Pilot Experimentation for Safe Exploration and Side-Effects Avoidance

C Henrik Åslund, Alessio Lomuscio

Imperial College London
c.aslund19@imperial.ac.uk, a.lomuscio@imperial.ac.uk

Abstract

We investigate the problem of safe exploration and side-effects avoidance in reinforcement learning agents. Specifically, the agents are deployed in certain kinds of factored environments and initialised with partial prior knowledge concerning the factoring. We introduce a novel model of the resulting agent–environment systems based on experimentation in, e.g., the medical, agricultural, material, and economic sciences, and how regulations in these domains limit both expected and unexpected adverse side-effects. We define a constrained optimisation problem for these systems as eventually maximising the expected return always subject to the regulations, which are specified as constraints in a variant of probabilistic computation tree logic. We introduce a model-based reinforcement learning algorithm solving this problem through runtime shielding, and we show that the learning process is always safe according to the regulations with high probability. We show that safety is traded off for slower convergence for early time steps, but that the regret approaches that of theoretically efficient procedures for late time steps. We illustrate how similar regulations could work in recommender engines through simulations, in which we experimentally corroborate the theoretical results and show that our algorithm outperforms the state-of-the-art.

1 Introduction

Reinforcement learning (RL) has seen considerable advances in applications such as games (Silver et al. 2017), robotics (Andrychowicz et al. 2020), fine-tuning of language models (OpenAI 2023), recommender engines, and the automation of science (Degrave et al. 2022). An RL agent performs optimisation in an environment, which is initially unknown and learned about through exploration. In games and other kinds of simulations, there is a large body of work that eventually achieves safety (Altman 1999; Ray, Achiam, and Amodei 2019). However, there is (i) no requirement for safety during exploration, and (ii) no problem with unexpected side-effects as they can be incorporated in future simulations. As for (i)—safe exploration (Brunke et al. 2021; García and Fernández 2015)—current methods either rely on the existence of faithful simulations (Mqirmi, Belardinelli, and León 2021, e.g.) or on strong assumptions about the environment (Koller et al. 2018, e.g.). As for (ii)—side-effects avoidance (Saisubramanian, Zilberstein, and Kamar 2021)—current methods lack strong formal gurantees (Turner, Ratzlaff, and Tadepalli 2020, e.g.). Adverse side-effects on health or the environment pose risks to safety in automated science, and side-effects such as misinformation or polarisation pose risks in news recommender engines; the assumptions of current methods are too strong for such applications. Rapid advances in machine learning, together with insufficient solutions, have even led some academics to warn of a global catastrophic risk (Bostrom 2014; Russell 2019; El Mahdi et al. 2019).

Safe exploration and side-effects avoidance are impossible without (at least partial) prior knowledge (Turchetta, Berkenkamp, and Krause 2016). Strong guarantees for safe exploration come from *shielding*, but they often require the environment dynamics as prior knowledge (Könighofer et al. 2022; Odriozola-Olalde, Zamalloa, and Arana-Arexolaleiba 2023). Only requiring prior knowledge of some structural elements of the environment is promising (Berkenkamp et al. 2017; Simão, Jansen, and Spaan 2021). Side-effects avoidance methods need to assume prior knowledge of the decision rules of human overseers (Armstrong and Mindermann 2018). Approaches applied to the automation of science, where *experimentation* may be used as a synonym of exploration, and recommender engines are lacking from the literature and motivate this work.

We make four key contributions:

1. We introduce a novel optimisation problem, which we call pilot experimentation. Pilot experimentation is the joint problem of safe exploration and side-effects avoidance in a novel model, which we call a cellular Markov decision process (cellular MDP). A constrained MDP is combined with factored MDPs, whereby the concepts of cells and regulators emerge. Cells model, e.g., different people or separate regions of the environment. Regulators model, e.g., regulatory agencies or moderators.
2. To solve the pilot experimentation problem, we introduce an algorithm, PE-UCRL, which is an upper confidence RL algorithm that contains a novel shielding subroutine.
3. We prove that PE-UCRL solves the pilot experimentation problem with high probability.
4. We illustrate how similar regulations could work in recommender engines through simulations, in which we ex-

perimentally corroborate the theoretical results and show that our algorithm outperforms the state-of-the-art.

The rest of the paper is organised as follows. we present related work below and preliminaries in Section 2. We define pilot experimentation in Section 3 and PE-UCRL in Section 4. We present theoretical results in Section 5 and experimental results in Section 6. We conclude in Section 7.

Related work. Safe exploration is concerned with safety during exploration in the real world. An early approach was ergodicity-preservation, i.e., regularising the reward function such that the RL agent is always able to return to the initial state (Martínez, Alenyà, and Torras 2015; Moldovan and Abbeel 2012). Such methods require much prior knowledge, which lead to approaches relying on the smoothness of the environment (Berkenkamp et al. 2017; Koller et al. 2018; Turchetta, Berkenkamp, and Krause 2016), where only the corresponding Lipschitz-constants are required as prior knowledge. Such methods are applicable to robotics, but for many other domains, the Lipschitz-bounds get prohibitively tight. The approach most similar to ours is to assume factored MDPs and constrain the RL agent to limit how much it alters (if at all) certain variables (Simão, Jansen, and Spaan 2021; Farquhar, Carey, and Everitt 2022).

Side-effects avoidance is concerned with unexpected adverse side-effects and has also been studied under the rubrics of low-impact methods and value alignment (Christiano et al. 2017; Hoang 2019; Kori, Glocker, and Toni 2022; Lindner et al. 2021; Saunders et al. 2018). An example is cooperative inverse RL: An RL agent is modelled as assisting its human user by learning about the reward function of the user (Hadfield-Menell et al. 2016, 2017; Shah et al. 2019). Instead of an RL agent–user teams, our work concerns RL agents learning human values via existing regulatory institutions. Closest to our work are methods where the impact on the environment regularises the reward (Krakovna et al. 2019; Turner, Ratzlaff, and Tadepalli 2020).

2 Preliminaries

In this section, we fix the notation. First, we describe the MDP model and extensions, second, the RL problem, and third, probabilistic computation tree logic (PCTL).

An RL agent interacting with an environment can be modelled as an MDP (Puterman 1994). An MDP M is a tuple

$$M := (\mathbf{S}, \mathbf{s}^{\text{init}}, \mathbf{A}, P, R) \qquad (1)$$

where: $\mathbf{S}$ is a finite set denoting the state space. $\mathbf{s}^{\text{init}}$ is a distinguished state denoting the initial state. $\mathbf{A}$ is a finite set denoting the action space. For any set $\mathbf{X}$, let $\mathit{Distr}(\mathbf{X})$ denote the set of probability distributions over $\mathbf{X}$. $P : \mathbf{S} \times \mathbf{A} \to \mathit{Distr}(\mathbf{S})$ is a probability function denoting the transition function. We write $P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$ for the probability of the current state being $\mathbf{s}'$ given that the RL agent took action $\mathbf{a}$ in the last state $\mathbf{s}$. Note that not all actions in $\mathbf{A}$ are necessarily available in every state. Precisely, $\sum_{\mathbf{s}' \in \mathbf{S}} P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) = 0$ if action $\mathbf{a}$ is not available in state $\mathbf{s}$, and $\sum_{\mathbf{s}' \in \mathbf{S}} P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) = 1$ if $\mathbf{a}$ is available in $\mathbf{s}$. The set of all available action in $\mathbf{s}$ is denoted by $\mathbf{A}(\mathbf{s})$. $R : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to [0, 1]$ denotes the reward function. We write $r = R(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ for the reward following a transition $(\mathbf{s}, \mathbf{a}, \mathbf{s}')$. We use **boldface** for lists and sets (but not tuples). (The reason for denoting states and actions as lists is explained below.)

