Investigating Adversarial Policy Learning for Robust Agents in Automated Driving Highway Simulations

Alessandro Pighetti[1[0009-0001-7166-5750]], Francesco Bellotti[1], Changjae Oh[2[0000-0000-6522-2451]], Luca Lazzaroni[1[0000-0001-8092-5473]], Luca Forneris[1[0009-0008-9595-3520]], Matteo Fresta[1[0009-0000-7265-7501]], and Riccardo Berta[1[0000-0003-1937-3969]]

[1] Department of Electrical, Electronic and Telecommunication Engineering (DITEN), University of Genoa, Via Opera Pia 11a, 16145 Genoa, Italy
{alessandro.pighetti, luca.lazzaroni, luca.forneris, matteo.fresta}@edu.unige.it, {francesco.bellotti,riccardo.berta}@unige.it

[2] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, England
c.oh@qmul.ac.uk

Abstract. This research explores an emerging approach, the adversarial policy learning paradigm, that aims to increase safety and robustness in deep reinforcement learning models for automated driving. We propose an iterative procedure to train an adversarial agent acting in a highway-simulated environment to attack a victim agent that is to be improved. Each training iteration consists of two phases. The adversarial agent is first trained to disrupt the victim-agent policy. The victim model is then trained to overcome the defects observed by the attack from the adversarial agent. The experimental results demonstrate that the victim agent trained with adversarial attacks outperforms the original agent.

Keywords: automated driving, adversarial policies, autonomous agents, proximal policy optimization, reinforcement learning, highway driving, decision making, highway-env simulator.

1 Introduction

Deep reinforcement learning (DRL) has emerged as a powerful approach to train online agent policies, enabling them to learn complex decision-making behaviors in dynamic environments, which is being used also in automotive applications (e.g., [1–3]). However, ensuring the safety and robustness of the learned policy remains a critical challenge. Different approaches are being studied to address safety and robustness in deep learning models. The adversarial paradigm [4] represents one of such emerging approaches in various research fields, such as computer vision and reinforcement

learning. It is defined as the procedure of purposefully attacking the model under development, also known as the victim, through different methods depending on its purpose and area of deployment. In particular, DRL policies are vulnerable to adversarial attacks involving perturbations to the models state, action or transition dynamics [4] or by applying destabilizing forces on the trained system [5].

The existing work in [6] shows that, in competitive two-player games, it is possible to build an adversarial policy for an agent physically present in the target environment to get the victim model in a failure state. Following this approach, we propose a new method to increase robustness of an a highway-driving DRL agent by implementing the adversarial training paradigm in the well-established highway-env [7] DRL environment. The method relies on adapting the iterative training procedure of the victim and adversarial agents, which is at the core of the adversarial learning (e.g., [8]).

The remainder of the paper is organized as follows. Section 2 describes the environment for the proposed training methodology. Section 3 highlights the core ideas behind our two-phase adversarial training process. Section 4 presents the experimental settings, while Section 5 discusses the results and Section 6 draws the conclusions.

2 Environment

We present a modified version of the highway-env [7] open-source simulator to develop a DRL training framework, given the simplicity and ease of customization which allows for quick prototyping and testing of different mechanics and details as well as a direct comparison to previous work [9, 10].

Fig. 1 depicts the environment we used for developing our method. The scene contains a customizable number of heuristic Non-Player Vehicles (NPVs) navigating the scene alongside two vehicles playing as the part of the victim (VV) and the adversarial vehicles (AV), that are guided by two different DRL policies. Two sets of actions are provided for the proposed experiment: the lower-level action space already present in the original framework (faster, slower, lane change right, lane change left) and a high-level action space (keep lane, go to right-most lane, overtake) illustrated in [10]. The observations for both agents are named “kinematic”, as per the original highway-env’s definition and consist of presence (whether a vehicle is present or not in the scene), longitudinal and lateral position and velocity. These observations are taken for the ego vehicle and the N = 7 (for our experiments) nearest vehicles.

Fig. 1. Snapshot of the highway environment. The victim vehicle is colored in light-green, the adversary in purple, while the NPVs in light-blue.

3 Training Cycle Definition

For training the victim and adversarial agents, respectively driving VV and AV, we defined a two-phase cycle, which is depicted in Fig. 2. We begin the process by freezing a pre-trained victim model (particularly, the one illustrated in [10]), and let it act in pure exploitation as a part of the highway environment. The goal of VV is to drive safely (i.e., no collisions with other vehicles) for as many kilometers as possible. In this first phase, the learner is AV acting upon the scene alongside with VV and the NPVs. The goal of AV is to hinder VV. To this end, the AV observation includes also the VV last observation and action. Unlike VV, which uses the aforementioned high-level action space designed to reach better "normal" driving performance, AV uses the original lower-level action space, which allows a greater degree of freedom in driving, which is needed to perform abrupt maneuvers to challenge VV. The idea behind the first phase of training is to build a strong adversarial policy and put the pre-trained vehicle in a danger/failure state.

