

Adversarial Black-Box Testing of Self-Driving Vehicle Algorithms with Reinforcement Learning

by

Oleksandra Nahorna

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Oleksandra Nahorna 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

When we hear autonomous or ego vehicles, we imagine a safe, self-driving device. However, this designation is unsuitable for describing a self-driving car because the car's automation is still being refined, and safety is being studied and constantly improved.

Right now, we can observe mostly the vehicles that are not fully autonomous to be present on the road in the physical world. The autonomy introduced requires the driver to be able to switch and take control of the vehicle in the case of an emergency occurrence. On the contrary, for the car to be able to apply full autonomy and be recognized and approved by the government to be placed on the road, the vehicle algorithm should be able to resolve and predict cases that may occur on the road and be able to respond to the situation utilizing the decision-making approach.

The physical world setting may be constrained in terms of stress-testing the ego-vehicle before its availability to the regular user-driver. Therefore, to broaden the ability to stress-test autonomous vehicles in various scenarios, the manufacturer of the autonomous vehicle and its algorithm may benefit from utilizing the ability to stress-test vehicles in the virtual vehicle setting.

For the work presented instead of enhancing the ability of safety enhancement of the autonomous vehicle by training it as a white box using the algorithm exposed by the manufacturer - we are introducing the adversarial virtual environment setting with generated accidents by the algorithm for the provider/manufacturer of the autonomous vehicle and its algorithm to be stress-tested in a black-box approach without the exposure of the technology (ego-vehicle algorithm) by the manufacturer.

Using the Reinforcement Learning technique, a single-agent or multi-agent attacker vehicle is trained to reproduce collisions in a virtual environment. The cases generated can be different, both simple and difficult to mimic a real-world setting.

When we describe autonomous vehicle adversarial blackbox testing, we identify the target vehicle as a representation of the autonomous vehicle by determining the action space, kinematics, and behaviour of the vehicle mentioned as autonomous throughout the work. This vehicle representation mimics the autonomous vehicle approach/action/decision-making within the 2D setting of the chosen virtual environment, with constraints driven by the virtual environment.

This work challenges realism, namely, it compares the likelihood of emergencies in generated car accidents to real-life cases. To do this, as in any other area, we study historical data on road accidents.

These situations are compared with selectively generated instances, and a successful comparison is recorded and described by creating a narrative for the generated situation. Thereby smoothing the boundary between virtual and real accidents on the road.

It is important to note that reproducing and preparing ego-vehicle for any situations that may arise on the road in a virtual environment can be an economical approach. After all, to train the model, only time is needed depending on the number of generated episodes, thereby reducing the cost of testing or pre-testing. And it will also help to recreate scenarios that are problematic for recreation in real, physical space. Testing cars using different algorithms can help to identify a pattern or algorithm suitable for a specific environment, type of road, and number of actors on the road. This work seeks to publicize and recommend using a virtual environment to improve the automotive industry.

Acknowledgements

I want to thank my supervisor, Professor Mark Crowley, for his invaluable mentorship throughout my master's journey. I also wanted to give my utmost thanks to my thesis readers, who allowed me to showcase the valuable theme of the safety concern addition of autonomous vehicle behaviour through their stress testing using adversarial attack via a virtual environment.

I also wanted to give my gratitude to our laboratory led by Professor Crowley which allowed me to explore and conduct research in the autonomous vehicle field, explore different stages of domain exploration, learning findings and their outcomes.

I hope the thesis described will be a great addition to bridging a gap in the exploration of enhancing autonomous vehicle safety and its adoption in the urban setting.

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	v
List of Figures	viii
List of Abbreviations	x
Chapter 1 Introduction.....	1
1.1 Summary of Contribution.....	2
1.2 Organization of Thesis	3
Chapter 2 Background.....	4
2.1 Definitions	4
2.1.1 Automation Levels	4
2.1.2 Autonomous Vehicle Testing Environments.....	6
2.2 Overview of Real-World Accidents	9
2.3 Chapter Summary	12
Chapter 3 Virtual Environment	14
3.1 Virtual environment used for This Study	14
3.1.1 Discrete Meta-Action Space.....	15
3.1.2 Vehicle Types.....	15
3.1.3 Road Types.....	16
3.2 Simulation Attacker Virtual Environment Definitions.....	23
3.3 Environment Preparation, Comparing Produced Scenarios with Real-World Cases	24
3.4 Realism in a Highway Env	34
3.5 Chapter Summary	34

Chapter 4 Contributions	36
4.1 Comparison of Policy Optimization Adoption.....	36
4.1.1 Highway Attack Env	36
4.1.2 Highway Env change with Stable Baselines 3 Library	49
Chapter 5 Conclusions.....	53
5.1 Future Work	53
5.1.1 Generate Test Cases for Physical Crash Tests.....	53
5.1.2 Adding Variety of the Vehicle Types as well as Environmental Types.....	54
5.1.3 Adding Weather Conditions	54
5.1.4 Adding a Benchmark for Policy Optimization Algorithms	54
5.1.5 Narratives Creation to Describe the Accident	55
5.1.6 Real-World Cases Generation	55
References	56
Appendix A Sample Appendix.....	64
Glossary.....	66

List of Figures

Figure 2.1: Levels of driving automation	5
Figure 2.2: Motor vehicle crash deaths per 100 million miles	9
Figure 2.3: Road fatalities per 100 000 inhabitants in Canada, 2022	10
Figure 2.4: Autonomous vs. Conventional Vehicle crashes [41]	11
Figure 2.5: Types of collision occurred in the U.S., 2022	12
Figure 3.1: Action space of Highway-Env	15
Figure 3.2: Road actors' representation	15
Figure 3.3: Aggressive and defensive vehicles parameters within a virtual environment	16
Figure 3.4: Highway environment road visualization	17
Figure 3.5: U-turn environment road visualization	18
Figure 3.6: Exit environment road visualization	20
Figure 3.7: Two-way environment road visualization	20
Figure 3.8: Roundabout environment road visualization	21
Figure 3.9: Intersection environment road visualization	22
Figure 3.10: Racetrack environment road visualization	23
Figure 3.11: Robohub robot with lidar and camera	25
Figure 3.12: Road detection and self-identification using LIDAR	25
Figure 3.13: Exit road type generated collision episodes	28
Figure 3.14: Initial and updated parameters of the intersection environment action space	29
Figure 3.15: Intersection Road type generated collision episodes	29
Figure 3.16: Two-way generated collision episodes	31
Figure 3.17: U-turn generated collision episodes	32
Figure 3.18: Merge generated collision episodes	33
Figure 4.1: Single-agent centralized optimization policy applied on a different road-types	38
Figure 4.2: Trust-region policy optimization applied on a different road-types	39
Figure 4.3: Multi-agent centralized optimization policy applied on a different road-types	41
Figure 4.4: Comparison of CPO, TRPO, MACPO, MATRPO on a highway road-type environment	43
Figure 4.5: Comparison of CPO, TRPO, MACPO, MATRPO on an exit road-type environment	44
Figure 4.6: Comparison of CPO, TRPO, MACPO, MATRPO on a merge road-type environment	45
Figure 4.7: Comparison of CPO, TRPO, MACPO, MATRPO on a U-turn environment	46

Figure 4.8: Comparison of CPO, TRPO, MACPO, MATRPO on a two-way environment	48
Figure 4.9: Inherited class for the TargettingAggressiveVehicle for the SB3 library	50
Figure 4.10 Manually tuned parameters from SB3 PPO, DQN, A2C	51
Figure 4.11 OPTUNA tuned parameters for DQN, PPO and A2C	52

List of Abbreviations

PPO – Proximal Policy Optimization

A2C – Advantage Actor Critic

RL – Reinforcement Learning

CPO – Constrained Policy Optimization

TRPO – Trust Region Policy Optimization

MATRPO – Multi-Agent Trust Region Policy Optimization

MACPO – Multi-Agent Constrained Policy Optimization

Chapter 1

Introduction

With the increase in available modes of transportation, there is a need to establish traffic regulations to ensure safe, high-quality, and coordinated interactions on the road.

Despite existing precautions such as driving, training, and legally established traffic regulations, road traffic injuries are the eighth leading cause of death for all age groups worldwide [1]. And road traffic accidents cost most countries 3% of their gross domestic product [1], [2]. Based on these statistics, approximately 9 out of 10 serious road accidents occur due to human actions, described in [3]. Using the body's available receptors, having experience after training, and driving practice when interacting with other objects on the road, a person is still prone to making mistakes or not responding on time to a rapidly changing situation on the road, which leads to road accidents.

It is precisely human carelessness (the human factor) that automated systems seek to address. The goal of automated vehicles is to ensure safe and high-quality driving. Safety and quality are achieved through algorithms programmed into the vehicle that allow it to find ways or scenarios for responding to incidents based on the information received by sensors in relation to external factors interacting with the ego-vehicle on the road. The sensors of the car have distinct functions depending on their type, such as determining the distance between objects or finding the location of the car on the road, etc. The sensors can recognize what a person's sensory functions cannot, thereby compensating for the amount of human intervention in automated driving.

The functionality and algorithms needed to improve unmanned driving are tested by researchers in this field both physically and virtually, using a real or virtual environment. Each environment has its own positive and negative characteristics. For example, the physical environment may be limited in its ability to reproduce specific emergencies, and the virtual environment cannot guarantee the achievement of full realism when compared to the real environment. At the same time, the physical environment can provide unique results and consolidate the results obtained when improving algorithms, and the virtual environment can help save money when testing algorithms and find problems before releasing the product.

Looking at that description, we can observe why we use a virtual environment for the scope of work described in this thesis. The initial idea of improving human or customer engagement while using a vehicle and at the same time reaching not only usefulness but safety as well—that is when

ego-vehicle testing appeared in our laboratory. The human interest in autonomous vehicles can be tracked from the early 16th century, from the three-wheeled automated cart designed by Da Vinci to the 20th century, when the concepts of autonomous cars started to form into what we define as ego-vehicles today. The history after has a lot of achievements in terms of autonomous vehicle concepts and components.

At our lab at the University of Waterloo, we are researching two domains, the driving vehicle's human behaviour and investigating ego-vehicle testing in the virtual environment. Our work will describe the latter by examining the differences between the usual approaches to testing vehicles and the adversarial approach. Adversarial testing will test an ego vehicle by stress-testing it with generated accidental situations on the road. The ego vehicle will be added as a blackbox where the other cars won't have information about it. Each vehicle will have an algorithm or policy optimization to determine the training outcome for a vehicle in the environment using the Reinforcement learning technique. The thesis aims to expand the ability to apply adversarial stress testing, improve the domains that can be compared and improve connection to the real world and generation of real types.

In this work, testing of optimization policies for an autonomous vehicle is performed under conditions of constructing and generating scenarios in a virtual environment. The selected `Highway-Env` environment [4], as well as the simulation method of adversarial testing [5] by generating emergencies by vehicles using a trained optimization policy in relation to an autonomous vehicle with given parameters, will be able to provide unique emergency situations for their processing when improving the ego-vehicle algorithm.

1.1 Summary of Contribution

We make the following contributions:

- Initially, we test the concept of adversarial ego-vehicle stress-testing on a broader range of roads besides highways, such as merges, exits, intersections, roundabouts, races, etc. By extending the applicable testing environments, we increase the range of settings (environment components such as roads) by which an algorithm or optimization policy can be compared, adding the ability to evaluate the performance of a particular algorithm.
- In addition, we use other optimization policies proposed in [4] to capture and compare their performance to the simulations proposed by [5]. Additionally, we added other widely used

algorithms for automotive simulations found in the research phase and adapted them to our environment and simulation methods.

- The last step is to search for realism in the generated cases by examining the collected data on accidents occurring on the roads. At this stage, a dataset of accidents was studied, the insights of which were compared with the results obtained during the generation and training of the simulation environment.

1.2 Organization of Thesis

In Chapter 2, we provide background information, including an overview of the definitions of autonomous vehicles and their characteristics, as well as real-world cases of vehicle crashes. Chapter 3 delves into the virtual environment used for the study as well as describing related works to expand its capabilities. Chapter 4 offers a detailed description of our contributions and algorithmic approaches used to achieve the purpose of the thesis. Finally, Chapter 5 highlights a summary of our contributions and experimental findings, as well as outlining prospective future study areas.

Chapter 2

Background

In this chapter, we will explore the prerequisites of the autonomous vehicle domain, explaining the level of autonomous vehicles, existing approaches to testing ego-vehicle systems, and providing a statistical overview of car accidents.

2.1 Definitions

2.1.1 Automation Levels

Self-driving cars are divided into several categories by the international organizations, like *Society of Automotive Engineers* [6], according to the level of automation and the need for human intervention (see the figure below). Level 0 does not include automation, levels 1 and 2 provide assisted automation with less need for human attention with each increase in level, levels 1-2 are called advanced driver assistance systems (ADAS), levels 3-5 are called automated driving systems (ADS) implies that the user does not drive the car. In contrast, automated functions are activated, and also with increasing level from 3 to 5 (conditional, high, and full system automation) the need to take control of the car decrease [6], [3].



SAE J3016™ LEVELS OF DRIVING AUTOMATION™

Learn more here: [sae.org/standards/content/j3016_202104](https://www.sae.org/standards/content/j3016_202104)

Copyright © 2021 SAE International. The summary table may be freely copied and distributed AS-IS provided that SAE International is acknowledged as the source of the content.

	SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in “the driver’s seat”		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	

Copyright © 2021 SAE International.

	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering OR • adaptive cruise control	• lane centering AND • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 2.1: Levels of driving automation, starting from 0 automated vehicles with support for the driver features up to the 5th level of fully automated vehicles, have been theoretically identified but not achieved yet [6]

While driving, a person uses multiple sensory sensations mentioned in [7]:

- cognitive (for example, perception, sensation, attention, thinking, imagination, speech, and memory),
- emotional (feelings, emotions, stress and affects), and
- volitional (decision making, conflict of motives, and goal setting).

However, this does not eliminate the possibility of errors on the driver's part or the presence of blind spots when driving.

Using a self-driving car's technological capability provides the opportunity to supplement and compensate for the lack of environmental information even at the first levels of automation. To

achieve this, automated vehicles use sensors, which increase the possibility of responding to accidents [8], [9]:

- exteroceptive (e.g., camera, stereo, light detection and ranging, radio detection and ranging, ultrasonic),
- proprioceptive (e.g., global navigation satellite system and internal measurement units, wheel odometry) sensor.

This, as a result, can increase road safety for the driver and others by compensating for blind spots and improving localization (the ability to perceive space and relate to it, understanding one's location) through the use of sensors.

2.1.2 Autonomous Vehicle Testing Environments

Despite the ability and desire to test automated machines in the real world, recreating specific events is a rather challenging task. In addition, government and local approval may be required, as well as regulations and approval from insurance agencies. Therefore, for testing a specific scenario, a virtual environment is preferable.

In addition, the system (virtual environment) can potentially become a component in solving the problem of self-driving cars in the area of navigation (such as faded or missing lane markings, inconsistent traffic signals and road works can confuse the car's sensors and lead to accidents, tunnels, or bridges, etc.). As described in [10], [11] by determining the location of the car in space, as well as the conditional distribution of the road according to road standards (according to the size, road width according to standards).

There are many virtual environments created for testing automated cars, among them there are both simplified and complex types, with more detailed graphics and descriptions of the system (for example, CARLA [12], highway environment from Gymnasium [13], etc.).

For the most part, when testing an autonomous vehicle, it is assumed that the ego vehicle itself is the cause of an accident or emergency. On the contrary, stress testing of an attack system (see Glossary) examines the behavior and decision-making of a self-driving car in conditions of an accident caused by external factors (attacker cars).

To characterize the impact on the occurrence of an accident on the road, indicators are included in the data of accident occurrence and may be included as parameters in some virtual environments such as:

- terrain (the exact location or average as if the accident occurred in an urban, rural area, or highway),
- time period (whether it is morning, day, or night),
- weather conditions (snow, rain, clear weather, etc.), and
- other indicators, such as the type of car and its characteristic components.

There are also a couple of techniques that are usually used for ego-vehicle training in virtual environments. From those techniques we can highlight Long Short-Term Memory (LSTM) [14], [15], [16], Cerebellar Model Articulation Controller (CMAC) [17], [18], [19], Decision Forest Regression [20], K-Nearest Neighbor (KNN) [21], [22], Bayes Decision Rule [23], [24], Recurrent Neural Networks (RNN) [25], [26], Deep Convolutional Neural Networks (CNN) [27], Support Vector Machines (SVM) [28], Neural Network Regression [29], Reinforcement Learning [30], [31] which are the most popular ones.

2.1.2.1 Stress Testing as a Technique for Vehicle Testing

Stress testing evaluates the performance and reliability of a component or system under extreme conditions. Automotive stress testing evaluates components, performance, and behaviour under conditions beyond their normal, predictable state. This is achieved by increasing external factors such as traffic on the road, environmental conditions such as weather (rain, snow, etc.), and internal factors such as increased speed or system load.

Normal testing of the vehicle and its parts dates back even before the appearance of autonomous vehicles as part of parts and mechanism testing. Normal testing mentioned in [32] is generalized as the system's functionality under anticipated operational conditions. Stress testing is a larger term that enforces and pushes systems beyond their normal boundaries to uncover possible faults and understand their behaviour under unfavourable conditions.

Stress testing as a component for autonomous driving systems testing described in [33] suggests that the stress testing mechanism enhances issue detection even beyond the adaptive cruise control,

automatic emergency braking, and lane keeping assist, which are the components of the autonomous driving systems, but also in terms of the response to surrounding elements, timing delays and network failures. In their work, the authors of [33] created an environment based on the statistical database of accident data from the Austrian motorways to identify the common action types that imply accident occurrence. Then, using the identified most frequent accident types in [33], they built a testing environment.

In the thesis, we will use stress testing to identify the failure of the target vehicle system by introducing outer factors that drive the vehicle's black box system to its limit and cause accidents.

Examples of the vehicle system stress-testing for its decision making mentioned in [34], and exploration of stress-testing of the vehicle perception, LIDAR system, mentioned in [35]. Another example provides an overview of adaptive stress-testing of the autonomous vehicle introducing a reward function applied towards simulated environment in [36].

2.1.2.2 Adversarial Approach for Vehicle Testing

An approach that we mentioned to stress test a target vehicle will include aggressive behaviour of the attacker vehicles to force the target vehicle to malfunction and lead to a collision, named the adversarial attack. The baseline and aim of the adversarial approach applied to the vehicle consist of finding weaknesses in the car's system control and decision-making by identifying challenging scenarios. The adversarial attacks method is mainly adopted from the machine learning adversarial approach that in comparison to the standard testing approaches provides and identifies a broader scope of the failure scenarios in the system.

