Safe Controller for Reinforcement Learning: Safety and Optimization

Samar Rahmouni, Prof. Giselle Reis

Carnegie Mellon University, Computer Science Department

Carnegie Mellon University

Introduction

- Reinforcement Learning (RL) is a great tool for finding an optimal solution in a stochastic environment as it converges to an optimal policy.
- RL works by trial-and-error.
- Trial-and-error cannot be deployed in real-world.
- Errors in the real world can be predicted deterministically, i.e., using a Safe Controller (SC).

Can a safe controller be integrated into RL to guarantee safety properties while still ensuring optimality of the solution?

Car Platooning : Case Study

Goal: Minimize the gap between vehicles while guaranteeing no crashes?

Why is it a problem we should care about?

1. Vehicles take less space on the road
2. Allowing more vehicles to occupy highways
3. Minimize traffic and ensure faster commute
4. Saves fuel
5. Central in the deployment of autonomous vehicles

Stochasticity in the environment

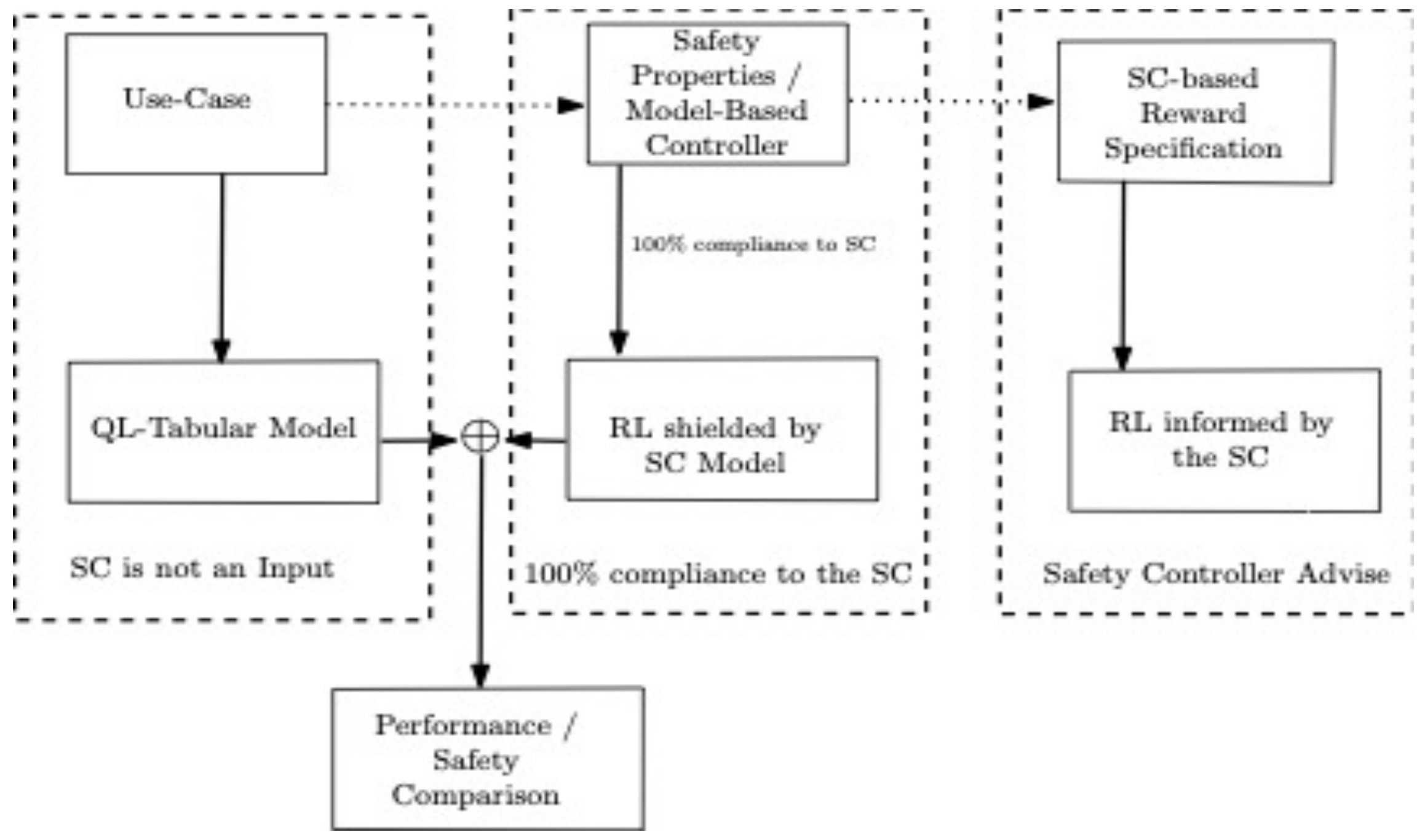
1. Stop signs and traffic lights
2. Faulty communication between vehicles
3. Faulty sensors
4. Malfunctioning breaks

It is a key problem that requires a tradeoff between optimization and safety.

