Enhanced Adversarial Strategically-Timed Attacks on Deep Reinforcement Learning

Chao-Han Huck Yang
Georgia Tech

Jun Qi
Georgia Tech

Pin-Yu Chen
IBM Research

I-Te Danny Hung
Columbia University

Yi Ouyang
Preferred Network Research USA

Xiaoli Ma
Georgia Tech

Abstract

Recent deep neural networks based techniques, especially those equipped with the ability of self-adaptation in the system level such as deep reinforcement learning (DRL), are shown to possess many advantages for optimizing robot learning systems (e.g., autonomous navigation and continuous robot arm control.) However, the learning-based systems and their associated models may be threatened by the risk of intentionally adaptive (e.g., noisy sensor confusion) and adversarial perturbations from the real-world. In this paper, we introduce timing-based adversarial strategies against a DRL-based navigation system by jamming in physical noise patterns on the selected time frames. To study the vulnerability of learning-based navigation systems, we propose two adversarial agent models: one refers to online learning; another one is based on evolutionary learning. Three open-source robot learning and navigation control environments are employed to study the vulnerability under adversarial timing attacks. Our experimental results show that the adversarial timing attacks can lead to a significant performance drop, and also suggest the necessity of enhancing the robustness of robot learning systems.

1 Introduction

Deep Reinforcement Learning (DRL) has gained a widespread applications in digital gaming, robotics and control. In particular, the principal DRL approaches, such as the value-based deep Q-network (DQN) [1], the Asynchronous Advantage Actor-Critic (A3C) [2], and the population-based Go-explore [3], have succeeded in mastering many dynamically unknown action-searching environments [3, 4]. Relying on the similarity between the adaptive and interacting behaviors, the design of DRL-based models are commonly used in the domain of navigation and robotics, and achieve a noticeable improvement than classical methods. However, despite the exciting performance enhancement, DRL-based models may incur some new challenges in terms of system robustness against adversarial attacks. For example, the DRL-based navigation systems are likely to propagate and even enlarge risks (e.g., delay and noisy pulsed-signals on the sensor networks of vehicle [5]) induced from the attackers. Besides, unlike the image classification tasks where only a single mission gets involved, navigation learning agent has to deal with a couple of dynamic states (e.g., inputs from sensors or raw pixels) and the related rewards. Our work mainly focuses on the robustness analysis of strategically-timed attacks with noises potentially conducted from the real world. More specifically, we formulate the adversarial attacks on two DRL-security settings:

- White-box attack: if attacker can access to model parameters, some potential function needs to be used to estimate the learning performance to jam in noise.
- Black-box attack: without the requirements of model parameters, the attacker trains another policy agent with the opposite reward objective via observation of the action made by the victim DRL network, the state, and the reward from the environment.

NeurIPS 2019 Workshop on Robot Learning: Control and Interaction in the Real World, Vancouver, Canada

To validate the adversarial robustness of a navigation system, we attempt a new and important research direction based on a 3D environment of (1) continuous robot arm control (e.g., Unity Reacher); (2) sensor-input navigation (e.g., Unity Banana Collector [6]); (3) a raw image input self-driving environment (e.g., Donkey Car) as shown in the Fig 1 (1), (2), and (3).

2 Related work

Scheduling Physical Attacks on Sensor Fusion. Sensor networks for the navigation system are susceptible to flooding-based attacks like Pulsing Denial-of-Service (PDoS) [7] and adversary selective jamming attacks [8]. The related work includes the security and robustness of background noise, spoofing pulses and jamming signal on the autonomous vehicles. For example, Yan et al. [9] shows that PDoS attacks can feasibly conduct on a Tesla Model S automobile equipped with standard millimeter-wave radars, ultra-sonic sensors, forward-looking cameras. Besides, to detect any anonymous network attacks, a sensing engine defined by some offline algorithms is required within a built-in network system. Furthermore, a recent work [10] also demonstrates that the LiDAR based Apollo-Auto system [11] could be fooled by adversarial noises during the 3D-point-cloud pre-processing phase as a malicious reconstruction.

