Observed Adversaries in Deep Reinforcement Learning

Eugene Lim[1] and Harold Soh[1,2]

***Abstract*— Recent research has shown that deep policies that are trained using reinforcement learning can be impaired by an external agent who acts in accord to environmental limitations. We assume that these external agents can influence some dimensions of the policies' observations. Just as it is important from a security perspective to sanitize inputs in software development, we demonstrate that observations received by the policies can be vulnerable to adversarial attacks in reinforcement learning if not properly sanitized. In this work, we focus on the setting where the influenced dimensions are irrelevant to the task that the policies are optimized for. We demonstrate the importance of input sanitation in preventing attacks, provide a hypothesis on the mechanics behind such attacks, and propose a feature selection method for sanitizing these task-irrelevant dimensions.**

I. Introduction

Decision making in robotics systems have significantly improved in recent years, thanks in part to advancements in artificial intelligence and machine learning. Deep learning in particular has been the dominant methodology for creating data-driven components in these systems [1], [2]. However, studies have shown that deep methods are susceptible to adversarial attacks [3], [4], and this raises concerns about the robustness of such methods.

Deep reinforcement learning (DRL) is a ubiquitous choice among these deep methods [5], [6], [7]. Over the years, DRL has been used to obtain policies for various multi-agent tasks, including those involving human-robot interaction (HRI) [8], [9], [10]. However, earlier studies had demonstrated that the performance of a DRL policies could be negatively impacted by adversarially perturbing the image observations using gradient-based attacks, similar to those that are employed against computer vision models [11].

Recently, Gleave et. al. showed that DRL policies are also susceptible to attacks by external agents that move in accord to environmental constraints [12]. Unlike the gradient-based attacks, these *observed adversaries* are unable to arbitrarily modify the victims' observations, yet are able to significantly degrade the victim's performance. In this work, we expand on [12] and demonstrate the extent of these observed adversary attacks. Just as it is important to sanitize inputs in software development, we demonstrate that observations are also in need for sanitation to protect against such attacks.

In this short paper, we will first detail experiments designed to investigate observed adversary attacks when task-irrelevant features are not properly sanitized. We focus our experiments on Proximal Policy Optimization (PPO) [7], a popular model-free RL method that has been widely used. We then present our results related to the severity of attacks and hypothesize the cause of such attacks. Finally, we discuss the implications on our findings on multi-agent decision making systems and potential future work that is needed to address the robustness of deep RL and advance the development of trustworthy agents.

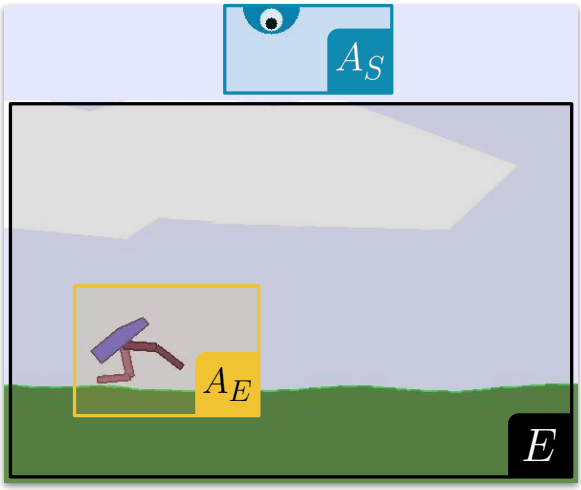

Fig. 1. The BipedalWalker-v3 environment augmented with a slime. Following the notations defined, E is the BipedalWalker-v3 environment, A_E is the agent controlling the walker to maximize returns defined by R, and A_S is the slime.

II. Background & Related Work

As machine learning becomes more widely adopted, there is an increasing demand for assurance that these learning systems are not vulnerable against adversaries. As such, the literature on adversarial attacks on machine learning algorithms has grown rapidly over the past decade, with adversarial attacks against deep computer vision models being one of the most active areas [3], [4]. These attacks on vision models generally involve searching for the smallest perturbation on image pixels that is required to increase the classification loss. Example of such attacks include the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). These methods can also be extended to black-box settings mainly by exploiting the transferability of adversarial examples [13], [14].

