White-Box Adversarial Policies in Deep Reinforcement Learning

Stephen Casper[1,2], Dylan Hadfield-Menell[1] and Gabriel Kreiman[2,3]

[1]MIT CSAIL
[2]Boston Children's Hospital
[3]Center for Brains, Minds, and Machines

Abstract

Adversarial examples can be useful for developing safer AI both by identifying vulnerabilities in a model and improving its robustness via adversarial training. In reinforcement learning, adversarial policies can be developed by training an adversarial agent to minimize a target agent's rewards. Prior work has studied black-box attacks where the adversary only sees the state observations and effectively treats the target agent as any other part of the environment. In this work, we study white-box adversarial policies to understand whether an agent's internal state can offer useful information for other agents. We make three contributions. First, we introduce white-box adversarial policies in which an attacker can observe a target agent's internal state at each timestep. Second, we demonstrate that white-box adversarial policies are more effective at finding weaknesses in a target agent, resulting in both faster initial learning and higher asymptotic performance. Third, we show that training against white-box adversarial policies can be used to make learners in single-agent environments more robust to domain shifts. Code is available at this https url.

Keywords

Adversarial attacks, Adversarial training, Robustness, Reinforcement learning

1. Introduction

The ability to discover and correct flaws with models is key for safer AI. One approach to this can be via constructing and training against *adversarial* attacks that are specifically crafted to make a system fail. Adversarial attacks in the form of subtle perturbations to inputs have been widely studied in supervised learning [1, 2]. However, compared to supervised learning, reinforcement learning (RL) agents can face an expanded set of threats [3, 4], including adversarial *policies* from other agents. Adversarial policies have been used both to attack target agents [5, 6] and to improve their robustness through adversarial training [7]. However, the standard approach for developing them has been to simply train an attacker against a black-box target until the attacker (over)fits a policy that minimizes the target's reward. This black-box approach sometimes works well, but it fails to utilize any information beyond what the attacker can directly observe, thus treating the target as any other part of the environment. This approach also typically requires cheap query access to the target, often for many millions of timesteps. Thus, we set out to expand on the conventional threat model with adversarial policies that exploit richer information from the target, known as white-box attacks, in order to encourage more robust performance.

The analog to training a black-box adversarial policy in supervised learning would be to make a zero-order search through a model's input space to find examples that make it fail. While black-box attacks like these have

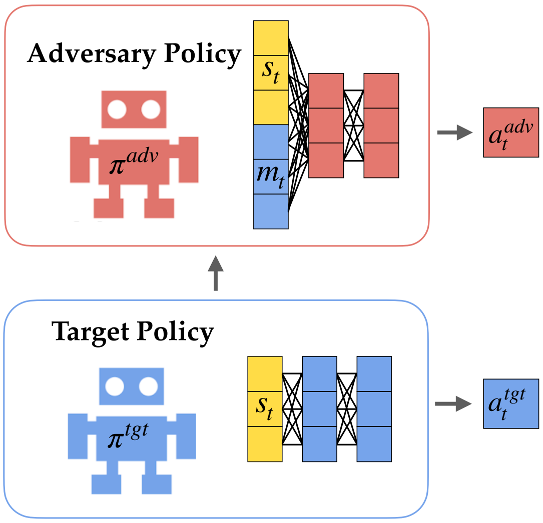


Figure 1: White-box adversarial policies. At each timestep, both the adversary (adv) and target (tgt) observe the state s_t. The adversary also observes information from the internal state of the target and concatenates this extra information, m_t, into its observations. We demonstrate how this type of white-box adversarial policy is more useful than black-box ones for identifying vulnerabilities using attacks and improving robustness using adversarial training.

SafeAI 2023: AAAI 2023 workshop, February 13–14, 2023, Washington

D.C.
scasper@@mit.edu (S. Casper)
https://stephencasper.com (S. Casper)

