A RULE-BASED SHIELD: ACCUMULATING SAFETY RULES FROM CATASTROPHIC ACTION EFFECTS

Shahaf S. Shperberg[1], **Bo Liu**[1], **Alessandro Allievi**[1,2], **Peter Stone**[1,3]
[1]University of Texas at Austin, [2]Robert Bosch LLC, [3]Sony AI
{shperbsh,bliu,pstone}@cs.utexas.edu, Alessandro.Allievi@us.bosch.com

ABSTRACT

Deploying autonomous agents in the real-world can lead to risks both to the agents and to the humans with whom they interact. As a result, it is essential for agents to try to achieve their objectives while acting as safely as possible. Thus, learning agents ought to learn not only about the effectiveness of actions, but also about their safety. While action effectiveness is task-dependent, information regarding the safety of actions can be preserved even if the task and/or the objective of the agent changes. The focus of this work is to leverage information from unsafe situations that the agent has experienced in order to obtain safety rules that identify which action from which state can lead to unsafe outcomes. These rules can be used for *shielding* the agent from repeating the same mistakes, as well as other mistakes that lead to the same catastrophic outcomes. In essence, before each action is selected for execution by the policy, actions which violate one of the safety rules from the current state are masked away and will not be selected. The cumulative set of safety rules can be used even when the agent faces multiple tasks, and can also be shared between different agents, so that mistakes that were made by one agent are not repeated by any of the agents that share the same rule-based shield. The process of learning a rule-based shield online is studied on a multi-task autonomous driving problem. Finally, the application of a rule-based shield to the Proximal Policy Optimization (PPO) algorithm is empirically evaluated and compared with the original PPO, with variants of PPO which use other online-learned shields, and with other baselines from the safe reinforcement learning literature. The results show that safety rules can significantly reduce the number of unsafe outcomes that agents experience, while even improving the cumulative rewards obtained by the agents.

1 INTRODUCTION

In continual and lifelong learning, agents face a variety of tasks in previously unknown environments. Since the dynamics of these environments cannot be fully modeled in advance, the agents are bound to occasionally take unsafe actions that lead to catastrophic outcomes (just as people do). The aim of this work is preventing agents from ever *repeating* such mistakes (which is more than can be said for many people!). Ideally, once an agent has made a catastrophic mistake, neither that agent nor any other agent should ever repeat it, nor any similar mistake that leads to the same unsafe outcome. Methods that aim to avoid mistake repetition in continuous domains need to possess the following four properties: i) an ability to identify catastrophic mistakes, i.e., which action, from which state, was the root cause of a catastrophe; ii) an ability to avoid taking actions that were previously discovered to be unsafe (avoid repeating identified mistakes); iii) low overhead in terms of both memory and runtime, so that agents can operate in real-time; and iv) an ability to generalize from experienced mistakes to a class of similar mistakes. The latter property is critical in continuous domains, in which the probability of an agent revisiting the exact same state is negligible.

This paper introduces a novel approach, called rule-based shielding, which is the first to support all four properties required to avoid mistake repetition. When an agent reaches an unsafe outcome, we assume that it can receive a set of safety rules (predicates) from an oracle (called a Mistake Analysis Entity, or MAE), which if followed, ensure that the same *equivalence class of mistakes* will not be repeated. For example, by observing a rear-end collision in an autonomous driving task, an oracle could provide a set of rules that prevent all future rear-end collisions. In this context, an agent needs to observe that an unsafe outcome was reached (e.g., a vehicle crash) and query the MAE with the corresponding trajectory in order to obtain a set of safety rules which prevent that particular mistake as well as any similar mistake. The safety rules are accumulated in a rule database and are used for verifying the safety of actions selected by the agent's policy before their deployment, thus shielding the agent from taking actions that are known to be unsafe from its current state. The rule-based shield method guarantees that the same mistake will never be repeated, and has the additional merit of being task-agnostic and shareable among different (homogeneous) agents. The reliance on an external MAE is an obvious limitation of this approach. While safety rules can be given to an agent by any

MAE, a natural way to obtain such rules is via domain experts. In fact, there are cases, such as aviation (FAA) or autonomous driving (Sinha et al., 2021), in which domain experts already analyze catastrophic outcomes. In addition, every time an agent makes a mistake, that mistake as well as other mistakes that belong to the same equivalence class are guaranteed not to be repeated again. Thus, the effort required to generate safety rules is expected to diminish at an exponential rate, as new mistakes become increasingly scarce. Moreover, generating rules to avoid one particular type of mistake is a much easier and more feasible task than designing criteria which prevent all possible unsafe outcomes in advance, as done in previous work. We note that program synthesis methods (e.g., (Holtz et al., 2021)) may be useful for automatically detecting mistakes and generalizing to equivalence classes of mistakes. Nonetheless, we leave the generation of automated MAE to future work and focus on mistake analysis provided by domain experts.

We apply the concept of accumulating safety rules to shield agents from unsafe actions to the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017), creating a variant called Proximal Policy Optimization with Rule-based Shield (PPO-RS). An empirical study is performed to evaluate PPO-RS in terms of mean episodic return, rate of catastrophic mistakes, and cumulative number of safety rules. This study is conducted using multi-task autonomous driving behavior control simulation based on the highway environment (Leurent, 2018), where each task has its own objective and setting (road structure, number of vehicles, etc.), but they all share the same underlying safety concern of not crashing. This domain will also be used as a case study to demonstrate important aspects in the interactive process of accumulating safety rules from observed catastrophic effects. In this empirical study, PPO-RS is compared to PPO as well as to five other baselines. The first baseline is an approach called ShieldPPO (Shperberg et al., 2022) which was designed to avoid mistake repetition in discrete domains. The second baseline is a newly introduced adaptation of ShieldPPO for continuous domains. The next two baselines are well-known algorithms in the safe reinforcement learning literature, Constrained Policy Optimization (CPO) (Achiam et al., 2017) and PPO-Lagrangian (Ray et al., 2019), a variant of PPO that uses Lagrangian methods to enforce safety constraints. The last baseline is a variant of PPO that models catastrophic outcomes as rewards by artificially assigning them high negative rewards. The results indicate that PPO-RS outperforms all baselines in this experimental setting. Furthermore, PPO-RS requires less than thirty rules in order to avert all collisions experienced during training, thus the overhead from using the rule-based shield is negligible in terms of both memory and runtime.

