Shield Decentralization for Safe Reinforcement Learning in General Partially Observable Multi-Agent Environments

Extended Abstract

Daniel Melcer
Northeastern University
Boston, MA, USA
melcer.d@northeastern.edu

Christopher Amato
Northeastern University
Boston, MA, USA
c.amato@northeastern.edu

Stavros Tripakis
Northeastern University
Boston, MA, USA
stavros@northeastern.edu

ABSTRACT

As reinforcement learning (RL) is increasingly used in safety-critical systems, it is important to restrict RL agents to only take safe actions. Shielding is a promising approach to this task; however, in multi-agent domains, shielding has previously been restricted to environments where all agents observe the same information. Most real-world tasks do not satisfy this strong assumption. We discuss the theoretical foundations of multi-agent shielding in environments with general partial observability and develop a novel shielding method which is effective in such domains. Through a series of experiments, we show that agents that use our shielding method are able to safely and successfully solve a variety of RL tasks, including tasks in which prior methods cannot be applied.

KEYWORDS

Shields; Multiagent; Partial Observability; Reinforcement Learning

ACM Reference Format:
Daniel Melcer, Christopher Amato, and Stavros Tripakis. 2024. Shield Decentralization for Safe Reinforcement Learning in General Partially Observable Multi-Agent Environments: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION & RELATED WORK

Deep reinforcement learning is gaining popularity as a general method to solve a wide variety of tasks [9, 10, 14]; however, training requires extensive exploration [13], and trained agents may still behave unpredictably [6], rendering RL potentially dangerous for many applications. Shielding addresses this issue in fully [1, 3] and partially observable [5, 8, 11] single-agent domains, but multi-agent shielding currently requires assumptions on communication [7] or observability [4, 12]. Therefore, we introduce a shielding method for communication-free multi-agent domains with general partial observability, and show that this method prevents safety violations.

2 DEC-POEAS, SAFETY, AND SHIELDING

A reinforcement learning environment may be complex: it may have an infinite state space, a complex stochastic transition structure, and a reward function which is irrelevant for safe operation. Therefore, shielding methods usually take as input an abstraction of the environment, and a safety specification defined over this abstraction [1, 7, 12]. We introduce a new abstraction that is able to capture partially observable multi-agent environments:

Definition 1 (Dec-POEA)**.** A *Decentralized Partially-Observable Environment Automaton (Dec-POEA)* is a tuple $\phi = (\mathbb{D}, Q, \mathbb{A} = \prod_{i\in\mathbb{D}} \mathbb{A}_i, \Omega = \prod_{i\in\mathbb{D}} \Omega_i, O = \prod_{i\in\mathbb{D}} O_i, \delta, s_0)$ where $\mathbb{D}$ is a set of agents, Q is a set of states, $\mathbb{A}$ is the action space of all agents, Ω is the set of joint observations, $O : Q \to \Omega$ returns the observation for a given state, $\delta : Q \times \mathbb{A} \to 2^Q$ represents the transition function of the Dec-POEA, and $s_0 \subseteq Q$ is the set of possible initial states.

The joint action and observation spaces are the Cartesian product of individual action and observation spaces for each agent.

A safety specification $\overline{\delta} : 2^{Q\times\mathbb{A}}$ represents a set of unsafe transitions; for example, all transitions which result in a collision.

We assume that the user provides a Dec-POEA and safety specification that correctly abstracts the underlying environment; i.e. a policy over ϕ that avoids any transition in $\overline{\delta}$ corresponds to a policy that acts safely in the underlying environment. We first apply existing shielding methods [3, 7] to compute a *Centralized Fully-Observable Shield* (CFOS) $\mathcal{S} : Q \to 2^{\mathbb{A}}$ that maps an underlying state in ϕ to a set of safe joint actions. A CFOS allows for *deadlock-free* action selection: if a joint action in $\mathcal{S}(q)$ leads to state q', there always exists at least one safe joint action in $\mathcal{S}(q')$. However, agents cannot use $\mathcal{S}$ directly: they have no access to q, and the safe actions for an agent depend on other agents' chosen actions.

Therefore, our challenge is to find a set of *individual shields* that each agent can follow that results in safe joint behavior:

Problem 1. Given CFOS $\mathcal{S}$, find a set of *Individual Shields* $\mathcal{D}_i : \Omega_i \to \mathbb{A}_i$ for each $i \in \mathbb{D}$ such that at every state $q \in Q$, the Cartesian product $\prod_{i\in\mathbb{D}} \mathcal{D}_i(O_i(q))$ is a non-empty subset of $\mathcal{S}(q)$.

3 CFOS DECOMPOSITION

While agents do not have access to the current environment state, the agents can still infer some information about the state based on their observation. Let $R_i(o_i) = \{q \in Q | O_i(q) = o_i\}$: for a given $o_i \in \Omega_i$, the set of states which result in agent i observing o_i.