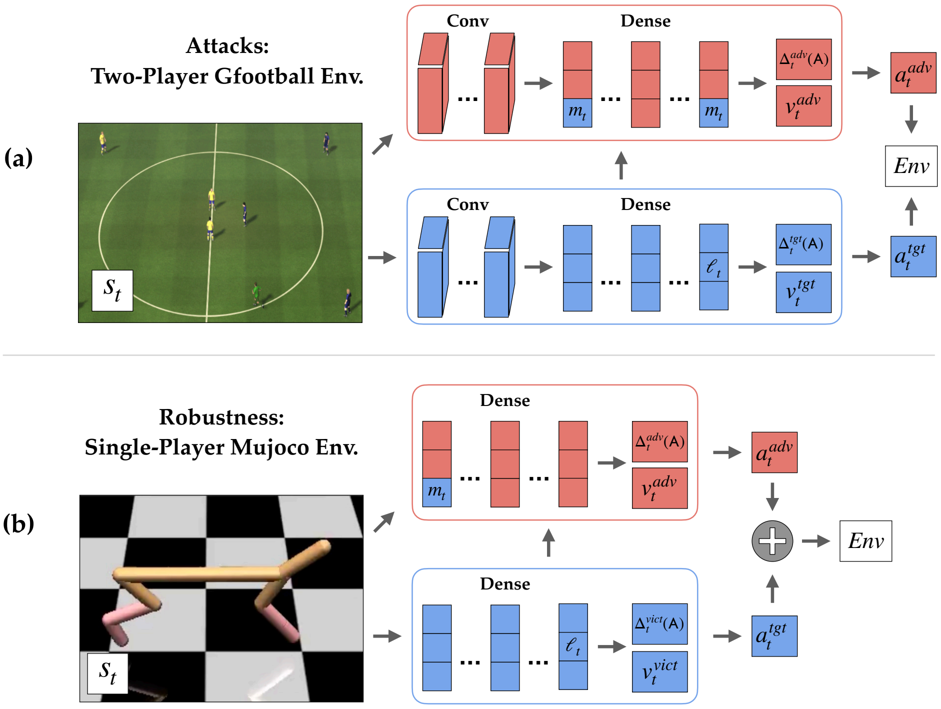


Figure 2: Our setup for (a) adversarial attacks in the two-player Google Research Football (Gfootball) environment and (b) robust adversarial reinforcement learning (RARL) in single-player Mujoco environments. At each timestep, the state observation s_t is passed to the adversary and target. The adversary is also given internal information m_t from the target which is concatenated into its observations or internal activations. The vector m_t can include the target agent's action distribution $\Delta_t^{tgt}(\mathcal{A})$, value estimate v_t^{tgt}, and/or latent activations ℓ_t. For the two-player Gfootball environment, both actions are passed into the environment's step function. For single-player Mujoco environments the adversary's action is added to the target's as a perturbation.

been studied in supervised learning [8], they are much less effective and query-efficient than white-box ones which permit access to the model's internal state. Thus, here we study how using information from the target can help an attacker learn an adversarial policy more quickly and effectively. Our version of white-box attacks are adversarial policies that can "read the target's mind." Fig. 1 depicts our general approach. At each timestep, both the adversary and target observe the state s_t. The adversary, however, is also able to observe internal information, m_t, from the target agent. In our experiments, m_t is a vector that consists of the target's action distribution $\Delta_t^{tgt}(\mathcal{A})$, value estimate v_t^{tgt}, and/or latent activations ℓ_t.

Specifically, we test this approach in two different settings. First, we test adversarial *attacks* using the two-player Google Research Football (Gfootball) environment [9] and large convolutional policy networks. Both the adversary's and target's actions are passed into the environment's step function. This setup is illustrated in Fig. 2a. Here, we show that white-box attackers are better for identifying weaknesses in the target agent, achieving both higher initial and asymptotic performance than black-box baselines. Second, we adopt the robust adversarial reinforcement learning (RARL) approach from [7, 10] for experiments in single-player Mujoco environments (HalfCheetah and Hopper) [11] with small fully-connected policy networks. The adversary acts by perturbing the target agent's actions. This is shown in Fig. 2b. Here, we find that white-box adversaries can be more useful for training robust agents whose policies are not only more robust to the adversary but also generalize better to environments with altered transition dynamics.

Given these results, we argue that adversarial policies that exploit inner information from the target agent pose greater opportunities for identifying and correcting weaknesses in reinforcement learners. More generally, our results demonstrate that observations from an agent's internal state can be useful for other agents that interact with it. Following a discussion of related works in section 2, Section 3 details our threat model and methods. Section 4 presents results, and Section 5 a discussion. For a high-level explanation and summary, see the Appendix. Code is available at https://github.com/thestephencasper/white_box_rarl.

2. Related Work

Adversarial Policies: Reinforcement learning agents can be vulnerable to several types of adversarial threats including input perturbations, action perturbations, reward perturbations, environments, and policies from other agents. Both [3] and [4] offer surveys of threats and defenses. Our focus is on adversarial policies. Conventionally, these attacks have been developed by simply training the adversary against the fixed target agent's policy. This approach has been used by [12, 5, 6, 13, 14, 15] for attacks. These adversaries were even observed unintentionally by [16] and [9] who found that in competitive multiagent environments, it was key to rotate players in a round-robin fashion to avoid agents overfitting against a particular opponent. Additionally, [17] introduced a approach based on planning, [5] tested the detectability of adversarial policies, [5, 18] explored defense techniques via obfuscating the attacker and using option-based policies respectively, [14, 19] experimented with defense via adversarial training, and [6, 20] offered methods of attacking a target whose reward is unknown.

Meanwhile, [7, 21, 22, 10, 23, 24] have studied Robust Adversarial Reinforcement Learning (RARL) in which an agent is trained alongside an adversarial policy that perturb's its state or actions in order for the agent to learn more robust control. [25] studied the stability of this approach. Others [26, 27, 28] have adversarially trained agents under observation or environment perturbations. To the best of our knowledge, however, no works to date have studied white-box attacks or RARL in modern reinforcement learning contexts.

Black vs. White-box Attacks: In supervised learning, adversarial attacks are simple to make using white-box access to the target's internal weights. Black-box attacks, however, typically require transfer, zero-order optimization, or gradient estimation, and they are usually less successful [8]. Several others including [26, 29, 30, 31, 27] have studied attacks against reinforcement learners based on perturbing the target agent's observations. [32] further demonstrated the use of a target's internal state by using the value function for scheduling maximally-effective adversarial observation perturbations. These types of attacks require an attacker to have the ability to manipulate agent observations and involve propagating the gradient for an adversarial objective through the policy network. In contrast, our white-box adversarial policies only differ from black-box ones from related work in whether the attacker, a reinforcement learner, can observe the target's internal state. Several works [33, 34, 35, 36, 37] have also trained agents with a theory of mind for their opponent in competitive tasks, but only in very simple tabular or cartpole environments. To our knowledge, we are the first to introduce policies which can exploit internal information from a target in complex environments.