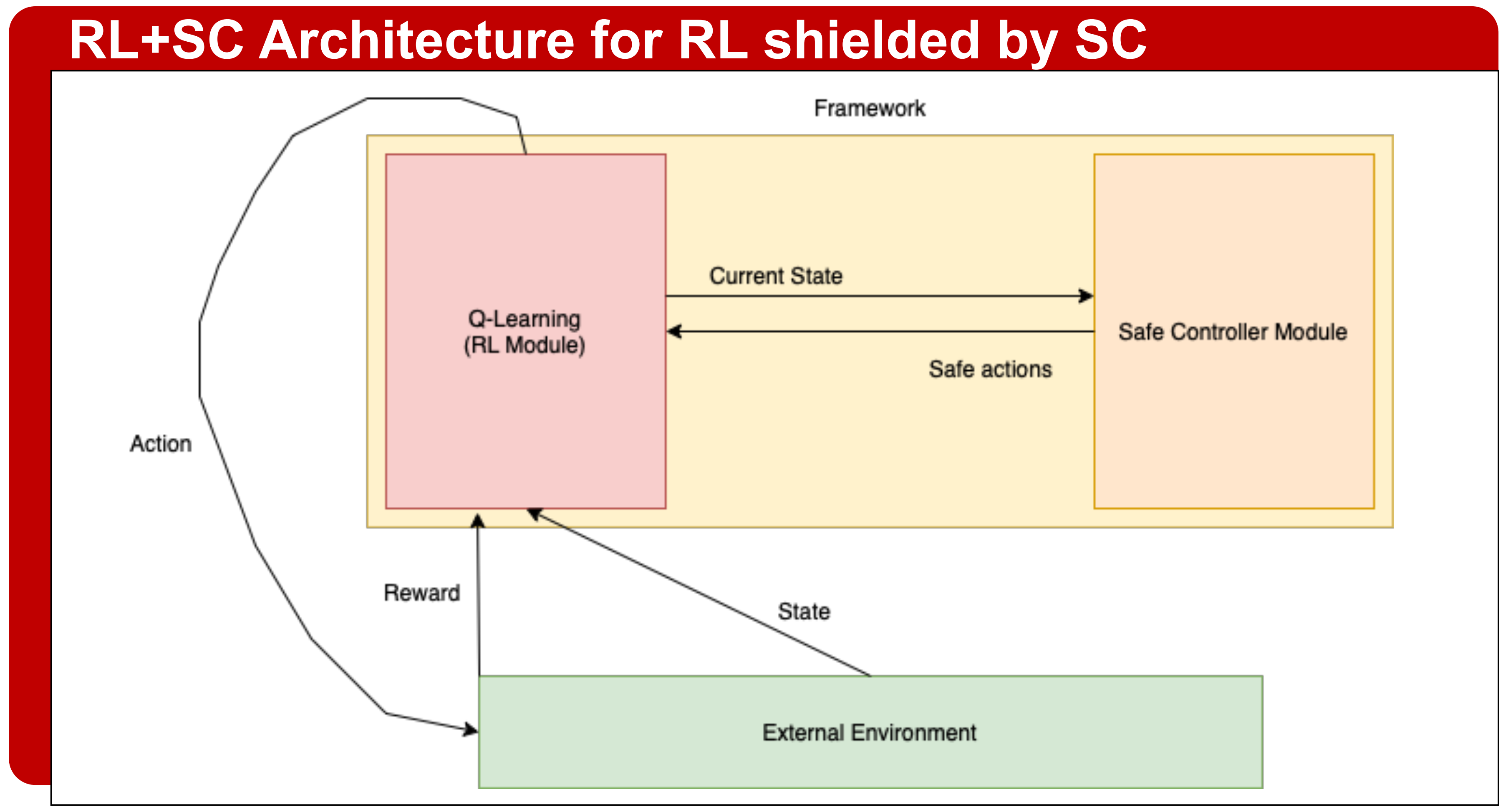
What has been done to tackle it?

- Safe controller (SC) that utilizes soft constraints to compute safe velocities [1]. **Safety.**
- Deep reinforcement learning on an actor/critic policy to optimize gap control [2]. **Optimality.**

Methodology



RL+SC Architecture for RL shielded by SC



Scenario

Model in Reinforcement Learning

•S - A state vector to denote *time, position, velocities*.

•A - An action vector for possible accelerations.

•R - A reward function, i.e., given the distance between the vehicles.

