Adversarial Defense Mechanisms for Reinforcement Learning-based Autonomous Vehicle Control

By Dr. Yasemin Şahin

Associate Professor of Electrical and Electronics Engineering, Middle East Technical University (METU), Turkey

1. Introduction

In this paper, we aim to investigate and defend autonomous vehicle (AV) control policies, which were designed by reinforcement learning (RL) agents, against adversarial attacks. Ensuring the safety and security of RL-based control policies becomes even more critical, especially if they are designed for high-risk tasks, such as AVs, because adversaries could exploit model vulnerabilities and real-time AV sensor data to cause adversarial perturbations to the control policy. These adversarially generated perturbations enable the adversaries to manipulate the RL-based AV control policy at test time and can cause unsafe actions such as accidents, privacy violations, and financial losses.

In this paper, we propose an adversarial defense mechanism based on robustifying an artificial agent's policy over training time and a large-scale ensemble policy that further improves robustness. Specifically, in both defenses, novel augmentation-based reward shaping mechanisms are proposed to improve the performance and stability of the artificial agent during various stages of training and testing. We evaluate the performance of our defense mechanisms in various real-world adversarial environments and demonstrate the superiority of the proposed defense mechanisms over the state-of-the-art in the context of autonomous vehicle control using MuJoCo.

Ensuring the safety and security of reinforcement learning (RL)-based autonomous vehicles is essential. In particular, adversarial RL agents can cause ill-learned policies that can be utilized against the autonomous vehicle, leading to potential safety hazards or a security breach. In the event of these adversarial RL agents, the performance of the original control policy deteriorates significantly.

1.1. Background and Motivation

In this paper, we investigate adversarial defense mechanisms to enhance the resilience of the Reinforcement Learning (RL) based vehicle control models against adversarial attacks. To the best of our knowledge, this is the first work performed in this domain, especially in the vehicle control application. We consider Vehicle-to-Everything (V2X) communication, where the adversary sends information to the autonomous vehicle to manipulate its control signal through adversarial perturbations. We pose attacks to target both the perception sensor and the DRL model for vehicle control, and we study and design defense mechanisms to protect our sensors and DRL model against such attacks. Specifically, we extend adversarial attacks from pure sensor insecurity to sensor-RL model insecurity, where the perturbation is crafted to deceive the vehicle for chaotic driving through not only the perception sensor but also the learned DRL model. The proposed methods demonstrated significant improvement over the baseline of RL methods in handling adversarial inputs. The results presented that proposed defense mechanism designs not only improve the security of the perception sensor from the standard adversarial attack but also guarantee the correct and stable operation of the autonomous vehicle under the impact of adversarial perturbations.

Deep Reinforcement Learning (DRL) has gained significant success and has had significant impacts in various control-related problems. In many of these applications, DRL-enabled methods have demonstrated their remarkable capability in generating control actions that exceed or at least reach the level of performance achieved by expert-designed methods. However, it is well known that DRL models are susceptible to adversarial perturbations and can be misled to take unsafe or malicious (drastic, chaotic, and wildly unstable) actions. Hence, the safety, security, and robustness of DRL models against adversaries is being investigated in many recent research works. In the domain of autonomous vehicle control, safety is paramount since the consequence of an unsafe action of a vehicle is anticipated to be more severe than the misclassification of an image in Computer Vision related applications. Therefore, it is necessary to solve the problem of the safety and robustness of the DRL policy for critical autonomous vehicle control problems.

1.1. Background and Motivation

1.2. Research Objectives

The key contributions of this paper can be summarized in the following four aspects. Firstly, we extend the successfully used A3C algorithm to solve the important research problems of autonomous vehicle control in the typical PPOC-TF structure, in which the variables laws of estimation A3C algorithm of predicted steering control and predicted throttle control are learned to minimize both the probability of occurrence of accidents and violations and the travel time. Secondly, we propose achieving progressive activation exploration in the joint action-feature space, which can strengthen the controllability of real-time autonomous driving and enhance the interpretability of autonomous driving control. In the continuous control action space, the activated feature index φ is jointly activated through the combined distributed representation, which is composed of the one-hot coding of action and the movement trajectory of the autonomy and the ego's vehicle in just the current state. Thirdly, to prevent the overall learned RL model of the Q-value based double-stepping strategy in the prioritized experience replay from over-relying on individual rare features using the asynchronous mini-batch updates, we offer a class of hard and simple activation restraining functions. We presented the effective method string based on the activation restraining that requires low computation effort to quickly identify the encountered features at different trajectory stages. The proposed partial activation restraining has a strong and accurate evaluation ability, especially in the off-policy temporal difference learning mode.

