arXiv:2608.19836v1 [cs.LG] 20 Aug 2026

Adaptive Probabilistic Shielding by Learning MDPs for Safe Reinforcement Learning

Astrid Horn Brorholt[1][0009−0007−1824−0554], Maris F. L. Galesloot[2][0009−0002−5112−8584], Nils Jansen[2,3][0000−0003−1318−8973], Kim Guldstrand Larsen[1][0000−0002−5953−3384], and Christian Schilling[1][0000−0003−3658−1065]

[1] Aalborg University, Aalborg, Denmark
{asgerhb,kgl,christianms}@cs.aau.dk
[2] Radboud University, Nijmegen, Netherlands
maris.galesloot@ru.nl
[3] Ruhr University Bochum, Bochum, Germany
n.jansen@rub.de

Abstract. Probabilistic shielding is a technique for safe reinforcement learning (RL). Typically, a static observer—called the shield—constrains the learning agent's actions to those for which acting safely remains feasible. Traditionally, the shield is computed from the transition probabilities of the underlying Markov decision process (MDP). Thus, this technique is not applicable when the MDP model is not given a priori, which, unfortunately, is the case in typical RL applications. In this paper, we study the problem of computing a shield in the setting where the transition graph of the MDP is known, but the transition probabilities are unknown. Our approach integrates probabilistic shielding with online model learning: as the RL agent explores the environment, we estimate the transition probabilities. From this estimate, we compute a shield. While the shield may be conservative initially, it adapts as the model estimate becomes more precise. Thus, the shield improves in tandem with the RL agent. This paradigm of *adaptive probabilistic shielding* raises a number of challenges, such as when to recompute the shield and how to balance between exploration and safety during learning. We empirically evaluate multiple variants of this paradigm across several environments.

Keywords: Safe reinforcement learning · Shielding · Model learning · Interval Markov decision process.

1 Introduction

Markov decision processes (MDPs) [32] are the standard models to capture *decision-making under uncertainty* in artificial intelligence (AI) [24]. Factors such as unknown or unpredictable environments, contextual changes at runtime, or incomplete data are commonly referred to as *uncertainty*. Specifically, MDPs capture settings where agents, in each *state* of their environment, choose to execute *actions* upon which the environment *probabilistically transitions* to a new state. Upon that transition, the agent receives a *reward*.

Common objectives for MDPs are to (1) maximize the expected cumulative reward and (2) adhere to safety constraints specified as temporal logic constraints [30]. In the past, the first objective was mostly considered by the AI community, and the latter objective by the formal verification community. Specifically, reinforcement learning (RL) is a major AI technique for decision-making under uncertainty [38]. For an unknown MDP, an RL agent aims to maximize the expected reward by collecting data through exploration of the environment across multiple episodes. A major limitation in RL is that during exploration, the agent will necessarily execute potentially devastatingly unsafe actions. In contrast, probabilistic model checking (PMC) is a formal verification technique that computes the probability of satisfying a safety constraint in an MDP [3]. The key limitation of PMC is that the MDP must be fully specified.

Shielded RL. In response to these key limitations, a tremendous body of work has brought together RL and formal methods in the area of safe RL [13], in particular within shielded RL [1,9]. Specifically, in *probabilistic pre-shielding*, PMC is used to compute a shield that blocks potentially unsafe actions at runtime [25,21,17]. Such *runtime verification approach* renders RL (more) safe during exploration, yet inherits PMC's strong assumption that the (safety-relevant) environment model, that is, the MDP, must be fully specified. One may be tempted to approach this problem by first gathering sufficient training data from RL, then using that data to learn a full MDP model of the environment, and finally computing a shield from that model. However, safety during the data collection is not considered, making such an approach hardly applicable in real-world scenarios.

Problem setting. In this paper, we overcome the aforementioned key real-world limitation of shielded RL and propose a practical and adaptive approach. To that end, we impose mild assumptions about the environment in which the RL agent operates. First, we assume that simulation access to the true environment MDP is available from the initial state, which is a common assumption in RL. Second, we assume that the topology, that is, the underlying graph of the MDP, is known, which is much more realistic than knowing the exact transition probabilities.

Our approach: Safe RL via Adaptive Probabilistic Shielding. Fig. 1 shows an overview of our approach. As usual, the RL agent executes an action in the (unknown) MDP, yielding a reward and causing the environment to transition to a new state. A key component of our approach is a *model estimator*, based on approaches from [37]. While the agent interacts with the environment, it collects data on the observed states and actions. From this data, the model estimator then constructs a *learned* MDP model of the environment. We use the state-of-the-art PMC tool PRISM [26] to compute a shield based on this learned MDP [12]. At any point during this process, the shield can be updated *adaptively*, and additional data can be collected under the updated shield.

Interval MDPs and shields. An essential part of our approach is to implement and compare estimators that yield different types of MDP models from moderate

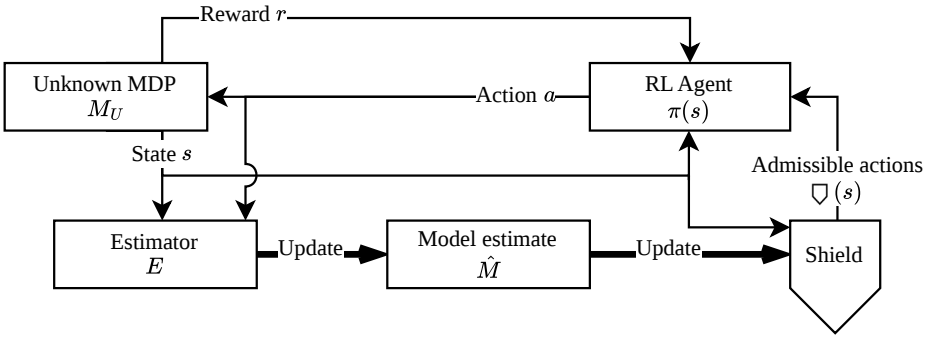

Fig. 1. A high-level overview of our adaptive probabilistic shielding approach.

amounts of data collected as a byproduct of the RL process itself. In particular, following [37,12], we create so-called interval MDPs (iMDPs) [29,36]. Intuitively, iMDPs capture data uncertainty robustly by defining upper and lower bounds on transition probabilities, based on, for instance, confidence intervals around point estimates of those probabilities. Then, PMC can provide upper and lower bounds on the safety probabilities for iMDPs. In [12], the worst-case estimates of the bounds are used to compute a *robust*, conservative shield. Consider the case where a particular action imposes a lower bound of 10% and an upper bound of 30% on the probability of reaching a safety-critical unsafe state. The application at hand may allow reaching such a state with a maximum probability of 20%. A shield with a *robust* uncertainty interpretation would block that action.

Safety vs. exploration. The key strength of our adaptive shielding approach is that gathering more data yields more accurate model estimates, which reduce the size of the probability intervals and allow for less conservative shields. The challenge, however, is that a too-conservative shield may, in the extreme, impede any exploration of the environment and thereby prevent the agent from gathering the necessary data to refine the shield and learn a good policy. One solution to this problem is to use an *optimistic* interpretation of uncertainty, as is common in robust RL and referred to as optimism in the face of uncertainty [28]. In the example above, the (optimistic) shield would then use the lower probability bound of 10% and allow the critical action. Another approach is to use common RL exploration techniques to address the exploration-exploitation dilemma [38].