Open-Source Decision Making: We study targets whose policies are transparent to other agents in the environment. Agents with open source policies pose a number of challenges and pitfalls for decision-making. Several works formalize these challenges in the context of decision theory or game theory [38, 39, 40, 41, 42]. Our work adds to this by empirically studying one such challenge: attacks in reinforcement learning.

3. Methods

3.1. Framework

We consider the goal of training an adversary against a target inside of a two player Markov Decision Process (MDP) defined by a 6-tuple: $(\mathcal{S}, \{\mathcal{A}_{adv}, \mathcal{A}_{tgt}\}, T, d_0, \{r_{adv}, r_{tgt}\}, \gamma)$ with $\mathcal{S}$ a state set, $\mathcal{A}_{adv}$ and $\mathcal{A}_{tgt}$ action sets for the adversary and target, $T : \mathcal{S} \times \mathcal{A}_{adv} \times \mathcal{A}_{tgt} \to \Delta(\mathcal{S})$ a state transition function which outputs a distribution $\Delta(\mathcal{S})$ over $\mathcal{S}$, d_0 an initial state distribution, γ a temporal discount factor, and r_{adv} and r_{tgt} reward functions for the adversary and target s.t. $r_{adv}, r_{tgt} : \mathcal{S} \times \mathcal{A}_{adv} \times \mathcal{A}_{tgt} \times \mathcal{S} \to \mathcal{R}$. We assume $r_{adv}(s) \approx -r_{tgt}(s) \;\; \forall s \in \mathcal{S}$. We only run experiments in which the target's policy is fixed, so the two-player MDP reduces to a single-player one. We will use $\pi_{adv} : \mathcal{S} \to \Delta(\mathcal{A}_{adv})$ and $\pi_{tgt} : \mathcal{S} \to \Delta(\mathcal{A}_{tgt})$ to denote the policy of an adversary and target, and $V_{adv}^{\pi_{adv}}, V_{tgt}^{\pi_{tgt}} : S \to \mathbb{R}$ to refer to their value functions.

3.2. Threat Model

There are multiple notions that have been used in supervised and reinforcement learning to characterize an adversary. These include being *effective* at making the target fail, being *subtle* and hard for an observer to detect (e.g., [32]), and being *target-specific* (e.g., [5]). Here, we use the first criterion and consider any policy that is *effective* at making another fail to be adversarial. For further discussion, see Appendix, A.1.

Previous works discussed in Section 2 have assumed a threat model in which the adversary only has black-box access to the target but can cheaply train against it for many timesteps. We both strengthen and weaken this. First, we make the permissive assumption that the adversary can observe the target's internal state at each timestep and is able to use this information as an observation in the same timestep (see Section 3.3 for details). This could be a plausible assumption if a malicious attacker could obtain access to a target agent's policy parameters – especially if its designers make the target open-source. However, a more realistic case for safety-critical settings in which an attacker may have white-box access to a target agent is if the agents *developers* use white box access to it to find and correct flaws in the agent's policy. Second, we consider the restrictive assumption that the number of timesteps for which the adversary can train against the target may be limited. Realistically, this could be the case if gathering experience is limited or costly for any reason.

3.3. White-Box Adversarial Policies

We train policies using Proximal Policy Optimization (PPO) [43] and Soft Actor Critic (SAC) [44]. Both involve training a value function estimator alongside the policy. We consider attackers that have access to (1) the target agent's action outputs, (2) its value estimate, and/or (3) the internal activations from its policy network. Our goal for (1) is to give the adversary a glimpse of the near future so that it can better counter the target agent's behavior. Our goal for (2) is to make it easier for the attacker to quickly learn its own value function because $V_{tgt}^{\pi_{tgt}}(s_t) \approx -V_{adv}^{\pi_{adv}}(s_t)$. Note this is only possible for targets that have a critic. Finally, our goal for (3) is to give the adversary rich and generally-useful information on how the target represents the state.

At timestep t, the environment state, s_t, is observed. The target processes the state and produces an action $a_t^{tgt} \sim \pi_{tgt}(s_t)$. At the same time, the white-box adversary queries the target to get its action output $\pi_{tgt}(s_t)$, value estimate $V_{tgt}(s_t)$, and/or latent activations $\ell_{tgt}(s_t)$ in the form of a vector $m(s_t)$. In a slight abuse of notation, we refer to $\ell_{tgt}(s_t)$ as ℓ_t and $m(s_t)$ as m_t. Thus, the adversary's policy function can be written as $\pi_{adv}(s_t) = f(s_t, m_t)$, and its value estimate can be written as $V_a^{\pi_a}(s_t) = g(s_t, m_t)$.

We train both adversaries that use large convolutional neural networks (CNNs) and small multilayer perceptrons (MLPs) as policy networks. These architectures are illustrated in Fig. 2. For the large CNNs, we concatenate m_t into the representation of the state twice: once at the first fully-connected layer, and once at the last. We do this so that the adversary can readily learn both complex and simple functions of m_t. In particular,we hypothesized that giving the adversary the target's value estimate in its final layer is helpful for learning its own value estimator, which ought to be approximately the negative of the target's. For the small MLPs policy networks, we only concatenate m_t with the observation once at the beginning for efficiency.