Although these gradient-based attacks on vision models are also shown to effective in RL settings [11], we must assume a powerful adversary who is able to directly modify the victim's observations. In contrast, Gleave et. al. worked

[1]Eugene Lim and Harold Soh are with Department of Computer Science, National University of Singapore, Singapore. Email: {elimwj,harold}@comp.nus.edu.sg

[2]H. Soh is also with the Smart Systems Institute, NUS.

under a more realistic thread model where the adversaries are simply external agents acting in a multi-agent environment alongside the victims [12]. They demonstrated that these observed adversaries can act under the constraints of the environment (i.e. the dynamics of the simulated robotics game) to invoke natural but adversarial observations. When these observations are seen by the victims, it can cause the victim to fail catastrophically.

In this short paper, we contribute to this line of work and investigate the susceptibility of deep policies in low-dimensional settings. Specifically, we focus the setting where there exist task-irrelevant dimensions in observations. Intuitively, these irrelevant dimensions that are highly influenced by the external agent (e.g. position of the external agent) but are not relevant to getting a higher returns. Formally, we define task-irrelevancy as:

Definition 1 (Task-Irrelevancy)**.** *The k-th observation feature is task-irrelevant if and only if for all states $s, s' \in \mathbb{R}^n$, if s and s' are equal with dimension k removed:*

$$(s_1, \ldots, s_{d-1}, s_{d+1}, \ldots, s_n) = (s'_1, \ldots, s'_{d-1}, s'_{d+1}, \ldots, s'_n),$$

then $r(s) = r(s')$.

The above definition captures the intuition that ignoring the dimension will not change the reward obtained.

III. Evil Slime Experiment

We design the *evil slime* experiment to test how learned NN policies react to observed adversaries. Specifically, we augment a single-agent environment E with a task-irrelevant agent A_S (represented as a slime) whose behavior has no effect on E. The agent A_E that resides in E aims to maximize its returns with respect to some reward function R. Although A_S does not affect A_E directly, A_E is able to observe the horizontal position of A_S. From an implementation perspective, an additional task-irrelevant feature $x_{n+1} \in \mathbb{R}$ representing the horizontal position of A_S is augmented to the n-dimensional input observation $\mathbf{x} \in \mathbb{R}^n$ of A_E produced by the original environment E.

At each time step, A_S can choose to move left or right. We further prevent A_S from moving arbitrarily far by constraining x_{n+1} to stay within some boundary. Figure 1 shows the slime augmented to the BipedalWalker-v3 environment.

During the training phase for agent A_E, the slime A_S moves left or right randomly with equal probability. Slimes that exhibit such random walk behavior are labelled `arand` and are colored blue in figures. For each environment, we train 20 victim agents A_E across different random seeds. To ensure the experiments are reproducible, we trained all policies using `stable-baselines3` implementation of Proximal Policy Optimization (PPO). We choose to focus on PPO due to its popularity. Both actor and critic network are defined by the default `MlpPolicy` model that uses two 64-node hidden layers. The code for this work is available on `github.com/clear-nus/observersary`.

After agent A_E completes its training, we freeze its policy and substitute A_S with one that always moves left (resp. right). It turns out that this simple behavior is detrimental to A_E and hence, considered "adversarial" behavior. Slimes that exhibit such behavior are labelled `aleft` (resp. `aright`) and are colored red in all figures. Figure 2 summarizes this experiment pipeline. Our primary metric will be the return (accumulated reward) A_E obtained at a test trial where an A_S is paired with a specific agent and both are executed in the environment.

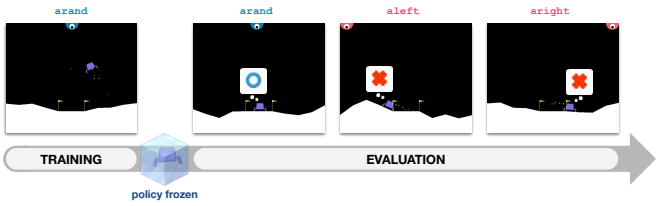

Fig. 2. Evil slime experiment on the LunarLander-v2 environment. We begin by training a policy to complete the task against `arand`. We then freeze the policy and evaluate it against three different opponents: `arand`, `aleft`, and `aright`. The latter two are considered adversarial as they act differently from a typical in-distribution slime.