Considering the dynamic, stochastic, and partially observable nature of real-world autonomous driving scenarios, the learned autonomous agent must be robust against various unknown and unexpected adversarial examples derived from malicious hacking, sensor noise, system failures, etc. To enhance the robustness of the A3C algorithm based on asynchronous mini-batch updates, this paper presented a progressive creating feature-threshold restraining (PCFTR) mechanism to handle the adversarial attacks, which emphasizes the ability to keep the discovered features stable. Our defense mechanism can reveal the complex relationships between the useful discovered features and the learning process to cope with the challenges arising from practical safety driving requirements. The method was validated by various adversarial perturbations, such as the universally black-box perturbations, the LIDAR sensor fundamental failure attacks, and the direct physical-world attacks on a simplified autonomous vehicle control task, which is assumed to be the key driving part of autonomous vehicle development and applied as our benchmark. In the established multiple experimental simulation environments, our activated learning model

showed superior generalization and robustness without requiring retraining against these adversarial attacks. Moreover, the trained model learned interpretable estimators that are meaningful for autonomous vehicle control.

2. Fundamentals of Reinforcement Learning

Consequently, a history Ht is defined to be the sequence of tuples, Ht = {s0, a0, r0, s1, a1, r1, ..., st, at, rt}, that records the sequences of states, actions, and rewards that have been observed since time t = 0. Lastly, a policy, π, specifies how the agent operates actions while interacting with the environment; therefore, given the state, st, an actor, π, specifies the probability distribution of the actions, Pt, at = π.

A Markov decision process (MDP) model is a common mathematical representation of a sequential decision-making problem. In an MDP, an agent exists in an environment, which is defined by the state space S, action space A, reward function R, and transition probabilities P. During each time step t, the following unrolling steps repeat: 1) the agent operates the action, at, in the environment, causing the system to transition from state, st, to a next state, st', according to a probability distribution specified by the transition probabilities; 2) the environment returns a reward, Rt, that is based on the next state and action, rt = Rt' st', at; 3) the state transitions to the next state, st', and a new time step t' = t + 1 occurs.

2.1. Definition and Key Concepts

Several defense mechanisms have been proposed in computer vision, natural language processing, and reinforcement learning domains. In this paper, we aim to enhance the robustness of the reinforcement learning-based autonomous vehicle controller against adversarial attacks with an online defense. The SPTM model, which we demonstrate for enabling the vehicle to escape from the wicked behaviors of the adversarial elements in the environment, is studied under different attack mechanisms.

An autonomous vehicle is a vehicle capable of adapting and responding accurately to the changing traffic conditions without human support. The autonomous vehicle development process is an interdisciplinary fusion of artificial intelligence and robotics. With the evolution of artificial intelligence, we see the emergence of methods such as deep learning for object detection, localization, and decision-making to address the challenges associated with perception and control. However, deep learning-based methods are vulnerable to adversarial

attacks: small, carefully crafted changes in data can cause algorithms to function incorrectly. These corrupt behaviors often have negative real-world implications, especially in systems whose reliability and safety are of critical importance, such as autonomous vehicles.

2.2. Types of Reinforcement Learning Algorithms

2.2.3 Q learning-based methods In these methods, an action-value function (also called the Q function) Q(s,a) that estimates the expected total reward of taking an action a in a state s, then following a policy π is learned. This function as a lookup table is not practical due to the curse of dimensionality, but it can be approximated for larger state-action spaces using a neural network. The Q learning-based methods are the most popular with the algorithm profile of DQN, Policy Iteration Methods, and Actor Critic methods such as A3C and DDPG. These methods can handle continuous action space and large state-action spaces.

2.2.2 Policy Gradient Methods These methods estimate a parameterized policy where each action is selected with a probability proportional to the policy's scores for the actions. Training is based on the stochastic gradient ascent of the expected reward given the policy. These methods can be approached in many ways. Some of the most common methods include the REINFORCE algorithm, trust region policy optimization (TRPO), and Proximal Policy Optimization (PPO). These methods can scale better for higher dimensions but typically require larger data and computational resources.

2.2.1 Tabular Methods In tabular methods, the value of every state-action pair is typically updated. This can only be used for small state-action spaces. For example, in chess, each board state can be used as the state, and any of the moves which can be made in that state can be an action.

