UVA CS 6316: Machine Learning : 2019 Fall
Course Project: Deep2Reproduce @
https://github.com/qiyanjun/deep2reproduce/tree/master/2019Fall

Safe Reinforcement Learning via Shielding

Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina y: Könighofer, Scott Niekum, Ufuk Topcu

Reproduced and Slides By: Carl Hildebrandt

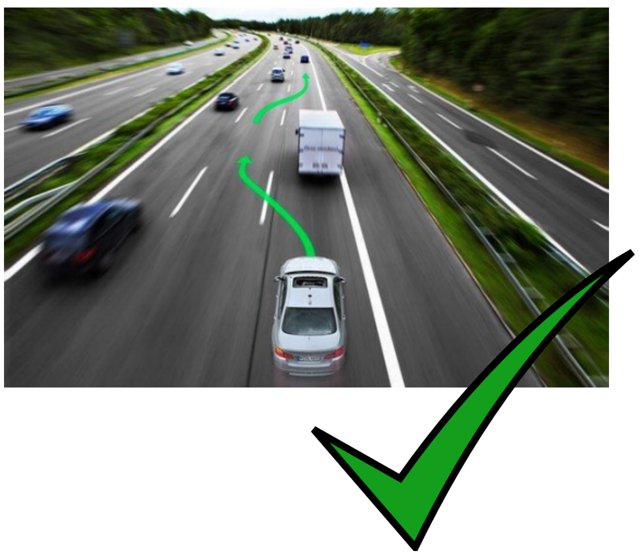
October 20, 2019

Motivation

Advances in machine learning have given rise to **autonomous systems**

One particular type of machine learning is **reinforcement learning** which is good at:

Complicated Tasks[1]



Dynamic Environments[2]

Safety[3]

1. https://towardsdatascience.com/planning-the-path-for-a-self-driving-car-on-a-highway-7134fddd8707
2. https://fortune.com/2016/09/08/self-driving-cars-environment/
3. https://www.wired.com/2017/03/uber-redeploys-self-driving-cars-wreck-arizona/

Background

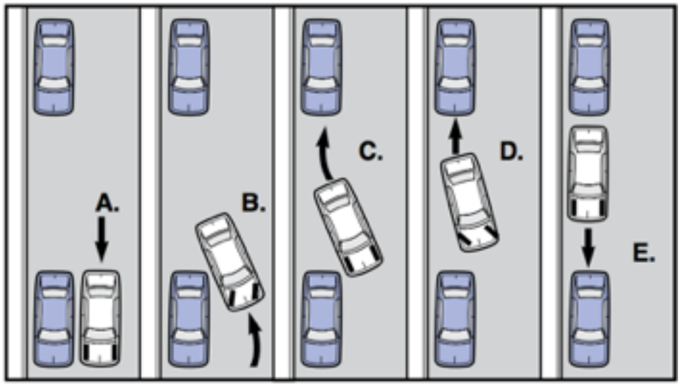
Reinforcement learning - discover policies that **maximize a reward**
Reinforcement learning - **does not guarantee safety**

Example:
Parallel Parking

Maximize Reward

Safety

1



2

1. https://1funny.com/wp-content/uploads/2009/04/bad-parking.jpg
2. https://www.drivingtips.com/minimizing-risks-of-accidents-while-driving/

Related Work

Definition: An exploration process is called **safe** if no undesirable states are ever visited[14]

Two mostly isolated threads of work:

Safety in reinforcement learning	Safety in formal methods
• Safety achieved by incorporating external knowledge[8][14] • Using temporal logic to generate invariance properties[8][1] • Using a teacher network [22][4] • Using a human teacher [15][19]	• Receding horizon control which combines continuous control and discrete correctness guarantees[24] • Simple safe controllers can be computed directly [9] and more complex ones can be computed using a combination of low level controllers[5] • Shield synthesize to enforce safety properties on controller[3] • …

Target Task

This paper sets out to generate a technique which **guarantees safety and correctness** of a learning agent

They aim to do this using the **correctness guarantees provided by formal methods** and combine it with the **optimality provided by reinforcement learning**

