Adversarial Behavior Exclusion for Safe Reinforcement Learning

Md Asifur Rahman , **Tongtong Liu,** and **Sarra Alqahtani**
Department of Computer Science, Wake Forest University
{rahmm21, liut18, sarra-alqahtani}@wfu.edu

Abstract

Learning by exploration makes reinforcement learning (RL) potentially attractive for many real-world applications. However, this learning process makes RL inherently too vulnerable to be used in real-world applications where safety is of utmost importance. Most prior studies consider exploration at odds with safety and thereby restrict it using either joint optimization of task and safety or imposing constraints for safe exploration. This paper migrates from the current convention to using exploration as a key to safety by learning safety as a robust behavior that completely excludes any behavioral pattern responsible for safety violations. Adversarial Behavior Exclusion for Safe RL (AdvEx-RL) learns a behavioral representation of the agent's safety violations by approximating an optimal adversary utilizing exploration and later uses this representation to learn a separate safety policy that excludes those unsafe behaviors. In addition, AdvEx-RL ensures safety in a task-agnostic manner by acting as a safety firewall and therefore can be integrated with any RL task policy. We demonstrate the robustness of AdvEx-RL via comprehensive experiments in standard constrained Markov decision processes (CMDP) environments under 2 white-box action space perturbations as well as with changes in environment dynamics against 7 baselines. Consistently, AdvEx-RL outperforms the baselines by achieving an average safety performance of over 75% in the continuous action space with 10 times more variations in the testing environment dynamics. By using a standalone safety policy independent of conflicting objectives, AdvEx-RL also paves the way for interpretable safety behavior analysis as we show in our user study.

1 Introduction

In the last two decades, RL has greatly evolved demonstrating its potential in a wide range of applications including robotics [Kober *et al.*, 2013], and autonomous driving [Grigorescu *et al.*, 2020]. To be deployed in the real world, ensuring safety is a crucial factor that RL inherently lacks due to its exploratory learning. One way safety can be ensured in RL is by modifying its objective to optimize both the safety goals and the task learning [Kim *et al.*, 2020] [Geibel, 2006]. In this approach, the RL agent must explore a significant number of safety-violating states which inherently leads to sub-optimal policies due to the conflict between the task learning and safety objectives [Thananjeyan *et al.*, 2021]. Another way for ensuring RL safety is to enforce safe exploration by endowing explicit constraints. Manual specification of such constraints [Levine *et al.*, 2018] though possible in environments with known dynamics, cannot be generalized to any slight changes in those environments. Besides, safety specifications based on estimating the environment dynamics during offline learning [Bastani, 2021] [Alshiekh *et al.*, 2018] are not sufficient to assure that the RL agent will behave safely during runtime for two reasons. First, the details of the environment dynamics can't be fully known at training time, which partially invalidates the initial assumptions about the environment modeling used to design the safety specifications. Second, RL agents are susceptible to even subtle perturbations in their observations and actions, known as adversarial examples [Chen *et al.*, 2019] [Lee *et al.*, 2020] which introduces novel perturbations in the environment model. Although a vulnerable policy under adversarial attacks cannot be regarded as truly safe in the physical world, there is little research studying the robustness of the safe RL methods against those attacks. In this paper, we investigate the following question: *how can we ensure a robust safety for the RL agent without impairing its task learning under deliberate perturbations?*

We propose the Adversarial Behavior Exclusion (AdvEx-RL) framework for safe RL. AdvEx-RL first trains an adversarial policy to interactively extract unsafe behaviors by maximizing the safety violations in the environment. Then, it learns a safety policy by maximizing its divergence from the adversarial policy. This approach is different from adversarial training which requires the agent to learn its task in a zero-sum game with the adversary. Instead, we train the adversarial policy to estimate the safety of each encountered state and to learn the underlying behavior most likely violates the safety constraints in the environment. Our contributions are three-folded: (1) developing a task-agnostic safety learning framework, AdvEx-RL, where the RL agent can use it as a safety firewall to avoid unsafe behaviors, (2) introducing

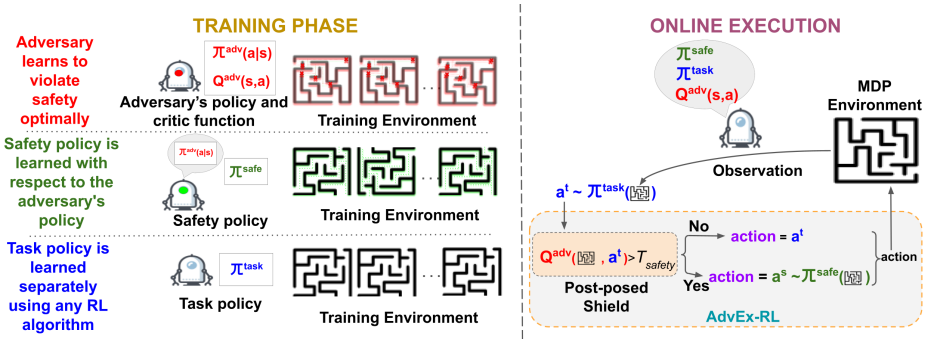

Figure 1: AdvEx-RL Safety Framework

a new safety learning method separate from task learning by deriving a state-action distribution that is most divergent from the state-action distribution of an approximated optimal adversary, and (3) providing a theoretical proof for the safety of AdvEx-RL along with empirical evaluations against 7 baselines in 3 continuous MuJoCo environments from [Thananjeyan *et al.*, 2021] and SafetyGym environments [Ray *et al.*, 2019] . Unlike prior safe RL studies, in our empirical evaluation, we consider factors such as robustness and frequency of deadlocks. We also demonstrate the interpretability of AdvEx-RL through a study conducted on 41 end users using an extended version of CAPS [McCalmon *et al.*, 2022].