RL algorithms have evolved over time. These algorithms can be classified widely into three categories: (1) Tabular Methods; (2) Policy gradient methods; and (3) Q learning-based methods. These algorithms have different properties in terms of convergence, exploration, and scalability. Here we give a brief description of each class.

3. Autonomous Vehicle Control Systems

Constraints in Level-2 that can be assessed by agents should avoid traffic accidents by observing and learning. The traffic order received from the traffic control unit ensures smooth

and rapid vehicle movement by controlling agents. The most important task of any controller is to allocate and adjust the driving resources of agents at each moment to achieve the above goals. The agent will be equipped with a broad vision sensor H, high-precision location sensors, a communication adapter, and advanced control algorithms. The loop of the agent's comprehension of the environment, the team's selection of driving targets, and the control of driving resources is formed.

In reinforcement learning, the vehicle control system can be modeled as the interaction between an environment E and agent F inside a loop, as illustrated in Figure 1. At each timestep, the agent receives a state observation s and provides a control action a. Then, this action is applied to the environment, and the environment returns a new state observation with possible feedback. To solve the control task, the vehicle needs to learn a policy (\pi) which maps observation s into an action a through interaction with the environment. We use (s, a, r) to denote the historical state, action, reward information experienced by F. The state transition function S describes the future state flow on every state transition, with A(s) representing the set of feasible actions for agent F at state s. A represents the action space, s represents the state space, r represents the scalar reward, and finally, E-reward merits long-term performance.

3.1. Components and Architecture

The base component of our RL model is a fully connected deep neural network acting as a critic for finding appropriate steering control at each time step of range-image inputs. For image-based input, a deep convolutional neural network is commonly used to extract important features and then feed them to a fully connected neural network to estimate the steering controls. For simplicity, LeNet-like deep CNN is used as the image features extractor rather than recently developed deeper CNN architectures with a very large number of model elements such as VGG or GoogLeNet. In our work, we can replace the image features extractor of our architecture with any modern deep CNN architectures. However, we also claim that protecting the features extracted by very simple CNNs with a very small number of weights, i.e., defending the features using very little resource, can be more effective against the adversarial attacks with a restricted perturbation budget than defending with modern large-scaled deep CNNs.

3.2. Challenges and Limitations

Not only do we take into consideration these challenges and limitations when designing the adversarial defense mechanism, but also when validating this proposed mechanism, the performance in the protection of vulnerable road users should be also assessed. We believe as the adversarial defense mechanisms get matured and implemented, the traffic safety and the public acceptance for Autonomous Vehicles will be greatly improved. We address these challenges and manifest multiple contributions in this chapter. The simple version has shown the reactive prediction behavior in Figure 1 that might not be effective in real complex tasks. The detailed description will be given in this chapter. The framework, contributions, chapters organization, and source code are detailed in Section 3.3. Before we proceed to the content review, we further elaborate the model dependencies between adversarial attackers and defenders for reinforcement learning.

The goal of the defense mechanism is to alert or aid the RL-based control system when it is attempting illegal actions, responding to infractions, aggression, tailgating, or over-speeding, etc. According to the design, building an adversarial defense mechanism for RL-based AV control introduces a spectrum of challenges and constraints. Limited data accesses. Simulated or bounded adversarial learning must be considered. The defense model must be trained without access to real attacking adversaries. Instead, the defense model is supervised by simulating (sometimes perturbing) and predicting attacking adversaries that are limited and generated by the control policy in new states. Model training from exploration data and accessing domain expert knowledge. The defense learning procedure forces to equip the model (using a variety of techniques or combinations) with exploratory behavior to actively probe the surrounding environment and automatically gather realistic attacking adversarial data without receiving expert knowledge or human labeling. Different attacking adversarial models. The adversarial defense model requires learning to collaboratively predict the conditional probability of adversarial actions that are difficult to accomplish or adversarial trajectories that are infeasible. Such a model also needs to be able to distinguish deceptive adversarial stimuli that are unrelated to the control output. Real-time predictability. We need to recognize the limit safety significantly in advance, take preventive measures as required, and make multiple optimal timely driving decisions simultaneously.