Extensions of the MDP model include (i) constrained MDPs (Altman 1999) and (ii) factored MDPs (Boutilier, Dearden, and Goldszmidt 1995). (i) A constrained MDP contains an additional element $\mathbf{C} : \mathbf{S} \to \mathbf{X}$. If $\mathbf{X} = \mathbb{R}^d$, where $d \in \mathbb{N}$, $\mathbf{C}$ is called a cost function. If $\mathbf{X}{=}2^{\mathbf{AP}}$, where $\mathbf{AP}$ is a finite set denoting atomic propositions, $\mathbf{C}$ is called a labelling function. A constraint is a proposition, i.e. a symbol sequence that takes a truth value. The symbols can either be values $\mathbf{c} \in \mathbf{X}$ or operators. (ii) In a factored MDP, P can be factorised such that $P{=}\prod_{i \in [n]} P_i$, where $[n]{:=}\{1, ..., n\}$, and $n \in \mathbb{N}$ is a fixed number. This requires assuming that the state and action spaces have certain structures. We simplify factored MDPs from the literature and consider the structure where $\mathbf{S} = \times_{i \in [n]} \mathbf{S}_i$ and $\mathbf{A} = \times_{i \in [n]} \mathbf{A}_i$ (states and actions are lists). Then, $P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) = \prod_{i \in [n]} P_i(s'_i \mid \mathbf{s}, a_i)$, where $\mathbf{s} \in \mathbf{S}$, $s'_i \in \mathbf{S}_i$, and $a_i \in \mathbf{A}_i$.

In RL problems, it is assumed that the developer of the RL agent knows the identities of some of the elements in the MDP but not others. A common division is

$$\text{Known: } \mathbf{S}, \mathbf{s}^{\text{init}}, \mathbf{A}. \text{ Unknown: } P, R. \qquad (2)$$

However, sometimes, R is known, and, at other times, P is known instead. The unknown elements are what the RL agent should learn. A deterministic policy is a function $\boldsymbol{\pi} : \mathbf{S} \to \mathbf{A}$, where $\mathbf{a} = \boldsymbol{\pi}(\mathbf{s})$ means that the RL agent takes action $\mathbf{a}$ in state $\mathbf{s}$. Policies are updated through learning, and, therefore, an RL agent can be seen as a sequence of policies. We denote the policy at time $t \in \mathbb{N}$ by $\boldsymbol{\pi}^t$. Note that $\boldsymbol{\pi}^t$ is a random variable. (Depending on the randomly generated history of states and actions, $\boldsymbol{\pi}^t$ can take different values.) Continuing, we write superscript t to denote random variables associated with time t also for other variables, e.g. states $\mathbf{s}^t$, actions $\mathbf{a}^t$, and rewards r^t. The unconstrained RL problem is to find a sequence of policies $\boldsymbol{\pi} = \boldsymbol{\pi}^t$ for all $t \in \mathbb{N}$ in order to

$$\begin{array}{c}\text{eventually}\\ \text{maximise}\end{array} \ \frac{1}{T}\mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{\tau=0}^{T} r^{\tau} \;\middle|\; M, \mathbf{s}^0{=}\mathbf{s}^{\text{init}}\right] \ \begin{array}{c}\text{initially given}\\ \text{Equation (2)}\end{array}$$

The symbol $\mathbb{E}_{\boldsymbol{\pi}}$ denotes the expectation under policy $\boldsymbol{\pi}$. $T \in \mathbb{N}$ is the time horizon. In episodic RL, T is finite (Osband and Roy 2017). In the context of safe exploration (for learning outside of simulations), we consider average RL, where $T \to \infty$.

Constraints in MDPs can be expressed in PCTL (Ciesinski and Größer 2004). PCTL contains propositional logic operators, e.g. $\neg$ and $\wedge$, and derived operators derived, e.g. $\vee$ and $\Rightarrow$. In addition, PCTL includes temporal operators, such as $\bigcirc$ and a quantifier $[\cdot]_{\leq q}$. The expression $\bigcirc\phi$ is read as 'in the next state, ϕ', and $[\phi]_{\leq q}$ is read as 'ϕ has a probability $\leq q$'. The syntax is given by the Backus–Naur form

$$\phi ::= ap \mid \psi \mid (\neg\phi) \mid (\phi \wedge \phi) \mid [\psi]_{\leq q}, \quad \psi ::= \bigcirc\phi, \qquad (3)$$

where $ap \in \mathbf{AP}$ and $q \in [0, 1]$. Let $\mathbf{M}$ be a set containing different MDPs $\tilde{M} \in \mathbf{M}$, which only differ in terms of their corresponding transition functions $\tilde{P}$. (We use $\sim$ to denote that $\tilde{X}$ may or may not be the true X.) Let $\models_{\boldsymbol{\pi}}$ be the satisfaction relation under policy $\boldsymbol{\pi}$. The semantics are below.

$$\begin{aligned}
&\mathbf{M}, \mathrm{s} \models_{\boldsymbol{\pi}} ap && \text{iff} \quad ap \in \mathbf{AP}(\mathrm{s}) \\
&\mathbf{M}, \mathrm{s} \models_{\boldsymbol{\pi}} \neg\phi && \text{iff} \quad \mathbf{M}, \mathrm{s} \not\models_{\boldsymbol{\pi}} \phi \\
&\mathbf{M}, \mathrm{s} \models_{\boldsymbol{\pi}} \phi \wedge \psi && \text{iff} \quad \mathbf{M}, \mathrm{s} \models_{\boldsymbol{\pi}} \phi \text{ and } \mathbf{M}, \mathrm{s} \models_{\boldsymbol{\pi}} \psi \\
&\mathbf{M}, \mathrm{s} \models_{\boldsymbol{\pi}} [\phi]_{\leq q} && \text{iff} \quad \mathbb{P}[\mathbf{M}, (\mathrm{s}^{\tau})_{\tau=0}^{\infty} \models_{\boldsymbol{\pi}} \phi \mid \mathrm{s}] \leq q \\
&\mathbf{M}, (\mathrm{s}^{\tau})_{\tau=0}^{\infty} \models_{\boldsymbol{\pi}} \bigcirc\phi && \text{iff} \quad \mathbf{M}, \mathrm{s}^{1} \models_{\boldsymbol{\pi}} \phi
\end{aligned} \tag{4}$$

Note that $\mathbb{P}$ refers to the probability under the worst-case choice of $\tilde{P}$ and that $\mathbf{AP}(\mathrm{s}) \in 2^{\mathbf{AP}}$ is the set of true atomic propositions in s. $\mathbf{AP}(\mathrm{s})$ is precisely defined in Section 3.

3 Pilot Experimentation Problem

In this section, we introduce the cellular MDP model. First, we describe its novel elements, cells and regulators, and motivate and illustrate them with examples from medical research, agricultural science, and recommender engines. Second, we define a cellular MDP as a tuple. Then, we motivate and define assumptions on partial prior knowledge. With the model and the assumptions, we define a novel constrained RL problem: pilot experimentation.

