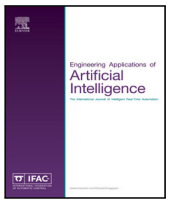
Engineering Applications of Artificial Intelligence 144 (2025) 110055

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Towards robust shielded reinforcement learning through adaptive constraints and exploration: The fear field framework

Haritz Odriozola-Olalde [a,b,*], Maider Zamalloa [a], Nestor Arana-Arexolaleiba [b], Jon Perez-Cerrolaza [a]

[a] Ikerlan Technology Research Centre, P.° J.M. Arizmendiarrieta 2, Arrasate-Mondragon, 20500, Basque Country, Spain
[b] Mondragon Unibertsitatea, Loramendi Kalea 4, Arrasate-Mondragon, 20500, Basque Country, Spain

ARTICLE INFO

Keywords:
Shielded reinforcement learning
Safety constraints
Fear Field
Curriculum learning
Adaptive exploration
Robustness

ABSTRACT

Machine Learning (ML) techniques, including Reinforcement Learning (RL), demonstrate potential as decision-making controllers. However, enhancing the robustness required for real-world deployment remains imperative. Within the realm of Safe RL, Shielded RL emerges as a solution, employing shields to block actions leading to unsafe states and offering safe alternatives through known policies. Yet, many Shielded RL methods rely on dynamic environment models, which may inaccurately predict future states, compromising controller robustness. We introduce the Fear Field framework to mitigate this issue for discrete Markov Decision Process-based (MDP) shields with strictly connected unsafe state spaces and fully observable states, which adjusts safe operation constraints based on disparities between model predictions and actual environmental dynamics. We employ parallel learning and Curriculum Learning (CL) strategies to mitigate lengthy training times in high state-space size environments. Additionally, an adaptive exploration algorithm enhances convergence rates amidst significant environmental dynamic shifts. In our case study, integrating CL and the adaptive exploration algorithm with the Fear Field framework reduces unsafe state occurrences by two orders of magnitude while enhancing convergence time following sudden environmental changes. The Fear Field framework significantly reduces unsafe states in the Frozen Lake Gridworld environment at low computational expense when model predictions deviate from reality, with negligible costs otherwise.

1. Introduction

Autonomous system controllers are in front of a new era due to the decision-making techniques based on ML. Among the variety of techniques in Machine Learning and linked to the optimal control theory (Brockman et al., 2016; Carleo et al., 2019), RL is showing promising results in the literature. RL utilises the exploration–exploitation paradigm to facilitate an agent's learning about its environment (Sutton and Barto, 2018; Zhu and Zhao, 2021). Each action the agent takes yields a reward, serving as a learning opportunity. Consequently, the agent refines its policy throughout the learning process.

However, RL techniques still lack safe operation guarantees. The unsafe states reached during both training and execution could lead the cyber–physical system to a catastrophic state, where its integrity or human lives could be compromised (Perez-Cerrolaza et al., 2023). In recent years, various authors have proposed many solutions to reduce the number of unsafe states that have been reached. Among them, Shielded Reinforcement Learning (Alshiekh et al., 2018) is found. Shielding consists of analysing the action proposed by the agent, blocking it if it leads to an unsafe state and offering a safe action. Most of those methods use an environment's dynamic model to predict the transition of the environment due to the action taken. These methods show correct behaviour if the dynamic model is similar to the real environment. However, reaching previously unseen states and/or dealing with outdated models could lead the agent to incorrectly identify actions by which the environment could transit to an unsafe state. The model could learn the new characteristics of the environment, thus adapting the shield to the new environment. But during the time taken on this process, the previously existing safety guarantees may be lost (Odriozola-Olalde et al., 2023b).

One of the critical aspects of a robust controller is the capacity to perform correctly when the environment's dynamic changes over time, i.e. in time-variant environments. A representative example of a time-variant environment is a rocket ship. As the rocket gains altitude, the fuel is burned, so its dynamic properties change as it loses mass. Thus,

* Corresponding author at: Ikerlan Technology Research Centre, P.° J.M. Arizmendiarrieta 2, Arrasate-Mondragon, 20500, Basque Country, Spain.
E-mail address: hodriozola@ikerlan.es (H. Odriozola-Olalde).

https://doi.org/10.1016/j.engappai.2025.110055
Received 9 April 2024; Received in revised form 2 December 2024; Accepted 8 January 2025
Available online 21 January 2025

if a controller is not robust enough, it could perform well in the rocket's initial stages but lead the system to unstable situations above a certain altitude. Many authors have addressed this problem in different control and decision-making problems, such as in Iterative Learning Control (ILC) (Tao et al., 2023; Wang et al., 2023d), Slide mode control (Wang et al., 2023b,a) or Model Predictive Control (MPC) (Arcos-Legarda and Gutiérrez, 2023; Chen et al., 2024).

By simulating humans' precautionary behaviour when they feel uncertainty, in this work, we provide a framework to leverage shielded RL method shortcomings in time-variant environments. The Fear Field framework adapts the constraints defined in the problem specifications according to the difference between the model prediction and the real environment transitions. Once the environment model is updated and matches the real environment dynamic, the shield reduces the number of reached unsafe states. The Fear Field framework is no longer needed until the subsequent mismatch of the model and reality. Since an MDP-based Shielded RL technique is used, the Fear Field framework is defined considering that state and action spaces are discrete, value-based learning agents (e.g. Q-Learning) are used, and the unsafe state space is strictly connected.

The experimentation environment consists of a state space of 100×100 states. Compared to the experiments made in previous work (Odriozola-Olalde et al., 2023a), the state space is ten times larger in both dimensions, so the learning convergence time has been increased significantly. With this increase in the state space, we tried to prove the effectiveness of the proposed solution in an environment with a higher resemblance to a real industrial application. The learning algorithm used in this work is the tabular Q-Learning. Parallel agents with shared memory and CL-based easy-to-hard learning processes have been included to minimise the training convergence time.

On the other hand, the CL process is also proposed to improve learning time and allow the agent to learn multiple paths to the goal. The robot starting point is chosen randomly to a defined Manhattan distance from the goal. Initially, this distance must have a low value so the starting state is close to the goal state. Gradually, the distance increases, so the previously learned experience is used in posterior learning processes. The distance is increased until the starting state reaches the theoretical starting point. This way, the learning process is progressive; thus, learning time is improved, and the robot explores most of the state space.

The Fear Field framework and CL combination still show convergence issues. Therefore, an adaptive exploration algorithm trained using CL is proposed to be integrated into the policy, which identifies whenever the robot gets stuck in a particular area and allows the agent to increase the exploration rate to find a new path to the goal.

Thus, the contribution of this work can be summarised as follows:

- Fear Field, a framework to compensate for the lack of safety guarantees of Shielded RL solutions on time-variant environments, regarding reliability, integrity and availability, is defined for multi-constraint and discrete n-dimension state space Shielded value-based RL problems with strictly connected unsafe state spaces.
- A parallelisation of agents that share knowledge and a CL strategy is integrated into the Shielded RL algorithm for discrete n-dimensional problems to improve the agent's learning convergence time in time-variant environments.
- An adaptive exploration algorithm is implemented to increase the exploration rate when needed, improving the convergence rate in time-variant environments.
- A novel discrete experimentation environment with a medium-sized state space and time-variant dynamics based on OpenAI Gym Frozen Lake is developed to validate the Fear Field framework behaviour. The results are compared against a non-Shielded RL agent and a Shielded RL agent.

This paper is structured as follows: First, related works to our contribution are summarised in Section 2. Afterwards, a theoretical background is given in Section 3. The problem statement is resumed in Section 4, identifying the challenges of time-variant environments regarding fulfilling safe constraints. In order to reduce constraint violation of Shielded RL in time-variant environments, the Fear Field framework solution is proposed in Section 5. The hypotheses we will cover in this work are presented in Section 6. The experimentation methodology and the metrics used to evaluate the Fear Field framework are presented in Section 7. Consecutively, in Section 8, the results obtained are published. The discussion of the results is developed in Section 9. Finally, in Section 10, conclusions are summarised, and future research lines are presented.

2. Related work

Firstly introduced by Alshiekh et al. (2018), Shielded RL has become a promising teacher-advice-based (Garcıa and Fernández, 2015) framework for RL problems with safe specifications. Most of the Shielded RL approaches do not consider that the controller may be working with a time-variant environment (Wang et al., 2024; Reed et al., 2024; Banerjee et al., 2024; Goodall and Belardinelli, 2024), which increases the complexity of the robustness-related problem. In the subsequent section, the state-of-the-art research of Shielded RL in time-variant environments is analysed, and the existing gaps are identified.

2.1. Time-variant environment

Shielding methods commonly use an environment dynamic model to predict the evolution of the environment for a sequence of actions. If the environment's dynamic changes, the model could become outdated and not predict correctly. Thus, safety guarantees could be lost (Zhu et al., 2019; Bastani and Li, 2021; Pranger et al., 2021; Odriozola-Olalde et al., 2023b). This phenomenon induces the controller to lose robustness. Thus, the scientific community began to study how to reduce the impact caused by an outdated model of the environment on the safety guarantees, as the model is used to synthesise the shield RL-based controller.

In this sense, Zhu et al. (2019) proposed a counterexample-guided inductive synthesis framework to search for a more explicit deterministic algorithm replicating the neural policy's behaviour, which is better for verification. This programme is used as a shield during runtime. After modifying the environment's dynamic behaviour, they observed that using a previously synthesised shield, the number of unsafe states reached is noticeably higher compared to the baseline environment. They observed that synthesising a new shield based on the previous shield requires less time than training a new policy. The main disadvantage of this approach is that safety guarantees are lost until the new shield is synthesised, which, in their experiments, this process can take from 239 s up to 581 s.

Nazmy et al. (2022) performed spacecraft manoeuvre tasks without any constraint violation in simulation after changing the environment's dynamic, specifically deploying a previously trained policy for Earth orbit parameters on a Moon orbital parameters environment. Expressing spacecraft position and velocity on canonical coordinates normalises the trajectory in a two-body regime. However, the orbit parameters used for the Moon (the new environment) are identical to the Earth orbit (original environment) ones; hence, the controller's robustness analysis must be done for planets with significantly different orbit parameters.

Xiong et al. (2022) proposed using adversarial attacks on their novel framework HiSaRL to evaluate its robustness. The mentioned attacks have been made on two fronts. First, the neural Lyapunov function's inputs are attacked to analyse the robustness against input perturbations. Second, framework outputs, i.e. the action to be applied,

are attacked to evaluate the robustness against output disturbances. While the Lyapunov function attack frequency and noise amplitude are less than 60% and 4 cm, respectively, no security violations are observed. However, this is not the case with higher values. Besides, safety is guaranteed in time-invariant environments; safety constraint violations might happen in time-variant environments.

For safe decision-making of urban rail transit autonomous operation, Zhao (2023) proposes the SSA-DRL framework, consisting of four modules: a Deep Reinforcement Learning-based (DRL) module for the agent leaner, a post-shield module to check the action's safety, a searching tree-based module to generate a safe action if the action is blocked by the shield and an additional learner module to study the self-protection ability. During the experimentation, the transferability of the SSA-DRL-based policy for non-learned railroad sections was studied, and outstanding performance was shown. However, the dynamic behaviour of the train is not modified; thus, it could be interesting to see how the SSA-DRL framework performs in these conditions.

Senthilvelan et al. (2022), and Senthilvelan et al. (2023) identified that the approaches based on a single static shield on continuous time-variant environments are not enough to guarantee a safe operation. To overcome this issue, they present the MAPE-K framework, a self-adapting hierarchical shielding method that changes the safe constraints when the environment suffers changes. However, they propose to relax the safety constraints when the environment dynamic changes, which could be unassumable in industrial applications.

Another case where the environment's conditions may change is when the policy trained on simulation is deployed on the lab prototype or, later on, the real cyber–physical system. Hsu et al. (2023) use a shield architecture, Value-Based Shielding, to improve a previous PAC-Bayes-based control. Two policies are trained for this task: the Performance and Backup policies. The Performance policy is trained to maximise the cumulative return while ignoring safety constraints. By contrast, the Backup policy considers the safety constraints, solving the Safety Bellman Equation based on Hamilton–Jacobi reachability analysis. The Backup policy is only used when the Performance policy induces an unsafe state in the environment. Even if the work done by Hsu et al. shows prominent results, such as training on simulation and laboratory prototypes and ending in a real cyber–physical system, it is observed that the safe operation constraint violation rate is still high. Also, it is observed that the proposed methodology is highly sensitive to hyperparameter values; thus, it requires a high effort to tune those hyperparameters properly.

Odriozola-Olalde et al. (2023a) introduced the Fear Field framework, where using the shield RL framework is capable of adapting the safe constraints when the model is outdated due to a time-variant environment. However, the Fear Field framework induces an increment in convergence time after the environment changes its dynamic, possibly related to the agent showing more conservative behaviour and taking fewer risks.

