# On Improving the Robustness of Reinforcement Learning Policies against Adversarial Attacks

Yesmina Jaafra, Christophe Bohn, Lucas Schott, Faouzi Adjed, Frédéric Pelliccia, Mehdi Rezzoug

# On Improving the Robustness of Reinforcement Learning Policies against Adversarial Attacks

Yesmina Jaafra

*Expleo France, 3 Avenue des Près, 78180 Montigny-le-Bretonneux, France.*
*E-mail: yesmina.jaafra@expleogroup.com*
*IRT - SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France. E-mail: yesmina.jaafra@irt-systemx.fr*

Christophe Bohn

*IRT - SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France.*
*E-mail: christophe.bohn@irt-systemx.fr*

Lucas Schott

*IRT - SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France. E-mail: lucas.schott@irt-systemx.fr*

Faouzi Adjed

*IRT - SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France. E-mail: faouzi.adjed@irt-systemx.fr*

Frédéric Pelliccia

*IRT - SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France.*
*E-mail: frederic.pelliccia@irt-systemx.fr*
*Apsys, 36 rue Raymond Grimaud, 31700 BLAGNAC, France. E-mail: frederic.pelliccia@apsys-airbus.com*

Mehdi Rezzoug

*IRT - SystemX, 8 Avenue de la Vauve, 91120 Palaiseau, France. E-mail: mehdi.rezzoug@irt-systemx.fr*

With deep neural networks as universal function approximators, the reinforcement learning paradigm has been adopted in several commonplace services such as autonomous vehicles, aircrafts and domestic assistance, which is raising new safety requirements. Indeed, a deep reinforcement learning agent obtains its states through observations, which may contain natural accuracy errors or malicious adversarial noises. Since the observations may diverge from the true environment states, they can lead the agent into taking risky suboptimal decisions. This vulnerability is well-known in computer vision literature where it has been emphasized via adversarial attacks. In terms of defense, various techniques have been proposed, including heuristic and certified methods, mainly to improve the robustness of deep neural networks-based classifiers. It is therefore necessary to propose solutions adapted to this learning challenge faced by reinforcement learning agents. In this paper, we propose two defense mechanisms based on reward shaping and adversarial training as a countermeasure against attacks on environment observations. The results reported from experiments conducted on autonomous vehicles controlled by reinforcement learning policies demonstrate that our approach successfully provide sufficient information to effectively learn the task in the context of highly perturbed environments. Furthermore, the defense mechanisms improve the robustness and generalization capacities of the learning models decreasing risky decisions in the presence of adversarial attacks.

*Keywords*: Deep reinforcement learning, Safe exploration, Observation perturbation, Adversarial training, Reward shaping, Autonomous vehicles.

## 1. Introduction

Among Machine Learning (ML) paradigms, Reinforcement Learning (RL) provides an effective framework for controllers to learn through information collected in real time and to behave appropriately without relying on a perfect model of the environment. With deep neural networks (DNNs) as universal function approximators, Deep Reinforcement Learning (DRL) has been applied to Cyber Physical Systems, where the search space

Proceedings of the 32nd European Safety and Reliability Conference.
*Edited by* Maria Chiara Leva, Edoardo Patelli, Luca Podofillini and Simon Wilson
 *Published by* Research Publishing, Singapore
ISBN: 981-973-0000-00-0 :: doi: 10.3850/981-973-0000-00-0_output

becomes intractable for tabular algorithms. These complex physical entities interacting with the real world are progressively used in commonplace services, such as autonomous vehicles, aircrafts , and domestic assistance (Jaafra et al. (2019)). Given the involvement of costly equipment and human safety, deploying DRL algorithms requires rigorous assessments before their deployment in order to prevent high-risk situations.

More specifically, it has been demonstrated that DRL is vulnerable to adversarial attacks, where an imperceptible perturbation incorporated in the networks input causes inconsistencies in the behavior of the algorithm. Such perturbations in the input space of DNN, may result in a significant variation of the predicted output as shown by Goodfellow et al. (2014). This well-known problem in computer vision literature, has been recently detected for neural network policies generated by state-of-the-art DRL algorithms . Indeed, when an RL agent obtains its current state via observations, the latter may contain uncertainty that originates from unavoidable equipment inaccuracies or malicious perturbations leading to catastrophic failures.