The algorithmic steps for this approach that will closely match the setting of the adversarial agentic algorithmic approach for this thesis can include adversarial agent identification, goal setting for the agent, utilizing optimization techniques for episode generation, performance evaluation by introduction of the parameters and boundaries, result refinement using the reinforcement learning approach of the agent training.

The approach's benefits include unique failure identification, improvement of the trained entity, and highlighting vulnerabilities and limitations. However, the drawback may be the unrealistic scenario generation, which may be complex in terms of testing, the parameters of the environment, and agent boundary identification, resulting in a computationally intensive training process.

Adversarial approaches for vehicle testing have shown successful malfunction identification in terms of dangerous maneuver detection in [37], generating safety-critical scenarios in [38], and Improving the Robustness of Multi-agent Autonomous Driving Policies in [39].

Similarly to those examples, we are going to create collision scenarios by training vehicles using single-agent and multi-agent policies to the adversarial approach of the target vehicle malfunction detection.

2.2 Overview of Real-World Accidents

Various sources identify several types of roads, such as rural, and urban, which in most cases reduces the variation and the possibility of considering safety according to road types outside this spectrum. After all, accidents often occur due to the complexity of the road and the number of actors (cars) on the road.

Based on [40] the percentage of deaths on rural types of roads in contrast to urban is 41% and 59%, respectively. Information on the number of traffic accidents from 1977 to 2022 can be seen in the figure presented below.

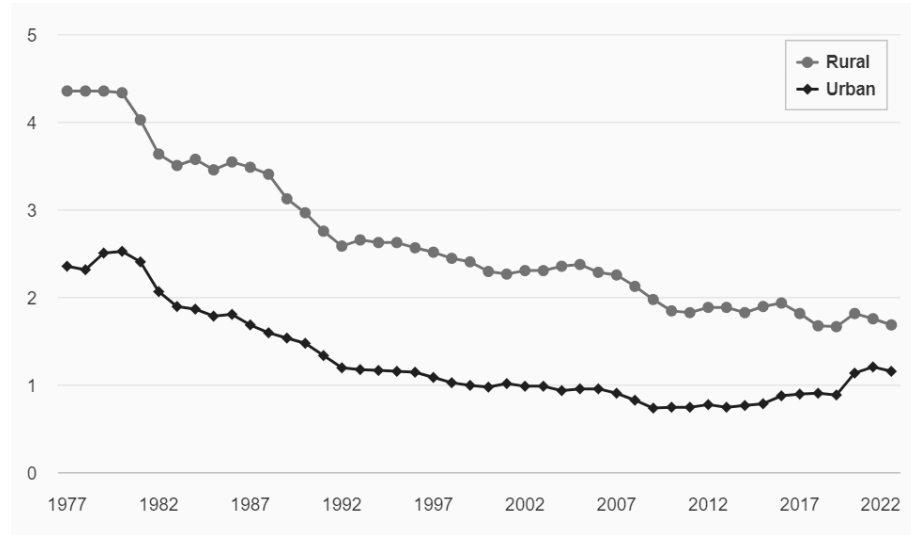


Figure 2.2: Motor vehicle crash deaths per 100 million miles (about 160934400 km) traveled by land use in the U.S. [40]

For a more detailed analysis of traffic crashes, data is often collected through formulated surveys, as well as publicly available data from insurance companies, law firms, and less readily available narratives from police departments around the world.

According to surveys conducted in Europe [41], the results show that there may be some dependence on the age category and type of roads. Without taking into account publicly available information about the influence of the number of passengers and their age spectrum when creating an emergency. As described, for example, in driving tests (one of the questions on the Canadian G1 test is an example), people 16 years of age tend to be disorganized when driving with other people of the same age as passengers.

Based on information provided by the World Health Organization [2], for children and young people aged 5 to 29, automotive accidents are the leading cause of death. People aged 18 to 59 account for two-thirds of all road fatalities. Men are three times as likely as women to die in road accidents.

The Insurance Institute for Highway Safety in [40] provided information that people over 50 years old tend to get into an accident on an intersection road type. The figure below compares the number of accidents that occurred on the road in different countries.

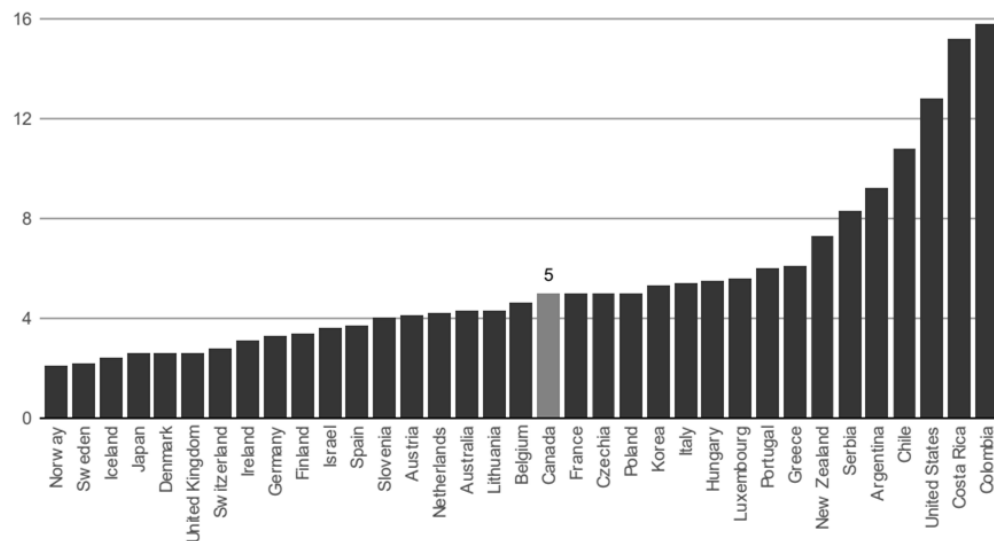


Figure 2.3: Road fatalities per 100 000 inhabitants in Canada, 2022 [43]

The images below from [44], [45] provide a comparison between the number of collisions on the road by a projected vehicle with level 2 automation and conventional means within the U.S., 2019.

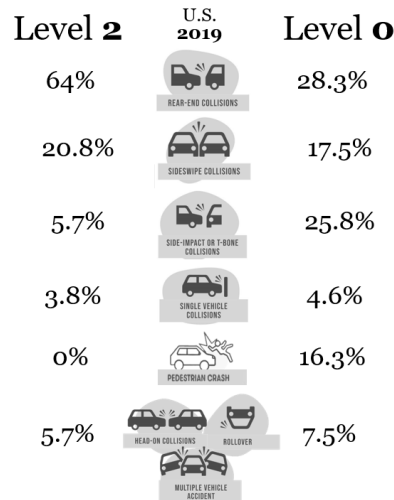


Figure 2.4: Autonomous vs. Conventional Vehicle Crashes using data from [44], starting from the lefthand side statistic provided for autonomous vehicle crashes versus the conventional vehicle crashes on the righthand side divided into the categories starting from the top to the rear-end, side-swipe, broadside, collision with an object, pedestrian and “other” accidents

For analysis, a dataset was selected that had information on the number of accidents involving both conventional and autonomous cars. Using existing data along the route of the accident, you can approximately determine the location and type of road on which the accident occurred. This is how the creation of a new category was carried out - the type of roads, by mapping the location of the origin of the collision.

Deepening the types of traffic studies and describing new characteristics can contribute to adding realism to the projected virtual environment.

A more detailed description of the created cases can lead to the addition of realism to the virtual environment and potentially reveal a pattern. For example, you can project types of collisions or explore possible algorithms (steps) for their occurrence. The figure below describes the most dangerous types of collisions, and the type of damage caused (death, non-fatal injuries, fatal collisions) [46], [47].

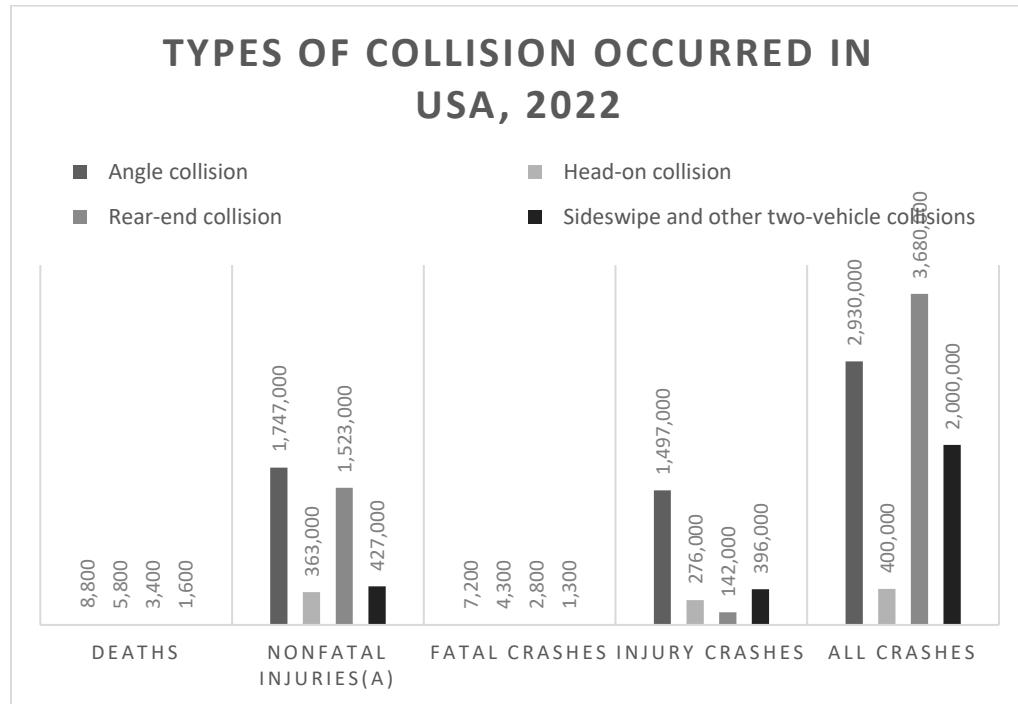


Figure 2.5: Types of collision occurred in the U.S., 2022 using data from [46], [47]

Depending on the type of collision, the following types of collisions are distinguished (give more explanations per each or a brief description):

- Head-on or front-end (or frontal impact) collision,
- Broadside (also called or T-bone, side-impact accident, cross-traffic accidents, lateral impact),
- Rear-end (or Rear Impact) collision,
- Sideswipe impact collision,
- Single-vehicle collision, object collision, rotational impact, and
- Rollover.

2.3 Chapter Summary

Many resources do not consider road type to be the deciding factor in a car accident. In our research, we will mainly consider those to be distinct characteristics for accident generation and flaw

identification in the target vehicle system that has malfunctioned due to the adversarial attack from the attacker vehicle.

The next chapter includes further investigations and environment definitions for which episodes were trained. Sections 3.1 and 3.3 show each road type of the default environment and its adapted versions for adversarial attack generations.

Chapter 3

Virtual Environment

3.1 Virtual environment used for This Study

`Highway-env` is a python library consisting of a collection of environments for simulating automobile interaction on diverse types of roads. This work was published in 2018 and is supported by its creator, Edouard Leurent, and is named “An Environment for Autonomous Driving Decision-Making” [4].

This collection presents several types of roads, which will be described in the following subsections, including a straight highway, merge, intersection, racetrack, roundabout, and parking. For each of these, the authors demonstrate basic solutions for optimization policies which are visualized and tested, such as Deep Deterministic Policy Gradient, Value Iteration for Markov Decision Process, Monte-Carlo Tree Search, and Deep Q-Network and Proximal Policy Optimization from the Stable Baselines3 library.

The Farhama Foundation [48] maintains the gymnasium library [49], the most widely used, open-source library of Reinforcement Learning environments for Python. The API of the gymnasium library [49] is actively used in `Highway-Env` as it utilizes a reinforcement learning approach of a loop of observing the state, taking actions, and receiving the reward or cost according to the rules of the changes made using the action by the agent.

The `Highway-env` project itself is an incubating project [13] that is actively developed and debugged, so it is great for creating innovative approaches or testing different optimization policies. The project consists of the following components:

- observation module with several parameters
- kinematics for describing the vehicle's motion
- vehicle actions, which can be Continuous Actions, Discrete Actions, Discrete Meta-Actions, Manual control dynamics aimed at roads and vehicles
- rewards used in all environments as well as goal specific one (like exiting merge-lane, etc.)
- graphics of road and vehicles in the code
- multi-agent setting such as increasing controlled vehicles and other configurations.

3.1.1 Discrete Meta-Action Space

The following figure shows the five actions that we use in our work. They can be divided into longitudinal and lateral control actions. Longitudinal control includes actions that affect the speed of the vehicle, such as accelerating or decelerating. Lateral control involve changing lanes using left or right turns. Idle action means the preservation of the current state.

```
ACTIONS_ALL = {  
    0: 'LANE_LEFT',  
    1: 'IDLE',  
    2: 'LANE_RIGHT',  
    3: 'FASTER',  
    4: 'SLOWER'  
}
```

Figure 3.1: Action space of Highway-Env

To perform these actions, the vehicle collects data about the environment using Lidar, Radar, Cameras. By default, some of these operations, such as lane changes at road boundaries or accelerating/decelerating beyond maximum/minimum velocity, may not always be accessible.

3.1.2 Vehicle Types

Highway-Env offers two types of transport - a car and a bicycle. In our case, we are interested in the car as a type of transport. The following images represent the characteristics of the car depending on the function it performs. Figure 3.2 shows an example of the visual distribution of cars. The agent is represented by the white box, aggressive drivers by the yellow box, typical drivers by the blue box, and conservative drivers by the green box [50]. The red colour occurs when a vehicle collision occurs, which can be with other vehicles or objects.

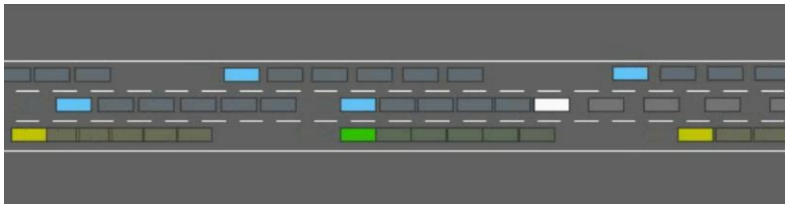


Figure 3.2: Road actors' representation: vehicles represented in a different coloring according to their characteristics. Each vehicle has a trajectory that is represented by faded colored rectangles.

Figure 3.3 represent the last two of the four existing classes: IDMVehicle, LinearVehicle, AggressiveVehicle, DefensiveVehicle.

```

class AggressiveVehicle(LinearVehicle):
    LANE_CHANGE_MIN_ACC_GAIN = 1.0 # [m/s2]
    MERGE_ACC_GAIN = 0.8
    MERGE_VEL_RATIO = 0.75
    MERGE_TARGET_VEL = 30

    ACCELERATION_PARAMETERS = [
        MERGE_ACC_GAIN / ((1 - MERGE_VEL_RATIO) * MERGE_TARGET_VEL),
        MERGE_ACC_GAIN / (MERGE_VEL_RATIO * MERGE_TARGET_VEL),
        0.5,
    ]

class DefensiveVehicle(LinearVehicle):
    LANE_CHANGE_MIN_ACC_GAIN = 1.0 # [m/s2]
    MERGE_ACC_GAIN = 1.2
    MERGE_VEL_RATIO = 0.75
    MERGE_TARGET_VEL = 30

    ACCELERATION_PARAMETERS = [
        MERGE_ACC_GAIN / ((1 - MERGE_VEL_RATIO) * MERGE_TARGET_VEL),
        MERGE_ACC_GAIN / (MERGE_VEL_RATIO * MERGE_TARGET_VEL),
        2.0,
    ]

```

Figure 3.3: Aggressive and Defensive Vehicles parameters within a virtual environment

From the image, we can observe a difference in the calculation of acceleration parameters and the merge velocity ratio between the two.

3.1.3 Road Types

In Highway-env, the main commonly used types of existing roads are racetrack_env, parking_env, roundabout_env, highway_env, and intersection_env.

There are also additional types that are not highlighted in the description of the virtual environment, but exit_env, lane_keeping_env, and two_way_env are also relevant for research.

By default, road creation uses several modifiable characteristics. For example, depending on the location, as well as road signs, the number of road lines, the number of cars and speed limits can be modified to generate a result that is closer to reality (most similar to the real indicators of the environment).

3.1.3.1 Highway

Highway road type occupies a leading position in the number of accidents and collisions on the road as described in the previous chapter. The highway driving environment imitates a car navigating a straight, multi-lane highway. Characteristics of this road type are the one direction move of the actors of the roads. The speed can be 20 to 30 m/s amidst varied traffic. Additionally, vehicles are being rewarded for maintaining high speed or preserving the rightmost position. The highway features four lanes, each with up to 50 cars, one of which the user controls. Also, different number of lanes can be adjusted for creation.

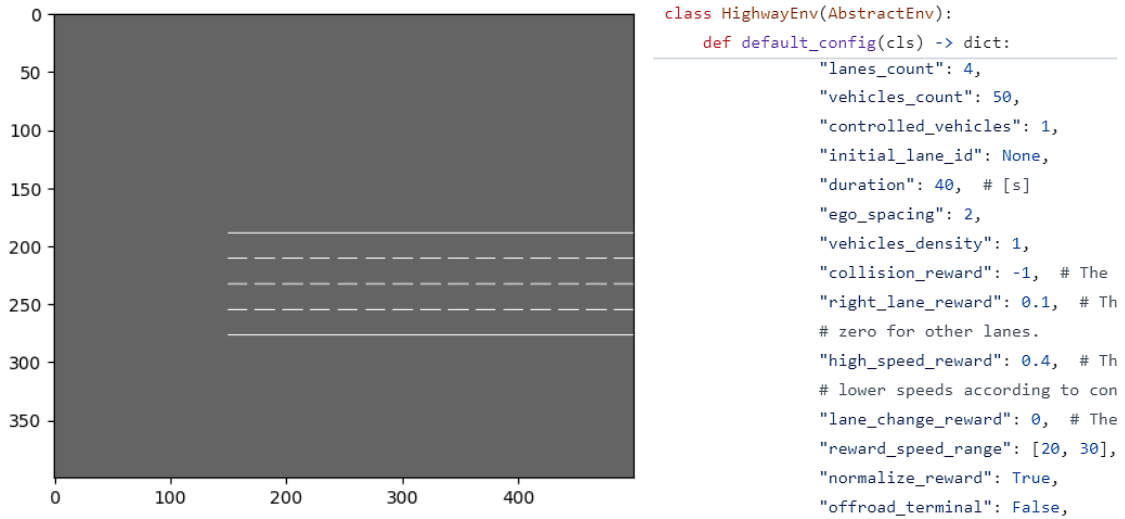


Figure 3.4: Highway environment road visualization

The controlled vehicle begins in a random lane and attempts to advance while avoiding collisions, as accidents may result in a negative reward and may end the episode. The simulation focuses on maintaining lane control and speed management.

