Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
KR meets Machine Learning and Explanation

SC^2: Safe Control via Shielding for CPCTL Specifications

Edwin Hamel-De le Court[1,†], **Gaspard Ohlmann**[2,†], **Francesco Belardinelli**[3]
[1]The University of Manchester, Manchester
[2]Independent Researcher, Mulhouse
[3]Imperial College, London
gaspard.ohlmann@outlook.com, edwin.hamel-delecourt@manchester.ac.uk,
francesco.belardinelli@imperial.ac.uk

Abstract

In real-world scenarios, reinforcement learning (RL) agents must not only maximize reward, but also behave safely, including during training. This has led to growing interest in Safe RL, where the objective is to learn an optimal policy among those satisfying relevant safety constraints. Most existing approaches focus on constraints expressed either as expected costs or as avoidance properties. However, safety in dynamical systems is often expressed using rich temporal logics such as Probabilistic Computation Tree Logic (PCTL). In this paper, we address the Safe RL problem under constraints expressed in CPCTL, a fragment of PCTL that generalizes avoidance constraints and enables the specification of nested behaviors. To this end, we leverage shielding, a technique that restricts the agent's actions during both training and deployment to enforce safety over an infinite horizon. We first introduce a general framework based on an augmentation method and provide its theoretical foundations. Building on this framework, we propose an algorithm that is provably safe at all times, including during training, while remaining optimal among all safe policies. Finally, we present an experimental evaluation demonstrating the effectiveness of our approach.

1 Introduction

Constrained Reinforcement Learning A central challenge in AI is the design of agents that can learn to act optimally in environments whose dynamics are initially unknown (Sutton and Barto 2018). Reinforcement Learning (RL), especially when combined with deep neural networks, has achieved remarkable empirical success in a wide range of domains, including strategic games (Mnih et al. 2015), robotics (Kober, Bagnell, and Peters 2013), and autonomous driving (Kendall et al. 2019).

Despite these advances, a major obstacle to deploying RL systems in real-world, safety-critical applications is the lack of guarantees regarding undesirable or unsafe behavior. In particular, standard RL algorithms freely explore the environment during training and may violate safety constraints arbitrarily often, while even the final learned policy may only be optimal in expectation and offer no guarantees on rare but catastrophic events. This has motivated a large body of work on *Safe* or *Constrained Reinforcement Learning*, where the objective is to maximize reward while satisfying explicit constraints on costs, risks, or safety specifications (Garcıa and Fernández 2015).

Constrained RL methods differ substantially depending on the nature of the constraint that is imposed. (Discounted) cumulative cost in expectation (CCE) (Achiam et al. 2019; Stooke, Achiam, and Abbeel 2020; Ray, Achiam, and Amodei 2019) fundamentally relies on cumulative cost formulations and do not directly capture temporal or logical safety requirements. Avoidance constraints, on the other hand (Alshiekh et al. 2017; le Court, Belardinelli, and Goodall 2025), can express infinite-horizon safety properties, but with limited expressivity. Multi-Objective Avoidance (MOA) constraints, where the agent is required to satisfy a conjunction of probabilistic avoidance constraints, allow for a gain in expressivity, but a limited number of approaches has addressed learning-based reward optimization under MOA (Yang et al. 2023; Goodall and Belardinelli 2024; Goodall and Belardinelli 2023) .

Main Contribution. In this paper, we go beyond the constraint classes discussed above by considering specifications expressed in Continuing Probabilistic CTL (CPCTL), a novel fragment of safe PCTL introduced in (Ohlmann, le Court, and Belardinelli 2025). CPCTL allows for arbitrary nesting of probabilistic and temporal operators, enabling the expression of rich safety specifications that cannot be captured by cumulative cost constraints, simple avoidance objectives, MOA objectives or even general PCTL flat formulas.

To the best of our knowledge, there currently exists no method that can optimize reward under general CPCTL specifications while providing safety guarantees throughout training and deployment. We address this gap by introducing SC^2, a shielding-based algorithm that enforces CPCTL constraints at runtime while learning a reward-optimal policy. Our approach guarantees safety at all times, including during exploration, and supports specifications that lie strictly beyond the expressiveness of existing shielding and constrained RL methods. To do so, as in (Court, Belardinelli, and Goodall 2025) for Avoidance constraints and (le Court, Ohlmann, and Belardinelli 2025) for (CCE) constraints, we consider an augmentation of the MDP. The shield does not remove actions or states based on the added parameters, but

[†]These authors contributed equally to this work.

it removes distributions themselves.

For reasons of space, the proofs of all results (as well as the code for experiments) is provided in the supplementary material.

1.1 Related Work

(Discounted) Cumulative Cost in Expectation (CCE). A (CCE) constraint is expressed as a bound on an expected cumulative cost, typically discounted over time. In this setting, the agent must satisfy a constraint of the form

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right] \leq C,$$

which intuitively expresses that the cumulative discounted cost has to remain below a given threshold C, while maximizing expected reward. This formulation naturally leads to the framework of Constrained Markov Decision Processes (CMDPs).

A large body of work addresses this problem using Lagrangian or primal–dual methods, where the constraint is incorporated into the objective through a multiplier that is updated during training (Achiam et al. 2019; Stooke, Achiam, and Abbeel 2020; Ray, Achiam, and Amodei 2019). These methods are typically model-free and scale well to high-dimensional problems, but their guarantees are asymptotic and do not necessarily hold throughout training.

More recently, several approaches have sought to strengthen the formal guarantees provided by expectation-based cost constraints. Examples include risk-aware state augmentation methods as well as shielding-inspired methods that restrict unsafe actions during training while still optimizing reward (Sootla et al. 2022; le Court, Ohlmann, and Belardinelli 2025). Nevertheless, these approaches still fundamentally rely on cumulative cost formulations and do not directly capture temporal or logical safety requirements.

Avoidance (A). A different and increasingly important class of constraints focuses on bounding the probability of reaching undesirable states. In its simplest form, an *avoidance* constraint requires that the probability of reaching a set of unsafe states labeled as *unsafe* does not exceed a threshold p, i.e.,

$$\mathbb{P}_\pi\left[\mathbf{F}(\mathit{unsafe})\right] \leq p,$$

or equivalently, that the agent avoids these states with sufficiently high probability (where $\mathbf{F}$ is the 'eventually' operator of temporal logic).

Avoidance constraints are particularly relevant in safety-critical systems, where even a single visit to a hazardous state may be unacceptable. Such constraints are closely related to chance constraints and probabilistic reachability in formal verification (Baier and Katoen 2008a). In the RL literature, they have been addressed using risk-sensitive objectives, reachability analysis, and abstraction-based techniques.

Several recent works combine RL with shielding or action-filtering mechanisms to enforce avoidance constraints at runtime. In these approaches, a safety mechanism restricts the agent to actions that do not excessively increase the probability of failure, while the underlying RL algorithm continues to optimize reward within the admissible action set (Jansen et al. 2020; le Court, Belardinelli, and Goodall 2025). Our approach can be seen as a generalization of this line of work to more expressive constraints.

Multi-Objective Avoidance (MOA). A natural generalization of avoidance constraints arises when multiple safety objectives must be satisfied simultaneously. In *Multi-Objective Avoidance* (MOA), the agent is required to satisfy a conjunction of probabilistic avoidance constraints, each corresponding to a different class of unsafe behavior. Formally, this takes the form

$$\bigwedge_{j=1}^{k} \mathbb{P}_{\geq p_j}\left[\mathbf{G}(\neg l_j)\right],$$

where each label l_j represents a distinct safety condition and $\mathbf{G}$ is the 'always' operator of temporal logic).

MOA constraints are significantly more expressive than single-objective avoidance, as they capture heterogeneous and potentially interacting safety requirements. They naturally arise in complex systems where different types of failures must be prevented simultaneously, such as collisions, deadlocks, and resource exhaustion. Despite their practical relevance, optimization under MOA constraints remains challenging. Existing approaches often reduce the problem to scalarized cost constraints or treat objectives sequentially, which may lead to overly conservative or suboptimal policies (Roijers et al. 2013). Although some recent shielding approaches support learning-based reward optimization techniques under MOA constraints (Yang et al. 2023; Goodall and Belardinelli 2024; Goodall and Belardinelli 2023), to the best of our knowledge, there is no current shielding approach that offers a strict guarantee that the MOA constraint is satisfied.

2 Preliminaries

In this section we provide preliminaries on Markov decision chains and processes (Sec. 2.1), and Probabilistic CTL (Sec. 2.2). We refer the interested reader to (Baier and Katoen 2008b) for an in-depth presentation.

2.1 Stochastic Models

Hereafter, for any measurable space E, we denote by $\mathcal{D}(E)$ the set of probability measures over E (Halmos 2013).

Markov Decision Processes. A (labeled) *Markov Decision Process* (Puterman 1994) is a tuple $\mathcal{M} = \langle S, A, P, s_0, AP, L, R\rangle$, where S is a set of *states*; A is a mapping that associates every state $s \in S$ to a nonempty finite set $A(s)$ of *actions*; $P : S \times A \to \mathcal{D}(S)$ is a *transition probability function* that maps every state-action pair (s, a) to a probability measure $P(s'|s, a)$ over S; $s_0 \in S$ is the *initial state*[1];

[1]This can be assumed wlog compared to a model with an *initial probability distribution*, as it is always possible to add a new initial state to a model of the latter kind, with an action from such

AP is a set of *atomic propositions*; $L : S \mapsto 2^{AP}$ is a *labeling function*; and $R : S \times \cup_{s\in S} A(s) \mapsto \mathbb{R}$ is the *reward function*.