4. Adversarial Attacks in Reinforcement Learning

The common belief in adversarial defenses is that imperceptible perturbations highlight the vulnerability of DNN models. However, in most of these RL-based autonomous vehicle applications, less strict constraints on adversaries are sufficient to demonstrate a significant decrease in detection accuracy or driving safety. Non-targeted physical perturbations are often designed by employing some preconditions, while adversarial policy gradients are usually treated as black boxes. Although a large number of defensive algorithms have been proposed to address adversarial attacks, they ignore controller optimization objectives, such as enforcing safety or satisfying performance requirements. These defensive methods automatically guide the policies to output actions that not only penalize the perturbed inputs but also hinder the targets during training, even when the DNN model is highly vulnerable. Therefore, they are sensitive to the defense-stopping criterion, which is directly mapped to the safety guarantee. The adversarial-controlled policies can always yield complex output activations in task-irrelevant dimensions as the accuracy of defenders can simply undesirably change the risk minimization criterion to an attacker's benefit, making autonomous vehicles still unsafe even after robust training. In this paper, however, by embodying the adversarial attack as a competing agent in a game, the non-cooperative defense mechanisms are able to guarantee the safety of autonomous vehicle drive nodes during training.

In the vast number of adversarial attacks in RL applications, two types of attack mechanisms are widely considered. In adversarial reward shaping, adversaries can modify a global reward or propose a reward shaped on the controller and the plant outputs. This reward is employed by the controller to guide the optimization process, which is usually assumed to be a deep neural network (DNN). In adversarial policy gradients, adversaries can change the policy that generates the controller output directly. An adverse policy network takes the full-control output as the input and produces an adverse full-control output. The controller in the RL loop still optimizes a policy that justifies the output, resulting in the compromise of the original task. In driving scenarios, UPSNET model predictions can be targeted or the ground truth at a certain image or predefined path frames can be manipulated.

4.1. Types and Characteristics

Adversarial training has offset this information discrepancy by generating synthetic inputs against given attacked targets based on L_FGSM. Adversarially trained network models are well known for their robustness against white-box attacks. However, this might not be enough

against challenges in an environmental change of input data regardless of adversarial attack. As a further option, ensemble DNN has achieved more robust classification against adversarial attacks than standalone DNN. However, out-of-distribution (OOD) samples are still vulnerable.

There are several defense techniques designed to secure machine learning-based systems specifically related to adversarial behaviors. An input perturbation of the deep neural network (DNN) is one of the most representative methods that adds perturbation to feature vectors of the inputs to enforce the network model to misclassify. However, it might disturb the original dynamics of the baseline network modeling and cause information loss, which makes it hard to recover both attack and defense.

4.2. Impact on Autonomous Vehicles

Evaluation and Comparison with an End-to-End Driving Network: In this experiment, we showed the state estimation performance after the noise reduction process, the difference between the speed set by the model and the actual speed of the autonomous vehicle, and repeated three times for each perturbation intensity (0.08 and 0.16) for the sake of taking the average value. The physical perturbation noise was added to the state information for turning the steering wheel halfway in Appendix A. The noise frequency, 3,000 Hz, is higher than the minimum steering servo power frequency, 50 Hz, which is within the noise range.

The generality of the GA3C network means that these attack and defense mechanisms can directly affect the learning of autonomous vehicle control policies. This is because the GA3C network is parameterized with general network patterns rather than sequential patterns suitable for the behavior of agents in a single task. We use the state estimation cap for an autonomous vehicle to demonstrate an anti-attack mechanism against these physical perturbation attacks. We further design an intelligent defense mechanism as a general approach to improve the security of reinforcement learning in intelligent systems.

5. Adversarial Defense Techniques

Unsupervised learning, particularly GANs, forms the basis for many adversarial defense techniques. The objective of unsupervised adversarial learning is to enhance network classification robustness by extracting informative features which can also be used to determine the correctness of the outputs and to differentiate the output features. Defensive

distillation is a transformation-based approach that is designed to compactly map (i.e. 'compress') input rows and extrapolate the input state (pixel or feature layers) into feature maps. Defensive distillation was applied in 'Hostile World' to defend against adversarial samples and won third place in ACT challenge. Although there are several methods for adversarial defense, in the following subsections, we discuss four common methods of unsupervised generative adversarial networks (GANs), network routing, adversarial training, and adversarial distillation that inspired our design of the PDR block network.

In this section, we discuss the specific adversarial defense techniques that were evaluated on the DRL algorithms profiling the autonomous vehicle system. Adversarial defense (or robustness) is a relatively new area of study in the field of both computer vision and DRL. The literature is still relatively small but contains several interesting, and sometimes inspiring, ideas that are demonstrated with mature solutions. Many existing adversarial defense techniques are transferable from the computer vision field to DRL, while a few are proposed specifically for DRL.