The most common possible type of collision on this type of road in the real world (as discussed in Section 2.3) are:

- rear-end collisions that are common in high-traffic areas or when vehicles abruptly slow down, leading following vehicles to collide with those in front, and
- sideswipe impact collisions that can occur during lane changes, particularly in blind zones, or when merging from on-ramps.

Additionally, a single-car collision occurs when a motorist loses control of his or her vehicle, maybe due to exhaustion, distraction, or bad weather, and collides with a guardrail, median, or another roadside object. As well as rollovers can occur at high speeds, especially with higher vehicles such as SUVs or trucks, and are frequently caused by unexpected direction changes or off the roadway.

3.1.3.2 U-turn

A defined U-turn zone on a divided highway or boulevard, usually marked with signs and occasionally protected by a dedicated turn lane or median opening. This permits vehicles to reverse directions and return the way they came safely.

In most cases, this type of road is not identified as a separate type. U-turn opportunities are often incorporated into road designs for traffic management.

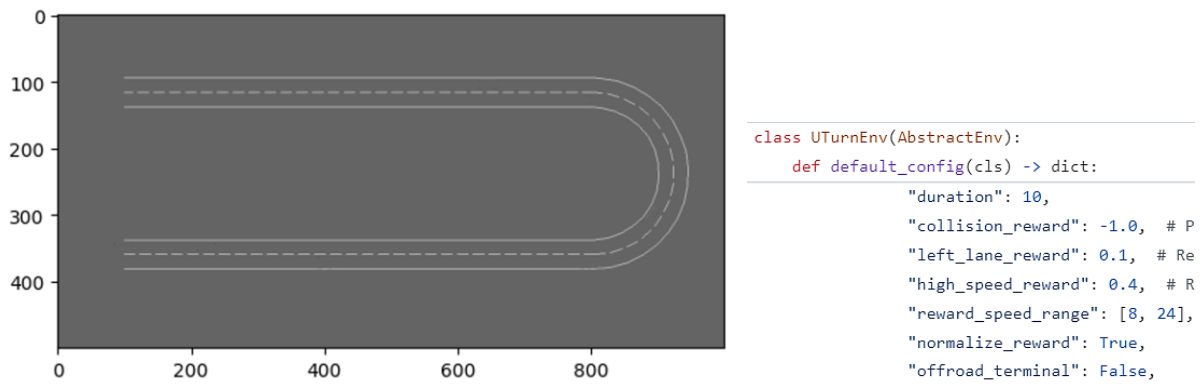


Figure 3.5: U-turn environment road visualization

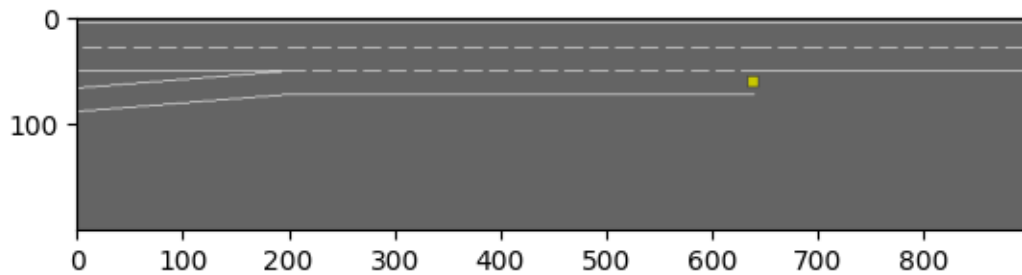
Actors (or simply vehicles as mentioned in the Glossary) in the environment are positioned to test the agent's anticipatory and collision-avoidance skills by requiring dangerous overtaking manoeuvres or impeding the agent's progress. Rewarding behaviours for the agent include staying in the leftmost lane, avoiding crashes, and keeping speed within a given range 8, 24 m/s.

Possible types of collision on this type of road can be:

- if a U-turn is not properly timed or sight is limited, incoming traffic may strike the vehicle on the side - broadside.
- rear-end collision – vehicle slowing or halting to make a U-turn may be hit from behind by another vehicle that fails to slow down in time.
- during a U-turn, a car may sideswipe another vehicle in an adjacent lane – sideswipe.
- a single-vehicle crash can occur when a driver loses control while making a U-turn and collides with a median, guardrail, or another roadside object.

3.1.3.3 Merge

Merging is a common feature on highways and freeways, allowing vehicles to enter and leave high-speed roads. A part of a road where two lanes merge into one is commonly encountered when an entry ramp connects to a highway or where a route narrows. Vehicles in the merging lane must change their speed and location to safely enter the main lane traffic flow.



The road's layout combines straight and curved lanes, along with a dedicated merging lane that progressively approaches the main highway. The environment default parameters penalize the ego-vehicle for making needless lane changes and failing to yield to merging traffic, and rewards it for maintaining a high speed, especially while driving in the right lane and avoiding crashes.

A possible type of collision on this type of road can be a collision with:

- an object – if a merge lane has no smooth connection into the highway.
- sideswipe impact collisions are the most prevalent form of accident in merging regions. It can happen when a merging vehicle fails to yield to main lane traffic, or when a car in the main lane leaves insufficient room for a merging vehicle, resulting in a side-by-side accident.
- rear-end (or rear impact) collisions occur when a car in the merging lane slows or stops abruptly, forcing the vehicle behind it to hit its rear. It can also happen in the main lane when a motorist suddenly slows down to enable a merging car to enter.

3.1.3.4 Exit

The exit road environment is a highway simulation used to evaluate vehicle behaviour on a multi-lane road featuring an exit ramp, depicted in figure below. The default configuration consists of a 6-lane straight highway where automobiles can traverse, with the option of using an exit lane. The

simulation includes 15 observable vehicles, including one controlled vehicle that may conduct discrete actions such as altering speed to one of three goal values 18, 24, or 30 m/s.

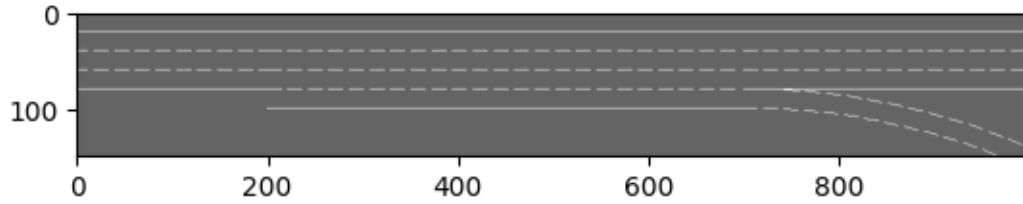


Figure 3.6: Exit environment road visualization

The default environment rewards the controlled vehicle for travelling at high speeds and completing the task while penalizing crashes, with rewards normalized to a range of 0 to 1. The environment is set at a vehicle density of 1.5, automobiles are quite close together, simulating a busy highway.

Possible types of collision on this type of road can be rear-end collision and swipe impact.

3.1.3.5 Two Lane – two-way

These are the most common types of roads, forming the backbone of most road networks. The agent must manoeuvre a two-way road with oncoming traffic while simulating a difficult risk management situation in the two-way road environment.

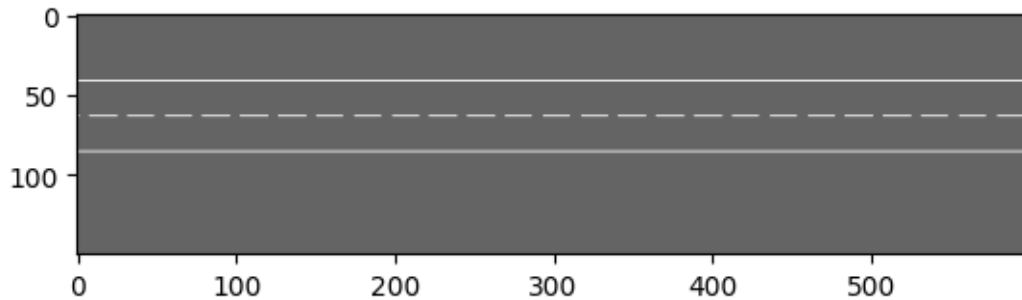


Figure 3.7: Two-way environment road visualization

A realistic highway scene is created by the environment's two lanes in each direction, continuous and striped lane markers. The agent's goal, which is controlled by a reward mechanism, is to maintain safety while balancing advancement by passing cars. With incentives designed especially for high-

speed travel and left-lane usage, the agent is rewarded for sustaining high speeds, especially when operating in the left lane.

On a two-way road, the most fatal accidents occur with head-on or side-swipe collisions. That type of road has the possibility of generating all types of collisions.

3.1.3.6 Roundabout

Increasingly popular in many countries, including the US, as an alternative to traditional intersections. The Roundabout road environment attempts to mimic how cars would behave when traversing a roundabout with several entry and exit locations. The setting includes a network of circular roads that are centred on a 20-meter radius. The lanes are delineated by lane markers that are continuous, striped, or sine wave

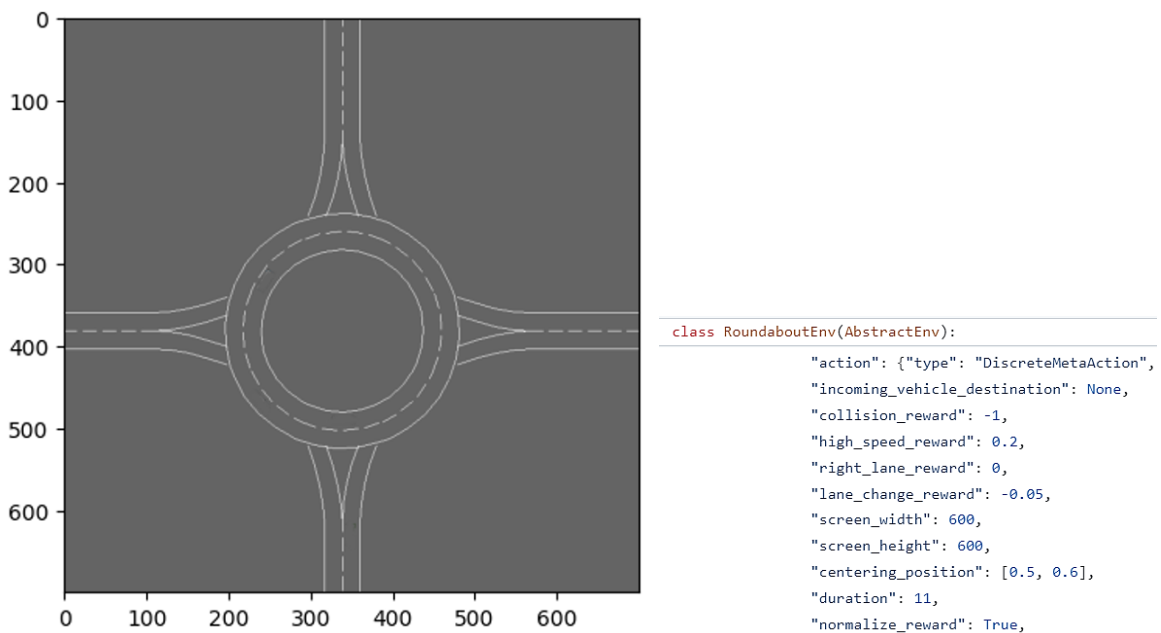


Figure 3.8: Roundabout environment road visualization

Vehicles approach the roundabout from a variety of entry routes. Lane changes, collisions, and off-road driving are all penalized by the environment, which encourages fast driving.

There are several types of collisions to expect due to the complexity of this type of road. Rear-end crashes are a usual form of collision in which two cars travelling too closely behind one another may collide if the lead vehicle abruptly slows down or stops. When cars approaching or leaving the

roundabout fail to yield to traffic already in the roundabout, it can result in side-impact or T-bone incidents at the intersections of the entrance and departure lanes. Furthermore, cars that attempt to manoeuvre inside the roundabout and deviate into adjacent lanes may be involved in sideswipe crashes.

3.1.3.7 Intersection

This type of road is found extensively in urban and suburban areas, intersections are critical for connecting different roads. The 4-way simulated junction environment known as the intersection road environment is intended for scenarios involving autonomous driving and multi-agent interaction. The surrounding area consists of a network of roads with distinct lanes for cars, each of which is given a different priority based on the direction in which it travels (horizontal roads have higher priority).

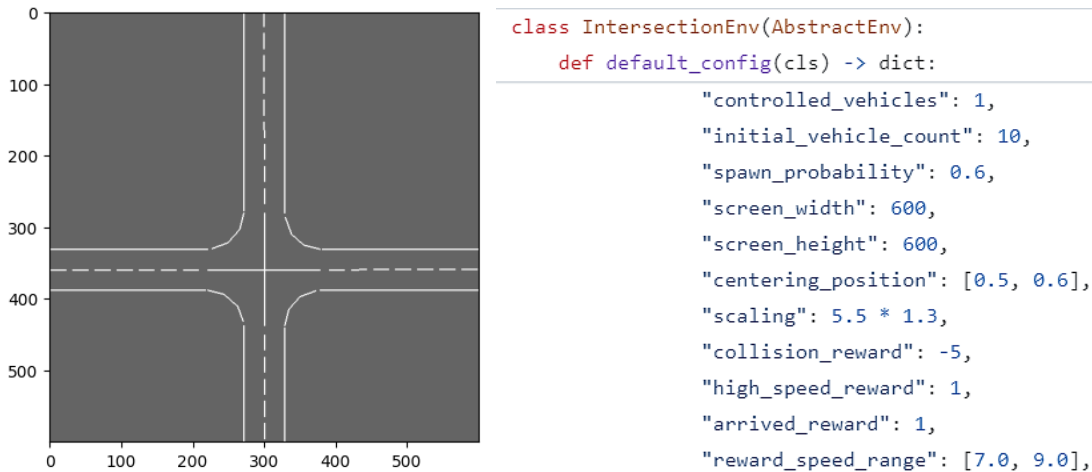


Figure 3.9: Intersection environment road visualization

There are fifteen cars in the simulation suggested by default configurations, one of which is controlled by the user. The action space for this environment allows the agent to alter their vehicle's longitudinal speed (making it slower, idler, or quicker) but disables lateral control by default (in the Section 3.3 we refactored parameters to include lateral control functions). Within a predetermined range, the observation space records the vehicles' kinematic data, such as their locations, velocities, and orientations. The incentive system of the environment penalizes crashes and rewards safe driving, sustaining high speeds, and safely arriving at destinations. Collision types are similar to roundabout.

3.1.3.8 Racetrack

The racetrack is an ongoing control environment created to test an agent's ability to pick up two vital skills: staying on the assigned course and avoiding other cars' crashes. The standard racetrack consists of a straight part at the beginning, then five carefully designed curved portions with different radii and speeds to evaluate the agent's manoeuvrability.

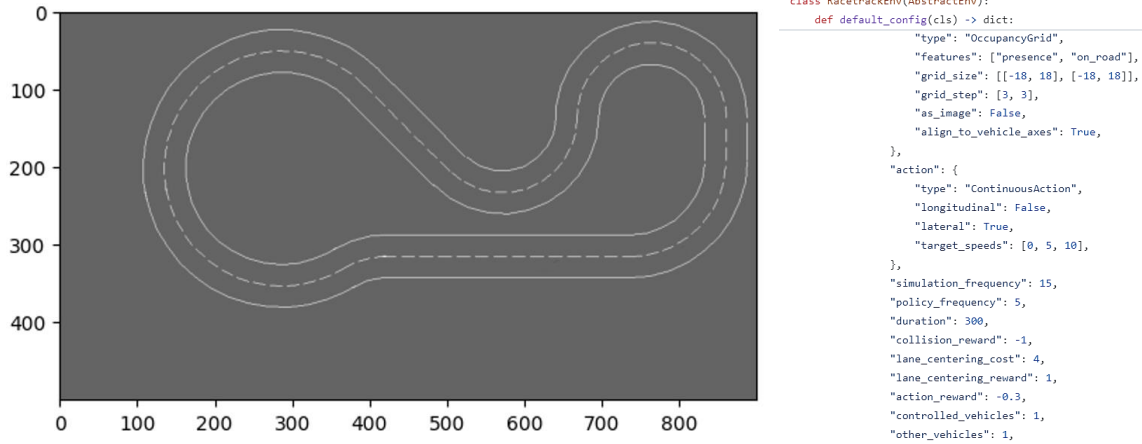


Figure 3.10: Racetrack environment road visualization

The controlled agent vehicle and other IDM-based vehicles that present dynamic impediments are among the vehicles in this environment. The agent's major responsibility is to maintain lane centering and prevent accidents, which carry heavy penalties. Reward points are awarded based on how near the agent stays to the lane centre. The agent controls lateral motions while maintaining longitudinal speed in a continuous action-based environment that employs an occupancy grid for observation. Hence, it is harder to find an optimal policy for this domain. Conversely, it may be easier to cause an adversarial crash.

The most common accidents that occur on racetrack found in the real-world are obstacle collision or side-swipe due to high-speed.

3.2 Simulation Attacker Virtual Environment Definitions

In [51], from 2017, authors presented hostile attacks aimed at interactions between the attacker and victim agents of reinforcement learning on five Atari games. That approach identified “maliciously defined target states” (simplified faults) in trained agents using A3C and DQN algorithms with a success rate of 70% for 3 out of 5 games.

In 2020, the authors of [52] utilized a black-box approach based on the implementation of a humanoid opponent. The black-box approach means that the attacker does not have direct access to the victim's observations. To test a hostile attack and build an adequate response, the authors of [52] proposed several highway environments in which a different number of attacking cars was presented, from 0 to 3, and regular cars. When training victim agents in an environment without an attacking car, rewards were given for following road recommendations without causing emergencies. When introducing an attacking car in a single-agent attacker domain, the victim is subject to receiving a penalty for an emergency, and the attacking car, in contrast, is encouraged if it lures the victim to end up in a crash, which is the victim's responsibility. The testing result showed that one attacking car significantly reduces the safety of an autonomous car, which leads to accidents and incidents. Hence, we are uncovering flaws in autonomous vehicle safety that the vehicle creator should address to improve the model in the generated situations.