Histories. A history in an MDP $\mathcal{M}$ is a finite word $\xi = (s_0 a_0 \dots a_{n-1} s_n)$ such that for each j, $s_j \in \text{succ}(a_{j-1})$ and $a_j \in A(s_j)$.

Policies. A *policy* (also called *controller*) π of $\mathcal{M}$ is a mapping that associates any history ρ of $\mathcal{M}$ to a probability measure over $A(\text{last}(\rho))$. It is memory-less if $\pi(\rho)$ only depends on $\text{last}(\rho)$, in which case we denote $\pi(\rho) = \pi(s)$, where $s = \text{last}(\rho)$. It is deterministic if for any finite path ρ, $\pi(\rho)$ is a Dirac measure.

2.2 Probabilistic and Temporal Specifications

In this section, we define the probabilistic temporal Logic PCTL (Ciesinski and Größer 2004), as well as its Continuing fragment CPCTL (Ohlmann, le Court, and Belardinelli 2025). These two languages allow for probabilistic specifications as well as their nesting.

Probabilistic Temporal Logic (PCTL) Let AP be a finite set of atoms and $q \in [0, 1]$. A formula of PCTL is generated by the nonterminal Φ in the following grammar:

$$\Phi ::= a \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \neg\Phi \mid \mathbb{P}_{\geq q}(\varphi), a \in AP \cup \{\bot, \top\},$$
$$\varphi ::= \mathbf{X}\,\Phi_1 \mid \Phi_1\,\mathbf{W}\,\Phi_2 \mid \Phi_1\,\mathbf{U}\,\Phi_2.$$

We call the formulas generated by Φ the *state formulas*, and the formulas generated by φ the *path formulas*. The satisfaction relation $\models$ is defined on both sets by induction as follows.

Definition 1 (Semantics)**.** *Given a Markov chain $\mathcal{M}$, state s, and path ξ, the satisfaction relation $\models$ is defined inductively as follows, where $\mathbb{P}_{\mathcal{M},\pi}$ denotes the probability measure induced on paths ξ by a policy π. (c.f. (Baier and Katoen 2008b) for full details).*

***State formulas**:*

$(\mathcal{M}, s) \models a$	*iff*	$a \in L(s)$
$(\mathcal{M}, s) \models \neg\Phi_1$	*iff*	$(\mathcal{M}, s) \not\models \Phi_1$
$(\mathcal{M}, s) \models \Phi_1 \wedge \Phi_2$	*iff*	$(\mathcal{M}, s) \models \Phi_1$ *and* $(\mathcal{M}, s) \models \Phi_2$
$(\mathcal{M}, s) \models \Phi_1 \vee \Phi_2$	*iff*	$(\mathcal{M}, s) \models \Phi_1$ *or* $(\mathcal{M}, s) \models \Phi_2$
$(\mathcal{M}, s) \models \mathbb{P}_{\geq q}(\varphi)$	*iff*	$\mathbb{P}_{\mathcal{M}^s}\big(\{\xi \in \textit{Paths}(s), (\mathcal{M}, \xi) \models \varphi\}\big) \geq q$.

***Path formulas**:*

$(\mathcal{M}, \xi) \models \mathbf{X}\Phi_1$	*iff*	$\mathcal{M}^{\xi[1]} \models \Phi_1$
$(\mathcal{M}, \xi) \models \Phi_1\,\mathbf{U}\,\Phi_2$	*iff*	$\exists\, j \in \mathbb{N}$, $(\mathcal{M}, \xi[j]) \models \Phi_2$ *and* $\forall\, i \leq j$, $(\mathcal{M}, \xi[i]) \models \Phi_1$,
$(\mathcal{M}, \xi) \models \Phi_1\,\mathbf{W}\,\Phi_2$	*iff*	$(\mathcal{M}, \xi) \models (\Phi_1\,\mathbf{U}\,\Phi_2) \vee (\mathbf{G}\,\Phi_1)$,
where $(\mathcal{M}, \xi) \models \mathbf{G}\,\Phi_1$	*iff*	$\forall j \in \mathbb{N}$, $\xi[j] \models \Phi_1$.

Satisfaction, Markov Chain $(\mathcal{M}, s) \models \Phi$ is alternatively denoted $\mathcal{M}^s \models \Phi$.

initial state whose associated probability distribution is the aforementioned initial probability distribution.

Satisfaction and probabilities in an MDP For a Markov Decision Process $\mathcal{M}$ and a policy π and a path formula ϕ, we write $\mathbb{P}_\pi(\phi)$ to denote the probability of satisfying ϕ in the Markov Chain induced by π in $\mathcal{M}$. (Baier and Katoen 2008b; Bertsekas and Shreve 2007). For Φ a state formula, we denote $\mathcal{M}_\pi \models \Phi$ to denote the satisfaction of Φ in the induced Markov Chain $\mathcal{M}_\pi$. For more details on MDPs and induced Markov chains, we refer to (Baier and Katoen 2008b; Bertsekas and Shreve 2007).

CPCTL The Continuing fragment of PCTL is a subset of the safe fragment and strictly generalizes the multi-objective avoidance case. Its syntax is given by

$$\Phi \;::=\; a \mid \neg a \mid \Phi \wedge \Phi \mid \mathbb{P}_{\geq q}(\varphi), \quad a \in AP \cup \{\bot, \top\},$$
$$\varphi \;::=\; \Phi_1\,\mathbf{W_c}\,\Phi_2,$$

where $\Phi_1\mathbf{W_c}\Phi_2 = \Phi_1\mathbf{W}(\Phi_1 \wedge \Phi_2)$.

Notice that CPCTL restricts disjunctions of state formulas and only allows the weak until $\mathbf{W_c}$ in its "continuing" form.

3 CPCTL Shielding: Theoretical Background

In this section, we define the problem and introduce the notions and results that are the theoretical backbone of our algorithm.

3.1 Problem Statement

We first introduce the problem of optimizing the reward of a policy π in a Markov Decision Process $\mathcal{M}$ under a CPCTL specification.

Definition 2. ***CPCTL Constrained optimization problem.*** *Let $\mathcal{M}$ be an MDP and Φ be a CPCTL formula. Find a policy π that has the highest discounted expected reward among the policies that are $\Phi-$safe, in other words*

$$\mathcal{R}_\mathcal{M}(\pi) = \max_{\tilde\pi \in \Pi_\Phi} \mathcal{R}_\mathcal{M}(\tilde\pi), \text{ where } \Pi_\Phi = \{\tilde\pi,\ \mathcal{M}_{\tilde\pi} \models \Phi\}.$$

3.2 The Bellman Operator

In this section, we introduce the Bellman operator for CPCTL. Our definition differs from that in (Ohlmann, le Court, and Belardinelli 2025) in the sense that we allow the policies to depend on the history of actions in addition to the history of states. We first introduce the technical definitions required to define the operator.

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we denote $\boldsymbol{x} \leq \boldsymbol{y}$ if and only if for all $j \in \{1, \dots, n\}$, $x_j \leq y_j$. For $E \subseteq (\mathbb{R}_+)^n$, we define the downward closure $E_\downarrow = \{\boldsymbol{x} \in (\mathbb{R}_+)^n,\ \exists \boldsymbol{y} \in E,\ \boldsymbol{x} \leq \boldsymbol{y}\}$. Let Φ be a CPCTL formula, $\mathcal{SF}(\Phi)$ the set of state subformulas of Φ and $\mathcal{PF}(\Phi)$ the set of path subformulas of Φ. We denote $sf(\Phi) = |\mathcal{SF}(\Phi)|$ and $pf(\Phi) = |\mathcal{PF}(\Phi)|$. Then, we define the set of **profiles** $\mathtt{P}(\Phi) = \{0, 1\}^{sf(\Phi)} \times [0, 1]^{pf(\Phi)}$. We index by $\Psi \in \mathcal{SF}(\Phi)$ the valuations $\boldsymbol{\mu}$ and by $\psi \in \mathcal{PF}(\Phi)$ the counters $\boldsymbol{\nu}$, so we write $\boldsymbol{\mu} = (\mu_\Psi)_{\Psi\in\mathcal{SF}(\Phi)}$ and $\boldsymbol{\nu} = (\nu_\psi)_{\psi\in\mathcal{PF}(\Phi)}$.

