ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230357

Approximate Model-Based Shielding for Safe Reinforcement Learning

Alexander W. Goodall[a;*] and Francesco Belardinelli[a]

[a]Imperial College London

Abstract. Reinforcement learning (RL) has shown great potential for solving complex tasks in a variety of domains. However, applying RL to safety-critical systems in the real-world is not easy as many algorithms are sample-inefficient and maximising the standard RL objective comes with no guarantees on worst-case performance. In this paper we propose approximate model-based shielding (AMBS), a principled look-ahead shielding algorithm for verifying the performance of learned RL policies w.r.t. a set of given safety constraints. Our algorithm differs from other shielding approaches in that it does not require prior knowledge of the safety-relevant dynamics of the system. We provide a strong theoretical justification for AMBS and demonstrate superior performance to other safety-aware approaches on a set of Atari games with state-dependent safety-labels.

1 Introduction

Due to the inefficiencies of deep reinforcement learning (RL) and lack of guarantees, safe RL [5] has emerged as an increasingly active area of research. Ensuring the safety of RL agents is crucial for their widespread adoption in safety-critical applications where the cost of failure is high. From formal verification, *Shielding* [3] has been developed as an approach for safe RL, that comes with strong safety guarantees on the agents performance. However, classical shielding approaches make quite restrictive assumptions which limit their capabilities in real-world and high-dimensional tasks. To this end, we propose approximate model-based shielding (AMBS) to address these limitations and obtain a more general and widely applicable algorithm.

The constrained Markov decision process (CMDP) [4] is a popular way of framing the safe RL paradigm. In addition to maximising reward, the agent must satisfy a set of safety-constraints encoded as a cost function that penalises unsafe actions or states. In effect, this formulation constrains the set of feasible policies, which results in a tricky non-smooth optimisation problem. In high-dimensional settings, several model-free methods have been proposed based on trust-region methods [1, 43, 29] or Lagrangian relaxations of the constrained optimisation problem [35, 10]. Many of these methods rely on the assumption of convergence to obtain any guarantees.

Furthermore, model-based approaches to safe RL have gained increasing traction, in part due to recent significant developments in model-based RL (MBRL) [17, 20, 21] and the superior sample complexity of model-based approaches [18, 26]. In addition to learning a reward maximising policy, MBRL learns an approximate dynamics model of the system. Approximating the dynamics using Gaussian process (GP) regression [40] or ensembles of neural networks are both principled ways to quantify uncertainty, and develop risk- and safety-aware policy optimisation algorithms [39, 31, 6] and model predictive control (MPC) schemes [30].

Shielding for RL was introduced in [3] as a correct by construction reactive system that forces hard constraints on the learned policy, preventing it from entering unsafe states determined by a given temporal logic formula. Classical approaches to shielding [3] require *a priori* access to a safety-relevant abstraction of the environment, where the dynamics are known or at least conservatively estimated. This can be quite a restrictive assumption in many real-world scenarios, where such an abstraction is not known or too complex to represent. However, these strong assumptions do come with the benefit of strong guarantees (i.e., *correctness* and *minimal-interference*).

In this paper we operate in a less restrictive domain and only assume that there exists some expert labelling of the states; we claim that this is a more realistic paradigm which has been studied in previous work [23, 15]. As a result, our approach is more broadly applicable to a variety of environments, including high-dimensional environments which have been rarely studied in the shielding literature [33]. We do unfortunately loose the strict formal guarantees obtained by classical shielding, although we develop tight probabilistic bounds and a strong theoretical basis for our method.

Our approach, AMBS, is inspired by *latent shielding* [23] an approximate shielding algorithm that verifies policies in the latent space of a world model [18, 20, 21] and is closely related to previous work on *approximate shielding of Atari agents* [15]. The core idea of our approach is that we remove the requirement for a suitable safety-relevant abstraction of the environment by learning an approximate dynamics model of the environment. With the learned dynamics model we can simulate possible future trajectories under the learned policy and estimate the probability of committing a safety-violation in the near future. In contrast to previous approaches, we provide a stronger theoretical justification for utilising world models as a suitable dynamics model. And while we leverage DreamerV3 [21] as our stand-in dynamics model, we propose a more general purpose and model-agnostic framework for approximate shielding.

Contributions. Our main contributions are as follows: (1) we formalise the general AMBS framework which captures previous work on latent shielding [23] and shielding Atari agents [15]. (2) We formalise notions such as *bounded safety* [14] in Probabilistic Computation-tree Logic (PCTL), a principled specification language for reasoning about the temporal properties of discrete stochastic systems. (3) We provide PAC-style probabilistic bounds on the probabil-

* Corresponding Author. Email: a.goodall22@imperial.ac.uk

ity of accurately estimating a constraint violation with both the 'true' dynamics model and a learned approximation. (4) We more rigorously develop the theory from [15] and theoretically justify the use of world models as the stand-in dynamics model. (5) We provide a much richer set of results on a small set of Atari games with state-dependent safety-labels and we demonstrate that AMBS significantly reduces the cumulative safety-violations during training compared to other safety-aware approaches. And in some cases AMBS also vastly improves the learned policy w.r.t. episode return. All technical proofs are provided in full in the supplementary material [44], along with extended results (learning curves) and implementation details, including hyperparameters and access to code. These materials are provided online at https://github.com/sacktock/AMBS.

2 Preliminaries

In this section we introduce the relevant background material and notation required to understand our approach and the main theoretical results of the paper. We start by formalising the problem setup and the specification language used to define bounded safety. We then introduce prior look-ahead shielding approaches and world models.

2.1 Problem Setup

In our experiments we evaluate our approach on Atari games provided by the Arcade Learning Environment (ALE) [8]. Atari games are partially observable, as the agent is only given raw pixel data and does not have access to the underlying memory buffer of the emulator. This means one observation can map to many underlying states by a probability distribution. Therefore, we model the problem as a partially observable markov decision process (POMDP) [34].

To capture state-dependent safety-labels, we extend the POMDP tuple with a set of atomic propositions and a labelling function; a common formulation used in [7]. Formally, we define a POMDP as a 10-tuple $\mathcal{M} = (S, A, p, \iota_{init}, R, \gamma, \Omega, O, AP, L)$, where, S is a finite set of *states*, A is a finite set of *actions*, $p : S \times A \times S \to [0, 1]$ is the *transition function*, where $p(s' \mid s, a)$ is the probability of transitioning to state s' by taking action a in state s, $\iota_{init} : S \to [0, 1]$ is the *initial state distribution*, where $\iota_{init}(s)$ is the probability of starting in state s, $R : S \times A \to \mathbb{R}$ is the *reward function*, where $R(s, a)$ is the immediate reward received for taking action a in state s, $\gamma \in (0, 1]$ is the discount factor, Ω is a finite set of *observations*, $O : S \times \Omega \to [0, 1]$ is the *observation function*, where $O(o \mid s)$ is the probability of observing o in state s, AP is a finite set of *atomic propositions* (or atoms), $L : S \to 2^{AP}$ is the *labeling function*, where $L(s)$ is the set of atoms that hold in state s.