Similar to Odriozola-Olalde et al. (2023a), Bethell et al. (2024) propose the ADVICE post-shielding technique that leverages a contrastive autoencoder (CA) model to distinguish safe and unsafe features, evaluating if the action proposed by the agent must be blocked by a parameter K that parameterise the risk aversion level of the shield. ADVICE manage to reduce safety violations by 50% when the policy trained in a specific environment is deployed in a similar but not the same environment. As ADVICE's drawback, it can be mentioned that initially, it requires some time to train the CA, inducing the possibility of the shield not blocking unsafe actions.

2.2. Conclusions

In Table 1, the literature shield RL methods applied in time-variant environments are analysed and classified. Almost all of them use an abstraction of the environment (model) to predict the next state the environment will transit. Thus, once a significant change happens in the environment, and until the abstraction is updated, previous safe operation guarantees are lost. To the best of our knowledge, the work done by Odriozola-Olalde et al. (2023a) is the only one that considers this issue in the Shielded RL domain and introduces a mitigation mechanism to overcome the shield's issues with time-variant environments.

Table 1
Shielded RL methods analysis in environment abstraction (model) usage, experimentation environment state-space size, if the time-variant environment is considered and if a feedback reward is given to the agent if the shield blocks the action.

Method	Model	State-space size	Time-variant env.	Mitigation mechanism
Zhu et al. (2019)	✓	Small	✓	✗
Nazmy et al. (2022)	✓	Large	✓	✗
Pranger et al. (2021)	✓	Medium	✓	✗
Zhao (2023)	✓	Small	✗	✗
Xiong et al. (2022)	✓	–	✗	✗
Senthilvelan et al. (2022, 2023)	✓	Small	✓	✗
Hsu et al. (2023)	✓	Large	✓	✗
Odriozola-Olalde et al. (2023a)	✓	Small	✓	✓
Bethell et al. (2024)	✗	Large	✓	✓

Regarding the environment's state-space size, only Hsu et al. (2023) consider a state-space size proximal to a real application.

In conclusion, further research is needed on Shielded RL with time-variant methods, mainly focusing on high state-space size environments (closer to real applications) and managing how to overcome the safety guarantees lost due to an outdated model. Also, further research is needed to evaluate whether the feedback reward is beneficial and to establish criteria indicating what type of problems can benefit from it.

3. Preliminaries

The current section will present a theoretical background to improve the following sections' comprehensiveness. Analytical definitions will be given later to particularise the problem analysed in this work.

3.1. Safety specification

Linear Temporal Logic (LTL) formulation is commonly used as specifications of a reactive system (Alshiekh et al., 2018; Jansen et al., 2020; Harris and Schaub, 2020; ElSayed-Aly et al., 2021; Junges et al., 2021; Giacobbe et al., 2021; He et al., 2022; Carr et al., 2022; Den Hengst et al., 2022; Nazmy et al., 2022; Waga et al., 2022; Nikou et al., 2022; Könighofer et al., 2023). This way, safety properties expressed with proposition logic can be extended to temporal operators (Den Hengst et al., 2022).

Definition 1. Let it be a **reactive system** with a finite set of Boolean input $AP_I = i_1, \dots, i_m$ and output $AP_O = o_1, \dots, o_n$ atomic propositions. The input alphabet $\Sigma_I = 2^{AP_I}$ and the output alphabet $\Sigma_O = 2^{AP_O}$ conform the alphabet $\Sigma = \Sigma_I \times \Sigma_O$. Words over the alphabet Σ are defined as traces $\bar{\sigma}$.

A reactive system's LTL safety specifications φ_s can be expressed as a safety deterministic finite automaton (SDFA) named φ_a.

Definition 2. A **Safety Deterministic Finite Automaton (SDFA)** is defined by the tuple $\varphi_a = (\mathcal{Q}, q_0, \Sigma, \delta, \mathcal{F})$; where $\mathcal{Q}$ is the finite set of states, q_0 is the initial state, $\delta : \mathcal{Q} \times \Sigma_I \to \mathcal{Q}$ is the transition function and $\mathcal{F} \subseteq \mathcal{Q}$ is the subset of safe states.

Remark 1. For the posterior definition of the Fear Field framework, it is considered that the subset of unsafe states $\mathcal{U}$, where $\mathcal{F} + \mathcal{U} = \mathcal{Q}$, in the SDFA is strictly connected.

Definition 3. A finite or infinite sequence of states $\bar{q} = q_0, q_1, \ldots \in Q^\infty$ induced by a trace $\bar{\sigma} = \sigma_0, \sigma_1, \ldots \in \Sigma^\infty$ such that $q_{i+1} = \delta(q_i, \sigma_i)$, is defined as a **run** (Known as episode in RL field). If all the states visited by a run $\bar{q}$ are only safe states, i.e. $q_i \in \mathcal{F}, \forall i \in \mathbb{N}$; the trace $\bar{\sigma}$ satisfies safety specifications φ_s and the SDFA φ_a.

3.2. Safety game

Suppose a safety abstraction of the environment is known for the SDFA, and safety specifications φ_s are given. In that case, it is possible to generate a reactive system that consistently generates an output that fulfils φ_s (Den Hengst et al., 2022). This process is known as *reactive system synthesis*. One of the strategies used for reactive synthesis is a *safety game*, where the problem is formulated as a two-player alternating game between an agent and an environment, which in this context is considered an adversarial environment.

Definition 4. A **safety game** is expressed as a tuple $\mathcal{G} : \langle \mathbb{G}, g_0, \Sigma, \delta, \mathcal{F} \rangle$, where $\mathbb{G}$ is the finite set of game states, g_0 is the initial game state, $\Sigma = \Sigma_I \times \Sigma_O$ is the alphabet, $\delta : \mathbb{G} \times \Sigma_I \times \Sigma_O \to \mathbb{G}$ is the transition function and $\mathcal{F} \subseteq \mathbb{G}$ is the set of safe game states.

In a particular state $g \in \mathbb{G}$ of the game, the environment chooses first a word $\sigma_I \in \Sigma_I$ and following the agent chooses a word $\sigma_O \in \Sigma_O$, making the game to transit to state $g' = \delta(g, \sigma_I, \sigma_O)$. A finite sequence of game states $\bar{g} = g_0, g_1, \ldots$ obtained on a safety game is called a *play*. A play is *won* iff all the game states visited are safe $\forall g_i \in \bar{g}, g_i \in \mathcal{F}$.

Definition 5. A function $\rho : \mathbb{G} \times \Sigma_I \to \Sigma_O$ where all plays that can be constructed are won by the agent, i.e. all plays are safe, is called a **winning memoryless strategy**.

3.3. Safe reinforcement learning

An agent interacts with its environment through trial and error in RL. The agent receives a reward that evaluates how well it did taking an action a_t in state s_t and transitioning the environment to state s_{t+1}. The learning process is based on the agent must try to maximise the expected reward (cumulative reward) obtained, thus the optimal policy $\pi*$ is found:

$$\pi^* \leftarrow \arg\max_{\pi \in \Pi} \mathbb{E}^{\pi}_{\mu}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}] \tag{1}$$

where Π is the set of policies, μ is the initial state distribution and γ is the discount factor.

Due to the exploration nature of RL, the agent could reach *unsafe states* $s_u \in S_u$ during the learning process. These unsafe states could become no-backwards states, where the integrity of the cyber–physical system controlled or of the humans involved could be in danger. *Safe Reinforcement Learning* is defined as the process of learning policies which maximise the expected reward in problems where system performance is non-trivial and ensure compliance with safe-operation specifications in both the learning process and execution (Garcıa and Fernández, 2015).

The modifications applied to RL algorithms to improve the fulfilment of the safe-operation specifications mentioned above can be divided into two fundamental trends (Garcıa and Fernández, 2015). The first consists of modifying the optimisation criteria (Slack et al., 2022; Zhao et al., 2022; Li et al., 2022; Flet-Berliac and Basu, 2022; Godbout et al., 2022; Jin et al., 2022; Gu et al., 2022). The second, in turn, consists of modifying the behaviour in terms of exploration (Turchetta et al., 2020; Zhu et al., 2019; Bastani and Li, 2021; Nazmy et al., 2022; Pranger et al., 2021; Xiong et al., 2022; Ro et al., 2022; Senthilvelan et al., 2022, 2023; Hsu et al., 2023; Odriozola-Olalde et al., 2023a), where two ways of proceeding can be observed: incorporating external knowledge and parameterising risk.

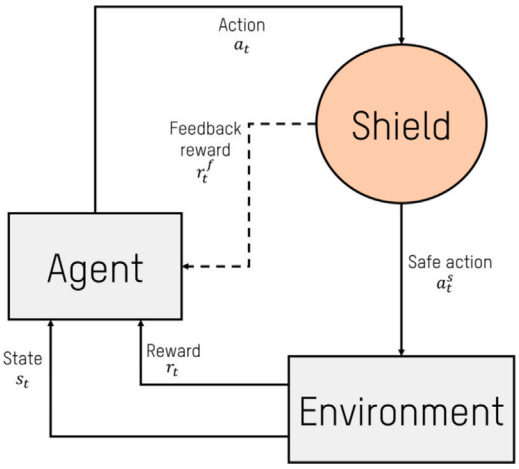


Fig. 1. Shielded reinforcement learning scheme.

Garcıa and Fernández (2015) identified approaches based on *teacher advise* in methods that modify the exploration behaviour by giving external knowledge. This teacher may be a human or a controller, but being an expert in the task is unnecessary. The teacher can provide the knowledge after the learner asks for it or by self-decision.

3.4. Shielded reinforcement learning

One of the teacher advice-based approaches (Garcıa and Fernández, 2015) for safe RL is Shielded Reinforcement Learning (Alshiekh et al., 2018; Yang et al., 2024; Gillet et al., 2024), where a shield intervenes when it predicts that the action proposed by the agent will transit the environment to an unsafe state. Thus, shielded RL is a reactive method, i.e. it only corrects the agent's proposed action when it foresees that the action will lead the environment to violate the safety specifications. That is why shielded RL fulfils the minimum interference premise, intervening only when necessary.

The shield is synthesised through the safety specifications φ_s, i.e. safety constraints, and an abstraction of the environment's dynamic $\mathcal{M}(\vec{s}_t^{\,pred} | \vec{s}_t, a_t)$, i.e. the environment model (Alshiekh et al., 2018) for a fully observable state problem. With the environment model and the safety specifications, a safety game is played between the agent and the environment, obtaining a safe set of plays. With that set of plays, a shield is constructed. Each timestep, the shield monitors the action a_t proposed by the agent for the state s_t. Leveraging the environment model, the shield can estimate the state s_{t+1} for which the environment will transit. If the expected transition state s_{t+1} is unsafe concerning the safety specifications φ_s, the shield will block the action a_t proposed by the agent and will offer a safe action a_t^s following a predefined safe policy π^s. It has been noted that this safe policy is usually predefined using severe constraints. Otherwise, if the action proposed by the agent leads the environment to a safe state, the shield will not interfere. The schematic of the Shielded RL algorithm is shown in Fig. 1.

In those cases where the shield blocks the action proposed by the agent, a *feedback reward* r_t^f may be given to the agent to bias it and consequently reduce the shield intervention rate in future. The safe policy that offers a safe action when the action proposed by the agent is blocked has to be predefined by the designer. It is recommendable to be very restrictive and conservative since it has to offer safe actions even when the environment's dynamic has changed significantly. A common approach for this type of safe policy, sometimes called recovery policy, is to completely stop the robot (Thumm and Althoff, 2022) or turn its actuators off and deploy safety measures, e.g. parachute in an Unmanned Aerial Vehicle (UAV) (Lazarus et al., 2020), to bring it to a safe state. However, these approaches resemble a safety envelope

recovery policy (Perez-Cerrolaza et al., 2023) rather than the one the shield should adopt. The shield's recovery policy should strive to maintain the agent's intended objective while incorporating additional safety measures.

3.5. Curriculum learning

In order to obtain better, faster and safer results, CL was developed as a methodology that optimises the agent learning process (the order of getting knowledge) (Narvekar et al., 2020; Gupta et al., 2021; Turchetta et al., 2020). A curriculum is a structured sequence of tasks represented by a sorted list, denoted as $\mathcal{T}$. Initially, the agent learns simpler and easier tasks, gradually increasing the difficulty. This approach aims to facilitate the learning process for more complex tasks (Gupta et al., 2021).

A set of tasks $\mathcal{T}$ is defined, where $m_i = (S_i, A_i, P_i, R_i)$ is each task of this set $m_i \in \mathcal{T}$, being $i \in \mathbb{N}$ the index of each task. Let it be $\mathcal{D}^{\mathcal{T}}$ the set of all possible transitions from tasks carried in $\mathcal{T}$, where:

$$\mathcal{D}^{\mathcal{T}} = \{(s, a, r, s') | \exists m_i \in \mathcal{T} \text{ s.t. } s \in S_i,\ a \in A_i,\ s' \sim p_i(\cdot|s, a), \\ r \leftarrow r_i(s, a, s')\} \tag{2}$$

Definition 6. A **Curriculum** $C = (\mathcal{V}, \mathcal{E}, g, \mathcal{T})$ is a directed acyclic graph, being $\mathcal{V}$ the set of vertices, $\mathcal{E} \subseteq \{(x, y) | (x, y) \in \mathcal{V} \times \mathcal{V} \wedge x \neq y\}$ the directed edges set and $g : \mathcal{V} \rightarrow \mathcal{P}(\mathcal{D}^{\mathcal{T}})$ the power set of $\mathcal{D}^{\mathcal{T}}$.