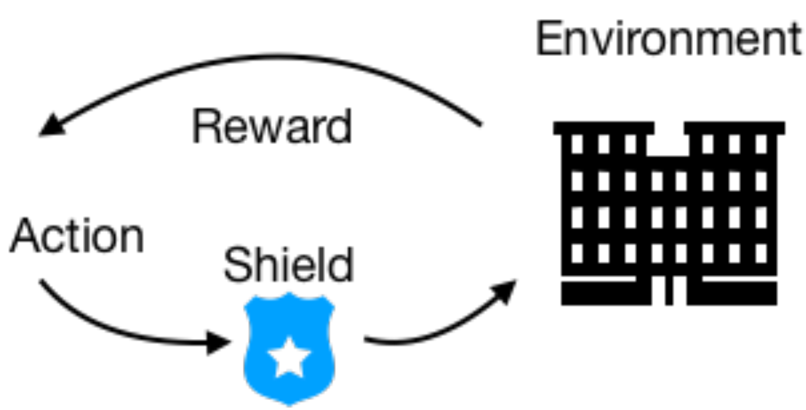
Goals:
1. Enforce safety properties in a traditional reinforcement learning setting.
2. Restrict the agent as little as possible by having minimum interference.
3. Create a system which clearly separates safety and optimality.

Correctness and Safety	+	Optimality	=	Safe Optimal Solution
Formal Methods		Reinforcement Learning		Safe Learning

An Intuitive Figure Showing WHY Claim

Reinforcement learning leans by taking an action in an environment and getting a reward which it uses to update its policy:

The idea of this paper is to block actions which are unsafe, before they are enacted in the environment:



Proposed Solution

1. Generate a set of system specifications and an abstraction of the agents environment expressed as temporal logic.

2. Synthesize a reactive system (shield) which enforces the safety properties of the systems specifications.

3. Modify the learning loop (as shown on the right) by placing the shield in 1 of 2 places:

    1. Before the learning agent, thus removing any unsafe actions.
    2. After the learning agent, thus monitoring the selected actions and correcting them only if an unsafe action is chosen.

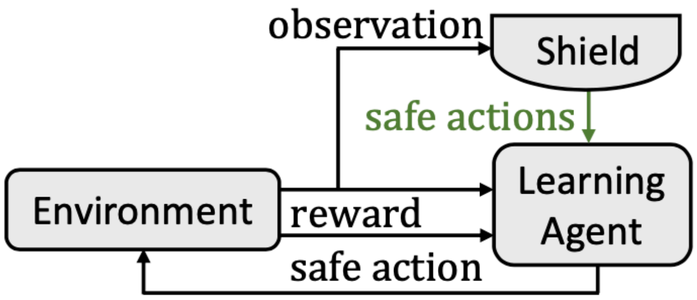


Fig. 1: Preemptive Shielding.

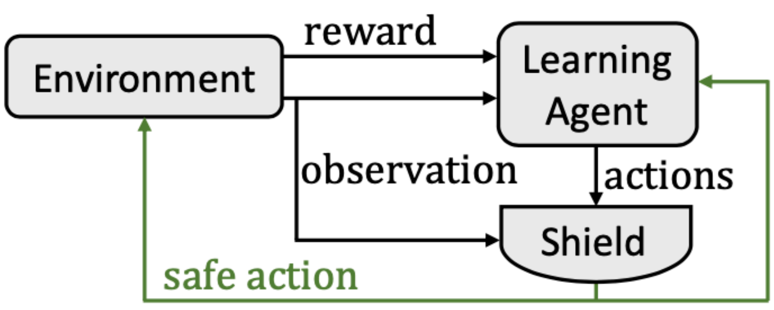


Fig. 2: Post-Posed Shielding.