2 Related Work

The aim of safe reinforcement learning (safe RL) (García & Fernández, 2015) is to maximize environment return when learning and deploying policies while respecting safety constraints. Most safe RL methods can be divided into two categories. Methods in the **first category** use a threshold to limit the unsafeness of learned policies. A prominent line of work that falls into this category is constrained optimization. Under this formulation, agents receive a cost per step (where cost is used for modeling unsafe interactions) from the environment and aim to keep the discounted cumulative cost below a given threshold whilst maximizing the environment return (Kadota et al., 2006; Moldovan & Abbeel, 2012). Constrained optimization is commonly defined using a Constrained Markov Decision Process (CMDP) (Altman, 1999), which is solved via different approaches, such as augmented Lagrangian-based methods (Liu et al., 2020), trust-region methods (Achiam et al., 2017) and Lyapunov-based methods (Chow et al., 2018). Another type of method in this category is based on minimizing the risk of being unsafe. In essence, the future return is modeled as a distribution, and a threshold is used for balancing the risk and return for learned policies (Mausser & Rosen, 1999; Di Castro et al., 2012; Carrara et al., 2019). The approaches in this category provide a methodological way to control the trade-off between the safety of agents and their achieved rewards. Yet, it can be very hard for environment designers to set safety threshold values that capture the desired balance between risk and reward. Thus, policies that are learned using these approaches can often be either too risky or too conservative. In addition, while such approaches learn from past mistakes, they do not have guarantees regarding the absence of mistake repetition and can often make the same mistake more than once.

Methods in the **second category** rely on safety criteria which are given to the agent *before training* to avert making catastrophic mistakes. The safety criteria encode information as to which actions are safe (or unsafe) to take from each state, and can be given to agents in different forms such as LTL expressions (Alshiekh et al., 2018) or predicates (e.g., safety envelopes (Desai et al., 2019; Bernhard et al., 2021)). These criteria can be then used for *shielding* agents from taking unsafe actions. With access to high-quality safety information, agents can avoid the effects of most unsafe actions.[1] Nonetheless, designing safety criteria can be a challenging task, as it requires foreseeing all mistakes which agents can possibly make that lead to catastrophic outcomes.[2]

[1] In some cases, unsafe outcomes cannot be averted, thus absolute safety cannot be ensured by any method.

[2] Under some criteria representation, it might be easy to define safety criteria. For example, for behavior control in autonomous driving tasks, it is simple to provide an LTL formula that encodes the constraint "never to reach a state with collision". Yet, it is computationally intractable to validate that actions do not violate the encoded constraints in such cases.

Some methods for safety do not fall under the two common categories. One common approach is to avoid repeating *mistakes*, i.e., avoid executing actions in states that have resulted in catastrophic outcomes. A straightforward way to avoid mistake repetition is to shape the reward of discovered unsafe state-action pairs to a high negative value (e.g., negative infinity) so that the agent learns to avoid them. However, the use of negative reward to account for safety has two main disadvantages. First, the value of a state-action pair corresponds to the expected environment return, which depends on the specific task that the agent is solving. Yet, safety constraints are usually independent of the specific task that that agent aims to achieve. Thus, modeling safety aspects using values learned from negative rewards limits the ability to transfer safety information between tasks. In addition, agents that model catastrophic outcomes via negative rewards are more prone to catastrophic forgetting that either causes mistake repetition or over-generalization, which results in overly risk-averse policies that yield low rewards (Dalal et al., 2018; Saunders et al., 2018a; Thananjeyan et al., 2021).

Recently, Shperberg et al. (2022) proposed an alternative way to avoid mistake repetition. This work uses the concept of shielding to refrain from taking unsafe actions, but rather than assuming that safety criteria are given to the agent in advance, catastrophic mistakes are assumed to be observable by the agent. Whenever an agent takes an action a from a state s which causes an unsafe outcome, the state-action pair (s, a) is stored in a database. This database is used for masking away unsafe actions every time an action is chosen for deployment from any state. The aim of that work is to learn a shield in an online manner based on the agent's experience while still maintaining some safety guarantees. While criteria-based approaches ensure that agents never take any unsafe action, the online shield guarantees that agents do not execute all actions that were *discovered* to be unsafe. In essence, the approach proposed by Shperberg et al. (2022) attempts to optimize the environment return subject to the constraint of not repeating the same mistake more than once. However, this approach quickly becomes ineffective for continuous domains. In these, identical states are seldomly repeated, thus the probability of finding the current state of an agent in the database is essentially zero. Consequently, the shield will never classify actions as unsafe, even if they were previously discovered to be unsafe for similar states. Moreover, even in non-trivial discrete domains the amount of mistakes that agents experience during their interaction with the environment can be very large, leading to untenable memory complexity, which would be required to store the monotonically increasing mistake database, and computational complexity, required to verify the safety of actions against such a database. Finally, it may be quite challenging to identify which actions are catastrophic (e.g., which action, from which state, was the root cause of the car's accident). To sidestep these challenges, the authors made the following assumptions: (i) the environments are discrete, (ii) the amount of common catastrophic mistakes (state-action pairs) is small enough to store in memory, and (iii) agents can identify catastrophic mistakes as they occur. Under these assumptions, Shperberg et al. (2022) introduced ShieldPPO, an application of the approach of using a mistake database as a shield to the PPO algorithm, and showed that ShieldPPO is capable of significantly outperforming PPO as well as other safe RL approaches on a simple discrete environment using different settings.

3 Background

In this section, we provide the necessary background on Markov Decision Processes with catastrophic actions and on the concept of shielding to ensure safety.

3.1 Markov Decision Process with Catastrophic Actions

Markov Decision Processes (MDPs) are a common formulation of many Reinforcement learning (RL) problems. An MDP is defined as a tuple $M = (S, \mathcal{A}, T, \gamma, R)$. In such a tuple, S is the state space, $\mathcal{A}$ is the action space, and T is the transition function, i.e., $T(s' \mid s, a)$ is the probability of reaching state s' as a result of taking action a from state s. Moreover, γ is the discount factor and $R : S \times \mathcal{A} \to \mathbb{R}$ is the reward function, such that $R(s, a)$ is the reward that the agent receives when executing action a from state s. The agent's objective is to learn a policy $\pi_\theta : S \to \mathcal{A}$ parameterized by θ, that optimizes the discounted cumulative reward: $\mathbb{E}_{(s_t,a_t)\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

Partially Observable Markov Decision Processes (POMDPs) are extensions of MDPs, for when the environment cannot be fully observed. A POMDP is also defined as a tuple $M' = (S, \mathcal{A}, T, \gamma, R, \Omega, \mathcal{O})$, where the first five elements are as defined in MDPs, Ω is the set of observations, and $\mathcal{O} : S \times A \times \Omega \to \mathbb{R}$ is a conditional observation probability function. In a POMDP, the subsequent state s' resulting from executing an action a from a state s is not observable to the agent. Instead, it receives an observation $o \in \Omega$ with probability $\mathcal{O}(o|s', a)$. In POMDPs, the agent's objective is to learn a policy $\pi'_\theta : \Omega \to \mathcal{A}$ parameterized by θ, that optimizes the discounted cumulative reward.