Definition 3 (Bellman Operator for CPCTL)**.** *The* extended Bellman operator $\mathcal{B}_\Phi$ *acts on elements* $V \in \mathcal{P}\left(\mathtt{P}(\Phi)\right)^{|S|}$, *and is such that, for any* $s \in S$, $\mathcal{B}_\Phi[V](s) = E_\downarrow$,

where E is the set of all pairs of tuples $(\boldsymbol{\mu}, \boldsymbol{\nu}) = ((\mu_\Psi)_{\Psi \in \mathcal{SF}(\Phi)}, (\nu_\psi)_{\psi \in \mathcal{PF}(\Phi)})$ such that there exists a distribution δ over the actions $A(s)$ and for every $a \in A(s)$ $(\boldsymbol{\mu}_a^{s'}, \boldsymbol{\nu}_a^{s'})_{s' \in S}$ with $(\boldsymbol{\mu}_a^{s'}, \boldsymbol{\nu}_a^{s'}) \in V(s')$ for all $s' \in S$ such that

Valuations for state formulas: *for all $\Psi \in \mathcal{SF}(\Phi)$,*

- *If $\Psi = a$, then $\mu_\Psi = 1$ if $a \in L(s)$ and 0 otherwise.*
- *If $\Psi = \neg a$, then $\mu_\Psi = 0$ if $a \in L(s)$ and 1 otherwise.*
- *If $\Psi = \Psi_1 \wedge \Psi_2$, then $\mu_\Psi = \mu_{\Psi_1} \mu_{\Psi_2}$.*
- *If $\Psi = \mathbb{P}_{\geq p}[\psi]$, then $\mu_\Psi = 1$ if $\nu_\psi \geq p$ and 0 otherwise.*

Counters for path formulas: *for all $\psi \in \mathcal{PF}(\Phi)$*

- *If $\psi = \Psi_1 \mathbf{W_C} \Psi_2$, then*

$$\nu_\psi = \begin{cases} 0 \text{ if } \mu_{\Psi_1} = 0, \\ 1 \text{ if } \mu_{\Psi_1} = \mu_{\Psi_2} = 1, \\ \sum_{s' \in S} \sum_{a \in A(s)} \delta(a)\, P(s'|s,a)\, \nu_{a,\psi}^{s'} \text{ otherwise.} \end{cases}$$

We now introduce the class of *inductive sets*, which are the sets that are preserved by the Bellman operator.

Definition 4. ***Inductive sets.*** *Let $V : S \to P(\mathcal{P}(\Phi))$. We say that V is **inductive** if for every $s \in S$, $\mathcal{B}_\Phi(V)(s) \subseteq V(s)$.*

As a particular case of inductive set, we introduce the realizability set V_Φ. To each state $s \in V$, it associates the set of profiles $(\boldsymbol{\mu}, \boldsymbol{\nu})$ such that there exists a policy π such that μ_Ψ corresponds to the satisfaction of Ψ in $\mathcal{M}_\pi^s$ and ν_ψ corresponds to the probability of satisfying ψ starting from s and following π.

Realizability set V_Φ. *Let $\Phi \in$ CPCTL and $\mathcal{M}$ be an MDP. We define V_Φ for all $s \in S$ as the set of $(\boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathcal{P}(\Phi)$ such that there exists a policy π satisfying*

- *For all $\Psi \in \mathcal{SF}(\Phi)$, $\mu_\Psi = 1$ when $\mathcal{M}_\pi \models \Phi$ and 0 otherwise.*
- *For all $\psi \in \mathcal{PF}(\Phi)$, $\nu_\psi = \mathbb{P}_\pi(\psi|s)$.*

Note that V_Φ is an inductive set satisfying $\mathcal{B}_\Phi(V_\Phi) = V_\Phi$.

The following result states that the realizability set V_Φ is, in fact, the largest possible inductive set.

Lemma 1. *Let V be any inductive value frontier. Then, for any $s \in S$, $V(s) \subseteq V_\Phi(s)$.*

Finally, we define the canonical valuation map Ξ that associate to each counters $\boldsymbol{\nu} \in \mathcal{PF}(\Phi)$ a valuation $\boldsymbol{\mu} \in \mathcal{SF}(\Phi)$.

Definition 5. ***Canonical Valuation Ξ.*** *Let $\Phi \in$ CPCTL, s be a state in $\mathcal{M}$, and $\boldsymbol{\nu}$ be a set of counters. We define ξ by induction as*

$$\begin{cases} \xi(a, \boldsymbol{\nu}, s) = 1 \text{ if } a \in L(s),\ 0 \text{ otherwise, for} a \in AP, \\ \xi(\Psi_1 \wedge \Psi_2, \boldsymbol{\nu}, s) = \xi(\Psi_1, \boldsymbol{\nu}) \xi(\Psi_2, \boldsymbol{\nu}), \\ \xi(\mathbb{P}_{\geq p}[\psi], \boldsymbol{\nu}, s) = 1 \text{ if } \nu_\psi \geq p,\ 0 \text{ otherwise}. \end{cases}$$

Finally, we set $\Xi(\boldsymbol{\nu}, \Phi, s) = (\mu_\Psi)_{\Psi \in \mathcal{SF}(\Phi)}$, simply denoted $\Xi(\boldsymbol{\nu}, s)$ where $\mu_\Psi = \xi(\Psi, \boldsymbol{\nu}, s)$.

3.3 Augmentation

In this section, we introduce the definitions and results that will constitute the theoretical backbone of our algorithm. The algorithm relies on an augmentation method, and a shielding mechanism that restricts the distribution of actions that the policy is allowed to take.

The augmented MDP $\mathcal{M} \times [\Phi]$

Definition 6. *Let $\mathcal{M}$ be a MDP and Φ be a* CPCTL *formula. We define $\mathcal{M} \times [\Phi]$ the **Markov Decision Process M augmented by** Φ as*

- ***Set of States:*** *$\overline{S} = S \times \mathcal{P}(\Phi)$.*
- ***Initial states:*** *The initial states are $\overline{I} = \{(s_0, \boldsymbol{\mu}, \boldsymbol{\nu}),\ s.t. (\boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathcal{P}(\Phi)\}$*
- ***Labels:*** *$L(s, \boldsymbol{\mu}, \boldsymbol{\nu}) = L(s)$.*
- ***Actions:*** *For $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \overline{S}_+$, an augmented action $\overline{a} = (a, \Theta)$ is given by an action $a \in A(s)$ and a profile map $\Theta : \text{succ}(s) = \{s_1, \ldots, s_{m(s)}\} \to \mathcal{P}(\Phi)$.*
- ***Transitions:*** *Let $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \overline{S}_+$ and $\overline{a} = (a, \Theta)$. The distribution $P(\overline{s}, \overline{a})$ is defined as the mixture of Dirac distributions $P(\overline{s}, \overline{a}) = \sum_{j=1}^{m(s)} P(s_j|s,a) \mathbf{1}_{\overline{s}_j}$, where $\overline{s}_j = (s_j, \boldsymbol{\mu}^j, \boldsymbol{\nu}^j)$.*

In Figure 1 and Figure 2, we illustrate the shape of the actions in the augmented MDP $\mathcal{M} \times [\Phi]$.

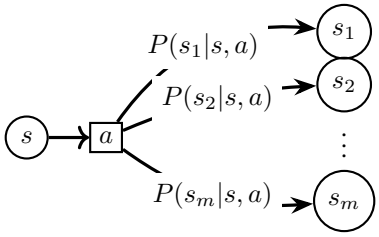


Figure 1: An action in $\mathcal{M}$

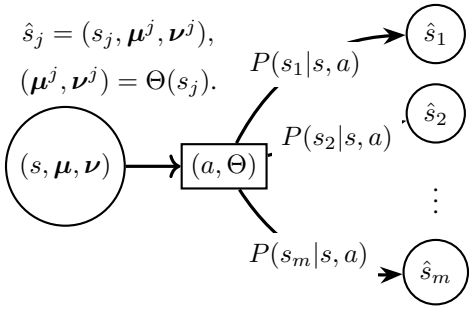


Figure 2: An augmented action in $\overline{\mathcal{M}}$.

Intuitively, for a state sub-formula Ψ of Φ, the valuation latent parameter $\boldsymbol{\mu}_\Psi$ corresponds to the satisfaction of the sub-formula Ψ from this state. For a path sub-formula ψ of Φ, the counter latent parameter $\boldsymbol{\nu}_\psi$ corresponds to the probability of satisfying ψ from this state. We introduce now

the valued policies, the policies that naturally correspond to policies in $\mathcal{M}$.

Definition 7. ***Valued distributions.*** *Let $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu})$ be an augmented state, and $\overline{\delta}$ be a distribution over the augmented actions $\overline{A}(\overline{s})$. We say that $\overline{\delta}$ is a **valued distribution** if there exist a **de-augmented distribution** δ and a choice of counters $\Theta_a : succ(a) \to \mathcal{P}(\Phi)$ for each action $a \in A(s)$ such that the distribution $\overline{\pi}$ satisfies*

$$\overline{\delta} = \sum_{a \in A(s)} \delta(a)\, \mathbf{1}_{(a,\Theta_a)}.$$

Moreover, for any $a \in A(s)$ and $s' \in \mathrm{succ}(s,a)$, $\Theta_a(s') = \left(\Xi(\boldsymbol{\nu}^{a,s'}, s'), \boldsymbol{\nu}^{a,s'}\right)$ for some $\boldsymbol{\nu}^{a,s'}$.

We now provide an example to illustrate which policies of the augmented MDP $\mathcal{M} \times [\Phi]$ are valued and which are not.

Example 1. *Let $\Phi = \mathbb{P}_{\geq p}[\phi]$, where $\phi = [a \mathbf{W}_{\mathbf{C}} b]$. We have $\mathcal{PF}(\Phi) = \{\phi\}$ and $\mathcal{SF}(\Phi) = \{\Phi, a, b\}$. The set of profiles is thus $\{0,1\}^3 \times [0,1]$.*

We consider a state s in a MDP $\mathcal{M}$, with two available actions and two successors for each action.