Implementation

Definition 1. *Safe reinforcement learning is the process of learning an optimal policy while satisfying a temporal logic safety specification φ^s during the learning and execution phases.*

1. System specifications are given as temporal logic. For instance to state that an autonomous car must never run out of fuel: *G(fuel_level > 0)*

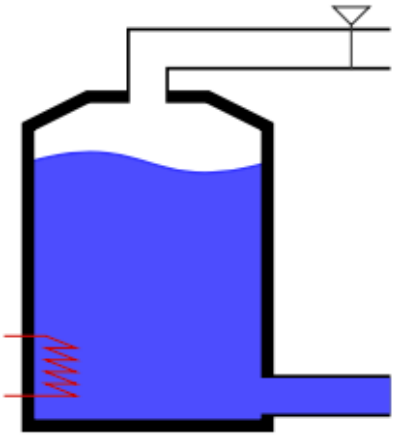
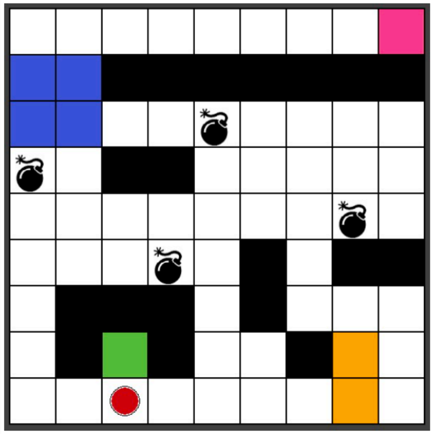
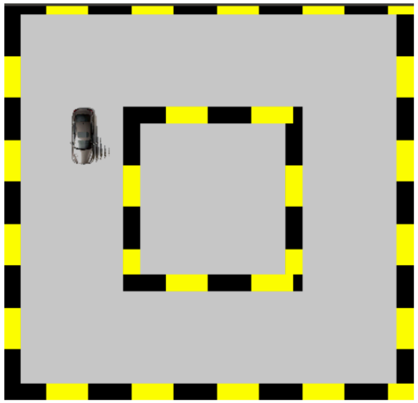
2. Convert the safety specifications into an automaton in which only safe states *F* may be visited: $\varphi^s = (Q, q_0, \Sigma, \delta, F)$

3. Convert the environment abstraction (often modeled as a Markov Decision Process) into an automaton: $\varphi^{\mathcal{M}} = (Q, q_0, \Sigma, \delta, F)$

4. Use reactive synthesis to enforce φ^s by solving a safety game built from φ^s and φ^M which is won if the system only ever visits safe states *F*.

Data Summary

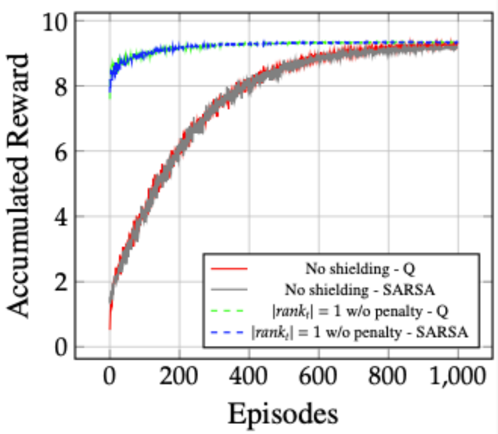
They use four test environments to test their proposed solution to safety.

Water Tank	Grid World	Self-Driving Car	Atari Seaquest™
			
Keep water warm while minimizing energy consumption. Water runs out of tank constantly. Cold water runs into take through controllable switch. Can not overflow or run empty.	Visit all colored regions in a given order while not crashing into walls or sitting on bombs for more than two consecutive steps.	Drive clockwise around the track. The car can only turn 7.5 degrees and moves 3 pixels each time step. Avoid crashing into walls.	Control a submarine to collect divers while avoiding obstacles. The submarine needs to surface before running out of oxygen.

Experimental Results

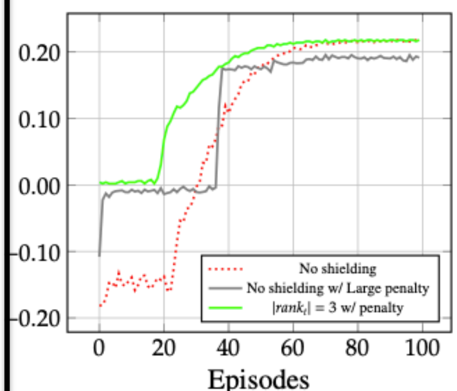
The main experimental results are shown below.