Remark 2. A typical case where the learning process is not developed through any curriculum-learning-based method is, in fact, a single-task CL case, which consists only of the target task m_t.

3.6. Goal-based environment

The environment with which the agent interacts can be classified according to the reward function. Sometimes, no prior knowledge of the environment is known, so shaping the reward function may be complex. In those situations, a possible solution is to define the environment as a goal-based environment:

Definition 7. Given an environment $E : \langle M, AP, L_I, L_O, \varphi_s, s_g \rangle$ where M is an MDP, $AP : AP_I \cup AP_O$ are the atomic propositions, L_I and L_O are atomic propositions' labels and s_g is the goal state; is considered as **goal-based environment** if the reward function takes the next form:

$$R(s, a, s') = \begin{cases} 1 & s_g = \delta(s, a) \\ 0 & otherwise \end{cases} \tag{3}$$

4. Problem statement

As mentioned in Section 3.4, on Shielded Reinforcement Learning, an abstraction of the environment (i.e. a dynamic model) and safety constraints are used to synthesise the shield. Thus, an inaccurate model or safety constraints could induce an inappropriate shield synthesis. As the shield's main objective is to predict if the action given by the agent will make the environment transit to an unsafe state, a correct synthesis of the shield must be assured; this way, the shield can continue identifying the unsafe actions correctly.

A change in the environment's dynamic leads to an inaccurate abstraction of the environment and could entail that the initial safe operation constraints are inappropriate for the new situation. This situation could lead to the shield incorrectly classifying the actions proposed by the agent and letting the environment transit to an unsafe state (Fig. 2) (Odriozola-Olalde et al., 2023a).

Consequently, a new shield will be needed to be synthesised. The shielding synthesise process will take some time because the abstraction of the environment needs to be updated. Even if the abstraction of the environment is expressed in an analytical form or as a Neural Network (NN), a particular dataset that carries the dynamic properties of the new environment will be needed.

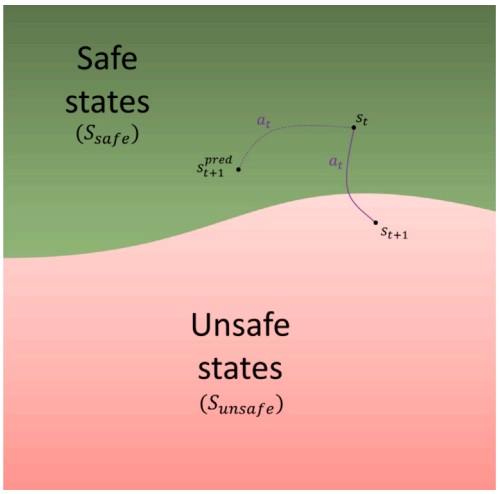


Fig. 2. Example of how changing the environment dynamic can affect decision-making. The model predicts that taking action a_t in the state s_t will lead the environment to a safe state s_{t+1}^{pred}, so the action a_t is not blocked. However, this action leads the environment to an unsafe state s_{t+1} (Odriozola-Olalde et al., 2023b).

During the period between the environment's significant dynamic change and the time the model is updated to the new scenario, Shielded Reinforcement Learning cannot provide the previous safety guarantees. Once the model is updated to the new scenario and correctly predicts the environment's new behaviour, safety guarantees will be assured again.

Therefore, the previously mentioned period is particularly interesting since a safety mechanism that mitigates the safety guarantees lost due to an outdated environment's dynamic model must be studied and developed to reduce the number of unsafe states reached.

When the initial training process on high-size state space and goal-based environments is carried out using Q-Learning as the learning algorithm, the Q-Value spreading from the goal state to the whole grid may take several training episodes. This issue increases with sparse reward functions (e.g. GridWorlds such as OpenAI Gym Frozen Lake) (Brockman et al., 2016). Therefore, this situation could induce a high computational cost or even the learning process to be unfeasible.

5. Fear field

As mentioned previously, a model of the dynamic of the environment $\mathcal{M}(\vec{s}_t^{pred} | \vec{s}_t, a_t)$ is given so the shield can be synthesised with some safety specification φ_s. Significant changes in the environment's dynamic can cause the model M associated with the Shielded Reinforcement Learning algorithm to become outdated, losing the assured safety guarantees until the model is adapted to the new scenario. The Fear Field framework's main objective is to reduce the number of unsafe states reached while the environment's model is outdated. However, it has to be mentioned that the Fear Field framework relies on some assumptions, such as discrete state and action spaces, value-based RL algorithms (such as Q-Learning), fully observable states, and the strict connectivity of the unsafe states space; that may be a limitation for some applications.

Motivational example: While driving a car on the road, if a sudden rainstorm starts, human behaviour is to drive more carefully due to the fear of an accident. Therefore, we reduce the speed, increase the distance to the front car, or avoid overtaking. Also, this conservative behaviour is adapted to our uncertainty about how the vehicle will

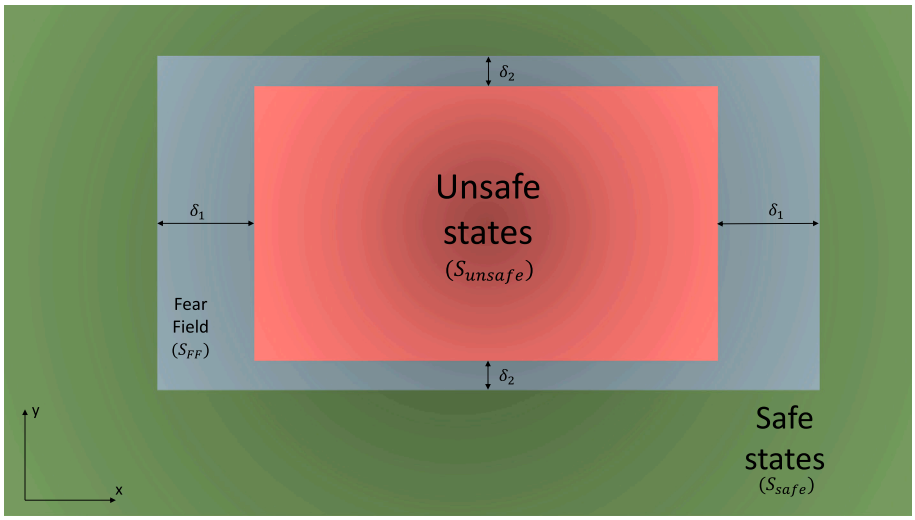


Fig. 3. Example of Fear Field deployment on a bidimensional (x,y) state-space. Note that δ_1 and δ_2 values differ since each dimension's prediction error and/or the safety factor α_i differ.

perform. With a drizzle, our conservativeness level is low, while with a sudden hail, we can even entirely stop the car. This behaviour is linked not only to environmental conditions but also to the car's conditions, such as brake and tyre wear or the vehicle's additional weight. Although the traffic constraints allow us to, e.g. drive at the road's speed limit, keep the recommended distance regarding the car in front, or overtake when necessary, when environmental conditions change, we do not know exactly how the car will behave. Hence, transferring this biological fear/conservativeness behaviour to an ML-based decision-making controller could be effective for scenarios not previously experienced.

As mentioned in the motivational example, when our confidence or knowledge about the environment is reduced, humans' natural reaction is to behave more cautiously until more knowledge of the environment is obtained. The Fear Field framework proposes a similar approach, adapting the safety constraints so the agent will behave more conservatively to trade performance for safety. Once it is identified that the model does not predict correctly and, therefore, needs to be updated, it adapts the constraints to be more restrictive than initially planned. It is established that the model needs to be updated when the Euclidean distance between the predicted state and the reached state is higher than a threshold $\vec{\lambda}$:

$$\left|\vec{s}_t^{pred} - \vec{s}_t\right| \geq \vec{\lambda} \tag{4}$$

As it is practically impossible for the model to be perfectly accurate, mainly when a complex non-linear dynamic is represented, a threshold $\vec{\lambda}$ has been used to filter those minor differences regarding the Euclidean distance that are produced by tolerable stochasticity inherent to the environment. As can be observed, the predicted and reached states and the lambda parameter are vectors. As state space is n-dimensional, the Euclidean distance is checked element-wise so different threshold values can be established for each state-space dimension.

The Fear Field subspace is defined by shrinking the safe state space in a $\vec{\delta}_t$ distance value in the required dimensions. The $\vec{\delta}_t$ distance components are proportional to the previously calculated Euclidean distances multiplied by a safety factor $\vec{\alpha}$:

$$\vec{\delta}_t = \begin{cases} \vec{\alpha} \cdot \left|\vec{s}_t^{pred} - \vec{s}_t\right|, \forall \alpha_i \geq 1 & \left|\vec{s}_t^{pred} - \vec{s}_t\right| \geq \vec{\lambda} \\ 0 & otherwise \end{cases} \tag{5}$$

As the safety value α_i can be tuned for each state space dimension, different values of δ_i can be obtained. Therefore, the safe state space could be shrunk with different values (Fig. 3).

The Fear Field framework aims to identify actions that can be categorised as safe by the shield but lead the environment to an unsafe state. Now, the predicted next state s_{t+1}^{pred} will not be identified by the shield as a safe state but rather a state that belongs to the Fear Field space. As the Fear Field states are proximal to the unsafe states space, and the Fear Field framework identifies that the model is predicting incorrectly, actions that lead the environment to states that belong to the Fear Field will be blocked.

Therefore, if the shield predicted state s_{t+1}^{pred} is an unsafe state or a state within the Fear Field space, the action will be blocked. This way, the shield will offer, following a predefined safe policy, an action that will lead the environment to a state s_{t+1}^{safe} that belongs to the new safe state space (obtained after shrinking the previous one) (Fig. 4a). In our case, the safe policy consists of taking the safe action with the highest Q-value.

While computing the Fear Field space, according to values of the width $\vec{\delta}$, a state s_k can belong to the Fear Field subspace according to δ_i but not according to δ_j (Fig. 4b), where $i, j, \ldots$ indicate each dimension of the state-space. In such cases, the recommended procedure is to take the safer approach and consider it part of the Fear Field subspace. As state space dimensions rise, establishing the required width all along the Fear Field subspace will become more complex, so a generalised approach has to be defined. Also, the width to be applied depends on the angle between each dimension δ value and the bordering hyperplane of the unsafe/safe state spaces.

Let it be a state that belongs to the Fear Field and is located on the border with the unsafe state space s_{border}, with a normal vector $\vec{u}$ to the Fear Field subspace (Fig. 4b). Then, for each s_{border} that conforms the border subspace, the required $\delta^{s_{border}}$ is stated as:

$$\delta^{s_{border}} = \max_n(\delta_i \cdot \vec{i} \cdot \vec{u}, \delta_j \cdot \vec{j} \cdot \vec{u}, \ldots) \tag{6}$$

where n is the number of state space dimensions, $\{\delta_i, \delta_j, \ldots\}$ are the width values in each dimension, and $\left\{\vec{i}, \vec{j}, \ldots\right\}$ are the unitary vectors of each dimension.

The width $\delta^{s_{border}}$ is applied perpendicular to the border, in the direction of $\vec{u}$. This way, the Fear Field subspace is defined as:

$$s \in S_{FF} \Leftrightarrow \exists s_{border} : \left|s - s_{border}\right| \leq \delta^{s_{border}} \tag{7}$$

As the Fear Field width is computed each timestep, it may vary from step to step. The model's prediction could be inaccurate, but in a certain step, be close to reality. Therefore, an increasing width value $\vec{\delta}$ will be applied instantaneously, allowing the shield to identify unsafe actions

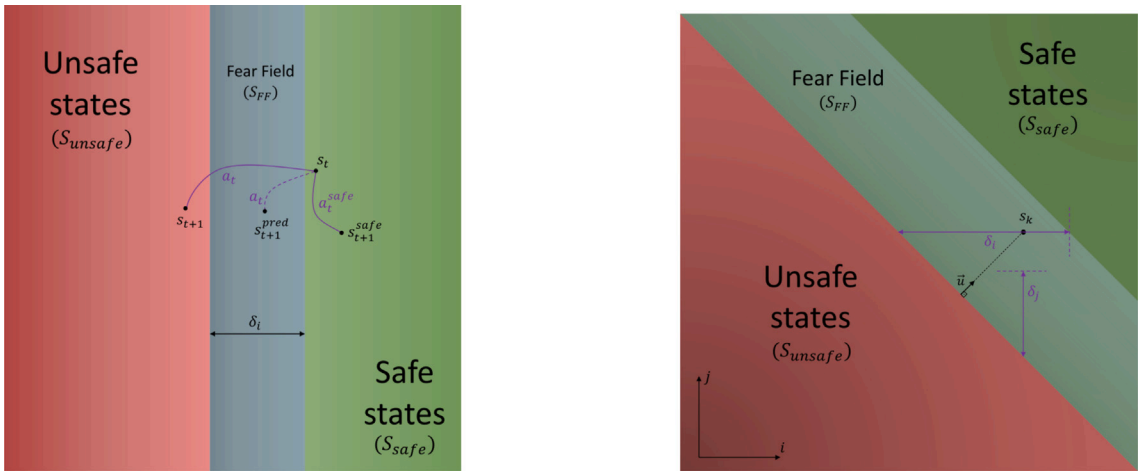


Fig. 4. **(a)** Fear Field framework applied in one of the state space dimensions. The outdated model predicts that the action a_t taken in state s_t will lead the environment to the Fear Field state s_{t+1}^{pred}. So if that action was not blocked, due to that the dynamic has changed significantly, it will transit to the unsafe state s_{t+1}. However, as the shield identifies the Fear Field states as states to be avoided, it will block the action a_t and propose, according to a safe policy, a safe action a_t^{safe} that will lead the environment to a safe state s_{t+1}^{safe} (Odriozola-Olalde et al., 2023a). **(b)** The state s_k belongs to the Fear Field subspace according to the width δ_i but not according to the width δ_j. The safer approach is considered; therefore, δ_i is selected to compute the Fear Field subspace.

as soon as possible. However, its value will be decreased only and only if, in all continuous numbers of n_{Steps} steps, $\vec{\delta}$ is reduced by $\vec{\delta}_{red}$ value or if its value is null (As happens when the model is being updated). The $\vec{\delta}_{red}$ value is defined as:

$$\vec{\delta}_{red} = \max_{n_{Steps}} \vec{\delta} \tag{8}$$

Remark 3. Each element of $\vec{\delta}$ is increased or reduced independently, e.g., if the model is refined in i dimension but not in j dimension, δ_i could be reduced or become null after n_{Steps} while δ_j keeps its value.