In terms of defenses, various techniques have been proposed recently, mainly assigned to DNN-based classifiers. Heuristic defenses are experimentally validated and operate without theoretical guarantees. Currently, the most common heuristic defense is adversarial training, which integrates adversarial samples into the training phase . Other heuristic defenses mainly achieve input transformations and denoising to alleviate the perturbation in the feature domains. On the contrary, certified defenses are provable methods providing theoretical error-rate guarantee reflecting their lowest accuracy under a well-defined type of attacks Raghunathan et al. (2018).

Motivated by these safety concerns, we consider in this paper the problem of improving the robustness of DRL models applied to autonomous control systems. To this end, we propose the following contributions: (i) Develop a safe reward function to shape and guide agents training using Safety of The Intended Functionality (SOTIF) standard (ii) Train an agent through an online DRL algorithm in presence of adversarial attacks generated by state-of-the-art techniques adapted to RL settings (iii) Present a case study on autonomous vehicles equipped with DNN policies that process environment observations to produce control actions.

The rest of this paper is organized as follows. In Section 2, we introduce the required background knowledge and related work on DRL, reward shaping and adversarial attacks. Section 3 gives a description of our methodology to implement the proposed defense mechanisms and section 4 presents and discusses the experiments results. Finally, we draw conclusions and perspectives in section 5.

## 2. Background and related work

In this section, we discuss the fundamentals of the DRL process and its application. Then, we recall the concept of reward shaping. Finally, we present a taxonomy of adversarial ML attacks and their recent implementations.

### 2.1. *Deep Reinforcement Learning*

The mathematical formulation of RL derives from Markov Decision Process (MDP) in terms of the state, action, reward, and dynamics of the system. At each time step, the agent observes the current state $s_t$ and performs an action $a_t$ based on its current policy $\pi$. After the action is executed, the agent observes its reward $r_t$ and next state $s_{t+1}$.

More formally, an RL task $T_i$ is defined according to the tuple $(S, A, p, r, \gamma, H)$ where $S$ is the set of states, $A$ is the set of actions, $p(s_{t+1}|s_t, a_t)$ is the state transition distribution, $r$ is a reward function, $\gamma$ is the discount factor and $H$ the horizon. A RL setting aims at learning a policy $\pi$ of parameters $\theta$ that maps each state $s$ to an optimal action $a$ maximizing the return of the agent trajectories $R_t = \sum_{k=t}^{t+H-1} \gamma^{k-t}\ r_{k+1}$. The discounted return stated above allows the definition of a state value function $V^{\pi}(s) = \mathbb{E}\ [R_t|s_t = s]$ and a state-action value function $Q^{\pi}(s, a) = \mathbb{E}\ [R_t|s_t = s, a_t = a]$ to measure, respectively, the current state and state-action returns estimated under the policy $\pi$. In high dimensional environments, the estimation of these value functions

becomes intractable. Through non-linear approximation, DRL proposes to control RL agents with function approximations based on DNN to learn the optimal policy or the value/reward functions.

There are three main approaches to solving RL problems. In value-based RL algorithms such as Q-learning, a value function is approximated to select the best action according to the maximum value attributed to each state and action pair. On the other hand, policy-based methods directly optimize a parameterized policy without using a value function. They use instead gradient descents like in the family of REINFORCE algorithms. Actor-critic (AC) methods combine the advantages of the two previous approaches by learning both a policy and a value function in order to reduce variance and accelerate learning (Sutton and Barto (2018)).

### 2.2. *Reward Shaping*

A crucial role is played by reward functions to build driving policies in large-scale applications. However, learning in such sparse settings is complicated and slow. A powerful technique for scaling up RL approaches to handle complex tasks is to transform domains knowledge into complementary rewards. The combination of the original and new rewards is known as reward shaping, inspired by the concept of operant conditioning from psychology discipline. It guides the RL agents to learn faster and more efficiently

The review of shaping methods reveals that, despite the current dominance of reward shaping, the concept itself extends perfectly beyond designing a powerful reward scheme and applies to any supervised transformation of the learning task including environment dynamics, internal parameters and the action space Marthi (2007). Nevertheless, it is the reward scheme that evolves in most modern shaping scenarios, where the learning agent is rewarded for meeting additional subgoals.