Adversarial Attacks on Deep Reinforcement Learning. Many works are denoted to adversarial attacks on neural network classifiers in either white-box settings or black-box ones [12, 13, 14]. Goodfellow et al. [15] proposed adversarial examples for evaluating the robustness of machine learning classifiers. Zeroth order optimization (ZOO) [14] was employed to estimate the gradients of black-box systems for generating adversarial examples. Besides, the tasks on RL-based adversarial attacks aim at addressing policy misconduct [16, 17, 18] or generalization issues [19]. In particular, Lin et al. [17] developed a strategically-timed attacking method in which at time t, an agent takes action based on a policy derived from a Potential Energy Function [20]. However, these approaches do not consider the update of online weights associated with the size of the action space. In this work, we further improve the potential estimate model from [17] by weighted-majority online learning and guarantee a bound, $regret_T$, in Eq. (4), and introduce a more realistic black-box timed attack setting.

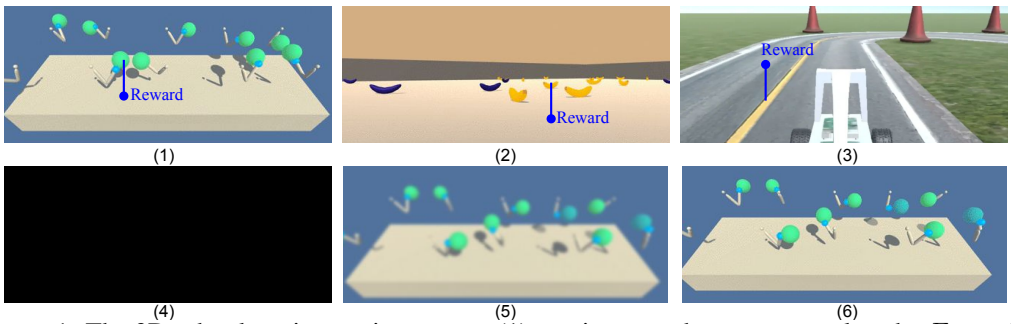

Figure 1: The 3D robot learning environments: (1) continuous robot arm control as the Env_1; (2) banana collector as the Env_2; (3) self-driving donkey car as the Env_3. Noisy observation under timing attack: (4) zero-out; (5) random sensor fusion; (6) adversarial perturbation.

3 Method

3.1 Noisy Observation from the Real World

We define a noisy DRL framework of a robot learning system under perturbation, where a noisy state observation $noisy\_s_t$ can be formulated as the addition of a state s_t and a noise pattern $noise(t)$:

$$noisy\_s_t = s_t + noise(t). \tag{1}$$

We propose three principal types of noise test (from T_1, T_2 to T_3) from the real world to impose adversarial timing attacks:

Pulsed Zero-out Attacks (T_1): Off-the-shelf hardwares [9] can affect the entire sensor networks by an over-shooting noise $noisy\_s_t = 0$ incurred from a timing attack in Eq.(1) as Fig. 1 (4).

Gaussian Average on Sensor Fusion (T_2): Sensor fusion is an essential part of the autonomous system by combining of sensory data from disparate sources with less uncertainty. We define a noisy sensor fusion system by a Gaussian filter for getting $noisy\_s_t$ in Eq.(2) and shown as Fig. 1 (5).

Adversarial Noise Patterns (T_3): Inspired by the fast gradient sign method (FGSM) [12, 16] based

DQN attacks, we use FGSM to generate adversarial pattern against the prediction loss of a well-trained DQN. We use $\epsilon = 0.3$ and a restriction of ℓ_∞-norm, where x is the all input including s_t and r_t; $y = a_t$ is an optimal output action by weighting over possible actions in the Eq.(2):

$$noise(t) = \epsilon \operatorname{sign}\left(\nabla_x J(\theta, x, y)\right). \tag{2}$$

To evaluate the performance of each timing selection algorithm in following sections, each model will receive noise patterns (from T_1, T_2 to T_3) and average the total reward as Table ??. In a perspective of system level, we take the random pulsed-signal as a attacking baseline. We jam in PDoS signals discussed in Sec. 3.1 randomly with maximum constrains $\mathbf{H}$ times (we use $\mathbf{H} = 40$ from [17] as a baseline) to block agent from obtaining the actual state observation in an episode.