5.1. Fear field integration into the shielded RL

As high values of $\vec{\delta}_t$ can make the problem unfeasible, we propose a new Fear Field states treatment: Only the actions that the shield predicts will lead the environment to an unsafe state ($\vec{s}_{t+1}^{pred} \in S_{unsafe}$) or into a state within the Fear Field with the transition vector pointing to an unsafe state ($\vec{s}_{t+1}^{pred} \in S_{FF} \wedge \lambda = 0$) will be blocked:

$$a_t \in A_{unsafe} \Leftrightarrow (\vec{s}_{t+1}^{pred} \in S_{unsafe}) \vee (\vec{s}_{t+1}^{pred} \in S_{FF} \wedge \lambda = 0) \tag{9}$$

where λ analyses if the predicted state transition vector is pointing to any unsafe state inside a $S^{\vec{\delta}_t}$ subspace defined as:

$$\begin{aligned} &S^{\vec{\delta}_t} = \left\{ \vec{s}_i \in S \mid i = 1, \dots, m \right\} \\ &s.t. \left| s_{i,j} - s_{t+1,j}^{pred} \right| \leq \delta_{t,j} \ \wedge \ \hat{u}_t \cdot (\vec{s}_i - \vec{s}_{t+1}^{pred}) \geq 0 \\ &\quad j = 1, \dots, n \end{aligned} \tag{10}$$

being m the number of states within $S^{\vec{\delta}_t}$, j a dimension of S, n the number of dimensions of S, $\hat{u}_t$ the normalised vector of the predicted transition vector $\vec{u}_t = \vec{s}_{t+1}^{pred} - \vec{s}_t$.

The $\left| s_{i,j} - s_{t+1,j}^{pred} \right| \leq \delta_{t,j}$ condition ensures that all the $s_i \in S^{\vec{\delta}_t}$ elements are within a dimension-dependant $\delta_{t,j}$ radius. On the other hand, the condition $\hat{u}_t \cdot (\vec{s}_i - \vec{s}_{t+1}^{pred}) \geq 0$ ensures that all s_i states are in the semiplane defined by the point $\vec{s}_{t+1}^{pred}$ and by the normal vector $\hat{u}_t$. In Fig. 5 can be observed that for a bi-dimensional S with $\alpha = [1, 1]$ the $S^{\vec{\delta}_t}$ subspace forms a semicircle shape.

Computing for all $\vec{s}_i^{unsafe} = \left\{ \vec{s}_i \in S^{\vec{\delta}_t} \cap S_{unsafe} \right\}$ states inside the $S^{\vec{\delta}_t}$ hyperspace, we can define the λ parameter as:

$$\lambda = \prod_{\vec{s}_i^{unsafe} \in S^{\vec{\delta}_t}} (\vec{s}_i^{unsafe} - \vec{s}_{t+1}^{pred}) \times \hat{u}_t \tag{11}$$

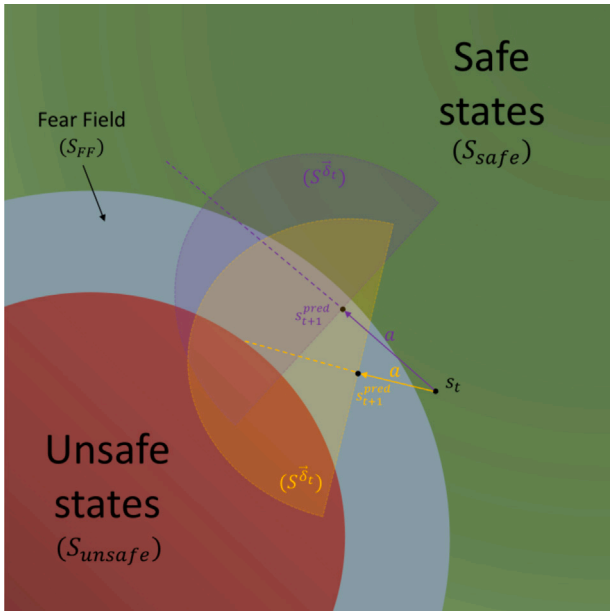


Fig. 5. Example of allowed (purple) and blocked (orange) actions that will transit the environment to a Fear Field state. In the allowed action, the predicted transition vector does not collide with any unsafe state within the $S^{\vec{\delta}_t}$ hyperspace; however, the orange action is facing directly to an unsafe state within the $S^{\vec{\delta}_t}$ hyperspace.

If there is any unsafe state $\vec{s}_i^{unsafe}$ in the direction of the predicted transition vector and inside the $S^{\vec{\delta}_t}$ hyperspace, λ value will be null, blocking the proposed action.

However, if the predicted state is not within the unsafe state space $\vec{s}_{t+1}^{pred} \in S_{unsafe}$ and the transition vector is not pointing to any unsafe state $\lambda \neq 0$ within the subspace $S^{\vec{\delta}_t}$, the action is allowed. Notice that by defining the constraints fulfilment this way, the cyber–physical system is allowed to reach a Fear Field state iff its predicted transition vector ensures that it is not facing an unsafe state (Fig. 5).

According to the enhancement to the Fear Field framework presented in this work, the algorithm used for the Fear Field framework is shown in Algorithm 1.

Algorithm 1 Fear Field algorithm

Given (Environment Dynamic Model $\mathcal{M}(\vec{s}_t^{pred}|\vec{s}_t, a_t)$, safety specifications φ_s, $n_{Dataset}$, n_{Steps})
Predict $\vec{s}_t^{pred}$ using $\mathcal{M}(\vec{s}_t^{pred}|\vec{s}_t, a_t)$
if $\left|\vec{s}_t^{pred} - \vec{s}_t\right| \geq \vec{\lambda}$ **then**
    Calculate $\vec{\delta}_t(s_t)$ value
    Generate S_{FF} s.t. $\vec{\delta}_t(s_t)$ and φ_s
    if $t = t_{LastTrain} + n_{Dataset}$ **then**
        Update $\mathcal{M}(\vec{s}_t^{pred}|\vec{s}_t, a_t)$
        $t_{LastTrain} = t$
    end if
else if $(\vec{s}_{t-n_{Steps}-1}, ..., \vec{s}_{t-1}) = (\vec{s}^{pred}_{t-n_{Steps}-1}, ..., \vec{s}^{pred}_{t-1})$ **then**
    Remove S_{FF}
end if
Check the safety of all actions
Take highest $Q(\vec{s}_t, a)$ safe action
Apply the action a_t to the environment
Save last n_{Steps} steps data in the last visited states FIFO buffer memory

5.2. Adaptive exploration in runtime

In scenarios where the shield suggests a corrective action, the agent may transition to an inadequately explored state. Consequently, the need to enhance the exploration rate arises. Specifically, in the context of the Q-learning approach, the epsilon (ϵ) value plays a crucial role in determining this exploration rate. To address this situation, we propose an adaptive exploration algorithm (refer to Algorithm 2), which dynamically adjusts the ϵ value.

To identify these situations, first, we store the past k_b number of states reached in the last visited states FIFO buffer memory. Then, we calculate the mode of the buffer states, obtaining the state value s_m and how many times (n_{rep}) the environment reached it in the past k_b number of steps. If in the previous k_b steps, the environment reaches a state in more than $n_{rep} \geq n_{loop}$ times, then we identify the agent is in an area vaguely explored or has entered into a loop. Once these situations are identified, we propose increasing the exploration rate ϵ value to ϵ_{adap} until the agent can find a path to the goal.

Algorithm 2 Adaptive exploration algorithm

Given (k_b, n_{loop}, ϵ_{adap})
Initialize the Buffer of k_b elements
while runtime **do**
    Introduce the reached state s_t into the last visited states FIFO buffer memory
    Calculate the mode of Buffer and obtain s_m and n_{rep}
    if $n_{rep} \geq n_{loop}$ **then**
        $\epsilon = \epsilon_{adap}$
    else
        $\epsilon = 0$
    end if
end while

In Odriozola-Olalde et al. (2023a) work, it was observed that in runtime, the convergence time after a significant environment dynamic change in the Fear Field framework was increased compared to the Shielded Tabular Q-Learning case. Thus, integrating the adaptive exploration algorithm can achieve better convergence time than the Shielded Tabular Q-Learning case.

6. Hypothesis

With the implementation of CL and vectorisation to the learning process, as well as the Fear Field framework with Adaptive exploration to both the learning and inference process, we expect an improvement in terms of performance and safety. As mentioned in Section 4, the use of Q-Learning on relatively-high-size state space and goal-based environments with sparse reward function often involves high convergence times; thus, we expect that CL and vectorisation will reduce the required learning time and homogenise the explored states during training **[H1]**. Also, we expect that using CL in the initial learning process will improve the convergence and reduce the number of unsafe states reached when the agent has to adapt to time-variant environments **[H2]**. When the policy learned in the initial training process is deployed in a time-variant environment, we expect that the Fear Field framework will adapt the constraints adequately, so the number of reached unsafe states will be reduced **[H3]**. Lastly, we expect to reduce the convergence time in time-variant environments after a significant change happens in the dynamics, using the Adaptive exploration algorithm **[H4]**.

7. Experiments

A modified version of the OpenAI Gym Frozen Lake (Brockman et al., 2016) environment is used for the experimentation. This environment consists of a reach-avoid problem, which can also be defined as a safety game, where the robot has to reach the goal while avoiding the holes (unsafe states). Once the robot falls into a hole, i.e., reaches an unsafe state, the episode ends. Each time an episode ends due to reaching an unsafe state, the goal or the maximum number of steps of the episode, the robot has to start from the beginning.

High-dimensional and stochastic environments are required to validate the Fear Field framework (Odriozola-Olalde et al., 2023a). The environment used for experimentation is an enhanced version of the Frozen Lake environment with a 100×100 grid size.

The state space size is directly correlated to the training convergence time, i.e. *the curse of dimensionality*, as Q-Learning is used as a learning algorithm. Also, the states that are not close to the optimal solution are visited less often (or maybe never), so if the agent reaches any of those states in inference, it will not be capable of finding the solution.

Regarding the convergence time, Carr et al. (2023) observed that in sparse reward domains, the probability of reaching the goal state with a random policy is improbable; thus, the probability of spreading the goal's Q value to the state space, i.e. learning the problem stated, can take an unassumable number of episodes.

In this work, we integrate CL into the training process, i.e. initially, the agent's starting state is close to the goal and gradually moves away until it reaches the real starting state. In our case, after the n_e number of episodes concludes, a new task $m_i \in \mathcal{T}$ where $i = [1, 200]$ is defined where the starting state of each episode is moved away to a Manhattan distance d_{manh}:

$$d_{manh} = \begin{cases} (Episode/n_e) + 1 & : \ Episode/n_e < 199 \\ 200 & : \ Episode/n_e \geq 199 \end{cases} \tag{12}$$

The Manhattan distance is applied to increase the size of the subspace (Fig. 6), and in case more than one state is to d_{manh} distance from the goal state, randomly, one of them is chosen. This way, the agent learns how to reach the goal from any state of the state space.