2 Related Work

We adopt 3 algorithmic concepts from the literature: (1) safe exploration via online shielding (2) value function-based safety estimation, and (3) the use of two policies. AdvEx-RL integrates the shielding concept from [Alshiekh *et al.*, 2018; ElSayed-Aly *et al.*, 2021; Bansal *et al.*, 2017; Fisac *et al.*, 2019] to prevent the agent from visiting unsafe states during runtime. We employ a post-posed [Alshiekh *et al.*, 2018] shield in AdvEx-RL, but instead of deriving the safety probabilities using formal methods, we use the value-function based safety estimation [Geibel and Wysotzki, 2005], [Hans *et al.*, 2008; Srinivasan *et al.*, 2020; Thananjeyan *et al.*, 2021].

In [Geibel and Wysotzki, 2005; Hans *et al.*, 2008], Q-learning is used for the risk estimation and to derive the risk-averse and rescue policies [Mihatsch and Neuneier, 2002]. The safety violations is estimated in [Srinivasan *et al.*, 2020] through a safety critic implemented using DQN. Then, the safety Q-function estimation is utilized to optimize a Lagrangian relaxation (LR) objective to derive a safety policy. [Srinivasan *et al.*, 2020] shows that any policy constrained under a safety Q-function is guaranteed to be safe. RecoveryRL [Thananjeyan *et al.*, 2021] adopts the concept of safety Q-function [Srinivasan *et al.*, 2020] to develop a shielding mechanism. They use offline demonstration data collected from human-supervised policy to train a safety estimator, Q-risk then derives a model-free recovery policy by defining an LR objective function that minimizes the Q-risk. The performance of [Thananjeyan *et al.*, 2021] is greatly reliant on the human-supervised offline data. In AdvEx-RL we employ a similar shielding mechanism by applying a threshold on the safety estimation from a critic network [Srinivasan *et al.*, 2020; Thananjeyan *et al.*, 2021]. However, instead of training a different DQN safety critic through exploring a pre-training environment [Srinivasan *et al.*, 2020] or on human supervised data [Thananjeyan *et al.*, 2021], we acquire the safety critic from the critic network of a trained adversary.

In AdvEx-RL, we follow the strategy of using two separate policies i.e. task policy and safety policy similar to[Bastani, 2021; Thananjeyan *et al.*, 2021]. In [Bastani, 2021], a shield mechanism is used to switch between the task and safety policies. The safety policy is derived using a non-linear model predictive controller (NMPC) that in the advent of probable safety violations, resets the agent to some fixed initial safety point within the environment. Whereas [Thananjeyan *et al.*, 2021] uses a similar concept but instead of resetting the agent to some fixed initial point, they reset the agent to nearby safe points. The safety policy in NMPC [Bastani, 2021] and MPC [Thananjeyan *et al.*, 2021] requires prior knowledge about the environment dynamics or demonstration data.

3 Problem Statement

We consider the standard CMDPs, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mu, \mathcal{P}(.|.,.), \mathcal{R}, \gamma, \mathcal{C})$ where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space; μ and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denote the initial state distribution and state transition dynamics respectively. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ is the reward function; γ denotes discount factor; and $\mathcal{C} = \{c_i : \mathcal{S} \times \mathcal{A} \xrightarrow{\mathcal{S}} \mathbb{R} \geq 0;\ i = 1, 2, .., T\}$ denotes the set of cost associated with safety constraint violations in any trajectory episode $\tau = \{s_0, a_0, ..., a_{T-1}, s_T\}$ with a maximum trajectory length of T. We assume that accomplishing the task goal or violating a safety constraint in $\mathcal{M}$ leads to episode termination. Let Π be the set of stationary MDP policies for $\mathcal{M}$ such that $\pi^{task}, \pi^{adv}, \pi^{safety} \in \Pi$ denote the task policy, adversary policy and safety policy respectively. The objective of task policy π^{task} is to learn the optimal control to maximize the expected discounted reward at time t; $\mathcal{R}_{\pi^{task}} = \mathbb{E}_{\tau \sim \pi^{task}}[\sum_{t'=t}^{T} \gamma^{t'-t} r_t]$. On the other hand, the objective of adversary policy π^{adv} is to maximize the expected discounted cost associated with safety violations $\mathcal{C}_{\pi^{adv}} = \mathbb{E}_{\tau \sim \pi^{adv}}[\sum_{t'=t}^{T} \gamma^{t'-t} c_t]$. AdvEx-RL works like a protective safety firewall integrated with a safety shield allowing task policy π^{task} and safety policy π^{safety} to be

learned and executed completely independent of each other as depicted in Fig.1.

4 Adversarial Behavior Exclusion for Safe Reinforcement Learning

AdvEx-RL (Fig.1) uses a post-posed shielding mechanism to assess the agent's current safety and act appropriately following either its task policy or safety policy. The task policy is any conventional RL policy optimized to learn a certain task while the safety policy is a task-agnostic policy that aims to only maximize the agent's safety in a certain environment. When the task policy takes the agent closer to a potentially dangerous/unsafe state, the safety policy is triggered by the shield to rescue the agent to a nearby safe state.