MDP model. In a cellular MDP, the environment is composed of *cells*. Cells are special cases of variables in factored MDPs. Cells are like variables in factored MDPs in that: (i) There is a fixed number $n \in \mathbb{N}$ of cells. (ii) Each cell $i \in [n]$ is associated with a factored state space $\mathbf{S}_i$ and a factored action space $\mathbf{A}_i$, which we call the intracellular state space and intracellular action space, respectively. Cells are different in that: (iii) The factored transitions are independent of intracellular states in other cells, i.e. $P_i(\cdot \mid s_i, a_i)$ is independent of $P_j(\cdot \mid s_j, a_j)$. Furthermore, $P_i = P_j =: P_{\#}$, where we call $P_{\#}$ the intracellular transition function. (Different transition dynamics in different cells can be modelled by augmenting the intracellular state spaces.) (iv) Cells are classified as belonging to one or more classes from a set of cell classes $\mathbf{G}$. In medical research, cells can model potential participants in a medical trial. Then, (iii) models that an experimental treatment given to one participant does not affect another participant who receives a different treatment. In (iv), examples of important classes to include in $\mathbf{G}$ are whether a potential participant has consented to participate in the trial, or whether they are pregnant. In agricultural science, cells can model different regions separated by land–water boundaries, large distances, or artificial laboratory structures such as in mesocosms. In recommender engines, cells can model users. For further details on examples, see Appendix A.

Also unlike variables in factored MDPs, cells are associated with *regulators*. A regulator can label a state as unsafe or safe; we say that the regulator reports an adverse side effect or no adverse side-effects, respectively. Regulators work like cost functions or labelling functions and side-effects work like costs or labels in constrained MDPs in that: (v) If $\mathbf{C}$ denotes the regulators, there is an $\mathbf{X}$ such that $\mathbf{C}$:$\mathbf{S}$→$\mathbf{X}$, and $\mathbf{c}$∈$\mathbf{X}$ is the side-effects. Regulators are different in that: (vi) Each regulator is associated with a cell. $\mathbf{X} = \mathit{Distr}\{\mathit{safe}, \mathit{unsafe}, \mathit{silent}\}^{n \times n}$, which means that $\mathbf{C}$ can be seen as a list $[\mathbf{C}_1, ..., \mathbf{C}_n]$, where $\mathbf{C}_i$ is the regulator associated with cell i. Let, e.g., $n = 2$, then $[\mathit{safe}, \mathit{unsafe}] \sim \mathbf{C}_1(\cdot \mid [s_1, s_2])$ means that the regulator in cell 1 is safe but reports adverse side-effects in cell 2. (vii) The outcome *silent* is special in the sense that nothing can be deduced about the safety of the intracellular state on this information alone. Furthermore, the absence of a regulator in a cell i can be modelled by letting $\mathbf{C}_i(\mathrm{s}) = \mathit{Distr}\{\mathit{silent}\}^n$ for all $\mathrm{s} \in \mathbf{S}$. In medical research, a regulator can model a regulatory agency or a single individual working within a regulatory agency. In the latter case, every regulator is a potential participant (vi), but not every potential participant is a regulator (vii). In agricultural science, it would be appropriate to only model regulatory agencies. In recommender engines, regulators can model moderators.

With cells and regulators define, we can summarise the definition of the cellular MDP as follows. A graphical representation is available in Appendix B.

Definition 1 (Cellular MDP)**.** A cellular MDP CM is a tuple

$$\mathit{CM} := (n, \mathbf{S}_1, ..., \mathbf{S}_n, \mathrm{s}^{\mathrm{init}}, \mathbf{A}, P_{\#}, R, \mathbf{C}, \mathbf{G}, L),$$

where: $n \in \mathbb{N}$ denotes the number of cells. $\mathbf{S}_1, ..., \mathbf{S}_n$ are finite sets denoting the intracellular state spaces. Letting $\mathbf{S} = \times_{i \in [n]} \mathbf{S}_i$, $\mathrm{s}^{\mathrm{init}}$ and R are defined as in Equation (1). So is $\mathbf{A}$ but with the additional assumption that there exists a finite set $\mathbf{A}_{\#}$ such that $\mathbf{A}(\mathbf{s}) \subseteq \times_{i \in [n]} \mathbf{A}_{\#}$ for all $\mathrm{s} \in \mathbf{S}$. $\mathbf{A}_{\#}$ denotes the intracellular action space. $P_{\#} : \mathbf{S}_{\#} \times \mathbf{A}_{\#} \to \mathit{Distr}(\mathbf{S}_{\#})$, where $\mathbf{S}_{\#} := \bigcup_{i \in [n]} \mathbf{S}_i$, denotes the intracellular transition function. $\mathbf{C}$ is a matrix probability distribution, where each element is defined by

$$C_{ij} : \mathbf{S}_i \times \mathbf{S}_j \to \mathit{Distr}\{\mathit{safe}, \mathit{unsafe}, \mathit{silent}\}.$$

The function $\mathbf{C}$ denotes the regulators and its outcomes are called side-effects. $\mathbf{G}(\ni \mathit{all})$ is a finite set denoting the cell classes. $L : [n] \to 2^{\mathbf{G}}$ ($\mathit{all} \in L(i)$ for all $i \in [n]$) denotes the cell classes.

Partial prior knowledge. In contrast to Equation (2), the division between what the developers know and do not know is as follows.

$$\text{Known:} n, \mathbf{S}_1, ..., \mathbf{S}_n, \mathrm{s}^{\mathrm{init}}, \mathbf{A}, R, \mathbf{G}, L. \text{ Unknown:} P_{\#}, \mathbf{C}. \tag{5}$$

It is easy to extend to cases where R is unknown, but, for ease of presentation, it is assumed to be known. Despite the identities of $P_{\#}$ and $\mathbf{C}$ being unknown, there is partial prior knowledge about them, that can be represented as constraints, which we call *physical constraints*. First, the developer knows which intracellular states and actions are associated with which independent cell. In medical research, a medical firm would know which participants are given which treatments, and it is similar within agricultural science. In recommender engines, the developer could track which users are given which recommendations. Second, the regulators do not disagree whether a consequence is safe or unsafe (a strong assumption), and the developer knows this.

Definition 2 (Physical Constraints)**.** The first physical constraint is cellular independence. Under cellular independence, there exist $P,P_\#$ such that, for all $\mathbf{s},\mathbf{s}'\in\mathbf{S}$ and $\mathbf{a}\in\mathbf{A}$,

$$P(\mathbf{s}' \mid \mathbf{s},\mathbf{a}) = \prod_{i\in[n]} P_\#(s'_i \mid s_i, a_i).$$

The second physical constraint is consistent regulators. Under consistent regulators, for all $\mathbf{s} \in \mathbf{S}$ and $h, i, j \in [n]$, both of the following hold.

$$\mathbb{P}[\mathit{unsafe} \sim C_{hi}(\cdot \mid s_h, s_i) \mid \mathit{safe} \sim C_{hj}(\cdot \mid s_h, s_j)] = 0$$
$$\mathbb{P}[\mathit{safe} \sim C_{hi}(\cdot \mid s_h, s_i) \mid \mathit{unsafe} \sim C_{hj}(\cdot \mid s_h, s_j)] = 0$$