An MDP with Catastrophic Actions (MDP-CA) and a POMDP with Catastrophic Actions (POMDP-CA) (Shperberg et al., 2022)) include an additional *ground truth* safety labeling function $L_\phi : S \times \mathcal{A} \to \{0, 1\}$. L_ϕ indicates which actions are safe for each state, i.e., it is safe to perform an action a at state s if and only if $L_\phi(s, a) = 1$.

3.2 SHIELDING

One way to ensure safety for autonomous agents is to prevent the execution of "unsafe" actions, i.e., taking an action from a state that leads to a catastrophic outcome. This prevention can be achieved using shielding, i.e., masking out unsafe actions from the agent's policy (Alshiekh et al., 2018). When the states of the environment are observable, a shield can be formally defined as a binary function $\mathcal{S} : S \times \mathcal{A} \rightarrow \{0, 1\}$, where a state and action pair (s, a) is considered safe with respect to $\mathcal{S}$ if and only if $\mathcal{S}(s, a) = 1$. The application of a shield $\mathcal{S}$ to an agent's policy π_θ is defined as follows:

$$\pi_\theta^{\mathcal{S}}(a|s) = \begin{cases} \frac{1}{Z}\, \pi_\theta(a|s)\, \mathcal{S}(s, a) & \text{if at least one of the actions is safe (i.e., } \sum_{a'} \mathcal{S}(s, a') \geq 1), \\ \pi_{\text{default}}(a|s) & \text{otherwise,} \end{cases} \tag{1}$$

In this equation, $Z = \sum_{a'} \pi_\theta(a'|s)\mathcal{S}(s, a')$ is the normalization term and π_{default} is some default policy (e.g., a do-nothing policy) in case no actions are safe from state s according to $\mathcal{S}$. One possibility is to apply $\mathcal{S}$ to π_θ during training in order to avoid exploring areas known to be unsafe. It is also possible to apply the shield to a fully trained policy during deployment. The accuracy of the shield is important to the success of this approach, as false positives can cause the agent to take unsafe actions and false negatives can prevent the agent from taking safe actions, which might affect its performance. The partial observability case is more challenging, as different states (some of which could be safe and some not) can lead to the same observation. Nonetheless, the approximation of treating observations as states, i.e., defining shields as binary functions $\mathcal{S} : \Omega \times \mathcal{A} \rightarrow \{0, 1\}$, works well in our experimental setting.

4 PROBLEM DEFINITION

As in Shperberg et al. (2022), we address the problem of operating in a sequence of K Markov decision processes (either a sequence of MDP-CAs or a sequence of POMDP-CAs), $(M_i)_{i=1}^K$, where all *models* (i.e., MDP-CAs or POMDP-CAs) share the same safety function $L_\phi : S \times \mathcal{A}$, and for every i, $S_i \subseteq S$ and $\mathcal{A}_i \subseteq \mathcal{A}$. In some cases, similar mistakes can occur in different models, e.g., if the sequence of models represents different autonomous driving scenarios, rear-ending the vehicle in front of the ego vehicle is a safety concern shared among many scenarios. By contrast, the hazard of hitting a deer that suddenly crosses the road is much rarer. To capture the concept of different events occurring at different frequencies, each model in the sequence is assumed to be drawn from an underlying long-tailed distribution $\mathbb{M}$.

When facing the k-th model in the sequence, the objective of the agent is to find a policy π that optimizes the discounted cumulative reward on all $\{M_i\}_{i \leq k}$:

$$J(\pi) = \sum_{i=1}^{k} \mathop{\mathbb{E}}_{(s_t^i, a_t^i) \sim \pi, M_i} \left[\sum_{t=0}^{\infty} \gamma_i^t R_i(s_t, a_t) \right]. \tag{2}$$

In essence, the agent needs to learn a *single* policy $\pi : S \rightarrow \mathcal{A}$ (or $\pi : \Omega \rightarrow \mathcal{A}$ for POMDP-CAs) that is projected onto S_i for each model in the sequence. However, in contrast to Shperberg et al. (2022), we do not assume that L_ϕ is observable from the environment. Instead, we assume that the agent can observe whether an unsafe outcome has occurred, but not information about which action from which state caused this outcome. For example, the environment gives an indication that a collision has occurred, but no indication of what the agent did in order to cause that collision (as the mistake could have occurred many steps before the collision itself). Formally, at every time step t, when taking an action a_t from the current state s_t, the agent receives from the environment, in addition to the reward r_t and the next state s_{t+1} (or observation o_{t+1}), a binary observation u_i, which indicates that an unsafe outcome has happened if and only if $u_i = 1$. For clarity of exposition, the following section considers the problem of a sequence of MDP-CAs (in which a shield is a binary function of state and actions). Nonetheless, the proposed method can be applied to sequences of POMDP-CAs (as done in the empirical evaluation) by replacing states with observations.

5 RULE-BASED SHIELDING

Once an unsafe outcome is experienced by the agent, the objective is to avoid ever repeating the mistake which led to this outcome, as well as to avoid similar mistakes that would lead to the same outcome. There are several challenges to overcome in order to achieve this objective. First, even when given an indication that a catastrophic outcome has occurred, identifying the agent's mistake can be hard, especially if the outcome was observed long after the mistake was committed. Next, generalizing from one mistake to similar mistakes is a task that can be hard to achieve. Finally, there is an inherent conflict between the aim of having a compact mistakes representation to save memory and the need to quickly verify if a state-action pair is to be considered unsafe so that the overhead resulting from running the