A. Are agents susceptible to observed adversary attacks in settings with low-dimensional observations?

To answer this question, we evaluated each victim against `arand`, `aleft`, and `aright`, each for 30 runs. To show the results persist across various task settings, we diversified by using environments with terminal states (CartPole-v1), discrete actions (Acrobot-v1), continuous actions (Pendulum-v1), randomized goals (LunarLander-v2), and complex dynamics (BipedalWalker-v3). The mean returns of each agent A_E over all 30 runs are aggregated over the 20 seeds. The statistics are summarized under the "Returns (Without Feature Selection)" columns of Table I. Besides Acrobot-v1, all environments see a drastic decrease in returns when evaluated against the adversarial slimes `aleft` and `aright`, with some environment (such as LunarLander-v2) suffering almost 65% decrease in mean returns.

B. What causes this adversarial behavior?

In the *evil slime* experiment, we expect the victim's optimal policy to ignore the slime's position. To exhibit this ignoring behavior, it is sufficient for the victim to zero the weights of its neural network policy that are associated to the slime's position. However, this is not what we observed in the trained policies as shown in Figure 3.

We hypothesize that this inability to zero off weights is a fundamental property of neural network. To test this, we train a classifier on the Iris data set [15] where each input $\mathbf{x} \in \mathbb{R}^4$ is appended with a randomly generated value $x_5 \sim \mathcal{N}(0,1)$. Figure 4 shows the L_1-norm of the input weights over 1000 training epochs. As hypothesized, the norms associated to x_5 is nowhere near zero.

TABLE I

RETURNS AGAINST VARIOUS OPPONENTS.

Environment	Returns (Without Feature Selection)			Returns (With Feature Selection)		
	arand	aleft	aright	arand	aleft	aright
cartpole-v1	500.00 ± 0.00	355.79 ± 149.07	374.24 ± 152.08	469.61 ± 117.42	428.40 ± 152.54	427.90 ± 151.41
acrobot-v1	-81.37 ± 4.58	-82.78 ± 5.51	-81.81 ± 4.14	-82.78 ± 5.19	-81.30 ± 4.83	-82.92 ± 5.05
pendulum-v1	-184.98 ± 21.71	-268.62 ± 79.73	-338.00 ± 164.31	-193.70 ± 18.71	-193.62 ± 24.52	-190.23 ± 19.83
lunarlander-v2	248.47 ± 28.94	104.80 ± 58.53	87.51 ± 67.30	236.22 ± 46.92	204.06 ± 83.95	201.64 ± 87.23
bipedalwalker-v3	282.54 ± 13.80	132.19 ± 114.08	173.17 ± 105.31	278.96 ± 22.16	143.81 ± 122.17	169.59 ± 108.38

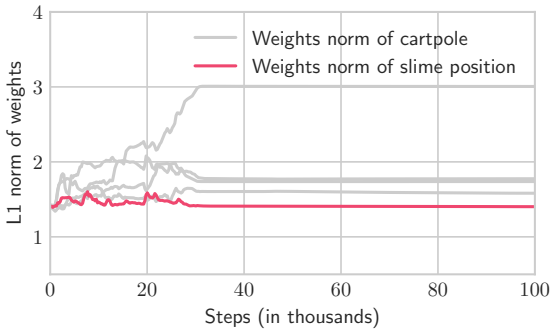

Fig. 3. L_1-norms associated with the input layers weights of the policy network.(Not the correct diagram yet)

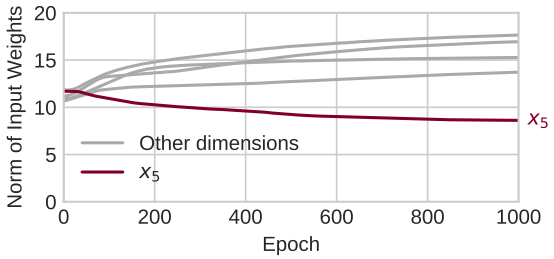

Fig. 4. L_1-norm of the weights associated to the inputs x_i over 1000 training epochs.