We note here that MDPs are a special case of POMDPs without partial observability. For example, we can suppose that in MDPs, $\Omega = S$ and the observation function $O : S \times \Omega \to [0, 1]$ collapses to the identity relation.

In addition, we are given a propositional safety-formula Ψ that encodes the safety-constraints of the environment. For example, a simple Ψ might look like,

$$\Psi = \neg\textbf{collision} \wedge (\textbf{red-light} \Rightarrow \textbf{stop})$$

which says "don't have a collision and stop when there is a red light", for $AP = \{\textbf{collision}, \textbf{red-light}, \textbf{stop}\}$.

At each timestep t, the agent receives an observation o_t, reward r_t and a set $L(s_t)$ of labels. The agent can then determine that s_t satisfies the safety-formula Ψ by applying the following relation,

$$s \models a \text{ iff } a \in L(s)$$
$$s \models \neg\Psi \text{ iff } s \not\models \Psi$$
$$s \models \Psi_1 \wedge \Psi_2 \text{ iff } s \models \Psi_1 \text{ and } s \models \Psi_2$$

where $a \in AP$ is an atomic proposition, negation ($\neg$) and conjunction ($\wedge$) are the familiar logical operators from propositional logic. The goal is to find a policy π that maximises reward, that is $\pi^* = \arg\max_\pi \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))]$, while minimising the cumulative number of violations of the safety-formula Ψ during training and deployment.

2.2 Probabilistic Computation-tree Logic

Probabilistic Computation Tree Logic (PCTL) is a branching-time temporal logic based on CTL that allows for probabilistic quantification along path formula [7]. PCTL is a convenient way of stating soft reachability and safety properties of discrete stochastic systems which makes it a commonly used specification language for probabilistic model checkers. A well-formed PCTL formula can be constructed with the following grammar,

$$\Phi ::= a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathbb{P}_J(\phi)$$
$$\phi ::= X\Phi \mid \Phi U \Phi \mid \Phi U^{\leq n} \Phi$$

where $J \subset [0, 1]$, $J \neq \varnothing$ is a non-empty subset of the unit interval, and next X, until U, and bounded until $U^{\leq n}$ are the temporal operators from CTL [7]. We make the distinction here between state formula Φ and path formula ϕ which are interpreted over states and paths respectively.

For state formula Φ, we write $s \models \Phi$ to indicate that the state $s \in S$ satisfies the state formula Φ, where the satisfaction relation is defined in a similar way as before (Section 2.1), but also includes probabilistic quantification, see [7] for details. On the other hand, path formula ϕ are interpreted over sequences of states or *traces*. In the following section we provide the satisfaction relation for path formula ϕ for the fragment of PCTL that we use. We also note that the common temporal operators eventually $\Diamond$ and always $\Box$, and their bounded counterparts $\Diamond^{\leq n}$ and $\Box^{\leq n}$ can be derived from the grammar above in a straightforward way, see [7] for details.

2.3 Bounded Safety

Now we are equipped to formalise the notion of *bounded safety* introduced in [14]. Consider a fixed (stochastic) policy $\pi : O \times A \to [0, 1]$ and a POMDP $\mathcal{M} = (S, A, p, \iota_{init}, R, \gamma, \Omega, O, AP, L)$. Together π and $\mathcal{M}$ define a transition system $\mathcal{T} : S \times S \to [0, 1]$ on the set of states S, where $\sum_{s' \in S} \mathcal{T}(s' \mid s) = 1$. A *finite trace* of the transition system $\mathcal{T}$ with length n (n transitions) is a sequence of states $s_0 \to s_1 \to \ldots \to s_n$ denoted by τ, where each state transition is induced by the transition probabilities of $\mathcal{T}$. Let $\tau[i]$ denote the i^{th} state of τ. A trace τ is said to satisfy bounded safety if all of its states satisfy the given propositional safety-formula Ψ, that is,

$$\tau \models \Box^{\leq n}\Psi \quad \text{iff} \quad \forall i \; 0 \leq i \leq n, \tau[i] \models \Psi$$

for some bounded look-ahead parameter n. In words, the path formula $\phi = \Box^{\leq n}\Psi$ says "always satisfy Ψ in the next n steps". Now in PCTL we say that a given state $s \in S$ satisfies Δ-bounded safety or formally $s \models \mathbb{P}_{\geq 1-\Delta}(\Box^{\leq n}\Psi)$ iff,

$$\mu_s(\{\tau \mid \tau[0] = s, \forall i \; 0 \leq i \leq n, \tau[i] \models \Psi\}) \in [1 - \Delta, 1] \quad (1)$$

where μ_s is a well-defined probability measure (induced by the transition system $\mathcal{T}$) on the set of traces starting from s and with finite length n, see [7] for additional details on the semantics of PCTL.

Throughout the rest of the paper we will denote $\mu_{s\models\phi}$ as shorthand for the measure $\mu_s(\{\tau \mid \tau[0] = s, \forall i \; 0 \leq i \leq n, \tau[i] \models \Psi\})$, where $\phi = \square^{\leq n}\Psi$ is the path formula that corresponds to bounded safety. By grounding bounded safety in PCTL we obtain a principled way to trade-off safety and exploration with the Δ parameter. Since strictly enforcing bounded safety can lead to overly conservative behaviour during training, which is not always desirable if we are to make progress toward the optimal policy.

2.4 Look-ahead Shielding

The general principle of look-ahead shielding is as follows, from the current state we compute the set of possible future paths and check if they incur any unsafe states [14]. If the agent proposes an action that leads the agent down a path with an unavoidable unsafe state, then the look-ahead shielding procedure should override the agent's action with an action leading down a safe path instead, see Fig. 1.

