Reducing Exploitability with Population Based Training

Pavel Czempin [1 2] **Adam Gleave** [1]

Abstract

Self-play reinforcement learning has achieved state-of-the-art, and often superhuman, performance in a variety of zero-sum games. Yet prior work has found that policies that are highly capable against regular opponents can fail catastrophically against *adversarial policies*: an opponent trained explicitly against the victim. Prior defenses using adversarial training were able to make the victim robust to a specific adversary, but the victim remained vulnerable to new adversaries. We conjecture this limitation was due to insufficient *diversity* of adversaries seen during training. We propose a defense using population based training to pit the victim against a range of opponents. We evaluate this defense's robustness against new adversaries in two low-dimensional environments. We find that our defense increases robustness against adversaries and show that robustness is correlated with the size of the opponent population.

1. Introduction

The discovery of *adversarial examples* (Szegedy et al., 2014) showed that capable image classifiers can be fooled by inputs that are easily classified by a human. Reinforcement learning (RL) policies have been shown to also be vulnerable to adversarial inputs (Huang et al., 2017; Kos & Song, 2017). However, attackers often are not able to directly perturb inputs of a victim agent. Gleave et al. (2020) instead model the attacker as controlling an adversarial agent in an environment shared with the victim. This adversarial agent does not have any special powers, but can *indirectly* influence the victim's observations by taking actions in the world. Gleave et al. train such *adversarial policies* using RL against a fixed victim policy, and are able to exploit state-of-the-art victim policies in zero-sum, two-agent, simulated robotics environments (Bansal et al., 2018).

This vulnerability is surprising given that self-play has produced policies that can defeat the world champions in Go (Silver et al., 2016) and Dota 2 (OpenAI et al., 2019). Critically, zero-sum games are *naturally* adversarial, with self-play being akin to adversarial training: so we might expect self-play policies to be naturally robust. Yet Gleave et al. (2020) can exploit victims with the adversary training for 3% as many timesteps as the original self-play procedure.

The resulting adversarial policies exhibit counterintuitive behaviour such as falling over and curling into a ball. Since the goal of this game is to prevent the opponent from moving to a specific area of the environment, this would be a weak strategy against most agents, but is able to destabilize the victim. Similar to adversarial examples, these policies showcase failure modes that affect a model but which a human would most likely be unaffected by. Such attacks are a critical danger to deep RL policies in high-stakes settings where there may be adversaries, such as autonomous driving or automated financial trading.

To address this, we propose using population based reinforcement learning (PBRL; Jaderberg et al., 2019) to train an agent against a diverse population of opponents. Whereas self-play trains an agent to be robust against *itself*, PBRL with a sufficient number of opponents will force an agent to be robust against a wide range of strategies. We conjecture that this will have similar benefit to adversarial training (Goodfellow et al., 2015) for classification models.

We evaluate PBRL as a defense in two simple two-player, zero-sum games. We find that the self-play baseline is easily exploited by an adversary using less than 60% as many timesteps as self-play. The PBRL trained policy is more robust: more timesteps are needed until a first adversarial policy can be found. However, we find selfplay to be comparatively more robust against attackers that train longer.

We make three key contributions. First, we introduce PBRL as an end-to-end robust training method for deep RL. Second, we evaluate this empirically, decreasing exploitability of the victim. Third, we investigate how attributes of the environment (such as dimensionality) and algorithm (such as population size) influence robustness.

[1] University of California, Berkeley [2] Technical University of Munich. Correspondence to: Pavel Czempin <pavel.czempin@berkeley.edu>.

This work was presented at the ICML Workshop New Frontiers in Adversarial Machine Learning in Conjunction with ICML 2021 (AdvML). This workshop does not have official proceedings and this paper is non-archival.

2. Related Work

Gleave et al. (2020) introduced the adversarial policy threat model, and the first attack: using RL to train an adversary against a fixed victim. Wu et al. (2021) added an auxiliary term to reward the victim paying attention to the adversary. They apply this attack to the original environments (Bansal et al., 2018), and a new environment: Roboschool Pong. Guo et al. (2021) develop a different attack for semi-competitive games, exploiting agents in the original environments and Starcraft 2.

Comparatively little attention has been given to defenses relative to attacks. Gleave et al. (2020) attempted to harden the victim by fine-tuning it against a fully trained adversary. The hardened victim was robust to that particular adversarial policy – but it was still vulnerable to repeating the attack. Furthermore, the hardened victim achieved lower performance against the original, non-adversarial opponent. Guo et al. (2021) use the same approach and find that fine-tuning on adversaries from a stronger attack can provide robustness against adversaries from weaker attacks. However, the authors don't evaluate the robustness of the hardened victim to an attack targeting itself rather than the original victim.