Also, our algorithm is vectorised, so multiple learning instances are executed, where agents involved share the knowledge they are obtaining. Knowledge sharing is done by sharing the same tabular Q-Learning values, and after each step, the Q-Values are updated according to the experience of each instance. The main goal of vectorisation is that sharing knowledge can accelerate the learning time and explore

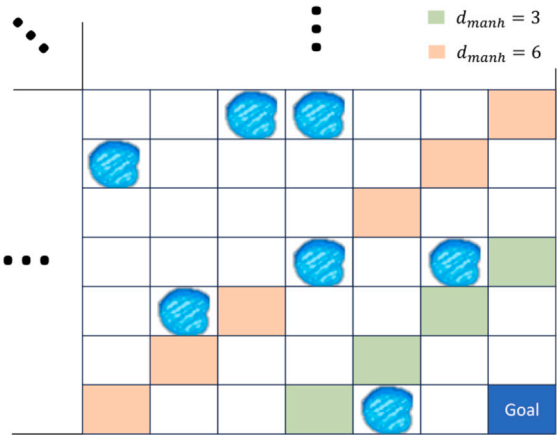


Fig. 6. Example of states that are to $d_{manh} = 3$ and $d_{manh} = 6$ distances from the goal.

the state space more homogeneously. As we use ϵ-greedy tabular Q-Learning as an RL algorithm, each agent's probability of taking a random action is independent.

The simulation of the time-variant phenomenon involves altering environmental characteristics to emulate changes in the weather conditions, as depicted in Fig. 7. These variations include different levels of slipperiness across the environment: non-slippery, slightly slippery, slippery, and severely slippery. In non-slippery conditions, the robot moves only one square per action taken. On slightly slippery, slippery and slippery conditions, the robot moves an additional one, two or three (successively) squares in the same direction as the chosen action.

Experiments are executed by the RL open-source library Skrl (Serrano-Muñoz et al., 2023), which contains various RL algorithms. The shielded RL and CL were included in the Skrl library to enable the experiments to be conducted. Regarding vectorisation, Skrl already integrates vectorised RL algorithms and environments; thus, we only had to introduce vectorisation to the shielded RL algorithm.

All experiments were performed using a workstation equipped with an Intel Core i9-13900K, 64 GB RAM, and an RTX 4090 GPU, running Ubuntu 22.04.3 LTS as OS. The hyperparameter values we used in experimentation are shown in Table 3 in Appendix.

7.1. Metrics

During experimentation, $N_{tests} = 50$ tests are made to obtain average results values, where each episode consists of a maximum of $N_{steps} = 300$ steps. For **H1**, we compare the studied learning approaches in terms of the metric: (1) *A heat map of visited states during the training process* (Eq. (13)). In **H2**, **H3** and **H4**, we compare the different safety approaches with two metrics: (2) *Average cumulative reward per episode* and (3) *Average reached unsafe states percentage* (Eq. (14)).

$$R_{avg} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(\frac{1}{N_{steps}} \sum_{k=1}^{N_{steps}} r_k^i \right) \tag{13}$$

$$P_{unsafe} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(\frac{N_{unsafe}^i}{N_{steps} \cdot N_{episodes}} \right) \tag{14}$$

The *Average cumulative reward per episode* is a widely used metric in Shielded Reinforcement Learning (RL) research (Alshiekh et al., 2018; Kochdumper et al., 2023; Zhang et al., 2019; Jansen et al., 2020; Anderson et al., 2020; Carr et al., 2023; Dey et al., 2023; Wang et al., 2023c). It provides insight into how well an agent performs on a given task.

As we experimented with time-variant environments for Shielded RL methods, we found it interesting to check how often the environment reaches unsafe states. Since in time-invariant environments, the shield reaches a null number of unsafe states. Therefore, regarding the safety metric, we decided to follow the *Total collision number* considered by many authors (Kochdumper et al., 2023; Zhu et al., 2019; Anderson et al., 2020; ElSayed-Aly et al., 2021; He et al., 2022) but we applied minor modification to show a relative view of how many steps of all the runtime processes are unsafe ones.

7.2. RL algorithms

We use the tabular Q-Learning algorithm as a baseline in the experiments performed. To compare against the baseline, we propose three other algorithms: a shielded tabular Q-learning algorithm, the previous shielded tabular Q-Learning including the Fear Field framework, and lastly, the previous shielded tabular Q-Learning + Fear Field framework but with the adaptive exploration algorithm. For **H2**, we compare the previously mentioned four algorithms when the training process has been done with and without a CL algorithm.

8. Results

The following lines show the results obtained in the initial process and during runtime with significant dynamic changes. First, the training phase results are presented, and afterwards, the results are obtained during the trained policy's inference.

8.1. Initial training experimentation (**[H1]**)

The initial training refers to the scenario where the agent has no knowledge and must learn how to solve the problem. Note that the environment does not suffer from any dynamic change during this initial training. The shield is used during the initial learning process, as it shortens the time needed until the policy is trained (Odriozola-Olalde et al., 2023a).

Two approaches have been tested initially: a vectorised learning agent with no CL algorithm where the initial point is always the upper left state and an agent with a CL algorithm where the initial point changes over time (as mentioned in Section 7). The training process does not converge in the case of the agent without CL (Fig. 8 left) because the positive reward (r = 100) is obtained iff the agent reaches the goal state. Therefore, we introduce (only in the agent without CL case) a dense reward function r_{dense} that evaluates the Manhattan distance between the actual state and the goal state and computes the reward as the inverse value of this distance:

$$r_{dense} = 1/d_{manh}(s_t, s_{goal}) \tag{15}$$

As shown in Fig. 8 (middle), introducing a dense reward function induces the agent to learn a possible path to the goal. As can be observed, the path is not very noticeable in the heat map since the agent could not achieve the goal in most of the initial episodes of the training process. As during the training process, an ϵ-greedy algorithm is used, it can be observed that the path to the goal is not totally narrow, so if there is a slight deviation from the path, the agent will be capable of bringing the robot back to the path again.

The case of the agent with the CL algorithm is different from the previous ones. It can be observed (Fig. 8 right) that the agent learns multiple paths all across the state space that lead to the goal state. Thus, depending on the robot's starting state, the agent will take the optimal path to the goal.

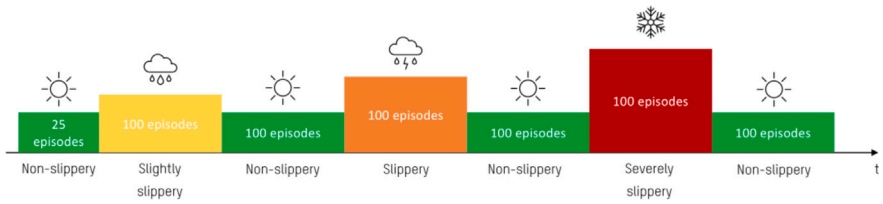


Fig. 7. Environment evolution over time on runtime experimentation.

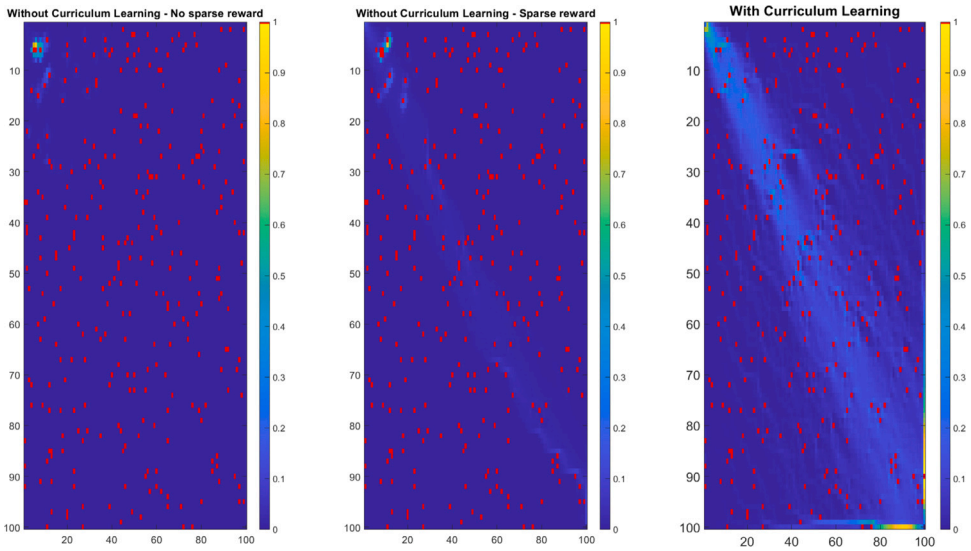


Fig. 8. Heat map of the initial learning process for an agent without CL and no dense reward (left), an agent without CL but dense reward (middle) and an agent with CL (right). Data is normalised, and value 1 (red) is reserved to represent the holes in the map.

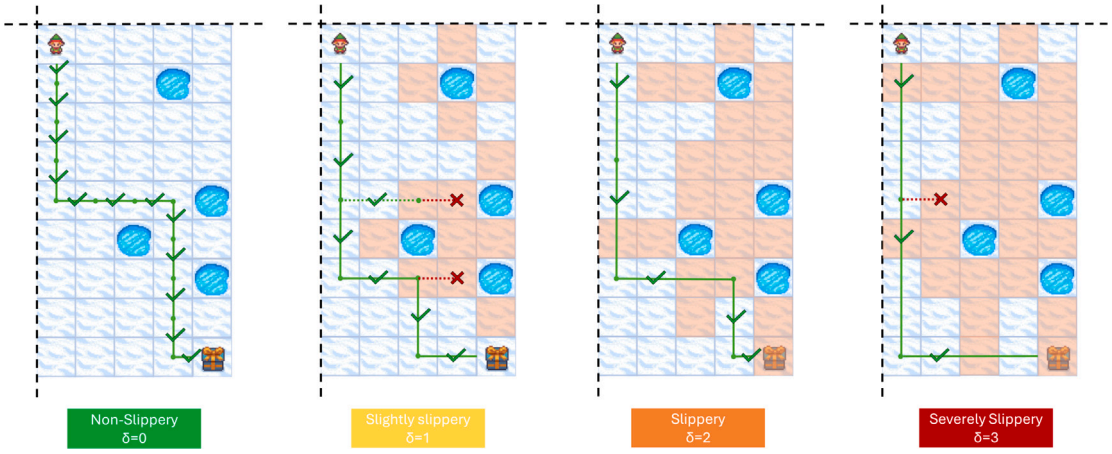


Fig. 9. Agents behaviour (Path taken to the goal) illustrative example in the four conditions (Non-Slippery, Slightly Slippery, Slippery and Severely Slippery) that it has to face in the final section of the Frozen Lake map. The states that belong to the Fear Field are presented as transparent light orange rectangles. A green check indicates that the action has been accepted by the shield and Fear Field framework, while the red cross indicates that the action has been blocked.

*8.2. Adaptation to the environment changes during runtime (**[H2], [H3] and [H4]**)*

During runtime, the agent has to solve the problem learned during the initial training process, but now the environment dynamic is changing significantly over time. This way, the agent must adapt and relearn the stated problem, avoiding violating the safety constraints.

In Fig. 9 it can be observed an illustrative example of the differences in path taking of an agent with Shielded RL and the Fear Field framework. Different domain conditions (Non-Slippery, Slightly Slippery, Slippery and Severely Slippery) are presented, where δ determines the distance (in a number of cells) that the Fear Field covers to avoid the robot's transition to an unsafe state. It can be observed that when the environment's conditions change, the agent tries to stick to the learned path in the Non-Slippery condition. However, it has to adapt to the new situation and find a new path to the goal that satisfies the safety constraints defined by the shield, and the condition of not reaching a Fear Field state iff the intended movement direction is facing an unsafe state. As can be observed in the Severely Slippery situation, when the agent faces a Fear Field state and the intended movement direction is not facing an unsafe state, the action is allowed.

Table 2
Comparative of reached unsafe state percentage per total performed steps using an initially trained policy without CL and with CL. In bold, the best value.

Learning algorithm	W/o CL	W/ CL	$\frac{w/o\ CL}{w/\ CL}$
Q-Learning	0,033825838	**0,027681916**	1,221947105
Shielded Q-Learning	**0,001417385**	0,003035718	0,466902606
Shielded Q-Learning + Fear Field	**1,65895E−05**	0,002573397	0,006446548
Shielded Q-Learning + Fear Field + Adaptive exploration	**0,000411926**	0,00054812	0,751524876

The effect that a policy initially trained with and without CL has over the four studied algorithms (Presented in Section 7.2) can be observed in Fig. 10 where the cumulative reward obtained is plotted. On the other hand, in Table 2 the percentage of reached unsafe states over all the steps is presented. The continuous line corresponds to the results obtained when applying CL during training, while the discontinuous line corresponds to the case where CL is not used. In the x-axis, the experiment episode is presented, where we indicate the periods of the environment evolution defined in Fig. 7. In the y-axis, the mean cumulative reward of all tests is presented, where we indicate with a light red zone bounded in $[-75, -105]$ approximately, the cumulative reward of an agent that reaches an unsafe state should achieve. If the cumulative reward is below a value of -75, we can confirm that the agent systematically reached an unsafe state. On the other hand, a cumulative reward of 100 means that the agent reaches the goal following an optimal path.