4.1 First, Learn to be Unsafe

AdvEx-RL first extracts unsafe behavior by interactively training an optimal adversary in $\mathcal{M}$. The adversary, similar to any conventional RL, learns an optimal policy π^{adv} to maximize the accumulated cost associated with the safety violations through the exploration-exploitation principle of RL. To improve the exploration of the adversarial policy, we use off-policy learning with entropy regularization by maximizing the following objective:

$$J(\phi) = \mathbb{E}_{(s_t,a_t)\sim\rho^{\pi^{adv}}}[c(s_t,a_t) + \alpha\mathcal{H}(\pi^{adv}_{\phi}(.|s_t))] \quad (1)$$

where $\mathcal{H}(\pi^{adv}_{\phi}(.|s_t))$ is the policy entropy and α is the temperature parameter. Using the maximum entropy policy objective [Haarnoja *et al.*, 2018a], the optimal state-action function $Q^{adv^*}(s,a)$ can be approximated through soft Q-function that is evaluated with Bellman backup operator $\tau^{\pi_{adv}}$ as:

$$\tau^{\pi^{adv}}Q^{adv}(s_t,a_t) \triangleq c(s_t) + \mathbb{E}_{s_{t+1}\sim p}[V(s_{t+1})] \quad (2)$$

where,

$$V(s_t) = \mathbb{E}_{a_t\sim\pi_{adv}}[Q^{adv}(s_t,a_t) - log\pi^{adv}(a_t|s_t)] \quad (3)$$

The optimal $Q^{adv^*}(s,a)$ provides the estimation of the expected cost over any trajectory $\tau \in \mathcal{M}$; $Q^{adv^*}(s,a) = \mathbb{E}_{\tau\sim\pi^{adv*}}[\sum_{t'=t}^{T}\gamma^{'-t}c(s_t)]$. Therefore, we approximate $Q^{adv}_{\varphi}(s,a)$ by minimizing soft bellman residual [Haarnoja *et al.*, 2018a] and use it to quantify safety violation as we will show in Section 4.3 . (Details on the training of the adversarial policy are given in Appendix[1] A, Algorithm 1.)

4.2 Learn to be Safe from The Worst Behavior

The second phase of AdvEx-RL is to learn a task-agnostic safety policy exploiting the adversarial policy π^{adv} under the following assumptions:

Assumption 1: Given CMDP $\mathcal{M}$, with the state space $S = S^{safe} \cup S^{unsafe}$, if a state $s \in S^{safe}$ have any neighboring unsafe state $\tilde{s} \in S^{unsafe}$, then the transition probability denoting safety violation upon taking an available action a at that state $P(\tilde{s}|s,a) > 0$.

[1]Appendix link

Assumption 2: The adversarial policy has sufficiently explored $\mathcal{M}$ and is optimal π^{adv^*} such that $Q^{adv^*} \geq Q^{adv'}$ where $\pi^{adv'}$ is any sub-optimal adversary policy. Therefore $Q^{adv^*}(.)$ can quantify the expected cost for any trajectory $\tau \in \mathcal{M}$.

Assumption 3: Every state has at least one neighboring safe state i.e. at any state s_t, there exists at least one safe action that can lead the agent to a neighboring safe state; $\exists a_t$ s.t $\mathcal{P}(s_{t+1}|s_t,a_t) > 0$ where $s_{t+1} \in S^{safe}$.

Theorem 1 (Reduction of safety violation probability): The probability of following a trajectory that violates safety can be reduced by increasing KL divergence between the adversarial policy π^{adv} and any arbitrary policy π'.

Proof

Let's consider the objective of the optimal adversary in $\mathcal{M}$ which maximizes the expected cost (Eq.1). Stationary policy $\pi^{adv} \sim \pi^{adv^*}$ represents a one-to-one correspondence [Puterman, 2014] with the state-action distribution in $\mathcal{M}$ which can be computed by:

$$\rho^{\pi^{adv}}(s,a) = \mu(s_0)\prod_{t=1}^{T}\mathcal{P}(s_t|s_{t-1},a_t)\pi^{adv}(a_t|s_{t-1}) \quad (4)$$

From inverse reinforcement learning principle [Ghasemipour *et al.*, 2019], the expected return of an arbitrary policy $\pi^{'}$ can be computed with respect to the optimal adversary policy (i.e. expert) π^{adv} as:

$$\begin{aligned}\mathbb{E}_{\tau\sim\pi'}[\sum_t c(s_t,a_t)] &= \mathbb{E}_{\tau\sim\pi'}[\sum_t \frac{\rho^{\pi^{adv}}(s_t,a_t)}{\rho^{\pi'}(s_t,a_t)} \\ &\quad \cdot log\frac{\rho^{\pi^{'}}(s_t,a_t)}{\rho^{\pi^{adv}}(s_t,a_t)}] \\ &\propto \mathbb{E}_{(s,a)\sim\rho^{\pi'}(s,a)}[\frac{\rho^{\pi^{adv}}(s_t,a_t)}{\rho^{\pi^{'}}(s_t,a_t)} \\ &\quad \cdot log\frac{\rho^{\pi^{'}}(s_t,a_t)}{\rho^{\pi^{adv}}(s_t,a_t)}] \\ &= -D_{KL}(\pi^{'}||\pi^{adv})\end{aligned} \quad (5)$$

This intuitively means that, as an arbitrary policy $\pi^{'}$ becomes more similar to an optimal adversary π^{adv} by minimizing the KL divergence $D_{KL}(\pi^{'}||\pi^{adv})$, the expected cost associated with safety violations increases. Meanwhile, according to the probabilistic inference [Levine, 2018], the probability of choosing a trajectory τ involving safety violation under π^{adv} can be given by

$$p_{\pi^{adv}}(\tau) \propto \mathbb{E}(\sum_t \gamma^t c_t) \quad (6)$$

Then optimal π^{adv} can be denoted as:

$$\begin{aligned}logp_{\pi^{adv}}(\tau) &= log\int p_{\pi^{adv}}(\tau)d\tau \\ &\geq \mathbb{E}_{\tau\sim\pi'}[\sum_t \gamma^t c_t)] - D_{KL}(\pi^{'}(\tau)||\pi^{adv}(\tau))\end{aligned} \quad (7)$$

The KL divergence term in the above equation acts like a penalty regularization [Goo and Niekum, 2022] that guides the arbitrary policy $\pi^{'}$ closer to the optimal adversary i.e $D_{KL} \rightarrow 0$ which in turn maximizes the expected cost. This also indicates that if the trajectory distribution under the arbitrary policy $\pi^{'}$ diverges from the trajectory distribution of the optimal adversary, it will then minimize the expected cost. Since diverging from a trajectory distribution is lower bounded by its state distribution [Ke *et al.*, 2021] and considering assumption 3, maximizing KL divergence of the arbitrary policy with respect to the optimal adversary policy guarantees to reduce the probability of selecting a trajectory leading to safety violations.