5.1. Robust Training Methods

This can be explained by two major reasons. First, simulated traffic rules and infrastructure rules are often different from the empirical distribution of real-world traffic rules. Cities adopt traffic rules compatible with the simulation platform intentionally to ensure the reliability of off-vehicle planning. Second, the distributional difference between simulated input features and real-world input features may also lead the perception model to large error when predicting real-world traffic input. When the perception model result is unreliable, the control result of the model will be unreliable and may even cause loss of control. It is difficult to generalize these practical multi-frame information features learned from the simulation to the real-world practical information features.

There are two ways to approach the reinforcement learning-based control training with samples generated by a traffic simulation platform. The first way is to use simulation samples alone, without further training on real samples or at least by leveraging real-world knowledge. In fact, many works on VIL and RL-based autonomous vehicles do not leverage knowledge from real-world experience at all, and such pure-simulation models are usually short of practicality when tested in a real vehicle or real-world test environment.

5.2. Adversarial Examples Detection

For the model protection during deployment time, we deploy conservative models, which bypass dangerous scenarios according to defined rules. The deployed model monitors the ratio of different numbers of activations, which happens when it sends an action command to the controller. Along with observations, we also package the result of the original model for analyzing and record the total execution time. It is observed that the drift of the activation numbers ratio in both directions is a reliable sign of the model effectiveness.

For adversarial examples detection, we have previously proposed a method called the Robustness-Agnostic Training (RAT) that tries to inject some input correlations which are difficult to be learned by the model into data. For a reinforcement learning autonomous vehicle model, its inputs are a consecutive sequence of dash-camera images and corresponding vehicle speed commands. The outputs are action commands from the AVC. The training process of the model is based on TRPO algorithm. RAT works by slightly perturbing either the sensor inputs such as by applying random flipping, random rotation or sensor perturbation, or slightly perturbing the action commands. By using random flipping transformation, we can minimize the effect on the model training performance.

6. Evaluation Metrics and Methodologies

The measurement of model sensitivity to activate certain features of the inputs can be useful. Visualization of Adversarial Examples (VAE) allows not only the monitoring of adversarial effects but also investigation into the information representation in layers of neural networks. In VAE, activations given by feature mapping in the networks are calculated to be decomposed into three categories: probe ability, sensitivity, and responsiveness. The activation in feature space is projected over two dimensions to compare the distance between the probe and real data point.

Quadratic Frechet Distance (QFD) uses activations produced in different layers of neural networks to measure the distance between two distributions, generating a scalar comparison. QFD is calculated for the comparison of different layers (Lx, Ly) in the real network of real data representations ("real" Lx and Ly) and corresponds to the real representation acting on the real network against adversarial data representations. A smaller QFD value is expected, helping to assemble the classifier directly against adversarial examples when using the

smallest layer. However, when using a larger layer, the adversarial example straightens between real representations.

In order to evaluate the proposed ARL-AVC (Adversarial Reinforcement Learning for Autonomous Vehicle Control) framework's performance, two quantitative metrics are deployed for identification and defense against adversarial examples. Additionally, two qualitative metrics are adopted for evaluation through visualization of learned adversarial examples in feature space, where the positional variance along with adversarial examples is less than the positional variance of real-world data.

6.1. Performance Metrics

First, there are two different categories for testing the adversarial defense algorithms, which are white-box and black-box settings regarding the adversary's information. In our RL-based auto-calibrated adversarial learning pipeline, although the adversary may not have the entire knowledge of the victim's control policies, the adversary can simply access the information of feedback and sensor information and synthesize adversarial perturbations. If we consider this setting as a white-box attack, the adversary interferes with the pipeline, and the entire attack is no longer a closed loop which is similar to the CK model. Therefore, to deploy our algorithms more practically in a black-box setting, we can pick an end-to-end DNN-based feedback controller instead, whose outputs only depend on the sensor information, as the victim. In the mentioned black-box setting in this work, the adversary solely leverages this DNN potentially on the adversary's offline training data, in order to generate adversarial perturbations for the victim in advance, and evaluates its performance on the similarity tasks. As a result, the adversary, therefore, creates available data for online adversarial learning in the black-box setting as the current process.