The most relevant prior defense is *Adversarially Robust Control* (ARC; Kuutti et al., 2021). They consider the semi-competitive setting of autonomous driving. They find that imitation-learned policies are vulnerable to adversarial vehicles trained to cause collisions – even when the adversary is limited to only cause preventable collisions. To improve robustness they fine-tune the imitation policies against an ensemble of adversaries that train concurrently with the main policy. Since autonomous driving is semi-competitive, an optimal policy against adversaries might fare poorly against regular agents, so Kuutti et al. add an auxiliary loss to keep the fine-tuned policy similar to the imitation policy. In contrast, we focus on the more challenging zero-sum setting which self-play was designed to work with.

Self-play is designed to solve zero-sum games – so it is natural to ask why it is vulnerable to such exploits in the first place. Indeed fictitious play, where each agent plays the best response to the opponent's historically observed strategy, is known to find the Nash equilibrium for two-player, zero-sum games (Robinson, 1951). Self-play is related to fictitious play, and has been shown to also converge to Nash equilibria provided that RL converges to a best response asymptotically (Heinrich et al., 2015). However, RL may never converge to a best response. A major culprit is non-transitivity: Balduzzi et al. (2019) show that in games like rock-paper-scissors, self-play may get stuck in a cycle. Even in a transitive game, deep RL algorithms may never converge to a best response if they are stuck in a local minimum, or if the policy network cannot represent the optimal policy.

3. Background

Following Gleave et al. (2020), we model the tasks as Markov games $M = (\mathcal{S}, (\mathcal{A}_\alpha, \mathcal{A}_\nu), \mu, \mathcal{T}, \gamma, (R_\alpha, R_\nu))$ (Littman, 1994) with state set $\mathcal{S}$, action sets $\mathcal{A}_i$, initial state distribution μ, transition dynamics $\mathcal{T}$, discount factor γ and reward functions R_i. The index $i \in \{\alpha, \nu\}$ represents the adversarial (α) and victim (ν) agent respectively.

We focus on two-player, zero-sum games as they have a clear competitive setup and obvious adversarial goals. Consequently $R_\alpha = -R_\nu$. Notably, the victim policy trained in a zero-sum game will be *harder* to exploit than those in a positive-sum setting, where they may have learned to cooperate with other agents (in an exploitable fashion) to maximize their overall reward.

The attacker has grey-box access to the victim. That is, they can train against a fixed victim, but cannot directly inspect its weights. The attacker has no additional capabilities to manipulate the victim or the environment.

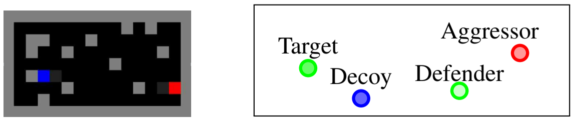
4. PBRL Defense

We would like to find a Nash equilibrium (π_ν, π_α), which for zero-sum games correspond to the minimax solutions: We take the $\arg\min_{\pi_\nu} \max_{\pi_\alpha}$ of the expectation $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_\alpha(S^{(t)}, A_\alpha^{(t)}, A_\nu^{(t)}, S^{(t+1)}) \mid \pi_\alpha, \pi_\nu\right]$, with random variables sampled from $S^{(0)} \sim \mu$, $A_i^{(t)} \sim \pi_i(\cdot \mid S^{(t)})$, and $S^{(t+1)} \sim \mathcal{T}(S^{(t)}, A_\alpha^{(t)}, A_\nu^{(t)})$.

However, as previously discussed self-play may not find the Nash equilibrium. Yet the self-play policies are nonetheless often highly capable – against their self-play opponent. We therefore conjecture that self-play gets stuck in a *local* Nash equilibrium. Notably, adversarially training against some π_α trained against π_ν might just cause π_ν to move to a new local Nash equilibrium that is robust to π_α, but not some unseen adversary π'_α.

We therefore propose using population based reinforcement learning (PBRL; Jaderberg et al., 2019), which is variation of population based training (PBT; Jaderberg et al., 2017). We train an agent, the *protagonist*, for robustness, by training it against a population of n opponents π_{o_i}. By jointly optimizing against multiple opponents we increase the coverage of the space of opponent policies. Since an adversary π'_α optimizes in a similar way as the opponents it is likely to be close to one of the opponent policies π_{o_i}. Moreover, given sufficient diversity in opponents it may be easier for the protagonist to learn a policy close to global Nash than to learn n strategies that overfit to each opponent.

Each of the n opponents has identical architecture and training objective, differing only in the seed used to randomly initialize their network. We alternate between training the



(a) Laser Tag (b) Simple Push

Figure 1: Illustrations of the (a) Laser Tag and (b) Simple Push environments. See Section 5.1 for details.

opponents against a fixed protagonist, and a protagonist against all fixed opponents. All agents (opponent and protagonist) are trained for the same total number of timesteps.