For joint action set $X \subseteq \mathbb{A}$, let a *decomposition* of X be a sequence of sets of individual actions $X[i] \subseteq \mathbb{A}_i$ for $i \in \mathbb{D}$, where the product $\prod_{i\in\mathbb{D}} X[i] \subseteq X$. Let $dec(X)$ be the set of all decompositions of X.

Consider if we were to arbitrarily choose $A(q) \in dec(\mathcal{S}(q))$ for every $q \in Q$. The individual shields $\mathcal{D}_i(o_i) = \cap_{q\in R_i(o_i)} A(q)[i]$ satisfy the subset constraint given in Problem 1. We then test if this intersection is non-empty for every observation—if so, this process (denoted as the *naive method*) produces a shield; otherwise, it fails.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

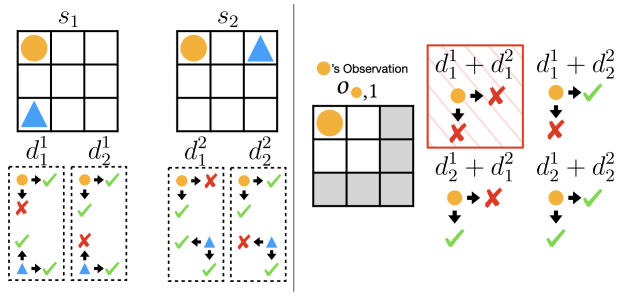

Figure 1: Illustration of the decomposition problem for an environment with a small observation radius. If decompositions d_1^1 and d_1^2 are chosen for states 1 and 2 respectively, the circle agent has no safe actions available when it observes that it is in the top-left and does not see triangle. Any other combination of decompositions would be safe.

If the naive method fails on an arbitrarily chosen set of decompositions, it is possible to test other decompositions for non-emptiness via brute-force search; however, this would be extremely inefficient. Rather, it is possible to define the problem in terms of boolean constraints, enabling us to use state-of-the-art SAT solvers to efficiently find a solution, or to confirm that no solution exists [2].

Our formulation contains three types of constriants, as follows:

$$\bigwedge_{q\in Q, d_q\in dec(\mathcal{S}(q)), i\in\mathbb{D}, a\in(\mathbb{A}_i\setminus d_q[i])} A(q) = d_q \implies a \notin \mathcal{D}_i(O_i(q)) \quad (1)$$

$$\bigwedge_{q\in Q} \bigvee_{d_q\in dec(\mathcal{S}(q))} A(q) = d_q \quad (2)$$

$$\bigwedge_{i\in\mathbb{D}, o_i\in\Omega_i} \bigvee_{a\in\mathbb{A}_i} a \in \mathcal{D}_i(o_i) \quad (3)$$

We encode these constraints into SAT clauses, with one class of variables to represent $A(q) = d_q$, and another class of variables to represent $a \in \mathcal{D}_i(o_i)$. We refer to this process as the *SAT method*.

It is also possible to incorporate history into this process by constructing a modified Dec-POEA ϕ^* such that each state in ϕ^* corresponds to a n-step sequence of states in ϕ. The modified transition function $\delta^*((q_{t-n},\ldots,q_t),a) = (q_{t-n+1},\ldots,\delta(q_t,a))$, and observation function $O^*((q_{t-n},\ldots,q_t)) = (O(q_{t-n}),\ldots,O(q_t))$.

4 EXPERIMENTS

We use environments **Gridworld-Collision** (GC) and **Particle-Momentum** (PM-P for position-only observation; PM-PV to observe position and velocity) found in earlier shielding works [7, 12], and introduce two new environments that include partial observability: In **Nearby-Obs-2** (NO-2), we re-use the maps and dynamics from Gridworld-Collision, but restrict observations of agent locations to a radius of 2. In **Flashlight** (Fl-6, Fl-10), agents only observe other agents up to one unit away on a 6x6 or 10x10 grid, but have an action available to temporarily increase this distance to 5; after doing so, the agents must wait a number of steps of "recharge time" prior to being able to increase the visibility again.

Training and evaluation setups for GC and PM replicates [12] as closely as possible. For NO-2, Fl-6, and Fl-10, we add a recurrent layer to the PM agent (sequence length of 4), anneal to and evaluate with $\epsilon = 0$, set $\gamma = 0.98$, and report discounted sum of rewards.

Table 1: Average sum of RL rewards over 100 test episodes and minimum history length (superscript) for which decentralization succeeds. GC and NO-2 show results for map "ISR". Unshielded agents include total safety violations during testing. Shielded agents incur zero safety violations. Prior work [12] fails to decentralize a shield for NO-2.

Env	[12]	Naive	SAT	Central	None (Violations)
GC	76.1	87.4^0	85.7^0	86.9	83.6 (2.0)
PM-PV	94.6	94.8^0	94.8^0	94.6	94.3 (1.6)
PM-P	83.1	84.4^1	86.6^1	81.3	52.7 (100.3)
NO-2	-	75.7^0	81.0^0	66.3	1.2 (140.7)

Table 2: RL performance for Fl-6 and FL-10, with varying recharge times (RT), and history length ($n = 0, 1$ for SAT). Shield decentralization fails with [12] or naive method.