Contributions and research questions. The main contribution of this paper is a novel, adaptive shielding algorithm that accounts for the uncertainty in estimating an MDP from data. We provide a thorough experimental evaluation structured around several concrete research questions. First, we evaluate if an adaptive shielding approach is beneficial to (1) obtain a safe and reward-optimal policy after training and (2) remain safe during training. Then, we investigate whether the choice of model estimator affects the performance of the shield and

how far the quality of the MDP estimate improves over time with respect to the (conservativeness) of the shield. Finally, we take into account various practical considerations, such as the number of model updates in relation to the number of RL episodes. The paper is structured as follows. In the remainder of the introduction, we discuss related work. In Section 2, we provide necessary background, followed by the definitions of shields for (interval) MDPs and model estimation in Section 3. Section 4 describes our adaptive probabilistic shielding algorithm. Finally, we present our research questions and experimental analysis in Section 5.

1.1 Related work

Probabilistic shielding. Classic shields provide unconditional safety guarantees, but this is often too conservative [17,25]. Shielding has been extended to many classes of models [10,5,8,23,1,33,19,6]. We consider probabilistic shields that permit a level of risk of reaching unsafe states [16,21,17,25]. In this paper, we focus on practical shields that guarantee the *admissibility* of probabilistic safety guarantees [21,31,12,25]. Specifically, our shielding method is based on [12].

Adaptive shielding. Several works have considered shields that change over time. Pranger et al. adaptively construct a finite-state environment abstraction from past observations to maintain a shield [31]. Similarly, Tappler et al. present an iterative approach using automata learning [39]. In other work, the shield adapts to changes in the environment, assuming white-box access to the (parametric) dynamics [34,11]. Goodall et al. estimate probabilistic safety by simulating future trajectories in a learned latent model [14]. Bethell et al. learn a shield using an auto-encoder and adjust the safety threshold in a subsequent RL phase [4].

Shielding based on model estimates. Galesloot et al. estimate iMDPs for shielding in *offline* RL [12]. Suilen et al. obtain iMDP estimates via optimistic exploration and then analytically compute a robust policy [37]. Another recent work collects environment data offline and then computes a robust shield for shielded RL, with the main contribution being a new algorithm to compute the shield [15].

Delimitation. The main differences of our problem setting are as follows. We assume a static environment with a known, finite state space and transition structure, but with unknown transition probabilities. We further assume black-box access instead of a settable simulator, which is why we specifically care about safety during the whole process, including data collection and exploration. Our method differs from previous work in that we integrate model estimation and updates of both the shield and the policy into a single integrated *online* RL procedure with adaptive probabilistic shields.

2 Preliminaries

Probability distributions. Given a set X, a probability distribution $p\colon X \to [0,1]$ satisfies $\sum_{x \in X} p(x) = 1$. Let $\Delta(X)$ denote the set of probability distributions over X and let $\mathrm{Unif}_X \in \Delta(X)$ denote the uniform distribution over X.

Intervals. A closed interval $[a, b] \subseteq \mathbb{R}$ for $a \leq b$ describes the set $\{x \in \mathbb{R} \mid a \leq x \leq b\}$. An open interval (a, b) describes $\{x \in \mathbb{R} \mid a < x < b\}$. Half-open intervals are defined analogously. Let $\mathbb{I} = \{[a, b] \mid 0 < a \leq b \leq 1\}$ denote the set of uncertain nonzero probabilities.

Markov decision processes. A *Markov decision process* (MDP) is a tuple $M = (S, A, s_0, T)$ where S is the finite set of states, A is the finite set of actions, $s_0 \in S$ is the initial state, and $T\colon S \times A \times S \to [0, 1]$ is the probabilistic transition function satisfying $\sum_{s' \in S} T(s, a, s') = 1$ for all s and a. A *run* is an alternating sequence $s_0 a_0 s_1 a_1 \ldots$ of states and actions such that $T(s_i, a_i, s_{i+1}) > 0$ for all i.

We consider two types of policies. A *deterministic policy* $\pi\colon S \to A$ maps each state to an action. A *nondeterministic policy* $\pi_N\colon S \to 2^A$ maps each state to a set of actions. A run $s_0 a_0 s_1 a_1 \ldots$ is an *outcome* of a deterministic policy π (resp. nondeterministic policy π_N) if $a_i = \pi(s_i)$ (resp. $a_i \in \pi_N(s_i)$) for all i.

An *unknown* MDP (uMDP) is a tuple $M_U = (S, A, s_0, T_U)$ where S, A, and s_0 are defined as for MDPs and $T_U\colon S \times A \to 2^S$ is a nondeterministic transition function (i.e., uMDPs are ordinary transition systems). Each MDP induces an unknown MDP by removing impossible transitions and dropping the probabilities; formally: $T_U(s, a) = \{s' \in S \mid T(s, a, s') > 0\}$. An *interval* MDP (iMDP) [22,29,35,20,37,36] is a tuple $M_I = (S, A, s_0, T_I)$ where again S, A, and s_0 are defined as for MDPs and $T_I\colon S \times A \times S \to \mathbb{I} \cup \{0\}$ is an *interval transition function*. Consider an MDP $M = (S, A, s_0, T)$ and an iMDP $M_I = (S, A, s_0, T_I)$ over S and A. We say that T_I *abstracts* T, written $T \in T_I$, if T is a probabilistic transition function and each interval in T_I contains the corresponding probability in T; formally: $\forall s \in S\ \forall a \in A\colon \sum_{s' \in S} T(s, a, s') = 1 \wedge \forall s' \in S\colon T(s, a, s') \in T_I(s, a, s')$. Analogously, we say that M_I abstracts M, written $M \in M_I$. Note that, because intervals in $\mathbb{I}$ must not include 0, if $T \in T_I$ and $T' \in T_I$, then the same transitions in T and T' have a non-zero probability; formally: $\forall T, T' \in T_I\ \forall s, s' \in S\ \forall a \in A\colon T(s, a, s') > 0 \implies T'(s, a, s') > 0$.

Reinforcement learning. We assume that the reader is familiar with reinforcement learning (RL) [38] and only recall some basic commonalities. Let $R\colon S \times A \times S \to \mathbb{R}$ be the reward function and $\gamma \in [0, 1)$ a discount factor. Given a deterministic policy π, the *expected cumulative discounted reward* from the initial state s_0 is $V^\pi(s_0) = \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right]$, where at each step $t \in \mathbb{N}$, the action is $a_t = \pi(s_t)$ and the expectation is taken over the probabilistic state transitions governed by the transition function T. In RL, an agent explores an environment (which is assumed to be an MDP) with the aim to learn a policy that maximizes the expected cumulative discounted reward from the exploration experience. This "training" takes place in episodes of multiple steps each. RL requires only black-box sampling access to the environment MDP from an initial state. Two major paradigms are *model-free* and *model-based* RL [38]. The latter learns an approximation of the MDP as the agent explores. On the one hand, the algorithm proposed in this paper learns such an approximation to construct the shield, and is therefore model-based. On the other hand, the algorithm also includes an RL component, for which our prototype implementation

uses Q-learning (which is model-free). We note that most other RL algorithms (model-free or model-based) could also be used in place of Q-learning.

During training, we use an *ε-greedy exploration/exploitation* strategy [38]: at each step, the agent chooses a random action with probability ε ("exploration") and follows the (partially) learned policy with probability $1-\varepsilon$ ("exploitation").

Safety and shielding. We consider safety properties $\varphi \subseteq S$ given as a set of safe states. A run $s_0 a_0 s_1 a_1 \dots$ is *safe* if $s_i \in \varphi$ for all i.