Regarding the Tabular Q-Learning case, it can be observed (Fig. 10 top left) that CL-based policy shows worse convergence than Non-CL-based policy for the *slightly slippery* episodes but better one in the *slippery* episodes. In "severely slippery" episodes, both CL-based and non-CL-based policies cannot converge. In Table 2, it can be observed that the CL-based policy reaches a slightly smaller number of unsafe states.

In the Shielded Q-Learning case, the CL-based policy performs worse than the Non-CL-based homonymous policy (Fig. 10 top right) since the Non-CL-based policy is able to converge at least in the *slightly slippery* episodes. Also, the CL-based policy reaches twice the percentage of unsafe states.

When the Fear Field framework is integrated into the Shielded Q-Learning algorithm (Fig. 10 bottom left), the Non-CL-based policy is able to converge also in the *slippery* episodes; however, during the *non-slippery* episodes alternated between slippery episodes, the agent is not able to converge. On the other hand, the CL-based policy does not converge during the slippery episodes but converges in all the episodes when the environment is *non-slippery*. Regarding the number of unsafe states reached, the Non-CL-based policy reaches approximately two orders of magnitude fewer unsafe states.

Lastly, with the Shielded Q-Learning + Fear Field + Adaptive exploration algorithm, it can be observed (Fig. 10 bottom right) that the CL-based policy performs better than the Non-CL-based policy, managing to converge faster. The number of unsafe states the environment reaches is slightly higher in the CL-based policy case than in the Non-CL-policy case. It has been mentioned that all the unsafe states reached in both cases happen when the environment changes from *non-slippery* to *severely slippery*, where the unsafe action taken is due to the adaptive exploration algorithm in 100% and 73% in CL-based and Non-CL-based cases respectively.

Fig. 11 shows the mean cumulative reward per episode for the four algorithms tested. As observed in our previous work (Odriozola-Olalde et al., 2023a), the Tabular Q-Learning, the Shielded Q-Learning and the Shielded Q-Learning + Fear Field reach unsafe states after a significant change happens in the environment dynamic. When the Fear Field framework is integrated, the number of unsafe states reached (**Q3**) is reduced by two orders of magnitude, especially in the Non-CL-based case. The shielded Q-Learning algorithm can converge after the environment returns to "non-slippery" conditions from some of the three slippery conditions. However, it is unable to converge in any of the slippery conditions.

In relation to the integration of an exploration algorithm, we can observe that the convergence time (**Q4**) is reduced, achieving convergence in all 50 tests and all three slippery conditions. Comparing the number of unsafe states reached by Shielded Q-Learning and Shielded Q-Learning + Fear Field + Adaptive exploration algorithms, the quantified reduction value is 5.54 times lower.

9. Discussion

Observing the results presented in the previous section, we noticed that introducing CL during the training process improves the learning capability of the agent. Two possible causes are identified. First, the CL approach that we proposed increases the exploration rate since, for each task, the robot's starting point is changed randomly. Second, since the knowledge is initially obtained in a proximal area of the goal and gradually increases in size, there is a backpropagation of Q-values linked to reaching the goal. In the case of CL, the agent was not able to learn its task if an additional sparse reward was introduced. Since the map size is increased 100 times, the chances of the robot reaching the goal by taking random action are nearly null.

Introducing the Fear Field framework to a Shielded RL agent reduced by nearly two orders of magnitude the number of unsafe states reached during the experimentation; however, the convergence of the agent is still not guaranteed when faced with a change in the environment's dynamic. Since the robot's behaviour is different in these situations from what it learned, we noticed that it usually enters zones of the state space with a low knowledge of it. With this in mind, we introduced an adaptive exploration algorithm that, once identified that the robot is in one of those low-explored regions, introduces a low random action probability that helps the robot to find an alternative path (usually a suboptimal one) learned through CL. Introducing this adaptive exploration algorithm significantly increases the agent convergence, allowing it to achieve the goal even when the robot faces a highly slippery environment with an additional three-block movement. However, the number of unsafe states reached in the tests is still not null, which we believe may be associated with the model's NN updating and deployment process. Refer to future work.

Regarding the research question **H1**, in this work, it is concluded that introducing a CL approach for the initial learning process improves the training time and also homogenises the number of visited states all over the FrozenLake map, learning multiple paths to the goal. Also, we have to remark that the non-CL-based training process only learned the proposed task if an additional reward function, which implicitly includes a knowledge of the solution (Eq. (15)), is given. Thus, CL-based approximation can improve the training process for problems without information about the solution.

Concerning the research question **H2**, we figured out that the CL approach-based initial learning process is sometimes beneficial in terms of safe operation and convergence. When the initial training process was done with CL, no significant improvement in convergence or safe operation was observed in general. However, in the experiments, it is observed that in the case of the Shielded RL + Fear Field algorithm, integrating CL penalised significantly in terms of the number of unsafe states reached. However, we observed that the Shielded RL + Fear Field + Adaptive exploration algorithm significantly benefits from the integration of CL in the training process in terms of convergence.

Integrating the Fear Field framework into the Shielded RL algorithm reduces the number of unsafe states reached. It is especially noticeable in the Shielded Q-Learning + Fear Field case, which obtains a reached unsafe state percentage of $1.65895 \cdot 10^{-5}\%$. Still, as the counterpart, the convergence is reduced as the agent is unable to converge on the

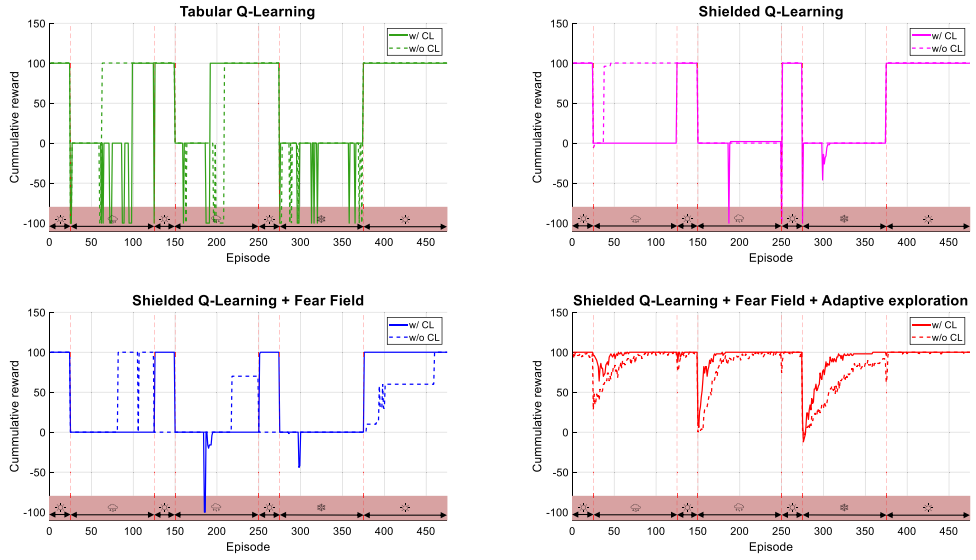


Fig. 10. Comparative of cumulative reward obtained by the robot in runtime for the four algorithms studied when the policy is trained with and without CL.

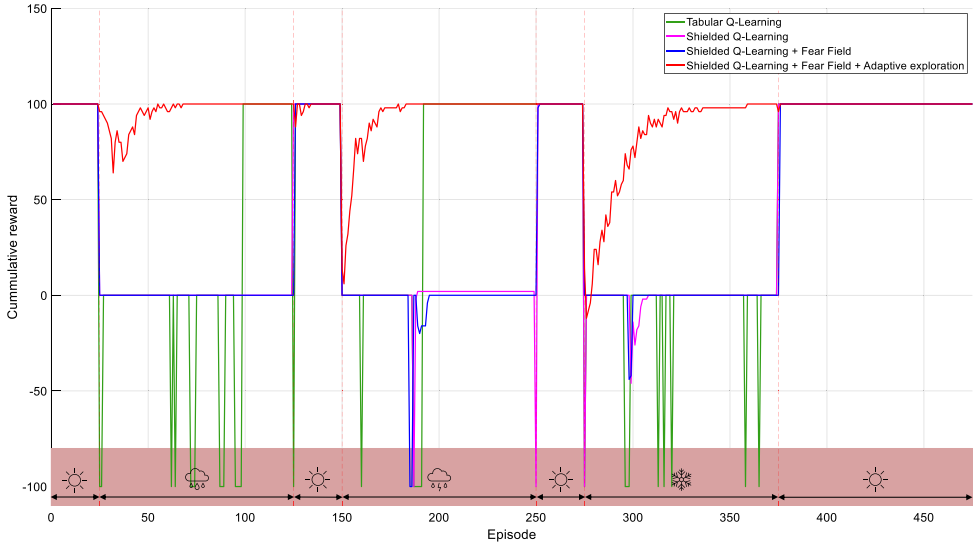


Fig. 11. Comparative of cumulative reward obtained by the robot in runtime for the four algorithms studied with a CL-based trained policy.

severely slippery episodes. Therefore, according to the research question **H3**, we can confirm that the number of the reached unsafe states is reduced compared to the Shielded RL approach with a fear-based constraint adaptation framework, but it influences the convergence time.

In the work of Odriozola-Olalde et al. (2023a), it has been identified that integrating the Fear Field framework decreases the number of reached unsafe states by one order of magnitude approximately with the counterpart of increasing the convergence time. As the Fear Field framework induces a more conservative approach in the agent, the agent may reach a situation where the learned solution is no longer feasible. Thus, the agent must re-learn a new solution or find another previously discovered solution. An adaptive exploration algorithm is introduced to work among a CL-based trained agent to cope with this situation. The results show that in conditions where other approaches cannot converge, the Shielded RL + Fear Field + Adaptive exploration converges and maintains a low number of reached unsafe states. So, regarding the research question **H4**, we can confirm that integrating an adaptive exploration algorithm can significantly improve the algorithm's convergence time while maintaining a similar safe operation behaviour.

As mentioned above, we observed that in the 100% and 73% cases (CL-based and Non-CL-based policy, respectively) where the environment reached an unsafe state, it is due to a random action following the adaptive exploration algorithm. Hence, we found again the paradigm of safety versus performance. The adaptive exploration algorithm hyperparameters, in addition to RL agents and NNs ones, must be further studied to find values that further reduce the number of reached unsafe states without compromising the controller's convergence.

The Fear Field framework has been designed for discrete state and action space problems and fully observable states; it also leverages value-based approaches, such as Q-Values in Q-Learning, to choose the

optimal safe action among the set of safe actions. Therefore, its applicability is limited to discrete or discretised continuous environments and value-based learning agents. In addition, the Fear Field framework has been designed for strictly connected unsafe state spaces, a situation that may not fit with some real problems' conditions. Thus, the Fear Field framework approach could be further researched to address these limitations. The computation of the safe set of actions for each state can be expensive when working in an environment with high action space dimensionality, so this issue must be addressed for future works.

Also, we considered the full state observability capability of the agent, which is hard to happen in reality. This type of problem, addressed in Partially Observable MDP (POMDP) problems, is a hot topic amidst the Safe RL community. So, we believe that it may be interesting to analyse further Fear Field framework enhancements or adaptations to operate in partial observability scenarios.

Despite the mentioned limitations and withdrawals of the Fear Field framework, it fills a gap that, to the best of our knowledge, nobody worked on: ensuring safety when the environment has changed, a new shield must be synthesised, and therefore, the operating system safety is degraded. It has been observed in this work that integrating the Fear Field framework always ensures an improvement in safety terms compared to a classic Shielded RL approach. Additionally, the Fear field framework could also be valuable in solving Sim2Real gap problems (which resemble the problem treated in this work). The Fear Field framework could be a valuable contribution to problems of this nature. Sim2Real gap is also a very active research area in the scientific community, with many research works aiming to leverage the potential of simulators and digital twins for automata training, avoiding the time and resource cost of training agents in real cyber–physical systems.

10. Conclusions and future work

Ensuring robustness, in terms of reliability, integrity and functional safety, during the learning and inference process of an AI-based controller is an open and well-known challenge in the research community (Perez-Cerrolaza et al., 2023). In the robotics domain, approaches such as Shielded Reinforcement Learning propose a model-based safety assurance for MDP problems. Shielded RL, like all data-driven controllers, may not ensure correctness when facing previously unseen environments or when the faced environment evolves over time, as happens in time-variant environments. RL allows us to re-learn the problem stated and, thus, adapt the controller to the new scenario. However, this adaptation process involves an adaptation period. If the environment change causes unexpected behaviour in the agent, safe operation guarantees may be lost until the re-learning process is finished.

This paper presents the Fear Field framework to cope with challenges posed by time-varying environments. Through the Fear Field framework, the agent is encouraged to act more carefully, adapting the safety operation constraints to a more conservative approach. The number of reached unsafe states is reduced by one order of magnitude, but as a counterpart, the convergence rate of the controller is reduced.

To overcome this issue, we propose introducing an adaptive exploration algorithm that identifies when the controller is unable to find a solution after the environment has changed and allows the agent to take a random action to explore and find a new path to the goal. The Adaptive exploration algorithm effect is especially remarkable in the most extreme environment variation (severely slippery), where it is the only RL algorithm capable of converging from all tested ones. However, we observed that this method increases the number of unsafe states reached. However, integrating the Fear Field framework is always beneficial in terms of safety compared to the Shielded RL approach.