In 2023, Zhou [5] compared the above approaches by testing the attacker and victim agents using the adaptation of the CPO, TRPO, HATRPO, and MACPO algorithms (adapting them using the open source [53], [54], [55]) expanding approaches towards multi-agent system adoption for adversarial testing. As in the previous work, the training was performed in a highway environment, where the attacker was rewarded if there was an opportunity to create a collision and had a cost if the situation had too unrealistic an effect or the collision occurred with the attacker agent, not with the victim. Initially, the victim agent had only an ideal state, but because the number of collisions was not significant, the resource proposed two other agents, such as the following victim vulnerable and the back victim vulnerable, to increase the number of accidents. The result in [5] showed the superiority of the MACPO algorithm over others when applied on the highway road type environment.

3.3 Environment Preparation, Comparing Produced Scenarios with Real-World Cases

To familiarize and get a better understanding of the supportive mechanism of autonomous vehicles, like LIDAR, RADAR, etc. The robots from ROBOHUB at University of Waterloo were investigated and actions were taken for localization and moving of the robot as well as mapping and understanding the information gathered by the mechanism into the linear or 2d representations. Refer to Figure 4.1, 4.2.



Figure 3.11: Robohub robot with lidar and camera

Similarly to the mapping of the data received from the autonomous entity, the aim of this project is to analyze the real-world cases in a 2d format or scale. Hence, the highway-attack-env can be a cost-effective solution for testing purposes of accidents generated without the entities being directly impacted (like in 3D testing with robots Robohub). In future works (see next chapter for more details) we could enhance the current adversarial black-box testing solutions to additional 2D and 3D workflow such as this.

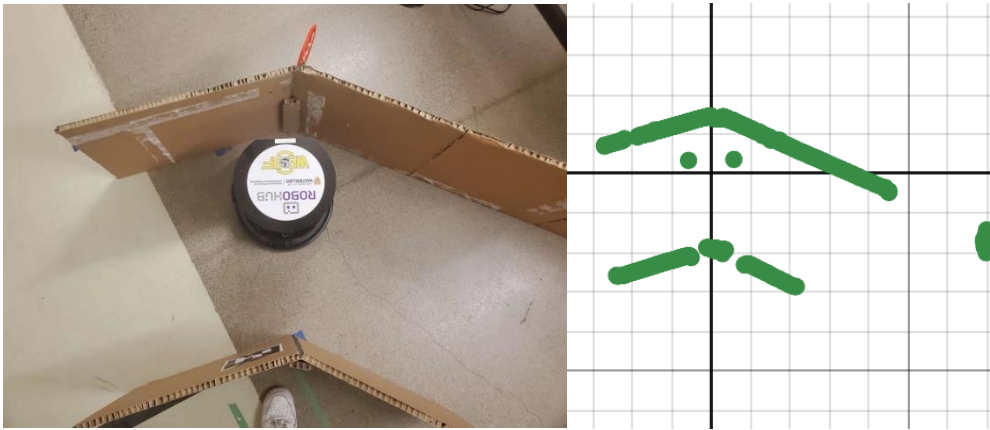


Figure 3.12. Road detection and self-identification using LIDAR

Initial testing of the current algorithms, or updates of outdated libraries to support newer solutions, were made to be compatible with version of 2023 of Highway-Env. The Highway-Env allows us to train the vehicle using the Reinforcement Learning approach.

Given that we are testing the accident generation with an adversarial approach randomized state for the vehicle accident generation, we can generate unique and different accidents to identify states and episodes for flaw detection in the black-box target ego vehicle system.

The vehicle observes the current state of the environment and evaluates what action to take based on the position, and distance according to the lane markings, obstacles and vehicles as well as a rewards function that gives rewards based on the preferred state or the speed defined in the environment.

In the original environment in which we are proceeding with training the adversarial approach the collision-avoiding avoiding, maintaining speed state is encouraged. For our case, we are encouraging the attacker vehicles to create and force collision onto the black-box target vehicle.

In the end, the vehicle learns the shortest available number of steps needed to receive a high reward and minimal cost for the action taken to reach the successful state of the episode—collision.

The first step in reproducing the research outcome completed in our lab was to recreate the training process.

A couple of problems needed to be overcome, such as outdated libraries and system differences, to train the model. At first, we tried different operation systems to see which one would support the outcome better. The CUDA's initial preferences for the device type were compatible with Linux-based systems, but in the end, we switched towards the CPU to run training on the Windows system. Some of the libraries were conflicting and initially adopted a reinforcement learning environment for Highway-Attack-Env, which was adaptable to our specific characteristics.

After adapting each algorithm by creating needed specifications and changing the outdated libraries, we successfully ran each algorithm and compared the initial results with the default parameters of using the Following victim: a vulnerable target car that has a predefined flaw in changing lanes to the left, which generated accidents of vehicles colliding with the leftmost border or left lane surrounding vehicles.

The number of experiments was **200000**, which we were recreating from the initially proposed number of episodes in our lab project. The cost for the attacker vehicle colliding with another attacking vehicle is 5 points (which was deducted from the reward). The reward for colliding with the

target vehicle is 10, enforcing the creation of episodes of collisions with the target vehicle—an adversarial approach.

The next step was adopting each algorithm to change the target vehicle so that it was perfect, with no flaws initially encoded into the vehicle's behaviour. Initially, we had two recreated environments for a highway road type with defined vehicle creation, step, cost, and reward functions, as well as generation specifications for rendering the episodes. The difference between a single-agent and a multi-agent approach was mainly in the collaborative enhancements towards the attacker's behaviour. The reward and cost functions were redefined to be collective.

After successfully adapting the environments, our next steps were to broaden the scope of the episode generation to include different road types. We defined complexity as increasing with the type of road. The next highway to be updated was the exit environment, followed by merge, two-way, lane keeping, intersection, roundabout, and race track.

An unexpected issue was when we initially built the environment in the exit road type. The redefined environment with extension towards an adversarial approach for a random point generation on the circular lane which was connected to the straight lane. After approaching those specifications in the kinematics the road type and vehicle creation on it using a random position was resolved.

In comparison to the initial work conducted in our lab in this project we for the future enhancements we aim to achieve real-world case simulation environment by switching to the ideal or perfect in terms of action scope autonomous target vehicle. Meaning the future work on this project would ideally get access to a real autonomous driving system used in industry as a target for the black-box testing.

3.3.1.1 Exit

In the exit-type environment, we had an initial problem with road specification and random vehicle generation. After resolving the kinematics and adding the reward enforcing for the collision with a target vehicle we adopted the goal identified in the environment for the lane parameter to be included

in the reward function. Two classes were inherited from the original `exit_env` to be adapted for a single-agent and multi-agent adversarial approach.

Initially, we identified a random point for the vehicle generation. Most of the generated accidents had highway-road-type episodes. Fewer of the actual episodes with accidents happened directly in the exit lane. The example generated during the off-ramp episode is provided in the Figure below.



Figure 3.13: Exit Road type generated collision episodes

One example, [56], describes an accident which happened with a speeding truck while exiting the highway. Which led to the rollover of the truck's load and impacted the traffic. Negotiating a curve while exiting can lead to a different type of collision which includes collision with a different number of actors, starting from solo-collision with an object.

Another similar collision with a one actor happened on the exiting ramp which led to the vehicle sliding outside of the road into a ditch [57].

Four actor collision described in [58] referring to collision happening due to losing control over the vehicle and veering out of control on the off ramp.

Speeding may be as well a high-risk leading to accident action. The news article [59] provides footage of two actors side-swipe collision. Similar collision can be observed on Figure 3.13.

3.3.1.2 Intersection

Intersection-environment was expanded towards acceptance of all action identified in the `DiscreteMetaAction` space (specified in Section 3.1.1)

```

class IntersectionEnv(AbstractEnv):
    ACTIONS: Dict[int, str] = {
        0: 'SLOWER',
        1: 'IDLE',
        2: 'FASTER'
    }
    ACTIONS_INDEXES = {v: k for k, v in ACTIONS.items()}

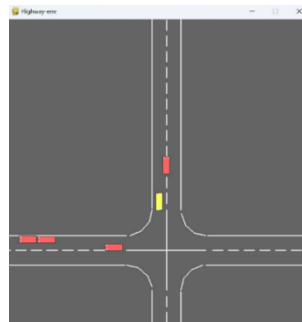
    # "longitudinal": True,
    # "lateral": False,
    # "target_speeds": [0, 4.5, 9]
    },
    "victim_action": {
        "type": "DiscreteMetaAction",
    },
    "attacker_action": {
        "type": "DiscreteMetaAction",
    },

```

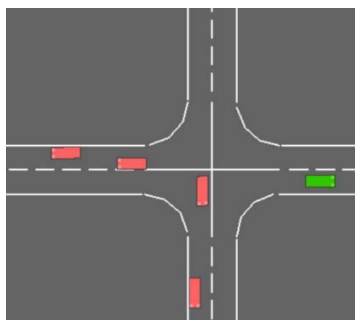
Figure 3.14: Initial and updated parameters of the Intersection environment action space identified accordingly from left to right hand side

We expanded and inherited the class of AbstractEnv and Intersection Road type creation for single-agent and multi-agent algorithms by changing the observation type according to the algorithm. The Figure 3.15 depicts generated examples of intersection-type accidents.

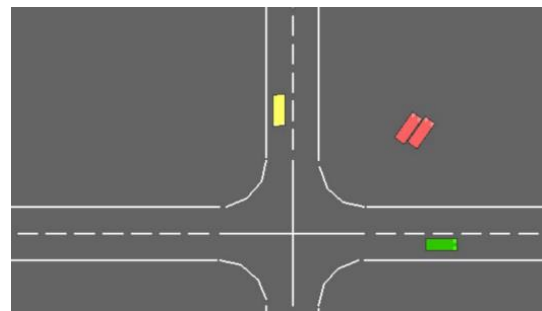
The reward and cost functions were adapted specifically for the agents. The Attacker agent has a reward for a collision with the target vehicle.



Multi-vehicle collisions (two-vehicle collision, wrong change-lanes, or central border override)



Multi-vehicle collision (colliding while intersecting on a high speed)



Over speeding when changing lanes – rollover from the road

Figure 3.15: Intersection Road type generated collision episodes

Real-world case of a front-end collision on the intersection of two vehicles [60]. One of them were going in the direction of the traffic light. Another vehicle, SUV, was speeding for 90 km/h while the red light was on leading to collision with a vehicle and overriding the pedestrians crossing at the time [60]. A couple of similar cases were generated on the intersection road-type leading to the collision with one or more vehicles as shown in Figure 3.15. The generated scenarios holding similar to the news article description – speeding and colliding with the vehicle going into a different direction.

Another case has a similar collision pattern where a vehicle was driving erratically and speeding after the notice from a police officer leading to the collision with another vehicle at the intersection [61].

Three vehicle collisions described in [62], happened due to the speed factor at the intersection. Two vehicles were colliding when travelling in a straight manner and travelling by turning the lane. As a further after effect of two vehicle collision, the remaining inertia of the vehicles led to another collision with the third vehicle.

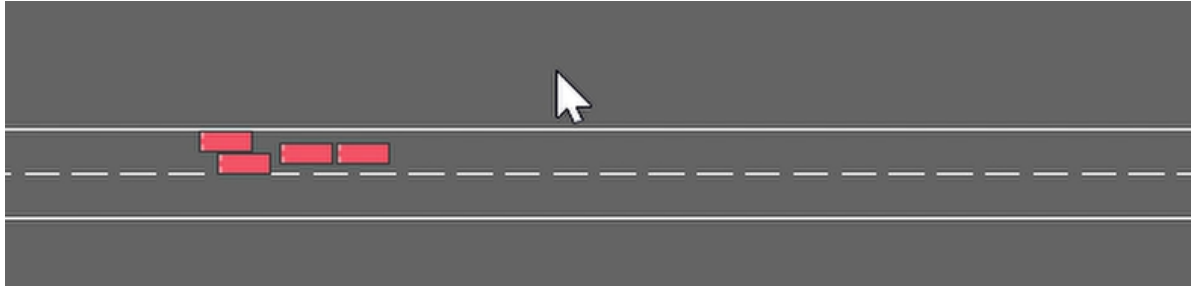
3.3.1.3 Two-way or two-lane

In this environment, we have a two-way direction for generating the vehicle. Initially, we enforced vehicles to be generated equally onto each part of the road lane. In this type of environment, the car is encouraged to stay in the same lane to encourage collision-free initial specifications of the environment.

During the generation process, we introduced different reward functions toward speed maintained, but with the introduction of a reward similar to the other environment reward for target-attacker collision, we did obtain a collision episode. Examples of some can be viewed in the Figure below.

This type of environment can be seen as a supplementary environment, or precisely the behaviour of this environment to be applied to other environment types to add complexity for collision generation per each environment.

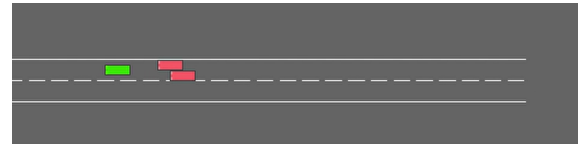
We found three examples of car accidents similar to generated ones on the two-way lane type environment were described in CTV News, these are shown in Figure 4.5



Multi-vehicle collision (domino effect)



Two-vehicle collision (high-speed or tailgating)



Two-vehicle collision (change/return to lane)

Figure 3.16: Two-way generated collision episodes

Head-on collision between a van and a pickup-truck [63]. The collision occurred due to high-speed travelling. A similar collision was generated using a two-actor approach of having a victim and attacker interactions which led to collision. In the scope of the project both vehicles were defined as lightweight vehicles. By generating and training the ego-vehicle on the collision-scenario we can potentially omit the real-accident on the road by performing actions or improving the existing model onto the algorithm of the ego-vehicle.

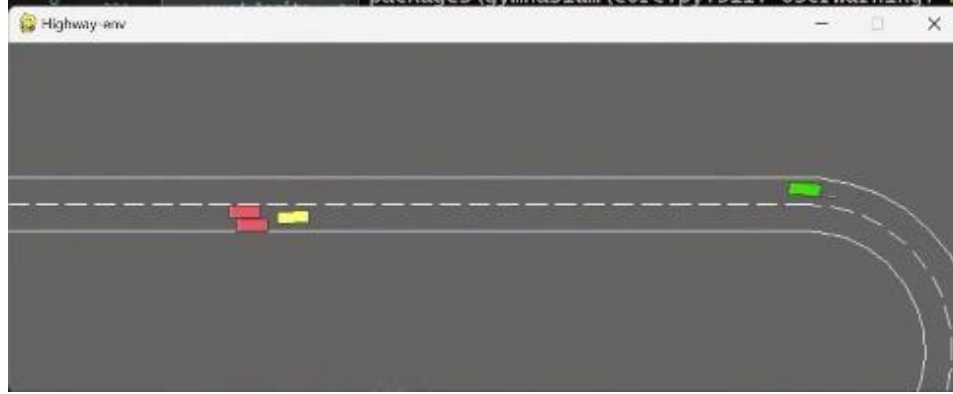
Another similar head-on collision between van and a vehicle which led to the van rollover from the road [64]. The head-on collision included six vehicles in the crash. The SUV crossed the center line on a two-way road and crashed into an incoming truck and the SUV collided with 4 other vehicles. While generating cases on a two-way road environment we also encountered a similar case when crashes of two vehicles resulted in the other actor's involvement [65].

3.3.1.4 U-turn

The U-turn identified road type has the longest duration environment which depends on the road dimensions. Accidents generated on the U-turn road type have the longest time-to-collision time. Initially, when we tried to generalize the termination condition for the episode for a collision to happen the amount of collision generated was generated before accessing the U-turn curve. After

changing the termination parameter, we received a broader range of collisions in a different part of the U-turn.

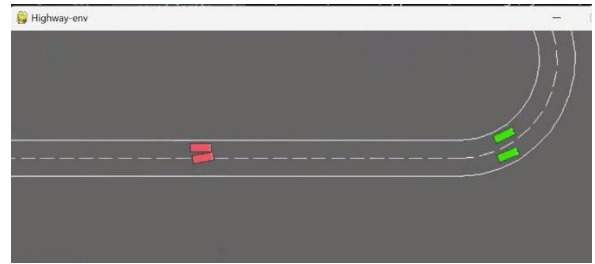
Like other environments, the reward function for the attacker vehicle to execute an adversarial approach is similar. The initial environment was inherited and adapted to the acceptance of single-agent and multi-agent algorithms.



Multi-vehicle collision (collision occurred while changing lanes)



Two-vehicle collision (failed to turn, overspeed)



Two-vehicle collision

Figure 3.17: U-turn generated collision episodes

The accident described in [66] shows that illegal actions such as U-turn made on a curve can lead to the life-threatening collisions, side-swap, or T-type. Similar to that accident, which includes 3 actors were generated on the U-turn road in the virtual environment. Using such an episode the autonomous vehicle system can be taught and improved in terms of actions to be taken to omit the collision.

Another example, in [67], describes a head-on collision of an aggressively driven vehicle which led to a head-on collision with another vehicle. During the simulation, a similar accident which included two actors was generated.

Another real-life example [68], describes a T-bone collision that happened during the night while the first actor performed a U-turn. By recreation or testing vehicles on similar episodes the actions made can be recreated for different investigation and improvement tasks.

Dangerous situations of performing an illegal U-turn described in [69]. Where the vehicles are performing U-turn actions which lead to near-misses and one collision. The action performed by vehicles are “driven” by a human behaviour of trying to cut the road while it is being under construction closure.

3.3.1.5 Merge

For the merge road type, to encourage collisions to happen within the merge or on-ramp lane, we enforced the initial lane for the target vehicle to be generated as a line with ID 0. The figure below represents generated episodes.

Inheriting the original merge environment and expanding it for the single-agent and multi-agent approach we encouraged the attacker vehicle to collide with the target vehicle using a reward introduction similar to other environments. Attacker vehicles in comparison to the target vehicle whether they are learning as a separate entity or as collaborative have a random point of their generation encouraging a variety of colliding episodes to be created during the learning process.

Merge road accidents also can happen with different numbers of actors included in the accident.