Water Tank



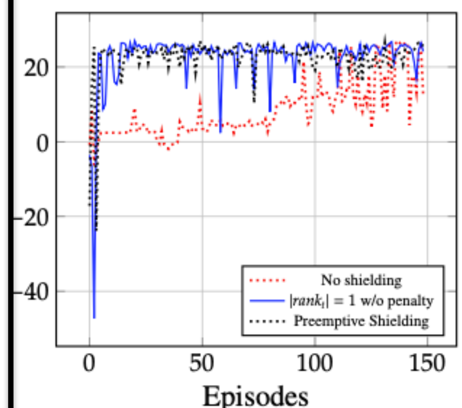
Shielded learning is shown in blue and green dashed lines. Unshielded learning is shown in red and gray solid lines.

Grid World



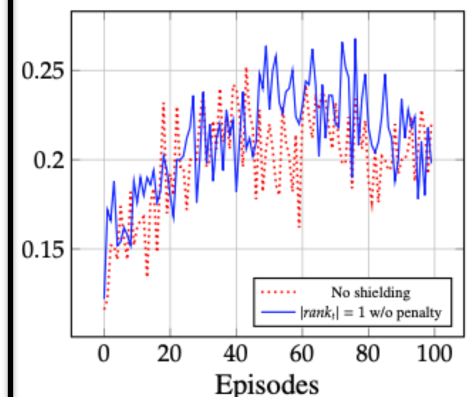
Shielded learning is shown in the solid green line. Unshielded learning is shown in red dashed line and gray line. Gray line represents large negative penalty.

Self-Driving Car



Red dashed line shows unshielded learning. Blue and black lines show shielded learning.

Atari Seaquest™

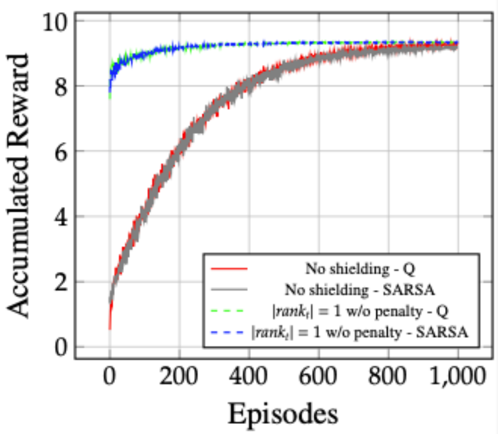


Red dashed line shows unshielded learning. Blue solid line shows shielded learning.

Experimental Analysis

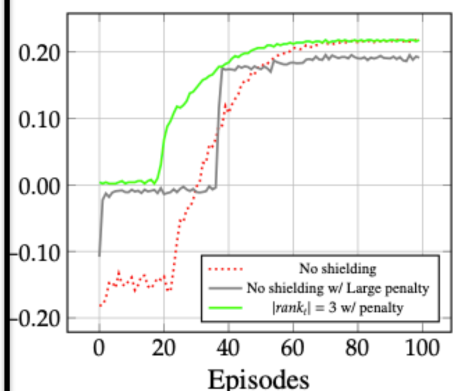
The main observation are:

Water Tank



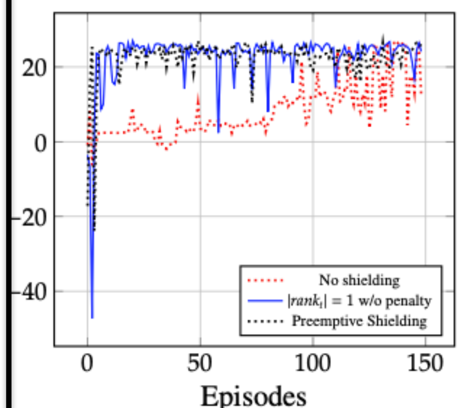
1) Both unshielded and shielded learning reaches optimal policy.
2) Shielded learning converges to optimal policy much faster.