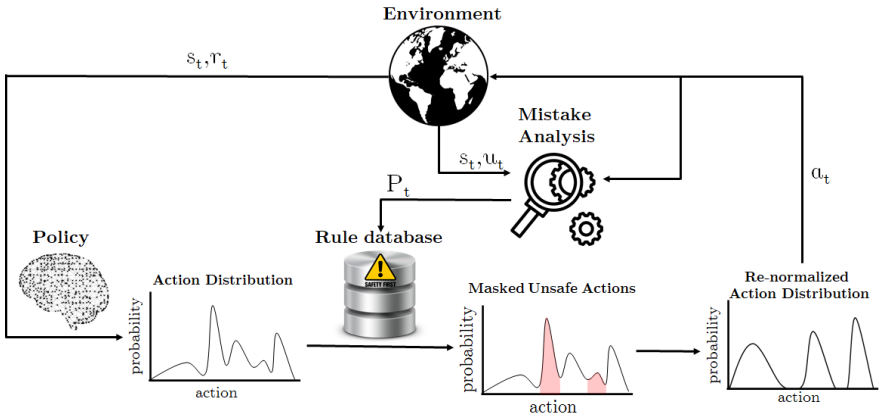


Figure 1: An illustration of the shielding via safety rules approach

shield is small. In this work, we sidestep these challenges by assuming that the agent can interact with an external Mistake Analysis Entity (MAE). This entity (which could be either a domain expert or an automated process) receives an indication that an unsafe outcome has occurred, as well as the trajectory of the agent. In return, the MAE identifies the mistake that caused the catastrophe and provides the agent with a set of *safety rules* corresponding to that mistake. A safety rule (or safety predicate) p can be treated as a small-scale shield, i.e., a binary function $p : S \times \mathcal{A} \to \{0, 1\}$, for which a state and action pair (s, a) is considered to be safe if $p(s, a) = 1$ and unsafe otherwise. In particular, when an MAE identifies a mistake (s, a) at time-step t, it returns a set of rules P_t which label the observed mistake as unsafe and do not introduce false positives (i.e., do not classify safe actions as unsafe). Formally, it is required that:

1. there exists a safety rule $p \in P_t$ such that $p(s, a) = 0$, and;
2. for all state-action pairs (s', a') and for all safety rules $p \in P_t$, $p(s', a') = 0 \Rightarrow L_\phi(s', a') = 0$.

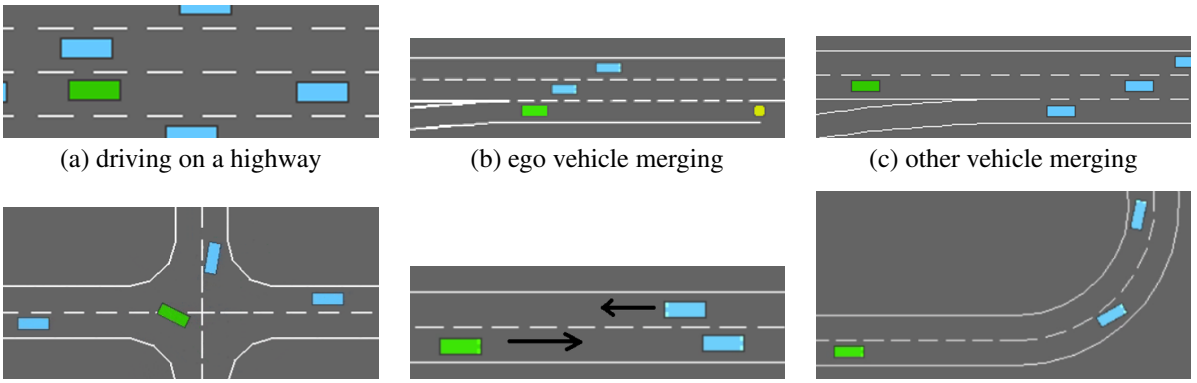
In addition, the combination of an MAE κ and a MDP-CA distribution $\mathbb{M}$ divides the set of mistakes into *equivalence classes* by defining an equivalence relation $\sim_{\kappa,\mathbb{M}}$. Two mistakes $m_1 = (s_1, a_1)$ and $m_2 = (s_2, a_2)$ are equivalent if by observing one mistake κ returns a set of rules which also capture the other. Formally, $m_1 \sim_{\kappa,\mathbb{M}} m_2$ if and only if for every set of rules P_t returned by κ: $\exists p \in P_t$ such that $p(m_1) = 0 \iff \exists p' \in P_t$ such that $p'(m_2) = 0$. This formulation captures the aspect that not all mistakes can be generalized (i.e., there could be a mistake that is not equivalent to any other mistake), and that different MAEs have different generalization capabilities (for example, given one trajectory that contains a mistake, the set of rules provided by one domain expert could account for more similar mistakes than the one provided by another).

The agent accumulates the different safety rules obtained from the MAEs at every time step into a rule database $\mathcal{P}$, which can be initialized with a predefined set of safety rules P_0. As a result, at every time-step t, $\mathcal{P} = \bigcup_{i=0}^{t} P_i$. This database is used as a shield $\mathcal{S}_\mathcal{P}$ in the following manner:

$$\mathcal{S}_\mathcal{P}(s, a) = \begin{cases} 0 & \exists p \in \mathcal{P} \text{ such that } p(s, a) = 0 \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

The complete method of using a rule-based shield is summarized below and illustrated in Figure 1. At each time-step t, the agent's policy, $\pi_t(s_t)$ encodes a probability distribution of actions which represents the current estimation of the probability that each action is the optimal action for the current state s_t. The shield $\mathcal{S}_\mathcal{P}$ is then applied to $\pi_t(s_t)$, i.e., the actions are verified against the rules in the database $\mathcal{P}$ in order to mask unsafe actions and π_t is renormalized to generate $\pi_t^{\mathcal{S}_\mathcal{P}}(s_t)$ (see Equation 1). Next, an action a_t is sampled from $\pi_t^{\mathcal{S}_\mathcal{P}}(s_t)$ and passed on to the environment, which returns to the agent a reward r_t and the next state, s_{t+1}. The action a_t and the new state s_{t+1} are also passed on to the Mistake Analysis Entity (MAE), along with an indication u_t of whether a catastrophic outcome has occurred. If such an outcome has indeed come to pass, the MAE provides a new set of rules P_t, which allow the shield to prevent the mistake at the root of the catastrophe from happening again, as well as mistakes that belongs to the same equivalence class. Note that the process of masking away the unsafe actions and renormalizing the action distribution could be replaced with a process of repeatedly sampling actions until finding an action verified to be safe against the rule database.

In this work, we sidestep the problem of obtaining a well-performing, automated MAE. Instead, the safety rules in our experiments are given by a domain expert (a co-author of the paper). Nonetheless, the focus of this work is not



(a) driving on a highway (b) ego vehicle merging (c) other vehicle merging

(d) turning left at an intersection (e) driving on a two-way road (f) making a u-turn

Figure 2: The set of scenarios that compose the multi-task autonomous driving behavioral control problem

the human-in-the-loop interaction, but rather introducing and demonstrating the rule-based framework for avoiding mistake repetition, which supports different types of MAEs. As a result, the assumption in this paper is that the benefit of learning to avoid any discovered mistakes is greater than the cost of interacting with the domain expert, as opposed to other work in which the concern is the cost of interaction (e.g., Saunders et al. (2018b)). This assumption is likely to hold in a domain like autonomous driving, in which many resources are invested, and learning to avoid mistakes could potentially save lives in the future. In addition, we note that domain-expert MAEs can violate the second requirement mentioned above of not introducing false positives. However, in practice, false positives which do not (significantly) affect the reward can be tolerated.