C. Could the network learn to ignore task-irrelevant features by cancelling out in later layers?

Although zeroing the weights is a sufficient condition to ignore task-irrelevant features, it might be possible the input to cancel out in its hidden layers. As such, we are interested to know if zeroing the weights is also a necessary condition to ignore task-irrelevant features? We postulate that in the presence of approximation error in floating point values, this is indeed a necessary condition. Our argument relies on the following theorem:

Theorem 1. *Let $(\mathbf{x}, x)$ be some $n+1$-dimensional input vector with $x \in \mathbb{R}$ be a feature we wish to analyze. Further let $\mathbf{W}^{(1)}$ and $\mathbf{w}$ be the weights between the input and the first hidden layer so that $\mathbf{z}^{(1)} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}x \in \mathbb{R}^d$ is the d-dimensional pre-activate values of the first hidden layer. Define $f_\theta(\mathbf{x}, x)$ be the output some fully-connected neural network. Suppose that span$\{\mathrm{d}f/\mathrm{d}\mathbf{z}^{(1)} | \mathbf{x}, x\} = \mathbb{R}^d$. If we want $\mathrm{d}f/\mathrm{d}x = 0$ for all input $(\mathbf{x}, x)$), then $\mathbf{w} = 0$.*

Proof. Using chain rule, we have

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{z}^{(1)}} \cdot \frac{\mathrm{d}\mathbf{z}^{(1)}}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{z}^{(1)}} \cdot \mathbf{w}.$$

Since span$\{\mathrm{d}f/\mathrm{d}\mathbf{z}^{(1)} | \mathbf{x}, x\} = \mathbb{R}^{d_1}$, the only choice of $\mathbf{w}$ that is orthogonal to $\mathbb{R}^d$ is 0. □

From Theorem 1, we see that as long as the assumption span$\{\mathrm{d}f/\mathrm{d}\mathbf{z}^{(1)} | \mathbf{x}, x\} = \mathbb{R}^d$ holds, the task-irrelevant feature x can only be ignored when $\mathbf{w} = 0$. We believe this assumption holds in practice due to the precision limits of floating point values.

To show this empirically, we trained 30 classification models on the Iris data set. For each model, we first sample 2000 input vectors x, and evaluate its $\mathrm{d}f/\mathrm{d}\mathbf{z}^{(1)}$. Then, we augment these vectors into a matrix and compute its rank. All 30 models return full rank, which supports the assumption.

This is further confirmed by Figure 5, which shows how the test performance degrades significantly outside of the training distribution $x_5 \sim \mathcal{N}(0,1)$. After every 50 epochs, we measure the test accuracy of the model after setting the random dimension x_5 to a fixed value between -20 to 20. As we move further away from the $x_5 = 0$, the test accuracy drops drastically.

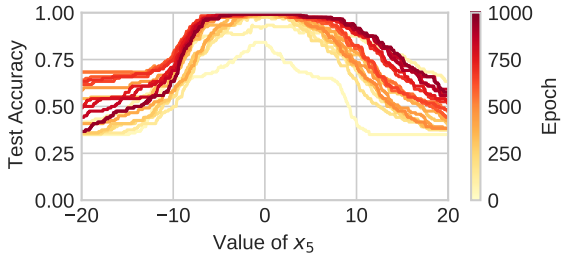

Fig. 5. Test accuracy when x_5 takes on value between -20 to 20. Each line corresponds to a model after training for n epoches, where n is defined by the color bar. We see a clear degradation in performance as we shift further away from the distribution $\mathcal{N}(0,1)$ seen during training.