We use this theorem as the learning principle of the safety policy, where the state-action distribution of policy π^{safe} is derived by maximizing its KL-divergence from π^{adv} according to the following objective:

$$J(\theta) = \underset{\theta}{argmax}\, E_{\tau\sim\pi^{safe}}[\mathcal{D}_{KL}(\pi_{\theta}^{safe}(\tau)||\pi_{\phi}^{adv}(\tau))] \quad (8)$$

π^{safe} is trained by taking samples from a set of trajectories $\tau_{\pi^{safe}}$; where the distribution of any trajectory $\tau = \{s_0, a_0, s_1, a_1, ..., a_T, s_T\}$ under π^{safe} is given by $\rho^{\pi^{safe}}(\tau) = \mu(s_0)\prod_{t=1}^{T}\pi^{safe}(a_t|s_t)\mathcal{P}(s_{t+1}|s_t, a_t)$ then:

$$\begin{aligned}\mathcal{D}_{KL}(\pi^{safe}(\tau)||\pi^{adv}(\tau)) = &\sum_{\tau\sim\tau_{\pi^{safe}}} \rho^{\pi^{safe}}(\tau) \\ &log[\prod_{t}\frac{\pi^{safe}(a_t|s_t)}{\pi^{adv}(a_t|s_t)}] \\ &= \mathbb{E}_{(s_t,a_t)\sim\pi^{safe}}[log\pi^{safe}(a_t|s_t) \\ &- log\pi^{adv}(a_t|s_t)]\end{aligned} \quad (9)$$

Here the KL divergence $\mathcal{D}_{KL}(\pi^{safe}(\tau)||\pi^{adv}(\tau))$ is upper bounded by Pinsker's inequality i.e. the square root of total variational distance between the trajectory distribution under π^{safe} and π^{adv}. To ensure that π^{safe} is not only divergent from π^{adv} but also will rescue the agent to a safe state, we change the objective in Eq.8 to state-pairwise safety learning by optimizing π^{safe} as:

$$\begin{aligned}J(\theta) = \underset{\theta}{argmax}\, \mathbb{E}[&\sum_{\tau^{safe}, s_t\sim\tau^{safe}, a_t\sim\pi^{safe}(s_t)} log\pi_{\theta}^{safe}(a_t|s_t) \\ &- (Q_{\psi}^{est_{adv}}(s_t, a_t) - log\pi_{\phi}^{adv}(a_t|s_t))]\end{aligned} \quad (10)$$

Where $Q^{est_{adv}}(s_t, a_t)$ is derived by minimizing objective:

$$\begin{aligned}J(\psi) = \sum_{(s_t,s_{t+1})\sim\tau^{safe},\, a_{t+1}\sim\pi^{safe}(s_{t+1})}&[\frac{1}{2}(Q_{\psi}^{est_{adv}}(s_t, a_t) \\ &- (\mathcal{C}(s_t) + \gamma Q_{\varphi}^{adv}(s_{t+1}, a_{t+1})))^2]\end{aligned} \quad (11)$$

Maximizing Eq.10 ensures that the trajectory distribution of the safety policy π^{safe} is diverse from the trajectory distribution of π_{ϕ}^{adv} while increasing the probability of sampling trajectories that have lower expected safety violation cost. Consequently, π^{safe} will rescue the agent to nearby states that have lower values for the adversary or in other words safer for the agent.

Sampling trajectories directly under π^{safe} would lead to poor exploration. Therefore, we train π^{safe} using a roll-out mechanism. This mechanism generates a demo trajectory τ_{demo} using either the adversarial policy π_{ϕ}^{adv}, the task policy π^{task}, or randomly sampled actions. For each state $s_t \in \tau_{demo}$; safety policy π^{safe} generates new trajectories $\tau_{\pi^{safe}} \sim (s_t, s_{t+1}, C(s_t))$ which are then used to train the safety policy π^{safe} by maximizing the objective function of Eq.10 in an off-policy fashion. (More details on AdvEx-RL safety policy training can be found in Appendix B, Algorithm 2).

4.3 Online Execution with Safety Shielding

We adopt the strategy of using a post-posed shield from [Alshiekh *et al.*, 2018] only during the online execution. Unlike [Srinivasan *et al.*, 2020; Thananjeyan *et al.*, 2021], which separately trains a DQN safety critic function Q_{risk}^{π} for safety estimation, we instead utilize the critic function of the optimal adversary Q_{φ}^{adv} that we trained previously (Eq.2) to implement the safety shield as:

$$Shield(s_t, a_t) : Q_{\varphi}^{adv}(s_t, a_t) > \mathbb{T}_{safety} \quad (12)$$

where $\mathbb{T}_{safety}$ is a predefined threshold value such that at any state s_t and for any action $a_t \sim \pi^{task}(s_t)$; if $Shield(s_t, a_t)$ is triggered, then the AdvEx-RL safety firewall replaces the selected action a_t by a safer action given by the safety policy $a_t^{safe} \sim \pi^{safety}(s_t)$. The value of $\mathbb{T}_{safety}$ is environment-specific and can be chosen based on a sensitivity test for each environment (see Appendix C for details about the sensitivity test. Algorithm 3 in Appendix D shows the online execution of AdvEx-RL.)

