.2 episodes, the agent reached the episode limit before crossing the bridge or entering an unsafe state. The agent with Prior 2 interestingly only entered unsafe states an average of 0.5 times per experiment, and converged to a near-optimal policy within about 300 episodes, successfully crossing the bridge 384.6 times. On the other hand, the agent with Prior 1 only crossed the bridge less than 30 times. We therefore increased the total number of episodes to 1500 and tried again, yet still over half the time it did not converge to a near-optimal policy (Figure 1b).

A similar pattern is observed for Case II, where the number of failed episodes tends to decrease as the prior becomes more informative. Interestingly, the agent with Prior 2 also exhibits a relatively low number of successful episodes. A potential explanation for this could be the low acceptable risk of $\Phi_{\text{max}} = 0.01$, as discussed in Remark 3.2. We then tested

Prior 1/Case I with a more lenient maximum acceptable risk of $\Phi_{max} = 0.33$, and found that the agent this time managed to converge to near-optimality within around 200 episodes, entering unsafe states 54.2 times and successfully crossing the bridge 404.3 times. We also tested Prior 3/Case I with a stricter $\Phi_{max} = 0.0033$ and found out that it entered unsafe states only 1.1 times and succeeded 421.3 times, converging to near-optimality within 150 episodes. Similar observations were made for Case II. For instance, in Prior 2 with $\Phi_{max} = 0.33$, the agent managed to increase the number of successful episodes from 17.6 to 444.7 while slightly increasing, as expected, the number of failures from 14.5 to 38.9. A more thorough analysis of these results is presented in the extended paper (Mitta et al. 2023). Finally, we tested Q-learning. Q-learning had almost no successful crossings of the bridge in the first 500 episodes, so we ran it for 1500 episodes and found that it only converged to a near-optimal policy about half the time, on average entering unsafe states 990.5 times and successfully crossing the bridge 414.6 times.

Discussion - Table 1 summarizes the number of successes and failures for each agent. The first result of note is how poorly Prior 1 performs with $\Phi_{max} = 0.01$ for both Case I and Case II. It mostly fails to converge to near-optimal behaviour even with 1500 steps as presented in Figure 1b, in fact seeming to converge slower than Q-learning. This occurs because the maximum allowable risk is set too low for the given prior. In particular, there are two main issues with this. The first issue is a type of degenerate behaviour specific to our algorithm and to the completely uninformative prior with overly strict Φ_{max}: given that the agent starts with no information on the transition probabilities, it is unable to tell which actions are safe and which are unsafe. In particular, with Φ_{max} at 1%, the first time the agent arrives at any state s from which it can observe some unsafe state, it immediately goes into safety mode as it judges that the risk of every action is above 1%. Since it has no information on which action is safest, it randomly selects an action.

If that randomly-selected action does not take the agent closer to a risky state, then after updating the agent's beliefs about the transition probabilities for that action, it will believe that action is the safest one from that state. Thus every time it encounters that state again, it will *always* select that action, never attempting any other actions. The state $(13, 1)$ has been visited significantly more often than any other state. This has occurred because the first time the agent encountered that state, it chose action $stay$, and as above, from then on always chose $stay$ in state $(13, 1)$. This would cause the agent to remain in $(13, 1)$ until it slipped off of that state. For further discussions refer to the extended paper (Mitta et al. 2023).