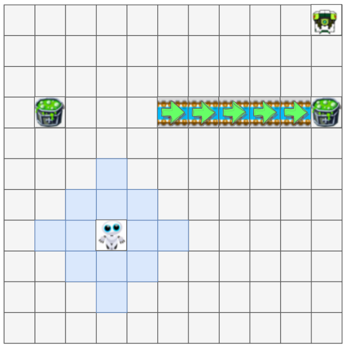

Figure 1. With a Manhattan distance look-ahead of 2 (blue squares) the robot can avoid the *left-most* vat of acid, but is unable to determine that the conveyor belt leads to an unavoidable unsafe state. Without a further look-ahead horizon, the robot is doomed to fall in the *right-most* vat of acid during exploration. *This image was created with the assistance of DALL·E 2.*

Bounded Prescience Shielding (BPS) [14] is an approach to shielding Atari agents that assumes access to a black-box simulator of the environment, which can be queried for look-ahead shielding. In the worst case BPS must enumerate all paths of length H from the current state, to find a safe path for the agent to follow. This of course comes with a substantial computational overhead, even with a relatively small horizon of $H = 5$.

Our approach most closely resembles *latent shielding*, [23] which rolls out learned policies in the latent space of a *world model* [17, 18] that approximates the 'true' dynamics of the environment for look-ahead shielding. By simulating futures in a low-level latent space rather than with a high-fidelity simulator, latent shielding can be used with a much larger look-ahead horizon ($H = 15$) and can be deployed during training. As a result we can hope to detect unavoidable unsafe states that require a larger look-ahead horizon to avoid.

Unfortunately *latent shielding* [23] is not the be all and end all and is still fairly short-sighted in comparison to our approach. Rolling out dynamics models for long horizons is impractical as we will quickly run into accumulating errors that arise from the approximation error of the learned model. Instead we use *safety critics* for bootstrapping the end of simulated trajectories, to obtain further look-ahead capabilities without running into accumulating errors.

2.5 World Models

World models were first introduced in the titular paper [17]. Built on the recurrent state space model (RSSM) [19], world models are trained to learn a compact latent representation of the environment state and dynamics. Once the dynamics are learnt the world model can be used to 'imagine' possible future trajectories.

We leverage DreamerV3 [21] as our stand-in dynamics model for look-ahead shielding and policy optimisation. DreamerV3 consists of the following components: (1) the image encoder $z_t \sim q_\theta(z_t \mid o_t, h_t)$ that learns a posterior latent representation given the observation o_t and recurrent state h_t. (2) The recurrent model $h_t = f_\theta(h_{t-1}, z_{t-1}, a_{t-1})$, which computes the next deterministic latents given the past state $s_{t-1} = (h_{t-1}, z_{t-1})$ and action a_{t-1}. (3) The transition predictor $\hat{z}_t \sim p_\theta(\hat{z}_t \mid h_t)$, which is used as the prior distribution over the stochastic latents (without o_t). (4) The image decoder $\hat{o}_t \sim p_\theta(\hat{o}_t \mid h_t, z_t)$ which is used to provide high quality image gradients by minimising reconstruction loss. (5) The prediction heads $\hat{r}_t \sim p_\theta(\hat{r}_t \mid h_t, z_t)$ and $\hat{\gamma}_t \sim p_\theta(\hat{\gamma}_t \mid h_t, z_t)$ trained to predict reward signals and episode termination respectively.

In DreamerV3 the latents represent categorical variables and the reward and image prediction heads both parameterise a *twohot symlog* distributions [21]. All components are implemented as neural networks and optimised with straight through gradients. In addition, KL-balancing [20] and free-bits [18] are used to regularise the prior and posterior representations and prevent a degenerate latent space representation.

Policy optimisation is performed entirely on experience 'imagined' in the latent space of the world model. We refer to π^{task} as the task policy trained in the world model to maximise expected discounted accumulated reward, that is, optimise the following,

$$\max \mathbb{E}_{\pi^{\text{task}}, p_\theta} \left[\sum_{t=1}^{\infty} \hat{\gamma}^{t-1} \hat{r}_t \right] \tag{2}$$

In addition to the task policy π^{task}, a task critic denoted v^{task}, is used to construct TD-λ targets [37] which are used to help guide the task policy π^{task} in a TD-λ actor-critic style algorithm. We refer the reader to [21] for more precise details.

3 Approximate Model-Based Shielding

In this section we present the AMBS framework, our novel method for look-ahead shielding RL policies with a learned dynamics model. We start by giving an overview of the approach, followed by a strong theoretical basis for AMBS. Specifically, we derive PAC-style probabilistic bounds on estimating the probability of a constraint violation under both the 'true' transition system and a learned approximation. We show these hold under reasonable assumptions and we demonstrate when these assumptions hold in specific settings.

3.1 Overview

AMBS is split into two main phases: the *learning phase*, which includes (i) dynamics learning, (ii) task policy optimisation with the standard RL objective, and (iii) safe policy learning and safety critic optimisation, and the *environment interaction phase*, which consists of collecting experience from the environment with the shielded policy. The new experience is then used to improve the dynamics model.

To obtain a shielded policy, recall that we are interested in verifying PCTL formulas of the form $\Phi = \mathbb{P}_{\geq 1-\Delta}(\square^{\leq n}\Psi)$ (Δ-bounded safety) on the transition system $\mathcal{T} : S \times S \to [0, 1]$ induced by fixing the task policy π^{task} for a given POMDP $\mathcal{M}$, where Ψ is the propositional safety-formula. Quite simply, the shielded policy picks actions according to π^{task} provided Δ-bounded safety is satisfied, otherwise the shielded policy picks actions with a backup policy that should prevent the agent from committing the possible safety-violation.

To check Δ-bounded safety, we can simulate the transition system $\mathcal{T}$ by rolling-out the learned world model p_θ with actions sampled from the task policy π^{task}; effectively obtaining an approximate transition system $\widehat{\mathcal{T}} : S \times S \to [0, 1]$. Even with access to the 'true' transition system $\mathcal{T}$, exact PCTL model checking is $O(\text{poly}(size(\mathcal{T})) \cdot n \cdot |\Phi|)$ which is too big for high-dimensional settings. Rather, we rely on Monte-Carlo sampling from the approximate transition system $\widehat{\mathcal{T}}$ to obtain an estimate $\tilde{\mu}_{s \models \phi}$ of the measure $\mu_{s \models \phi}$, which specifies the probability of satisfying $\phi = \Box^{\leq n}\Psi$ (bounded safety) from s, under the 'true' transition system $\mathcal{T}$.