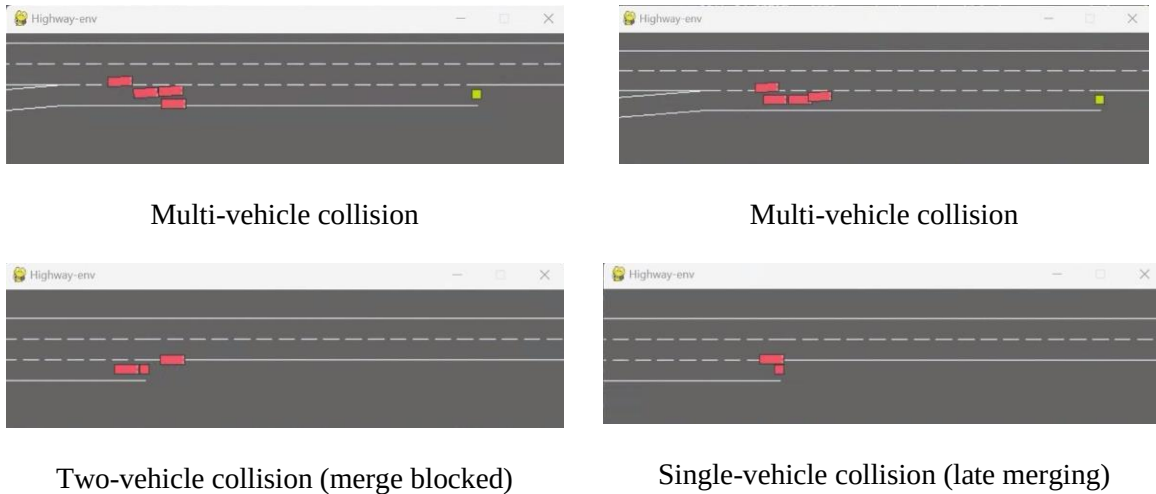


Figure 3.18: Merge generated collision episodes

Article [70] describes two vehicle collision during merge which resulted in serious injuries to the passengers. The actions taken by two vehicles trying to merge into the same lane led to the rear-on collision happening. Similar cases with merging were created during the test on the merge road type supported environment.

Another example [71] describes a driver losing control while accelerating on a ramp which led to collision with the guard rail.

3.4 Realism in a Highway Env

After introducing the virtual environment, named `Highway-env`, we can observe and identify restrictions that may imply the reality or the meaning of the realism we can reach within the virtual environment. First of all, let us note that the police narratives described in the previous section and their comparison to generated episodes from our model may breach the gap of the accident generation, which we can define as feasible or "real" after their comparative analysis. But we also should mention that the realism of the accident itself is constrained by the non-photorealistic 2D setting provided in a `Highway-env` and the lack of 3D components and sensory systems that can be seen in other environments, like CARLA [12]. On the other hand, the `Highway-env` introduces its own parameters that make the environment a scaled representation of what we can observe in the physical world setting; such parameters are scaled measurements of the speed, distance, kinematics, and behaviour of the road and the vehicle chosen. Furthermore, in future enhancements, we identified parameters that can also be added to the virtual environment and enhance its "realism" representation, such as weather conditions, time frame (morning, noon, night), and additional elements like obstacles on the road (pitfalls), vehicle types (truck, streetcar, etc.), road types (narrow road, etc.). Those mentioned are seen as an introduction to kinematics and vehicle behaviour in accordance with the state of the environment under which those conditions are applied. By itself, the environment is a great representation of the 2D settings for training the model and the agent within the constrained environment already introduced in the `Highway-env`.

3.5 Chapter Summary

In this chapter, we explore the chosen virtual environment and its default parameters. Then, we provide a summary of the approaches to the virtual environment given the scope of the stress-testing

methods, like applying an adversarial approach to a black-box ego vehicle system. Next, we identify the parameters that were conducted in our lab for the Highway-attack-environment.

We are expanding the single-agent and multi-agent adversarial simulation approach to the different road types and providing examples of generated accidents within each type. These examples are compared with real-life accident descriptions retrieved from news articles provided for each road type.

.

Chapter 4

Contributions

All road-type created environments for a single-agent and multi-agent approach for the adversarial attack testing described in the previous Chapter will be applied to test various optimization policies to train and determine those that perform effectively for the black box target ego vehicle testing.

In this chapter, we will describe the outcome and generated results using cost, return and cost-return functions to identify the performance of the single-agent or multi-agent optimization policies on the different road types for reaching the optimum value.

The optimization policies of PPO, DQN, A2C, CPO, TRPO, MACPO and MATRPO will be applied as reinforcement learning agents directly and/or as a part of SB3.

4.1 Comparison of Policy Optimization Adoption

For testing purposes, we will generalize single-agent optimization policy and multi-agent optimization policy with recommended parameters described in the Highway-Attack-Env to apply them to different road-type environments. Furthermore, each optimization policy was changed to obtain the parameters for their visualization through the learning process. As additional parameters were created to troubleshoot model performance, we added the kinematics and behaviour parameters to be represented in the log files and road parameters from the road to identify and address any drawbacks of the vehicle and road generation experienced during the first stage of environment creation.

4.1.1 Highway Attack Env

For Highway-Attack-Env mentioned in Section 3.3, which presents two agents for simulating accidents described in the previous section and uses the adapted policy optimization publicly available based on the rl-zoo environment. The advantage of this approach is the availability of algorithms with a similar structure in which you only need to change the agents to the attacker and target. This is described in section 3.2 of the previous work analysis for the MACPO, CPO, TRPO, and MATRPO algorithms. Various approaches can be added as a component towards the default models and changes in their function representations. The types of consideration for top implementations were additions towards the representation of the values collected at the end of the

training process and changes towards the neural network model, activation functions, and component additions like long short-term memory implementations, etc.

We aimed to extend `Highway-Env` to allow training on other types of roads besides highways and to reach the efficient way for the agent to learn and preserve the original aim of creating an adversarial attack on the black box target ego vehicle by enforcing single-agent in a single-agent policy optimization and multi-agent approaches to conduct actions and make decisions in the way of the real-world scope that can be described and obtained insights from. Our goal was to modify the simulation environments and train vehicle control policies.

Additionally, while we initially considered the target vehicle with prebuilt flaws in the systems that were previously suggested for the accident generation in our lab in [5], we retrained and changed those action parameters when the model with flaws achieved similar findings and results.

To conduct the test, we enforced similar where possible reward functions alongside the default for the environment goals and rewards of the speed perseverance and lane preferences. The perfect target as an ego vehicle has a similar ability to recognize its surroundings and spot the forthcoming actions of the surrounding vehicles as initially proposed in our lab. The next Sections will present the plot with the learning outcomes of the different policy approaches to resolve different road-type environments and reach an optimum as a result of the training process.

4.1.1.1 Single-agent centralized optimization policy

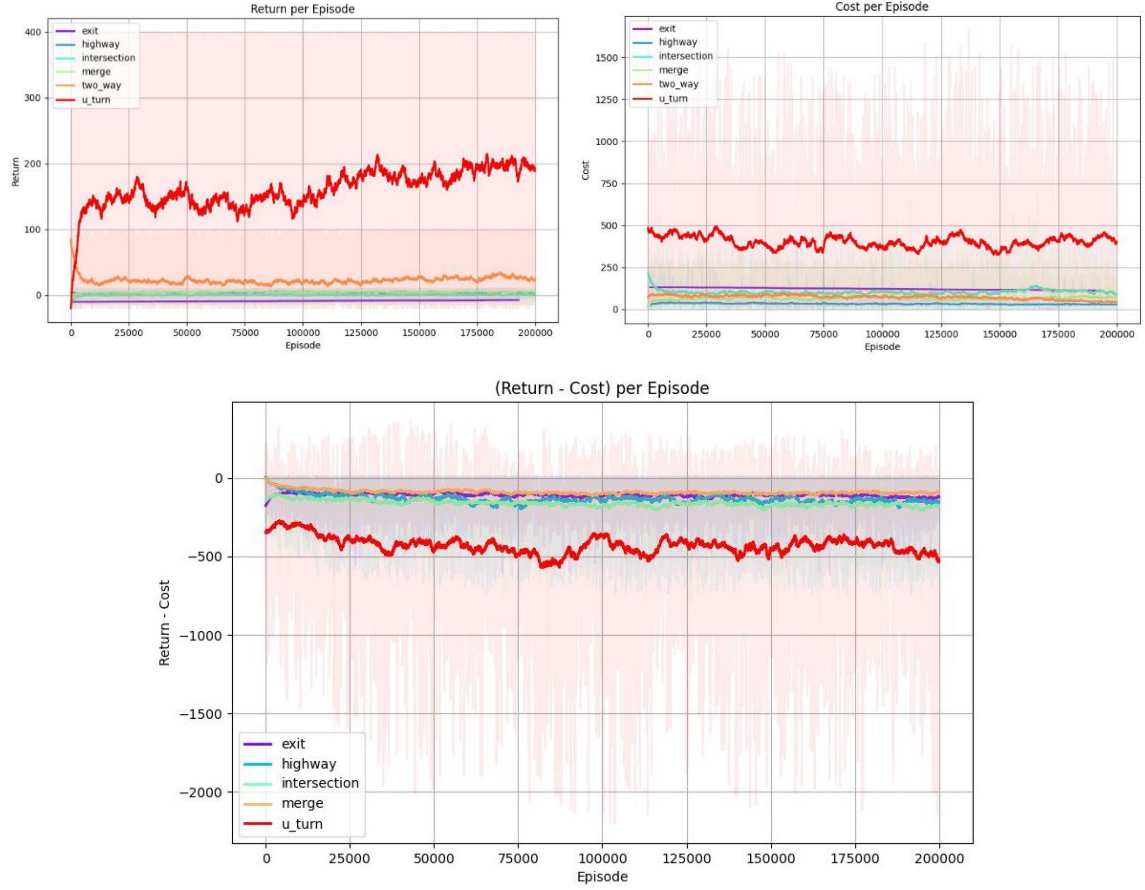


Figure 4.1: Single-agent centralized optimization policy applied on a different road-types

For the exit road type environment, Figure 4.8, we can observe relatively low returns, which may lead to the conclusion that agents have challenges in policy optimization with a given introduced constraints towards accidents generation on the exit lane.

Comparatively to other scenarios, the exit road-type scenario is as simple as the highway road-type scenario, which suggests possibilities of complexity introductions to future training, like introducing higher rewards for exiting goals achieved by all vehicles and restricting the road duration to limit the road of the highway after the exit turn.

4.1.1.2 Trust-region Policy Optimization

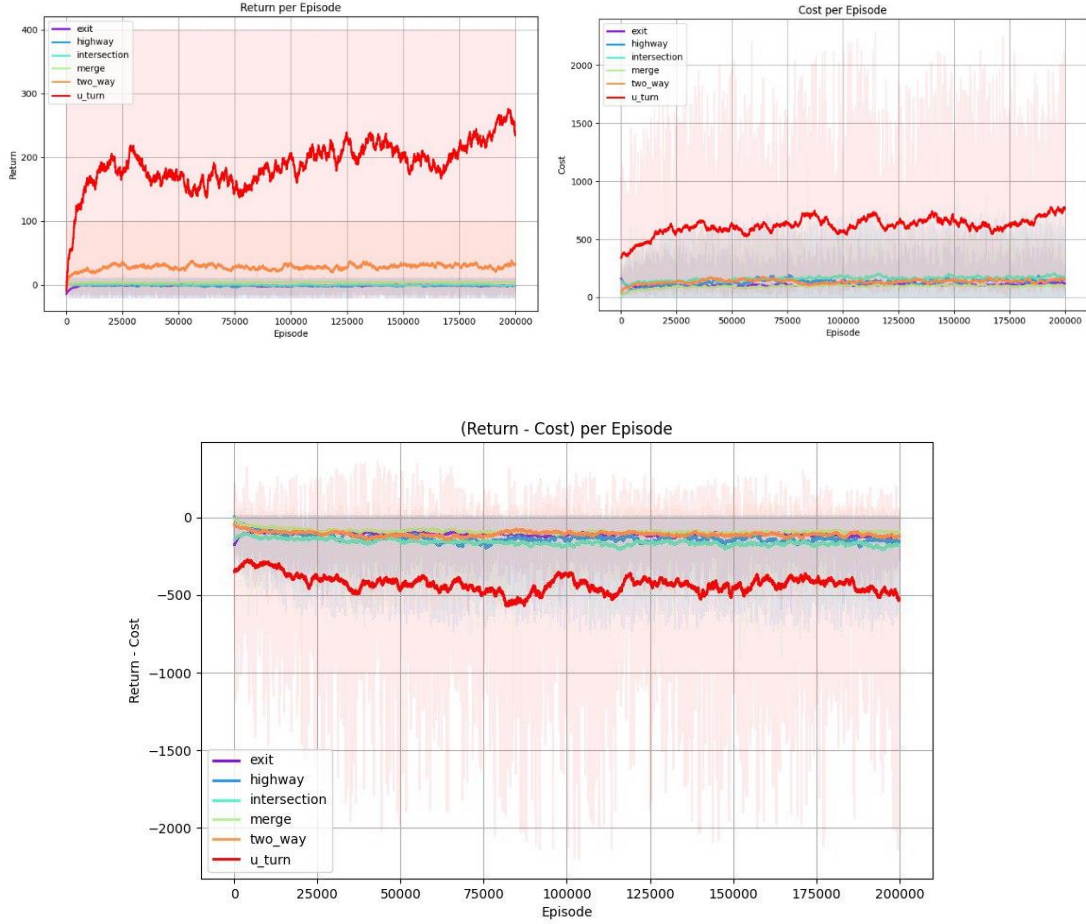


Figure 4.2: Trust-Region policy optimization applied on a different road-types

The Trust-policy optimization technique ensures that agent learns while making an actions or decisions within the environment. Constraints in that algorithms are depicted as a trust region, actions that the agent takes meant to maximize cumulative return while maintaining stability and performance in the environment. Given the figures depicted in 4.9, we can observe the performance of an agent on a different road-types environments.

First, let us overview the least performing road-type. We can see that the agent that is learning within the exit-environment has a lower reward compared to other environments. By analyzing and going through the costs within the environment, such as enforcing exiting, collision with target vehicle, omitting collision with attacker-vehicles, we can make a conclusion that the environment is

difficult for that algorithm. Introducing the randomness of the starting point leads to the vehicle being generated sometimes outside the possible exit road, resulting in them to reverting to the usual highway environment scenario.

The highway-environment is the second one after the exit-environment with comparatively lower rewards. This type of scenario is the least complex which produces returns from 0 to 50. Which suggests that other policies may be more beneficial for that type of environment. The cost complements the return which suggests for the highway to be simple environment.

Agent performance within the intersection-type environment seems to be less volatile. However, other types of optimizations may be more beneficial in terms of performance in comparison to TRPO. Even though comparing to the exit and highway environment the intersection is more complex, according to the costs we can conclude that the agent manages to subsequently find converging point for this type of the environment.

In terms of the merge-environment, we can observe the upward trend in terms of agent learning within the environment. The overall performance suggests that there should be more fine-tuning for the agent to merge in an efficient and safe manner. The cost for the merge-type environment depicts moderate complexity of this road-type scenario, yet the agent maintains satisfactory performance. The main reason for this outcome may be the initial parameters initiated for the agent as to the target vehicle to be enforced to be generated on the merging lane and for the attacker vehicle to have a random lane to be chosen.

In a two-way scenario, we can observe the tendency of an agent to maintain satisfactory performance while executing the basic maneuvers while struggling with more complex actions. That can be described by the environment having higher complexity, requiring the agent to maintain the speed, avoid oncoming traffic, as well as ensuring change of the lane while overpassing other vehicles. As mentioned before the “attacker-vehicle” is rewarded for collision with a target vehicle and receives a penalty when colliding with other attacker-vehicles. The cost for the two-way environment suggests that this type of environment is moderately difficult to learn.

The U-turn type of scenario demonstrates the good agent performance comparing to other road-types scenarios. Initially it may be complex, but the agent eventually learns and has superior performance as shown by the high returns received. It is worth mentioning that this type of environment is the most time consuming as per duration of the road, which is why it is slightly further

depicted within the figure 4.2. The cost of the environment compared to return suggests that the U-turn is the most complex environment to learn.

Summarizing, we can observe that TRPO supported agent outputs stable learning process. The complexity of environments, like exit, intersection, two-way, and merge may be addressed by further fine-tuning of the environments specifically with TRPO usage.

4.1.1.3 Multi-agent centralized optimization policy

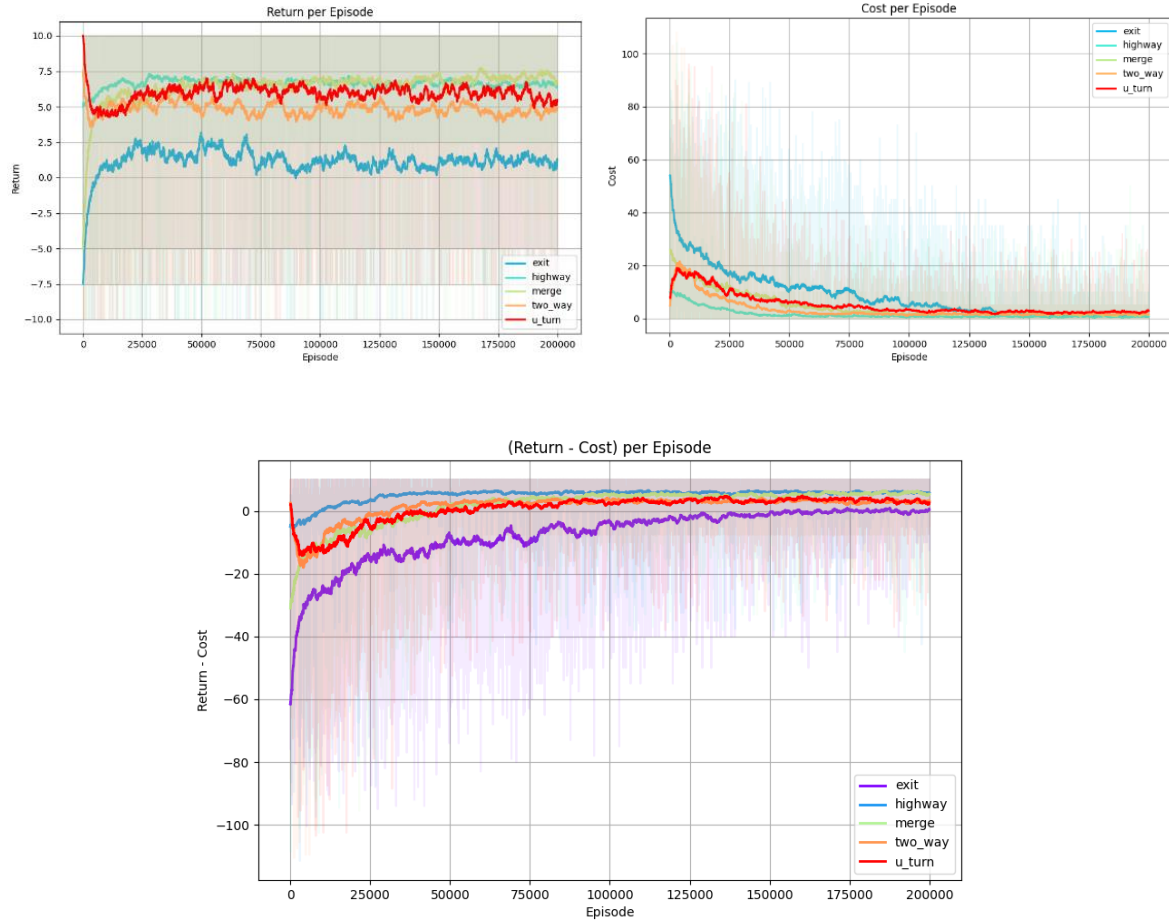


Figure 4.3: Multi-agent centralized optimization policy applied on a different road-types

Figure 4.3 shows how the Multi-agent Centralized Policy Optimization is trained on the different road types.