Given a deterministic policy $\pi\colon S \to A$, a *shield* is any nondeterministic policy $\shield\colon S \to 2^A$ over the same states and actions. The *shielded policy* $\pi_\shield$ is a deterministic policy that acts similarly to π but only chooses actions allowed by the shield, i.e., $\forall s \in S\ \forall a \in A\colon \pi_\shield(s) = a \implies a \in \shield(s)$. Additionally, the shield only alters actions when necessary: $\forall s \in S\colon \pi(s) \in \shield(s) \implies \pi_\shield(s) = \pi(s)$. The action chosen by $\pi_\shield(s)$ when $\pi(s) \notin \shield(s)$ depends on the implementation of the RL agent. In our implementation, we use Q-learning [41], for which it is straightforward to read out the best admissible action in $\shield(s)$.

3 Probabilistic Shielding Using an Estimator

In this paper, we construct shields based on the method described in [12], which was originally developed for the *offline* RL problem where a fixed dataset is given. Our approach differs in that we continuously adapt the shield based on newly generated data, and we must consider exploration to collect new data within and beyond the shield's allowed actions. As we will see later, such a setting is particularly challenging and yields trade-offs between safety and exploration.

3.1 Probabilistic shielding approaches for (interval) MDPs

Next, we recall how to obtain a shield $\shield$ for an (interval) MDP M and a safety specification φ. We assume a horizon $h \in \mathbb{N}$ and parameters $\theta, \kappa \in [0,1]$, which we explain below. Before the formalization, we first describe the high-level idea. Intuitively, the shield ensures the existence of a series of h actions from the current state s such that the chance of a safety violation along these steps is below θ. For tractability, the shield is memoryless and thus ignores accumulated past risk before reaching the current state s. Since the restriction is probabilistic and memoryless, there may still be a chance of reaching a state where this guarantee does not hold [12,25,17]. Whenever no sufficiently safe action is available, the shield only allows actions that are κ-close to the safest available action.

Now we formalize this idea. Let $\varphi|h$ be the set of all *h-safe runs* $s_0 a_0 s_1 a_1 \dots$ with safe h-prefix, i.e., $s_i \in \varphi$ for $i \leq h$. Let $\mathbb{P}_\pi^{M,\varphi|h}(s)$ be the probability of an MDP M producing an h-safe run by following policy π starting in state s. We denote the related safety optimization problem by $\mathbb{P}_{\max}^{M,\varphi|h}(s) = \max_\pi \mathbb{P}_\pi^{M,\varphi|h}(s)$. Moreover, let the probability of producing a run in $\varphi|h$ after taking action a in state s be $\mathbb{P}_{\max}^{M,\varphi|h}(s,a) = \sum_{s' \in S} T(s,a,s') \mathbb{P}_{\max}^{M,\varphi|h-1}(s')$.

Normally, the shield allows all actions guaranteeing a safe h-step run with probability at least $1-\theta$; formally: $\shield_\theta(s) = \{a \in A \mid \mathbb{P}^{M,\varphi|h}_{\max}(s,a) \geq 1-\theta\}$. However, a shielded policy may still reach a state s where this shield definition would not allow any action (i.e., $\shield_\theta(s) = \emptyset$). In that case, the shield instead allows all actions that are κ-close to the safest available action; formally: $\shield_\kappa(s) = \{a \in A \mid \mathbb{P}^{M,\varphi|h}_{\max}(s,a) \geq \max_{a'} \mathbb{P}^{M,\varphi|h}_{\max}(s,a') - \kappa\}$. Our shield combines these two cases:

$$\shield(s) = \begin{cases} \shield_\theta(s) & \text{if } \shield_\theta(s) \neq \emptyset \\ \shield_\kappa(s) & \text{otherwise.} \end{cases} \tag{1}$$

Following [12], we extend shields to iMDPs by defining the probability of producing a run for iMDPs as follows. Let $\mathbb{P}^{M_I,\varphi|h}_{\max\,\mathrm{opt}}(s) = \max_\pi \mathrm{opt}_{M \in M_I} \mathbb{P}^{M,\varphi|h}_\pi(s)$ be the probability of producing a run under the model $M \in M_I$ corresponding to the optimization direction $\mathrm{opt} \in \{\min, \max\}$. Moreover, let the iMDP version of the probability producing a run in $\varphi|h$ after taking action a in state s be

$$\mathbb{P}^{M_I,\varphi|h}_{\max\,\mathrm{opt}}(s,a) = \operatorname*{opt}_{T^\dagger \in T_I} \sum_{s' \in S} T^\dagger(s,a,s') \mathbb{P}^{M_I,\varphi|h-1}_{\max\,\mathrm{opt}}(s'). \tag{2}$$

We use the probabilistic model checker PRISM [26] to efficiently compute such probabilities on iMDPs. Statistically speaking, a worst-case assumption ($\mathrm{opt} = \min$) makes the shield *robust* against estimation errors. Following [12], we define a (pessimistic) *robust shield* for an iMDP M_I similarly to Eq. (1) using $\mathbb{P}^{M_I,\varphi|h}_{\max\min}(s,a)$. Instead of assuming the worst-case MDP M from an estimate M_I, one can also assume the best case ($\mathrm{opt} = \max$), specified as $\mathbb{P}^{M_I,\varphi|h}_{\max\max}(s)$. We call this alternative an *optimistic shield*. We say that the *attitude* of a shield is either robust or optimistic, depending on how it was constructed from an iMDP.

3.2 Estimators from data for unknown MDPs

We recall three existing estimators, based on those that appeared in [37]. While exploring the black-box MDP, we count how many times a transition triple (s,a,s') has been observed. For that, we use a transition database $D\colon S \times A \times S \to \mathbb{N}$. Let $D(s,a) = \sum_{s'} D(s,a,s')$ be the *total count* for a state-action pair (s,a).

An *estimator* is a function $E(M_U, D)$ that maps a uMDP M_U and a transition database D to either an estimated MDP $\hat{M}$ or iMDP $\hat{M}_I$, depending on the estimator. Below, we describe three estimators that learn (i.e., estimate the transition function of) MDPs or iMDPs, with or without guarantees: *MAP* (E_{MAP}), *PAC* (E_{PAC}), and *LUI* (E_{LUI}). These approaches estimate the transition function locally for each (s,a)-pair using knowledge of the graph in the form of a given uMDP M_U, where $T_U(s,a)$ denotes the set of successor states.

MAP. The first approach finds a *point estimate* $\hat{T}$ of the MDP's transition function T based on the data D. Following [37], we define point estimates as maximum a-posteriori (MAP) estimation with respect to a symmetric prior weight

assigned to each successor state, which we denote as a single $w \in \mathbb{N}$. Then

$$\hat{T}(s,a,s') = \frac{w + D(s,a,s') - 1}{\left(\sum_{t \in T_U(s,a)} w + D(s,a,t)\right) - |T_U(s,a)|}$$

defines the MAP point estimate. It can be viewed as a maximum-likelihood probability with some additive smoothing from w. The MAP estimator E_{MAP} maps (M_U, D) to an estimated MDP $\hat{M}$ with point estimates $\hat{T}$.