In contrast to physical constraints, *regulatory constraints* are not dictated by how the world works but can be decided by society. Complete safety cannot be guaranteed while simultaneously allowing for learning. Therefore, regulatory constraints in medical research and agricultural science work by first allowing small experiments and gradually scaling these up. Regulations in recommender engines are less developed and are to a large extent based on self-regulation. In contrast, regulations in medical research are highly developed. For example, Friedman et al. (2015) include: First entering three participants into the clinical trial. Wait and observe if adverse side-effects occur and, in that case, potentially end the trial. Exclude, e.g., pregnant persons from trials. For further details, see Appendix A. We note three important parameters: (i) how long to wait to observe adverse side-effects such as toxicity, (ii) the number of participants, and (iii) whether the participants belong to certain classes. We formalise the parameters using a variant of PCTL, which we call pilot experimentation PCTL (PE-PCTL). (i) does not require any modification of PCTL, but it is helpful to define the operator

$$\bigcirc^{\leq\tau} := \bigwedge_{\sigma\in[\tau]} \bigcirc^{\sigma} \text{ where } \bigcirc^{\sigma+1} := \bigcirc\bigcirc^{\sigma}, \bigcirc^{1} := \bigcirc$$

where τ is how long to wait. To formalise (ii) and (iii), we introduce the symbol $\text{№}_g \geq j$, where $g \in 2^{\mathbf{G}}$ and $j \in [n]$. j denotes the bound on the number of cells allowed to experience adverse side-effects. g denotes the cell classes this applies to.

Definition 3 (Regulatory Constraints)**.** PE-PCTL is PCTL, where ap in Equation (3) is replaced by $\text{№}_g \geq j$ and Equation (4) is replaced by

$$\mathbf{M}, \mathbf{s} \models_{\boldsymbol{\pi}} \text{№}_g \geq j \text{ iff } \sum_{i\in[n]} \mathbb{I}[g \in L(i)]\mathbb{I}[\mathit{isUnsafe}(s_i)] \geq j$$

where $\mathit{isUnsafe}(s_i) \equiv \mathit{true}$ iff $\prod_{h\in[n]} C_{hi}(\mathit{unsafe}|s'_h) > 0$ for all $\mathbf{s}' \in \{\mathbf{s}' \in \mathbf{S} : s'_i = s_i\}$. If $\varPhi$ is a formula in PE-PCTL, then it is a regulatory constraint.

The final piece of prior knowledge is that the developer knows the identity of a distinguished policy $\boldsymbol{\pi}^{\text{init}}$ denoting the initial policy. Furthermore, the developer knows that $\boldsymbol{\pi}^{\text{init}}$ is safe, although $\boldsymbol{\pi}^{\text{init}}$ may be very far from optimal. It is a common assumption in safe exploration (Biyik et al. 2019; Koller et al. 2018; Roderick, Nagarajan, and Kolter 2021, e.g.). For example, $\boldsymbol{\pi}^{\text{init}}$ could be a foundation model acquired through supervised learning. In the examples of medical research or agricultural science, there are usually default treatments known to be safe.

Definition 4 (Safe Initial Policy)**.** The initial policy is $\boldsymbol{\pi}^{\text{init}}$. Let $\mathbf{S}^{\infty}$ be the set of states that are reachable from $\mathbf{s}^{\text{init}}$ under $\boldsymbol{\pi}^{\text{init}}$. For all $\mathbf{s}\in\mathbf{S}^{\infty}$ and $i,j\in[n]$, $C_{ij}(\mathit{unsafe}|s_i,s_j)=0$.

Safe exploration and side-effects avoidance. A safe exploration problem is a constrained reinforcement learning problem of eventually maximising reward while always satisfying the safety constraints. The pilot experimentation problem is a safe exploration problem applicable in cellular MDPs with the prior knowledge above. Unlike other safe exploration problems, it also covers side-effects not originally specified.

Definition 5 (Pilot Experimentation Problem)**.** Let pk be an object containing the prior knowledge in Equation (5) and definitions 2 and 4, and let $\varPhi$ be a regulatory constraint. Find a sequence of policies $\boldsymbol{\pi} = \boldsymbol{\pi}^t$ for all $t \in \mathbb{N}$ in order to

$$\begin{array}{rll} \text{eventually maximise} & \lim\limits_{T\to\infty} \dfrac{1}{T}\mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{\tau=0}^{T} r^{\tau} \,\middle|\, CM, \mathbf{s}^0 = \mathbf{s}^{\text{init}}\right] & \\ \text{always subject to} & \{CM\}, \mathbf{s}^{\text{init}} \models_{\boldsymbol{\pi}} \varPhi & \text{initially given } pk \end{array}$$

4 Shielded RL Algorithm

In this section, we introduce a procedure for solving the pilot experimentation problem. First, we adapt upper confidence RL algorithms to cellular MDPs to get a baseline algorithm. Then, we adapt the baseline algorithm further by making it avoid adverse side-effects and shielding violations to the regulatory constraints.

Cellular Ucrl. Upper confidence RL algorithms, such as UCRL2 (Jaksch, Ortner, and Auer 2010) and KL-UCRL (Filippi, Cappé, and Garivier 2010), can be directly applied to cellular MDPs. Direct application does, however, result in inefficient learning: The RL agent ignores the cellular independence in Definition 2 from the prior knowledge pk and, therefore, fails to do a certain form of transfer learning. We call it cellular transfer learning, and it consists in observing an intracellular transition in one cell and using that to learn about intracellular transitions in other cells. Upper confidence RL algorithms can be split into two alternating phases: the *on-policy* phase, where the RL agent interacts with the environment, and the *off-policy* phase, where it computes the policy update. Cellular transfer learning can be enabled by modifying each phase. In the on-policy phase, we modify *switchPhase*, the condition when to switch to the off-policy phase. In the off-policy phase, we modify the calculation of the confidence set $\mathbf{CM}$. Continuing, we pick UCRL2 as the specific algorithm we build upon, and in comparison with UCRL2

$$\mathbf{CM} \subseteq \mathbf{CM}_{\text{UCRL2}}, \mathit{switchPhase} \Leftarrow \mathit{switchPhase}_{\text{UCRL2}}.$$

We call the resulting algorithm Cellular Ucrl, and details are available in Appendix C.

Pilot Experimentation UCRL (Pe-Ucrl). A problem with Cellular UCRL is that it ignores side-effects. We modify both the on-policy and off-policy phases and call the resulting algorithm Pe-Ucrl. In the on-policy phase, we

```
Algorithm 1: PE-SHIELD
Input: last behaviour policy $\boldsymbol{\pi}^{\Phi}$, target policy $\boldsymbol{\pi}^{+}$, confidence set $\mathbf{CM}$, last state $\mathbf{s}$, prior knowledge $pk$
Output: updated behaviour policy $\boldsymbol{\pi}^{\Phi}[k]$
  $\mathbf{J} \leftarrow \mathbf{0}$ if first call to algorithm
  $cells \leftarrow [pk.n]$          ▷ $pk.X$ denotes $X$ contained in $pk$
  $\boldsymbol{\pi}^{\Phi}[k] \leftarrow \boldsymbol{\pi}^{\Phi}[k-1]$
  while $|cells| \geq 1$ do
      sample $i$ from $cells$, remove $i$ from $cells$
      $jumble \leftarrow [(\pi_i^{\Phi}[k-1] = pk.\pi_i^{\text{init}}) \wedge (s_i \in pk.\mathbf{s}^{\text{init}})]$
      $J_i \leftarrow J_i + 1$ if $jumble$
      $jumble \leftarrow jumble \wedge \begin{cases} true & \text{w.p. } \frac{1}{\max\{1,J_i\}} \\ false & \text{otherwise} \end{cases}$
      if $jumble$ then          ▷ jumbling speeds up convergence
          $\pi_i \leftarrow \pi_i^{\text{init}}$
      else
          $\pi_i^{\Phi}[k] \leftarrow \pi_i^{+}$, $verified \leftarrow$ VERIFY($\boldsymbol{\pi}^{\Phi}[k]$, $\mathbf{CM}$, $\mathbf{s}$, $pk$)
          $\pi_i^{\Phi}[k] \leftarrow \pi_i^{\Phi}[k-1]$ if $\neg verified$
```