Grid World



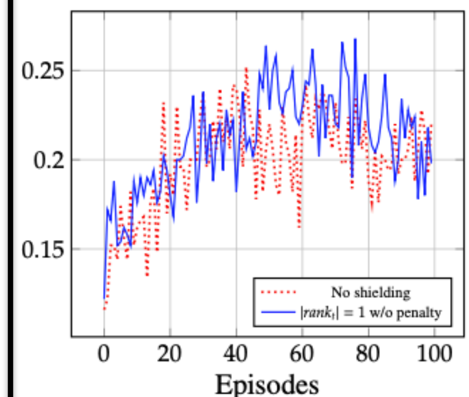
1)Both unshielded techniques get negative rewards - implying they violate a safety constraint during training
2) High negative reward does not convert to optimal policy

Self-Driving Car



1)Unshielded technique still crashes at the end of training.
2)Shielded techniques learn more rapidly.

Atari Seaquest™



1)Shielding did not change the performance of the learning agent.
2) Safety properties never violated when shield was implemented.

Result Reproduction

Why I selected this paper:

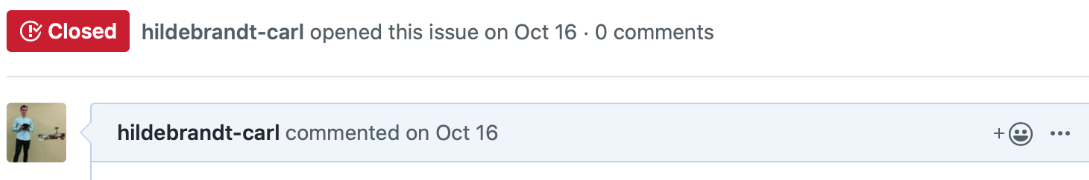
Water Tank	Grid World	Self-Driving Car	Atari Seaquest™
		Carl	

- There are 4 scenarios (1 per group member):
  - The **entire approach** is applied to **each scenario.**
  - Reproducing 1 set of results requires understanding and implementing the entire paper.
- I selected to implement the self-driving car as this is the closest to my research.

Result Reproduction

Why replicating papers is beneficial:

- You get a **deeper understanding** of the approach.
- You confirm that the **results are correct**, and not tailored to look good.
- You benefit the developer by **checking their work**.

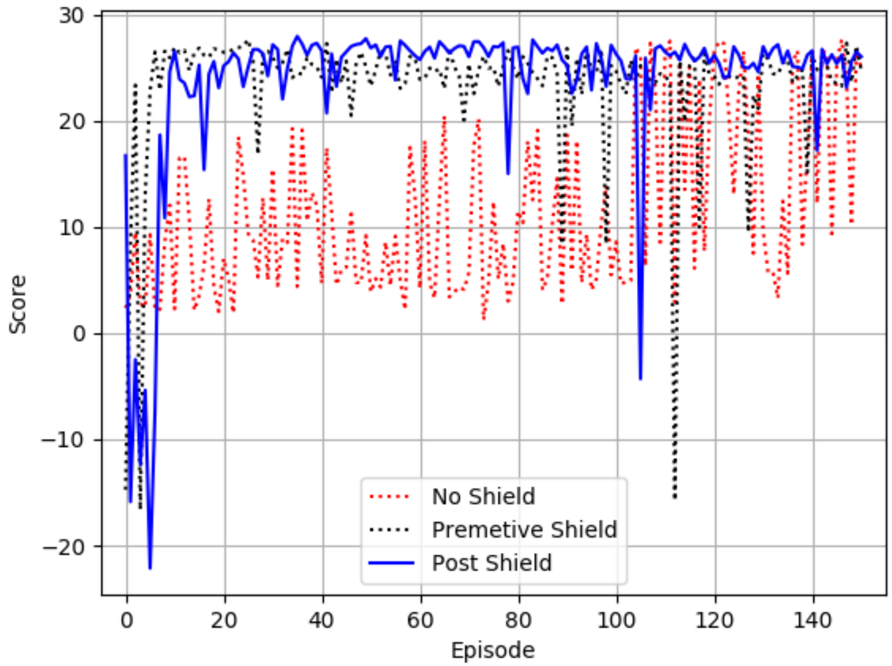


- I found a security issues on their Github repo and was able to submit a issue which was fixed and closed.