Pacman - We evaluated the performance of RCRL on a *Pacman* example. The agent (Pacman) must get to food tokens without getting caught by the ghost. Note that because both the agent and the ghost move through the maze, the Pacman MDP has about 10 times more states than the BridgeCross, and up to 5 times more possible next states at any given state. Upon picking up the second piece of food, the agent is given a reward of 1 and the learning episode stops. Every other state incurs a reward of 0 and if the ghost catches Pacman, the learning episode stops with reward 0. The agent has access to five joystick actions at each state, $A = \{right, up, left, down, no\_act\}$ and will move in the direction selected, or if that direction moves into a wall, then it will stay still. The ghost will with 90% probability move in the direction that takes it closest to the agent's next location, and with 10% probability will move in a random direction. For this setup, we assumed an observation boundary $O = 3$ and compared two values of the risk horizon, $m = 2, 3$. We therefore kept constant the other parameters and hyper-parameters: the learning rate $\mu = 0.85$; the discount factor $\gamma = 0.9$; the maximum number of steps per episode $max\_steps = 400$; the maximum acceptable risk $\Phi_{max} = 0.33$; the prior, which we set to be a completely uninformative prior as in the BridgeCross example; the maximum number of episodes, which we set as 1500 or the number of episodes before the total rate of successful episodes exceeded 75%. As in Table 1, the agent with a risk horizon of $m = 2$ steps exceeded a success rate of 75% after 311 episodes, having failed 77 times. The agent with the larger risk horizon of $m = 3$ only took 275 steps to exceed that success rate, and only failed 68 times. Figures 1c–1d display the number of steps taken by the agent to win (or 400 if they lose) for each agent, as well as the running average number of steps over the previous 50 episodes.

Discussion - The improvement in performance from $m = 2$ to 3 is likely due to the increased foresight of the agent leading it to move away from excessively risky scenarios further in advance, potentially avoiding entering a state from which entering a dangerous state is unavoidable. However, it may also be simply due to the fact that increasing the risk horizon leads to an overall increase in risk estimates, which will naturally cause more actions to be considered too risky and may reduce the number of failures. In other words, we may have been in a situation where decreasing the maximum acceptable risk Φ_{max} would have led to similar improvements, and the increase in risk horizon was behaving functionally more like a decrease in Φ_{max}. Both risk-aware agents compare very favourably against the Q-Learning agent, which did not succeed once across 1500 episodes.

5 Conclusions

We proposed a new approach, Risk-aware Cautious Reinforcement Learning (RCRL), to address the problem of safe exploration in MDPs. A definition of the risk related to taking an action in a given state has made use of the agent's beliefs about the MDP transitions and the safest available actions in future states. We have approximated the expectation and variance of the defined risk and have derived a convergence result that justifies the use of those approximations. We have also shown how to derive an approximate bound on the confidence that the risk is below a certain level. All these ingredients comprise RCRL, a Safe RL architecture that couples risk estimation and safe action selection with RL. We tested RCRL and showed that it significantly outperform on Q-learning, both in terms of maintaining safety during exploration, as well as of the rate of convergence to an optimal policy. As this approach can be easily interleaved with other RL algorithms we expect similar improvements against other baselines.

References

Abbeel, P.; Coates, A.; and Ng, A. Y. 2010. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research*, 29(13): 1608–1639.

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2017. Safe Reinforcement Learning via Shielding. arXiv:1708.08611.

Brunke, L.; Greeff, M.; Hall, A. W.; Yuan, Z.; Zhou, S.; Panerati, J.; and Schoellig, A. P. 2021. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *CoRR*, abs/2108.06266.

Cai, M.; Hasanbeig, H.; Xiao, S.; Abate, A.; and Kan, Z. 2021. Modular deep reinforcement learning for continuous motion planning with temporal logic. *IEEE Robotics and Automation Letters*, 6(4): 7973–7980.

Cantelli, F. P. 1929. Sui confini della probabilita. In *Atti del Congresso Internazionale dei Matematici: Bologna del 3 al 10 de settembre di 1928*, 47–60.

Casella, G.; and Berger, R. L. 2021. *Statistical inference*. Brooks/Cole Cengage Learning.

Cheng, R.; Orosz, G.; Murray, R. M.; and Burdick, J. W. 2019. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *AAAI*, volume 33, 3387–3395.

Chow, Y.; Ghavamzadeh, M.; Janson, L.; and Pavone, M. 2018a. Risk-Constrained Reinforcement Learning with Percentile Risk Criteria. *JMLR*, 1–51.

Chow, Y.; Nachum, O.; Duenez-Guzman, E.; and Ghavamzadeh, M. 2018b. A Lyapunov-based Approach to Safe Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 8092–8101.

Coraluppi, S. P.; and Marcus, S. I. 1999. Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. *Automatica*, 35(2): 301–309.

Garcia, J.; and Fernandez, F. 2012. Safe Exploration of State and Action Spaces in Reinforcement Learning. *Journal of Artificial Intelligence Research*, 45: 515–564.