PAC. The point estimates of the MAP estimator do not account for the uncertainty arising from estimating probabilities from data. Point estimates can be turned into *probably approximately correct* (PAC) intervals via Hoeffding's inequality [18], such that the estimated iMDP contains the true MDP with high probability $1 - \delta$, for $\delta \in [0, 1]$ [2,37,12]. As such, by the union bound, we distribute δ over all transitions as $\delta_T = {}^{\delta}/_{\sum_{s,a} k(s,a)}$, where $k(s,a)$ denotes the number of successor states $k(s,a) = |T_U(s,a)|$ if $|T_U(s,a)| > 1$ and $k(s,a) = 0$ otherwise. Then, $\eta_{s,a} = {}^{\log[2/\delta_T]}/_{2 \cdot D(s,a)}$ denotes the range of the PAC interval around the point estimate for (s,a). Using each $\eta_{s,a}$, we construct the intervals

$$\hat{T}_I(s,a,s') = \left[\max(\xi, \hat{T}(s,a,s') - \eta_{s,a}), \min(1, \hat{T}(s,a,s') + \eta_{s,a})\right] \tag{3}$$

where $\xi \in (0,1)$ is a small constant that ensures intervals for transitions with nonzero probability (as given by the unknown MDP M_U) map to $[\xi, 1]$. The PAC estimator E_{PAC} maps (M_U, D) to an iMDP $\hat{M}_I$ with intervals $\hat{T}_I$ from Eq. (3).

LUI. The third approach that we consider is the *linearly updating intervals* (LUI) estimator from [37], which in turn is based on [40]. While it does not retain PAC guarantees, it iteratively learns probabilities by updating intervals. We assign each unknown transition a prior interval $\tilde{T}_I(s,a,s'_i) = [\underline{T}_i, \overline{T}_i]$ and prior strength $[\underline{n}_i, \overline{n}_i]$. The strength influences the prior's effect on the updated intervals. At any point, we find new intervals given the database D by distinguishing cases based on whether the current intervals agree with the new data. For any (s,a) and s'_j, let $F_j = {}^{D(s,a,s'_j)}/_{D(s,a)}$ denote the relative occurrence of transition (s,a,s'_j) in the database D. Then, the updates are:

$$\underline{T}_i \leftarrow \begin{cases} \frac{\overline{n}_i \underline{T}_i + D(s,a,s'_i)}{\overline{n}_i + D(s,a)} & \text{if } F_j \geq \underline{T}_j \text{ for all } s'_j, \\ \frac{\underline{n}_i \underline{T}_i + D(s,a,s'_i)}{\underline{n}_i + D(s,a)} & \text{otherwise.} \end{cases} \tag{4}$$

$$\overline{T}_i \leftarrow \begin{cases} \frac{\overline{n}_i \overline{T}_i + D(s,a,s'_i)}{\overline{n}_i + D(s,a)} & \text{if } F_j \leq \overline{T}_j \text{ for all } s'_j, \\ \frac{\underline{n}_i \overline{T}_i + D(s,a,s'_i)}{\underline{n}_i + D(s,a)} & \text{otherwise.} \end{cases} \tag{5}$$

The (strength) intervals are found from total counts $[\underline{n}_i + D(s,a), \overline{n}_i + D(s,a)]$. Initial intervals are valid when $0 < \underline{T}_i \leq \overline{T}_j \leq 1$ and $\overline{n}_i \geq \underline{n}_i \geq 1$. Similarly to E_{PAC}, E_{LUI} maps (M_U, D) to an iMDP $\tilde{M}_I$ with intervals $\tilde{T}_I$ using Eq. (4).

Guarantees and convergence. While only E_{PAC} provides statistical guarantees from finite data [2], all three estimators converge to the true probabilities as the number of visits to each transition tends to infinity [37].

4 Adaptive Probabilistic Shielding

In this section, we develop the paradigm that we call *adaptive probabilistic shielding*. Before we present our algorithm, we motivate the problem it addresses.

4.1 Problem Statement

We consider an RL application in a safety-critical real-world scenario. In particular, we do not know the underlying MDP model (only the underlying uMDP) and hence do not assume access to a settable simulator. While safety violations may not be entirely avoidable, we place great importance on them, as they may occur during real-world data collection outside a simulator [27]. Hence, our goal is to obtain a policy π subject to three sub-goals: (i) achieve a given admissibility threshold of the safety specification, (ii) achieve a high expected reward, and (iii) achieve a low number of safety violations during training.

To highlight the intricacy of our problem, we point out that goal (iii) is in direct competition with the other two goals. This is because higher safety during training requires more conservative exploration, which may prevent the discovery of a better-performing policy (e.g., a faster and/or safer route to a goal state). Since we do not assume prior knowledge of the environment's transition dynamics, one may have to take more risks to learn them; thus, an action deemed less safe due to higher uncertainty may only be determined to be safer after enough exploration. While RL solves (ii), it does not achieve (i), and it may also perform poorly regarding (iii), since it typically relies on (random) exploration of the environment. To additionally achieve (i), we could apply shielded RL; however, since we do not know the MDP, we would need to learn an MDP or iMDP model, which would itself require exploration for the data collection and thus again fail to achieve goal (iii).

4.2 Adaptive Probabilistic Shielding

Our answer to this dilemma is to interweave all three procedures (policy learning, shield construction, and model estimation) into a single adaptive learning loop. Generally, we collect data from the RL agent's exploration of the environment. From time to time, we use that data to update our model estimate, and from that improved estimate, we obtain a refined shield that allows us to continue exploring the environment more safely and/or less conservatively.

One may be tempted to think that this process will, given enough episodes, converge to the ideal solution of learning the underlying MDP and thus the best possible shield. However, this is not necessarily the case, and indeed, we observed that such an approach can fail in practice. The issue is that the conservative

Algorithm 1 Safe RL via Adaptive Probabilistic Shielding

Input: black-box MDP $M = (S, A, s_0, T)$; uMDP $M_U = (S, A, s_0, T_U)$; reward function R; safety specification φ; estimator E; shield update delay $u \in \mathbb{N}$; shield parameters $\theta, \kappa \in [0, 1]$ and $h \in \mathbb{N}$; exploration rate $\varepsilon \in [0, 1]$; number of episodes $N \in \mathbb{N}$; maximum episode length $L \in \mathbb{N}$

1: $\pi \leftarrow$ Initialize RL policy
2: $D(s, a, s') \leftarrow 0, \quad \forall\, (s, a, s')$ ▷ *Initialize transition database*
3: **for** i **in** 0 **to** $N - 1$ **do**
4: **if** $i \equiv 0 \mod u$ **then**
5: $\hat{M} \leftarrow E(M_U, D)$ ▷ *See Section 3.2*
6: $\shield \leftarrow$ Synthesize shield from $\hat{M}$, θ, κ, and h for φ ▷ *See Section 3.1*
7: $s \leftarrow s_0$
8: **for** j **in** 0 **to** $L - 1$ **do**
9: **if** Random bit with probability ε of being true **then**
10: $a \sim \mathrm{Unif}_A$ ▷ *"Explore" – uniform choice among all actions*
11: **else**
12: $a = \pi_{\shield}(s)$ ▷ *"Exploit" – agent chooses the best safe action*
13: $s' \sim T(s, a)$ ▷ *Take a step with action a in the environment*
14: $\pi \leftarrow$ Update policy with transition (s, a, s') and reward $R(s, a, s')$
15: $D(s, a, s') \leftarrow D(s, a, s') + 1$
16: $s \leftarrow s'$

return $(\pi, \shield, \hat{M})$

shield, from the beginning, may simply prevent exploration of large parts of the state space, even if the corresponding actions were to be perfectly safe under the true MDP model. More specifically, the shield cannot distinguish between actions that are already known to be risky (which it should indeed block) and actions for which the model estimate is too coarse to make a definite judgement. This is particularly pronounced for the more pessimistic robust shields.