•ϕ - A transition function s.t. $\phi(s_t,a_t) = s_{t+1}$

•γ - A discounting factor in [0,1]

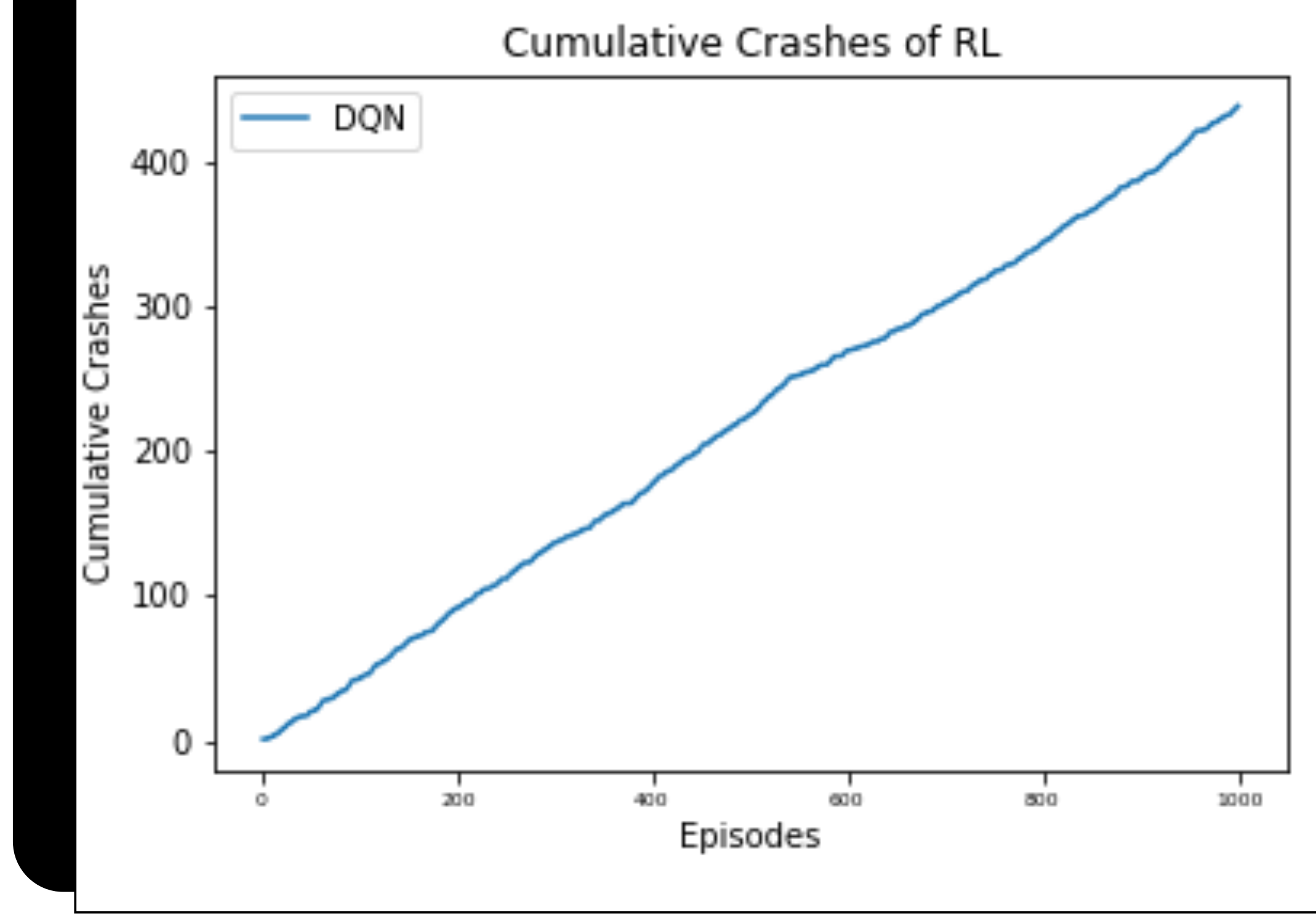
Assumes a joint state representation of the agents

Model in the Safe Controller

- Local Knowledge Base
  - Set of grounded facts p@t
- Events = ev@t
  - Equivalent to task@0
- Executable semantics
- System configuration search to ensure that no path given those accelerations result in a crash.