Deadlock Side Effect: Providing safety in 2 separate policies is prone to deadlock as the agent may loop between the same states due to switching between π^{safe} and π^{task}. Suppose at any timestep t, for action $a_{t_c} \sim \pi^{task}(.|s_{t_c})$ at a critical state s_{t_c}, the shield is triggered. From this state s_{t_c} onward, the shield will make sure that the agent keeps following a safe trajectory using π^{safe} until it reaches a safe state s_{t_s} such that for action $a_{task} \sim \pi^{task}(.|s_{t_s})$; $Q_{\varphi}^{adv}(s_{t_s}, a_{task}) < T_{safety}$ is satisfied. Afterward, the agent can select actions using its task policy π^{task} unless the shield is triggered again. However, due to the presence of external perturbation or inherent weakness within the task policy π^{task}, the agent might cycle back to the same old critical state s_{t_c} while selecting action a_{t_c}; resulting in an inadvertent deadlock.

Although deadlock can hamper the agent's task, it can save the agent from dangerous situations. For example, a deadlock can be a safe, temporary solution for an expensive robot until it gets rescued by human operators. To examine the proposed AdvEx-RL for deadlocks, we empirically analyze the frequency of deadlocks in the tested environments using the deadlock detection proposed in [Ye *et al.*, 2022].

5 Practical Implementation

The adversarial policy was trained using SAC [Haarnoja *et al.*, 2018b] since it provides better exploration. The task policy was also trained using SAC but it can be trained using any

RL algorithm. The safety policy was trained by performing gradient descent on the objective function in Eq.10. The post-posed shield was implemented as a safety assurance layer that explicitly replaces any unsafe action selected by the task policy with a safe action chosen by the safety policy during execution.

6 Experiments

The experiments in this paper are conducted to answer the following questions: **(1)** how robust is AdvEx-RL compared to the baselines under deliberate uncertainty in form of external perturbations and altered environment dynamics? **(2)** how does AdvEx-RL's safety policy affect the agent's task performance? **(3)** how much does the safety policy contribute to the safety of AdvEx-RL? (ablation analysis) **(4)** how often does the deadlock occur in AdvEx-RL and the baselines? and **(5)** how transparent and interpretable is the behavior generated by AdvEx-RL to the end users? All the codes[2] relevant to the experiments are available online.

6.1 Environments

We conducted our experiments on three continuous MuJoCo CMDPs [Thananjeyan *et al.*, 2021] (i) Maze (ii) Navigation 1, and (iii) Navigation 2. In these environments, the agent's task is to reach the goal state while avoiding collisions with obstacles, walls, or boundaries. In addition, we also conducted experiments on SafetyGym environments [Ray *et al.*, 2019]. (See Appendix E for more details about the environments.)

6.2 Baselines

We have tested AdvEx-RL against 7 baselines; SAC (without any safety measures), Lagrangian Relaxation (LR) [Thananjeyan *et al.*, 2020], Safety Q-Functions for RL (SQRL) [Srinivasan *et al.*, 2020], Risk Sensitive Policy Optimization (RSPO) [Mihatsch and Neuneier, 2002], Critic Penalty Reward Constrained Policy Optimization (RCPO) [Tessler *et al.*, 2018], Reward Penalty (RP) [Thananjeyan *et al.*, 2021], and Recovery RL Model Free (RRL-MF) [Thananjeyan *et al.*, 2021]. (More details on the baselines are in Appendix F. In addition, see Appendix G for further implementation details of AdvEx-RL and the baselines.)

6.3 Performance Metrics

Assuming a maximum episode length T_{max} in any environment, the following cases might happen: (1) the agent accomplishes its task within T_{max} without any safety violations, (2) it violates at least one safety constraint and terminates, or (3) it exhausts T_{max} without accomplishing its task or violating any safety constraints. Considering these cases, we use the following two performance metrics in our experiments: **(i) Safety(%):** This metric measures the portion of time the agent acts safely over its maximum episode length. Given T_{max}; the function $F(.)$ counts the total number of timesteps before the episode termination caused by safety violation. Then Safety% is measured as a function of trajectory τ:

$$Safety(\%) = \begin{cases} \frac{F(\tau)}{T_{max}} \times 100 & ; \text{ if } \exists s_t \sim \tau \text{ e.g. } \mathcal{C}(s_t) > 0 \\ 1 \times 100 & ; \text{ otherwise} \end{cases} \tag{13}$$

(ii) Success-Safety(%): Assuming the AdvEx-RL agent's task is to reach a goal state $\mathbb{G}$, both the reward and success are measured in terms of how close the agent is to $\mathbb{G}$. The agent is considered successful in accomplishing its task when the episode ends while it is within a predefined minimum distance from $\mathbb{G}$ $Min_{distance}$. If the maximum distance from $\mathbb{G}$ is given by $Max_{distance}$ and the agent's current Euclidean distance from the goal is $\mathbb{D}(s_t, \mathbb{G})$, then the Success-Safety % measures the trade-off between safety and task objectives by:

$$Success - safety(\%) = \frac{\mathbb{D}(s_t, \mathbb{G}) - Max_{distance}}{Min_{distance} - Max_{distance}} \times Safety(\%) \tag{14}$$

6.4 Robustness Analysis

While prior safe RL works focus on the performance optimality, we argue that optimality is not enough and we have to study the robustness of the safety solutions against deliberately crafted perturbations. Therefore, we evaluate the robustness of the safety performance of the baselines and AdvEx-RL under different magnitudes and types of uncertainty. We injected two types of uncertainty into the testing environments; external action space perturbations and changes to the environment dynamics (such as air resistance and noise). For the first type, we kept the environment dynamics the same as the training environment while incorporating two white-box action space perturbations; random and alternative adversarial action (AAA) perturbations from [Tessler *et al.*, 2019]. (See Appendix H for details.)