More practically, in order to improve the return assignment by making the correct behavior apparent at early stages of training, we apply a shaping function $F$ acting similarly to the native reward function $r$. At each transition, $F$ operates an assessment of the trajectory, and returns a corresponding reward value yielding a more supportive environment to the RL agent. In the most general form, namely the additive form Randløv and Alstrøm (1998), the new environment is defined as a transformation of the original MDP to a shaped MDP with supplementary rewards $(r + F)$. Besides this approach, other important works of reward shaping include the Potential-based reward shaping (PBRS) Ng et al. (1999) and its variants, the potential-based advice (PBA), the dynamic PBRS and the dynamic potential-based advice (DPBA) Harutyunyan et al. (2015). They express $F$ as the difference of potential function $\phi$ defined over a source $s$ and a destination state $s^{'}$ : $F = \gamma\phi(s^{'}) - \phi(s)$.

### 2.3. *Adversarial attacks*

Among security attacks, the malicious input generated by inserting crafted perturbations into the original input is identified as an adversarial example. Formally, given $f(\cdot)$ a DNN classifier, the adversarial example $\hat{x}$ is created by adding an imperceptible perturbation $\delta$ to the initial example $x$. The perturbation $\delta$ is computed by iteratively approximating the optimization problem until the resulting adversarial example is classified in targeted class $c$ where $\hat{x} = x + arg \ \min_{\delta_x} ||\delta||$ until $f(x + \delta) = c$.

The attacks on DRL can be divided into four categories based on the functional components of the DRL process: reward, action, state or model spaces. In this study, we are interested in adversarial attacks on DRL state observations. For example, Huang et al. (2017) provided a first attempt to evaluate the robustness of deep reinforcement learning policies through attacks based on fast gradient sign method. The experiments show a significant decrease in the accuracy of the DRL algorithms. Lin et al. (2017) considered a more complicated case where the adversary is allowed to attack only a subset of time steps, and used a generative model to predict the future states and actions in order to formulate the misleading actions.

The countermeasures proposed to deal with adversarial attacks on DRL include many strate-

gies based on adversarial training, randomization schemes, denoising methods, and provable defenses Ren et al. (2020). In our work, we will rely on adversarial training as defense mechanism. It attempts to improve the robustness and the generalization of DNN policies outside of the standard training manifold by learning a better distribution. Researchers such as Kos and Song (2017) proposed to re-train the model with perturbations generated by adding noises to states and rewards where the attacker is considered to be competing in a game with the agent. Pinto et al. (2017) configured interactions between both the agent and the attacker as a zero-sum minimax objective function where the agent improves its policy by trying to win the attacker.

## 3. Approach

In this section we propose our approach to deal with adversarial attacks on DRL. The strategy adopted to build on the framework robustness includes adversarial training and the design of a safer reward function.

### 3.1. *Task environment*

First, we consider an MDP environment implemented in the 2D open-source simulator for autonomous driving Highway-v0 Leurent (2018). The agent task consists in driving a car on an infinite 4 lanes unidirectional highway. The ego-vehicle controlled by the agent is inserted in the traffic flow with the exo-vehicles that follow the Intelligent Driver Model and MOBIL model Kesting et al. (2007). The setting implemented to collect the environment states consists of the transversal position of the agent $y_{ego}$ and its velocity $v_{ego}$. In order to account for interactions between the traffic participants, we use relative data for exo-vehicles. Hence, the $z_i$ are the positions of the exo-vehicles relative to the agent in the longitudinal direction of the road and $v_i$ their relative velocities. Furthermore, we identify the exo-vehicles according to their topological relation with the ego-vehicle. In this regard, $bl$, $fl$, $b$, $f$, $br$, and $fr$ represent respectively the closest exo-vehicles to the agent, in the back-left, front-left, back, front, back-right, and front-right positions.

The goal of the agent is to drive as fast and as long as possible while avoiding the accidents. The episode ends when the agent has a collision with another vehicle or reaches a time limit. The ego-vehicle can be controlled with a finite discrete set of tactical decisions implemented by low-level controllers: no-action, right/left change lane, accelerate/decelerate.