3.2 Fully Observable Setting

We start by considering the fully observable setting, that is we are concerned with checking PCTL formula of the form $\Phi = \mathbb{P}_{\geq 1-\Delta}(\Box^{\leq n}\Psi)$ (Δ-bounded safety) on the transition system $\mathcal{T} : S \times S \to [0, 1]$ induced by fixing some given policy π in an MDP $\mathcal{M} = (S, A, p, \iota_{init}, R, \gamma, AP, L)$. We state the following result,

Theorem 1. *Let $\epsilon > 0$, $\delta > 0$, $s \in S$ be given. With access to the 'true' transition system $\mathcal{T}$, with probability $1 - \delta$ we can obtain an ϵ-approximate estimate of the measure $\mu_{s \models \phi}$, by sampling m traces $\tau \sim \mathcal{T}$, provided that,*

$$m \geq \frac{1}{2\epsilon^2} \log\left(\frac{2}{\delta}\right) \tag{3}$$

Proof. The proof is an application of Hoeffding's inequality [25]. See supplementary material [44] for details. □

Suppose that we only have access to an approximate MDP, where the transition function p is estimated with a learned approximation $\hat{p}$, that is, $\widehat{\mathcal{M}} = (S, A, \hat{p}, \iota_{init}, R, \gamma, AP, L)$. As earlier, π and $\widehat{\mathcal{M}}$ define an approximate transition system $\widehat{\mathcal{T}}$. We show that we can obtain an estimate $\tilde{\mu}_{s \models \phi}$ for $\mu_{s \models \phi}$ by sampling traces from $\widehat{\mathcal{T}}$.

Theorem 2. *Let $\epsilon > 0$, $\delta > 0$ be given. Suppose that for all $s \in S$, the total variation (TV) distance between $\mathcal{T}(s' \mid s)$[1] and $\widehat{\mathcal{T}}(s' \mid s)$ is bounded by some $\alpha \leq \epsilon/n$. That is,*

$$D_{TV}\left(\mathcal{T}(s' \mid s), \widehat{\mathcal{T}}(s' \mid s)\right) \leq \alpha \; \forall s \in S \tag{4}$$

Now fix an $s \in S$, with probability $1 - \delta$ we can obtain an ϵ-approximate estimate of the measure $\mu_{s \models \phi}$, by sampling m traces $\tau \sim \widehat{\mathcal{T}}$, provided that,

$$m \geq \frac{2}{\epsilon^2} \log\left(\frac{2}{\delta}\right) \tag{5}$$

Proof. The proof follows from Theorem 1 and the *simulation lemma* [27] adapted to our purposes, see supplementary material [44]. □

The assumption we make in Theorem 2 on the TV distance being bounded for all states $s \in S$ (Eq. 4) is quite strong. In reality we may only require that the TV distance between $\mathcal{T}(s' \mid s)$ and $\widehat{\mathcal{T}}(s' \mid s)$ is bounded for the safety-relevant subset of the state space.

[1] Unless s' is fixed we define $\mathcal{T}(s' \mid s)$ as the distribution of next states s' given s rather than as an explicit probability.

Tabular Case. It may also be interesting to consider under what conditions the TV distance between $\mathcal{T}(s' \mid s)$ and $\widehat{\mathcal{T}}(s' \mid s)$ are bounded for a given state. In the tabular case we can obtain an approximate dynamics model $\hat{p}(s' \mid s, a)$ for the 'true' dynamics $p(s' \mid s, a)$ by using the maximum likelihood principle for categorical variables. Quite simply let,

$$\hat{p}(s' \mid s, a) = \frac{c(s', s, a)}{v(s, a)} \tag{6}$$

where $c(s', s, a)$ is the visit count of s', s, a or equivalently the number of times (s, a) has lead s' and $v(s, a)$ is the visit count of (s, a) or similarly the number of times a has been picked from s.

Theorem 3. *Let $\alpha > 0$, $\delta > 0$, $s \in S$ be given. With probability $1-\delta$ the total variation (TV) distance between $\mathcal{T}(s' \mid s)$ and $\widehat{\mathcal{T}}(s' \mid s)$ is upper bounded by α, provided that all actions $a \in A$ with non-negligable probability $\eta \geq \alpha/(|A||S|)$ (under π) have been picked from s at least m times, where*

$$m \geq \frac{|S|^2}{\alpha^2} \log\left(\frac{2|A||S|}{\delta}\right) \tag{7}$$

Proof. The proof is an application of Hoeffding's inequality [25] to categorical distributions, see supplementary material [44]. □

The strong dependence on $|S|$ here can make this result quite unwieldy. However, for low-rank or low-dimensional MDPs this dependence on $|S|$ can be suitably replaced. For example in *gridworld* environments, where there are only 4 possible next states, $|S|$ can be replaced with 4. Additionally, for deterministic policies we can ignore the dependence on $|A|$.

3.3 Partially Observable Setting

Sample complexity bounds like the one we obtained for the tabular MDP case are a lot harder to obtain for general POMDPs. To say anything meaningful about learning in POMDPs, various assumptions can be made about the structure of the environment [28]. We will reason about the underlying POMDP $\mathcal{M}$ in terms of *belief states*. Belief states infer a distribution over possible underlying states given the history of past observations and actions. Formally they are defined as a filtering distribution $p(s_t \mid o_{t \leq t}, a_{\leq t})$. Importantly, conditioning on the entire history of past observations and actions exposes more information about the possible underlying states.

However, maintaining an explicit distribution over possible states is difficult, particularly when we have no idea what the space of possible states is. Instead it is common to learn a latent representation $b_t = f(o_{t \leq t}, a_{\leq t})$ of the history of past observations and actions that captures the important statistics of the filtering distribution, so that $p(s_t \mid o_{t \leq t}, a_{\leq t}) \approx p(s_t \mid b_t)$. We will denote b_t the belief state, although it is actually a compact latent state representation of $o_{t \leq t}, a_{\leq t}$, rather than a distribution over possible states.

Theorem 4. *Let b_t be a latent representation (belief state) such that $p(s_t \mid o_{t \leq t}, a_{\leq t}) = p(s_t \mid b_t)$. Let the fixed policy $\pi(\cdot \mid b_t)$ be a general probability distribution conditional on belief states b_t. Let f be a generic f-divergence measure (TV or similar). Then the following holds:*

$$D_f(\mathcal{T}(s' \mid b), \widehat{\mathcal{T}}(s' \mid b)) \leq D_f(\mathcal{T}(b' \mid b), \widehat{\mathcal{T}}(b' \mid b)) \tag{8}$$

where $\mathcal{T}$ and $\widehat{\mathcal{T}}$ are the 'true' and approximate transition system respectively, defined now over both states s and belief states b.