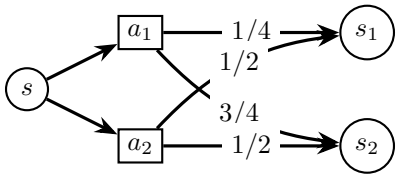


Figure 3: Valued distributions - Example

We have $\overline{A}(\overline{s}) = \bigcup_{i=1,2}\{(a_i, \Theta),\ \Theta \in (\{0,1\}^3 \times [0,1])^2\}$ and define

$$\begin{cases} \Theta_1 : s_1 \mapsto ((0,1,1), 0.5),\ s_2 \mapsto ((1,1,1), 0.3), \\ \Theta_2 : s_1 \mapsto ((1,0,0), 0.5),\ s_2 \mapsto ((1,1,1), 0.3), \end{cases}$$

We consider $\overline{\delta}_1$ the choice defined as

- *picks the augmented action (a_1, Θ_1) with probability $1/4$.*
- *picks the augmented action (a_1, Θ_2) with probability $1/4$.*
- *picks the augmented action (a_2, Θ_2) with probability $1/2$.*

Then, $\overline{\delta}_1$ is not a valued choice.

We consider $\overline{\delta}_2$ the choice defined as

- *picks the augmented action (a_1, Θ_1) with probability $1/3$.*
- *picks the augmented action (a_2, Θ_2) with probability $2/3$.*

Then, $\overline{\delta}_2$ is a valued choice and is equal to the mixture

$$\overline{\delta}_2 = \frac{1}{3}\mathbf{1}_{(a_1,\Theta_2)} + \frac{2}{3}\mathbf{1}_{(a_2,\Theta_2)}.$$

State and Path compatibility We now provide two conditions, the state and path compatibilities. These two conditions will ensure that the latent valuations and counters respectively encode the satisfaction of the state formulas and the probabilities of the path formulas.

Definition 8. ***Path and state compatibilities.***

*Let $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \overline{S}_+$ be an augmented state. We say that $\overline{s}$ is **state-compatible** if and only if for every $\Psi \in \mathcal{SF}(\Phi)$,*

- *If $\Psi = \Psi_1 \wedge \Psi_2$, $\mu_\Psi = 1 \Rightarrow (\mu_{\Psi_1} = 1) \wedge (\mu_{\Psi_2} = 1)$.*
- *If $\Psi = \mathbb{P}_{\geq p}[\psi]$, then $(\mu_\Psi = 1) \Rightarrow (\nu_\psi \geq p)$.*
- *If $\Psi = a \in AP$, then $(\mu_\Psi = 1) \Rightarrow (a \in L(s))$.*
- *If $\Psi = \neg a,\ a \in AP$, then $(\mu_\Psi = 1) \Rightarrow (a \notin L(s))$.*

*Let $\hat{\delta}$ be a valued distribution characterized by its de-augmented distribution δ and the profiles $\Theta_a(s') = (\Xi(\boldsymbol{\nu}^{a,s'}), \boldsymbol{\nu}^{a,s'})$ for $a \in A(s)$ and $s' \in succ(a)$. It is **path-compatible** if and only if for every $\psi = \Psi_1 \mathbf{W}_{\mathbf{c}} \Psi_2 \in \mathcal{PF}(\Phi)$,*

$$\nu_\psi \leq \max\left(\mu_{\Psi_1} \sum_{a \in A(s)} \sum_{s' \in succ(a)} \delta(a) P(s'|s,a) \nu_\psi^{a,s'}, \mu_{\Psi_1}\mu_{\Psi_2}\right).$$

Definition 9. ***(V-shielded policies)*** *Let $\overline{\pi}$ be a valued policy. We say that $\overline{\pi}$ is $V-$shielded if for any any reachable history $\overline{\xi} = (\overline{s}_1, \ldots, \overline{s}_n)$,*

- ***State compatibility:*** *$\overline{s}_n$ is state compatible.*
- ***Path compatibility:*** *The distribution of actions $\overline{\pi}(\overline{\xi})$ is path compatible.*
- ***Realizability:*** *For each augmented action $\overline{a} = (a, \Theta_a)$ in $\overline{A}(\overline{s}_n)$, $P_{\overline{\pi}}(\overline{a}|\overline{\xi}) \neq 0$ implies $\Theta_a(s') \in V(s')$.*

The memory-less and valued policies of the augmented MDP $\mathcal{M} \times [\Phi]$ correspond to history dependent policies of $\mathcal{M}$. We make this intuition more precise in the next definition.

Definition 10. ***(Projection of $\mathcal{M} \times [\Phi]$ onto $\mathcal{M}$).*** *Let $\overline{\pi}$ be a memory-less and valued policy of $\mathcal{M} \times [\Phi]$. We define π, the projection of $\overline{\pi}$ onto $\mathcal{M}$, as:*

- *The memory states are of the form $m(\xi) = (\boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathcal{P}(\Phi)$ for histories $\xi = (s_0 a_0 \ldots a_{n-1} s_n)$.*
- *The initial state is s_0 and the initial memory state is $m_0 = (\boldsymbol{\mu}_0, \boldsymbol{\nu}_0)$, where $(s_0, \boldsymbol{\mu}_0, \boldsymbol{\nu}_0)$ is the initial state of $\overline{\pi}$.*
- *For any state $s \in S$, history ξ and memory state $m = m(\xi)$, where $m = (\boldsymbol{\mu}, \boldsymbol{\nu})$, we define the distribution of actions $\pi(s,m)$ as*

$$\pi(s, m(\xi)) = \sum_{a \in A(s)} \delta(a) \mathbf{1}_a,$$

and the memory state becomes $m(\xi a s') = \Theta_a(s')$ after reaching the state s' with the action a, where

$$\overline{\pi}(s, \boldsymbol{\mu}, \boldsymbol{\nu}) = \sum_{a \in A(s)} \delta(a) \mathbf{1}_{(a,\Theta_a)}.$$

The following theorem formalizes the intuition that shielded policies in $\mathcal{M} \times [\Phi]$ correspond to safe policies in $\mathcal{M}$.

Theorem 1. ***(Safety of shielded policies in $\mathcal{M} \times [\Phi]$.)*** *The following properties hold:*

(i) $(\mathcal{M} \times [\Phi])_{\overline{\pi}} \models \Phi$ *implies* $\mathcal{M}_\pi \models \Phi$.

(ii) Any V*-shielded policy of* $\mathcal{M} \times [\Phi]$ *satisfies* $(\mathcal{M} \times [\Phi])_{\overline{\pi}} \models \Phi$.

Proof. (i) We first show that for any $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu}) = (s, m)$ and $\overline{P}_n(\overline{s})$ (resp $P_n(m)$) the probability of reaching the augmented state $\overline{s}$ (resp. the state s with memory m), we have $\overline{P}_n(\overline{s}) = P_n(s, m)$. $P_{n+1}(s, m)$ satisfies

$$P_{n+1}(s, m) = \sum_{s' \in S} \sum_{m' \in \mathcal{P}(\Phi)} P_n(s', m') P(s, m | s', m'),$$

since $P(s, m|s', m') = \sum_{a \in A(s),\ \Theta_a(s) = m'} \delta_{\overline{s}}(a) P(s'|a)$ where $\overline{\pi}(\overline{s}) = \sum_{a \in A(s)} \delta_s(a) \mathbf{1}_{(a, \Theta_a)}$ we obtain using the induction assumption

$$P_{n+1}(s, m) = \sum_{\overline{s}' = (s', \boldsymbol{\mu}', \boldsymbol{\nu}') \in \overline{S}} \overline{P}_n(\overline{s}') P(\overline{s}' | \overline{s}) = \overline{P}_{n+1}(\overline{s}),$$

since $\overline{P}(\overline{s}|\overline{s}') = \sum_{a \in A(s),\ \Theta_a(s) = (\boldsymbol{\mu}', \boldsymbol{\nu}')} \delta_{\overline{s}}(a) P(s'|a)$.

We deduce that for any history of states $\overline{\xi} = (\overline{s}_1 \dots \overline{s}_n)$ and the corresponding history of state-memory pairs $\xi = ((s_1, m_1) \dots (s_n, m_n))$ with $\overline{s}_j = (s_j, m_j)$, we have $\mathbb{P}_\pi(\xi) = \mathbb{P}_{\overline{\pi}}(\overline{\xi})$. Since for any $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu})$, $L(\overline{s}) = L(s)$, we deduce that for any finite word ω over the subset of atomic propositions AP,

$$\mathbb{P}_\pi(\omega) = \sum_{\xi | L(\xi) = \omega} \mathbb{P}_\pi(\xi) = \sum_{\overline{\xi} | L(\overline{\xi}) = \omega} \mathbb{P}_{\overline{\pi}}(\xi) = \mathbb{P}_{\overline{\pi}}(\omega).$$

(ii) follows from the coherence theorem available in (Ohlmann, le Court, and Belardinelli 2025). □

From the previous theorem, we know that the projection of a memory-less V-shielded policy $\overline{\pi}$ correspond to a history dependent policy π in $\mathcal{M}$. However, not all history dependent policies in $\mathcal{M}$ are the projection of a memory-less V-shielded policy in $\mathcal{M} \times [\Phi]$. We introduce the following theorem, stating that considering only those policies is enough to solve the optimization problem in $\mathcal{M}$.