### 3.2. *Reward shaping for safe driving*

The basic reward function implemented in Highway-env is expressed in 1. It prompts the agent to reach high speed mainly by avoiding collisions.

$$R_v = \max\left(-1, \frac{V_{ego} - \frac{\sum_{i=0}^{n} V_{exo_i}}{n}}{V_{max} - \frac{\sum_{i=0}^{n} V_{exo_i}}{n}}\right) \quad (1)$$

Nevertheless, the speed-based reward function fails to enhance the agent performance in terms of collisions number and episodes duration, notably in high-density traffic scenarios. Since the focus of this work is shed on RL safety, we propose to shape the basic reward function by extending it to a safe design reward in relation with Safety Of The Intended Functionality (SOTIF) and defined in ISO Standards under the reference ISO/PAS 21448:2019 [a]. SOTIF offers a proper understanding of road vehicles safety requirements granting the absence of unreasonable risk caused by the hazardous behavior of the intended functionality. More formally, let $C_1$ and $C_2$ denote two vehicles represented in a follower-leader topology. We define three functions $P(x)$, $V(x)$ and $A(x)$ to describe the position, the velocity and the acceleration of each vehicle, respectively. The three following operational metrics have been considered:

**Time Inter Vehicles (TIV).** Heavy traffic on highways requires the optimization of inter-distances between the follower and leader vehicles in order to provide safety solution in transport. This distance allows to react in case of critical situation taking into consideration the reaction time of the driver and the braking distance of the car. Therefore, it's possible to derive the safety time between two cars $C_1$ and $C_2$ as $TIV = \frac{|P(C_2) - P(C_1)|}{V(C_1)}$.

**Time To Collision (TTC).** TTC has proven to be

[a] https://www.iso.org/standard/70939.html

a cue for decision-making in traffic and a pertinent metric for rating the severity of a conflict. It is defined as the time span left until a collision between two vehicles occurs if the course and speed difference are maintained and no evasive action is taken . The TTC is expressed as $TTC = \frac{|P(C_2)-P(C_1)|}{V(C_1)-V(C_2)}$.

**Braking Time (BT).** It is a main component of the stopping distance in addition to the Driver Reaction Time. The braking distance can be defined as the distance the car will travel once the driver has reacted and applied the brakes. The BT is then specified as $BT = \frac{V(C_2)-V(C1)}{A(C_1)}$.

A new operational reward $R_o$ is designed by integrating the safety measures described above:

$$R_o = \begin{cases} \min(r_f, r_{ft}, r_{bt}) & \text{if change lane} \\ r_{fb} & \text{else,} \end{cases} \quad (2)$$

where $r_f$, $r_{ft}$ and $r_{bt}$ are functions of TTC and TIV defining, respectively, the risk from exo-vehicles in front-same lane, front-target lane and before-target lane. On the other hand, $r_{fb}$ is a function of braking time of the ego-vehicle (same lane). The safe reward function is obtained by shaping the speed-based reward function with the following minimum bounding combination: $R = \min(R_v, R_o)$.

### 3.3. *Adversarial training*

Our second contribution to improve the robustness of RL agent policies consists in adopting an approach focusing on adversarial training techniques. In this paper, we implement a Proximal Policy Optimization algorithm (PPO) Schulman et al. (2017), however, any other actor-critic method could be retained. The critic component yields an estimation of the value function $V^{\pi}(s_t)$ of the state $s_t$ represented by the observation $x_t$ and following the current policy $\pi$ learned by the actor network.

We consider in this work a gradient-based white-box method inspired by Papernot et al. (2016). It aims at attacking the actor network of the RL agent by crafting perturbed observations $\hat{x}$ to replace the actual observations $x$ returned by the environment, and then allow the agent to decide the action $a = \pi(\hat{x})$. In the case of discrete action spaces, an attack is effective if the agent modifies its decision. In the context of a DNN policy, we operate in two steps. First, we propose to use the gradient of loss with respect to every component of the input (i.e. Jacobian matrix) to extract the sensitivity direction. Then a saliency map is computed to select the dimension which generates the maximum error using as little perturbations as possible.