An important consideration is what to do in case that there are no safe actions to take. Here, the different catastrophic outcomes need to be evaluated with respect to different considerations, such as minimizing the negative impact (e.g., the risk of injury or the kinetic energy), ethical concerns (for instance, the trolley problem (Thomson, 1976)), and legality. A potential solution is to encode rules for ordering of catastrophic outcomes and to incorporate them into the default policy of the shield (as required by Equation 1). We leave the deliberation of deciding between different unsafe actions to future work. In this paper, we implemented a default policy which picks randomly among unsafe actions.

6 Case Study: The Highway Environment

We demonstrate and evaluate the approach of using rule-based shields to avoid repeating mistakes on a multi-task autonomous driving behavioral control problem, based on the Highway environment (Leurent, 2018). In this problem, the ego vehicle is faced with the following six different scenarios, which are illustrated in Figure 2: driving on a highway, merging into another lane, coping with another vehicle that is merging into the ego vehicle's lane, taking a left turn at an intersection, driving on a two-way road, and making a U-turn. Formally, the agent faces a new problem instance (POMDP-CA) at every episode, which is selected uniformly from the set of six scenarios. In addition, there are several control parameters for each scenario, e.g., the locations of all vehicles, the starting velocities, and the behavior of the other vehicles, whose values are also uniformly drawn at each episode. The observations the agent receives in all scenarios are the location, velocity, and heading of the ego vehicle and of the ten other vehicles which are closest to it. In addition, the agents can observe the location of obstacles on the road (e.g., the yellow block at the end of the merge lane in Figure 2(b)). The observation values of the ego vehicle are provided with respect to a global reference frame, while all other observation values are provided relative to the ego vehicle. The action space is composed of five actions: FASTER and SLOWER, which change the speed of the ego vehicle in $5\ \mathrm{m/s}$ interval increments, RIGHT_LANE and LEFT_LANE, which instruct the ego vehicle to switch lanes, and IDLE, which instructs the ego vehicle to maintain its current lane and velocity. Finally, there are some variations in the reward function between the different scenarios, as each scenario has a different objective, but all reward functions penalize collisions (which are also terminal states) and encourage maintaining high speeds. Note that for all scenarios and instances, the catastrophic outcomes are states in which there is a collision.

The usage of rules to ensure safety for autonomous vehicles has gained much momentum over recent years. In a 2018 report proposing an overall framework to measure and foster automated vehicles (AV) safety, RAND proposed formalizing the concept of roadmanship, an important leading measure of safety, via the definition of an automotive "safety envelope"—a set of boundary conditions under which the system must operate to ensure conformance with a prescribed safety concept (Fraade-Blanar et al., 2018). Desirable measures of roadmanship identified by RAND

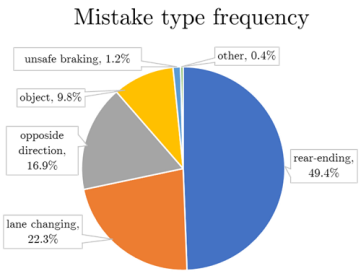
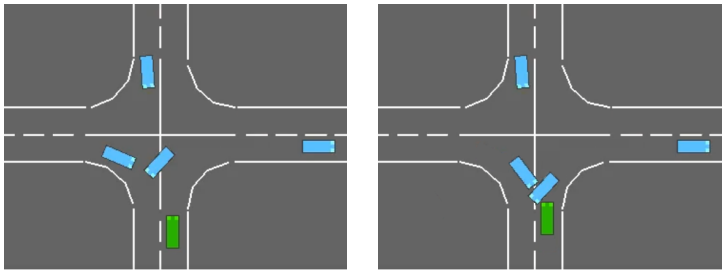


Figure 3: Frequency of mistakes



(a) before the collision (b) during the collision

Figure 4: An avoidable collision not covered by the RSS rules

include quantifying an AV's ability to abide by the official and unofficial rules of the road, the predictability and anticipatory nature of its behavior, the instances in which the AV is an initiator of or a responder to unsafe behavior, and the entropy caused by an AV's own behavior to that of the surrounding road users. The Responsibility Sensitive Safety (RSS) model proposed by Mobileye (Shalev-Shwartz et al., 2017), is a prominent example of a safety envelope that is derived from five behavior safety rules, the infringement of which can be used to measure quantities related to an AV's roadmanship. In related work, Censi et al. propose embedding safety in AV behavior via rulebooks, a unified hierarchical framework able to incorporate and prioritize rules derived from various sources such as traffic laws, common sense, liability, and ethics considerations (Censi et al., 2019). Rulebooks can be coupled with graph-based motion planning to directly provide trajectories that maximally respect the predefined rules. Rules with higher priority are placed at the top of the rulebook hierarchy, thus posing harder constraints on the planning solver. Extensions of this work, using rulebooks for planning via optimal control techniques have already been proposed in the literature (Xiao et al., 2021). It should be noted that in this work the authors focus more on the framework to integrate and optimize over rules, rather than on mechanisms to define such rules. It is stipulated that rules that guarantee safety of humans must be placed at the top of the hierarchy, and expressed analytically, and it is recognized that rules and priorities concerning safety and liability, must be informed by and derived from regulations and the public discourse. Finally, we note the RSS model is of particular interest, as it aims to set a standard for mandated autonomous driving safety, and, to the best of the authors' knowledge, is the only safety envelope model for AVs that has been made publicly available in its entirety. Consequently, the RSS model is currently the only viable source of comparison in the literature described above.

Despite the numerous efforts to design safety rules for AVs, none of the existing approaches is able to guarantee that an agent will avert all avoidable collisions. Furthermore, we hypothesize that due to the notoriously complex nature of the driving task, it is infeasible to define a set of rules that provides full coverage of safe behaviors across all possible driving scenarios a-priori. An initial set of rules can be designed to prevent collisions in most common scenarios, as well as in some rare ones. Yet, as the scenarios that agents run into are drawn from a long-tailed distribution, AVs will continue to encounter new situations, with some leading to avoidable collisions. Nonetheless, the rule-based shield approach ensures that once an AV has encountered a new situation and made a mistake that incurred a catastrophic outcome, all AVs sharing the same mistake database will be able to avoid repeating similar mistakes in kindred scenarios. Subsequently, the overall safety of agents would monotonically improve.

We note that the focus of this work is not to produce better safety rules for autonomous driving. Instead, we use the task of behavioral control for autonomous driving as a challenging continuous domain for the purpose of evaluating the rule-based shield approach, and to demonstrate that the process of learning from mistakes experienced by agents is essential for achieving safety, as some mistake-inducing scenarios cannot possibly be predicted in advance.