6.5 Results

Evaluation of Safety Robustness: The results of the safety robustness analysis are presented in Fig.2 and Fig.3. The results are averaged over 100 test episodes. Evidently, the baselines showed optimal safety performance under no external perturbations or dynamics changes. However, a significant deterioration of both safety(%) and success-safety(%) can be seen for most of the baselines with increasing in perturbation rates. Performance deterioration is more visible in the case of the AAA perturbation than random perturbation, which indicates that the prior safe RL techniques are not robust against deliberate perturbations. Moreover, changes in the environment dynamics negatively impact the safety robustness of the baselines as well as AdvEx-RL. Interestingly, in Fig.2, we observe that SAC provides reasonable safety which is due to its default robustness properties [Eysenbach and Levine, 2021]. RRL-MF provides the second-best safety robustness, particularly for higher perturbation rates and the reason for this could be its usage of explicit unsafe demonstration data provided by experts during its training. In both uncertainty types, AdvEx-RL clearly outperforms the baselines with a minimum safety(%) over 75% and success-safety(%) over 80%. This empirically proves the robustness of AdvEx-RL

[2]https://github.com/asifurrahman1/AdvEx-RL

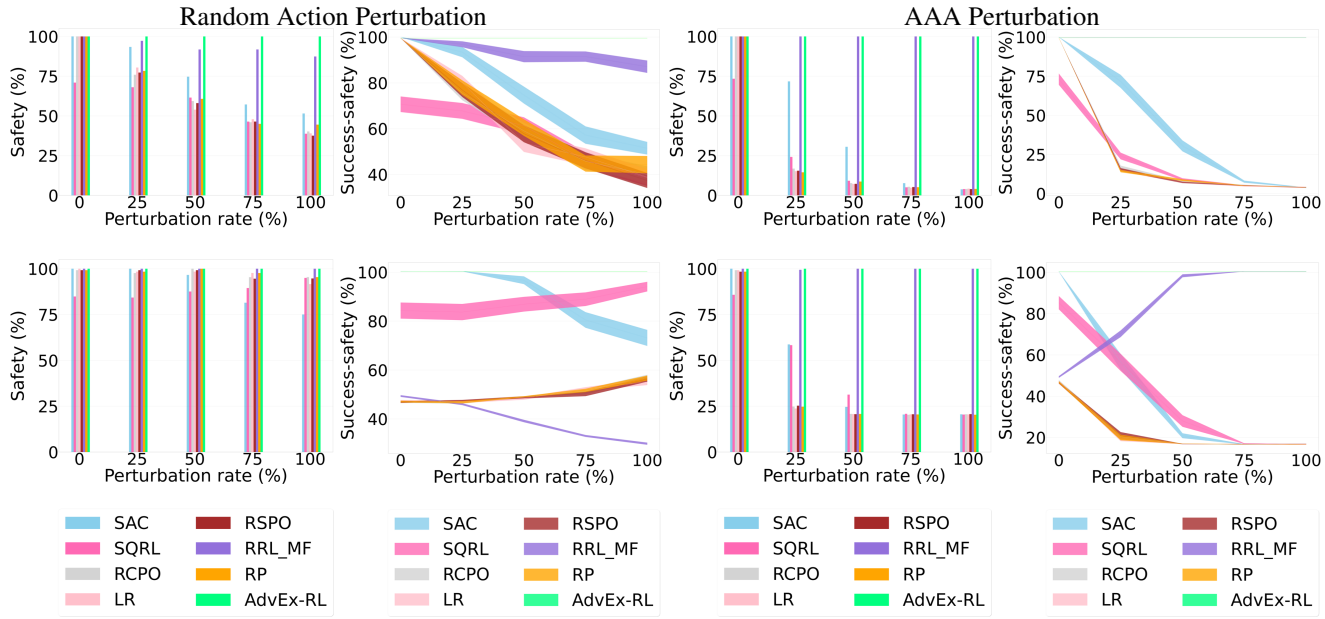

Figure 2: Safety(%) and success-safety(%) performance of the baselines and AdvEx-RL under the influence of various rates of external action perturbations in the Maze (top row) and Navigation 2 (bottom row) environments suggest that baseline methods are not robust, and their performance decreases when the attack rate increases. During this analysis, the dynamics of the environments were kept the same during training and testing. (See Appendix H for results on Navigation 1).

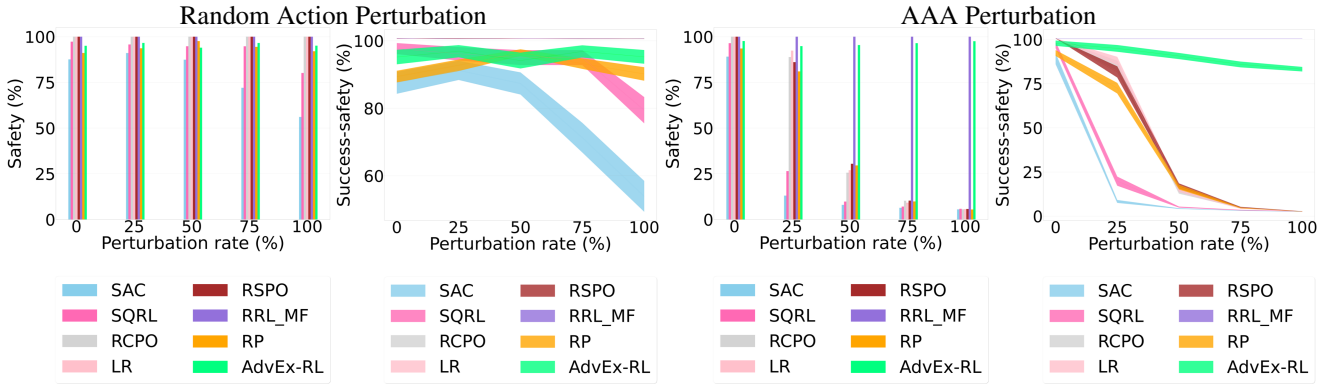

Figure 3: The robustness performance of AdvEx-RL and baselines on the Navigation 1 test environment with 10 times more variation in dynamics than its training environment while being exposed to external perturbations. The results show that AdvEx-RL is more robust to variations in the environment dynamics than other baselines. (Please refer to Appendix K for the detailed experiment)