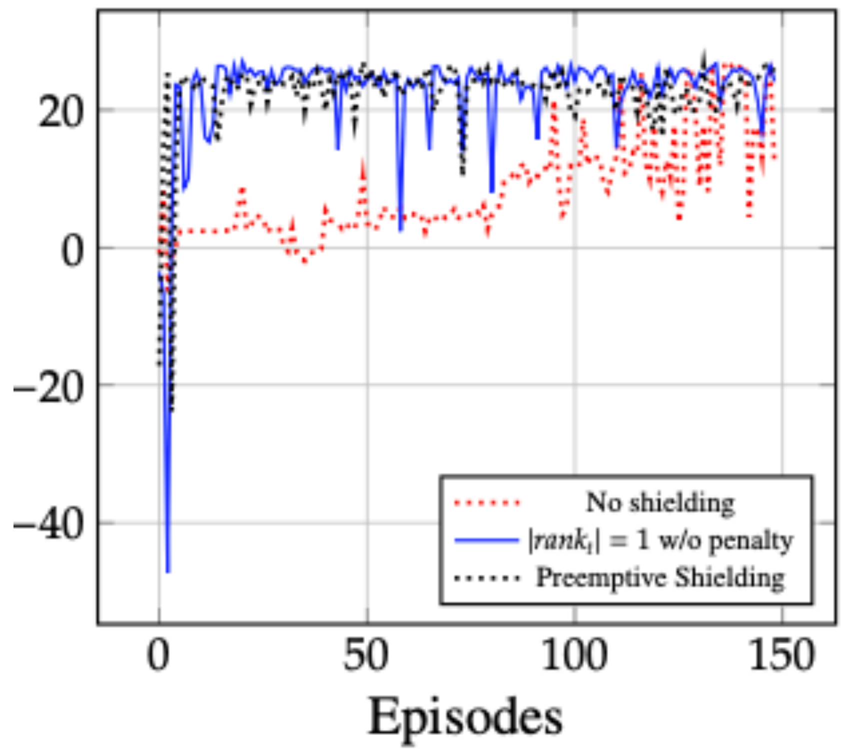
Here is a link to my review of the code:
https://github.com/hildebrandt-carl/SafeReinforcementLearning

Result Reproduction

My Results



Original Results



Conclusion and Future Work

- They provide a method for reinforcement learning under safety constraints expressed as temporal logic specifications.
- Their technique enforces safety constraints without changing the (often complex) inner workings of reinforcement learning algorithm.
- Demonstrate that the shielded agents perform at least as well as unshielded agents. However in most cases improve performance.
- They show that their shielded agents do not violate their safety constraints.
- Downside is that you need an approximate model which describes which actions are unsafe.
- Future work is not mentioned in this paper.

References

References are given the same number as in the paper for convenience.

[8] Garcıa, Javier, and Fernando Fernández. "A comprehensive survey on safe reinforcement learning." Journal of Machine Learning Research 16.1 (2015): 1437-1480.

[14] Moldovan, Teodor Mihai, and Pieter Abbeel. "Safe exploration in Markov decision processes." arXiv preprint arXiv:1205.4810 (2012).

[1] Baier, Christel, and Joost-Pieter Katoen. Principles of model checking. MIT press, 2008.

[22] Quintía Vidal, Pablo, et al. "Learning on real robots from experience and simple user feedback." (2013).

[19] Thomaz, Andrea Lockerd, and Cynthia Breazeal. "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance." Aaai. Vol. 6. 2006.

[4] Clouse, J. "On Integrating Apprentice Learning and Reinforcement Learning TITLE2." (1997).

[15] Pecka, Martin, and Tomas Svoboda. "Safe exploration techniques for reinforcement learning–an overview." International Workshop on Modelling and Simulation for Autonomous Systems. Springer, Cham, 2014.

[24] Wongpiromsarn, Tichakorn, Ufuk Topcu, and Richard M. Murray. "Receding horizon temporal logic planning." IEEE Transactions on Automatic Control 57.11 (2012): 2817-2830.

[5] DeCastro, Jonathan A., and Hadas Kress-Gazit. "Nonlinear controller synthesis and automatic workspace partitioning for reactive high-level behaviors." Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control. ACM, 2016.

[9] Henzinger, Thomas A., and Peter W. Kopke. "Discrete-time control for rectangular hybrid automata." Theoretical Computer Science 221.1-2 (1999): 369-392.

[3] Bloem, Roderick, et al. "Shield synthesis." International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, Berlin, Heidelberg, 2015.