4. Experiments

4.1. Identifying Vulnerabilities

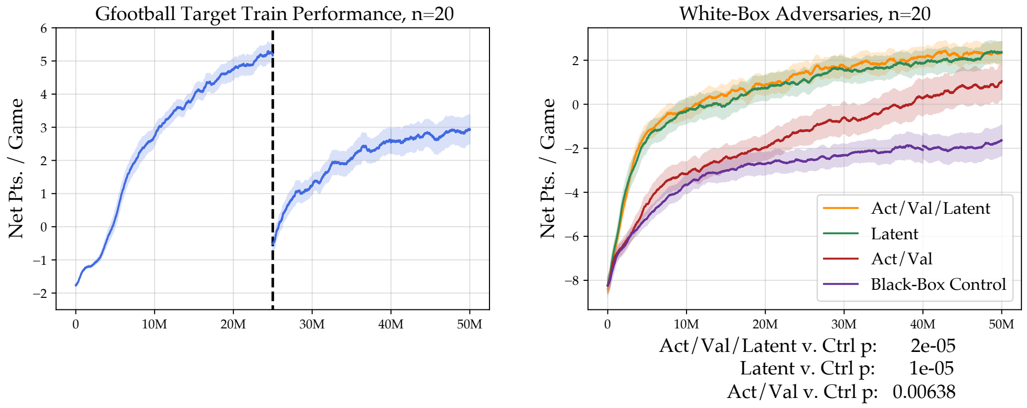
Environment: We use the two-player Google Research Football environment (Gfootball) [9]. Each agent in the environment controls a set of 11 football (soccer) teammates. The states are $72 \times 96 \times 4$ pixels with the four channels encoding the left team positions, right team positions, ball position, and active player position. Observations were stacked over four timesteps to give a perception of time, resulting in observations of $72 \times 96 \times 16$ pixels. The agents' policy networks had a ResNet architecture [45], and the action space was discrete with size 19. We used the same reward shaping as in [9] in which an agent was rewarded 1 for scoring, -1 for being scored on, and 0.1 for advancing the ball one tenth of the way down the field. We trained all Gfootball agents using Proximal Policy Optimization [43] using the Stable Baselines 2 implementation [46].

Target Agents: First, we trained target agents to develop adversarial policies against. For Gfootball, this was done in two stages for a total of 50 million timesteps. First, the targets were trained against a 'bot' agent for 25 million timesteps with an entropy reward to encourage exploration. Second, they were trained for another 25 million timesteps against an agent from the first phase with an entropy penalty to encourage more deterministic play. We found this to result in more consistent behavior from adversaries. In Fig. 3 (a) shows the learning curves for these targets.

Adversaries: We trained four types of adversaries, each of which uses observes different information, m_t, from the target's internal state:

1. **Black-Box Control:** $m_t = \varnothing$. This is the same threat model used by [16], [5] and others mentioned in Section 2.
2. **Action & Value:** $m_t = V_{tgt}(s_t) \oplus \pi_{tgt}(s_t)$ where $\oplus$ is the concatenation operator. Here, the adversary sees the scalar value and an $|\mathcal{A}_{tgt}|$-sized observation giving the target agent's distribution over discrete output actions.
3. **Latent:** $m_t = \ell_t$ where ℓ_t gives the latent activations from some layer during the forward pass through the target's network from s_t. Here, we use those of the final layer from which both the target agent's actions and value are computed.
4. **Full:** $m_t = V_{tgt}(s_t) \oplus \pi_{tgt}(s_t) \oplus \ell_t$. This combines the Action & Value and Latent threat models.

Results: We train each adversary for 50 million timesteps. Fig. 3b shows the training curves for these attackers. All improve significantly over the black box control, both by having faster initial learning and a higher



(a) (b)

Figure 3: Results for white-box adversarial attacks. (a) Training curves for Gfootball target agents. The curves give the mean and standard error of the mean across $n = 20$ target agents. The first 25 million timesteps of training is against a rule-based "bot," and the action entropy is rewarded while the second 25 million timesteps is against a peer and the action entropy is penalized. (b) Learning curves over 50 million timesteps for various adversarial attackers against the target agents from (a) starting from random initialization. The top three curves show the performance of white-box adversaries with access to the target's action distribution and value estimate and/or its latent activations. The bottom shows a black-box control. Notably, the best white-box adversaries do as well after 5 million timesteps as the black box control does after 50 million. As in (a), the curves give the mean and standard error of the mean across $n = 20$ targets. Three p value are shown below giving the results of a one-sided t test for the hypothesis that each white-box agent beat the black-box control.

asymptotic performance. The two types of white-box adversaries that could observe the target's latents performed the best. Both do as well after 5 million timesteps as the black box control does after 50 million. For the action/value, latent, and full attacks, the p values from a one-sided t test for the hypothesis that they were superior to the black box controls were 0.00638, 0.00001, and 0.00002 respectively, demonstrating clear improvements.

