A Very Cool and Catchy Title

Authors' names go here

1 Introduction

The Formula 1 is a open-wheel single-seater formula racing cars. Its tournament, the FIA Formula One World Championship, has been one of the world's premier forms of racing since 1950. Participants are all top racers, however, the champions are those that talented in choosing the correct racing line for a given track. This put them leagues above the others. However, most of these are estimates based on the experience of racers and it would be useful for them to be able to see the ideal racing line during practice and training.

In this project, we're aiming to solve this problem with the help of Reinforcement Learning techniques in order to help visualize the mistake that racers made to allow better adjustment and practice session. Specifically, we'll train an agent that can identify the optimal racing line given a track using multiple on-policy and off-policy update methods. That way, we can highlight the mistake that the racer made in comparison with the optimal racing line.

2 Background Related Work

Following, you should provide the necessary background and discuss related work in the RL literature. This section should also be about a page. Citations should be in BibTeX format [**?**]. Some approaches have proposed using RL for robot soccer [**?**].

3 Technical Approach

3.1 Environment Setup

To simulate the racing environment, we'll assume that the environment has an idealistic design where there are no elevation or bumps on the track. We'll also ignore all types of friction like air friction and terrain. The width of racetrack is kept fixed throughout and the agent can only choose the actions that select a fixed value for acceleration and deceleration of the car at each time step. There is, however, a cap speed which limits the velocity. All elements regarding the racecar like tire degradation, inertia, momentum, etc. are also ignored.

In the scenario that the agent failed to take a curve and is thrown off the racetrack, the agent will continue from the nearest location on the racetrack in

the next timestep and the velocity as well as the acceleration is resetted to zero. This will required the agents to again increase the acceleration and velocity

At first, we'll be using the racing car environment from gymnasium Python package for simplicity. Then, to add more complications into the environment later on, we'll move toward a more flexible environment that is more tailored to our specific need.

3.2 Methodology

3.2.1 Baseline

3.2.2 Deep Q-Network

3.2.3 Deep Deterministic Q-Network

3.2.4 Proximal Policy Optimization

4 Evaluation

5 Timeline and Individual Responsibilities