3.2 Enhanced White-Box Strategically-Timed Attack by Online Learning

White-box adversarial setting. Recently, since various of pre-defined DRL architecture and models (e.g., Google Dopamine [21]) are released for public use and as an B2B solution, an adversarial attacker may access the open-source and design an efficient strategically-timed attack accordingly.
Weighted-Majority Potential Energy Function. We first propose an advanced type of adversarial attacks originated from online learning, based on the algorithm of weighted majority algorithm (WMA). The procedures of WMA are shown in (3), where we introduce d experts for weighting the revenues incurred by taking d actions. The weights of experts are equally initialized to 1 and then iteratively updated as step (12). At each time t, steps (7) and (8) suggest that we obtain both $a_t^{\max}$ and $a_t^{\min}$ which correspond to the actions of maximum and minimum costs. The decision of attacking the state relies on the threshold value $c(s_t, \mathbf{w}_t, a_t^{\max}, a_t^{\min})$. If $c(s_t, \mathbf{w}_t, a_t^{\max}, a_t^{\min})$ is greater than a pre-specified constant threshold β, we attempt to attack the states by adding pulses to make user have random observations. The choices of $c(s_t, \mathbf{w}_t, a_t^{\max}, a_t^{\min})$ are based on the difference of two potential energy functions (inspired by [17] and [16]) defined as (3)[1]:

$$c(s_t, \mathbf{w}_t, a_t^{\min}, a_t^{\max}) = \frac{\mathbf{w}_t^T \exp(-\mathbf{Q}(s_t, a_t^{\max}))}{\sum_{a_t^{(k)}} \mathbf{w}_t^T \exp(-\mathbf{Q}(s_t, a_t^{(k)}))} - \frac{\mathbf{w}_t^T \exp(-\mathbf{Q}(s_t, a_t^{\min}))}{\sum_{a_t^{(k)}} \mathbf{w}_t^T \exp(-\mathbf{Q}(s_t, a_t^{(k)}))} \tag{3}$$

We use the strategically-timed attacks in [17] as a baseline with $\beta = 0.3$ to evaluate our WMA-enhance algorithms. Then, we further discuss a learning bound for this advanced WMA-policy estimation.
Proposition 1: Assuming that the total number of rounds $T > 2\log(d)$, the weighted algorithm enjoys the bound as Eq.(4), where Z_t denotes a normalization term at time t.

$$regret_T = \sum_{t=1}^{T} \frac{\mathbf{w}_t^T \exp(-\mathbf{Q}(s_t, a_t))}{Z_t} - \min_{i \in [d]} \sum_{t=1}^{T} \frac{\exp(-\mathbf{Q}(s_t, a_t^{(i)}))}{Z_t} \leq \sqrt{2\log(d)T}. \tag{4}$$

Proposition 1 [20] suggests that the weighted revenues are more likely to reach the global optimal in theory, since the regret at time T is upper bounded by a constant value.

3.3 Black-Box Strategically-Timed Attack by Adversarial Evolutionary Strategy

Black-box adversarial setting. Since an adversarial insidious attacking agent is hardly recognizable, an adversarial agent is able to drive the equilibrium of DRL-based system with an opposite objective reward without any information of targeted DRL-model. Thus, we propose an adversarial-strategic agent (ASA) via a population-based training method based on parameter exploring policy gradients [22] (PEPG) for optimizing a black-box system. The PEPG-ASA algorithm can dynamically select sensitive time frames for jamming in an physical noise patterns in Section 3.1, which is likely to minimize the total system-rewards from an off-online observation of the input-output pairs without accessing actual parameters from the given DRL framework as below:

- observation: records of state S from [s_0, s_1,..., s_n] and adversarial reward against victim navigation DRL-agent R_{advs} from [r_0, r_1,..., r_n], an adversarial reward R_{adv} as a black-box security setting.
- adversarial reward R_{adv}: a negative absolute value of the environmental reward R_{env}.

[1]For potential energy estimation on policy-based model (e.g., A3C), we use a weighted-majority average of $c\left(s_t, w_t\right) = \max_{a_t} w_t^T \pi\left(s_t, a_t\right) - \min_{a_t} w_t^T \pi\left(s_t, a_t\right)$.