Starting from the exit-type environment with the lowest performance, we can observe that the initial starting value is set to -60. After the episodes are continued, we can observe improvement in the value reaching the point of -10 and below. The exit road type environment is comparably easy like the highway environment. To generate the accidents and enforce collisions we introduced a higher time constraint for the vehicles to collide and added a higher reward for the cars to collide with each other. Regarding the coordination characteristic of the multi-agent CPO algorithm, we can conclude that the exit-type environment is less compatible with the coordinated off-ramp collision generations. For future improvements, we consider increasing the exiting lanes for the coordinated multi-agent algorithm to have broader actions to lead the target vehicle toward the collision.

The curve converges slower when the MACPO algorithm is applied to the merged environment, suggesting that the model learns slower in comparison to other environments. Such a behaviour can be described as the target vehicle being enforced to be generated on the merge line, whereas attacking vehicles are randomly generated. This leads to different approaches for accident generation.

The algorithm's performance on the highway environment shows gradual improvement after a couple of runs, which can be described by the environment's simplicity. The highway road type shows better performance for the cooperative adversarial approach.

A two-way road environment involves bidirectional traffic. The MACPO collaborative approach requires effective lane management and interaction with vehicles travelling in the opposite direction to create and facilitate the target collision occurrence. The model handles this scenario well, achieving stable, near-zero performance.

The U-turn scenario's stable performance suggests that the MACPO algorithm performs highly coordinated attacks on the target vehicle.

Summarising we can note that Multi-Agent Constrained Policy Optimisation extends the ability of the Single-Agent Constrained Policy Optimisation and outperforms it in the advantages of the parallel learning by simulation exploration of the environment space which fastens the learning ability and convergence of the model. The multi-agency approach's introduction of the agent specialization allows the agent to identify and focus on a specific task, creating a diverse and effective strategy sporting the environment goal.

4.1.1.4 Comparison of the different policy optimization techniques per highway road-type environment

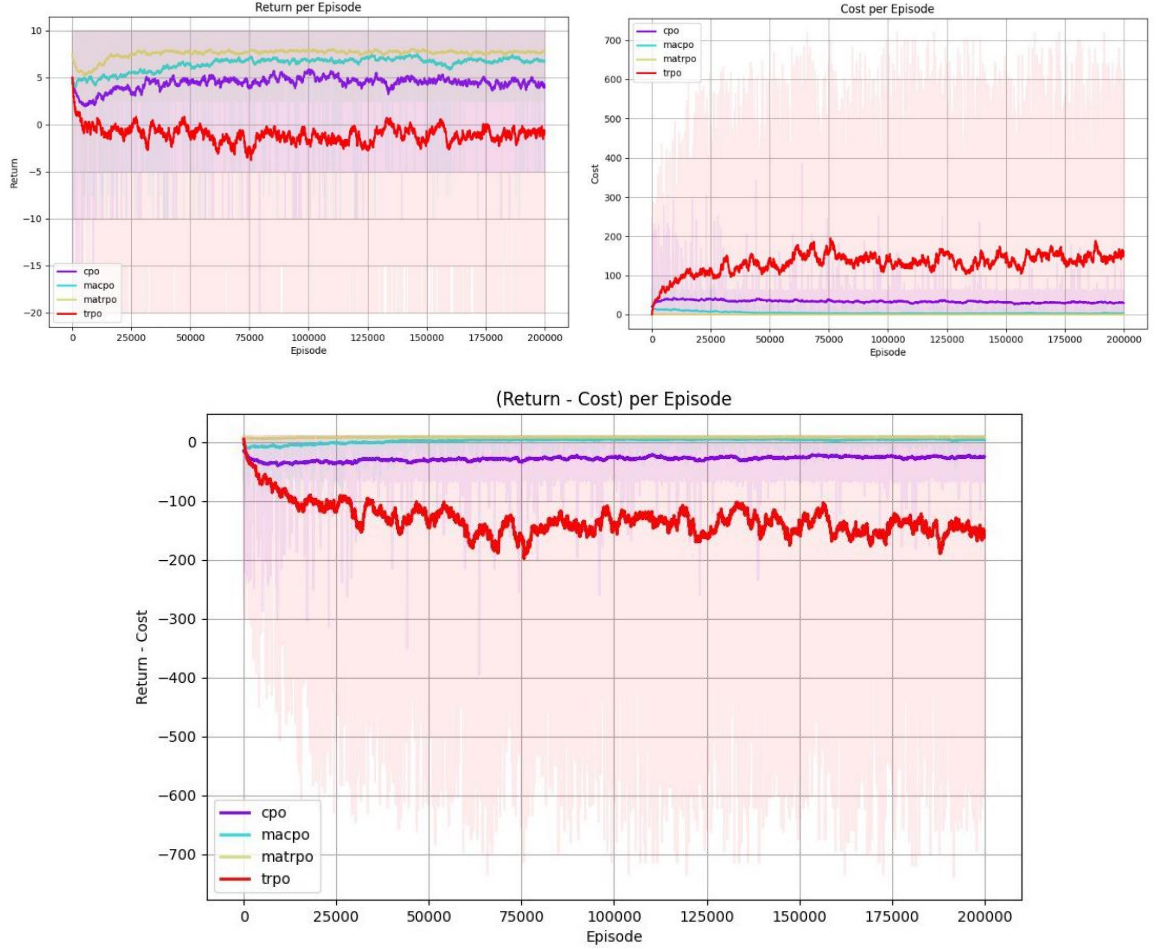


Figure 4.4: Comparison of CPO, TRPO, MACPO, MATRPO on a highway road-type environment

Reference to the work from our lab [5]. Similarity in CPO performance achieving on the highway road-type.

Apart from that we can observe consistency in terms of CPO performance.

In terms of the MACPO-supported highway environment, we can observe similar learning performance to the CPO. The additional complexity of coordination and cooperation added by the multi-agent environment ensures that the attacker-agents aim at colliding with the target vehicle. There is less collision costs related to the attacking agents colluding with each other.

In terms of TRPO we can observe that complexity given by the TRPO constrained goal, there are fluctuations between the cost and reward which indicates that TRPO is less compatible with highway road type environment.

On the other hand, MATRPO depicts superior performance. Applying multi-agency towards the TRPO-constrained characteristics, like conservative updates, MATRPO has stable learning and suppresses fluctuations between the reward and cost, unlike in TRPO. We can observe a better outcome when we try to solve the problem of learning a coordinated attack policy for multiple vehicles by treating the vehicles as separate, cooperating agents.

4.1.1.5 Comparison of the different policy optimization techniques per exit road-type environment

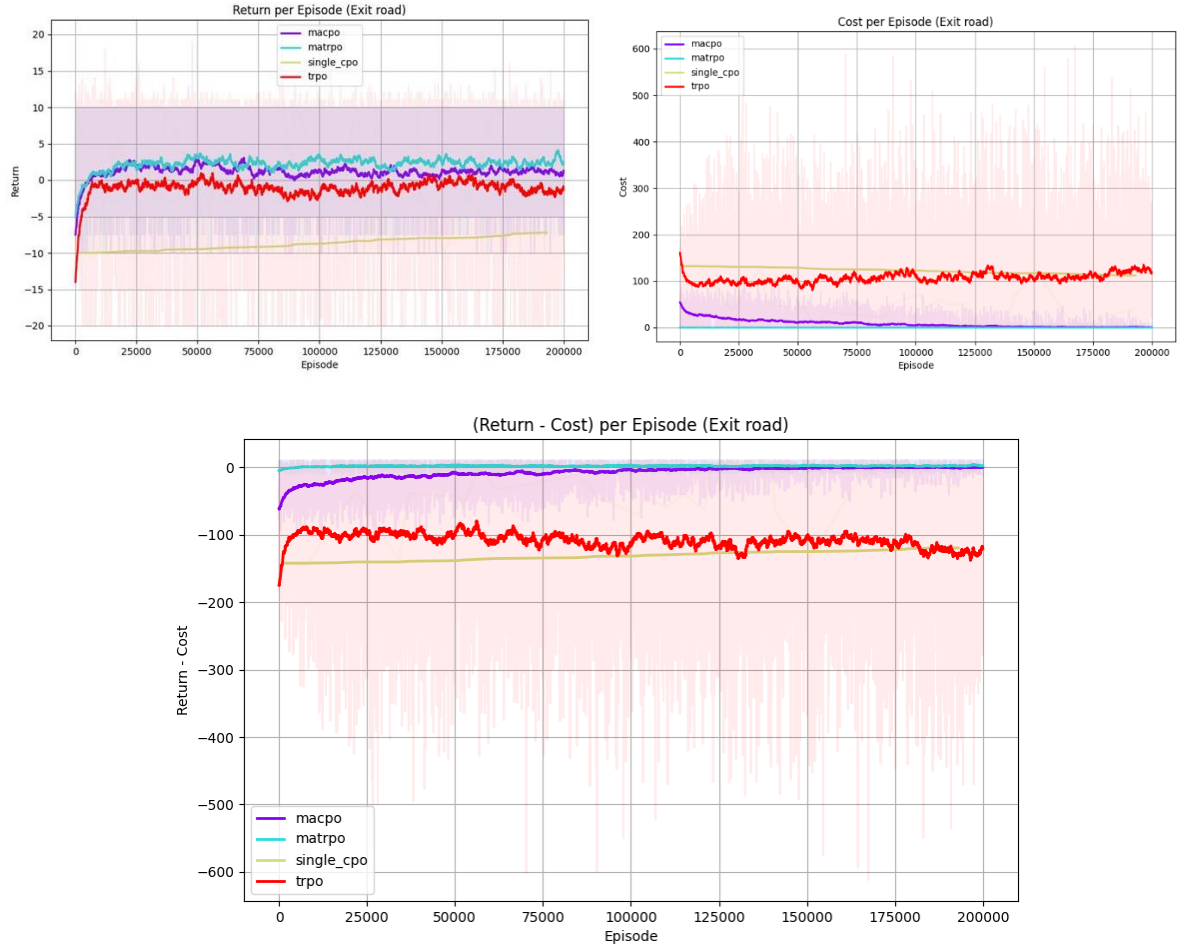


Figure 4.5: Comparison of CPO, TRPO, MACPO, MATRPO on an exit road-type environment

As in other environments, each model explores the environment for the first few episodes and then converges towards stable values. Overall, the ramp road-type environment is more constrained regarding the variety of the episodes generated. At first, the generated episodes were mostly similar to the highway nature setting, as the exit lane was sometimes located after the point to be accessed by the randomly generated on-road vehicles. Comparatively merge road was enforced to be a starting point. In the exit road, we utilized the default goal of the environment for the car to exit the road using the exit lane. The return-cost plot, Figure 4.5, suggests that coordinated actions between the agents perform better in the exit-road-type environment.

4.1.1.6 Comparison of the different policy optimization techniques per merge road-type environment

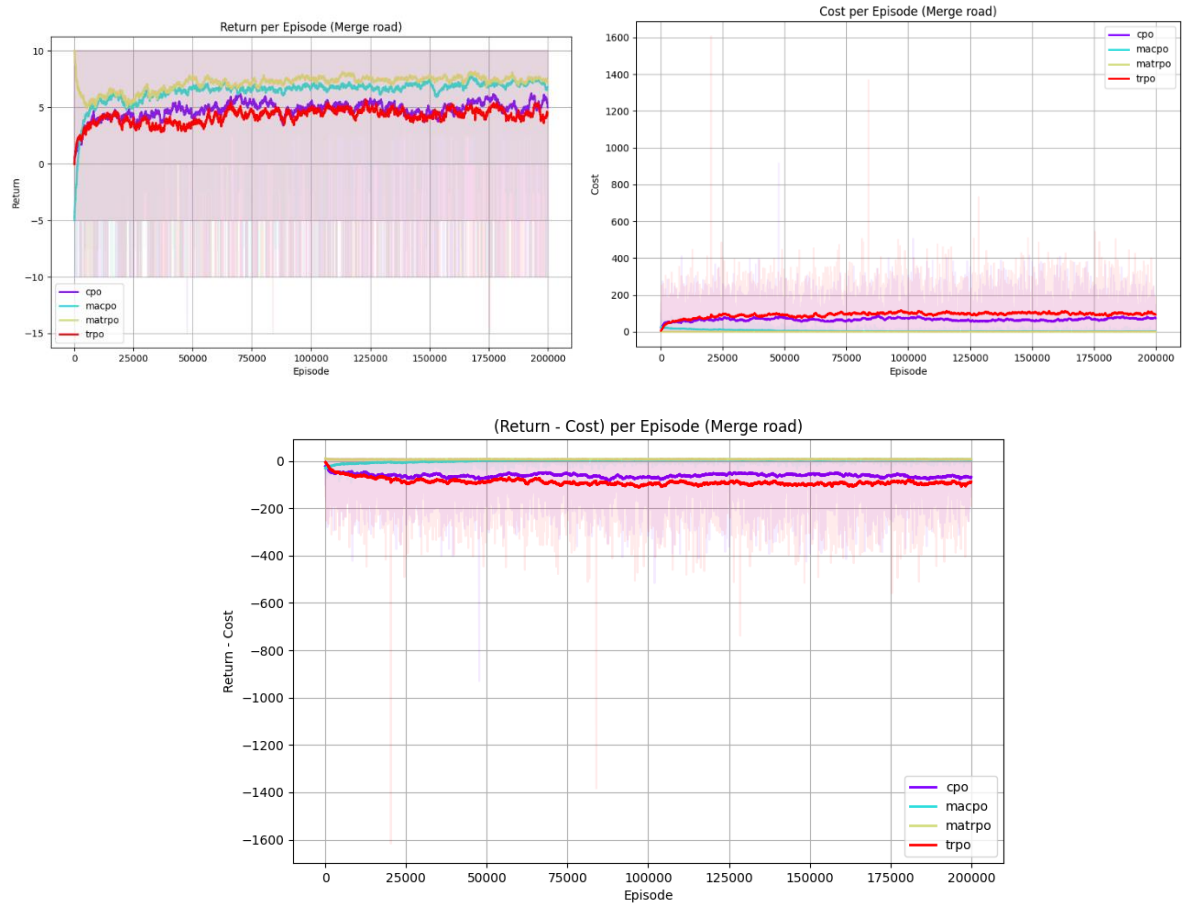


Figure 4.6: Comparison of CPO, TRPO, MACPO, MATRPO on a merge road-type environment

The distinct characteristic of the merge type of the environment is that we enforced the target vehicle to be generated on the merge lane connected to the highway. As for the attacker vehicles with an adversarial goal to create collisions with the target vehicle – those are generated randomly, sometimes on the merge lane or the highway 3-lane road. In the Figure 4.6, CPO has stable learning in a single-agent mode compared to the TRPO algorithm for the merged road-type environment. In the multi-agent setting, we can observe that the trust region policy (MATRPO) maintains learning stability and slightly outperforms MACPO.

4.1.1.7 Comparison of the different policy optimization techniques per U-turn road-type environment

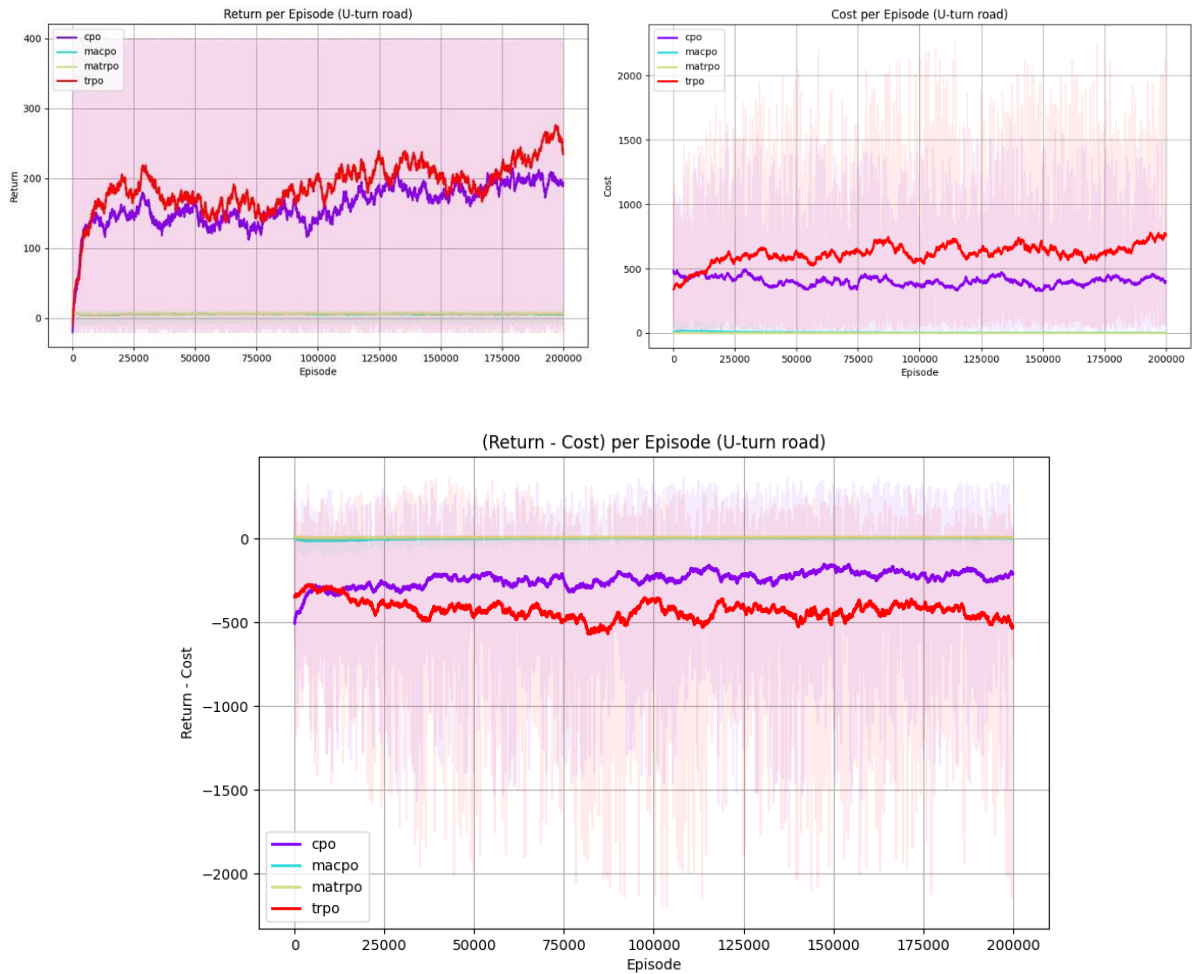


Figure 4.7: Comparison of CPO, TRPO, MACPO, MATRPO on a U-turn environment

The U-turn environment is the most distant compared to other tested environments for the single-agent and multi-agent adversarial approach. The U-turn has three components that generate accidents: the straight lane (highway), curve and straight lane (highway) closing this road. The duration of the time referred to the episode to be terminated is the longest compared to other environments as a U-turn curve is longer in the lane dimensions. Apart from the usual rewards related to collisions with target vehicle generation, this road type has a speed reward for joining the curve, maintaining the curve speed, and exiting the curve speed.