Proof. The proof is a direct application of the data-processing inequality [2] and we note similar bounds have been derived in previous work [13]. See supplementary material [44] for details. □

What Theorem 4 says, is that by minimising the divergence between the next belief state b' in the 'true' transition system $\mathcal{T}$ and the approximate transition system $\widehat{\mathcal{T}}$ (given the current belief b), we minimise an upper bound on the divergence between the next underlying state s' in $\mathcal{T}$ and $\widehat{\mathcal{T}}$. This objective is precisely encoded in the world model loss function of DreamerV3 [21]. And so by using DreamerV3 (or similar architectures) we aim to minimise the TV distance between $\mathcal{T}$ and $\widehat{\mathcal{T}}$ in the hope that we may use the results stated earlier in this section.

4 AMBS with DreamerV3

In this section we present our practical implementation of AMBS with DreamerV3 [21]. We start by detailing the important components of the algorithm, before describing the shielding procedure in detail and justifying some of the important algorithmic decisions. An outline for the full procedure is presented in the supplementary material [44].

4.1 Algorithm Components

RSSM with Costs. We augment DreamerV3 [21] with a cost predictor head $\hat{c}_t \sim p_\theta(\hat{c}_t \mid h_t, z_t)$ used to predict state dependent costs. Similar to the reward head, the cost predictor is also implemented as a neural network that parameterises a *twohot symlog* distribution [21]. Targets for the cost predictor are constructed as follows,

$$c_t = \begin{cases} 0, & \text{if } s_t \models \Psi \\ C, & \text{otherwise} \end{cases} \tag{9}$$

where $s_t \models \Psi$ can be determined using the labels $L(s_t)$ provided at each timestep (see Section 2.1), and $C > 0$ is a hyperparameter that determines the cost of violating the propositional safety-formula Ψ. We claim that using a cost function in this way rather than a binary classifier, allows the agent to distribute its uncertainty about a constraint violation over consecutive states, since the predicted cost at a latent state (h_t, z_t) may lie anywhere on the real number line.

Safe Policy. The safe policy denoted π^{safe} is used as the backup policy [22] if we detect that a safety-violation is likely to occur in the near future when following the task policy π^{task}. The goal of the safe policy is to prevent the agent from committing the safety-violation that was 'likely' to happen in the near future.

Unlike classical shielding methods that synthesise a backup policy ahead of time [3], we must learn a suitable backup policy as we have no knowledge of the safety-relevant dynamics of the system a priori. Although, we note that if such a backup policy is available ahead of time, either because it is easy to determine (i.e. breaking in a car) or significant engineering effort has gone into creating a backup policy, then this can be easily integrated into our framework. However, in general we train a separate policy π^{safe} and critic v^{safe} using the same TD-λ actor-critic style algorithm used for the task policy [21]. However, we minimise expected discounted costs and ignore any reward signals entirely and so specifically we optimise the following objective,

$$\min \mathbb{E}_{\pi^{\text{safe}}, p_\theta}\left[\sum_{t=1}^{\infty} \hat{\gamma}^{t-1} \hat{c}_t\right] \tag{10}$$

In theory, by minimising expected costs we should learn a good backup policy, since by construction we only incur a cost > 0 when Ψ is not satisfied.

Safety Critics. Safety critics are used to estimate the expected discounted costs under the state distribution of the task policy. In essence, they give us an idea of how safe a given state $s \in S$ is if we were to follow the task policy π^{task} from that state s. As a result we can use them to bootstrap the end of 'imagined' trajectories for further look-ahead capabilities, without having to roll-out the dynamics model any further. We jointly train two safety critics v_1^C and v_2^C with a TD3-style algorithm [12] to prevent overestimation, which is important for reducing overly conservative behaviour. We also train a separate safety discount predictor head $\hat{\gamma}_t^{\text{safe}} \sim p_\theta(\hat{\gamma}_t^{\text{safe}} \mid h_t, z_t)$ to help with overestimation. Specifically, v_1^C and v_2^C estimate the following quantity,

$$V^C(s) = \mathbb{E}_{\pi^{\text{task}}, p_\theta}\left[\sum_{t=1}^{\infty} (\hat{\gamma}_t^{\text{safe}})^{t-1} \hat{c}_t \,\middle|\, s_0 = s\right] \tag{11}$$

The safety discount head is a binary classifier that predicts safety-violations, we use it to implicitly transform the MDP into one where violating states are terminal, this ensures that the safety critics are always upper bounded by C and any costs incurred after a safety-violation are not counted. We note that this is an important property to maintain for checking bounded safety.

4.2 Shielding Procedure

During environment interaction we deploy our approximate shielding procedure to try and prevent the task policy π^{task} from violating the safety-formula Ψ. Actions that lead to violations in the learned world model p_θ do not count since they are not actually committed in the 'real world'.

Algorithm 1 details the approximate shielding procedure. It takes as input an approximation error ϵ, desired safety level Δ (from the PCTL formula), the current latent state $\hat{s} = (z, h)$, an action a proposed by the task policy, the 'imagination' horizon H, the look-ahead shielding horizon T, and the number of samples m, along with the RSSM components and the learned policies. The shielding procedure works by sampling m traces from the approximate transition system $\widehat{\mathcal{T}}$ (obtained by sampling actions with π^{task} in p_θ), checking whether each trace satisfies bounded safety and returning the proportion of satisfying traces. If the estimate obtained $\tilde{\mu}_{s\models\phi}$ is in the interval $[1 - \Delta + \epsilon, 1]$ then the proposed action a is verified safe, otherwise we pick an action with the safe policy π^{safe} instead.

Proposition 5. *Suppose we have an estimate $\tilde{\mu}_{s\models\phi} \in [\mu_{s\models\phi} - \epsilon, \mu_{s\models\phi} + \epsilon]$, if $\tilde{\mu}_{s\models\phi} \in [1 - \Delta + \epsilon, 1]$ then it must be the case that $\mu_{s\models\phi} \in [1 - \Delta, 1]$ and therefore $s \models \mathbb{P}_{\geq 1-\Delta}(\square^{\leq n}\Psi)$.*

Proof. Suppose $\tilde{\mu}_{s\models\phi} \in [\mu_{s\models\phi} - \epsilon, \mu_{s\models\phi} + \epsilon]$, $\tilde{\mu}_{s\models\phi} \in [1 - \Delta + \epsilon, 1]$ and $\mu_{s\models\phi} \notin [1 - \Delta, 1]$. Then $\tilde{\mu}_{s\models\phi} - \mu_{s\models\phi} > \epsilon$ which contradicts $\tilde{\mu}_{s\models\phi} \in [\mu_{s\models\phi} - \epsilon, \mu_{s\models\phi} + \epsilon]$. This implies that indeed $\mu_{s\models\phi} \in [1 - \Delta, 1]$ and therefore $s \models \mathbb{P}_{\geq 1-\Delta}(\square^{\leq n}\Psi)$ by Eq. 1. □

Proposition 5 justifies checking that the condition $\tilde{\mu}_{s\models\phi} \in [1 - \Delta + \epsilon, 1]$ holds in Algorithm 1. Since if $\tilde{\mu}_{s\models\phi}$ is an ϵ-estimate of $\mu_{s\models\phi}$ then $\tilde{\mu}_{s\models\phi} \in [1 - \Delta + \epsilon, 1]$ necessarily implies that the current state s satisfies Δ-bounded safety. We present one more proposition that is required to understand how we determine (or verify) that a trace satisfies bounded safety.