against perturbations and demonstrates its generalizability to unseen scenarios. (Please refer to Appendix I, to see results in SafetyGym environments. Also the switching between task policy and safety policy is provided in Appendix J.)

Ablation analysis: AdvEx-RL, has three components that contribute to its safety performance i.e (1) the task policy, (2) the safety policy, and (3) the shield. To evaluate the contribution of the safety policy to the overall performance, we conducted a thorough ablation study , the details of which can be found in Appendix L.

Deadlock analysis: We analyzed the presence of deadlock in the baselines while subjected to different rates of AAA perturbation (ranging from 0% to 100%). Using a 20-step look-ahead for deadlock cycle detection across 100,000 test episodes, we found no deadlock cycles in the baselines and AdvEx-RL in all three environments. (See code supplements for deadlock experiment details).

Interpretability Analysis: Since safety-assurance approaches, in general, compromise system performance, we must ensure that human practitioners and users trust them, lest they ignore them and negate their effectiveness. Trustworthiness regarding how safe an agent is depends on how transparent its behavior is to the end users. For the joint task-safety optimization techniques [Srinivasan *et al.*, 2020;

Tessler *et al.*, 2018; Liu *et al.*, 2022], it is not possible to pinpoint what influences the agent's behavior, task, or safety objective. AdvEx-RL, however, uses two separate policies for the task and safety and is, therefore, capable of explaining what influences its behavior at each time step. To analyze the interpretability of AdvEx-RL, we extend the explainable RL method CAPS [McCalmon *et al.*, 2022] into safety-CAPS to explain the impact of safety violations on the agent's behavior and to provide a directed graph with an explanation of the agent's safety policy by extending CAPS with two criteria; risk estimation and the episode length in time steps. We use the safety-CAPS graphs to answer the question of "why the agent is taking certain actions at certain states?". We then add more details to the CAPS graph about how, when, and why the safety policy and safety shield are triggered during safety violations. Episode length (TS) measures the time steps the agent takes to accomplish its task. Perturbing the agent's policy during safety violations could result in the agent taking a longer time to reach its goal state. Hence, we use this metric to answer the question of "why the agent is not taking action a' instead of a at state s_c?" by attaching the timesteps to each abstract state of the CAPS graphs. The risk estimation (RE), on the other hand, measures how likely the agent will fail its task when it takes action a' instead of following its policy and taking a at a critical state s_c due to a safety violation. To estimate the risk, we build on the risk estimation approach proposed in [Uesato *et al.*, 2018] for uncovering failures in RL agents by sampling the agent in states where its safety is violated. We use a neural network with two fully-connected layers of 64 neurons each, and train it to predict RE from each state s_t. An example of the safety-CAPS graph for Navigation 2 is displayed in Fig.4.

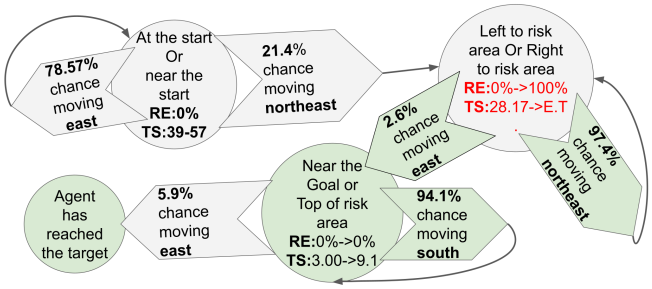

Figure 4: An example of the graphs generated by safety-CAPS for Navigation 2.

We conducted a user study using the Navigation 2 environment and presented 41 Amazon Mechanical Turk (AMT) workers with the safety-CAPS graphs and asked them 8 questions (See Appendix M for the study and the details of the questions) about their interpretability of AdvEX-RL policies. A summary of the accuracy of the users' answers to each question is shown in Figure.5.

In questions 1,2,5, we asked the users to identify optimal actions the agent will take at a certain state with its task policy (Q1), task policy under attack (Q2), and task policy under attack but with AdvEx-RL's safety policy (Q3). The participants demonstrated a good understanding of the environment, with an accuracy rate above 80% in those questions. Notably, the accuracy rate of Q5 is above 90%, which indicates that the users can understand the purpose and impact of the safety policy on the agent's behavior. This clearly shows the interpretability of AdvEx-RL for end-users regardless of their RL background.