add a subroutine ACTION-PRUNING, to solve the following problem: Given a cellular MDP CM, find a pruned cellular MDP CM^-. CM^- is identical apart from the transition function P^-, which, for all $\mathbf{s}, \mathbf{s}' \in \mathbf{S}$ and $\in \mathbf{A}(\mathbf{s})$, satisfies

$$P^-(\mathbf{s}'|\mathbf{s},\mathbf{a}) \geq 0 \text{iff } C_{ij}(unsafe|s'_i, s'_j) = 0 \text{for all } i,j \in [n] \quad (6)$$

ACTION-PRUNING relies on the assumption of consistent regulators from Defintion 2. It is applied to each $\widetilde{CM} \in \mathbf{CM}$ yielding the pruned confidence set $\mathbf{CM}^- \subseteq \mathbf{CM}$, which is used for planning instead. If a new pruning is performed by ACTION-PRUNING, then PE-UCRL proceeds to the planning part of the on-policy phase even if not *switchPhase*.

In the off-policy phase, reward-shaping is used to ensure that PE-UCRL explores potential side-effects sufficiently. Let $I(\mathbf{s})$=1 if only *silent* has been observed for state $\mathbf{s}$ and =0 otherwise. Let $N[k](\mathbf{s}, \mathbf{a})$ be the number of times that state–action pair $(\mathbf{s}, \mathbf{a})$ has been visited before the kth on-policy phase (and $\delta \in\,]0, 1[$ is a constant). (Continuing, we denote X after the kth on-policy phase by $X[k]$.) The reward-shaping is defined by

$$R(\mathbf{s},\mathbf{a},\mathbf{s}') := R(\mathbf{s},\mathbf{a},\mathbf{s}') + I(\mathbf{s})\sqrt{\frac{7\log(2|\mathbf{S}||\mathbf{A}|t[k]/\delta)}{2\max\{1, N[k](\mathbf{s},\mathbf{a})\}}} \quad (7)$$

The final modification is the addition of a subroutine PE-SHIELD to the end of the planning part of the off-policy phase. PE-SHIELD is a shield that outputs a policy $\pi^{\Phi}[k]$ that satisfies the regulatory constraints Φ. As input, PE-SHIELD takes $\boldsymbol{\pi}^{\Phi}[k-1]$, the behaviour policy from the previous on-policy phase, and $\boldsymbol{\pi}^{+}$, the target policy calculated through extended value iteration. The idea behind PE-SHIELD is to iterate over cells $i \in [n]$ and gradually replace $\boldsymbol{\pi}^{\Phi}[k-1]$ with $\boldsymbol{\pi}^{+}$ as long as Φ is satisfied according to a probabilistic model-checker VERIFY. The details are shown in Algorithm 1. In VERIFY, all states in $\mathbf{S}^{\infty}$ from Definition 4 are assumed to be safe. Otherwise, VERIFY works as any other probabilistic model-checker, e.g. Algorithm V in Pugelli et al. (2013). Algorithm V has a time complexity polynomial in $\mathrm{O}(|\mathbf{S}|^2|\mathbf{A}|)$, which is not worse than the planning part of UCRL2. PE-UCRL is our proposition to solve the pilot experimentation problem and corroborating evidence is presented in the following sections.

5 Safety and Convergence Analyses

In this section, we analyse PE-UCRL theoretically. First, we prove that it is always safe in the sense that it always satisfies the regulatory constraints. Under additional assumptions, we prove that it converges to the optimal policy.

Safety. PE-UCRL solves the safety part of the pilot experimentation problem without any additional assumptions.

Theorem 1 (Safety). *With probability $\geq 1 - \delta$,* PE-UCRL *satisfies Φ from Definition 5. Its time complexity between every time step in the off-policy phase is polynomial in $\mathrm{O}(n|\mathbf{S}|^2|\mathbf{A}|)$. Its time complexity between every time step in the on-policy phase is in $\mathrm{O}(n^2 + \max_{i\in[n]} |\mathbf{S}_i|)$.*

For proof, see Appendix D. Theorem 1 shows that the time complexity for the off-policy phase is comparable to other upper confidence RL algorithms for small n. However, The time complexity for the on-policy phase may be high. Below, we make an easily satisfied assumption, under which the time complexity is greatly reduced.

Definition 6 (State Transience Invariance). A cellular MDP is state transient invariant if, for any $s_{\#} \in \mathbf{s} \in \mathbf{S}$ and $a_{\#} \in \mathbf{a} \in \mathbf{A}(\mathbf{s})$, there exists $s'_{\#} \in \mathbf{s}' \in \mathbf{S}_{\#}$ such that $P(s'_{\#}|s_{\#}, a_{\#}) > 0$ and $C_{ij}(unsafe \mid s'_i, s'_{\#}) = 0$.

Corollary 1.1. *Assume: The cellular MDP is state transience invariant. Then, the time complexity of* PE-UCRL *between every time step in the on-policy phase is in $\mathrm{O}(n^2)$.*

Convergence. PE-UCRL solves the convergence part of the pilot experimentation, but, as in all convergence theorems for RL algorithms, strong assumptions are necessary. Before we state the assumptions, we introduce a novel form of regret analysis suitable for safe exploration problems. Classical regret analysis assumes that the available prior knowledge takes the form in Equation (2), whereas we assume the more general object pk from Definition 5. Assume, e.g., that $pk.n = 1$ and $pk.\Phi = [\bigcirc\text{№}_{all} \geq 1] \leq 1$. Then, no algorithm can get sublinear regret, i.e., no algorithm converges. This example illustrates a trade-off between expressivity and explorability. *Expressivity* is the set of possible physical and regulatory constraints contained in the prior knowledge. *Explorability* is the number of states the agent can safely explore at any time. Note that our definition of explorability is different from learnability (Yang, Littman, and Carbin 2022). Because of the trade-off, our regret analysis considers algorithm–prior pairs instead of algorithms alone.

Definition 7 (Regret). Let CM^- be the pruned true cellular MDP CM from Equation (6). Let $\boldsymbol{\pi}^{\star}$ be the optimal policy from Definition 5. The regret is defined by

$$\underset{(\textsc{Alg},prior)}{\mathrm{Regret}(T)} := \mathbb{E}_{\boldsymbol{\pi}^{\star}}\left[\sum_{\tau=0}^{T} r^{\tau} \,\middle|\, CM^-, \mathbf{s}^0 = \mathbf{s}^{\text{init}}\right] - \sum_{\tau=0}^{T} r^{\tau},$$

where ALG is an algorithm and *prior* is prior knowledge.

The literature on safe exploration has focused on limiting expected adverse side-effects, whereas the side-effects avoidance literature has focused on the unexpected. We say that adverse side-effects are unexpected if $prior = pk$. We say that adverse side-effects are expected if $prior = pk{+}\mathbf{C}$, where $pk{+}\mathbf{C}$ is an object equal to pk except that it, in addition, contains reported side-effects. We assume that side-effects are expected for the intermediary results and generalise to unexpected adverse effects for the final theorem.