Garcia, J.; and Fernandez, F. 2015. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research 16*.

Geibel, P.; and Wysotzki, F. 2005. Risk-Sensitive Reinforcement Learning Applied to Control under Constraints. *Journal of Artificial Intelligence Research*, 24: 81–108.

Ghavamzadeh, M.; Mannor, S.; Pineau, J.; and Tamar, A. 2015. Bayesian Reinforcement Learning: A Survey. *Foundations and Trends in Machine Learning*, 8(5-6): 359–483. ArXiv: 1609.04436.

Giacobbe, M.; Hasanbeig, H.; Kroening, D.; and Wijk, H. 2021. Shielding Atari Games with Bounded Prescience. In *Autonomous Agents and MultiAgent Systems*.

Hasanbeig, H.; Abate, A.; and Kroening, D. 2019a. Certified Reinforcement Learning with Logic Guidance. *arXiv preprint arXiv:1902.00778*.

Hasanbeig, H.; Abate, A.; and Kroening, D. 2019b. Logically-Constrained Neural Fitted Q-Iteration. In *Autonomous Agents and MultiAgent Systems*, 2012–2014.

Hasanbeig, H.; Kroening, D.; and Abate, A. 2020. Deep reinforcement learning with temporal logics. In *Formal Modeling and Analysis of Timed Systems*, 1–22. Springer.

Jansen, N.; Könighofer, B.; Junges, S.; Serban, A. C.; and Bloem, R. 2019. Safe Reinforcement Learning via Probabilistic Shields. arXiv:1807.06096.

Li, X.; and Belta, C. 2019. Temporal Logic Guided Safe Reinforcement Learning Using Control Barrier Functions. arXiv:1903.09885.

Mannucci, T.; van Kampen, E.-J.; De Visser, C.; and Chu, Q. 2017. Safe exploration algorithms for reinforcement learning controllers. *IEEE Transactions on Neural Networks and Learning Systems*, 29(4): 1069–1081.

Mao, H.; Schwarzkopf, M.; He, H.; and Alizadeh, M. 2019. Towards safe online reinforcement learning in computer systems. In *Neural Information Processing Systems*.

Miryoosefi, S.; Brantley, K.; Daume III, H.; Dudik, M.; and Schapire, R. E. 2019. Reinforcement learning with convex constraints. *Advances in Neural Information Processing Systems*, 32.

Mitta, R.; Hasanbeig, H.; Wang, J.; Kroening, D.; Kantaros, Y.; and Abate, A. 2023. Safeguarded Progress in Reinforcement Learning: Safe Bayesian Exploration for Control Policy Synthesis. *arXiv preprint arXiv:2312.11314*.

Moldovan, T. M.; and Abbeel, P. 2012. Safe Exploration in Markov Decision Processes. *arXiv:1205.4810*.

Pecka, M.; and Svoboda, T. 2014. Safe exploration techniques for reinforcement learning–an overview. In *International Workshop on Modelling and Simulation for Autonomous Systems*, 357–375. Springer.

Perkins, T. J.; and Barto, A. G. 2002. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec): 803–832.

Puterman, M. L. 2014. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

Sutton, R. S.; Bach, F.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT Press Ltd.

Taylor, A. J.; Dorobantu, V. D.; Dean, S.; Recht, B.; Yue, Y.; and Ames, A. D. 2021. Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty. In *Conference on Decision and Control*, 6469–6476. IEEE.

Torrey, L.; and Taylor, M. E. 2012. Help an Agent Out: Student/Teacher Learning in Sequential Decision Tasks. In *Adaptive and Learning Agents workshop (at AAMAS-12)*.

Turchetta, M.; Berkenkamp, F.; and Krause, A. 2016. Safe exploration in finite Markov decision processes with Gaussian processes. In *Advances in Neural Information Processing Systems*, 4312–4320.

Watkins, C. J. C. H. 1989. *Learning from delayed rewards*. Ph.D. thesis.

Wen, M.; and Topcu, U. 2018. Constrained cross-entropy method for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, 7450–7460.