Q3 and Q6 are true or false questions where we asked the users to identify if the agent will terminate in a dangerous state under an attack without (Q3) or with (Q6) the safety policy. Roughly 70% and 85% of the users correctly answered Q3 and Q6, respectively. The increase in the accuracy of Q6 demonstrates that the safety-CAPS graph with the safety policy can better convey the reason behind the agent's actions. Q4 and Q7 are counterfactual reasoning type questions where we asked the users why the agent terminated in dangerous states under attack (Q4) but successfully avoided the dangerous states using the safety policy (Q7). We think the low accuracy for those questions is due to their difficulty and demand in terms of logic and analytical skills, which can be challenging to non-technical participants who do not have RL background. Lastly, Q8 is a comprehensive measure of the users' understanding of the overall impact of the safety policy on the agent's behavior. Approximately 73% of the users successfully understood how the safety policy protects the agent from choosing unsafe actions. At the same time, we also need to consider the fact that the length of the study and the dependent relationship of questions generally pose challenges to the participants. If they fail to understand the central idea of the study, they tend to perform badly subsequently. All the participants who incorrectly answered Q8 have at least answered 2 or 3 questions incorrectly before. Therefore, we believe our AdvEx-RL is interpretable by non-technical users using the graphs generated by safety-CAPS.

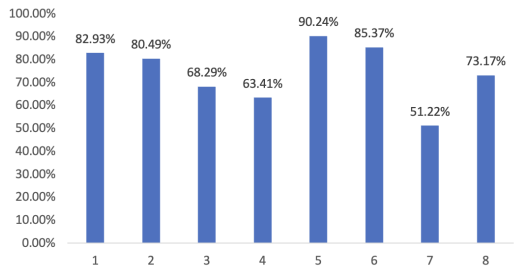

Figure 5: The summary of the accuracy of users' answers to the 8 questions in the user study.

7 Conclusion

In this paper, we introduced an alternative view on safety learning for RL through our task-agnostic safety framework AdvEx-RL. We empirically showed that AdvEx-RL is effective in ensuring safety even in uncertain conditions. Through a user study conducted on 41 non-technical end users, we also demonstrated the transparency of AdvEx-RL by explaining its behavior using safety-CAPS. In future work, we plan to extend this framework to multi-agent settings along with an explainable safety method.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under grant no. 2105007.

References

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

[Bansal *et al.*, 2017] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

[Bastani, 2021] Osbert Bastani. Safe reinforcement learning with nonlinear dynamics via model predictive shielding. In *2021 American Control Conference (ACC)*, pages 3488–3494. IEEE, 2021.

[Chen *et al.*, 2019] Tong Chen, Jiqiang Liu, Yingxiao Xiang, Wenjia Niu, Endong Tong, and Zhen Han. Adversarial attack and defense in reinforcement learning-from ai security view. *Cybersecurity*, 2(1):1–22, 2019.

[ElSayed-Aly *et al.*, 2021] Ingy ElSayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. Safe multi-agent reinforcement learning via shielding. *arXiv preprint arXiv:2101.11196*, 2021.

[Eysenbach and Levine, 2021] Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.

[Fisac *et al.*, 2019] Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8550–8556. IEEE, 2019.

[Geibel and Wysotzki, 2005] Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

[Geibel, 2006] Peter Geibel. Reinforcement learning for mdps with constraints. In *European Conference on Machine Learning*, pages 646–653. Springer, 2006.

[Ghasemipour *et al.*, 2019] Seyed Kamyar Seyed Ghasemipour, Shane Gu, and Richard Zemel. Understanding the relation between maximum-entropy inverse reinforcement learning and behaviour cloning. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019.

[Goo and Niekum, 2022] Wonjoon Goo and Scott Niekum. Know your boundaries: The necessity of explicit behavioral cloning in offline rl. *arXiv preprint arXiv:2206.00695*, 2022.

[Grigorescu *et al.*, 2020] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[Haarnoja *et al.*, 2018a] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[Haarnoja *et al.*, 2018b] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[Hans *et al.*, 2008] Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft. Safe exploration for reinforcement learning. In *ESANN*, pages 143–148. Citeseer, 2008.

[Ke *et al.*, 2021] Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 313–329. Springer, 2021.

[Kim *et al.*, 2020] Youngmin Kim, Richard Allmendinger, and Manuel López-Ibáñez. Safe learning and optimization techniques: Towards a survey of the state of the art. In *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, pages 123–139. Springer, 2020.

[Kober *et al.*, 2013] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[Lee *et al.*, 2020] Xian Yeow Lee, Sambit Ghadai, Kai Liang Tan, Chinmay Hegde, and Soumik Sarkar. Spatiotemporally constrained action space attacks on deep reinforcement learning agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4577–4584, 2020.

[Levine *et al.*, 2018] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018.

[Levine, 2018] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[Liu *et al.*, 2022] Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, pages 13644–13668. PMLR, 2022.

[McCalmon *et al.*, 2022] Joe McCalmon, Thai Le, Sarra Alqahtani, and Dongwon Lee. Caps: Comprehensible abstract policy summaries for explaining reinforce-

ment learning agents. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 889–897, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

[Mihatsch and Neuneier, 2002] Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2):267–290, 2002.

[Puterman, 2014] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

[Srinivasan *et al.*, 2020] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020.

[Tessler *et al.*, 2018] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

[Tessler *et al.*, 2019] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

[Thananjeyan *et al.*, 2020] Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *IEEE Robotics and Automation Letters*, 5(2):3612–3619, 2020.

[Thananjeyan *et al.*, 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 and Automation Letters*, 6(3):4915–4922, 2021.

[Uesato *et al.*, 2018] Jonathan Uesato, Ananya Kumar, Csaba Szepesvari, Tom Erez, Avraham Ruderman, Keith Anderson, Krishmamurthy, Dvijotham, Nicolas Heess, and Pushmeet Kohli. Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures, 2018.

[Ye *et al.*, 2022] Zhaohui Ye, Yanjie Li, Ronghao Guo, Jianqi Gao, and Wen Fu. Multi-agent pathfinding with communication reinforcement learning and deadlock detection. In *Intelligent Robotics and Applications: 15th International Conference, ICIRA 2022, Harbin, China, August 1–3, 2022, Proceedings, Part I*, page 493–504, Berlin, Heidelberg, 2022. Springer-Verlag.