The first assumption uses a standard quantity in convergence analyses of RL algorithms: the diameter of an MDP. We adapt it to cellular MDPs by defining it as

$$D := \max_{\mathbf{s},\mathbf{s}' \in \mathbf{S}} \min_{\boldsymbol{\pi}:\mathbf{S}\to\mathbf{A}} \mathbb{E}_{\boldsymbol{\pi}}[\theta(\mathbf{s},\mathbf{s}') \mid CM],$$

where $\theta : \mathbf{S}\times\mathbf{S} \to Distr(\mathbb{N})$ is a random function. $\theta(\mathbf{s},\mathbf{s}')$ is defined as the minimum number of time steps Δ it takes to reach state $\mathbf{s}' = \mathbf{s}^{t+\Delta}$ from state $\mathbf{s} = \mathbf{s}^t$ for any time t. The diameter can be used for the following standard assumption.

Definition 8 (Communicating Cellular MDP.)**.** A cellular MDP is communicating if $D < \infty$.

Intermediary result 1. Assume: The cellular MDP is state transient invariant and communicating. Then, there exists a term $oh_{pk+\mathbf{C}}(T)$ equal to the difference in regret between PE-UCRL and UCRL2. For details, see Lemma 1 in Appendix D. Since $oh_{pk+\mathbf{C}}(T) \geq 0$, we can view it as an overhead, which is the price for stronger safety guarantees. $oh_{pk+\mathbf{C}}(T)$ is constant if $pk{+}\mathbf{C}.n = 1$ and $pk{+}\mathbf{C}.\varPhi = [\bigcirc\text{№}_{all} \leq 1] \geq 1$. That means the overhead is not sublinear and convergence cannot be guaranteed. We define suitable constraints on prior knowledge below.

Definition 9 (Explorability–Expressivity Trade-Off)**.** Prior knowledge $prior$ strikes a suitable trade-off between explorability and expressivity if two conditions are met: The regulatory constraints belong in an explorable fragment of PE-PCTL, and the reward function is monotonically increasing. The regulatory constraints $\varPhi$ belong in an explorable fragment of PE-PCTL if $\varPhi$ implies that, for all $\mathbf{s} \in \mathbf{S}$ and $j \in [n]$, there exist $t \in \mathbb{N}$ and $i \in [n]$ such that

$$C_{ij}(safe \mid s_i, s_j) > 0 \text{ if } \mathbf{s}^t = \mathbf{s}.$$

For all $i \in [n]$, let $R_i : \mathbf{S}_i \to [0,1]$. Let $f : [0,1]^n \to [0,1]$. The reward function R is monotonically increasing if

$$R = f(\mathbf{R}) \text{ such that } f(\mathbf{R}(\mathbf{s}')) \geq f(\mathbf{R}(\mathbf{s})),$$

where $R_i(\mathbf{s}'_i) \geq R_i(\mathbf{s}_i)$ for all $i \in [n]$, and $R_i(\mathbf{s}'_i) > R_i(\mathbf{s}_i)$ for some $i \in [n]$

Intermediary result 2. Assume: The cellular MDP is state transient invariant and communicating. $pk{+}\mathbf{C}$ strikes a suitable trade-off between explorability and expressivity. Then, $oh_{pk+\mathbf{C}}(T)$ approaches a constant as $T \to \infty$. For details, see Lemma 2 in Appendix D. Quantitative bounds for different regulations within the explorable fragment of PE-PCTL require different proofs and are left for future work.

Assuming prior knowledge of side-effects does not solve the pilot experimentation problem. However, without assumptions about the regulators, the cellular MDP could be such that they are always silent, and convergence is impossible. Below, we prove convergence after assuming that all side-effects can eventually be reported by some regulator.

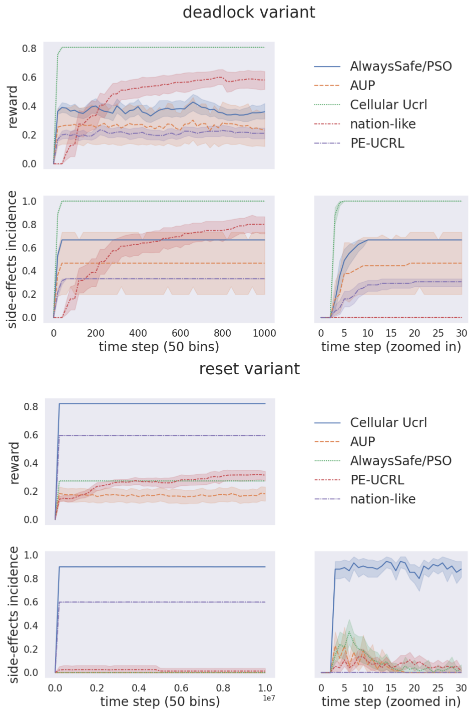

Figure 1: Reward and side-effects incidence over time. Averages and standard deviations for ≥ 20 samples are shown. Algorithm names abbreviate algorithm–regulations pairs.

Definition 10 (Complete Reporting)**.** Reporting is complete if, for all $s_\# \in \mathbf{S}_\#$, there exist $s'_\# \in \mathbf{S}_\#$ and $i,j \in [n]$ such that $C_{ij}(\{safe, unsafe\} \mid s'_\#, s_\#) > 0$.

Theorem 2 (Convergence)**.** *Assume: The cellular MDP is state transient invariant and communicating. The prior knowledge pk strikes a suitable trade-off between explorability and expressivity. Reporting is complete. Then, with probability $\geq 1-\delta$, for large $T \in \mathbb{N}$,*

$$\underset{(\text{PE-UCRL},pk)}{\mathrm{Regret}}(T) \in \tilde{\mathrm{O}}(D|\mathbf{S}|\sqrt{T|\mathbf{A}|})$$

For proof, see Appendix D. In summary, PE-UCRL is guaranteed to converge in some environments, thereby, solving the pilot experimentation problem.

6 Implementation and Experimental Results

In this section, we introduce implementations of environments that can be modelled as cellular MDPs. Then, we summarise four benchmark algorithms. Finally, we present the results comparing PE-UCRL with the benchmark algorithms in the environments.

Environments. Environment suites for testing RL algorithms for both safe exploration (Ray, Achiam, and Amodei 2019) and side-effects avoidance (Wainwright and Eckersley 2020) have been proposed. It is problematic, for our purposes, that they cannot be *exactly* modelled as cellular MDPs. Therefore, we introduce two novel environments.

Note that many existing implementations approximate cellular MDPs and that making PE-UCRL work for approximate cellular MDPs is an important area for future work.

Similarly to previously proposed environments (Krueger, Maharaj, and Leike 2020, e.g.), both of our environments model polarisation in recommender engines (caused by misinformative content and other kinds of manipulative content). Note that there is little consensus on whether recommender engines have a large impact on polarisation (Ribeiro et al. 2020), but, in our environments, we assume the affirmative hypothesis for illustrative purposes. We refer to the environments as the reset variant and the deadlock variant.