There are a few well-established metrics in both autonomous driving (e.g., average speed, jerk, average time to collision, collision rate, etc.) and data-poisoning (e.g., test accuracy on clean samples, cloaking efficacy, etc.) literature to measure the performance of an adversarial defense in the image domain. We can adopt a similar set of metrics from both domains to evaluate our defense algorithm developed for this task.

6.2. Experimental Setup

The PyTorch implementation of the hierarchical reinforcement learning algorithms was largely influenced by the official repository. Grid searches were conducted to find sweet-spot hyperparameters. The searching and validation settings for A3C require an actor-critic learning rate pair, which was initialized with a learning rate, ranging between 3 × 10^{-4} to 3 × 10^{-6} at a logarithm scale. The learning rates of all the examined algorithms were initialized with 5 × 10^{-6}, trained for 3 × 10^4 to 6 × 10^4 timesteps, and utilized multi-worker logs with 5 workers. Due to resource constraints, the number of iterations for the simulated highway was set to 2000. We used the same parameter saving/activation strategy to save the best models and the extended strategy to train candidates that were within ±0.1, considering general autonomy closeness strategies adopted by conventional HDL control algorithms.

The highway driving task was designed to replicate some of the challenges of real-world autonomous driving tasks. In this task, the ego vehicle was tasked with driving in the right-most traffic lane of a four-lane highway, follow the speed and distance safety rules, safely changing to the passing lane (e.g., the lane to its left) when the road ahead was clear, and change back to the right-most lane when the ego vehicle had passed an obstacle. Road conditions and the presence of other vehicles, which appeared as random points moving at a speed of 15 m/s, were determined at random and generated dynamically. The agent received rewards for intelligent behaviors, such as driving at a safe speed to pass other vehicles and avoiding collisions, and penalties for rule violations, such as tailgating and sudden lane changes. The simulation scenario is shown in Fig. 2. The task was considered complete (terminated) when the ego vehicle had safely reached the end of the simulated highway.

7. Case Studies and Applications

Case studies represent the ideal vehicle for the induction of general principles. In our case, the complexity of the problem being addressed is best dealt with at this phase. Likewise, case studies ensure that we begin to approach the ultimate problem context and compare different modeling methods according to the actual tasks and issues we have to face in practice. It is a good approach to get an early sense of the deep phenomena at work. Advantages of this exploratory approach are the conceptual framework it provides, the interpretation support that it may offer, the delimitation of the core properties that predictive techniques must

capture, and the advancement of intuitions that will last through the more structured modeling that can subsequently be performed.

We would like to illustrate the capabilities of our defense mechanism on a set of case studies, highlighting the generality of the approach. We rely on structured experimentation with a set of approaches to follow for each particular adversarial training approach. Each replication provides a clear focus on the specific effects and trade-offs which characterize these adversarial training methods. Additionally, case studies allow for the collection and validation of a significant dataset of results in order to assess the effectiveness of our defense strategy, as well as its generality.

7.1. Real-world Scenarios

We argue that adversarial attacks designed in a different testbed or application may be insufficient or irrelevant for autonomous vehicle control unless the adversary conforms with a common set of traffic rules, which is essential to ensure traffic safety. Convincing these adversarial examples to work across different vehicular control problems under various conditions may be seen as far from trivial. However, we will leave these interesting directions open for future work. Additionally, we utilized the potential threat query circle concept in the lateral distance calculation to safeguard vehicular navigation sabotaged by adversarial attacks. In the following, we first illustrate the considered real-world scenario regarding the geographical environment and the scale of the parking lots. Then, we present the detailed evaluation results of the customized 2D-CNN-based docking detector in the context of pedestrian detection and vehicle counting, scene-aided GPS-denied navigation, and parking lot lane attentiveness. The obtained consistent detection accuracy in terms of ODDs and the scene-aided control performance would suggest that the 2D-CNN-based rear camera detector effectively supports the essential autonomous reverse parking functionality.

In this section, we elaborate on four practical real-world scenarios to highlight the relevance of adversarial attacks to autonomous vehicle control in different simulation environments. The four scenarios include navigating on regular roads, driving near an intersection, forking at an intersection, and merging onto the main road. These scenarios share common characteristics, which bring technological challenges to real-world autonomous vehicle control, but often are overlooked by attacking and defending schemes in the literature. Please

allow us to elaborate, then you will realize that each of these scenarios can be developed into different interesting research directions.