D. Can we identify the task-irrelevant dimensions?

Motivated the definition of task-relevancy defined earlier on, we propose the following method to identify task-irrelevant dimensions. First, we collect a set of observations O from the environment using an agent and a slime who take

random actions. For each observation s and for each dimension d, we artificially construct a new set of observations $\bar{O}_s = \{(s_1, \ldots, s_{d-1}, s'_d, s_{d+1}, \ldots, s_n) \mid s' \in O\}$ by replacing dimension d of s with $\{s'_d \mid s' \in O\}$. Then, we compute the variance of the value evaluated on all observations in $\bar{O}_s$ using the victim's critic network. Intuitively, the lower the variance, the less task-relevant dimension d is. This variance is computed for all observations in O and the mean of these variances are used to rank the relevancy of the dimension.

Since we cannot trust the critic network for values that are outside the training distribution, our algorithm ensures that the values of each dimension of the constructed observations are seen during the training step.

The column "Returns (With Feature Selection)" in Table I shows the performance of the same 20 agents but with the least irrelevant feature blinded off by setting it to a constant value. For some environments, we see a drastic improvement in returns when the agent is evaluated against the adversaries `aleft` and `aright`. CARTPOLE-V1 returns jumps from 355.79 to 427.90, PENDULUM-V1 from -338.00 to -193.62, and LUNARLANDER-V2 from 87.51 to 201.64. However, this protection against adversaries comes with a price: we see a degradation of performance when the agent is evaluated against the non-adversarial `arand`. For instance, some CARTPOLE-V1 agents are unable to achieve maximum returns after applying feature selection.

IV. DISCUSSION

The results above suggest that PPO-trained agents can be vulnerable to observed adversaries even in relatively simple environments. Nevertheless, these adversaries can be guarded against by proper sanitation of the input observations. Using our proposed method of features selection, we are mostly able to identify the task-irrelevant dimensions and ignore them, thus improving the performance against adversaries that could use these dimensions as potential attack vectors. However, there are still cases where our method fails to identify the task-irrelevant dimensions, such as the case for `BipedalWalker-v3`, where most input dimensions are task-relevant only in very specific joint configurations.

To protect against observed adversaries effectively, we must first identify the core mechanisms behind the attack. One major limitation of our work is that we assumed the task-irrelevant dimensions are nicely decoupled from the other dimensions in the observation. This assumption does not hold in the environment used by Gleave et. al. [12] as it is indeed important to know the position of the external agent when trying to avoid them.

In some HRI settings which involves social or collaborative robots that are expected to perform in environments with other human agents, the threat of observed adversaries become particularly problematic. As seen in this work, it is highly possible that an adversarial human need not physically interact with the robot to cause significant performance degradation. These attacks can drastically undermine trust and limit adoption of such robots.

V. ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/ 2021-08-011).

REFERENCES

[1] A. Punjani and P. Abbeel, "Deep learning helicopter dynamics models," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3223–3230.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016. [Online]. Available: http://jmlr.org/papers/v17/15-522.html

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013. [Online]. Available: https://arxiv.org/abs/1312.6199

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. [Online]. Available: https://arxiv.org/abs/1412.6572

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[6] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," 2015. [Online]. Available: https://arxiv.org/abs/1502.05477

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[8] H. Modares, I. Ranatunga, F. L. Lewis, and D. O. Popa, "Optimized assistive human–robot interaction using reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 655–667, 2016.

[9] M. Khamassi, G. Velentzas, T. Tsitsimis, and C. Tzafestas, "Robot fast adaptation to changes in human engagement during simulated dynamic social interaction with active exploration in parameterized reinforcement learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 881–893, 2018.

[10] B. Xie and C. H. Park, "A multimodal social robot toward personalized emotion interaction," *CoRR*, vol. abs/2110.05186, 2021. [Online]. Available: https://arxiv.org/abs/2110.05186

[11] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," 2017. [Online]. Available: https://arxiv.org/abs/1702.02284

[12] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJgEMpVFwB

[13] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," 2016. [Online]. Available: https://arxiv.org/abs/1602.02697

[14] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy*, 03 2016, pp. 372–387.

[15] R. Fisher, "Iris," UCI Machine Learning Repository, 1988.