In order to apply the rule-based shield with a domain expert acting as an MAE, the cumulative collection of safety rules is externally stored and is accessed by the agent before every action deployment. Each time the agent encounters an unsafe situation during training (collision), the training process is paused, and a notification is sent to the expert along with the agent's trajectory and a recorded video of the episode. The corresponding set of safety rules is provided by the expert and added to the cumulative collection, after which the training process resumes. In our experiments, the safety rules were provided by a co-author of this paper and were encoded either as Python code or as propositional logic expressions. Finally, the number of times each rule actively prevents an agent from taking an unsafe action is counted, in order to estimate the frequency of each mistake type.

The main types (equivalence classes) of mistakes experienced by the agent during training are as follows: rear-ending the vehicle in front, colliding with another vehicle due to lane changing, hitting a vehicle travelling in the opposite direction, crashing into an object on the road, and braking leading to being rear-ended by a following vehicle unable to

stop on time; [3] the frequency of each mistake type experienced by the agents in our empirical evaluation can be found in Figure 3. Most of these mistakes are covered by the Responsibility-Sensitive Safety (RSS) model. For example, the most common mistake of rear-ending the leading vehicle can be avoided by a rule imposing a lower bound on the distance to be maintained from it such that the ego vehicle is guaranteed to be able to slow down sufficiently to avoid a collision in case the leading vehicle brakes: [4]

$$d_{\min} = \left[v_r \, \rho + \frac{1}{2} a_{\max,\text{accel}} \, \rho^2 + \frac{(v_r + \rho \, a_{\max,\text{accel}})^2}{2 a_{\min,\text{brake}}} - \frac{v_f^2}{2 a_{\max,\text{brake}}} \right]_+$$

where c_r is the rear vehicle, c_f is the vehicle in front of it, v_r, v_f are the longitudinal velocities of the cars, and $\rho, a_{\max,\text{brake}}, a_{\max,\text{accel}}, a_{\min,\text{brake}}$ correspond to the response time, the maximum deceleration of c_f, the maximum acceleration speed of c_r during the initial response time, and the minimum deceleration of c_r after the initial response time. The above constraint on the minimal distance can be translated to a few simple rules in the highway domain, depending on the direction in which the vehicle is heading and its relative position with respect to the other vehicle. An example of such rule is: if both vehicles are in the same lane, heading in the same direction, and the ego vehicle is behind another vehicle, then the ego vehicle cannot accelerate if its velocity after acceleration would cause a collision if the other vehicle were to brake now and the ego vehicle were to start braking at the next time step. This rule can be easily encoded, where the lane is determined by the relative position of the other vehicle, the direction is determined by the heading of the vehicles, and the distance criterion simplifies to: $(v_r+5)-(v_r+5)^2/10 < d+6-(v_f)^2/10$. In the latter criterion, d is the distance between the cars, which is given as part of the observation, 5 is the acceleration/deceleration value associated with the FASTER and SLOWER actions (which also induces the value in the denominators), and 6 is the length of the vehicle, a constant value in the highway domain. In our experiments, the rules did not introduce any hyperparameters, only information which is encoded in the state and domain knowledge; thus no tuning was required.

While the RSS model is gaining traction in the AV industry as a promising framework to ensure formal safety, we discovered instances of avoidable collisions which are not covered by RSS. For example, in one episode, shown in Figure 4, a collision between two vehicles at an intersection resulted in one vehicle being pushed in front of the ego vehicle, with the ego vehicle subsequently crashing into it. Although the collision between the two other vehicles can easily be predicted by the naive forward model assumed by the RSS, the RSS fails to account for the ensuing change in momentum of the leading vehicle, thus not averting an avoidable collision of the ego vehicle. In general, if the agent predicts that other vehicles are about to collide, it should exercise caution. This example supports our hypothesis that accounting for all safety concerns in advance is not feasible, and that it is crucial to experience some mistakes in order to learn to avoid making them.

7 Empirical Evaluation

In this section, we empirically study the effect of applying a rule-based shield on the learning process in the multi-task autonomous driving behavior control problem described in Section 6. For that aim, a rule-based shield is applied to the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017), resulting in a variant called PPO-RS. Recall that applying a shield during training affects the exploration process, as unsafe actions are not allowed even during data collection. In addition, a shield is only concerned with safety, yet the objective of the agents is to maximize return. Consequently, we are interested in quantifying the effect of the rule-based shield in terms of the rate of unsafe actions (mistakes) performed by agents and in terms of mean undiscounted episodic return.

To study the impact of using rule-based shields, PPO-RS is compared to the following six algorithms: 1) The vanilla version of PPO, which does not use a shield; 2) A variant of PPO that models Catastrophic Outcomes as Rewards (PPOCaR) by artificially assigning them highly negative values; 3) PPO-Lagrangian (Ray et al., 2019), a variant of PPO that uses Lagrangian methods to enforce safety constraints by using an adaptive penalty coefficient; 4) Constrained policy optimization (CPO) (Achiam et al., 2017)), which enforces constraints throughout training by solving trust region optimization problems at each policy update; 5) A tabular version of ShieldPPO that stores mistakes directly in memory, as done by Shperberg et al. (2022), and; 6) A parametric version of ShieldPPO that attempts to generalize over unsafe actions. Both PPO-Lagrangian and CPO require a hyperparameter C, which controls the unsafeness (represented as a limit on the commutative cost) of policies. Since the aim is to learn to avoid repeating collisions, we use a strict threshold value of $C = 0.01$ for both algorithms in our experiments.

A parametric version of ShieldPPO was proposed by Shperberg et al. (2022), yet it was neither implemented nor evaluated. One of the challenges in generalizing unsafe actions is the balance between the types of examples, i.e.,

[3] While in this case, the agent is not at blame, the agent should avoid the collision if possible.

[4] This rule applies for straight roads, for curved roads all values should be first transformed to a lane-based coordinate system.