The overall learning process for the agents relies on repeated interactions with the environment. The input received at the end of the episode is used to improve policies over time. From Figure 4.7, we can observe that Multi-agent algorithms excel in U-turn road scenarios, which is mainly achieved by the cooperative techniques introduced by the multi-agent setting. The MACPO algorithm has a stable return to cost, which suggests its slight outperformance of the MATRPO. Single-agent methods are struggling in the U-turn setting with the slightly higher performance of CPO over TRPO.

4.1.1.8 Comparison of the different policy optimization techniques per two-way road-type environment

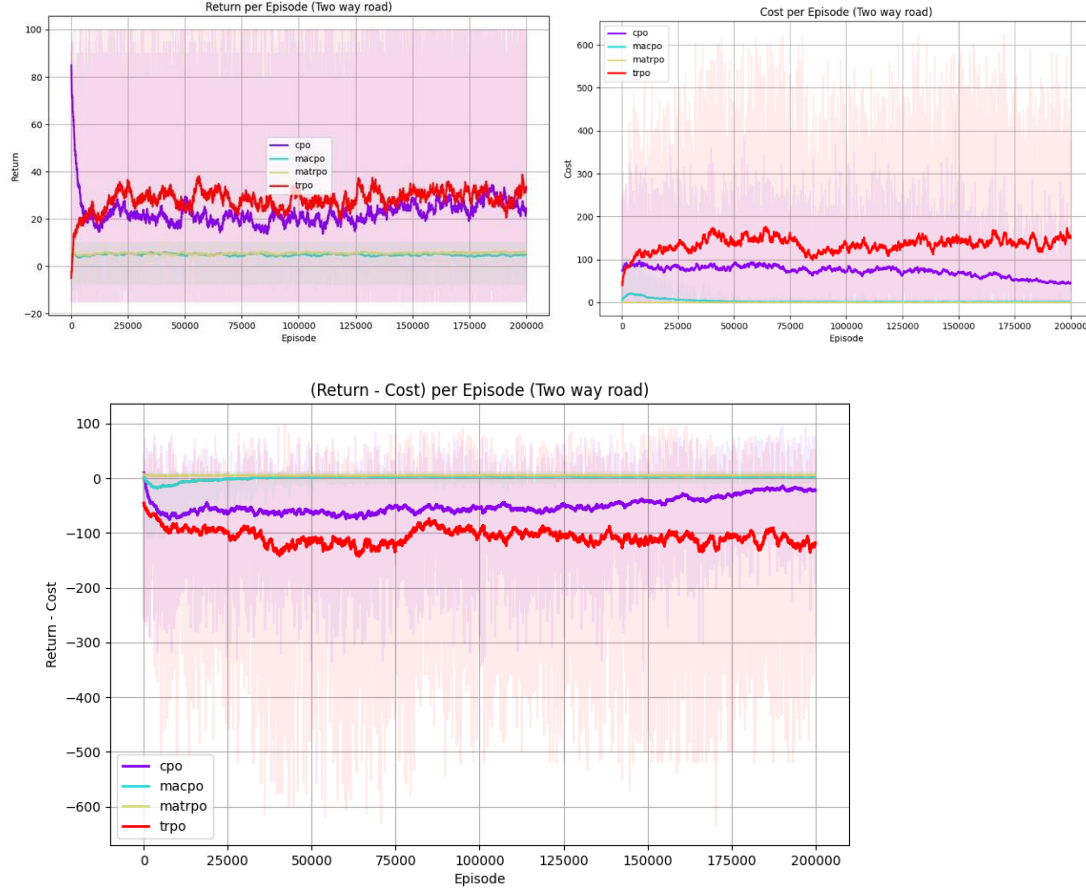


Figure 4.8: Comparison of CPO, TRPO, MACPO, MATRPO on a two-way environment

The distinct characteristic of the two-way type of environment is the bidirectional driving setting – the vehicles manage ongoing traffic. We had multiple learning approaches for this type of environment. At first, we introduced only one kind of direction for the cars to be created. On that setting we enforced vehicles to explore adversarial learning within the one lane, rewarding for the lane managing approach additionally to the enforcement of the target vehicle collision by the attacker cars. After repeating that approach in the other direction, we combined those two. The vehicles were rewarded for maintaining speed, not going in the opposite traffic in addition to the collision enforcement by the attacker vehicles towards the target one.

Looking at Figure 4.8, we can observe that the MACPO and MATRPO have a better return-cost tradeoff, suggesting that a coordinated approach better fits the two-way road environment. Comparatively, the MATRPO algorithm slightly outperforms the MACPO algorithm in the two-way setting in other environments.

In future enhancements, we expect to combine the two-way environment bidirectional traffic characteristic toward other environments to introduce the additional complexion of the environments.

4.1.2 Highway Env change with Stable Baselines 3 Library

After these modifications, we can use two libraries, `stable_baselines3` and `stable-baselines3-contrib`, which provide access to various policy optimizations for reinforcement learning.

In this regard, the positive aspect is that when making changes on one type of road, depending on the type of action, you can test various available policies in the above-mentioned libraries. The parameters can be called in the model directly.

So, for this method, we modify the built-in `AggressiveVehicle` class from `Highway_Env` with some changes in the behaviour of the car, such as pursuing an ego-car.

On the road environment side, the class is inherited with a change in the encouragement to pursue the ego-car at increased speed and the encouragement to cause an accident with the ego-car by adding the inherited `AggressiveVehicle`.

When considering stable baseline 3 library to be applied within the highway-attacker environment the action method was considered in mind. So far those that satisfied the discrete action support like DQN, PPO, A2C, TRPO.

To utilize the SB3 library without major modification to its initial characteristics, we created a custom aggressive vehicle (named `TargetingAggressiveVehicle`) inherited from `AggressiveVehicle`. This vehicle has two distinct characteristics: it enforces tailgating or close following the ego vehicle, speeds to overspeed the ego vehicle, and can catch up if the aggressive vehicle is generated further from the ego vehicle, Figure 4.9.

```

class TargetingAggressiveVehicle(AggressiveVehicle):
    def __init__(self, road, position, heading=0, speed=0):
        super().__init__(road, position, heading, speed)

    def act(self, action=None):
        ego_vehicle = self.road.vehicles[0]
        if ego_vehicle:
            self.target_velocity = ego_vehicle.speed + 10
            self.follow_ego(ego_vehicle)

        super().act(action)

    def follow_ego(self, ego_vehicle):
        if self.lane_index[0] < ego_vehicle.lane_index[0]:
            self.lane_change_left()
        elif self.lane_index[0] > ego_vehicle.lane_index[0]:
            self.lane_change_right()

        if self.position[0] < ego_vehicle.position[0]:
            self.target_velocity = self.speed + 5
        elif self.position[0] > ego_vehicle.position[0]:
            self.target_velocity = self.speed - 5

class CustomHighwayEnv(HighwayEnv):
    def _create_vehicles(self):
        other_vehicles_type = TargetingAggressiveVehicle |
        other_per_controlled = near_split(self.config["vehicles_count"], num_bins=s

        self.controlled_vehicles = []
        for others in other_per_controlled:
            # Create the ego vehicle
            ego_vehicle = Vehicle.create_random(self.road, speed=25, lane_id=self.c
            ego_vehicle = ControlledVehicle(self.road, ego_vehicle.position, ego_ve
            self.controlled_vehicles.append(ego_vehicle)
            self.road.vehicles.append(ego_vehicle)

        for _ in range(others):
            vehicle = other_vehicles_type.create_random(self.road, spacing=1 /
            vehicle.randomize_behavior()
            self.road.vehicles.append(vehicle)

```

Figure 4.9: Inherited class for the TargetingAggressiveVehicle and Environment inherited class with parameters changed for reward introduction and vehicle addition for the SB3 library policy optimization training

We tended to simplify Highway-attack-env agents so that they were easily accessible for the SB3 library application.

The next step in generating the scenario was to choose and inherit the environment, introducing a reward function for the TargetingAggressiveVehicle to enforce its adversarial behaviour, Figure 4.9. We then added the vehicles to the custom environment by introducing our updated aggressive vehicle with a target vehicle, as actors or agents of the environment. Each environment road type had a similar extension that we added of changed vehicle introduction and the reward function addition.

The first run for each optimization policy was creating the optimization function and main function to facilitate the reinforcement learning approach for the custom road-type environment. We tested DQN, PPO, and A2C, which closely matched DiscreteMetaActions(Section 3.1.1, Figure 3.1) specifications over 20000 episodes.

The initial learning outcome represented via the tensorboard suggested that further parameters must be changed. After changing a few of them, like a learning rate, buffer size, and target update interval we obtained a slightly better performance than the initial results. Further investigating the parameter tuning we found a similar issue addressed for the difference between the reinforcement learning agents from Farhama Foundations and those from the SB3 library described in the issue.

To further explore hyperparameter tuning ability we investigated the possibility of the addition of the function that creates trials to test the case on, such library was OPTUNA library.

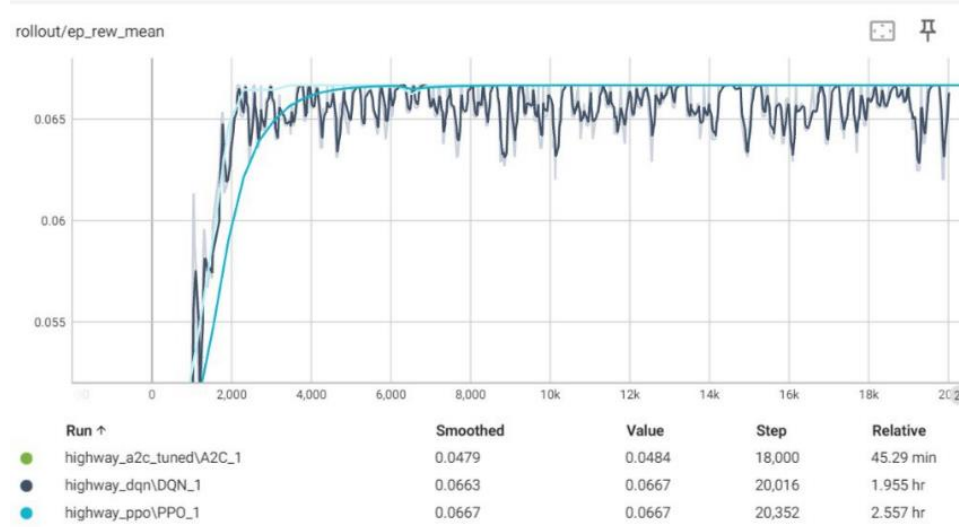


Figure 4.10 Manually tuned parameters from sb3 PPO, DQN, A2C

OPTUNA as an automated solution allows the user to insert ranges towards the parameters to be tested.

For the DQN, based on the suggested shifts between the default parameters of the SB3 library function and the proposed in [72] parameters, we identified ranges for the DQN, PPO, and predicted ranges for the A2C as it was not investigated in [72]. Such parameters were learning rate, gamma, buffer and batch size, and value and entropy coefficients. Improvements after parameter tuning and their visual comparison using a tensorboard are highlighted in Figures 4.10 and 4.11.

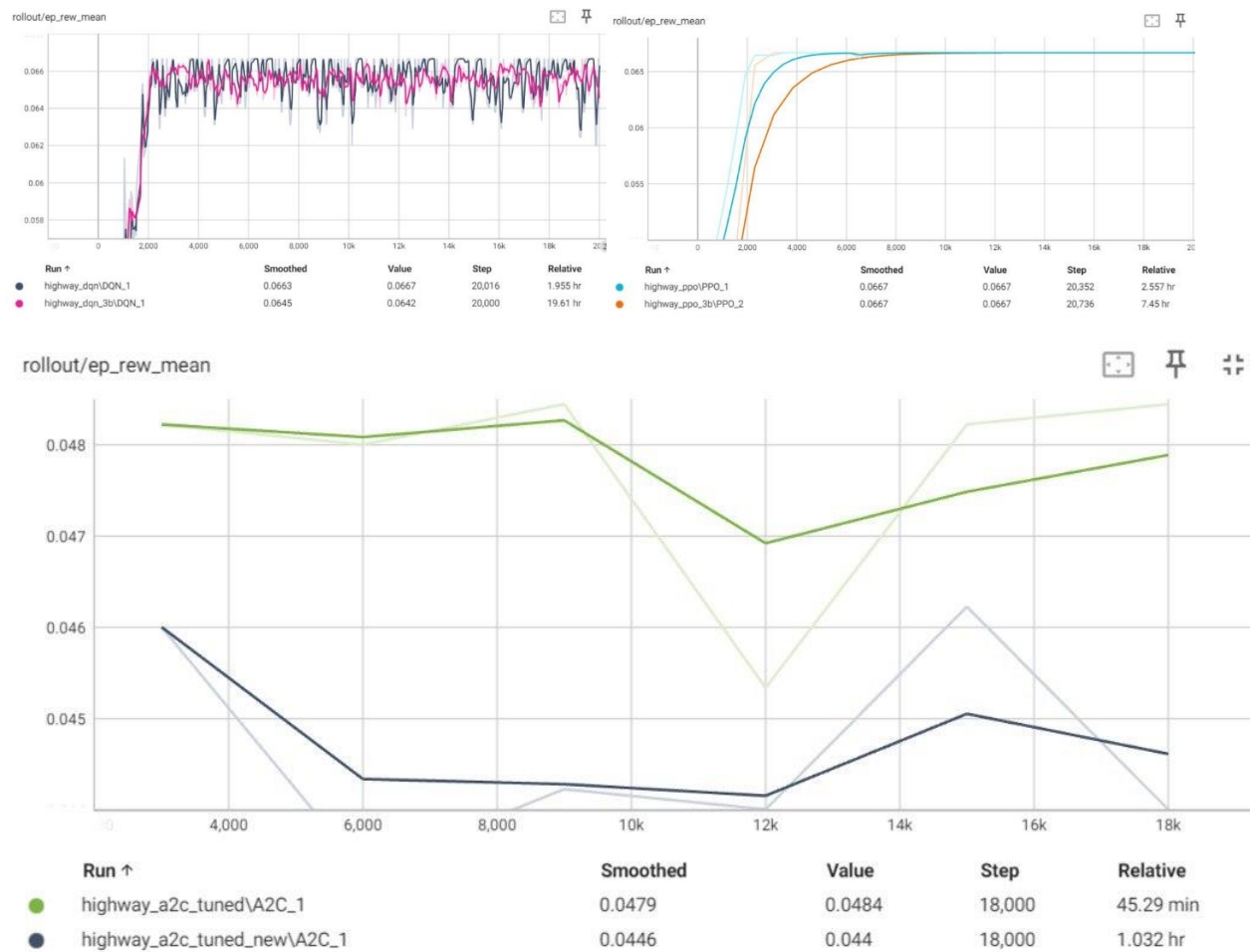


Figure 4.11 OPTUNA tuned parameters for DQN, PPO and A2C

Examples provided in Figures 4.10 and 4.11 describe the reward obtained by the agent. The agent's goal or aim is maximizing its reward. Further tuning of the parameters for the model from the SB3 library needs to be performed for the agent to be learning efficiently [72].

Chapter 5

Conclusions

The initial objective of the thesis was to consider and adopt an environment for direct stress testing of an autonomous vehicle in a such way that we preserve an adversarial approach alongside the black box configuration of the vehicle being tested, named the target vehicle.

When working in the Highway-Env virtual environment, the main components of the environment and the possibility of building custom scenarios using documentation were studied. The highway-attack-env environment was compared with the original highway-env, in the process some conflicting outdated libraries were eliminated to adapt the environment to the new task. Further work was carried out adapting existing algorithms and extending them to new road types by configuring and inheriting the main types of roads presented in highway-env.

The thesis addressed the problem of virtual environment realism of the Highway_Env in the context of this adversarial black box testing problem. By comparing the generated scenarios with real-world situations, we can conclude that the virtual environment is a relevant component for analysis or research in the automotive industry. In addition, the initial characteristics described in the highway-env environment such as kinematics, behaviour, and definition of speed indicators according to the type of road provide the virtual environment with the opportunity to achieve realism on aspects of the driving scenario relevant to autonomous driving. Such as the approach used is realistic enough for the problem strategic control of a vehicle, to test that level of behaviour in other algorithms and with different parameters.

In addition, this thesis included testing of various existing algorithms directly in the working virtual environment. The algorithms were compared according to their use on several types of roads, as well as among themselves to identify patterns and recommendations for using the algorithms.

5.1 Future Work

5.1.1 Generate Test Cases for Physical Crash Tests

For example, many companies now provide the ability to assess physical damage to vehicles, like companies related to the car insurance or the companies that are related towards the detail or simulation assessments like Altair Radios [73]. Thus, the episode(s) generated by the approach demonstrated in this thesis can become the necessary dataset for further damage analysis, as well as

for assessing details that can be improved to ensure the safety of passengers and the driver, depending on the type of collision. An approximate range of areas of interest in this work could be Insurance, Automotive manufacturing, Various companies researching the physical characteristics of cars, as well as areas of training for further research in this area.

5.1.2 Adding Variety of the Vehicle Types as well as Environmental Types

The environment currently only provides the ability to use a car and a bicycle as a vehicle. To improve and approach realism in the presented virtual environment, we could add and adapt the characteristics of other vehicles (for example, a truck, etc.) to the environment in the future. When studying the capabilities of the virtual environment, a dummy truck vehicle was created. Since research with other vehicles was not carried out, the created element does not appear in the thesis. Adding specific roads for testing, in the process of implementing and testing the capabilities of creating elements in the selected environment, we created roads of the narrow road type. This type of road was not mentioned in the study since its testing was not performed in the previous section.