An obvious way to maximize $E[R_{adv}|s, a_{adv}, \pi_{adv}]$ is to estimate ∇E. Differentiating this form of the expected return with respect to ρ and applying sampling methods, where ρ in Eq. (5) are the parameters determining the distribution over θ, the agent can generate h from $p(h|\theta)$ and yield the following gradient estimator:

$$\nabla_{\boldsymbol{\rho}} E(\boldsymbol{\rho}) \approx \frac{1}{N} \sum_{n=1}^{N} \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho}) r\left(h^n\right). \tag{5}$$

The probabilistic policy, which is commonly used for the policy gradient, is replaced with a probability distribution over the parameter θ for PEPG. The key advantage of the approach is the deterministic actions; thus an entire track of history can be generated from sampling a single parameter.

4 Results

4.1 3D Control and Robot Learning Environment Setup

Our testing platforms were based on the most recently released open-source 'Unity-3D' environments [6] for robotic applications. $\mathbf{Env_1}$ **Reacher:** A double-jointed arm could move to desired position. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm. Every action is a vector with four numbers, corresponding to torque applicable to two joints. Each entry in the action vector should be a numerical value between -1 and 1.
$\mathbf{Env_2}$ **Banana Collector:** A reward of $+1$ is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana from a first-person view vehicle to collect as many yellow bananas as possible while avoiding blue bananas. The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around agent's forward direction. Four discrete actions are available to associate with four moving directions.
$\mathbf{Env_3}$ **Donkey Car:** Donkey Car is an open source embedded system for radio control vehicle with off-line RL simulator. The state input is the image from the front camera with 80×80 pixels, the actions is equal to two steering values ranging from -1 to 1, and the reward is cross track error (CTE). We use a modified reward from [23] divided by 1k to balance track-staying and maximize its speed.

4.2 Performance Evaluation

We applied two classical DRL algorithms, namely DQN and A3C, to evaluate the learning performance relative to well-trained DRL models in.
Baseline (aka no attack): We modify DQN and A3C models from the open-source Dopamine 2.0 [21] package to avoid an overparameterized model with reproducibility guarantee.
Adversarial Robustness (aka under attack): Assuming the presence of one adversarial attacker, we highlight some important results. Overall, although the WMA (white-box setting) outperforms the PEPG-ASA (black-box setting), it also requires much more information of a navigation system during the online potential-energy estimation and training.

5 Conclusion

This work introduces two novel adversarial timing attacking algorithms for evaluating DRL-based model robustness under white-box and black-box adversarial settings. The experiments suggest that the improved performance of DRL-based continuous control and robot learning models can be significantly degraded in the adversarial settings. In particular, both valued and policy-based DRL algorithms are easily manipulated by an black-box adversarial attacking agent. Besides, our work points out the importance of the robustness and adversarial training against adversarial examples in DRL-based navigation systems. Our future work will discuss the visualization and interpretability of robot learning and control systems in order to secure the system. To improve model defense, we could also adapt the adversarial training [12] to train DQN & A3C models by noisy states.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[2] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[3] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

[4] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[5] Tor A Johansen, Andrea Cristofaro, Kim Sørensen, Jakob M Hansen, and Thor I Fossen. On estimation of wind velocity, angle-of-attack and sideslip angle of small uavs using standard sensors. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 510–519. IEEE, 2015.

[6] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

[7] Xiapu Luo and Rocky KC Chang. On a new class of pulsing denial-of-service attacks and the defense. In *NDSS*, 2005.

[8] Alejandro Proano and Loukas Lazos. Selective jamming attacks in wireless networks. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6. IEEE, 2010.

[9] Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle.

[10] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019.

[11] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[13] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

[14] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[15] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097*, 2018.

[16] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

[17] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017.

[18] Yi Han, Benjamin IP Rubinstein, Tamas Abraham, Tansu Alpcan, Olivier De Vel, Sarah Erfani, David Hubczenko, Christopher Leckie, and Paul Montague. Reinforcement learning for autonomous defence in software-defined networking. In *International Conference on Decision and Game Theory for Security*, pages 145–165. Springer, 2018.

[19] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. *arXiv preprint arXiv:1703.02702*, 2017.

[20] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[21] Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G Bellemare. Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*, 2018.

[22] Frank Sehnke, Christian Osendorfer, Thomas Rückstieß, Alex Graves, Jan Peters, and Jürgen Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010.

[23] Bharat Prakash, Mark Horton, Nicholas R Waytowich, William David Hairston, Tim Oates, and Tinoosh Mohsenin. On the use of deep autoencoders for efficient embedded reinforcement learning. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, pages 507–512. ACM, 2019.