More formally, we aim at crafting a perturbation $\delta_x$ of $x$ that minimizes the probability of the optimal action predicted by the actor using the Jacobian matrix of the probability function learned by the actor policy $\pi$. Let's consider its logit outputs $\prod(x)$ where $\prod_{a_j}(x)$ expresses the probability of the action $a_j$ in the policy output given an input $x$ and $a_j \in A$ the different possible actions. Therefore, determining the appropriate perturbation to attack the observation $x$ consists in solving the following optimization problem: $arg\min_{\delta_x} ||\delta_x|| \;\; s.t. \;\; \pi(x) \neq \pi(x + \delta_x)$.

Furthermore, since it's preferable to change all input features by no more than a small quantity, the perturbation is bounded by a parameter $\varepsilon$ defining the budget that the adversary is allowed to introduce in the input. Building an adversarial example $\hat{x}$ from a given input $x$ requires calculating a perturbation map $H(.)$ with respect to the input features $x_i$, based on the jacobian $\nabla_x \prod(x)$:

$$H(i) = \Big(\sum_{a_j \neq a_d, a_j \in A} \frac{\partial \prod_{a_j}(x)}{\partial x_i}\Big) - \Big(\frac{\partial \prod_{a_d}(x)}{\partial x_i}\Big) \quad (3)$$

where $a_d = arg\max_{a \in A} \pi(a|x)$ is the action corresponding to the highest probability computed by $\pi$ in $x$. The Jacobian matrix identifies how the elements of the environment observation affect the logit outputs of different discrete actions. Precisely, the proposed algorithm iteratively perturbs the feature $x_i$ with the highest value $H(i)$ to affect the logit outputs significantly. The proposed method creates an adversarial observation $\hat{x}$ which reduces the probability of the selected action $a_d$ and increases the probability of all other actions. This process is repeated until $\pi(\hat{x}) \neq \pi(x)$.

## 4. Simulation experiment

In this section, we first describe the simulation configuration used to run experiment scenarios

for the RL agent in Highway-v0 simulator. Then we depict and interpret the results of experiments conducted to evaluate the defense mechanisms.

### 4.1. *Experimental design*

The experimentation is based on the PPO algorithm, which performs comparably to state-of-the-art RL approaches while being much simpler to implement and tune. The agent is trained in the $2D$ open-source simulator for autonomous driving Highway-v0 described in section 3.1. The perturbations applied on the dynamics of the environment imply the velocities and positions of the exo-vehicles as introduced in section 3.3. For the experimental setup, hyper-parameters tuning is performed following a holdout validation methodology which consists in holding out part of the training set used to evaluate several candidate models and then selecting the best one (for more details, cf. Géron (2019).

To quantify the safety and effectiveness of the proposed robust RL approach in both training and testing process, we use three metrics that evaluate the average cumulative return, the episode length and the average velocity of the ego-vehicle. The metrics are computed under 10 rollouts average with a duration of training and testing respectively equal to two millions and fifty thousand steps per experiment. The agent purpose is to reach the highest possible speed avoiding collisions. Each episode terminates when the agent has a collision with another vehicle or reaches a time limit. The results are reported in the form of graphs for training to apprehend the evolution and convergence of learning, and boxplot for testing for a more global and final vision of the performance.

The experiments evaluating the proposed defense mechanisms involve a set of models and scenarios that are defined in the following.

**Unsafe model.** Policy of the RL agent that has been trained using the basic reward function ($R_v$).
**Safe model.** Policy of the RL agent that has been trained using the safe reward function ($R$).
**Attack_Attack.** Main scenario where the safe agent is trained and tested in the presence of adversarial attacks. **NoAttack_Attack.** Witness scenario that specifies a safe agent trained with no adversarial attacks and tested with attacks. **Attack_NoAttack.** Complementary scenario to assess the generalization capacities when the safe agent is trained with adversarial attacks and tested with no attacks. **NoAttack_NoAttack.** Standard scenario where the safe agent is trained and tested with no attacks.

### 4.2. *Results*

The evaluation of our contributions consists in analyzing to what extent can the safe reward function and the adversarial training improve the policy of the RL agent in the presence of perturbed observations.