Our final step is to extend the ε-greedy exploration strategy to explore beyond the shield's boundaries. When the exploration strategy decides to explore a random action in state s, we choose this action from the full set of actions A, rather than just from $\shield(s)$ allowed by the shield. We show the impact of this extension empirically in the next section in (RQ6).

Algorithm 1 shows the pseudocode of our proposed approach. Notably, we only use white-box access to the environment via the uMDP M_U, while we only access the underlying MDP M implicitly via the black-box functions T and R.

In our implementation, we use Q-learning [41] to find a policy π. For that, we initialize π as an empty Q-table and an empty database D of observed transition triples. In the first iteration of the outer for-loop in Line 3, $i = 0$ satisfies the condition in Line 4, which triggers the computation of the first model estimate $\hat{M}$ and shield $\shield$. Since the transition database is still empty, this estimate and shield are most conservative. The inner for-loop in Line 8 represents Q-learning for a single episode with the extended ε-greedy exploration strategy described above under $\shield$. In particular, we either select a random action ("explore") or select the

current best action that is allowed by the shield ("exploit"). The selected action is then executed in the environment MDP M. Its output is recorded in the transition database D and used along with the immediate reward r to update the Q-table. Every u episodes, we update the model estimate and shield. After exceeding the training budget of N episodes, we return the final versions of the policy, the shield, and the model estimate.

5 Experimental Evaluation

In this section, we evaluate our proposed adaptive probabilistic shielding approach from different angles. We aim to answer the following research questions:

(RQ1) Can our approach learn a safe and optimal policy?
(RQ2) What is the effect on the environment exploration?
(RQ3) Does the model estimate improve over time?
(RQ4) Do the model estimates become sufficiently precise?
(RQ5) What is the impact of the specific model estimator?
(RQ6) Is unshielded exploration beneficial?
(RQ7) How often should the shield be updated?
(RQ8) What is the impact of the shield lookahead (h)?

5.1 Implementation and Baseline Methods

The implementation is available online [7]. We train the agent using standard Q-learning [41] with the hyperparameters $\alpha = 0.1$ (learning rate), $\gamma = 0.9$ (discount factor), and $\varepsilon = 0.05$ (exploration probability). We use reward shaping to penalize safety violations, with the penalty varying by environment. Instead, we separately record the number of episodes that were unsafe.

By default, we use the LUI estimator E_{LUI} with prior strengths $[\underline{n}_i, \overline{n}_i] = [5, 10]$ and a ("pessimistic") robust shield attitude with parameters $\theta = 0.05$, $\kappa = 0.01$, and $h = 100$, and update the estimate and shield every $u = 1000$ episodes. We underline these defaults in the following figures and tables.

We compare to two baselines. The first baseline is a standard unshielded RL agent trained with reward shaping; this baseline is expected to perform poorly in terms of safety due to the lack of a shield. The second baseline is a shielded RL agent that uses a probabilistic shield computed with the same method but given the ground-truth MDP (which our method cannot access); this "oracle" baseline acts as a benchmark and is expected to outperform all other methods.

5.2 Description of Environments

In total, we consider five different environments with mixed safety and optimization objectives. The first environment is described in previous literature while the remaining were developed as additional benchmarks for our problem setting.

The **aircraft** environment [37,24] ($|S| = 1665$) represents a collision avoidance system of an aircraft that must keep a minimum vertical distance to another

plane that passes horizontally. The other plane changes altitude at random, and the agent's aircraft may fail to follow the instructions with a small probability.

The **antlion** environment ($|S| = 400$) requires the agent (an ant) to circumnavigate a stationary predator in order to reach a goal on the other side. Instead of moving in the intended direction, the agent may slip toward the predator, with the probability increasing with proximity to it. Reaching the goal yields a reward, while taking a step has a cost that decreases with proximity to the goal.

The **sinkholes** environment ($|S| = 400$) features multiple goals with varying rewards and multiple holes that must be avoided. If the agent falls into a hole, it either escapes with a small probability or returns to the starting state. As the agent moves, it may slip in a random direction instead, with the probability varying by the state. The cost of moving decreases with proximity to the goal.

The **crossroads** environment ($|S| = 202$) illustrates deferred risk. In the initial state, the agent must choose between two roads, which are then followed for 100 steps. One route is safe, while the other route is more rewarding but has a risk of slipping into an unsafe state at each step.

The **gravity** environment ($|S| = 2000$) rewards the agent for visiting a sequence of checkpoints near a gravity well without crashing into the latter. Every step has a small cost, and there is a probability that the agent is dragged towards the well, which increases with proximity. Later checkpoints are riskier to visit, and the agent can end the episode early by going to one of two exit points.

5.3 Experimental Results and Discussion of Research Questions

Our experiments consists of 21 hyperparameter configurations (e.g., the choice of the model estimator), and we repeated each run 100 times.

(RQ1): Can our approach learn a safe and optimal policy? We are interested in both safety (Fig. 2 column I) and performance (Fig. 2 column II) over time. (Note that the penalty stemming from the reward shaping is not included in these plots.) For most environments, the adaptive shield leads to about the same number of safety violations as the oracle shield. The unshielded baseline is generally less safe, especially in the crossroads and gravity environments, despite a strong reward penalty. Still, the negative outcomes were overshadowed by the more frequent positive rewards in the Q-learning algorithm. We also note that the oracle baseline can generally explore more freely than the adaptive method because of its less conservative shield. Yet, in the aircraft environment, the adaptive method actually achieves a slightly more rewarding policy, profiting from slightly riskier behavior. This may seem counterintuitive given the robust attitude of the shield, which is generally more conservative than the oracle baseline. The reason we still see this behavior is that, during the exploration, profitable states are visited more often, making these seem safer as compared to less explored states with wider interval estimates. Indeed, column III reveals that the shield mostly falls back to the $\triangledown_\kappa$ variant in this case.

Updates to the adaptive shield can be seen to temporarily affect performance negatively in the antlion, sinkholes, and gravity environments. At the