Theorem 2. ***(Optimality of shielded policies in*** $\mathcal{M} \times [\Phi]$***).*** *For any memory-less and valued policy* $\overline{\pi}$ *of* $\mathcal{M} \times [\Phi]$, $\mathcal{R}_{\mathcal{M}}(\pi) = \mathcal{R}_{\mathcal{M} \times [\Phi]}(\overline{\pi})$. *Moreover, for any* π *policy of* $\mathcal{M}$ *satisfying* $\pi \models \Phi$, *there exists* $\overline{\pi}$ *a* V_Φ*-shielded policy of* $\mathcal{M} \times [\Phi]$ *such that* $\mathcal{R}_{\mathcal{M} \times [\Phi]}(\overline{\pi}) = \mathcal{R}_{\mathcal{M}}(\pi)$.

Proof. (Sketch) We consider π a policy of $\mathcal{M}$ satisfying $\pi \models \Phi$. We will define $\overline{\pi}$, a policy on $\mathcal{M} \times [\Phi]$. For any history $\overline{\xi} = (\overline{s}_1, \overline{a}_1, \dots, \overline{a}_{n-1}, \overline{s}_n)$, we define $\xi = (s_1, a_1, \dots, s_{n-1}, a_{n-1})$, where for any $k \leq n$, $\overline{s}_k = (s_k, \boldsymbol{\nu}^k, \boldsymbol{\nu}^k)$ and $\overline{a}_k = (a_k, \Theta_{a_k})$.

We define

$$\overline{\pi}(\overline{\xi}) = \sum_{a \in A(s)} \delta(a) \mathbf{1}_{(a, \Theta_a)},$$

where δ is defined as $\pi(\xi) = \sum_{a \in A(s)} \delta(a)$, and for $s' \in \text{succ}(s)$, $\Theta_a(s') = (\Xi(\boldsymbol{\nu}^{a,s'}), \boldsymbol{\nu}^{a,s'})$, where for any ψ path sub-formula of Φ, $\boldsymbol{\nu}_\psi^{a,s'} = P(\psi | \pi, s', m(\xi a s'))$. It holds

- Since any reachable augmented state $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu})$ satisfies $\boldsymbol{\mu} = \Xi(\boldsymbol{\nu})$, the policy $\overline{\pi}$ is state compatible.
- Since $\mathbb{P}_\pi(\psi | s, m(\xi)) = \sum_{a \in A(s)} \delta(a) \sum_{s' \in \text{succ}(a)} P(s'|a)$ for any path formula ψ, we obtain that $\overline{\pi}$ is path compatible.
- We consider an augmented history of the form $\overline{\xi} = (\overline{s}_1 \overline{a}_1 \dots \overline{a}_{n-1} \overline{s}_n)$, where $\overline{s}_k = (s_k, \boldsymbol{\mu}_k, \boldsymbol{\nu}_k)$ and $\overline{a}_k = (a_k, \Theta^k)$. We define $\xi = (s_1 a_1 \dots a_{n-1} s_n)$ and show by induction on n that $\mathcal{R}_n(\overline{\pi}, \overline{\xi}) = \mathcal{R}_n(\pi, \xi)$, where $\mathcal{R}_n$ denotes the discounted expected reward in n steps.

□

3.4 From Local Constraints to Unconstrained

In this section, we define a new augmented MDP, $\hat{\mathcal{M}}_V$. In this new MDP, we only consider the $V-$shielded policies of $\mathcal{M} \times [\Phi]$, which allows us to reduce the problem of optimizing the reward among the constrained problem in $\mathcal{M} \times [\Phi]$ to the unconstrained problem of optimizing the reward in $\hat{\mathcal{M}}_V$. In $\hat{\mathcal{M}}_V$, only the distributions of augmented actions that will ensure that an augmented policy is $V-$shielded are available.

Definition 11 (The augmented MDP $\hat{\mathcal{M}}_V$)**.** *Let* $\mathcal{M}$ *be an MDP and* V *be an inductive set. We define the unconstrained MDP* $\hat{\mathcal{M}}_V$ *as:*

- *States* $\hat{S}_V = S \times \texttt{Pr}(\Phi)$.
- *Initial states* $\hat{I}_V = \{s_0\} \times V(s_0)$.
- *Labels* $\hat{L}_V(s, \boldsymbol{\mu}, \boldsymbol{\nu}) = L(s)$.
- *Actions: Let* $\hat{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \hat{S}_V$.

$$\hat{A}_V(\hat{s}) = \Big\{ (\delta, \Lambda),\ \delta \in \mathcal{D}(A(s)), \Lambda \in \Big([0,1]^{pf(\Phi)} \Big)^{|S|},\ \forall i,\ \Theta(s_i) = (\Xi(\Lambda(s_i)), \Lambda(s_i)) \in V(s_i) \text{ and } \forall \psi = \Psi_1 \mathbf{W_C} \Psi_2 \in \mathcal{PF}(\Phi),$$

$$\nu_\psi \leq \max \Big(\mu_{\Psi_1} \sum_{s_i \in S} \sum_{a \in A(s)} \delta(a) P(s_i | s, a) \Lambda(s_i)_\psi,\ \mu_{\Psi_1} \mu_{\Psi_2} \Big) \Big\}.$$

- *Transitions: Let* $\hat{a} = (\delta, \Lambda)$ *and* $s' \in S$. *Then*

$$P_V(s', \Xi(\Lambda(s')), \Lambda(s') | (s, \boldsymbol{\mu}, \boldsymbol{\nu}), \hat{a}) = \sum_{a \in A(s)} \delta(a) P(s' | s, a).$$

  For any other $\hat{s}' = (s', \boldsymbol{\mu}', \boldsymbol{\nu}') \in \hat{S}_V$, $P_V(\hat{s}' | (s, \boldsymbol{\mu}, \boldsymbol{\nu}), \hat{a}) = 0$.
- *Rewards:* $\hat{r}(\hat{s}, \hat{a}) = \sum_{a \in A(s)} \delta(a) r(s, a)$.

Since V is inductive, we obtain that the agent will never get "stuck". We make this intuition more precise in the following lemma.

Lemma 2. *Let* V *be an inductive set, and* $\hat{s}$ *be any reachable state of* $\hat{I}_V$. *Then,* $\hat{A}_V(\hat{s}) \neq \emptyset$.

We now explain how $\mathcal{M} \times [\Phi]$ and $\hat{\mathcal{M}}_V$ are related.

Definition 12. ***(Projection of $\hat{M}_V$ onto $\mathcal{M}\times[\Phi]$).*** *Let $\hat{\pi}$ be a memory-less and deterministic policy of $\hat{M}_V$. We define the valued and memory-less policy $\overline{\pi}$, the projection of $\hat{\pi}$ onto $\mathcal{M} \times [\Phi]$, as follows. For any augmented state $\overline{s}$, $\hat{s} = \overline{s} \in \hat{S}_V$, so there exists δ and Λ such that $\hat{\pi}(\hat{s}) = \mathbf{1}_{(\delta,\Lambda)}$. We define the distribution of actions $\overline{\pi}(\overline{s})$ as*

$$\overline{\pi}(\overline{s}) = \sum_{a \in A(s)} \delta(a)\mathbf{1}_{(a,\Theta_a)},$$

where for all $a \in A(s)$ and $s' \in succ(a)$, $\Theta_a(s') = \Lambda(a, s')$.

We now introduce two theorems. The first one states that any policy of the unconstrained MDP $\hat{\mathcal{M}}_V$ corresponds to a safe policy in $\mathcal{M} \times [\Phi]$, and the second one states that considering only the $V-$shielded policies that can be seen as a policy in $\hat{\mathcal{M}}_V$ is enough to recover optimality of the reward.

Theorem 3. ***(Safety from $\hat{\mathcal{M}}_V$ to $\mathcal{M} \times [\Phi]$).*** *For any deterministic and memory-less policy $\hat{\pi}$ of $\hat{\mathcal{M}}_V$, the projected policy $\overline{\pi}$ of $\mathcal{M} \times [\Phi]$ is V-shielded and thus Φ-safe.*

Proof. The fact that $\overline{\pi}$ is V-shielded follows from the definition of $\hat{A}(\hat{s})_V$. The fact that $\overline{\pi}$ is thus $\Phi-$safe comes from the coherence theorem available in (Ohlmann, le Court, and Belardinelli 2025). □

Theorem 4. ***(Optimality from $\hat{\mathcal{M}}_V$ to $\mathcal{M} \times [\Phi]$)***

(i) For any deterministic and memory-less policy $\hat{\pi}$ of $\hat{\mathcal{M}}_V$, the projection $\overline{\pi}$ of $\hat{\pi}$ onto $\mathcal{M} \times [\Phi]$ satisfies $\mathcal{R}(\overline{\pi}) = \mathcal{R}(\hat{\pi})$.

(ii) Moreover, for any $V-$shielded policy $\overline{\pi}$ of $\mathcal{M} \times [\Phi]$, there exists a policy $\hat{\pi}$ of $\hat{\mathcal{M}}_V$ such that $\mathcal{R}(\overline{\pi}) = \mathcal{R}(\hat{\pi})$.