Algorithm 1 Approximate Model-Based Shielding (AMBS)

Input: approximation error ϵ, desired safety level Δ, current state $\hat{s} = (z, h)$, proposed action a, 'imagination' horizon H, look-ahead shielding horizon T, number of samples m, RSSM p_θ, safe policy π^{safe} and task policy π^{task}.

Output: shielded action a'.

for $i = 1, ..., m$ **do**
  Let $\hat{s}_0 = (h, z)$
  // Sample trace
  From $\hat{s}_0$ play a_0 and sample trace $\tau = \hat{s}_{1:H}$ with π^{task} and p_θ.
  Using $\tau = \hat{s}_{1:H}$ compute sequences $\hat{c}_{1:H}$ and $\hat{\gamma}_{1:H}$ with p_θ.
  // Check trace is satisfying
  $X_i = \mathbb{1}\left[\text{cost}(\tau) < \gamma^{T-1} \cdot C\right]$ using Eq. 12 or Eq. 13.
end for
Let $\tilde{\mu}_{s \models \phi} = \frac{1}{m}\sum_{i=1}^{m} X_i$
If $\tilde{\mu}_{s \models \phi} \in [1 - \Delta + \epsilon, 1]$ **return** $a' = a$ **else return** $a' \sim \pi^{\text{safe}}$

Proposition 6. *Suppose we have a trace τ with cost given by,*

$$cost(\tau) = \sum_{t=1}^{H} (\hat{\gamma}_t)^{t-1} \hat{c}_t \tag{12}$$

Under the 'true' transition system $\mathcal{T}$ if $cost(\tau) < \gamma^{H-1} \cdot C$ then necessarily $\tau \models \square^{\leq H}\Psi$.

Proof. The proof is a straightforward argument. By construction $c_t = C$ if and only if $\tau[t] \not\models \Psi$, therefore $\text{cost}(\tau) < \gamma^{H-1} \cdot C$ implies that $\forall t \ 1 \leq t \leq H$ we have $c_t = 0$, which implies that $\forall t \ 1 \leq t \leq H$ we have $\tau[t] \models \Psi$. $\square$

Furthermore, with safety critics we can compute the cost of a trace in a similar way:

$$\text{cost}(\tau) = \sum_{t=1}^{H-1} (\hat{\gamma}_t)^{t-1} \hat{c}_t + \min\left\{v_1^C(h_H, \hat{z}_t), v_2^C(h_H, \hat{z}_t)\right\} \tag{13}$$

Provided that the safety critics accurately capture the expected discounted costs from beyond the 'imagination' horizon H then we can check bounded safety with a larger horizon $T > H$ and derive something similar to Proposition 6. In this situation it may be clear now why we want the safety critics to be upper bounded by C. Since a state $s \in S$ that satisfies bounded safety could have a value $V^C(s) > \gamma^{T-1} \cdot C$ if two or more violations occurred beyond the horizon T; further in the future than what we care about.

5 Experimental Evaluation

In this section we present the results of running AMBS equipped with DreamerV3 [21] on a set of Atari games with state-dependent safety-labels [14]. First, we detail the corresponding safety-formula for each of the games. We then cover with the training details, followed by the results and a brief discussion.

Atari Games with Safety-labels. We evaluate each agent on a set of five Atari games provided in the ALE [8]. Following [32] we use sticky actions (action repeated with probability 0.25), frame skip ($k = 4$), and provide no life information unless it is explicitly encoded in the safety-formula. We leverage the state-dependent safety-labels provided in prior work [14] to construct the safety-formula, Table 1 outlines each of the Atari environments used in our experiments. For a more precise description of each of the environments we refer the reader to [32]. We opt for this setting, as opposed to more standard safe RL benchmarks like SafetyGym [35], as it has been used in previous work [14, 15] and it provides a richer set of safety-constraints and corresponding behaviours that much be learnt.

Table 1. Atari environments and their corresponding safety-formula

Environment	safety-formula Ψ
Assault	$\neg$**hit** $\wedge$ $\neg$**overheat**
DoubleDunk	$\neg$**out-of-bounds** $\wedge$ $\neg$**shoot-bf-clear**
Enduro	$\neg$**crash-car**
KungFuMaster	$\neg$**loose-life** $\wedge$ $\neg$**energy-loss**
Seaquest	(**surface** $\Rightarrow$ **diver**) $\wedge$ $\neg$**hit** $\wedge$ $\neg$**out-of-oxygen**

Training Details Each agent is trained on a single Nvidia Tesla A30 (24GB RAM) GPU and a 24-core/48 thread Intel Xeon CPU with 256GB RAM. Due to constraints on compute resources, each agent is trained for 10M environment interactions (40M frames) as opposed to the typical 50M [20, 21] and our experiments are only run on one seed.

To test robustness, all hyperparameters are fixed over all runs for all agents. The hyperparameters of DreamerV3 for the Atari benchmark are detailed in [18], most notably the 'imagination' horizon $H = 15$. The AMBS hyperparameters are also fixed in all experiments as follows, bounded safety level $\Delta = 0.1$, approximation error $\epsilon = 0.09$, number of samples $m = 512$, and look-ahead shielding horizon $T = 30$. Additional implementation details and hyperparameter settings can be found in the supplementary material [44].

Results We evaluate the effectiveness of AMBS equipped with DreamerV3 [21] by comparing it with the following algorithms, vanilla DreamerV3 without any shielding procedure or access to the cost function, a safety-aware DreamerV3 that implements an augmented Lagrangian penalty framework (LAG) [41, 35], since this method is based on DreamerV3 it should act as a more meaningful baseline compared to similar model-free counterparts like PPO- and TRPO- Lagrangian [35] that are not optimised for Atari.