4.2. Improving Robustness

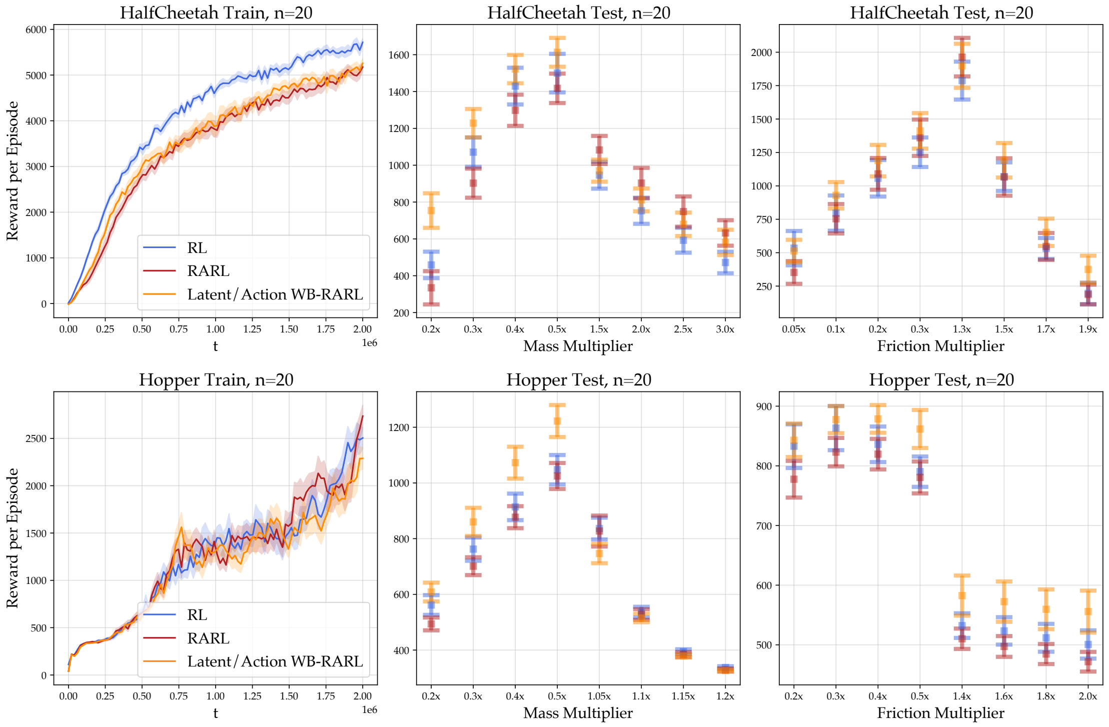
Environment: To evaluate white-box robust adversarial reinforcement learning (RARL), we used HalfCheetah-v3 and Hopper-v3 Mujoco environments from OpenAI Gym. [11]. In both environments, the agent controls a body in a 3D simulated physics environment. Observations are continuous-valued vectors specifying the position of the body, and actions are continuous-valued vectors for controlling it. The agents' policy networks had a small MLP architecture with two hidden layers of 256 neurons each. We trained all gym agents using SAC [44] with the Stable Baselines 3 implementation [47].

Training: In alternation, we trained a target agent and an ensemble of three adversaries who perturbed the target's actions. For each training episode for the target, a random adversary from the three was chosen to make the perturbations. We experiment with three methods:

1. **RL Control:** The target agent is trained with no adversary.
2. **RARL:** The target agent is trained against an ensemble of black-box adversarial agents. This is the approach used by [10].
3. **Latent/Action White-Box RARL (WB-RARL):** The target agent is trained against an ensemble of white-box adversaries that each observe its latent activations from the penultimate layer of the policy network and action outputs. Thus, $m_t = \pi_{tgt}(s_t) \oplus \ell_t$

Results: We trained a total of 40 agents of each type for 2 million timesteps and selected the 20 with the best final performance. Fig. 4a shows the evaluation performance for the HalfCheetah and Hopper agents in an adversary-free environment over the course of training. Performance is comparable between all three conditions with the RL controls seeming to perform the best in HalfCheetah.

To test the robustness of the learned policies, we use the same approach as [7] and [10]. After RARL, we test on a set of adversary-free environments with the transition dynamics altered. We selected a range of 8 mass and 8 friction coefficients to modify the environment dynamics by and tested the agents on all 8 × 8 combinations. The full arrays of results are shown in Fig. 5 in Appendix A.2. And the mean results over all friction coefficients and mass coefficients are plotted in Fig. 4b-c respectively.



(a) (b) (c)

Figure 4: Results for white-box adversarial training. Training and testing performance for (top) HalfCheetah and (bottom) Hopper agents. (a) Performance over training for robust adversarial reinforcement learning (RARL) experiments. Results are obtained from adversary-free testing environments. The curves show the mean and standard error of the mean across $n = 20$ agents. We then tested the final agents across a range of environments with perturbed mass and friction coefficients. The full results are shown in Fig. 5 in Appendix A.2. Here, (b-c) show the mean and standard error of the mean for testing results averaged across the friction and mass coefficients respectively. Again, all errorbars show standard error of the mean across $n = 20$ agents. In general, agents trained with white-box adversarial training perform as well or better than controls.

In Fig. 4b-c, WB-RARL agents generally perform as well or better than the other two. And on average, WB-RARL performs the best over all testing environments. For RL, RARL, and WB-RARL, the HalfCheetah agents achieve mean episode rewards of 902, 914, and 1019, and the Hopper agents achieve 673, 645, and 716 respectively. We performed four one-sided t-tests to test the hypotheses that the WB-RARL agents had superior overall testing performance. For HalfCheetah, the p values were 0.085 and 0.111 for comparing the WB-RARL agents to the RL and RARL ones respectively. For Hopper, the corresponding p values were 0.095 and 0.009. These suggest that the WB-RARL agents are more robust to these domain shifts.

5. Discussion and Broader Impact

Our goal in this work is to better understand opportunities from adversarial policies in reinforcement learning by studying white-box adversarial attackers. We show that allowing an adversarial policy to observe the internal state of the target agent, can result in (1) better initial and asymptotic performance for adversarial attackers and (2) more effective adversarial training for improving the robustness of a learned policy. These results suggest that using white-box adversarial policies to identify and correct flaws with reinforcement learners may be a useful strategy for developing safer, more reliable reinforcement learning systems.