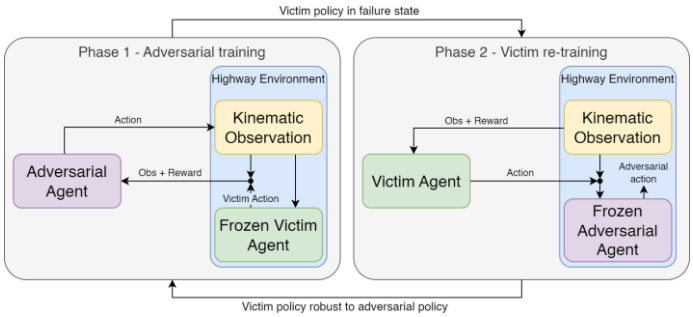

Fig. 2. Outline of the proposed two-phase adversarial training cycle.

As the main effect of the first training phase, we expect VV to collide significantly more frequently than in the normal situation (i.e., without trained AV) and maintain a lower mean cruising speed. We conclude the first training phase when the metrics indicate a clear degradation of VV performance and proceed to the second phase. Here, AV is frozen and VV becomes the subject of the training. Thus, we resume the victim's model training through a continued learning procedure without any changes to the original reward function or state and fine-tune it in the adversarial scenario. The goal of this second phase is to make VV policy robust to the newly introduced adversarial attack, thus enhancing VV capability to deal with unexpected dangerous behaviors by other vehicles, that would previously induce a failure state of the agent. This two-phase process is then repeated a certain number of times, until the predefined stopping criteria is met (e.g., in terms of VV policy robustness) or results suggest using different hyperparameter values.

4 Experiment

For the implementation, we opted for the Gymnasium Python framework (formerly known as OpenAI Gym [11]), which provides an intuitive API for DRL model training and environment configuration. The DRL model deployed for both victim and adversarial agents is the Proximal Policy Optimization (PPO) algorithm [12]. By repeatedly updating the policy parameters through a clipped surrogate objective function, PPO balances the exploration and exploitation learning stages, assuring stability and dependable performance. The neural network at the core of the algorithm is a 2-layer multilayer perceptron, with 64 neurons per layer for both analyzed models. The environment policy frequency (i.e., decision rate) is set to 1 Hz., to give the agents sufficient time to observe the effect of their previous action. We set the maximum training episode duration to 60 seconds. In both training phases an episode is terminated whenever a collision occurs. Also, when training AV, the episode is terminated if AV is overtaken by VV, since AV did not succeed in making VV fail. The number of NPVs is set to a randomly chosen value from 10 to 15 for each episode in both training phases.

The reward function (RF) is the key to the success of any DRL agent training, since RF numerically shapes the model behavior. For the first training phase (in which the VV policy is frozen), RF aims to reward the AV's ability to make VV fail. Thus, RF of AV is simply the opposite of RF with which VV was trained. That RF included rewards for high speed and right-most lane, and huge penalty for collision, as detailed in [10]. It is important to stress that the considered quantities for the AV training (i.e., speed, lane collision) are referred to the VV. However, the RF of AV additionally includes a penalty for its own collision with NPVs (a collision would prevent AV from continuing its disturbance task), while a rear-front collision with VV is not penalized (we do not reward it to prevent AV from learning to really chase for VV, which would be an unrealistic behavior). Finally, if VV manages to overtake AV, the episode is terminated, with a training penalty. In the second phase, the AV policy is frozen, while VV is fine-tuned with its "normal" RF, which is described in [10]. It is important to highlight that, in each phase, only one agent is trained, so to prevent non-stationary conditions that would hinder the training.

5 Results

The initial victim policy was trained for a total of 600k steps before being frozen for the first cycle of adversarial training. Phases 1 and 2 of the process took 300k steps each, for a total of 600k steps for the execution of the whole first cycle. For the preliminary experiment presented in this paper, we considered a single cycle iteration.

We evaluate the victim's policy before and after adversarial re-training. Particularly, the increase in robustness of the VV policy is evaluated by comparing the episode metrics reported in Table 1 averaging 1000 test episodes. Tests were conducted in the highway 3-lane environment in two conditions: with and without AV in the scene, while NPVs are always present. In the first phase, the emergent behavior adopted by our best AV model can be described as a continuous, yet not random, swerving procedure. As

soon as AV detects the presence of VV behind, it visibly tries to move up or down the lanes to block VV. If possible, AV tends to steer at the very last possible moment to cut right in front of VV and thus have a chance of causing a "good" collision (rear-front). As shown in Table 1, this adversarial behavior leads VV to increase the collision rate by one order of magnitude (14.7% vs. 1.4% of the episodes) and lower its speed significantly when compared to the case without the AV. The VV model also shows a reduction in average deceleration and an increase in average acceleration. We argue that this is due to the fact that VV drives more slowly, due to the behavior of AV (less average deceleration), then it tries to abruptly accelerate to quickly perform the overtaking.