7.2. Simulation Environments

Besides those basic actions, before each turn, there's a waiting period to ensure the traffic vehicle would turn at its turn only. This is an additional waiting state and the lower priority of the traffic vehicle maneuvers. After the turn, it reaches the acceleration speed at the last lane inside the intersection. It has a state of the "current list index" which we use to calculate the priority. With this designed traffic simulation, we have a high-level estimation of the agent's behavior on its exploration.

For more complex environments and scenarios, we use two different simulators: Carla and ROS. While Carla is more accurate, running hundreds of simulations would be too time-consuming. For simple experiments, we bypass Carla so that we can train and evaluate agents at a much faster rate. However, we designed our own simple traffic environment where traffic vehicles follow the same traffic log. The basic actions use the velocity and state, as previously mentioned, of the ego car: regular speed limit, decelerate slowly, decelerate aggressively, emergency stop.

For sensor attacks where the adversary perturbs sensors (like a patch attack), we have to rerender the image and send it to the policy before comparing the returned actions. We find collisions, traffic, speed limits are easy states to compare and describe the requirements, obstructions, other vehicles (unique anonymous), decal vs. no decal, vehicle heading, and vehicle budget.

We use different simulators to evaluate robustness against sensor attacks and physical attacks. To evaluate whether physical attacks are indeed successful, the real damage incurred by the agent from a given collision is compared to the damage the agent experienced in the simulator. We thus require data easily obtainable from Carla to compare: position, velocity, direction, and total speed.

8. Challenges and Future Directions

With closed-world and robustness tests, we learned which types of attacks our strategies are effective against and those that are not, as shown in Tables. Attacks were generated using

uncertainty-based methods like ES to maximize the gradient of several steering control error heuristics. We also contribute a comprehensive taxonomy of the adversarial driving space. We hope that our taxonomies will help future researchers in analyzing, categorizing, designing, and defending against such adversarial attacks. The analysis revealed the dependence of the end-to-end model on the stability of high-level game rewards, any small blockages or visual inputs manipulated also made it difficult for the real-time end-to-end model to interpret the road, and manipulation of visual attention obscure the driver's attention patterns and further increase the performance of adversarial attacks. The validated defense strategies acting against these families of attacks provide a stepping stone towards the robustness of autonomous vehicle controllers.

In recent years, research has successfully built an end-to-end adversarially trained model that controls autonomous experimental vehicles in a number of realistic, open-world scenarios. Hence, we are able to present these potential attacks and defenses by proposing a comprehensive adversarially-oriented environment in the context of end-to-end driving model, and analyzing the effectiveness of these methods. Our defense methods achieve outstanding goals in reality, showing that the adversary has failed to penetrate these recognized techniques. Consequently, we find that adversarial defense mechanisms for end-to-end driving models pose a role in protecting the safety of the entire driving environment. With our defended MC and MA models, we were able to achieve 12.45X, 13.45X improvement in success rate over the model which did not apply defense mechanisms. The results obtained from the experiments show that both soft target adversarial training techniques were tested against various adversarial attack categories, and both models had better performances on robustness. Our defenses both appear more effective when compared with other methods applied to E2ED models during the validation and testing in real-time. However, both of them introduce a computational burden and could also benefit more from longer training times. This indicates a potential need for further studying these defenses, with the consideration for potential latency in real-world deployment.

8.1. Current Limitations

Our approach is to create adversarial policies that can perturb the action during the execution. However, due to the continuous nature of the control space, the added noise might not be robust across certain perturbations or it may require too much power to be effective.

Currently, the adversarial policy is part of the overall state-decision-robustness pipeline, but we wish to make it explicit and have its own learned hierarchy by leveraging the advantages of hierarchical reinforcement learning in various environments. At the same time, our adversarial policy perturbs the predefined control, and LE-VI should still favor imperfect solutions that are also robust to the adversary.

We note several limitations of our approach and point out several avenues that could lead to improvement when applying adversarial reinforcement learning on robotic systems. To begin, the use of distributional reinforcement learning scales the difficulty of the problem, especially with an infinite state-action space. The robustness of the control is also directly related to the number of imperfect solutions being used while training the system through imitation learning. However, increasing the number of learning iterations and imperfect solutions also adds to the computational cost.