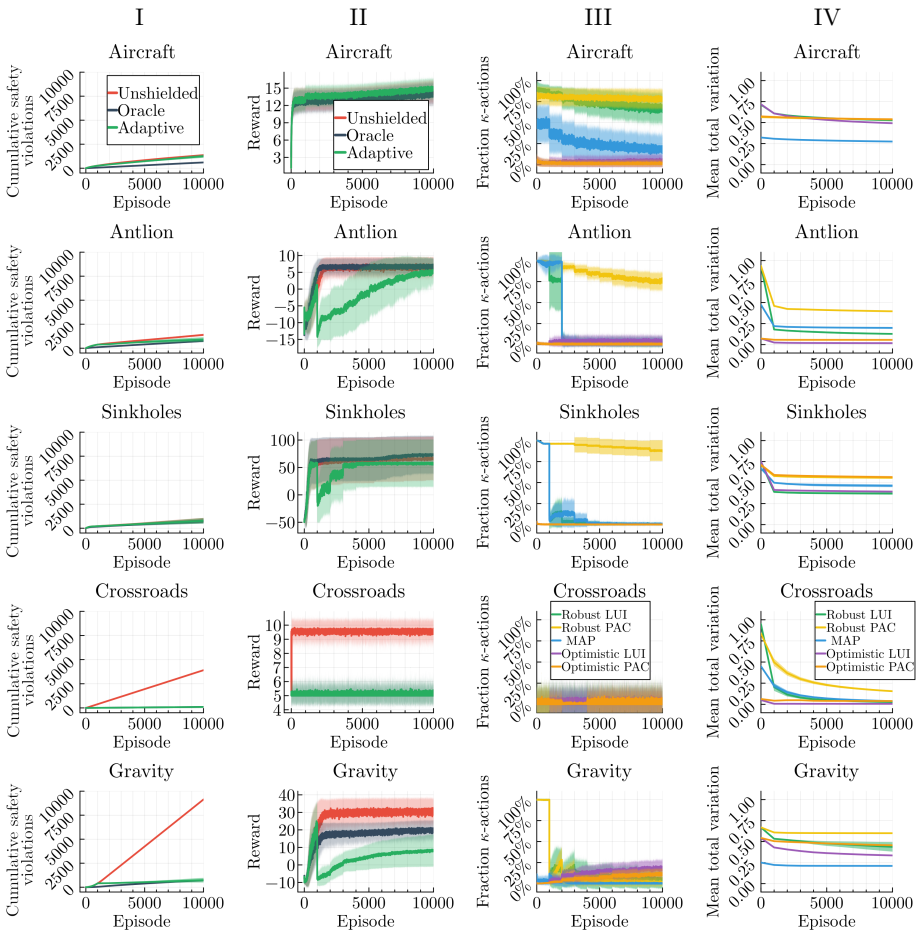

Fig. 2. Mean outcome of 100 repetitions for different configurations. We plot the standard deviation as a ribbon around the lines. Vertical grid lines mark updates of the adaptive shield. Column I: Cumulative safety violations during training. Column II: Reward during training. Column III: Per-episode rate of using the fallback ∇_κ. Column IV: Average total variation between true model and model estimate during training.

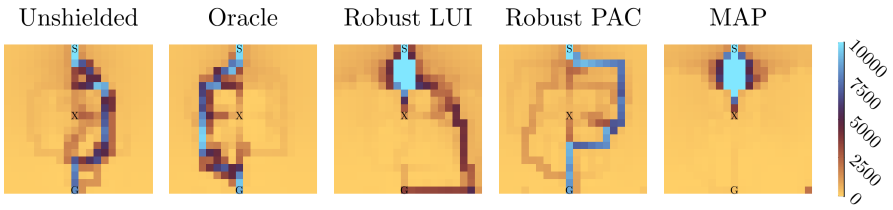

Fig. 3. Heat map showing how frequently a state is visited in the antlion environment. The initial state, predator, and goal are at the bottom, center, and top, respectively.

1000 episode mark, the shield is updated for the first time, leading to a drop in the reward performance. This drop is to be expected: as seen in column III, the initial shield typically uses the $\triangledown_\kappa$ variant because every action seems unsafe. The first model update is most impactful and hence typically leads to a very different shield, and hence the agent effectively experiences a different environment. Over time, the reward performance recovers as the policy adapts to the new shield.

Overall, the shields have a significant positive impact on safety and explore the environment more safely. This safety may come at a cost when risky behavior is profitable, but this is a desirable trade-off in many applications.

(RQ2): What is the effect on the environment exploration? To assess how adaptive shields affect exploration, Fig. 3 visualizes the number of times a state has been visited in the antlion environment during one algorithm execution over 10,000 episodes, comparing the unshielded and oracle baselines as well as adaptive shields with the estimators *robust LUI* (default), *robust PAC*, and *MAP*.

The LUI and PAC interval estimators both find the goal by following some narrow paths during learning. Meanwhile, the MAP estimator almost never leaves the area around the initial states. This is consistent with the low average performance of the MAP estimator in Table 1 and further investigated below in (RQ5). The unshielded agent passes close to the antlion but takes a more circuitous route, similar to what the oracle shield allows. Both appear to explore more freely around the paths they take, rather than staying on a narrow route permitted by the adaptive shield. This highlights the need for exploring states outside of what the shield allows, as also investigated in (RQ6) below.

(RQ3): Does the model estimate improve over time? We plot mean *total variation* (TV), i.e., $1/|S||A| \sum_{s,a} 1/2 \sum_{s' \in S} |T^\dagger(s,a,s') - T^*(s,a,s')|$, between the true MDP's transition probabilities T^* and the probabilities $T^\dagger$ returned by PMC used for the shield (e.g., for robust/optimistic shields, as found from Eq. (2)). We omit transitions with probability 1, since their probability is known.

As the agent explores the environment, we obtain a more precise estimate of the model. Exploration also comes with risk; thus, it is not desirable to obtain a perfect estimate of all transitions – in particular not those transitions that are rarely visited by the policy. The mean TV after each model update is shown in column IV of Fig. 2. The most significant change occurs in the first update after 1000 episodes, indicating that this is sufficient to collect data for key transitions, which will be further explored during the rest of the training. While the estimates for these transitions keep improving, this has little impact on the (global) metric.

(RQ4): Do the model estimates become sufficiently precise? We examine how often the shield allows an action because it satisfies the θ threshold ($\triangledown_\theta$) respectively how often it has to use the fallback ($\triangledown_\kappa$) (cf. Eq. (1)). Column III of Fig. 2 shows the fallback frequency. Until the first model update (1000 episodes), the robust and MAP estimates primarily use $\triangledown_\kappa$, whereas the optimistic estimates, which consider almost all actions safe from the beginning, primarily use $\triangledown_\theta$. The slowest improvement of the estimate is observed in the aircraft environment.

Table 1. Comparison for different model estimators. The numbers in each cell respectively denote the reward in the final evaluation (left) and the probability that the final policy produces an unsafe episode (right). The numbers are the mean outcomes of 100 repetitions and bold entries mark the best result for each column.

Estimator	Aircraft		Antlion		Sinkholes		Crossroads		Gravity	
Robust LUI	14.89	8.3%	5.27	4.5%	57.05	3.6%	5.12	**0.0%**	8.27	3.9%
Robust PAC	**15.54**	8.2%	6.33	**2.8%**	46.59	3.8%	5.12	**0.0%**	−2.56	**0.0%**
MAP	13.24	**4.0%**	−8.99	8.0%	49.16	**3.0%**	5.12	**0.0%**	−2.35	**0.0%**
Optimistic LUI	14.17	5.2%	6.66	6.0%	67.78	4.1%	5.11	**0.0%**	21.40	19.6%
Optimistic PAC	14.35	6.8%	6.36	10.0%	65.35	3.8%	5.12	**0.0%**	23.51	47.8%
Unshielded	14.36	7.3%	6.62	9.0%	65.62	3.7%	**9.57**	40.1%	**30.35**	99.2%
Oracle	13.98	4.1%	**6.78**	5.6%	**73.32**	3.8%	5.12	**0.0%**	19.86	4.5%

Table 2. Comparison for the two exploration variants. Setup as in Table 1.

Exploration	Aircraft		Antlion		Sinkholes		Crossroads		Gravity	
Unif_A	14.89	**8.3%**	**5.27**	**4.5%**	57.05	**3.6%**	**5.12**	**0.0%**	**8.27**	3.9%
$\mathrm{Unif}_{\triangledown(s)}$	**15.04**	9.8%	1.93	5.1%	**57.29**	3.7%	5.00	**0.0%**	1.58	**0.6%**

This is because, unlike the static obstacles in the other environments, learning the behavior of the randomly moving opponent requires more data. In all other environments, $\triangledown_\theta$ is used most of the time after one or two model updates for all but the robust PAC estimator, which sometimes fails to obtain a sufficiently precise estimate (due to its higher data requirements).