Env	RT	SAT-0	SAT-1	Central	None (Violations)
Fl-6	3	-0.6	77.6	84.1	84.4 (4.6)
	4	-52.3	74.5	84.0	84.1 (5.6)
	5	-53.6	58.3	83.2	84.1 (4.1)
	6	-59.2	40.3	80.4	83.3 (5.7)
Fl-10	2	-49.9	48.5	38.3	20.4 (32.7)
	3	-51.3	52.4	29.7	26.4 (9.5)

Tables 1 and 2 show average sum of rewards over 10 seeds (50 seeds for NO-2) using random starting locations. Our method uniquely succeeds for enforcing a safety specification without communication between agents, and often performs comparably to agents with a centralized shield. Augmenting the shield with history can improve performance by allowing for additional safe actions, even when history is not strictly necessary to enforce safety. Note that RL task performance was generally high-variance, but all executions with shielded agents have zero safety violations.

5 CONCLUSION

Multi-agent shielding was previously limited to domains with communication, or where all agents observed enough information to determine the environment state. This paper presents, to our knowledge, the first shielding method without such assumptions; the resulting shields are often permissive enough to allow agents to solve reinforcement learning problems under a shielding protocol. We are currently investigating methods to further improve the scalability of this method, and we plan to develop a process for iteratively improving a shield to increase resulting RL performance.

ACKNOWLEDGMENTS

This work has been supported in part by NSF CCF award #2319500, *FMitF: Track I: Safe Multi-Agent Reinforcement Learning with Shielding*, and used the Discovery cluster, supported by Northeastern University's Research Computing team.

ETHICS STATEMENT

Shielding can be a powerful tool to prevent mistakes as the result of an incorrectly trained RL agent. However, as with other shielding works, there is an inherent risk that the creator or user of an RL system could be overconfident in a shielded RL agent—it is possible for an environment or safety specification to be incorrectly specified, or for there to be a bug in the implementation of the shield synthesis tool. Care must be taken when applying shielding to a given problem, and there should be redundant systems in place for any safety-critical process.

REFERENCES

[1] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. 2018. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32. AAAI Conference on Artificial Intelligence, New Orleans, LA, 10 pages.

[2] Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximillian Heisinger. 2020. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling Entering the SAT Competition 2020. In *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions (Department of Computer Science Report Series B, Vol. B-2020-1)*, Tomas Balyo, Nils Froleyks, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda (Eds.). University of Helsinki, Alghero, Italy, 51–53.

[3] Roderick Bloem, Bettina Könighofer, Robert Könighofer, and Chao Wang. 2015. Shield synthesis: Runtime enforcement for reactive systems. In *Tools and Algorithms for the Construction and Analysis of Systems: 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings 21*. Springer, International Conference on Tools and Algorithms for the Construction and Analysis of Systems, London, UK, 533–548.

[4] Steven Carr, Nils Jansen, Suda Bharadwaj, M.T.J. Spaan, and Ufuk Topcu. 2021. Safe Policies for Factored Partially Observable Stochastic Games. In *Robotics: Science and System XVII*, Dylan A. Shell, Marc Toussaint, and M. Ani Hsieh (Eds.). Robotics: Science and Systems, Virtual, 11 pages. https://doi.org/10.15607/RSS.2021.XVII.079 Robotics: Science and Systems XVII, 2021 ; Conference date: 12-07-2021 Through 16-07-2021.

[5] Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. 2022. Safe reinforcement learning via shielding for pomdps. , 21 pages.

[6] Jack Clark and Dario Amodei. 2016. Faulty reward functions in the wild. https://openai.com/research/faulty-reward-functions

[7] Ingy ElSayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. 2021. Safe Multi-Agent Reinforcement Learning via Shielding. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 483–491.

[8] Sebastian Junges, Nils Jansen, and Sanjit A. Seshia. 2021. Enforcing Almost-Sure Reachability in POMDPs. In *Computer Aided Verification*, Alexandra Silva and K. Rustan M. Leino (Eds.). Springer International Publishing, Cham, 602–625.

[9] Will Knight. 2020. AI Helps Warehouse Robots Pick Up New Tricks. https://www.wired.com/story/ai-helps-warehouse-bots-pick-new-skills/

[10] Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. 2018. Data center cooling using model-predictive control. *Advances in Neural Information Processing Systems* 31 (2018), 10 pages.

[11] Giulio Mazzi, Alberto Castellini, and Alessandro Farinelli. 2021. Rule-based Shielding for Partially Observable Monte-Carlo Planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 31. International Conference on Automated Planning and Scheduling, Virtual, 243–251.

[12] Daniel Melcer, Christopher Amato, and Stavros Tripakis. 2022. Shield Decentralization for Safe Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems* 36 (2022), 13 pages.

[13] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

[14] Peter R. Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J. Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Kompella, HaoChih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D. Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. 2022. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature* 602, 7896 (Feb. 2022), 223–228. https://doi.org/10.1038/s41586-021-04357-7