More generally, our results show that information about an agent's internal state offers useful information for other agents interacting with it. This may be the case regardless of whether the setting is adversarial, cooperative, or indifferent. In multiagent settings, it is important to bear in mind that a policy which makes use of white-box information from another agent need not be implemented *by* nor *against* a conventional reinforcement learner. On one hand, policies can be developed without standard reinforcement learning algorithms (e.g., PPO or SAC). For example, human video game players constantly develop strategies to exploit the weaknesses of computer-controlled competitors to great effect. On the other hand, so long as a target agent computes "actions" via latent information, this information could be given to other agents seeking to interact with it. One case in which using adversarial policies against non-reinforcement-learners can be useful is for finding flaws in language models. The inability to differentiate through the sampling of discrete textual tokens makes the task of finding failure modes for language models one that adversarial policies can be useful for (e.g. [48]). Future work on versions of white-box adversarial policies for debugging language models may be useful.

Concerning adversarial attacks in particular, one risk of any work that focuses on attack methods is that they could be used for malicious attacks. This is an important concern, but we emphasize that it is better to develop an understanding of adversarial vulnerabilities through exploratory research than from incidents in the real world. We also stress the benefits of adversarial training and the fact that white box access to an agent can be kept from malicious attackers if appropriate measures are taken. For this reason, we expect white-box adversarial policies to be much more practical for those working to make systems more robust than for malicious attackers.

A limitation is that while we show that white-box attacks can be useful, the improvements from granting the adversary white-box access in the RARL experiments were only modest. And even though white-box attacks can help train adversarial policies more quickly, these attacks may still demand many timesteps. Future work on similar black-box attacks that use a model of the target learned from black-box (and potentially even offline) access may be valuable. Studying ways to more effectively leverage target agent information in fewer training timesteps may also be useful. Additional progress like this toward better understanding opportunities from adversaries in reinforcement learning will be a promising direction for expanding the toolbox for safer and more trustworthy AI.

6. Acknowledgments

We thank Lucas Janson for valuable ideas and feedback throughout the course of this work. We also appreciate discussions with Adam Gleave and Pavel Czempin.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[2] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).

[3] I. Ilahi, M. Usama, J. Qadir, M. U. Janjua, A. Al-Fuqaha, D. T. Huang, D. Niyato, Challenges and countermeasures for adversarial attacks on deep reinforcement learning, IEEE Transactions on Artificial Intelligence (2021).

[4] S. H. Silva, P. Najafirad, Opportunities and challenges in deep learning adversarial robustness: A survey, arXiv preprint arXiv:2007.00753 (2020).

[5] A. Gleave, M. Dennis, N. Kant, C. Wild, S. Levine, S. Russell, Adversarial policies: Attacking deep reinforcement learning, arXiv preprint arXiv:1905.10615 (2019).

[6] T. Fujimoto, T. Doster, A. Attarian, J. Brandenberger, N. Hodas, The effect of antagonistic behavior in reinforcement learning (2021).

[7] L. Pinto, J. Davidson, R. Sukthankar, A. Gupta, Robust adversarial reinforcement learning, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 2817–2826.

[8] S. Bhambri, S. Muku, A. Tulasi, A. B. Buduru, A survey of black-box adversarial attacks on computer vision models, arXiv preprint arXiv:1912.01667 (2019).

[9] K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, et al., Google research football: A novel reinforcement learning environment, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 4501–4510.

[10] E. Vinitsky, Y. Du, K. Parvate, K. Jang, P. Abbeel, A. Bayen, Robust reinforcement learning using adversarial populations, arXiv preprint arXiv:2008.01825 (2020).

[11] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, arXiv preprint arXiv:1606.01540 (2016).

[12] V. Behzadan, W. Hsu, Adversarial exploitation of policy imitation, arXiv preprint arXiv:1906.01121 (2019).

[13] W. Guo, X. Wu, S. Huang, X. Xing, Adversarial policy learning in two-player competitive games, in: International Conference on Machine Learning, PMLR, 2021, pp. 3910–3919.

[14] X. Wu, W. Guo, H. Wei, X. Xing, Adversarial policy training against deep reinforcement learning, in: 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 1883–1900.

[15] J. Guo, Y. Chen, Y. Hao, Z. Yin, Y. Yu, S. Li, Towards comprehensive testing on the robustness of cooperative multi-agent reinforcement learning, in: Pro-

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 115–122.

[16] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, I. Mordatch, Emergent complexity via multi-agent competition, arXiv preprint arXiv:1710.03748 (2017).

[17] A. Pozanco, S. Fernández, D. Borrajo, et al., Anticipatory counterplanning, arXiv preprint arXiv:2203.16171 (2022).

[18] P. Dasgupta, Using options to improve robustness of imitation learning against adversarial attacks, in: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III, volume 11746, International Society for Optics and Photonics, 2021, p. 1174610.

[19] P. Czempin, A. Gleave, Reducing exploitability with population based training, arXiv preprint arXiv:2208.05083 (2022).

[20] T. Fujimoto, T. Doster, A. Attarian, J. Brandenberger, N. Hodas, Reward-free attacks in multi-agent reinforcement learning, arXiv preprint arXiv:2112.00940 (2021).

[21] H. Shioya, Y. Iwasawa, Y. Matsuo, Extending robust adversarial reinforcement learning considering adaptation and diversity (2018).