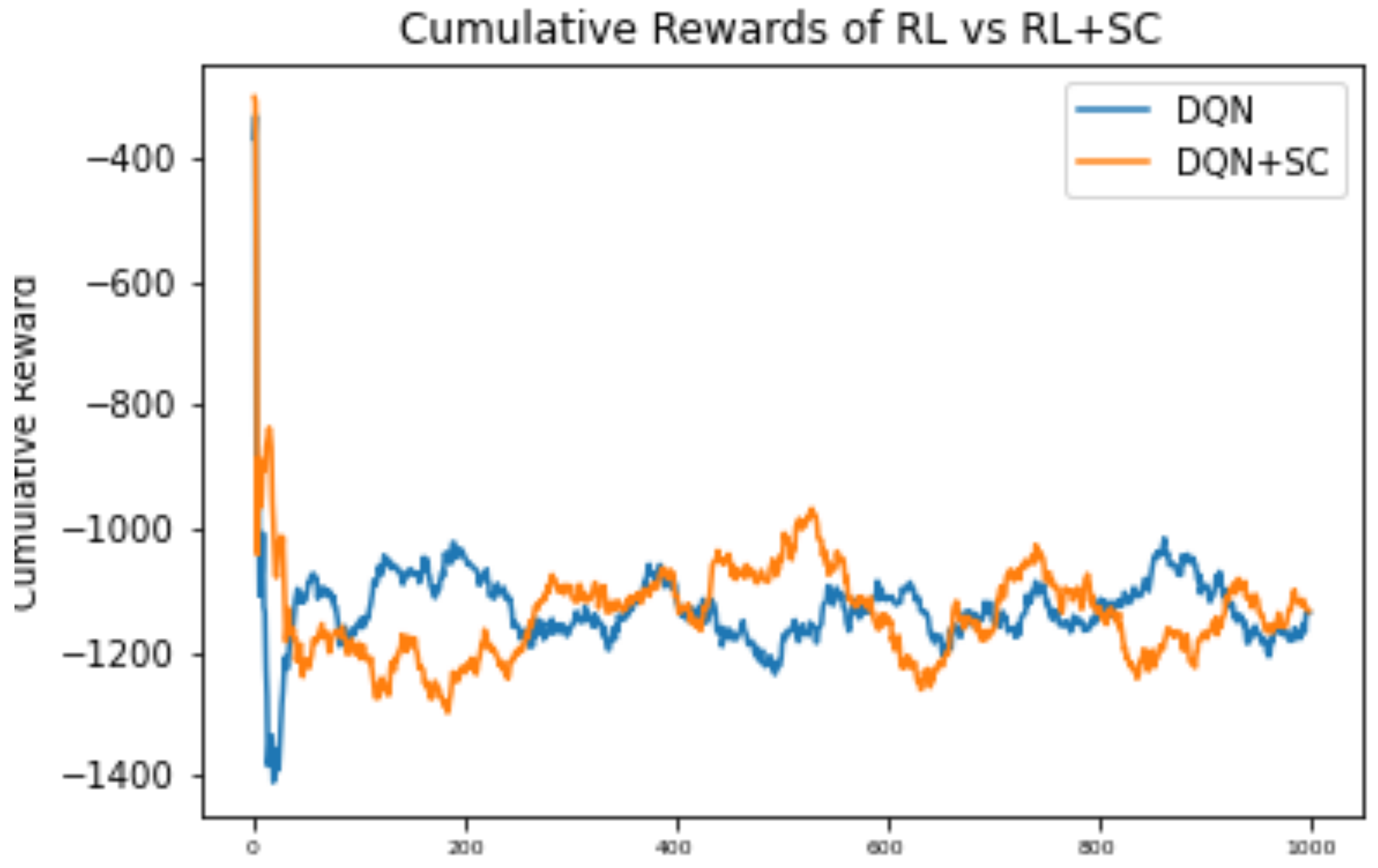
We take inspiration from this model to implement it alongside the RL

Results : Comparing Safety Guarantees



- Linear cumulative crashes indicate that almost every episode ends with a crash.
- The RL never learns not to bump into other vehicles.
- Using a SC, safety is always guaranteed.

Results : Comparing Optimization i.e., min gap



- RL+SC achieves the max cumulative compared to basic RL.
- It also never reaches the max-min of the basic RL.
- It never takes the negative reward from crashing.
- They both achieve a similar average of cumulative reward, indicating that both converge to an optimal solution, i.e., stay constant which is evident in the resulting policy.

Note that rewards are negative since we want to minimize the gap.

Results: Comparing Training Time

- The RL+SC takes on average longer as the computation of the SC gets more complex.
- It takes on average less episodes to train.

Next Steps

- Add a layer of uncertainty by considering **faulty communication and faulty sensors.**
- Consider **Logic-based inference RL** (see third box in the methodology diagram). Rather than enforcing the decision of the SC, the RL is informed by communication of a reward system towards improving its policy.

References

[1] Yuri Gil Dantas, Vivek Nigam, and Carolyn Talcott. A formal security assessment framework for cooperative adaptive cruise control. In *2020 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, 2020.

[2] Ramadan, Amr & Abdulaaty, Omar & Hussein, Ahmed & Shehata, Omar. (2020). Reinforcement Learning Based Approach for Multi-Vehicle Platooning Problem with Nonlinear Dynamic Behavior.

Acknowledgments

We would like to thank Vivek Nigam for useful discussions and continuous valuable feedback on the project. The following work is done as part of a CS Senior Honors Thesis.