As we introduced an environment with a relatively high-size state space, the previous Q-Learning approach did not perform correctly both in the learning and inference processes, as it needed to introduce a new parameter, which implicitly contains the solution to the problem, within the reward function. Therefore, the use of CL is proposed to cover this type of large environment, where the agent begins its learning process close to the goal, and the starting state is brought away from the goal over time. We observed that during the learning process, the agent learned many different paths to the goal, which can switch from one path to another if any deviation brings the robot to a state out of the main path. Generally, the CL-based controller performed similarly to the non-CL-based controller in safety and convergence. Nevertheless, combining CL with an Adaptive exploration algorithm has improved the convergence and maintained the number of unsafe states reached.

In conclusion, our proposed method improves the Shielded RL algorithm both in reached unsafe states (reduction of one order of magnitude) and convergence (Shielded RL is incapable of converging during the three slippery conditions). Also, integrating CL into the learning process has been beneficial in large state space-size environments.

In future work, we aim to validate the Fear Field framework in a real industrial application simulation or target. We also have to focus on a systematic optimisation of the hyperparameters of the RL algorithm, the Fear Field framework and especially the Adaptive exploration algorithm, which we believe affects the convergence time and the constraint violation rate. We identified that the cause of reaching unsafe states, even with the Fear Field framework active, could be related to the model training and deployment process. As we rely only on one model for the predictions, any incidence during the retraining directly impacts the predictions and, thus, the controller's safety.

Continuous state and action space environments are challenging scenarios for Safe RL techniques. Since the Fear Field framework has been proposed for discrete value-based RL algorithms, a study on how to generalise this work contribution to continuous and non-value-based algorithms (e.g. PPO and DDPG) could be a significant contribution to the Safe RL community. A possible path to generalise the Fear Field framework could be leveraging Shielded RL techniques based on Control Barrier Functions (Cao et al., 2023; Zhang et al., 2023) existent in the literature.

The motivational example shows that the Fear Field framework could fit into Autonomous Driving Vehicle (ADV) applications. The ADV is a high-complexity environment where training the AI-based decision-making controller in all the possible operational conditions it may face during its whole life cycle may be unfeasible. Therefore, it may be necessary to train the agent for limited and broad possible scenarios and refine the policy once deployed, leveraging the safety guarantees provided by the Fear Field framework. In the short future, we will leverage the Safety Gymnasium (Ji et al., 2023) benchmark environment with simplified ADV dynamics *racecar* agent to validate the Fear Field framework in a more realistic ADV-related environment.

The Fear Field framework may also have the potential to be integrated into a wide range of other applications, e.g. Traffic Collision Avoidance Systems in Aeronautics or Multi-Robot Systems in automated warehouses. Shielded RL has shown potential for giving formal safety guarantees for AI-based controllers; hence, further solutions (such as the Fear Field framework) need to be investigated and validated to afford the model inaccuracy situation that many model-based runtime safety assurance methods face.

CRediT authorship contribution statement

Haritz Odriozola-Olalde: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Maider Zamalloa:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Nestor Arana-Arexolaleiba:** Writing – review & editing, Supervision, Project administration. **Jon Perez-Cerrolaza:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially developed within the MEDUSA project's framework: "Red tecnológica de ingeniería aplicada al desarrollo de soluciones inteligentes para conducción autónoma centrada en la persona (CER-20231011), Red de Excelencia CERVERA" funded by the "Ministerio de Ciencia, Innovación y Universidades, Gobierno de España" via "Centro para el Desarrollo Tecnológico y la Innovación E.P.E. (CDTI) (Spain)", supported by the European Union's Recovery and Resilence Facility (RRF). This work was partially developed within the Group of Excellence in Intelligent Systems for Industrial Systems (SISI) - Reference IT1676-22. Basque Government - Education Department

Appendix. Hyperparameters

The values of the hyperparameters used during experimentation are shown in Table 3. It contains the agent's training parameters values, the environment's dynamic NN-based model's parameters, the parameters related to the Fear Field framework, the reward function values and the adaptive exploration algorithm parameters.

As the main objective of this work is to improve the safety of the agent when it faces time-variant environments, we do not focus on optimising the hyperparameter values related to the performance of the agent; however, some of the values are tested to reduce the training necessary time. We leave to define a systematic optimisation procedure for future work.

For agent hyperparameter values, we followed the default values defined in SKRL (Serrano-Muñoz et al., 2023) Frozen Lake Q-learning illustrative example, modifying the epsilon greedy algorithm's ϵ and $\epsilon - decay$ values in a range of $\epsilon \in \{0.1, 0.2, \dots, 1\}$ and $\epsilon_{decay} = \left\{-1.2 \times 10^{-5}, -1.2 \times 10^{-4}, -1.2 \times 10^{-3}\right\}$. For the Non-CL case, we found that a high exploration value $\epsilon = 0.6$ with a decay over 5000 episodes $\epsilon_{decay} = -1.2 \times 10^{-4}$ is a good combination. In contrast, for the CL-based case, a slight chance $\epsilon = 0.1$ of taking a random action to explore is beneficial over the whole $\epsilon_{decay} = 0$ learning process.

Regarding the NN training hyperparameters, we started from the default values as we leveraged the Keras library (Chollet et al., 2015). We modified the topology, the buffer and batch size, and the epochs to optimise them for the Fear Field framework's features. For the NN topology, we tried $n_{hidden-layers} = \{1, 2\}$ with $n_{neurons-per-layer} = \{8, 10, 12, 24, 48\}$ and observed that $n_{hidden-layers} = 1$ and $n_{neurons-per-layer} = 12$ provide the best performance per less training required time relation. For buffer and batch size and epochs, we tested $Buffer - size = \{800, 1000, 1200\}$, $Batch - size = \{6, 10, 50, 100\}$ and $Epochs = \{50, 100, 200, 500\}$. For the experimented values, we select the ones that show a good performance while keeping the NN retraining time as low as possible.

For the Fear Field hyperparameters and the reward function, we kept the values obtained by Odriozola-Olalde et al. (2023a) as they performed well.

Lastly, for the adaptive exploration algorithm, we tried $k_b = \{10, 20, 30\}$, $n_{loop} = \{3, 5, 10\}$ and $\epsilon_{adap} = \{0.1, 0.2\}$. We found that the visited states FIFO buffer memory size of the last 20 $\left\{s_{t-19}, s_{t-18}, \dots, s_t\right\}$ states capture reasonably well if the agent is facing unexplored states. A value of $n_{loop} = 3$, which counts if at least the 15% of the states stored in the last visited FIFO buffer memory correspond to a single state, seems enough to identify if the agent gets stuck in the unexplored region. Regarding the adaptable exploration rate, we found that just a low value $\epsilon_{adap} = 0.1$ is enough for the agent to find a path that brings it to a previously explored region.

Table 3
Hyperparameters values used in experimentation.

	Hyperparameter	Value
Agent	ϵ	0.6 | 0.1 (Without/With CL)
	ϵ decay	-1.2×10^{-4} per episode | 0 (Without/With CL)
	γ	0.999
	α	0.4
NN training	Topology	1-12-1
	Buffer size	1000
	Batch size	6
	Epochs	100
	Learning rate	0.001
	Optimiser	Adam
	Activation function	ReLU
Fear field	$n_{Dataset}$	1000
	n_{Steps}	800
	λ	0
Rewards	Step taken	−0.01
	Hit a wall	−1
	Fall in hole	−100
	Reach goal	100
	Action blocked by the shield	−10
	Action blocked by Fear Field	−2
Adaptive exploration	k_b	20
	n_{loop}	3
	ϵ_{adap}	0.1

Data availability

The authors do not have permission to share data.

References

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U., 2018. Safe reinforcement learning via shielding. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32, http://dx.doi.org/10.1609/AAAI.V32I1.11797.

Anderson, G., Verma, A., Dillig, I., Chaudhuri, S., 2020. Neurosymbolic reinforcement learning with formally verified exploration. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems. Vol. 33, Curran Associates, Inc., pp. 6172–6183.

Arcos-Legarda, J., Gutiérrez, Á., 2023. Robust model predictive control based on active disturbance rejection control for a robotic autonomous underwater vehicle. J. Mar. Sci. Eng. 11 (5), http://dx.doi.org/10.3390/jmse11050929.

Banerjee, A., Rahmani, K., Biswas, J., Dillig, I., 2024. Dynamic model predictive shielding for provably safe reinforcement learning. arXiv preprint arXiv:2405.13863.

Bastani, O., Li, S., 2021. Safe reinforcement learning with nonlinear dynamics via model predictive shielding. In: 2021 American Control Conference. ACC, IEEE, pp. 3488–3494. http://dx.doi.org/10.23919/ACC50511.2021.9483182.

Bethell, D., Gerasimou, S., Calinescu, R., Imrie, C., 2024. Safe reinforcement learning in black-box environments via adaptive shielding. arXiv preprint arXiv:2405.18180.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Openai, W.Z., 2016. OpenAI gym. http://dx.doi.org/10.48550/arxiv.1606.01540, arXiv preprint.

Cao, H., Xiong, H., Zeng, W., Jiang, H., Cai, Z., Hu, L., Zhang, L., Lu, W., 2023. Safe reinforcement learning-based motion planning for functional mobile robots suffering uncontrollable mobile robots. IEEE Trans. Intell. Transp. Syst. 1–18. http://dx.doi.org/10.1109/TITS.2023.3330183.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., Zdeborová, L., 2019. Machine learning and the physical sciences. Rev. Modern Phys. 91 (4), 045002. http://dx.doi.org/10.1103/RevModPhys.91.045002.

Carr, S., Jansen, N., Junges, S., Topcu, U., 2022. Safe reinforcement learning via shielding for POMDPs. http://dx.doi.org/10.48550/arxiv.2204.00755, arXiv preprint.

Carr, S., Jansen, N., Junges, S., Topcu, U., 2023. Safe reinforcement learning via shielding under partial observability. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37, pp. 14748–14756. http://dx.doi.org/10.1609/aaai.v37i12.26723.

Chen, S., Preciado, V.M., Morari, M., Matni, N., 2024. Robust model predictive control with polytopic model uncertainty through system level synthesis. Automatica 162, 111431. http://dx.doi.org/10.1016/j.automatica.2023.111431.

Chollet, F., et al., 2015. Keras. https://keras.io.

Den Hengst, F., François-Lavet, V., Hoogendoorn, M., van Harmelen, F., 2022. Planning for potential: efficient safe reinforcement learning. Mach. Learn. 111 (6), 2255–2274. http://dx.doi.org/10.1007/S10994-022-06143-6.

Dey, S., Dasgupta, P., Dey, S., 2023. Safe reinforcement learning through phasic safety-oriented policy optimization. In: Pedroza, G., Huang, X., Chen, X.C., Theodorou, A., Hernández-Orallo, J., Castillo-Effen, M., Mallah, R., McDermid, J.A. (Eds.), Proceedings of the Workshop on Artificial Intelligence Safety 2023 (SafeAI 2023) Co-Located with the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023), Washington DC, USA, February 13-14, 2023. In: CEUR Workshop Proceedings, Vol. 3381, CEUR-WS.org, URL https://ceur-ws.org/Vol-3381/22.pdf.

ElSayed-Aly, I., Bharadwaj, S., Amato, C., Ehlers, R., Topcu, U., Feng, L., 2021. Safe multi-agent reinforcement learning via shielding. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. Vol. 1, International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), pp. 483–491. http://dx.doi.org/10.5555/3463952.3464013.

Flet-Berliac, Y., Basu, D., 2022. SAAC: Safe reinforcement learning as an adversarial game of actor-critics. In: RLDM 2022 - the Multi-Disciplinary Conference on Reinforcement Learning and Decision Making. Providence, United States, URL https://hal.science/hal-03771734.

Garcıa, J., Fernández, F., 2015. A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. 16 (1), 1437–1480, URL http://jmlr.org/papers/v16/garcia15a.html.

Giacobbe, M., Hasanbeig, M., Kroening, D., Wijk, H., 2021. Shielding atari games with bounded prescience. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), pp. 1507–1509. http://dx.doi.org/10.5555/3463952.3464141.

Gillet, S., Marta, D., Akif, M., Leite, I., 2024. Shielding for socially appropriate robot listening behaviors. In: 2024 33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN).

Godbout, M., Heuillet, M., Raparthy, S.C., Bhati, R., Durand, A., 2022. A game-theoretic perspective on risk-sensitive reinforcement learning. In: SafeAI@AAAI. URL https://api.semanticscholar.org/CorpusID:247321889.

Goodall, A.W., Belardinelli, F., 2024. Leveraging approximate model-based shielding for probabilistic safety guarantees in continuous environments. In: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems. pp. 2291–2293. http://dx.doi.org/10.5555/3635637.3663137.

Gu, S., Chen, G., Zhang, L., Hou, J., Hu, Y., Knoll, A., 2022. Constrained reinforcement learning for vehicle motion planning with topological reachability analysis. Robotics 11 (4), http://dx.doi.org/10.3390/robotics11040081.