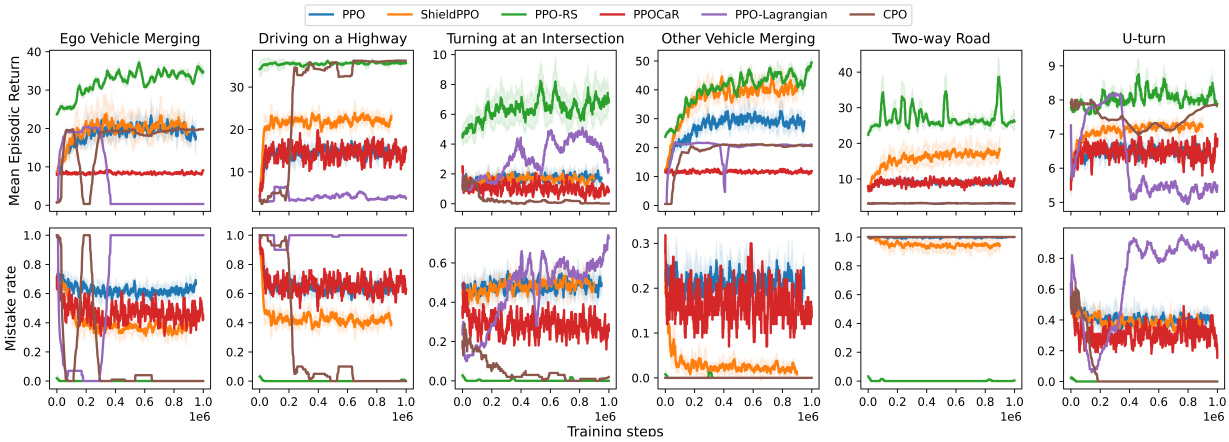


Figure 5: Single agent setting — mean undiscounted episodic return and cost rate achieved on the different tasks

the agent observes many more safe action effects than unsafe ones. For that reason, we used a buffer of safe actions effects ($\mathcal{B}_{\text{safe}}$), along with a separate buffer for unsafe actions effects ($\mathcal{B}_{\text{unsafe}}$). Then, at every update we sample an equal number of examples from each buffer and minimize the sum of the average binary cross entropy losses:

$$L(\psi) = \frac{1}{|\mathcal{B}_{\text{safe}}|}\mathbb{E}_{(o,a)\sim\mathcal{B}_{\text{safe}}}\left[\sum_{i=1}^{|\mathcal{B}_{\text{safe}}|}\log\left(\mathcal{S}_\psi(o,a)\right)\right] + \frac{1}{|\mathcal{B}_{\text{unsafe}}|}\mathbb{E}_{(o',a')\sim\mathcal{B}_{\text{unsafe}}}\left[\sum_{i=1}^{|\mathcal{B}_{\text{unsafe}}|}\log\left(1-\mathcal{S}_\psi(o',a')\right)\right], \quad (4)$$

where $S_\psi : \Omega \times \mathcal{A} \longrightarrow [0,1]$ is the shield, parameterized by ψ.

Our evaluation focuses on two experimental settings. In the first setting, a single agent is faced with a sequence of driving instances. In this experiment, we hypothesize that PPO-RS would result in a much lower rate of mistakes compared to all baselines. In addition, we expect PPO-RS to converge faster, as it collects data more efficiently. Moreover, we predict that PPO-RS would result in a higher asymptotic return than the baselines in some of the driving tasks, as its ability to better avoid mistake repetition would aid it to avoid some local optima.

The second experimental setting considers multiple agents which simultaneously face different driving tasks. Here, each agent comes up against its own sequence of POMDP-CAs, where the different sequences are drawn from the same underlying distribution. This setting was proposed by Shperberg et al. (2022) in an attempt to represent, for example, a fleet of autonomous cars, each driving in the same city every day. Here, we evaluate the effect of shield sharing by comparing a shield that is shared between all agents to the case where each agent has its own individual shield and the case in which agents do not use shields at all. Our expectation is that the mistake rate when sharing a shield should drop at a close to linear rate with respect to the number of agents compared to the case in which individual shields are used.

Each of the two experiments was repeated five times with different seeds. The results of the experiments are reported in the form of plots, which show the mean episodic return and mean rate of mistakes (collisions) achieved by each algorithm on each task in the multi-task autonomous driving problem. In addition, the standard error of the different runs is shown as a shaded area around each line in the corresponding plot.

The results of the first (single agent) experiment are reported in Figure 5. First, we note that the agent almost never visited the exact same state twice in any of the tasks, thus the tabular version of ShieldPPO performed the same as PPO. Consequently, only the parametric version of ShieldPPO is shown in the figure. When comparing ShieldPPO with PPO, it is evident that ShieldPPO outperforms PPO in most environments, both in terms of episodic return and in terms of rate of mistakes. This result is the first indication that ShieldPPO can learn to generalize over mistakes and be applied to continuous domains. Nonetheless, PPO-RS was able to significantly outperform both ShieldPPO and PPO by quickly reaching a mistake rate of almost zero, converging faster, and resulting in better asymptotic performance. Furthermore, PPO-RS is guaranteed not to repeat the same mistake more than once and not to classify safe actions as unsafe, in contrast to the parametric version of ShieldPPO which does not possess similar guarantees.

The performance of the three safe-RL baselines varies between the different tasks. The results of PPOCaR support previous finding and show that modeling safety via negative rewards is not effective. PPOCaR is outperformed by most other tested approaches, in some cases even by PPO. PPO-Lagrangian usually results in low episodic returns while still making many mistakes. By contrast, CPO often manages to converge to low mistake rates and in some cases

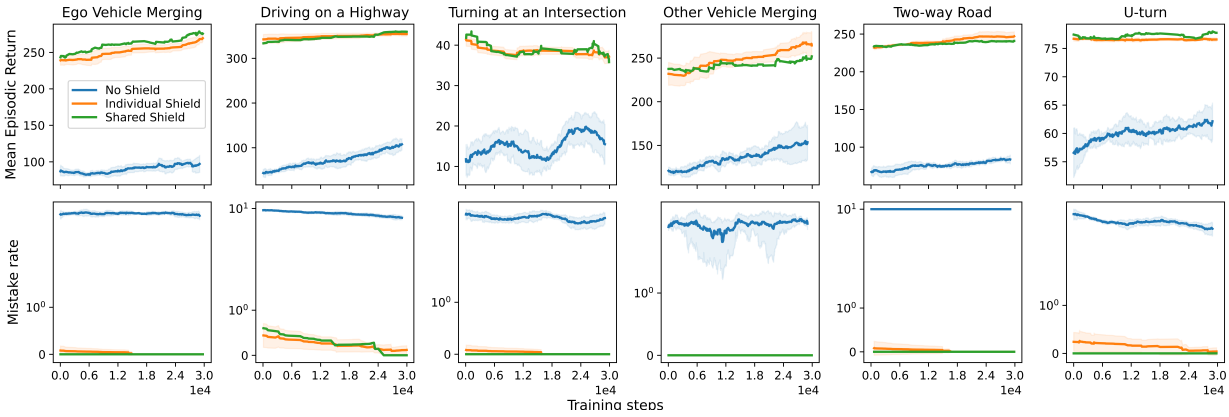


Figure 6: Multi agent setting — mean undiscounted episodic return and cost rate achieved on the different tasks

to achieve a high return. For example, in the task of driving on a highway, CPO was able to achieve an asymptotic episode return and cost rate which are similar to those of PPO-RS. Nonetheless, in all other tasks CPO was dominated by PPO-RS. Moreover, in some tasks CPO converges to a policy with a very low return (e.g., in the task of turning at an intersection CPO achieve the lowest return among all algorithms), and in the task of driving in a two-way road CPO converged to a policy that always collides with the vehicles going in the opposite direction.