#### 4.2.1. *Training with safe reward function*

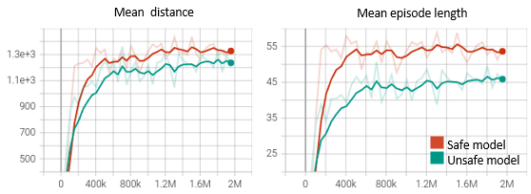


Fig. 1. Safe Vs Unsafe models trained without attack: Average traveled distance across episodes (left), average episode length (right).

First, we evaluate the global performance of the safe reward function in experiment scenarios without attack. Training results are depicted in figure 1 where we can see through the curves of average episodic length and traveled distance that the safe function improves drastically the performances of the agents, compared to the basic reward function.

We can also state that the safe model reaches higher episode length values denoting an improved capacity of collision avoidance. This interpretation is compliant with the reward shaping effort fostering more safety in the interaction between the neighboring vehicles as described in section 3.2. The testing results presented in figure 2 confirm the conclusions stated above. Furthermore, we observe that the mean velocity of the safe agent (90.1 km/h) is slightly lower than the unsafe one (99.9 km/h) but is still very appropriate to the highway context and comparatively higher than exo-vehicles average speed (70.3 km/h). This means that the safe policy has been able to operate

an efficient tradeoff between higher velocity and safer driving decreasing the risk of collisions.

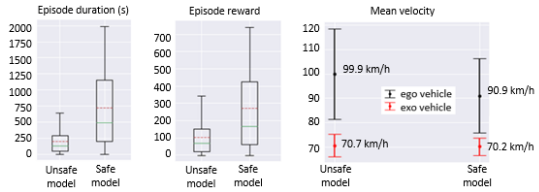


Fig. 2. Boxplots of Safe Vs Unsafe models tested without attack, from left to right: average episode length, average cumulative episode reward and average velocities across episodes.

In figure 3(a), we assess the efficiency of the safe reward defense mechanism in the training environment over eight experiments with a different intensity of disturbance $\varepsilon$ applied in each one. The major finding is that the safe model is more robust and resistant to the increase of the parameter $\varepsilon$. Indeed, the metrics curves of the unsafe model are almost flattened starting from $\varepsilon = 0.1$. In figure 3(b), we expand the y-axis (returns) of the scenario run with an attack intensity $\varepsilon = 0.5$.

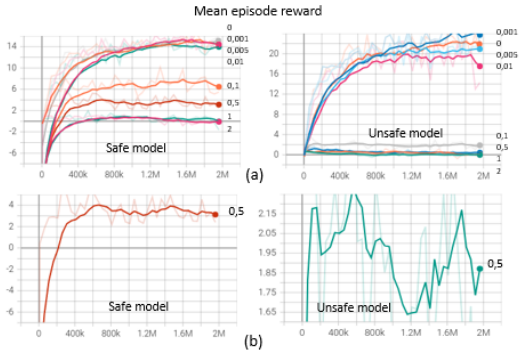


Fig. 3. Adversarial training: (a) Safe Vs Unsafe models for all $\varepsilon$ values (b) Safe Vs Unsafe models for $\varepsilon = 0.5$.

While the safe model is showing a typic case of convergence, the unsafe model curve points out a failure of the learning process. This is probably due to the fact that the basic reward function $R_v$ is unable to provide sufficient information to learn the problem in the context of highly perturbed observations.

#### 4.2.2. *Performance of adversarial training*

The evaluation of the second defense mechanism is depicted in figure 4, where we consider policies performance in four different scenarios emphasizing the contribution of adversarial training in improving RL agents robustness and generalization capacities. Some interpretations can be given in this respect. In the following, the figures are presented in terms of episode reward (in first position) and episode length (between brackets).

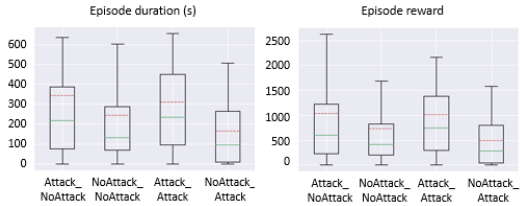


Fig. 4. Boxplots of robustness test scenarios: average cumulative episode reward (left), average episode length (right).