In addition, we provide results for two state-of-the-art (single GPU) model-free algorithms, Rainbow [24] and Implicit Quantile-Network (IQN) [11]. While these model-free algorithms don't really provide a meaningful baseline, we include them (to the same end as [14]) to demonstrate that state-of-the-art Atari agents frequently violate quite straightforward safety properties that we would expect them to satisfy. We also note that both BPS [14] and classical shielding [3] are too unwieldy to be used during training in this setting, and so we cannot obtain a meaningful comparison between these approaches and AMBS.

We compare the five algorithms by recording the cumulative number of violations and the best episode score obtained during a single run of 10M environment interactions (40M frames). Table 2 presents the results. We also provide learning curves for each of the agents which can be found in the supplementary material [44].

Discussion. As we can see in Table 2, our approach (AMBS) compared with the other DreamerV3 [21] based agents dramatically reduces the total number of safety-violations during training in all five of the Atari games. In terms of reward, AMBS also achieves comparable or better best episode scores in all five of the Atari games. This

Table 2. Best episode scores and total violations for the five Atari games. *Blank (−) denotes that the agent failed to converge.*

		DreamerV3	DreamerV3 (AMBS)	DreamerV3 (LAG)	IQN	Rainbow
Assault	Best Score ↑	14738	**44467**	19832	9959	9632
	# Violations ↓	18745	**12638**	16802	24462	24019
DoubleDunk	Best Score ↑	**24**	**24**	**24**	**24**	-
	# Violations ↓	877499	**66248**	359018	188363	-
Enduro	Best Score ↑	2369	2367	2365	2375	**2383**
	# Violations ↓	167933	*132147*	174217	129012	**108000**
KungFuMaster	Best Score ↑	97000	**117200**	97200	51600	59500
	# Violations ↓	427476	**10936**	567559	284909	612762
Seaquest	Best Score ↑	4860	**145550**	1940	34150	1900
	# Violations ↓	73641	**40147**	64679	53516	67101

is likely because the objective of maximising rewards and minimising constraints are suitably aligned, which we note might not always be the case. LAG fails to reliably reduce the total number of safety-violations in all the Atari games, which could suggest this method is more sensitive to hyperparameter tuning. Clearly the extra machinery introduced by AMBS helps us to obtain a dynamic algorithm that can effectively switch between reward maximising and constraint satisfying policies, which on this set of Atari games demonstrates a significant improvement in performance compared to other safety-aware and state-of-the-art algorithms. Interestingly, the model-free algorithms do better across the board on *Enduro*, it is not immediately clear why this is the case and this result may need further investigation. Perhaps it is the case that DreamerV3 is not quite as suited to Enduro as IQN or Rainbow, although we note that AMBS improves on DreamerV3 w.r.t. safety-violations while maintaining comparable performance.

6 Related Work

Model-based RL. Model-based RL as a paradigm for learning complex policies has become increasingly popular in recent years due to its superior sample efficiency [18, 26]. Dyna – "an integrated architecture for learning, planning and reacting" [36] proposed an architecture that learns a dynamics model of the environment in addition to a reward maximising policy. In theory, by utilising the dynamics model for planning and/or policy optimisation we should learn better policies with fewer experience. Model-based policy optimisation (MBPO) [26] is a sample-efficient neural architecture for optimising policies in learned dynamics model. More recently, more sophisticated neural architectures such as Dreamer [18] and DreamerV2 [20] have been proposed that have demonstrated state-of-the-art performance on the DeepMind Visual Control Suite [38] and the Atari benchmark [8, 32] respectively. Our work is built on DreamerV3 [21] which has demonstrated superior performance in a number of domains (with fixed hyperparameters) including the Atari benchmark [8, 32] and MineRL [16], which is regarded as a notoriously difficult exploration challenge in the RL literature.

Shielding. Shield synthesis for RL was first introduced in [3] as a method for preventing learned policies from entering an unsafe region of the state space defined by a given temporal logic formula. In contrast to other safety-aware approaches based on soft penalties [1, 39], shielding enforces hard constraints by directly overriding unsafe actions with a verified backup policy. The reactive shield, which includes the backup policy, is computed ahead of time to ensure no training violations (*correctness*). To compute the reactive shield a safety automaton is constructed from a safety-relevant abstraction of the game with known dynamics and a safety game [9] is then solved.

Look-ahead shielding [14, 23, 42, 15] is a more dynamic approach to shielding that aims to alleviate some of the restrictive requirements that come with classical shielding. The key benefits of look-ahead shielding include, (1) better computational efficiency as only the reachable subset of the state space needs to be checked, (2) can be applied online in real-time from the current state. Although we note that look-ahead shielding can be done ahead of time [42] similar to classical shielding. Inspired by *latent shielding* [23] our approach can very much be categorised as an online (approximate) look-ahead shielding approach.

7 Conclusions

In this paper we introduce and describe approximate model-based shielding (AMBS), a general purpose framework for shielding RL policies by simulating and verifying possible futures with a learned dynamics model. In contrast with the majority of work on shielding [3], we apply our approach in a much less restrictive setting, where we only require access to an expert labelling of the states, rather than having the safety-relevant dynamics be known. In addition, we obtain further look-ahead capabilities in comparison to previous work, such as *latent shielding* [23] and BPS [14], by utilising safety critics that are trained to predict expected costs.

Compared with other safety-aware approaches, we empirically show that DreamerV3 [21] augmented with AMBS demonstrates superior performance in terms of episode return and cumulative safety-violations, on a set of Atari games with state-dependent safety-labels. We also develop a rigorous set of theoretical results that underpin AMBS, which includes strong probabilistic guarantees in the tabular setting. We stress the importance of these results, as it allows operators to meaningfully choose hyperparameter settings based on their specific requirements w.r.t. safety-guarantees.

All the components of our algorithm must be learnt from scratch, which unfortunately means that there are few guarantees during early stage training. Although in principle we could jump start the learning process with offline or expert data, this would be an interesting paradigm to explore. Important future work also includes, empirically verifying our theoretical results for the tabular setting and applying AMBS to more standard safe RL benchmarks, like Safety-Gym [35], that are not frequently used in the shielding literature.

Acknowledgements

This work was supported by UK Research and Innovation [grant number EP/S023356/1], in the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence (www.safeandtrustedai.org).

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel, 'Constrained policy optimization', in *International conference on machine learning*, pp. 22–31. PMLR, (2017).

[2] Syed Mumtaz Ali and Samuel D Silvey, 'A general class of coefficients of divergence of one distribution from another', *Journal of the Royal Statistical Society: Series B (Methodological)*, **28**(1), 131–142, (1966).

[3] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu, 'Safe reinforcement learning via shielding', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, (2018).

[4] Eitan Altman, *Constrained Markov decision processes: stochastic modeling*, Routledge, 1999.