Gupta, K., Mukherjee, D., Najjaran, H., 2021. Extending the capabilities of reinforcement learning through curriculum: A review of methods and applications. SN Comput. Sci. 3 (1), 1–18. http://dx.doi.org/10.1007/S42979-021-00934-9, 2021 3:1.

Harris, A.T., Schaub, H., 2020. Spacecraft command and control with safety guarantees using shielded deep reinforcement learning. In: AIAA Scitech 2020 Forum. Vol. 1 PartF, American Institute of Aeronautics and Astronautics Inc, AIAA, pp. 386–403. http://dx.doi.org/10.2514/6.2020-0386.

He, C., León, B.G., Belardinelli, F., 2022. Do androids dream of electric fences? Safety-aware reinforcement learning with latent shielding. In: Proceedings of the Workshop on Artificial Intelligence Safety 2022 (SafeAI 2022) Co-Located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI2022). abs/2112.11490 URL https://ceur-ws.org/Vol-3087/paper_50.pdf.

Hsu, K.-C., Ren, A.Z., Nguyen, D.P., Majumdar, A., Fisac, J.F., 2023. Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees. Artificial Intelligence 314, 103811. http://dx.doi.org/10.1016/j.artint.2022.103811.

Jansen, N., Könighofer, B., Junges, S., Serban, A., Bloem, R., 2020. Safe reinforcement learning using probabilistic shields. In: 31st International Conference on Concurrency Theory (CONCUR 2020). Schloss-Dagstuhl-Leibniz Zentrum für Informatik, http://dx.doi.org/10.4230/LIPIcs.CONCUR.2020.3.

Ji, J., Zhang, B., Zhou, J., Pan, X., Huang, W., Sun, R., Geng, Y., Zhong, Y., Dai, J., Yang, Y., 2023. Safety gymnasium: A unified safe reinforcement learning benchmark. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems. Vol. 36, Curran Associates, Inc., pp. 18964–18993.

Jin, P., Tian, J., Zhi, D., Wen, X., Zhang, M., 2022. TRAINIFY: A CEGAR-driven training and verification framework for safe deep reinforcement learning. In: International Conference on Computer Aided Verification. Springer, pp. 193–218. http://dx.doi.org/10.1007/978-3-031-13185-1_10.

Junges, S., Torfah, H., Seshia, S.A., 2021. Runtime monitors for Markov decision processes. In: International Conference on Computer Aided Verification. In: LNCS, Vol. 12760, Springer, pp. 553–576. http://dx.doi.org/10.1007/978-3-030-81688-9_26.

Kochdumper, N., Krasowski, H., Wang, X., Bak, S., Althoff, M., 2023. Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes. IEEE Open J. Control Syst. 2, 79–92. http://dx.doi.org/10.1109/OJCSYS.2023.3256305.

Könighofer, B., Rudolf, J., Palmisano, A., Tappler, M., Bloem, R., 2023. Online shielding for reinforcement learning. Innov. Syst. Softw. Eng. 19 (4), 379–394. http://dx.doi.org/10.1007/s11334-022-00480-4.

Lazarus, C., Lopez, J.G., Kochenderfer, M.J., 2020. Runtime safety assurance using reinforcement learning. In: 2020 AIAA/IEEE 39th Digital Avionics Systems Conference. DASC, IEEE, pp. 1–9. http://dx.doi.org/10.1109/DASC50938.2020.9256446.

Li, G., Yang, Y., Li, S., Qu, X., Lyu, N., Li, S.E., 2022. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. Transp. Res. C 134, 103452. http://dx.doi.org/10.1016/j.trc.2021.103452.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M.E., Stone, P., 2020. Curriculum learning for reinforcement learning domains: A framework and survey. J. Mach. Learn. Res. 21 (181), 1–50, URL http://jmlr.org/papers/v21/20-212.html.

Nazmy, I., Harris, A., Lahijanian, M., Schaub, H., 2022. Shielded deep reinforcement learning for multi-sensor spacecraft imaging. In: 2022 American Control Conference. ACC, IEEE, pp. 1808–1813. http://dx.doi.org/10.23919/ACC53348.2022.9867762.

Nikou, A., Mujumdar, A., Sundararajan, V., Orlic, M., Feljan, A.V., 2022. Safe ran control: A symbolic reinforcement learning approach. In: 2022 IEEE 17th International Conference on Control & Automation. ICCA, IEEE, pp. 332–337. http://dx.doi.org/10.1109/ICCA54724.2022.9831850.

Odriozola-Olalde, H., Arana-Arexolaleiba, N., Zamalloa, M., Perez-Cerrolaza, J., Arozamena-Rodríguez, J., 2023a. Fear field: Adaptive constraints for safe environment transitions in shielded reinforcement learning. In: Proceedings of the IJCAI-23 Joint Workshop on Artificial Intelligence Safety and Safe Reinforcement Learning (AISafety-SafeRL 2023) Co-Located with the 32nd International Joint Conference on Artificial Intelligence(IJCAI2023), Macau, China, August 21-22, 2023. In: CEUR Workshop Proceedings, Vol. 3505, CEUR-WS.org, URL https://ceur-ws.org/Vol-3505/paper_3.pdf.

Odriozola-Olalde, H., Zamalloa, M., Arana-Arexolaleiba, N., 2023b. Shielded reinforcement learning: A review of reactive methods for safe learning. In: 2023 IEEE/SICE International Symposium on System Integration. SII, IEEE, pp. 1–8. http://dx.doi.org/10.1109/SII55687.2023.10039301.

Perez-Cerrolaza, J., Abella, J., Borg, M., Donzella, C., Cerquides, J., Cazorla, F.J., Englund, C., Tauber, M., Nikolakopoulos, G., Flores, J.L., 2023. Artificial intelligence for safety-critical systems in industrial and transportation domains: A survey. ACM Comput. Surv. http://dx.doi.org/10.1145/3626314.

Pranger, S., Könighofer, B., Tappler, M., Deixelberger, M., Jansen, N., Bloem, R., 2021. Adaptive shielding under uncertainty. In: 2021 American Control Conference. ACC, IEEE, pp. 3467–3474. http://dx.doi.org/10.23919/ACC50511.2021.9482889.

Reed, R., Schaub, H., Lahijanian, M., 2024. Shielded deep reinforcement learning for complex spacecraft tasking. arXiv preprint arXiv:2403.05693.

Ro, J.W., Lüttgen, G., Wolter, D., 2022. Reinforcement learning with imperfect safety constraints. In: Pedroza, G., Hernández-Orallo, J., Chen, X.C., Huang, X., Espinoza, H., Castillo-Effen, M., McDermid, J.A., Mallah, R., hÉigeartaigh, S.O. (Eds.), Proceedings of the Workshop on Artificial Intelligence Safety 2022 (SafeAI 2022) Co-Located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI2022), Virtual, February, 2022. In: CEUR Workshop Proceedings, Vol. 3087, CEUR-WS.org, URL https://ceur-ws.org/Vol-3087/paper_38.pdf.

Senthilvelan, P., Li, J., Tei, K., 2022. Safe reinforcement learning through hierarchical shielding with self-adaptive techniques. In: 9th International Conference on Industrial Engineering and Applications. http://dx.doi.org/10.18178/wcse.2022.04.14.

Senthilvelan, P., Li, J., Tei, K., 2023. Similarity-based shield adaptation under dynamic environment. In: 2023 IEEE 3rd International Conference on Software Engineering and Artificial Intelligence. SEAI, IEEE, pp. 33–39. http://dx.doi.org/10.1109/SEAI59139.2023.10217461.

Serrano-Muñoz, A., Arana-Arexolaleiba, N., Chrysostomou, D., Bøgh, S., 2023. Skrl: Modular and flexible library for reinforcement learning. J. Mach. Learn. Res. 24 (254), 1–9, URL http://jmlr.org/papers/v24/23-0112.html.

Slack, D., Chow, Y., Research, G., Dai, B., Wichers, N., 2022. SAFER: Data-efficient and safe reinforcement learning via skill acquisition. http://dx.doi.org/10.48550/arxiv.2202.04849, arXiv preprint arXiv:2202.04849.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

Tao, H., Zheng, J., Wei, J., Paszke, W., Rogers, E., Stojanovic, V., 2023. Repetitive process based indirect-type iterative learning control for batch processes with model uncertainty and input delay. J. Process Control 132, 103112. http://dx.doi.org/10.1016/j.jprocont.2023.103112.

Thumm, J., Althoff, M., 2022. Provably safe deep reinforcement learning for robotic manipulation in human environments. In: 2022 International Conference on Robotics and Automation. ICRA, IEEE, pp. 6344–6350. http://dx.doi.org/10.1109/ICRA46639.2022.9811698.

Turchetta, M., Kolobov, A., Shah, S., Krause, A., Agarwal, A., 2020. Safe reinforcement learning via curriculum induction. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA.

Waga, M., Castellano, E., Pruekprasert, S., Klikovits, S., Takisaka, T., Hasuo, I., 2022. Dynamic shielding for reinforcement learning in black-box environments. In: International Symposium on Automated Technology for Verification and Analysis. Springer, pp. 25–41. http://dx.doi.org/10.48550/arxiv.2207.13446.

Wang, J., Alattas, K.A., Bouteraa, Y., Mofid, O., Mobayen, S., 2023b. Adaptive finite-time backstepping control tracker for quadrotor UAV with model uncertainty and external disturbance. Aerosp. Sci. Technol. 133, 108088. http://dx.doi.org/10.1016/j.ast.2022.108088.

Wang, H., Qin, J., Kan, Z., 2024. Shielded planning guided data-efficient and safe reinforcement learning. IEEE Trans. Neural Netw. Learn. Syst. 1–12. http://dx.doi.org/10.1109/TNNLS.2024.3359031.

Wang, B., Shen, Y., Li, N., Zhang, Y., Gao, Z., 2023a. An adaptive sliding mode fault-tolerant control of a quadrotor unmanned aerial vehicle with actuator faults and model uncertainties. Internat. J. Robust Nonlinear Control 33 (17), 10182–10198. http://dx.doi.org/10.1002/rnc.6631.

Wang, J., Yang, S., An, Z., Han, S., Zhang, Z., Mangharam, R., Ma, M., Miao, F., 2023c. Multi-agent reinforcement learning guided by signal temporal logic specifications. http://dx.doi.org/10.48550/ARXIV.2306.06808, CoRR abs/2306.06808.

Wang, R., Zhuang, Z., Tao, H., Paszke, W., Stojanovic, V., 2023d. Q-learning based fault estimation and fault tolerant iterative learning control for MIMO systems. ISA Trans. 142, 123–135. http://dx.doi.org/10.1016/j.isatra.2023.07.043.

Xiong, Z., Agarwal, I., Jagannathan, S., 2022. HiSaRL: A hierarchical framework for safe reinforcement learning. In: Proceedings of the Workshop on Artificial Intelligence Safety 2022 (SafeAI 2022) Co-Located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI2022), Virtual, February, 2022. In: CEUR Workshop Proceedings, Vol. 3087, CEUR-WS.org, URL https://ceur-ws.org/Vol-3087/paper_17.pdf.

Yang, Y., Chen, L., Zaidi, Z., van Waveren, S., Krishna, A., Gombolay, M., 2024. Enhancing safety in learning from demonstration algorithms via control barrier function shielding. In: Proceedings of the 19th ACM/IEEE International Conference on Human-Robot Interaction. pp. 820–829. http://dx.doi.org/10.1145/3610977.3635002.

Zhang, W., Bastani, O., Kumar, V., 2019. Mamps: Safe multi-agent reinforcement learning via model predictive shielding. arXiv preprint arXiv:1910.12639.

Zhang, Y., You, J., Shi, L., Shao, J., Zheng, W.X., 2023. Constrained coverage of unknown environment using safe reinforcement learning. In: 62nd IEEE Conference on Decision and Control (CDC). IEEE, pp. 3415–3420. http://dx.doi.org/10.1109/CDC49753.2023.10383702, URL https://ieeexplore.ieee.org/document/10383702/.

Zhao, Z., 2023. How to ensure a safe control strategy? Towards a SRL for urban transit autonomous operation. arXiv preprint arXiv:2311.14457.

Zhao, W., He, T., Liu, C., 2022. Model-free safe control for zero-violation reinforcement learning. In: Faust, A., Hsu, D., Neumann, G. (Eds.), Proceedings of the 5th Conference on Robot Learning. In: Proceedings of Machine Learning Research, Vol. 164, PMLR, pp. 784–793, URL https://proceedings.mlr.press/v164/zhao22a.html.

Zhu, H., Xiong, Z., Magill, S., Jagannathan, S., 2019. An inductive synthesis framework for verifiable reinforcement learning. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation. Association for Computing Machinery, pp. 686–701. http://dx.doi.org/10.1145/3314221.3314638.

Zhu, Z., Zhao, H., 2021. A survey of deep rl and il for autonomous driving policy learning. IEEE Trans. Intell. Transp. Syst. 23 (9), 14043–14065. http://dx.doi.org/10.1109/TITS.2021.3134702.