Reset variant. In this toy example, there are three cells corresponding to a regular user, a child, and a moderator. The content recommendation actions of the recommender engine agent can push users towards more or less extreme views. The most extreme content is both the most rewarding and most likely to be reported as unsafe by the moderator. The *reset assumption* is satisfied: Any policy will eventually reset the system to the initial state. The reset assumption is reasonable in complex systems when there is some process trying to maintain homeostasis, e.g., schools and established news agencies countering polarising misinformation. Convergence can be guaranteed in the reset variant.

Deadlock variant. The only difference is that the most extreme intracellular state of polarisation is so extreme that the user can never be deradicalised, i.e., the user gets stuck in a deadlock intracellular state. Therefore, convergence cannot be guaranteed by any algorithm—only safety can be.

Further details are available in Appendix E.

Comparisons. Merely comparing algorithms in safe exploration would result in unfair comparisons as prior knowledge might differ, cf. Section 5. In the experiments, prior knowledge only differs in terms of regulatory constraints, so we compare algorithm–regulations pairs.

(PE-UCRL, $[\bigcirc \bigcirc (\text{№}_{all} \geq 1)]_{\leq 0.5}$) is the algorithm–regulations pair representing our proposed solution. Verification uses PRISM (Kwiatkowska et al. 2020), and, in particular, its interval MDP engine[1]. The symbol № is implemented as a variable №_g for each $g \in \mathbf{G}$.

Baselines. We designed two, which we compare against:

(CELLULAR UCRL, n/a) can be seen as an ablation of PE-UCRL without ACTION-PRUNING and PE-SHIELD. Further ablations are available in Appendix F.

(NATION-LIKE, update w.p. 0.2 every 50 time steps) is a more sophisticated baseline. However, it lacks formal guarantees. For details, see Appendix C.

State-of-the-art. We restrict our comparisons to solutions that can achieve safety in discrete non-episodic MDPs:

(AUP, 10 auxiliary functions and coefficient $=1$), where AUP is short for attainable utility preservation (Turner, Ratzlaff, and Tadepalli 2020), is a method from the literature on avoiding side-effects. In brief, it introduces a regulariser with a coefficient $=1$. The regulariser works by solving a multi-objective optimisation problem over the reward function and 10 auxiliary reward functions, which are sampled randomly. We apply the regulariser to SIDE-EFFECTS-AVOIDING UCRL.

(ALWAYSSAFE/PSO, delicate variables $i{:}L(i){=}children$) is the implementation in cellular MDPs for two different algorithms: AlwaysSafe (Simão, Jansen, and Spaan 2021) and (Farquhar, Carey, and Everitt 2022). The reason that they (surprisingly) coincide for cellular MDPs is the cellular independence assumption from Definition 2. These are methods from the literature on safe exploration. Delicate variables are cells that the RL agent must not alter.

Further details are available in Appendix C.

Results. In the experiments, we evaluated the algorithm–regulations pairs on two metrics: reward and side-effects incidence for each time step. Side-effects incidence is defined as the fraction of cells that are in an unsafe intracellular state at any one time (regardless of whether the regulators are reporting them). How the reward and the side-effects incidence change over time is shown in Figure 1.

The experiments corroborate theorems 1 and 2, i.e., (PE-UCRL, $[\bigcirc \bigcirc (\text{№}_{all} \geq 1)]_{\leq 0.5}$) *is safe and converges.* In the deadlock variant, the side-effects incidence for (PE-UCRL, $[\bigcirc \bigcirc (\text{№}_{all} \geq 1)]_{\leq 0.5}$) does not decrease, but it is always at an acceptable level. The side-effects are also bounded for the state-of-the-art algorithm–regulations pairs. Both baselines approach maximal levels of side-effects over time, although not equally quickly. In the reset variant, the side-effects incidence for (PE-UCRL, $[\bigcirc \bigcirc (\text{№}_{all} \geq 1)]_{\leq 0.5}$) is low initially, and over time it decreases even further. In contrast, the baselines get high levels of side-effects. The state-of-the-art algorithm–regulations pairs have acceptable levels of side-effects but do not approach the optimal reward. Convergence can only be guaranteed in the reset variant, and (PE-UCRL, $[\bigcirc \bigcirc (\text{№}_{all} \geq 1)]_{\leq 0.5}$) is the *only* algorithm–regulations pair that converges.

Limitations are that convergence is slower and that the reset assumption helps the algorithms. In the reset environment, (PE-UCRL, $[\bigcirc \bigcirc (\text{№}_{all} \geq 1)]_{\leq 0.5}$) performs worse than (ALWAYSSAFE/PSO, delicate variables $i{:}L(i){=}children$) for the early time steps. Furthermore, the reset assumption helps PE-UCRL by letting it see the transitions necessary to update the non-exploratory intracellular policies many times. However, we hypothesise that if ergodicity-preserving regularisation (Moldovan and Abbeel 2012) was added to PE-UCRL, then it would not matter if the environment satisfied the reset assumption or not. Exploring this hypothesis is beyond the scope of this paper though.

Additional results are available in Appendix F.

7 Conclusions

In summary, we proposed a novel model, the cellular MDP and a novel algorithm, PE-UCRL. We showed that for cellular MDPs, it is possible to strike a suitable trade-off between exploration and exploitation, which ensures that PE-UCRL is safe and converges. We corroborated the theoretical results with experiments and show that PE-UCRL outperforms state-of-the-art algorithms in cellular MDPs.

[1] https://github.com/davexparker/prism/tree/imc2

Future work includes scaling Pe-Ucrl by, e.g., using neural approximations for learning representations, abstractions for verification, and ergodicity-preserving regularisation for exploration. We currently investigate weakening the assumptions on the regulators by considering scenarios in which they disagree or are manipulated by the RL agent.

References

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Andrychowicz, M.; Baker, B.; Chociej, M.; Józefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; Schneider, J.; Sidor, S.; Tobin, J.; Welinder, P.; Weng, L.; and Zaremba, W. 2020. Learning dexterous in-hand manipulation. *Int. J. Robotics Res.*, 39(1).

Armstrong, S.; and Mindermann, S. 2018. Occam's razor is insufficient to infer the preferences of irrational agents. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 5603–5614.

Berkenkamp, F.; Turchetta, M.; Schoellig, A. P.; and Krause, A. 2017. Safe Model-based Reinforcement Learning with Stability Guarantees. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 908–918.

Biyik, E.; Margoliash, J.; Alimo, S. R.; and Sadigh, D. 2019. Efficient and Safe Exploration in Deterministic Markov Decision Processes with Unknown Transition Models. In *2019 American Control Conference, ACC 2019, Philadelphia, PA, USA, July 10-12, 2019*, 1792–1799. IEEE.

Bostrom, N. 2014. *Superintelligence*. Dunod.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting Structure in Policy Construction. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, 1104–1113. Morgan Kaufmann.

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

Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 4299–4307.

Ciesinski, F.; and Größer, M. 2004. On Probabilistic Computation Tree Logic. In Baier, C.; Haverkort, B. R.; Hermanns, H.; Katoen, J.; and Siegle, M., eds., *Validation of Stochastic Systems - A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, 147–188. Springer.

Degrave, J.; Felici, F.; Buchli, J.; Neunert, M.; Tracey, B.; Carpanese, F.; Ewalds, T.; Hafner, R.; Abdolmaleki, A.; de Las Casas, D.; et al. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419.

El Mahdi, E.; et al. 2019. *Le fabuleux chantier: Rendre l'intelligence artificielle robustement bénéfique*. EDP Sciences.

Farquhar, S.; Carey, R.; and Everitt, T. 2022. Path-Specific Objectives for Safer Agent Incentives. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 9529–9538. AAAI Press.