**Robustness.** The boxplots related to scenarios Attack_Attack and NoAttack_Attack show that the agent trained and tested with adversarial attacks is substantially behaving better than the attacked agent trained in standard conditions. This defense mechanism offers 85%(100%) of room for improvement. Thus, we can confirm that the adversarial training prevents drastic failures of RL agents in presence of attacks against environment observations. Practically, the safety aspect is visible in the decreasing of collision risk expressed by higher expectations of episode length.

**Generalization.** This aspect is analyzed in 2 steps. First, let's compare NoAttack_NoAttack and NoAttack_Attack scenarios. In the case of standard training, the policy performance is decreasing when tested in an environment different from the training one. Concretely, the model trained without attack loses 47% of average reward (50% of episode length) when it's tested in an attacked environment. This can be explained by the inherent feature of deep learning policies which are sensitive to significant change in data distribution Lake et al. (2017). On the other hand, we raise almost the same performance of Attack_Attack and Attack_NoAttack scenarios reflected by a small improvement of 9% (3%) when training with attacks then testing in standard conditions. Hence, the defense mechanism of adversarial training not only enhances the robustness, but also the generalization capacities of RL policies.

## 5. Conclusion

Despite its great advances, DRL is vulnerable to adversarial attacks, which prevents its deployment in real-life critical systems. This problem has directed our concern to develop defense mechanisms based on reward shaping and adversarial training. The results reported for a case study conducted on autonomous vehicles are promising. The reward shaping has successfully provided sufficient information to accelerate training convergence in the context of perturbed environment. Furthermore, the adversarial training has specifically fostered the robustness and generalization capacities of the obtained models in the presence of attacks. An important direction of future work is to implement learnable adversaries, which we assume to be more harmful and susceptible to demonstrate higher resilience to defense mechanisms. We also intend to enhance the naïve application of adversarial training by reformulating the defense strategy as a modified MDP.

## References

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".

Goodfellow, I. J., J. Shlens, and C. Szegedy (2014). Explaining and harnessing adversarial examples. *arXiv:1412.6572*.

Harutyunyan, A., S. Devlin, P. Vrancx, and A. Nowé (2015). Expressing arbitrary reward functions as potential-based advice. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 29.

Huang, S., N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel (2017). Adversarial attacks on neural network policies. *arXiv:1702.02284*.

Jaafra, Y., A. Deruyver, J. L. Laurent, and M. S. Naceur (2019). Context-aware autonomous driving using meta-reinforcement learning. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 450–455.

Kesting, A., M. Treiber, and D. Helbing (2007). General lane-changing model mobil for car-following models. *Transportation Research Record 1999*(1), 86–94.

Kos, J. and D. Song (2017). Delving into adversarial attacks on deep policies. *arXiv:1705.06452*.

Lake, B. M., T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman (2017). Building machines that learn and think like people. *Behavioral and brain sciences 40*.

Leurent, E. (2018). An environment for autonomous driving decision-making. *GitHub*.

Lin, Y.-C., Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun (2017). Tactics of adversarial attack on deep reinforcement learning agents. *arXiv:1703.06748*.

Marthi, B. (2007). Automatic shaping and decomposition of reward functions. In *Proceedings of the 24th International Conference on Machine learning*, pp. 601–608.

Ng, A. Y., D. Harada, and S. Russell (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, Volume 99, pp. 278–287.

Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami (2016). The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE.

Pinto, L., J. Davidson, R. Sukthankar, and A. Gupta (2017). Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826. PMLR.

Raghunathan, A., J. Steinhardt, and P. S. Liang (2018). Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems 31*.

Randløv, J. and P. Alstrøm (1998). Learning to drive a bicycle using reinforcement learning and shaping. In *ICML*, Volume 98, pp. 463–471. Citeseer.

Ren, K., T. Zheng, Z. Qin, and X. Liu (2020). Adversarial attacks and defenses in deep learning. *Engineering 6*(3), 346–360.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017). Proximal policy optimization algorithms. *arXiv:1707.06347*.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement learning: An introduction*. MIT press.