A total of 28 rules were required in order to avoid mistake repetition in all five runs, where each run required between 24 and 26 of these rules. Most rules were added in the first $40,000$ time steps, and four other rules (on average) were gradually added to account for rarer mistakes (as depicted by the spikes in the mistake rate of PPO-RS).

In the second (multi-agent) experiment, we considered the sum of returns and the joint mistake-rate of ten different agents. Figure 6 show the results for each type of shield sharing: No Shield, in which agents do not use shield at all (i.e., running vanilla PPO), Individual Shield, in which each agent maintains its own copy of rule-based shield, and finally, Shared Shield, in which all agents use the same set of rules. In accordance with our hypothesis, the number of cumulative mistakes made by the agents is almost ten times lower when sharing a shield as opposed to using individual shields. Yet, the sharing of a shield did not significantly affect the rate at which the policy converges, as the time required to learn a good policy for each of the tasks dominates the time required to learn to avoid making most mistakes in the highway domain.

8 Discussion and Conclusion

This paper makes progress towards enabling autonomous agents to avoid mistake repetition. The approach taken in this paper is based on building a shield in an online manner in order to prevent agents from taking actions that were previously discovered to have catastrophic effects. This shield can be used for protecting an agent across multiple tasks and objectives. Moreover, this shield can be shared among several agents to further reduce the rate of mistakes. To this end, this paper makes three main contributions. First, an implementation of a parametric shield, which is suitable for continuous domains, is presented. This parametric shield is applied to PPO, resulting in the ShieldPPO variant, which greatly improves over PPO's performance on the highway autonomous driving multi-task problem. Furthermore, ShieldPPO uses observations alone and does not rely on domain expert input. The next contribution of the paper is the Rule-based Shield framework, which enables agents to receive safety rules in response to observed catastrophic outcomes from an external mistake analysis entity (MAE). These rules allow generalizing to an entire equivalence class based on a single trajectory while being compact and computationally efficient. Finally, the rule-based shield approach is applied to PPO, creating PPO-RS. Using a domain expert as an MAE, it is empirically shown that PPO-RS significantly dominates PPO, ShieldPPO, and other safe-RL baselines on the highway problem. In addition, the experiments demonstrate that even well-crafted safety rules, such as those defined in the Responsibility-Sensitive Safety model, cannot prevent all mistakes a priori. These results support the need for learning to avoid mistakes based on experience, which is the fundamental notion underpinning this work. An important challenge left for future work is making an automated MAE, which does not rely on human intervention. Generating a set of rules which are capable of capturing complete classes of mistakes based on one (of few) examples while not introducing false positives is a hard task. Nonetheless, a potentially promising direction is considering a symbolic representation or latent states as a ground for program synthesis methods.

ACKNOWLEDGEMENTS

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by the National Science Foundation (CPS-1739964, IIS-1724157, FAIN-2019844), the Office of Naval Research (N00014-18-2243), Army Research Office (W911NF-19-2-0333), DARPA, General Motors, Bosch, and Good Systems, a research grand challenge at the University of Texas at Austin. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, pp. 2669–2678. AAAI Press, 2018.

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

Julian Bernhard, Patrick Hart, Amit Sahu, Christoph Schöller, and Michell Guzman Cancimance. Risk-based safety envelopes for autonomous vehicles under perception uncertainty. *CoRR*, abs/2107.09918, 2021.

Nicolas Carrara, Edouard Leurent, Romain Laroche, Tanguy Urvoy, Odalric-Ambrym Maillard, and Olivier Pietquin. Budgeted reinforcement learning in continuous state space. In *NeurIPS*, pp. 9295–9305, 2019.

Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry S. Yershov, Scott Pendleton, James Guo Ming Fu, and Emilio Frazzoli. Liability, ethics, and culture-aware behavior specification using rulebooks. In *ICRA*, pp. 8536–8542. IEEE, 2019.

Yinlam Chow, Ofir Nachum, Edgar A. Duéñez-Guzmán, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *NeurIPS*, pp. 8103–8112, 2018.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerík, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *CoRR*, abs/1801.08757, 2018.

Ankush Desai, Shromona Ghosh, Sanjit A. Seshia, Natarajan Shankar, and Ashish Tiwari. SOTER: A runtime assurance framework for programming safe robotics systems. In *DSN*, pp. 138–150. IEEE, 2019.

Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. *arXiv preprint arXiv:1206.6404*, 2012.

FAA. Aviation accident & incident data. URL https://www.faa.gov/data_research/accident_incident/.

Laura Fraade-Blanar, Marjory S Blumenthal, James M Anderson, and Nidhi Kalra. *Measuring automated vehicle safety: Forging a framework*. 2018.

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

Jarrett Holtz, Simon Andrews, Arjun Guha, and Joydeep Biswas. Iterative program synthesis for adaptable social navigation. In *IROS*, pp. 6256–6261. IEEE, 2021.

Yoshinobu Kadota, Masami Kurano, and Masami Yasuda. Discounted markov decision processes with utility constraints. *Comput. Math. Appl.*, 51(2):279–284, 2006.

Edouard Leurent. An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env, 2018.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. IPO: interior-point policy optimization under constraints. In *AAAI*, pp. 4940–4947. AAAI Press, 2020.

Helmut Mausser and Dan Rosen. Beyond var: from measuring risk to managing risk. In *CIFEr*, pp. 163–178. IEEE, 1999.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *ICML*. icml.cc / Omnipress, 2012.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7, 2019.

William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *AAMAS*, pp. 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018a.

William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *AAMAS*, pp. 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *CoRR*, abs/1708.06374, 2017.

Shahaf S. Shperberg, Bo Liu, and Peter Stone. Learning a shield from catastrophic action effects: Never repeat the same mistake, 2022.

Amolika Sinha, Sai Chand, Vincent Vu, Huang Chen, and Vinayak Dixit. Crash and disengagement data of autonomous vehicles on public roads in california. *Scientific data*, 8(1):1–10, 2021.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: safe reinforcement learning with learned recovery zones. *IEEE Robotics Autom. Lett.*, 6(3):4915–4922, 2021.

Judith Jarvis Thomson. Killing, letting die, and the trolley problem. *The monist*, 59(2):204–217, 1976.

Wei Xiao, Noushin Mehdipour, Anne Collin, Amitai Bin-Nun, Emilio Frazzoli, Radboud J. Duintjer Tebbens, and Calin Belta. Rule-based optimal control for autonomous driving. In *ICCPS*, pp. 143–154. ACM, 2021.