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané, 'Concrete problems in ai safety', *arXiv preprint arXiv:1606.06565*, (2016).

[6] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause, 'Constrained policy optimization via bayesian world models', *arXiv preprint arXiv:2201.09802*, (2022).

[7] Christel Baier and Joost-Pieter Katoen, *Principles of model checking*, MIT press, 2008.

[8] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, 'The arcade learning environment: An evaluation platform for general agents', *Journal of Artificial Intelligence Research*, **47**, 253–279, (jun 2013).

[9] Roderick Bloem, Bettina Könighofer, Robert Könighofer, and Chao Wang, 'Shield synthesis: Runtime enforcement for reactive systems', in *International conference on tools and algorithms for the construction and analysis of systems*, pp. 533–548. Springer, (2015).

[10] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh, 'Lyapunov-based safe policy optimization for continuous control', *arXiv preprint arXiv:1901.10031*, (2019).

[11] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos, 'Implicit quantile networks for distributional reinforcement learning', in *International conference on machine learning*, pp. 1096–1105. PMLR, (2018).

[12] Scott Fujimoto, Herke Hoof, and David Meger, 'Addressing function approximation error in actor-critic methods', in *International conference on machine learning*, pp. 1587–1596. PMLR, (2018).

[13] Tanmay Gangwani, Joel Lehman, Qiang Liu, and Jian Peng, 'Learning belief representations for imitation learning in pomdps', in *Uncertainty in Artificial Intelligence*, pp. 1061–1071. PMLR, (2020).

[14] M Giacobbe, Mohammadhosein Hasanbeig, Daniel Kroening, and Hjalmar Wijk, 'Shielding atari games with bounded prescience', in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, (2021).

[15] Alexander W Goodall and Francesco Belardinelli, 'Approximate shielding of atari agents for safe exploration', *arXiv preprint arXiv:2304.11104*, (2023).

[16] William H Guss, Cayden Codel, Katja Hofmann, Brandon Houghton, Noboru Kuno, Stephanie Milani, Sharada Mohanty, Diego Perez Liebana, Ruslan Salakhutdinov, Nicholay Topin, et al., 'Neurips 2019 competition: the minerl competition on sample efficient reinforcement learning using human priors', *arXiv preprint arXiv:1904.10079*, (2019).

[17] David Ha and Jürgen Schmidhuber, 'Recurrent world models facilitate policy evolution', in *Advances in Neural Information Processing Systems*, eds., S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, volume 31. Curran Associates, Inc., (2018).

[18] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi, 'Dream to control: Learning behaviors by latent imagination', in *International Conference on Learning Representations*, (2020).

[19] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson, 'Learning latent dynamics for planning from pixels', in *International conference on machine learning*, pp. 2555–2565. PMLR, (2019).

[20] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba, 'Mastering atari with discrete world models', in *International Conference on Learning Representations*, (2021).

[21] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap, 'Mastering diverse domains through world models', *arXiv preprint arXiv:2301.04104*, (2023).

[22] Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft, 'Safe exploration for reinforcement learning.', in *ESANN*, pp. 143–148. Citeseer, (2008).

[23] P He, B Gonzalez Leon, and F Belardinelli, 'Do androids dream of electric fences? safety-aware reinforcement learning with latent shielding'. CEUR Workshop Proceedings.

[24] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver, 'Rainbow: Combining improvements in deep reinforcement learning', in *Proceedings of the AAAI conference on artificial intelligence*, volume 32, (2018).

[25] Wassily Hoeffding, 'Probability inequalities for sums of bounded random variables', *The collected works of Wassily Hoeffding*, 409–426, (1994).

[26] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine, 'When to trust your model: Model-based policy optimization', *Advances in neural information processing systems*, **32**, (2019).

[27] Michael Kearns and Satinder Singh, 'Near-optimal reinforcement learning in polynomial time', *Machine learning*, **49**, 209–232, (2002).

[28] Jonathan N Lee, Alekh Agarwal, Christoph Dann, and Tong Zhang, 'Learning in pomdps is sample-efficient with hindsight observability', *arXiv preprint arXiv:2301.13857*, (2023).

[29] Yongshuai Liu, Jiaxin Ding, and Xin Liu, 'Ipo: Interior-point policy optimization under constraints', in *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4940–4947, (2020).

[30] Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao, 'Constrained model-based reinforcement learning with robust cross-entropy method', *arXiv preprint arXiv:2010.07968*, (2020).

[31] Yuping Luo and Tengyu Ma, 'Learning barrier certificates: Towards safe reinforcement learning with zero training-time violations', *Advances in Neural Information Processing Systems*, **34**, 25621–25632, (2021).

[32] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling, 'Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents', *Journal of Artificial Intelligence Research*, **61**, 523–562, (2018).

[33] Haritz Odriozola-Olalde, Maider Zamalloa, and Nestor Arana-Arexolaleiba, 'Shielded reinforcement learning: A review of reactive methods for safe learning', in *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1–8. IEEE, (2023).

[34] Martin L Puterman, 'Markov decision processes', *Handbooks in operations research and management science*, **2**, 331–434, (1990).

[35] Alex Ray, Joshua Achiam, and Dario Amodei, 'Benchmarking safe exploration in deep reinforcement learning', *arXiv preprint arXiv:1910.01708*, **7**(1), 2, (2019).

[36] Richard S Sutton, 'Dyna, an integrated architecture for learning, planning, and reacting', *ACM Sigart Bulletin*, **2**(4), 160–163, (1991).

[37] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[38] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al., 'Deepmind control suite', *arXiv preprint arXiv:1801.00690*, (2018).

[39] Garrett Thomas, Yuping Luo, and Tengyu Ma, 'Safe reinforcement learning by imagining the near future', *Advances in Neural Information Processing Systems*, **34**, 13859–13869, (2021).

[40] Christopher KI Williams and Carl Edward Rasmussen, *Gaussian processes for machine learning*, volume 2, MIT press Cambridge, MA, 2006.

[41] Jorge Nocedal Stephen J Wright. Numerical optimization, 2006.

[42] Wenli Xiao, Yiwei Lyu, and John Dolan, 'Model-based dynamic shielding for safe and efficient multi-agent reinforcement learning', *arXiv preprint arXiv:2304.06281*, (2023).

[43] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge, 'Projection-based constrained policy optimization', in *International Conference on Learning Representations*.

[44] Alexander W. Goodall and Francesco Belardinelli, 'Approximate Model-Based Shielding for Safe Reinforcement Learning', arXiv preprint arXiv:2308.00707 (2023).