The total number of training timesteps for all policies is $n+1$ times the number of timesteps the protagonist is trained for. When logging timesteps for PBRL training, we report the number of timesteps the protagonist agent trains, since this is the relevant metric for protagonist training. Note that this means the compute necessary for training PBRL is $n + 1$ times higher than self-play at the same number of training steps (although PBRL is more parallelizable).

We train all policies – self-play, PBRL and adversarial – using Proximal Policy Optimization (PPO; Schulman et al., 2017). PPO is widely used and has achieved good results with self-play in complex environments (Bansal et al., 2018). Furthermore, prior work on adversarial policies in higher-dimensional two-agent environments uses PPO (Gleave et al., 2020). We use the PPO implementation in `RLlib` (Liang et al., 2018), from the `ray` library (Moritz et al., 2018), due to its support for multi-agent environments and parallelizing RL training.

5. Experiments

We evaluate the PBRL defense in two low-dimensional environments, described in Section 5.1. In Section 5.2 we confirm that baseline self-play policies are vulnerable to attack. To the best of our knowledge, these are the lowest-dimensional environments in which an adversarial policy has been found. Finally, in Section 5.3 we find that PBRL improves robustness against adversarial policies, and explore the relationship with population size.

Unless otherwise noted, in all experiments we train 5 seeds of victim policies. We attack each victim using 3 seeds of adversaries for a total of 15 adversaries. Unless omitted for legibility, 95% confidence intervals are shown as shaded regions for training curves and bars in bar plots.

5.1. Environments

Laser Tag is a symmetric game with incomplete information (Lanctot et al., 2017). The players see 17 spaces in front, 10 to the sides, and 2 spaces behind their agent. The two agents move on a grid world and get points for tagging each other with a light beam. Obstacles block movement and beams. We make the environment zero-sum by also subtracting a point from the tagged player.

Simple Push is a continuous environment introduced by Mordatch & Abbeel (2018) and released with Lowe et al. (2017). The environment is asymmetric, with one agent the *aggressor* and the other the *defender*.[1]

The environment contains two randomly placed landmarks. Only the defender knows which of these is the true target, the other landmark acts as a "decoy" for the aggressor. The aggressor receives positive reward based on the defenders distance to the true target. Subtracted from this is a relative penalty, based on its own distance. Unlike vanilla Simple Push, where the defender's rewards are solely based on its own distance, we make the environment zero-sum by giving the defender the negative of the aggressor's reward.

Initial experiments, whose training curves can be found in Figure 7 of the appendix, showed that the attack from Gleave et al. (2020) does not find an adversarial policy in vanilla Simple Push. As Simple Push is very low dimensional (a two-dimensional continuous control task), we develop a variant with a "cheap talk" communication channel (see Section A in the appendix) that increases the dimensionality but does not otherwise change the dynamics. This allows adversarial policies to be found, so experiments are performed with a one-hot coded communication channel of 50 tokens.

5.2. Self-play Baseline

Before evaluating our PBRL defense, we first consider the exploitability of the self-play baseline by the standard adversarial policy attack from Gleave et al. (2020). We find self-play policies in both Laser Tag and Simple Push to be vulnerable. However, in Simple Push the attack only succeeds when there is a communication channel. In Laser Tag adversarial policies can be found, however variance is high and some selfplay victims are hard to attack.

These results add nuance to the conclusion from Gleave et al. (2020) that adversarial policies are easier to find in higher-dimensional environments. Our environments are significantly lower-dimensional than those considered by Gleave et al., suggesting the minimum dimensionality for attack is smaller than previously believed. However, the fact that policies are only vulnerable in Simple Push given a communication channel, and the high variance in the victim robustness in Laser Tag, supports the overall claim that dimensionality is an important mediator for exploitability.

[1] Note, that the notion of *aggressor* and *defender* in this environment is orthogonal to *adversary* and *victim* in the sense of adversarial policies.

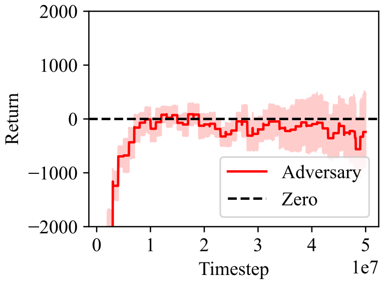
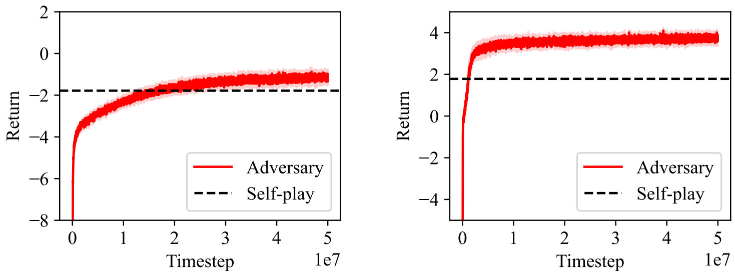


Figure 2: Training curve of adversaries in *Laser Tag*. After fewer than 15 million timesteps of training on average a successful adversarial policy can be found. However variance is high and adversary performance deteriorates against some strong victims over time.