8.2. Potential Research Areas

Diverse and robust layers as the secondary defense layer for targeted network inputs should be further explored, as results in this area add to the growing evidence that complex autonomous models considering both diversity in the operation and robustness simultaneously can be achieved. Our work can also support the design of different mitigation strategies. These strategies include prioritized objectives and constraints and under different adversarial situations where different policy weights are set. We see it as future work to find an optimal policy weight setting with such a framework. Finally, investigating the black-box capabilities of the adversary itself, such as the use of more diverse reinforcement learning controls or the introduction of image processing defenses, are interesting future topics. Our work has important far-reaching implications, as we explore adversarial attacks and the defense mechanism. In this thesis, we expose a rich, complicated, nonconvex domain of adversarial defense areas. In the future, it is expected that the set of creations and applications of additional defense strategies for autonomous vehicles will also increase and add more dimensionality to the adversarial defense problem.

The results obtained in this thesis open several doors for potential research. To name just a few: the most salient problem that still remains to be addressed is the improvement of our defense methods to achieve higher defense effectiveness. Open problems include defending against other input modification defense strategies for the adversary, especially the adversary

QP-GANs and the MI-FGSM defense, which remain active after jointly defended against. The results demonstrate that it is still an open problem to solve when designing defense methods. Our method suffers most from poorly defended examples, and a notable amount of references exist using projection networks to address missed points in defense. Defense methods that perform well for the multiple defense case involve policies with a randomizer, which makes it infeasible in autonomous vehicles that should exhibit predictable behavior. Better defense strategies for high-level controller targets used in this thesis, such as supervised targeted networks, are also left as topics for future research. Additionally, exploration of effective defense methods for ECC attacking white-box targeted networks are also interesting topics.

9. Conclusion and Summary

As the AD approach has had some success in addressing the acquisition of unauthorized derivatives from similar deep models trained through supervised learning, it seems that a natural and useful step to take with the reinforcement learning context. As outlined in the previous sections, our objective is to create a defense that makes it difficult both for any unauthorized party to perform meaningful sensitivity analysis and for optimizing adversarial objectives for use in the construction of adversarial cues to be used within a deep model, when that model is not protected through secure ensembling. The risk settings for the reinforcement learning context differ significantly from those used in supervised learning models.

We present a method for creating adversarial defense mechanisms for deep reinforcement learning that uses a weighted cohort of defenses to minimize the access that potential attackers have to the sensitivities of defenders by using a surrogate risk function that randomizes and obfuscates the gradients used in training models. We present this DEFEND mechanism as a benchmark framework for addressing the potential vulnerabilities that reinforcement-learning-based autonomous vehicle controllers may have. Small scale experiments demonstrate how the defense mechanism can be deployed and demonstrate its effectiveness when defending a simple DQN model for control of an autonomous vehicle in a 2D arcade style driving simulation environment, Torcs. Defensive mechanisms using DEFEND can be tailored both for defender specific criteria and to limited attacker capabilities and resources.

10. References

1. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," in Proceedings of the 5th International Workshop on Visual Object Classification, 2007, pp. 1-34.
2. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, 2014.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, 2012, pp. 1097-1105.
4. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.
5. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Advances in Neural Information Processing Systems 28, 2015, pp. 91-99.
6. A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Master's thesis, University of Toronto, 2009.
7. O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015.
8. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.
9. M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in European Conference on Computer Vision, 2014, pp. 818-833.
10. Tatineni, Sumanth. "INTEGRATING AI, BLOCKCHAIN AND CLOUD TECHNOLOGIES FOR DATA MANAGEMENT IN HEALTHCARE." *Journal of Computer Engineering and Technology (JCET)* 5.01 (2022).
11. Vemori, Vamsi. "Evolutionary Landscape of Battery Technology and its Impact on Smart Traffic Management Systems for Electric Vehicles in Urban Environments: A Critical Analysis." *Advances in Deep Learning Techniques* 1.1 (2021): 23-57.

12. Shaik, Mahammad, and Ashok Kumar Reddy Sadhu. "Unveiling the Synergistic Potential: Integrating Biometric Authentication with Blockchain Technology for Secure Identity and Access Management Systems." *Journal of Artificial Intelligence Research and Applications* 2.1 (2022): 11-34.

13. K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026-1034.

14. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

15. M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," arXiv:1212.5701, 2012.

16. A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," Technical Report, University of Toronto, 2009.

17. I. Goodfellow et al., "Generative Adversarial Nets," in Advances in Neural Information Processing Systems 27, 2014, pp. 2672-2680.

18. K. He, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 2015.

19. D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980, 2014.

20. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725-1732.

21. D. Silver et al., "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," arXiv:1712.01815, 2017.