(iii) Finally, there exists a memory-less and deterministic policy $\hat{\pi}_0$ of $\hat{\mathcal{M}}_V$ such that with $\hat{\Pi}$ the set of all policies of $\hat{\mathcal{M}}_V$, $\mathcal{R}(\hat{\pi}_0) = \max_{\hat{\pi}\in\hat{\Pi}} \mathcal{R}(\hat{\pi})$.

Proof. (i) We show by induction that with $\mathcal{R}_n(\overline{s})$ the expected discounted reward in n steps starting from $\overline{s}$ in $\mathcal{M} \times [\Phi]$ and $\mathcal{R}_n(\hat{s})$ the expected discounted reward in n steps starting from $\hat{s}$ in $\hat{\mathcal{M}}_V$, it holds with $\hat{s} = \overline{s}$ that $\mathcal{R}_n(\overline{\pi}) = \mathcal{R}_n(\hat{\pi})$. The initialization holds as the reward in zero steps is zero. Furthermore, with $\hat{\pi}(\hat{s}) = \mathbf{1}_{(\delta,\Lambda)}$,

$$\mathcal{R}_{n+1}(\hat{s}) = r(\delta, \Lambda) + \gamma \sum_{\hat{s}'} P(\hat{s}'|\delta, \Lambda)\mathcal{R}_n(\hat{s}').$$

Using the definition of $\hat{r}$, the definition of the transition function in $\hat{\mathcal{M}}_V$ and the induction hypothesis, we obtain

$$\mathcal{R}_{n+1}(\hat{s}) = \sum_{a\in A(s)} \delta(a)r(s, a) + \gamma \sum_{\hat{a}\in A(s)} \sum_{s'\in \text{succ}(a)} \delta(a)\mathcal{R}_n(s', \Lambda(a, s')).$$

So $\mathcal{R}_{n+1}(\hat{s}) = \mathcal{R}_n(\overline{s})$.

(ii) We consider $\overline{\pi}$ a policy on $\mathcal{M} \times [\Phi]$ and define $\hat{\pi}$ the history dependent policy of $\hat{\mathcal{M}}_V$. Its memory states are of the form $\{(a_j, \zeta_j)\}_{j\in J}$, where $\zeta_j = (a^j_1, \dots, a^j_{n-1})$.

Let $\hat{\xi} = (\hat{s}_1\hat{a}_1 \dots \hat{a}_{n-1}\hat{s}_n)$ be a history in $\hat{\mathcal{M}}_V$, where for all $j \in \{1, \dots, n\}$, $\hat{s}_j = (s_j, \boldsymbol{\mu}_j, \boldsymbol{\nu}_j)$ and for all $j \in \{1, \dots, n-1\}$, $\hat{a}_j = (\delta_j, \Lambda_j)$, and assume that the memory state is of the form $m(\hat{\xi}) = \{(\alpha_j, \hat{\zeta}_j)\}_{j\in J}$. For each $\hat{\zeta}_j = (a^j_1, \dots, a^j_{n-1})$, we define $\overline{\zeta}_j = (\hat{s}_1(a^j_1, \Theta^j_1) \dots (a_{n-1}, \Theta^j_{n-1})\hat{s}_n)$, where $(\Theta^j_k)_a = \Lambda(a^j_k, s_{k+1})$.

For every $j \in J$, since $\overline{\pi}$ is valued, there exists δ_j and Θ^j_a such that

$$\overline{\pi}(\overline{\zeta}_j) = \sum_{a\in A(s)} \delta^j(a)\mathbf{1}_{(a,\Theta^j_a)}.$$

We define the distribution of actions $\hat{\pi}(\hat{\xi})$ as

$$\hat{\pi}(\hat{\xi}) = \sum_{j\in J} \alpha_j \mathbf{1}_{(\delta_j,\Lambda_j)},$$

where $\Lambda_j(a, s') = \Theta^j_a(s')$. Finally, after taking the action (δ_j, Λ_j) and reaching the state $\hat{s}' = (s', \boldsymbol{\mu}, \boldsymbol{\nu})$ the memory state becomes

$$m(\hat{\xi}) = \bigcup_{a\in A(s),\ s'\in \text{succ}(a)} \{(\alpha_j \cdot \delta^j(a), \zeta_j a)\}_{j\in J}.$$

First, since $\overline{\pi}$ is $V-$shielded, we indeed have $(\delta_j, \Lambda_j) \in \hat{A}(\hat{s}_n)_V$. With an argument similar to the one used for Theorem 4-(i) or Theorem 4, it follows by construction that $\mathcal{R}(\overline{\pi}) = \mathcal{R}(\hat{\pi})$.

(iii) Since $\hat{\mathcal{M}}_V$ satisfies the conditions for the lower semi-continuous model (Definition 8.7 of (Bertsekas and Shreve 2007)), it admits a memoryless, deterministic, and optimal policy (Corollary 9.17.2 of (Bertsekas and Shreve 2007)).

□

4 The Shielding Algorithm

In this section, we use the theoretical foundations established in the previous section to design an algorithm to solve the constrained optimization problem of maximizing the reward in an MDP $\mathcal{M}$ while satisfying a CPCTL constraint Φ. We state the theoretical guaranties on this algorithm provided by the previous section. Then, we provide details on the implementation of the algorithm.

4.1 The Algorithm

We present the high-level SC^2 algorithm in Algorithm 1. We show that SC^2 is safe and optimal, even during training.

We now introduce the following theorem, stating that the policy given by the algorithm is safe.

Theorem 5 (Safety of the output policy)**.** *The policy π that the algorithm outputs satisfies $\mathcal{M}_\pi \models \Phi$.*

Proof. Let $\hat{\pi}$ be the deterministic optimal policy in $\hat{\mathcal{M}}_V$. It follows from Theorem 3 that $\overline{\pi}$, the projection of $\hat{\mathcal{M}}_V$ onto $\mathcal{M} \times [\Phi]$ is Φ-safe. It follows from Theorem 1 that π, the projection of $\overline{\pi}$ onto $\mathcal{M}$ satisfies $\mathcal{M}_\pi \models \Phi$. □

Algorithm 1 SC^2

Input: MDP $\mathcal{M}$, CPCTL formula Φ
Output: A solution π to the Φ-constrained policy optimization problem
1: Compute an inductive frontier V_Φ for $\mathcal{M}$
2: Compute a deterministic optimal policy $\hat{\pi}^\star$ of $\hat{\mathcal{M}}_V$
3: Return the projection π^* of $\hat{\pi}^\star$ on $\mathcal{M}$

Moreover, the algorithm satisfies the following optimality property.

Theorem 6 (Optimality of the algorithm)**.** *With $\hat{\pi}^*$ the optimal policy in $\hat{\mathcal{M}}_V$, $\hat{\pi}^*$ can be chosen to be deterministic and memory-less, and the projection π^* on $\mathcal{M}$ is safe and solves, with Π_Φ the set of Φ-safe policies on $\mathcal{M}$,*

$$\mathcal{R}_{\mathcal{M}}(\pi^*) = \max_{\pi \in \Pi_\Phi} \mathcal{R}_{\mathcal{M}}(\pi).$$

Proof. The result follows from Theorem 2 and Theorem 4. □

Finally, the additional safety result holds.

Theorem 7 (Safety of the sampling)**.** *Let $\overline{\pi}_i$ be shielded policies and denote for any augmented state $\overline{s}$*

$$\overline{\pi}_i(\overline{s}) = \sum_{a \in A(s)} \delta_i(a) \mathbf{1}_{(a,\Theta_a^i)}.$$

For $\alpha_i \in [0,1]$ such that $\sum_i \alpha_i = 1$, we define the mixed policy $\overline{\pi}$ as

$$\overline{\pi}(\overline{s}) = \sum_{i=1}^{m} \alpha_i \sum_{a \in A(s)} \delta_i(a) \mathbf{1}_{(a,\Theta_a^i)}.$$

For every reachable augmented state $\overline{s}$, with $\mu_\Psi(s, \boldsymbol{\mu}, \boldsymbol{\nu}) = \mu_\Psi$ and $\nu_\psi(s, \boldsymbol{\mu}, \boldsymbol{\nu}) = \nu_\psi$

$$\forall \Psi \in \mathcal{SF}(\Phi),\ \mu_\Psi(\overline{s}) = 1 \Rightarrow \mathcal{M}_{\overline{\pi}}^{\overline{s}} \models \Psi,$$
$$\forall \psi \in \mathcal{PF}(\Phi),\ \nu_\psi(\overline{s}) \leq \mathbb{P}_{\overline{\pi}}(\psi|\overline{s}).$$

Proof. We show the result by induction over the subformulas of Φ. Note that any reachable state $\overline{s} = (s, \boldsymbol{\mu}, \boldsymbol{\nu})$ satisfies $\boldsymbol{\mu} = \Xi(\boldsymbol{\nu})$, and is thus self state-compatible.

We first consider Ψ a state sub-formula and assume that the result holds for all strict sub-formulas $\Psi \in \mathcal{SF}(\Psi)$ and $\psi \in \mathcal{PF}(\Psi)$. We consider a reachable state $\overline{s}$.

