

# Crash Now, Don't Crash Later:

## Reinforcement Learning for Autonomous Vehicles

Peter Adam Aboud  
Ryerson University  
adam.aboud@ryerson.ca

Christian Wang  
Ryerson University  
christian.wang@ryerson.ca

Adam Srebrnjak Yang  
Ryerson University  
a1yang@ryerson.ca

Mohammad Saleh  
Joonghani  
Ryerson University  
msalehjo@ryerson.ca

### I. INTRODUCTION AND MOTIVATION

Millions of people a year die due to automotive accidents. Most of these casualties are caused by human errors due to emotional interference, physical limits, and insufficient knowledge. These casualties could be mitigated by automation; however, automation has its own set of problems. This project serves to explore the implementation of different algorithms to solve self-driving car problems. This exploration will assess the performance of two reinforcement learning algorithms: Q-Learning, and Deep Deterministic Policy Gradient (DDPG) on a two-dimensional self-driving car.

### II. METHODS

For the project, we will divide our experiment into three tiers:

- 1) *Tier 1: A self-learning agent completing a 2d race track*
- 2) *Tier 2: The same agent meeting a performance threshold*
- 3) *Tier 3: An agent conducting the race experiment with obstacles or traffic*

We will analyze two different reinforcement learning algorithms: QL, and DDPG on each tier. These algorithms will be implemented in OpenAI gym environments: Highway-V0 and CarRacing-V0. Additionally, we will visualize the metrics used in our experiment. The following Python Libraries may be used to execute our experiments: Gym, TensorFlow, Matplotlib, and Keras, Numpy, Scipy.

#### A. Environment Selection and Setup

- What gyms we are using OPENAI
- What environments
  - o CarRacing-V0

- This top-down racing environment allows the agent to learn from pixels as it navigates a randomly generated racetrack.
- o Highway-V0
  - A more comprehensive environment that features traffic, a multilane roadway, roundabouts, intersection negotiation, and merging scenarios.
- We will use Google Cloud Platform (GCP) to train and run our simulations.

#### B. Algorithm Selection

Our exploration will begin with an implementation of Q-learning. If we are successful in our implementation we will proceed with other algorithms.

- Q-Learning
  - o Model-free reinforcement learning algorithm that seeks to learn the optimal action policy.
  - o Methods for reaching that optimal policy may include:
    - Monte Carlo
    - Temporal Difference
    - Dynamic Programming
- DDPG
- Deep Q Network

#### C. Evaluation Metrics

- The following are not weighted equally in descending order of importance:
  - o Vehicle collisions
  - o Time off track
  - o Time to complete
- Other metrics we will consider include:

- Dispersion across Time (DT)
- Short-term Risk across Time (SRT)
- Long-term Risk across Time (LRT)

### III. INTENDED EXPERIMENTS

A number of experiments will be carried out in order to evaluate our implementations of each tier and for each algorithm:

- Tier 1 experiments:
  - Changing the track after learning
- Tier 2 experiments:
  - Reward adjustment to optimize for speed.
  - Episodic limits
- Tier 3 experiments:
  - Adding obstacles to the track upon instantiation
  - Adding obstacles at stochastic intervals
  - Adding additional actors to track
  - Randomized variable assignment.

### IV. PLANNING AND MILESTONES

#### A. Milestone 1

- February 14
- Environments, workflow, repository, GCP, configured and installed.

#### B. Milestone 2

- February 28
- Q learning implemented and working for Tier 1

#### C. Milestone 3

- March 7
- DDPG implemented and working for Tier 1

#### D. Milestone 4

- March 14
- Q learning and DDPG for Tier 2
- Start project report

#### E. Milestone 5

- March 21
- Tier 2 completed and finalized (including all information required to get a B)

#### F. Milestone 6

- April 4
- Complete any extra materials/experiments/implementations related to Tier 3

#### G. Milestone 7

- April 11
- Project wrap up
- README updating, documentation, images, Project asset creation, data visualization, etc.

### REFERENCES

- [1] <https://gym.openai.com/envs/CarRacing-v0>
- [2] <https://github.com/eleurent/highway-env>
- [3] <https://www.tensorflow.org/agents/tutorials/>
- [4] <http://narimanfarsad.com/cps824/index.html>
- [5] <https://sites.ualberta.ca/~szepesva/rlbook.html>