(RQ5): What is the impact of the specific model estimator? We investigate how the different model estimators impact the results. Specifically, we compare the iMDP estimators PAC and LUI, both with the robust and the optimistic attitudes, and the MDP estimator MAP. We fix the priors of MAP to $w = 10$ and of LUI to $[\underline{n}_i, \overline{n}_i] = [5, 10]$, and the parameters of PAC to $\delta = 0.1$ and $\xi = 10^{-8}$. Varying priors as an additional dimension of experimental parameters is left for future research.

In Table 1, we show the evaluation of the final policies. Each cell shows the reward, evaluated empirically as the mean over 1000 episodes, and the relative safety of this policy, computed analytically using the PRISM model checker [26].[4]

Counterintuitively, robust estimators do not always lead to safer policies, as seen in the aircraft environment. Since the agent initially finds an imperfect route, which is considered relatively safe compared to less-explored states, the shield later forces the agent to stay on it, even if an unexplored yet safer alternative may exist. Instead, the optimistic shield specifications allow the agent to explore more states unless prior experience indicates that doing so is unsafe. The

[4] We did not use PRISM to compute the reward because it does not support a mix of positive and negative rewards.

Table 3. Comparison for different update delays u. Setup as in Table 1.

u	Aircraft		Antlion		Sinkholes		Crossroads		Gravity	
250	14.94	12.9%	−10.74	9.7%	78.71	5.4%	**5.12**	**0.0%**	−2.32	**0.0%**
500	**14.99**	10.5%	−10.46	11.0%	**84.78**	4.1%	**5.12**	**0.0%**	2.95	1.9%
1 000	14.89	8.3%	5.27	4.5%	57.05	**3.6%**	**5.12**	**0.0%**	8.27	3.9%
1 500	14.81	7.8%	**6.17**	**3.7%**	54.40	**3.6%**	**5.12**	**0.0%**	11.02	3.5%
2 000	14.72	**7.0%**	6.06	4.0%	57.93	**3.6%**	**5.12**	**0.0%**	**12.92**	8.2%

Table 4. Comparison for varying shield horizon (h in $\varphi|h$). Setup as in Table 1.

h	Aircraft		Antlion		Sinkholes		Crossroads		Gravity	
6	14.50	**8.2%**	**6.42**	**3.1%**	48.50	**3.3%**	**9.57**	40.1%	**24.61**	47.1%
12	14.78	**8.2%**	5.90	3.8%	54.81	3.4%	**9.57**	40.1%	23.16	33.3%
25	**14.89**	8.3%	5.71	4.2%	56.31	3.7%	**9.57**	40.1%	22.16	21.5%
50	**14.89**	8.3%	4.67	4.9%	57.12	3.4%	**9.57**	40.1%	19.36	13.2%
75	**14.89**	8.3%	4.39	4.4%	56.35	3.7%	**9.57**	40.1%	13.55	6.9%
100	**14.89**	8.3%	5.27	4.4%	57.05	3.7%	5.12	**0.0%**	8.27	4.0%
125	**14.89**	8.3%	5.37	4.0%	57.09	3.7%	5.12	**0.0%**	5.21	2.7%
150	**14.89**	8.3%	5.14	3.6%	57.07	3.9%	5.12	**0.0%**	3.81	2.5%
175	**14.89**	8.3%	5.42	3.6%	**57.19**	3.7%	5.12	**0.0%**	2.90	2.0%
200	**14.89**	8.3%	5.33	**3.1%**	57.18	3.6%	5.12	**0.0%**	1.24	**1.5%**

sinkholes and gravity environments show the biggest variation in the results. This is because the goals (checkpoints) in the sinkholes (gravity) environment are far apart, and hence finding a good route strongly depends on the exploration.

(RQ6): Is unshielded exploration beneficial? In Line 10 of Algorithm 1, the ε-greedy exploration chooses a random action from the set of all actions Unif_A, thus disregarding the shield. In Table 2, we compare this strategy to the alternative that only admissible actions are allowed in state s with $\mathrm{Unif}_{\triangledown(s)}$. Generally, with the latter variant, the agent eventually stops exploring new actions once the shield has found at least one admissible route. For some environments, this change has no significant effect on the reward because the route that was identified first was sufficiently good, while in the antlion and gravity environments, the restricted variant prevents the agent from uncovering more promising routes.

(RQ7): How often should the shield be updated? The model estimate and subsequent shield update are the most expensive operations in our approach. We examine the effect of the update delay u. Table 3 shows that the choice of u can be impactful, but no single choice is more preferable across the environments.

(RQ8): What is the impact of the shield lookahead (h)? We examine the effect of varying the lookahead horizon h of the shield. Table 4 shows the results. Aircraft episodes end after 20 steps, making longer horizons redundant. The gravity

environment is less safe at lower lookahead. In the crossroads environment, for $h \leq 75$, the agent prefers the more rewarding (but less safe) route.

6 Conclusion and Future Work

In this paper, we have proposed the paradigm of *adaptive probabilistic shielding* for safe reinforcement learning. We assume only access to a nondeterministic environment model (i.e., we do not know the transition probabilities) and consequently do not have access to a simulator. In settings where safety violations are costly, safe exploration is a challenge. We tackle this challenge with a practical, integrated procedure that simultaneously explores the environment, updates a model estimate, maintains a probabilistic shield, and learns a policy under that shield. Our focus has been on the empirical evaluation, in which we investigated various research questions regarding the success and impact of our design choices.

While our shield implementation is based on a recent approach for interval MDPs [12], the procedure can be extended to support other shields. One direction is to vary the safety thresholds of the shields computed during the course of the algorithm, which we have kept to a user-defined constant in our approach. For instance, we imagine first using an optimistic shield that encourages exploration and then gradually raising the safety threshold to obtain a safer shield in the end. As another direction, one can incorporate more elements from model-based RL algorithms. For instance, we may replace the uniformly random exploration step with a biased choice toward under-explored states and actions.

Acknowledgments. This research was partly supported by the European Research Council (ERC) Starting Grant 101077178 (DEUCE), the Villum Investigator Grant S4OS under reference number 37819, and the Independent Research Fund Denmark under reference number 10.46540/3120-00041B.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: AAAI. pp. 2669–2678. AAAI Press (2018). https://doi.org/10.1609/AAAI.V32I1.11797
2. Ashok, P., Kretínský, J., Weininger, M.: PAC statistical model checking for Markov decision processes and stochastic games. In: CAV. LNCS, vol. 11561, pp. 497–519. Springer (2019). https://doi.org/10.1007/978-3-030-25540-4_29
3. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
4. Bethell, D., Gerasimou, S., Calinescu, R., Imrie, C.: Safe reinforcement learning in black-box environments via adaptive shielding. In: ECAI. pp. 2450–2457. FAIA, IOS Press (2025). https://doi.org/10.3233/FAIA251092
5. Brorholt, A.H., Jensen, P.G., Larsen, K.G., Lorber, F., Schilling, C.: Shielded reinforcement learning for hybrid systems. In: AISoLA. pp. 33–54. LNCS, Springer (2023). https://doi.org/10.1007/978-3-031-46002-9_3