- If $\Psi = b \in AP$, then $\mu_\Psi(\overline{s}) = 1 \Rightarrow b \in L(s)$ by state compatibility, so $\mathcal{M}_{\overline{\pi}}^{\overline{s}} \models \Psi$.
- If $\Psi = \Psi_1 \wedge \Psi_2$, then $\mu_\Psi(\overline{s}) = 1$ implies $\mu_{\Psi_1}(\overline{s}) = \mu_{\Psi_2}(\overline{s}) = 1$, so by induction, $\mathcal{M}_{\overline{\pi}}^{\overline{s}} \models \Psi_1$ and $\mathcal{M}_{\overline{\pi}}^{\overline{s}} \models \Psi_2$ which implies $\mathcal{M}_{\overline{\pi}}^{\overline{s}} \models \Psi$.
- If $\Psi = \mathbb{P}_{\geq p}[\psi]$, then $\mu_\Psi(\overline{s}) = 1$ implies $\nu_\psi(\overline{s}) \geq p$. By induction, $\mathbb{P}_{\overline{\pi}}(\psi|\overline{s}) \geq \nu_\psi(\overline{s}) \geq p$ so $\mathcal{M}_{\overline{\pi}}^{\overline{s}} \models \Psi$.

We now consider a path formula $\psi = \Psi_1 \mathbf{W}_{\mathbf{c}} \Psi_2$, and show by induction over $n \in \mathbb{N}$ that with

$$A_n(\overline{s}) = \{\overline{\xi},\ \overline{\xi}_0 = \overline{s},\ \overline{\xi} \models (\mu_{\Psi_1} = 1) \mathbf{W}_{\mathbf{c}}^{\leq n} (\mu_{\Psi_2} = 1)\},$$

any reachable state $\overline{s}$ satisfies

$$\nu_\psi(\overline{s}) \leq \mathbb{P}(A_n(\overline{s})).$$

Where for an augmented path $\bar{\xi}$, we denote $\bar{\xi} \models (\mu_{\Psi_1} = 1) \mathbf{W}_{\mathbf{c}}^{\leq n} (\mu_{\Psi_2} = 1)$ if and only with $\bar{\xi} = \bar{s}_0 \bar{s}_1 \ldots$, $\bar{s}_k = (s_k, \boldsymbol{\mu}_k, \boldsymbol{\nu}_k)$, either for all $k \leq n$, $(\boldsymbol{\mu}_k)_{\Psi_1} = 1$, or there exists $n_0 \leq n$ such that for all $k \leq n_0$, $(\boldsymbol{\mu}_k)_{\Psi_1} = 1$ and $(\boldsymbol{\mu}_{n_0})_{\Psi_1} = (\boldsymbol{\mu}_{n_0})_{\Psi_2} = 1$.

First, $\mathbb{P}(A_{-1}(\overline{s})) = 1$ for any $\overline{s}$ so the property holds. Now, we consider a reachable $\overline{s}$.

- If $\mu_{\Psi_1}(\overline{s}) = 0$, $\nu_\psi(\overline{s}) = 0$ by state compatibility so $\nu_\psi(\overline{s}) \leq \mathbb{P}(A_n(\overline{s}))$.
- If $\mu_{\Psi_1}(\overline{s}) = \mu_{\Psi_2}(\overline{s}) = 1$, then $\mathbb{P}(A_n(\overline{s})) = 1$ so $\nu_\psi(\overline{s}) \leq \mathbb{P}(A_n(\overline{s}))$.
- If $\mu_{\Psi_1}(\overline{s}) = 1$ and $\mu_{\Psi_2}(\overline{s}) = 0$, then $\overline{\xi} \in A_{n+1}(\overline{s})$ if and only if there exists $\overline{s}'$ such that $\overline{\xi}_1 = \overline{s}'$ and $\overline{\xi}_{\geq 1} \in A_n(\overline{s}')$. Hence,

$$\mathbb{P}(A_{n+1}(\overline{s})) = \sum_{i=1}^{m} \alpha_i \sum_{a \in A(s)} \delta_i(a) \sum_{s' \in S} P(s'|a) \mathbb{P}(A_n(\overline{s}')),$$

where $\overline{s}' = (s', \Theta_i(a))$. By induction over n, $\nu_\psi(\overline{s}') \leq \mathbb{P}(A_n(\overline{s}'))$, so

$$\nu_\psi(\overline{s}) \leq \mathbb{P}(A_{n+1}(\overline{s}')).$$

Now, with

$$B_n(\overline{s}) = \{\overline{\xi},\ \overline{\xi}_0 = \overline{s},\ \overline{\xi} \models \Psi_1 \mathbf{W}_{\mathbf{c}}^{\leq n} \Psi_2\},$$

Any path in $A_n(\overline{s})$ is also in $B_n(\overline{s})$, as $\mu_{\Psi_j}(\overline{s}') = 1 \Rightarrow \mathcal{M}_{\overline{\pi}}^{\overline{s}'} \models \Psi_j$ for $j = 1, 2$. Since for any n, $A_{n+1}(\overline{s}) \subseteq A_n(\overline{s})$ and $B_{n+1}(\overline{s}) \subseteq B_n(\overline{s})$, we obtain

$$\mathbb{P}_{\overline{\pi}}(\psi|\overline{s}) = \mathbb{P}\left(\cap_{k=1}^{\infty} B_k(\overline{s})\right) = \lim_{k \to \infty} \mathbb{P}(B_k(\overline{s})),$$

and

$$\nu_\psi(\overline{s}) \leq \lim_{k \to \infty} A_k(\overline{s}) \leq \mathbb{P}_{\overline{\pi}}(\psi|\overline{s}).$$

□

4.2 Implementation

We propose an implementation of SC^2 for discrete environments that follows Algorithm 1. We rely on Value Iteration for computing the inductive frontier () in step 1 and on PPO () to optimize the MDP in step 2. In PPO, the policy of the agent is represented by a neural network π_θ called the *actor*. The neural network π_θ outputs a probability distribution, from which we sample to obtain an action (δ, Λ) in $\hat{\mathcal{M}}_V$. We simulate (δ, Λ) using the original MDP $\mathcal{M}$. More precisely, we sample from δ to obtain an augmented action (a, Θ) with $\Theta = \Lambda(a)$. In turn, to simulate the augmented action (a, Θ), we step with the action a in the original $\mathcal{M}$ from state s. This leads the agent to a state s' of the original MDP, that we augment with $\Theta(s')$ to obtain the next state $(s', \Theta(s'))$ of $\hat{\mathcal{M}}_V$. That pipeline guarantees safety *in practice*, as Theorem 7 applies to *any* policy used during training.

To obtain an action of $\hat{\mathcal{M}}_V$ from sampling from the actor's output distribution, we interpolate with an action already in $\hat{\mathcal{M}}_V$. More precisely, in standard implementations of PPO (e.g. Stable Baselines 3[2]), and other actor-critic policy optimization algorithms (like e.g. SAC (Haarnoja et al. 2018)), the actor can output after sampling either discrete actions, or continuous actions in $\mathbb{R}^n$. We follow that convention in our implementation. Thus, we obtain straightforwardly from sampling the actor's output distribution $\pi_\theta(s)$ a couple (δ, Λ), where δ is a probability distribution over the set of actions $A(s)$ of $\mathcal{M}$, and Λ is a function mapping a couple (a, s') with $a \in A(s)$ and $s' \in \text{succ}(a)$ to a value in $[0; 1]$. We then interpolate (δ, Λ) with an action $(\delta_{safe}, \Lambda_{safe})$ of $\hat{\mathcal{M}}_V$, which always exists by Lemma 2. More precisely, we compute a number $t \in [0; 1]$ such that $(t\delta + (1-t)\delta_{safe}, t\Lambda + (1-t)\Lambda_{safe})$ is in the action space of $\hat{\mathcal{M}}_V$, and step with that action in $\hat{\mathcal{M}}_V$.

5 Experiments

We demonstrate the viability of SC^2 on four environments from the benchmark suite of the MASA-Safe-RL library (Goodall et al. 2025), and learn a reward-maximizing policy under a CPCTL constraint. Our implementation is based on the MASA-Safe-RL library, and we use the JAX-based implementation of PPO (Schulman et al. 2017) provided with its standard parameters. Experiments were run on Windows 10 on an Intel Core i7-12650H CPU. For each environment, we report the episodic reward achieved by the agent, averaged in Figure 4. The episodic reward on each environment is averaged on three different seeds, except for the Pacman environment where five seeds are used. By construction, the policy, even during training, satisfies the CPCTL constraint, as shown in the previous section.

Across all environments, a subset of states is labelled `unsafe`. We evaluate two instances of the nested CPCTL constraint

$$\Phi_q = \mathbb{P}_{\geq q}(\mathbf{G}\, \mathbb{P}_{\geq 0.6}(\mathbf{G}\, \neg \texttt{unsafe}))\,, \qquad q \in \{0.5, 0.9\}.$$

The inner predicate $\mathbb{P}_{\geq 0.6}(\mathbf{G}\, \neg \texttt{unsafe})$ expresses that the current state offers a sufficiently high infinite-horizon safety prospect. States that do not satisfy this predicate can therefore be read as "low-safety-prospect" situations—states where, even if the controller keeps operating as intended, the probability of avoiding unsafe states indefinitely is too low, and where, in an applied setting, one might prefer a fallback such as human takeover. The outer probability operator in Φ_q then enforces a reliability guarantee: with probability at least q, the execution always remains in states where human intervention is not required. We instantiate $q \in \{0.5, 0.9\}$ to showcase how tightening this reliability requirement restricts behaviour and impacts achievable reward, depending on the environment.