By constructing existing road types according to their real appearance, each street or specific intersection can be assessed in terms of safety or identified as an area with various levels of attention required to avoid a car accident.

5.1.3 Adding Weather Conditions

In the second chapter, we mentioned weather conditions as a characteristic that is included in the data collection when forming a dataset for describing road accidents. By adding characteristics inherent in climatic conditions as "error" parameters to the operation of a vehicle, it is possible to identify new scenarios of vehicle behaviour or describe clear actions that should be taken by an autonomous vehicle.

5.1.4 Adding a Benchmark for Policy Optimization Algorithms

During the readers' review stage, suggestions highlighted the further benchmarking scope for the policy optimization algorithms, such as a broader comparison of the single-agent approaches with multi-agent ones by adding the median between the two as a single-agent parallelization approach. In such a manner, future studies can further analyze the policy optimization techniques regarding model

training for adversarial stress testing with modifications and further investigate the differences between the single-agent and multi-agent approaches for the goal of the proposed environment.

5.1.5 Narratives Creation to Describe the Accident

In addition, since we know the location of the car, its speed, and the distance between the car of interest and the surrounding elements, it is possible to create narratives of the car's actions. Narratives of car behaviour can be created by describing them with placeholders for parameters in the text or by using a visual adapter to create image captions for actions performed in the generated scenario. Determination of the collision type depends on the location of the vehicles in the accident or their visual determination using a visual adapter.

It is also possible to use this environment as a primary test or for test case generation before testing them in the real world. Risk assessment and comparison of different algorithms can contribute to the creation of a more secure and safe mechanism for an autonomous system.

This environment can also be used as a prerequisite for other 3D environments, such as CARLA [12].

5.1.6 Real-World Cases Generation

We also conducted and researched some approaches to recreating real-world accidents using the database of road incidents like those available on the Bing traffic application. By creating a snapshot of the road with the accident identified and reading it using a large language model visual recognition system like Molmo by AI2 as well as comparing the given information by the Bing system itself about how serious the accident is, where it happened and what date time the occurrence was given. By utilizing that information, we tried to recreate the road construction by defining the LLM with simple road connection details like a straight lane, circular lane, and sine lane, to describe the road network on the screenshot taken from the Bing traffic application. And then to be recreated for testing purposes using our environment.

References

- [1] World Health Organization, "Road traffic mortality," 13 December 2023. [Online]. Available: <https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/road-traffic-mortality>.
- [2] World Health Organization, "Road traffic injuries," 13 December 2023. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>.
- [3] U.S. Department of Transportation, "Automated Driving Systems: A Vision for Safety," September 2017. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf.
- [4] E. Leurent., " An environment for autonomous driving decision-making," 2018. [Online]. Available: <https://github.com/eleurent/highway-env>.
- [5] X. Zhou, "Aggressiveness-regulated Multi-agent Stress Testing of Autonomous Vehicles," 2023.
- [6] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 31 April 2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/.
- [7] C. J. Nash, D. J. Cole and R. S. Big, "A review of human sensory dynamics for application to models of driver steering and speed control," *Biological Cybernetics*, p. 91–116, 16 April 2016.
- [8] J. Lu, Z. Peng, S. Yang, Y. Ma, R. Wang, Z. Pang, X. Feng, Y. Chen and Y. Cao, "A review of sensory interactions between autonomous vehicles and drivers," *Journal of Systems Architecture*, August 2023.
- [9] S. Waslander, "Autonomous Mobile Robotics," 29 September 2014. [Online]. Available: http://wavelab.uwaterloo.ca/sharedata/ME597/ME597_Lecture_Slides/ME597-4-Measurement.pdf. [Accessed 2023].

- [10] Lee, Gober & Reyna, PLLC, "Examining Autonomous Car Accidents and Statistics," 14 February 2024. [Online]. Available: <https://www.lgrrlawfirm.com/blog/examining-autonomous-car-accidents-and-statistics-2/>.
- [11] M. Abdel-Aty and S. Ding, "A matched case-control analysis of autonomous vs human-driven vehicle accidents," *Nature Communications*, 18 June 2024.
- [12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and Vladlen, "CARLA: An open urban driving simulator," *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1-16, 2017.
- [13] The Farama Foundation, "Projects".
- [14] L. Zhao, N. Farhi, Y. Valero and Z. Christoforou, "Long Short-Time Memory Neural Networks for Human Driving Behavior Modelling," *Transportation Research Procedia*, pp. 2589-2596, 13 December 2023.
- [15] N. Negi, O. Jelassi, S. Clemencon and S. Fischmeister, "A LSTM Approach to Detection of Autonomous Vehicle Hijacking," *In Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*, pp. 475-482, 2019.
- [16] R. Bani-Hani, S. H. Aljbour and M. Shurman, "Autonomous Vehicles Trajectory Prediction Approach Using Machine Learning Test," *14th International Conference on Information and Communication Systems (ICICS)*, pp. 1-6, 2023.
- [17] J. S. Albus., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller," pp. 237-240.
- [18] J. Zhao, B. Liang and Q. Chen, "The key technology toward the self-driving car," *International Journal of Intelligent*, vol. 6, no. 1, pp. 2-20, 2 January 2018.
- [19] C.-M. Lin and C.-H. Chen, "Car-Following Control Using Recurrent Cerebellar Model Articulation Controller," *Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3660-3673, 19 November 2007.

- [20] F. Alam, R. Mehmood and I. Katib, "Comparison of Decision Trees and Deep Learning for Object Classification in Autonomous Driving," *Smart Infrastructure and Applications*, p. 135–158, 21 June 2019.
- [21] M. Alqarqaz, M. Bani Younes and R. Qaddoura, "An Object Classification Approach for Autonomous Vehicles Using Machine Learning Techniques," pp. 1-17, 3 February 2023.
- [22] A. Naim, J. Bowkett, S. Karumanchi, P. Tavallali and B. Kennedy, "Deterministic Iteratively Built KD-Tree with KNN Search for Exact Applications," pp. 1-7, 7 June 2021.
- [23] M. J, X. H, S. K and L. H, "A Bayesian Driver Agent Model for Autonomous Vehicles System Based on Knowledge-Aware and Real-Time Data," 6 January 2021.
- [24] W. Kontara and S. Ahn, "Bayesian Methods in Automated Vehicle's Car-following Uncertainties: Enabling Strategic Decision Making," *Systems and Control*, pp. 1-22, 25 October 2022.
- [25] S. Grigorescu, B. Trasnea, T. Cocias and G. Macesanu, "A Survey of Deep Learning Techniques for Autonomous Driving," *Machine Learning*, vol. 37, p. 362–386, 24 March 2020.
- [26] X. Mo, Y. Xing and C. Lv, "Graph and Recurrent Neural Network-based Vehicle Trajectory Prediction For Highway Driving," *Robotics*, 8 July 2021.
- [27] Y. Tian, K. Pei, S. Jana and B. Ray, "DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars," p. 303–314, 3 June 2018.
- [28] Y. Liu, X. Wang, L. Li, S. Cheng and Z. Chen, "A Novel Lane Change Decision-Making Model of Autonomous Vehicle Based on Support Vector Machine," pp. 26543-26550, 21 February 2019.
- [29] G. Garimella, J. Funke, C. Wang and M. Kobilarov, "Neural Network Modeling for Steering Control of an Autonomous," *International Conference on Intelligent Robots and Systems (IROS)*, pp. 2609-2615, 28 September 2018.

- [30] X. Pan, Y. You, Z. Wang and C. Lu, "Virtual to Real Reinforcement Learning for Autonomous Driving," *Proceedings of the British Machine Vision Conference (BMVC)*, 26 September 2017.
- [31] L. Wang, J. Liu, H. Shao, W. Wang, R. Chen, Y. Liu and S. L. Waslander, "Efficient Reinforcement Learning for Autonomous Driving with Parameterized Skills and Priors," 8 May 2023.
- [32] L. Trinh, Q.-H. Luu, T. M. Nguyen and H. L. Vu, "A novel framework for adaptive stress testing of autonomous vehicles in multi-lane roads," February 2024. [Online]. Available: <https://arxiv.org/abs/2402.11813>.
- [33] D. Nalic, H. Li, A. Eichberger, C. Wellershaus, A. Pandurevic and B. Rogic, "Stress Testing Method for Scenario-Based Testing of Automated Driving Systems," *IEEE Access*, 18 November 2020. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9290063>.
- [34] J. B. A. C. H. Z. R. M. Stanley Bak, "Stress Testing Autonomous Racing Overtake Maneuvers with RRT," in *IEEE Intelligent Vehicles Symposium (IV)*, Germany, 2022.
- [35] M. I. B. L. R. S. M. J. K. Harrison Delecki, "How Do We Fail? Stress Testing Perception in Autonomous Vehicles," in *IEEE IROS*, 2022.
- [36] M. Koren, S. Alsaif, R. Lee and M. J. Kochenderfer, "Adaptive Stress Testing for Autonomous Vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, China, 2018.
- [37] R. Song, M. O. Ozmen, H. Kim, R. Muller, Z. B. Celik and A. Bianchi, "Discovering Adversarial Driving Maneuvers against Autonomous Vehicles," Anaheim, 2023.
- [38] X. Cai, X. Bai, Z. Cui, P. Hang, H. Yu and Y. Ren, "Adversarial Stress Test for Autonomous Vehicle via Series Reinforcement Learning Tasks with Reward Shaping," in *IEEE Transactions on Intelligent Vehicles*, 2024.
- [39] A. Sharif and D. Marijan, "Adversarial Deep Reinforcement Learning for Improving the Robustness of Multi-agent Autonomous Driving Policies," 2023.

- [40] Insurance Institute for Highway Safety (IIHS), "Fatality Facts 2022".
- [41] C. Lyon, D. Mayhew, M.-A. Granić, R. Robertson, W. Vanlaar, H. Woods-Fry, C. Thevenet, G. Furian and A. Soteropoulos, "Age and road safety performance: Focusing on elderly and young drivers," *IATSS Research*, vol. 44, no. 3, pp. 212-219, 2020.
- [42] "Canada: Road Safety Country Profile 2023," OECD Publishing, [Online]. Available: <https://www.itf-oecd.org/sites/default/files/canada-road-safety.pdf>.
- [43] NST Law, "Self-Driving Car Statistics".
- [44] Đ. Petrovića, R. Mijailović and D. Pešić, "Traffic Accidents with Autonomous Vehicles: Type of Collisions, Manoeuvres and Errors of Conventional Vehicles' Drivers," *Transportation Research Procedia*, vol. 45, pp. 161-168, 24 September 2021.
- [45] National Safety Council, "Type of Crash".
- [46] Shlosman Law Firm, "Four of the Most Dangerous Types of Car Accidents".
- [47] The Farama Foundation, "Announcing The Farama Foundation: The future of open source reinforcement learning," 2022.
- [48] The Farama Foundation, "Gymnasium Documentation: An API standard for reinforcement learning with a diverse collection of reference environments".
- [49] S. Zhang, L. Wen, H. Peng and H. E. Tseng, "Quick Learner Automated Vehicle Adapting its Roadmanship to Varying Traffic Cultures with Meta Reinforcement Learning," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 22 September 2021.
- [50] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu and M. Sun, "Tactics of Adversarial Attack on Deep Reinforcement Learning Agents," *IJCAI 2017*, 8 March 2017.
- [51] S. Zhang, H. Peng, S. Nagesh Rao and H. E. Tseng, "Generating Socially Acceptable Perturbations for Efficient Evaluation of Autonomous Vehicles," *2020 IEEE/CVF Conference*

on *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1341-1347, 19 June 2020.

- [52] S. Gu, J. G. Kuba, M. Wen, R. Chen, Z. Wang, Z. Tian, J. Wang, A. Knoll and Y. Yang, "Multi-Agent Constrained Policy Optimisation," 6 October 2021.
- [53] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang and Y. Yang, "Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning," 23 September 2021.
- [54] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta and J. G. Araújo, "CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms," *Journal of Machine Learning Research*, 16 November 2021.
- [55] C. Fox, "Transport truck off ramp accident," CTV Toronto, [Online]. Available: <https://toronto.ctvnews.ca/charges-laid-after-transport-truck-hauling-fruit-rolls-off-ramp-from-hwy-427-1.3661792>.
- [56] "Collision closes Highway 407 exit ramp," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/collision-closes-highway-407-exit-ramp-to-bayview-1.653405>.
- [57] "A car crash on the Kennedy Road exit ramp off of Highway 401," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/at-least-four-cars-involved-in-kennedy-rd-crash-1.373965>.
- [58] "Exit onto Highway 407 - an accident site involving one of the motorcycles," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/new-video-shows-motorcycles-speeding-along-hwy-400-prior-to-crash-1.6578528>.
- [59] "Highway 401 eastbound ramp acceleration," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/2024/4/3/siu-clears-opp-officer-in-connection-with-collision-that-serious.html>.
- [60] "Collision at the intersection of Sandalwood and Dixie Road," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/driver-in-fatal-2021-crash-in-brampton-ont-turns-himself-in-opp-1.6735621>.

- [61] "Three vehicles were involved in an initial crash at the intersection," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/2023/9/13/3-car-crash-in-west-toronto-sends-6-people-to-hospital.html>.
- [62] "High-speed collision," CTVnews, [Online]. Available: <https://toronto.ctvnews.ca/2022/3/17/two-ontario-students-critically-injured-in-deadly-texas-crash-st.html>.
- [63] "Head-on collision," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/two-of-3-victims-in-caledonia-crash-were-heading-home-from-laser-tag-1.3347037>.
- [64] "Multi-vehicle crash," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/mother-and-son-killed-in-multi-vehicle-crash-in-caledonia-1.1061839>.
- [65] "Collision with a vehicle making a U-turn," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/woman-describes-twin-sister-killed-in-motorcycle-crash-as-her-gift-of-life-1.3497024>.
- [66] "Jeep Wrangler in attempt to make a U-turn collided with a black Honda pick-up truck," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/suspect-in-fail-to-remain-crash-in-entertainment-district-surrenders-to-police-1.4021405>.
- [67] "While making a u-turn Jeep Liberty t-boned the Mazda," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/fiery-richmond-hill-crash-leaves-two-dead-1.338858>.
- [68] "Motorists making illegal U-turns to access Don Valley Parkway," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/police-cracking-down-on-motorists-making-illegal-u-turns-to-access-don-valley-parkway-1.4547248>.
- [69] "Two cars tried to merge into the same lane near Evans Avenue at about 8 p.m., causing a collision," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/young-boy-seriously-hurt-in-hwy-427-crash-1.223250>.

- [70] "Entering the ramp at a high speed and colliding with the guard rail," CTVNews, [Online]. Available: <https://toronto.ctvnews.ca/labour-day-highway-blitz-off-to-inauspicious-start-1.431633>.
- [71] "Github issues related to HighwayEnv: List of Trained Models," Farama-Foundation, [Online]. Available: <https://github.com/Farama-Foundation/HighwayEnv/issues/209#issuecomment-912463146>.
- [72] A. Engineering, "Altair Radioss™ Delivers Robust Results Quickly for Crash-Safe Vehicle Designs," [Online]. Available: <https://pdf.aeroexpo.online/pdf/altair-engineering-gmbh/altair-radioss/171591-8978.html>.
- [73] Statistics Canada , "The Ultimate List of Canada Driving Statistics for 2024 (Source: <https://tests.ca/driving-statistics/>)," Tests.ca, 202.
- [74] E. Leurent, "highway-env Documentation," Farama Foundation, 2018.
- [75] M. Koren, S. Alsaif, R. Lee and M. J. Kochenderfer, "Adaptive Stress Testing for Autonomous Vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, China, 2018.

Appendix A

Sample Appendix

Mentioned in a Future work section. **ADDING NEW TYPES OF TRANSPORTATION**

The selected environment has two main types of transport, the functionality of which is modified to suit the needs of the person using the environment.

For the ‘lane_keeping’ environment, the environment provides an additional type of vehicle, a bicycle, with assembled characteristics for its use on the road and a car, for example, the characteristics of screen below.

```
class BicycleVehicle(Vehicle):
    """
    A dynamical bicycle model, with tire friction and slipping.

    See Chapter 2 of Lateral Vehicle Dynamics. Vehicle Dynamics and Control. Rajamani, R.
    """

    MASS: float = 1 # [kg]
    LENGTH_A: float = Vehicle.LENGTH / 2 # [m]
    LENGTH_B: float = Vehicle.LENGTH / 2 # [m]
    INERTIA_Z: float = (
        1 / 12 * MASS * (Vehicle.LENGTH**2 + Vehicle.WIDTH**2)
    ) # [kg.m2]
    FRICTION_FRONT: float = 15.0 * MASS # [N]
    FRICTION_REAR: float = 15.0 * MASS # [N]

    MAX_ANGULAR_SPEED: float = 2 * np.pi # [rad/s]
    MAX_SPEED: float = 15 # [m/s]
```

Exploring several types of roads and their actors (interacting elements such as cars, pedestrians, etc.), many resources categorize actors according to the number of fatalities. For example, the Ontario Traffic Research resource provides information about road actors, a figure below.

Category	Number of Fatalities	Percentage of Total Fatalities*
Pedestrians	116	22%
Speed-Related	110	21%
Drinking and Driving	98	18%
Large Trucks	85	16%
Unbelted Occupants	82	15%
Motorcyclists	81	15%
Drug-Impaired	77	15%
Inattentive Driving	74	14%
Cyclists	23	4%
Senior Drivers	17	3%
Young Drivers	9	2%

* Many fatal crashes involve more than one of the factors listed. These percentages do not add up to 100.

Based on this information, you can add additional vehicles in order to project and train the model with new indicators (new road actors).

Glossary

Blackbox – Refers to the unknown component, in this thesis the blackbox has a meaning of not being known by the tester algorithm or mode of the target vehicle, on which interactions are performed. For example, the adversarial attack happening using the pretrained attacker vehicles onto the target vehicle configuration of which can be changed by the user or manufacturer or a company that wants to use virtual environment to test their ego-vehicle.

Ego vehicle – Alternatively, **autonomous** or **unmanned vehicle**, refers to the vehicle that has particular level of automation to support driving. There are 5 levels of automation that are described in Chapter 2 of this thesis.

Stress-testing – Referred to the testing approach which meant exhausting the initial algorithm of the system tested to define any bugs or error appearance before being launched for use. In the automotive industry, stress testing involves subjecting vehicles and their components to intense, real-world conditions to identify potential points of failure and ensure durability.

Actors of the road – In the thesis, we will use the terms **agent**, **actor**, or **vehicle** interchangeably to refer to the entity of the road (such as car, target or attacker vehicle).