The second phase aims to improve the pre-trained policy of VV, which is the actual goal of the overall process, through a continued learning procedure. VV has now to learn to deal with AV, which hinders the normal behavior of VV through selective lane and speed changes. Comparing the third and first column of Table 1, we see that, the VV policy at the end of this phase is still negatively influenced by the AV presence, mainly in terms of collision number and average kilometers travelled, compared to the original VV without AV. The average left-lane change number is also increased, suggesting that the VV has learned to overtake vehicles more often in the adversarial scenario. On the other hand, comparing the second and third column, we see with no surprise that, in the adversarial scenario, the re-trained agent outperforms the original one in all the metrics.

Table 1. Performance over 1000 episodes of the victim models

Considered metrics	Baseline	Baseline vs. AV	Re-trained vs. AV	Re-trained
Mean episode duration (s)	59	53	57	**60**
Number of collisions	14	147	65	**2**
Average travelled kilometers	1.84	1.5	1.82	**1.95**
Average speed (km/h)	111	89	**118**	**118**
Average deceleration (m/s^2)	-5.38	-4.52	-5.47	**-5.57**
Average acceleration (m/s^2)	1.62	1.84	**1.28**	1.34
Average left-lane changes	0.32	0.38	0.9	**0.96**
Average right-lane changes	**1.33**	1	1.09	1.27

Finally, testing the agent in a normal, non-adversarial scenario (fourth and first column), we see that the re-trained model collides less frequently, while traveling more kilometers at a slightly higher speed and accelerates less abruptly. This indicates a significant agent improvement in terms of safety and overall robustness.

6 Conclusions and future work

We explored the implementation of an adversarial learning procedure to increase robustness of a DRL agent for automated driving in a 3-lane highway-env simulated environment. Experimental results show that the re-trained model's policy has proven to

be more robust and safer, outperforming the original one in all the metrics. These initial results suggest that research should be conducted on the reward function, improving the realism of the AV disturbances, and loss function customization for the employed learning algorithm to stabilize the training phase. Moreover, the method may be transferred to more complex driving scenarios and simulators.

References

1. Folkers, A., Rick, M., Buskens, C.: Controlling an Autonomous Vehicle with Deep Reinforcement Learning. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 2025–2031. IEEE, Paris, France (2019)
2. Bellotti, F., Lazzaroni, L., Capello, A., Cossu, M., De Gloria, A., Berta, R.: Explaining a Deep Reinforcement Learning (DRL)-Based Automated Driving Agent in Highway Simulations. IEEE Access. 11, 28522–28550 (2023). https://doi.org/10.1109/ACCESS.2023.3259544
3. Lazzaroni, L., Bellotti, F., Capello, A., Cossu, M., De Gloria, A., Berta, R.: Deep Reinforcement Learning for Automated Car Parking. In: Berta, R. and De Gloria, A. (eds.) Applications in Electronics Pervading Industry, Environment and Society. pp. 125–130. Springer Nature Switzerland, Cham (2023)
4. Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., Hsieh, C.-J.: Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations, http://arxiv.org/abs/2003.08938, (2021)
5. Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.: Robust Adversarial Reinforcement Learning, http://arxiv.org/abs/1703.02702, (2017)
6. Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., Russell, S.: Adversarial Policies: Attacking Deep Reinforcement Learning, http://arxiv.org/abs/1905.10615, (2021)
7. Leurent, E.: An Environment for Autonomous Driving Decision-Making, https://github.com/eleurent/highway-env, (2018)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM. 63, 139–144 (2020). https://doi.org/10.1145/3422622
9. Campodonico, G., Bellotti, F., Berta, R., Capello, A., Cossu, M., De Gloria, A., Lazzaroni, L., Taccioli, T., Davio, F.: Adapting Autonomous Agents for Automotive Driving Games. In: De Rosa, F., Marfisi Schottman, I., Baalsrud Hauge, J., Bellotti, F., Dondio, P., and Romero, M. (eds.) Games and Learning Alliance. pp. 101–110. Springer International Publishing, Cham (2021)
10. Pighetti, A., Forneris, L., Lazzaroni, L., Bellotti, F., Capello, A., Cossu, M., De Gloria, A., Berta, R.: High-Level Decision-Making Non-player Vehicles. In: Kiili, K., Antti, K., de Rosa, F., Dindar, M., Kickmeier-Rust, M., and Bellotti, F. (eds.) Games and Learning Alliance. pp. 223–233. Springer International Publishing, Cham (2022)
11. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym, http://arxiv.org/abs/1606.01540, (2016)
12. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms, http://arxiv.org/abs/1707.06347, (2017)