5.1 Environments

We evaluate SC^2 on four benchmark environments (three gridworlds and one buffer-control task) of the MASA benchmark suite.

[2]https://github.com/DLR-RM/stable-baselines3

Media streaming. The agent controls a buffer of capacity 20. Packets depart according to a Bernoulli process with rate $\mu_{out} = 0.7$. The action space is $A = \{\textit{fast}, \textit{slow}\}$, where arrivals follow Bernoulli rates $\mu_{\textit{fast}} = 0.9$ and $\mu_{\textit{slow}} = 0.1$, respectively. The reward penalizes outages: the agent receives -1 when the buffer is empty and 0 otherwise. The state is augmented with a counter cost c tracking how often *fast* has been selected. Unsafe states are those with $c > C$, where $C = \lfloor \texttt{episode\_length}/2 \rfloor$, capturing an upper bound on aggressive transmission with high probability.

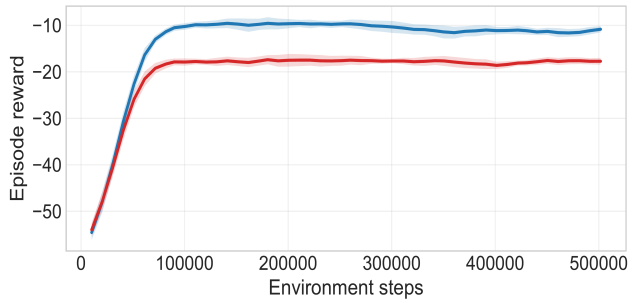
Colour bomb gridworld. This task is in a 15×15 gridworld. The agent chooses among four moves $\{\textit{left}, \textit{right}, \textit{up}, \textit{down}\}$ in every non-terminal state. Actions are noisy: with probability $1 - \texttt{random\_action\_probability}$ the intended move is executed, while with probability `random_action_probability` a different move is sampled uniformly from the remaining three directions (e.g., *left* is replaced by *right*, *up* or *down* with probability $\texttt{random\_action\_probability}/3$ each). Entering a yellow, blue, or pink region yields reward $+1$ and ends the episode. Upon entering a green or red region, the agent may either remain within the region or exit to any adjacent white cell; all other transitions yield reward 0. Bomb cells are unsafe.

Mini-Pacman and Pacman We also consider a 8×10 mini-pacman and a 15×19 pacman environment, with one ghost, and collectible coins (+1 reward) in every position. By ignoring the dynamics of the coins, we can leverage a *safety abstraction* of the environment for efficient Value Iteration. We note that even with the safety abstraction, the total number of states of the 15×19 exceeds 100k, demonstrating that our approach is still feasible for large state spaces.

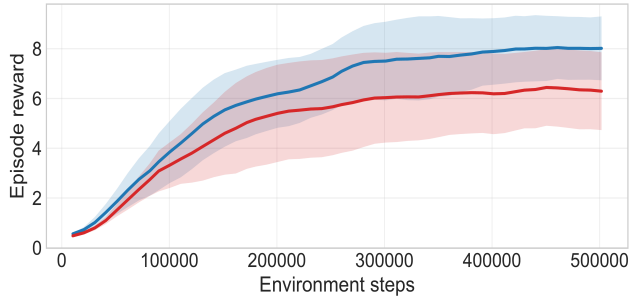
6 Conclusion

We introduced a new approach to Safe Reinforcement Learning in which safety requirements are specified in CPCTL, a fragment of PCTL that strictly generalizes classical avoidance and multi-objective avoidance constraints. Our contribution is twofold. On the theoretical side, we characterize CPCTL-constrained control via an extended Bellman operator over profiles of valuations and counters, and show how these can be propagated through an appropriate augmentation of the MDP. On the algorithmic side, we design SC^2, a shielding-based method that enforces CPCTL constraints at all times, including during exploration, while remaining optimal among safe policies.

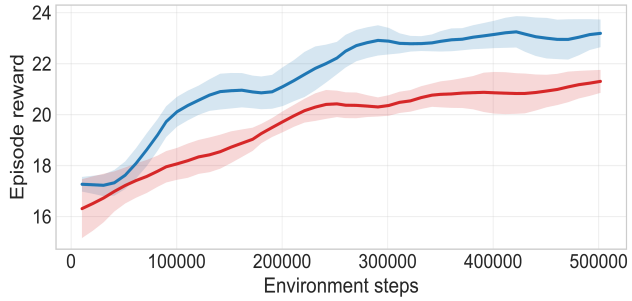
A central step is the reduction of the constrained optimization problem in the original MDP to an unconstrained one in the augmented MDP $\hat{\mathcal{M}}_V$, where only V-shielded distributions are permitted. We show that memory-less policies in $\hat{\mathcal{M}}_V$ correspond to history-dependent policies in the original MDP that satisfy the CPCTL specification and achieve the same expected return. Consequently, any policy produced by SC^2 is provably safe, and optimal among all safe policies.



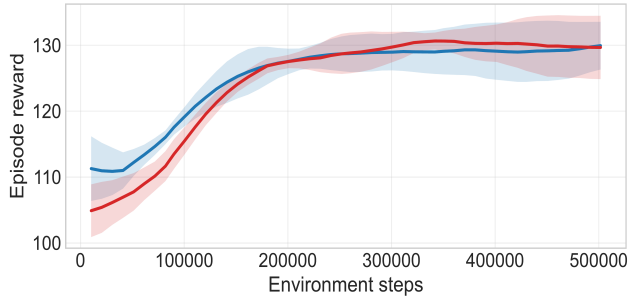
(a) Media Streaming



(b) Colour Bomb Gridworld



(c) Mini Pacman



(d) Pacman

Figure 4: Episodic rewards incurred by the agent during training in the four environments. Blue corresponds to constraint $\Phi_{0.9}$ and red to constraint $\Phi_{0.5}$.

Overall, our results demonstrate that reinforcement learning agents can satisfy expressive temporal safety specifications with formal guarantees, without sacrificing reward optimality.

Acknowledgments

The research presented in this paper was supported by the EPSRC grant number EP/X015823/1, ”An abstraction-based technique for Safe Reinforcement Learning”.

AI Declaration

Generative AI tools have been used to help with wording, grammar, spelling, and routine coding support. They have also been used to detect latex mistakes. These tools were not used to generate the scientific ideas, theoretical results, proofs, or experimental design of the paper.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1805.05800*.

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2017. Safe reinforcement learning via shielding. *CoRR* abs/1708.08611.

Baier, C., and Katoen, J.-P. 2008a. *Principles of Model Checking*, volume 26202649.

Baier, C., and Katoen, J.-P. 2008b. *Principles of model checking*. MIT press.

Bertsekas, D. P., and Shreve, S. E. 2007. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific.

Ciesinski, F., and Größer, M. 2004. *On Probabilistic Computation Tree Logic*. Berlin, Heidelberg: Springer Berlin Heidelberg. 147–188.

Court, E. H.-D. l.; Belardinelli, F.; and Goodall, A. W. 2025. Probabilistic shielding for safe reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 39(15):16091–16099.

Garcıa, J., and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16(1):1437–1480.

Goodall, A. W., and Belardinelli, F. 2023. Approximate model-based shielding for safe reinforcement learning. In Gal, K.; Nowé, A.; Nalepa, G. J.; Fairstein, R.; and Radulescu, R., eds., *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 883–890. IOS Press.

Goodall, A. W., and Belardinelli, F. 2024. Leveraging approximate model-based shielding for probabilistic safety guarantees in continuous environments. 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*, 2291–2293. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Goodall, A. W.; Adalat, O.; Hamel De-le Court, E.; and Belardinelli, F. 2025. MASA-Safe-RL: Multi and single agent

safe reinforcement learning. https://github.com/sacktock/MASA-Safe-RL/. GitHub repository.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR* abs/1801.01290.

Halmos, P. R. 2013. *Measure theory*, volume 18. Springer.

Jansen, N.; Könighofer, B.; Junges, S.; Serban, A.; and Bloem, R. 2020. Safe Reinforcement Learning Using Probabilistic Shields. In Konnov, I., and Kovács, L., eds., *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 3:1–3:16. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Kendall, A.; Hawke, J.; Janz, D.; Mazur, P.; Reda, D.; Allen, J.; Lam, V.; Bewley, A.; and Shah, A. 2019. Learning to drive in a day. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, 8248–8254. IEEE.

Kober, J.; Bagnell, J.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32:1238–1274.

le Court, E. H.; Belardinelli, F.; and Goodall, A. W. 2025. Probabilistic shielding for safe reinforcement learning. In Walsh, T.; Shah, J.; and Kolter, Z., eds., *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, 16091–16099. AAAI Press.

le Court, E. H.-D.; Ohlmann, G.; and Belardinelli, F. 2025. Prosh: Probabilistic shielding for model-free reinforcement learning.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Ohlmann, G.; le Court, E. H.-D.; and Belardinelli, F. 2025. Synthesis of safety specifications for probabilistic systems.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA: John Wiley & Sons, Inc., 1st edition.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking safe exploration in deep reinforcement learning. In *arXiv preprint arXiv:1910.01708*.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48:67–113.

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.; Wang, J.; and Bou-Ammar, H. 2022. Saute rl: Almost surely safe reinforcement learning using state augmentation.

Stooke, A.; Achiam, J.; and Abbeel, P. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In *ICML*.

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT press, 2nd edition.

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.