[22] X. Pan, D. Seita, Y. Gao, J. Canny, Risk averse robust adversarial reinforcement learning, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 8522–8528.

[23] K. L. Tan, Y. Esfandiari, X. Y. Lee, S. Sarkar, et al., Robustifying reinforcement learning agents via action space adversarial training, in: 2020 American control conference (ACC), IEEE, 2020, pp. 3959–3964.

[24] P. Zhai, J. Luo, Z. Dong, L. Zhang, S. Wang, D. Yang, Robust adversarial reinforcement learning with dissipation inequation constraint (2022).

[25] K. Zhang, B. Hu, T. Basar, On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems, Advances in Neural Information Processing Systems 33 (2020) 22056–22068.

[26] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, G. Chowdhary, Robust deep reinforcement learning with adversarial attacks, arXiv preprint arXiv:1712.03632 (2017).

[27] T. Oikarinen, W. Zhang, A. Megretski, L. Daniel, T.-W. Weng, Robust deep reinforcement learning through adversarial loss, Advances in Neural Information Processing Systems 34 (2021).

[28] L. Schott, M. Césaire, H. Hajri, S. Lamprier, Improving robustness of deep reinforcement learning agents: Environment attacks based on critic networks, arXiv preprint arXiv:2104.03154 (2021).

[29] B. Lütjens, M. Everett, J. P. How, Certified adversarial robustness for deep reinforcement learning, in: Conference on Robot Learning, PMLR, 2020, pp. 1328–1337.

[30] E. Korkmaz, Adversarially trained neural policies in the fourier domain, in: ICML 2021 Workshop on Adversarial Machine Learning, 2021.

[31] E. Korkmaz, Investigating vulnerabilities of deep neural policies, in: Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 1661–1670.

[32] J. Kos, D. Song, Delving into adversarial attacks on deep policies, arXiv preprint arXiv:1705.06452 (2017).

[33] A. Davidson, Using artificial neural networks to model opponents in texas hold'em, Unpublished manuscript (1999).

[34] A. J. Lockett, C. L. Chen, R. Miikkulainen, Evolving explicit opponent models in game playing, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007, pp. 2106–2113.

[35] H. He, J. Boyd-Graber, K. Kwok, H. Daumé III, Opponent modeling in deep reinforcement learning, in: International conference on machine learning, PMLR, 2016, pp. 1804–1813.

[36] V. Behzadan, W. Hsu, Rl-based method for benchmarking the adversarial resilience and robustness of deep reinforcement learning policies, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2019, pp. 314–325.

[37] Y. Faghan, N. Piazza, V. Behzadan, A. Fathi, Adversarial attacks on deep algorithmic trading policies, arXiv preprint arXiv:2010.11388 (2020).

[38] J. Y. Halpern, R. Pass, Game theory with translucent players, International Journal of Game Theory 47 (2018) 949–976.

[39] A. Demski, S. Garrabrant, Embedded agency, arXiv preprint arXiv:1902.09469 (2019).

[40] A. Critch, A parametric, resource-bounded generalization of löb's theorem, and a robust cooperation criterion for open-source game theory, The Journal of Symbolic Logic 84 (2019) 1368–1381.

[41] S. Casper, Achilles heels for agi/asi via decision theoretic adversaries, arXiv preprint arXiv:2010.05418 (2020).

[42] A. Critch, M. Dennis, S. Russell, Cooperative and uncooperative institution designs: Surprises and problems in open-source game theory, arXiv preprint arXiv:2208.07006 (2022).

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[44] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, arXiv preprint arXiv:1801.01290 (2018).

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern

recognition, 2016, pp. 770–778.

[46] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, Stable baselines, https://github.com/hill-a/stable-baselines, 2018.

[47] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, N. Dormann, Stable-baselines3: Reliable reinforcement learning implementations, Journal of Machine Learning Research 22 (2021) 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.

[48] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, G. Irving, Red teaming language models with language models, arXiv preprint arXiv:2202.03286 (2022).

[49] N. Papernot, P. McDaniel, I. Goodfellow, Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, arXiv preprint arXiv:1605.07277 (2016).

[50] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).

[51] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, The space of transferable adversarial examples, arXiv preprint arXiv:1704.03453 (2017).

[52] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial examples are not bugs, they are features, arXiv preprint arXiv:1905.02175 (2019).

A. Appendix

A.1. Understanding Adversarial Policies

The notion of an *adversary* for a deep learning system was popularized by [1, 2] and subsequent research. These works developed adversarial images that are both *effective*, meaning that they fool an image classifier, and *subtle*, meaning that they only differ from a benign image by a very small-norm perturbation. While they often transfer to other models [49, 50, 51, 52], these adversaries are also typically *target-specific* in the sense that they are created specifically to fool a particular model.

As in supervised learning, "effectiveness" is used as part of the definition for adversarial policies across the literature. "Target-specificity" sometimes is, but many RL works (e.g., [12]) including ours do not require an adversary to be target-specific. Finally, "subtlety" has not been adopted as a standard for adversaries research in RL. A notion of subtlety for adversaries in RL that would be analogous to adversaries in supervised learning would be that the adversary produces distributions over actions or trajectories that are very similar to a benign agent. However, in this and all related work in RL of which we know, no notion of subtlety is part of the definition of an adversarial policy. So ultimately, we use "adversarial" here to simply refer to a policy which is good at beating a target.