6. Brorholt, A.H., Larsen, K.G., Schilling, C.: Compositional shielding and reinforcement learning for multi-agent systems. In: AAMAS. pp. 399–407. IFAAMAS (2025), https://dl.acm.org/doi/10.5555/3709347.3743554
7. Brorholt, A.H., Galesloot, M.F.L.: Code and data for "Adaptive probabilistic shielding by learning MDPs for safe reinforcement learning" (2026). https://doi.org/10.5281/zenodo.21874278
8. Carr, S., Bakirtzis, G., Topcu, U.: Compositional shield synthesis for safe reinforcement learning in partial observability. CoRR **abs/2509.12085** (2025). https://doi.org/10.48550/ARXIV.2509.12085
9. David, A., Jensen, P.G., Larsen, K.G., Legay, A., Lime, D., Sørensen, M.G., Taankvist, J.H.: On time with minimal expected cost! In: ATVA. LNCS, vol. 8837, pp. 129–145. Springer (2014). https://doi.org/10.1007/978-3-319-11936-6_10
10. Elsayed-Aly, I., Bharadwaj, S., Amato, C., Ehlers, R., Topcu, U., Feng, L.: Safe multi-agent reinforcement learning via shielding. In: AAMAS. pp. 483–491. ACM (2021). https://doi.org/10.5555/3463952.3464013
11. Feng, Y., Zhu, J., Platzer, A., Laurent, J.: Adaptive shielding via parametric safety proofs. Proc. ACM Program. Lang. **9**(OOPSLA1), 816–843 (2025). https://doi.org/10.1145/3720450
12. Galesloot, M.F.L., Rhemrev, T., Jansen, N.: Robust probabilistic shielding for safe offline reinforcement learning. CoRR **abs/2605.10293** (2026). https://doi.org/10.48550/ARXIV.2605.10293
13. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. **16**, 1437–1480 (2015). https://doi.org/10.5555/2789272.2886795
14. Goodall, A.W., Belardinelli, F.: Approximate model-based shielding for safe reinforcement learning. In: ECAI. FAIA, vol. 372, pp. 883–890. IOS Press (2023). https://doi.org/10.3233/FAIA230357
15. Hamel-De le Court, E., Badings, T., Abate, A., Belardinelli, F., Fabiano, F.: Robust shielding for safe reinforcement learning. CoRR **abs/2606.00270** (2026). https://doi.org/10.48550/ARXIV.2606.00270
16. Hamel-De le Court, E., Belardinelli, F., Goodall, A.W.: Probabilistic shielding for safe reinforcement learning. In: AAAI. pp. 16091–16099. AAAI Press (2025). https://doi.org/10.1609/AAAI.V39I15.33767
17. Heck, L., Macák, F., Andriushchenko, R., Češka, M., Junges, S.: Shields to guarantee probabilistic safety in mdps. In: CAV (2026), https://arxiv.org/abs/2605.10888
18. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963), http://www.jstor.org/stable/2282952
19. Huynh, K.V., Parker, D., Feng, L.: Robust permissive controller synthesis for interval mdps. CoRR **abs/2510.03481** (2025)
20. Jaeger, M., Bacci, G., Bacci, G., Larsen, K.G., Jensen, P.G.: Approximating Euclidean by imprecise Markov decision processes. In: ISoLA. LNCS, vol. 12476, pp. 275–289. Springer (2020). https://doi.org/10.1007/978-3-030-61362-4_15
21. Jansen, N., Könighofer, B., Junges, S., Serban, A., Bloem, R.: Safe reinforcement learning using probabilistic shields (invited paper). In: CONCUR. pp. 3:1–3:16. LIPIcs (2020). https://doi.org/10.4230/LIPICS.CONCUR.2020.3
22. Jonsson, B., Larsen, K.G.: Specification and refinement of probabilistic processes. In: LICS. pp. 266–277. IEEE Computer Society (1991). https://doi.org/10.1109/LICS.1991.151651
23. Kim, K., Corsi, D., Rodríguez, A., Lanier, J., Parellada, B., Baldi, P., Sánchez, C., Fox, R.: Realizable continuous-space shields for safe reinforcement learning. In:

L4DC. pp. 932–945. PMLR (2025), https://proceedings.mlr.press/v283/kim25c.html
24. Kochenderfer, M.: Decision Making Under Uncertainty. MIT Press (2015)
25. Könighofer, B., Bloem, R., Jansen, N., Junges, S., Pranger, S.: Shields for safe reinforcement learning. Commun. ACM **68**(11), 80–90 (2025). https://doi.org/10.1145/3715958
26. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV. LNCS, vol. 6806, pp. 585–591. Springer (2011). https://doi.org/10.1007/978-3-642-22110-1_47
27. Lacerda, B., Faruq, F., Parker, D., Hawes, N.: Probabilistic planning with formal performance guarantees for mobile service robots. Int. J. Robotics Res. **38**(9) (2019). https://doi.org/10.1177/0278364919856695
28. Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D., Peters, J.: Robust reinforcement learning: A review of foundations and recent advances. Mach. Learn. Knowl. Extr. **4**(1), 276–315 (2022). https://doi.org/10.3390/MAKE4010013
29. Nilim, A., Ghaoui, L.E.: Robust control of Markov decision processes with uncertain transition matrices. Oper. Res. **53**(5), 780–798 (2005). https://doi.org/10.1287/OPRE.1050.0216
30. Pnueli, A.: The temporal logic of programs. In: FOCS. pp. 46–57. IEEE Computer Society (1977). https://doi.org/10.1109/SFCS.1977.32
31. Pranger, S., Könighofer, B., Tappler, M., Deixelberger, M., Jansen, N., Bloem, R.: Adaptive shielding under uncertainty. In: ACC. pp. 3467–3474. IEEE (2021). https://doi.org/10.23919/ACC50511.2021.9482889
32. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics, Wiley (1994). https://doi.org/10.1002/9780470316887
33. Reed, R., Lahijanian, M.: Learning-based shielding for safe autonomy under unknown dynamics. In: ACC. pp. 4940–4946. IEEE (2025)
34. Senthilvelan, P., Li, J., Tei, K.: Similarity-based shield adaptation under dynamic environment. In: SEAI. pp. 33–39. IEEE (2023). https://doi.org/10.1109/SEAI59139.2023.10217461
35. Strehl, A.L., Littman, M.L.: An analysis of model-based interval estimation for Markov decision processes. J. Comput. Syst. Sci. **74**(8), 1309–1331 (2008). https://doi.org/10.1016/J.JCSS.2007.08.009
36. Suilen, M., Badings, T., Bovy, E.M., Parker, D., Jansen, N.: Robust Markov decision processes: A place where AI and formal methods meet. In: Principles of Verification (3). pp. 126–154. LNCS, Springer (2024). https://doi.org/10.1007/978-3-031-75778-5_7
37. Suilen, M., Simão, T.D., Parker, D., Jansen, N.: Robust anytime learning of Markov decision processes. In: NeurIPS (2022), https://doi.org/10.52202/068431-2087
38. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (2018), http://www.incompleteideas.net/book/the-book-2nd.html
39. Tappler, M., Pranger, S., Könighofer, B., Muskardin, E., Bloem, R., Larsen, K.G.: Automata learning meets shielding. In: ISoLA. LNCS, vol. 13701, pp. 335–359. Springer (2022). https://doi.org/10.1007/978-3-031-19849-6_20
40. Walter, G., Augustin, T.: Imprecision and prior-data conflict in generalized Bayesian inference. J. Stat. Theory Pract. **3**(1), 255–271 (2009)
41. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**, 279–292 (1992). https://doi.org/10.1007/BF00992698