Filippi, S.; Cappé, O.; and Garivier, A. 2010. Optimism in reinforcement learning and Kullback-Leibler divergence. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 115–122. IEEE.

Friedman, L. M.; Furberg, C. D.; DeMets, D. L.; Reboussin, D. M.; and Granger, C. B. 2015. *Fundamentals of clinical trials*. Springer.

García, J.; and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16: 1437–1480.

Hadfield-Menell, D.; Milli, S.; Abbeel, P.; Russell, S. J.; and Dragan, A. D. 2017. Inverse Reward Design. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6765–6774.

Hadfield-Menell, D.; Russell, S.; Abbeel, P.; and Dragan, A. D. 2016. Cooperative Inverse Reinforcement Learning. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 3909–3917.

Hoang, L. N. 2019. Towards Robust End-to-End Alignment. In Espinoza, H.; hÉigeartaigh, S. Ó.; Huang, X.; Hernández-Orallo, J.; and Castillo-Effen, M., eds., *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, volume 2301 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jaksch, T.; Ortner, R.; and Auer, P. 2010. Near-optimal Regret Bounds for Reinforcement Learning. *J. Mach. Learn. Res.*, 11: 1563–1600.

Koller, T.; Berkenkamp, F.; Turchetta, M.; and Krause, A. 2018. Learning-Based Model Predictive Control for Safe Exploration. In *57th IEEE Conference on Decision and Control, CDC 2018*, 6059–6066. Miami, FL, USA: IEEE.

Könighofer, B.; Bloem, R.; Ehlers, R.; and Pek, C. 2022. Correct-by-Construction Runtime Enforcement in AI - A Survey. In Raskin, J.; Chatterjee, K.; Doyen, L.; and Majumdar, R., eds., *Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday*, volume 13660 of *Lecture Notes in Computer Science*, 650–663. Springer.

Kori, A.; Glocker, B.; and Toni, F. 2022. Visual Debates. arXiv:2210.09015.

Krakovna, V.; Orseau, L.; Martic, M.; and Legg, S. 2019. Penalizing Side Effects using Stepwise Relative Reachability. In Espinoza, H.; Yu, H.; Huang, X.; Lécué, F.; Chen, C.; Hernández-Orallo, J.; hÉigeartaigh, S. Ó.; and Mallah, R., eds., *Proceedings of the Workshop on Artificial Intelligence Safety 2019 co-located with the 28th International Joint Conference on Artificial Intelligence, AISafety@IJCAI 2019*, volume 2419 of *CEUR Workshop Proceedings*. Macao, China: CEUR-WS.org.

Krueger, D.; Maharaj, T.; and Leike, J. 2020. Hidden Incentives for Auto-Induced Distributional Shift. arXiv:2009.09153.

Kwiatkowska, M.; Norman, G.; Parker, D.; and Santos, G. 2020. PRISM-games 3.0: Stochastic Game Verification with Concurrency, Equilibria and Time. In Lahiri, S. K.; and Wang, C., eds., *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II*, volume 12225 of *Lecture Notes in Computer Science*, 475–487. Springer.

Lindner, D.; Turchetta, M.; Tschiatschek, S.; Ciosek, K.; and Krause, A. 2021. Information Directed Reward Learning for Reinforcement Learning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 3850–3862.

Martínez, D.; Alenyà, G.; and Torras, C. 2015. Safe robot execution in model-based reinforcement learning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, 6422–6427. IEEE.

Moldovan, T. M.; and Abbeel, P. 2012. Safe Exploration in Markov Decision Processes. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Mqirmi, P. E.; Belardinelli, F.; and León, B. G. 2021. An Abstraction-based Method to Check Multi-Agent Deep Reinforcement-Learning Behaviors. In Dignum, F.; Lomuscio, A.; Endriss, U.; and Nowé, A., eds., *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, 474–482. ACM.

Odriozola-Olalde, H.; Zamalloa, M.; and Arana-Arexolaleiba, N. 2023. Shielded Reinforcement Learning: A review of reactive methods for safe learning. In *IEEE/SICE International Symposium on System Integration, SII 2023, Atlanta, GA, USA, January 17-20, 2023*, 1–8. IEEE.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Osband, I.; and Roy, B. V. 2017. Why is Posterior Sampling Better than Optimism for Reinforcement Learning? In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 2701–2710. PMLR.

Puggelli, A.; Li, W.; Sangiovanni-Vincentelli, A. L.; and Seshia, S. A. 2013. Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties. In Sharygina, N.; and Veith, H., eds., *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, 527–542. Springer.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley. ISBN 978-0-47161977-2.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning.

Ribeiro, M. H.; Ottoni, R.; West, R.; Almeida, V. A. F.; and Jr., W. M. 2020. Auditing radicalization pathways on YouTube. In Hildebrandt, M.; Castillo, C.; Celis, L. E.; Ruggieri, S.; Taylor, L.; and Zanfir-Fortuna, G., eds., *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, 131–141. ACM.

Roderick, M.; Nagarajan, V.; and Kolter, J. Z. 2021. Provably Safe PAC-MDP Exploration Using Analogies. In Banerjee, A.; and Fukumizu, K., eds., *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, 1216–1224. PMLR.

Russell, S. 2019. *Human compatible: Artificial intelligence and the problem of control*. Penguin.

Saisubramanian, S.; Zilberstein, S.; and Kamar, E. 2021. Avoiding Negative Side Effects Due to Incomplete Knowledge of AI Systems. *AI Mag.*, 42(4): 62–71.

Saunders, W.; Sastry, G.; Stuhlmüller, A.; and Evans, O. 2018. Trial without Error: Towards Safe Reinforcement Learning via Human Intervention. In André, E.; Koenig, S.; Dastani, M.; and Sukthankar, G., eds., *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Shah, R.; Krasheninnikov, D.; Alexander, J.; Abbeel, P.; and Dragan, A. D. 2019. Preferences Implicit in the State of the World. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T. P.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017. Mastering the game of Go without human knowledge. *Nat.*, 550(7676): 354–359.

Simão, T. D.; Jansen, N.; and Spaan, M. T. J. 2021. AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training. In Dignum, F.; Lomuscio, A.; Endriss, U.; and Nowé, A., eds., *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, 1226–1235. ACM.

Turchetta, M.; Berkenkamp, F.; and Krause, A. 2016. Safe Exploration in Finite Markov Decision Processes with Gaussian Processes. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 4305–4313.

Turner, A. M.; Ratzlaff, N.; and Tadepalli, P. 2020. Avoiding Side Effects in Complex Environments. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wainwright, C. L.; and Eckersley, P. 2020. SafeLife 1.0: Exploring Side Effects in Complex Environments. In Espinoza, H.; Hernández-Orallo, J.; Chen, X. C.; ÓhÉigeartaigh, S. S.; Huang, X.; Castillo-Effen, M.; Mallah, R.; and McDermid, J. A., eds., *Proceedings of the Workshop on Artificial Intelligence Safety, co-located with 34th AAAI Conference on Artificial Intelligence, SafeAI@AAAI 2020, New York City, NY, USA, February 7, 2020*, volume 2560 of *CEUR Workshop Proceedings*, 117–127. CEUR-WS.org.

Yang, C.; Littman, M. L.; and Carbin, M. 2022. On the (In)Tractability of Reinforcement Learning for LTL Objectives. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 3650–3658. ijcai.org.