A.2. Full Robust Adversarial Reinforcement Learning Results

As discussed in Section 4.2, we tested agents on environments with altered mass and friction parameters. For both the HalfCheetah and Hopper environments, we used a set of 8×8 different mass and friction values. Testing results across all testing environments for control, RARL, and WB-RARL agents are shown here in Fig. 5. Under each grid, the mean for all results in the grid is displayed. Under the RL and RARL grids (columns 1 and 2), the p value from a one-sided t-test for the hypothesis that WB-RARL is superior to RL and RARL is shown.

A.3. High-Level Summary

Here, we provide a summary of this work which does not assume that the reader has a technical background.

"Reinforcement Learning" (RL) is the process by which an agent learns via some formalized process of trial and error to accomplish a goal. Humans are reinforcement learners. And so are some algorithms that are commonly studied in machine learning research today. For example, is common to use reinforcement learning algorithms to train AI systems to play video games. Using experience, they can infer what types of actions lead to higher scores and adjust their behavior accordingly.

Multiagent RL describes settings in which there is more than one agent acting in some setting. Past research has shown that in multiagent settings, training "adversarial" reinforcement learners to make other reinforcement learners fail can be useful. One one hand, an adversarial agent can often learn to act in a way that renders the "target" agent unable to accomplish its goals. For example, an adversary can sometimes act in ways that make a target in a two player video game seem to take actions that are as bad as – or even worse than – random ones. On the other hand, training a target agent against an adversarial agent can make it more robust to some failures. For example, this might make the target particularly effective at avoiding failures due to changes to its environment.

In this work, we study a new approach to adversarial attacks and adversarial training in RL. We experiment with "white-box" attacks in which the adversary can observe the internal state of the target. For humans, this would be analogous to one person playing a game against someone else while being able to view scans of their brain. We show that these white-box adversarial agents

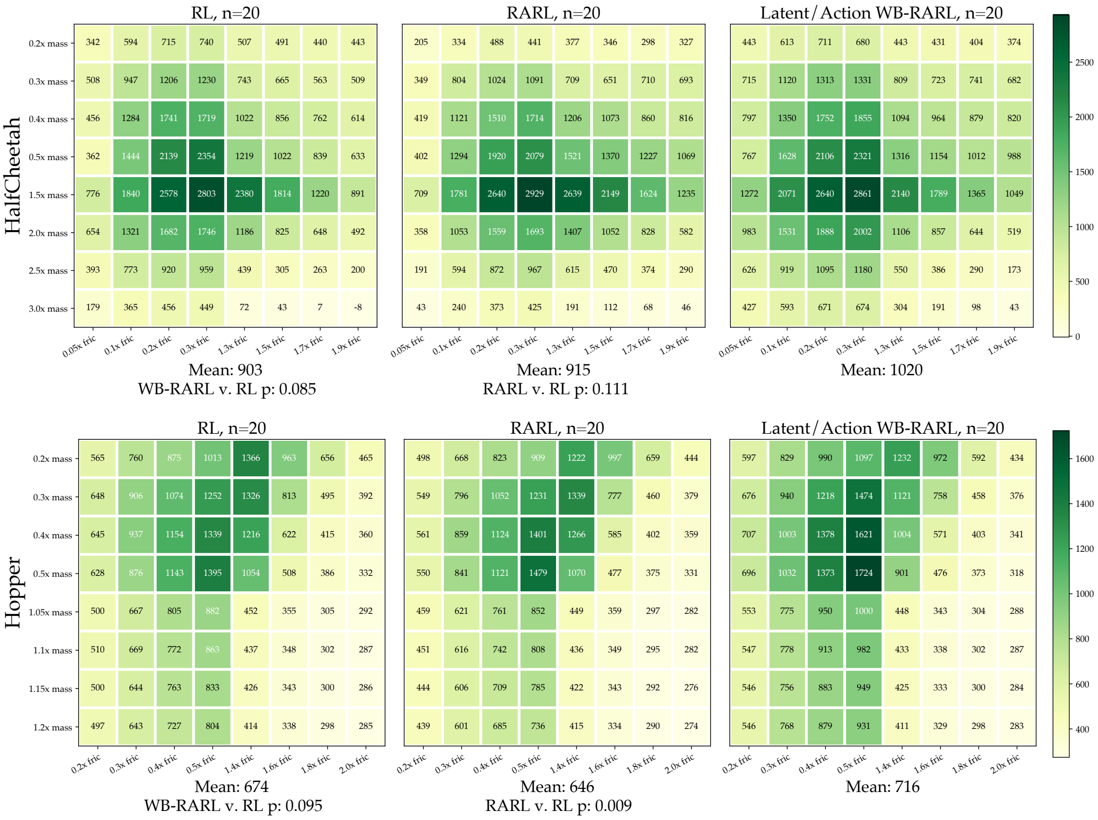


Figure 5: Evaluations for Robust Adversarial Reinforcement Learning Experiments for $n = 20$ agents with (top) HalfCheetah and (bottom) Hopper agents. Each grid shows mean episode reward for adversary-free environments with the mass and friction coefficients altered. Under each grid, the mean for all results in the grid is displayed. Under the RL and RARL grids cols 1 and 2), the one-sided p value for the hypothesis that WB-RARL is superior to RL and RARL is shown.

are more effective than controls for both attacks and adversarial training. We argue that this helps us to better understand opportunities from adversarial RL. And based on these results, we argue that white-box adversaries may be very useful for discovering and correcting flaws in reinforcement learners.