(a) Aggressor.

(b) Defender.

Figure 3: *Simple Push* average return of adversary controlling aggressor (left) and defender (right) *with* 50-token communication channel. Adversary controlling aggressor leads to weak attack, defender leads to strong attack.

Laser Tag. We train self-play policies in the symmetric Laser Tag environment for 25 million timesteps. This should be adequate to produce a strong policy, as the paper introducing the environment (Lanctot et al., 2017) trained self-play for only 3 million timesteps. Figure 2 shows the average return of the adversaries trained against these victims. We train adversaries for 50 million timesteps, twice as many as the victims, in order to reason about the adversaries' behavior given more compute. Since the game is symmetric, an agent with a return above zero outperforms its opponent.

Successful adversarial policies can be found: if the adversary were to stop training once it outperforms the victim, on average fewer than 15 million timesteps are necessary for a successful attack. The loose confidence interval suggests high variability in different seeds. Some of the trained victims are very robust while others are not. On average attacker performance deteriorates after 20 million timesteps.

Simple Push. Since Simple Push has an asymmetric observation space, we train self-play using a separate policy for either player. We train the agents for 25 million timesteps, which we expect to be more than sufficient given the 625,000 timesteps used in prior work (Lowe et al., 2017). While prior work used MADDPG, not PPO, in exploratory experiments we find PPO to perform comparably to MADDPG.

Again we train adversaries for 50 million timesteps, twice as many as the victims, in order to reason about the adversaries' behavior given more compute. Figure 3b shows when the adversary controls the defender the adversary achieves almost twice the return on average (in red) as the self-play baseline (in black) at 25 million timesteps, which is the point where adversary and victim trained for the same number of timesteps. By contrast, Figure 3a shows that when controlling the aggressor the adversary needs around 20 million timesteps just to match the victim. Return after training the adversary for twice as many timesteps as the victim is only slightly higher than the baseline.

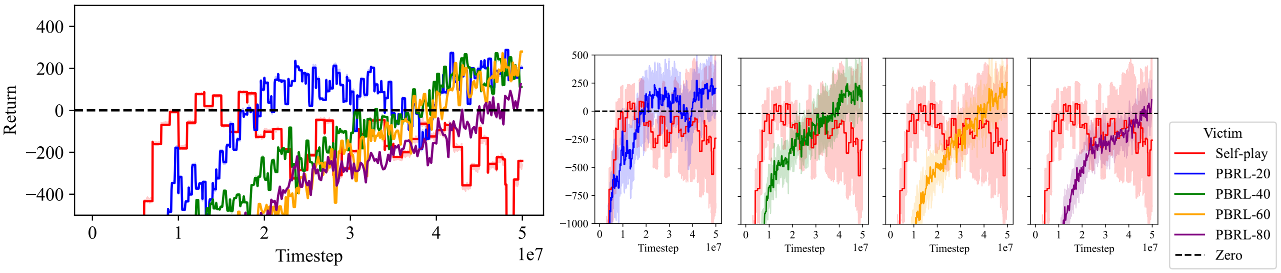
These results suggest defender policies in Simple Push are more robust to adversarial policies than aggressor policies. We conjecture this asymmetry is due the defender having more information than the aggressor: it knows the target landmark. Consequently, the aggressor needs to observe the opponent to learn the true target landmark. The defender could exploit this and perform movements that fool a victim aggressor. Due to the stronger nature of the attack controlling the defender, we focus our defense in the upcoming section on adversaries that control the defender agent.

5.3. PBRL Defense

In this section, we evaluate the effectiveness of our PBRL defense by trying to exploit PBRL-trained policies. We find some improvement in robustness relative to the self-play baseline in both environments. In Laser Tag larger populations increase the number of timesteps needed to find the first adversarial policy – at the cost of requiring more computational resources. In Simple Push there are no significant improvements with more than $n = 2$ opponents.

We evaluate our defense by attacking the PBRL-trained protagonist agent, and the baseline self-play agent. We focus on the relative performance of the adversary compared to what the victims achieve on-distribution. We continue training adversaries until 50 million timesteps against the same fixed 25-million-timestep victim to evaluate how many timesteps are needed to attack more robust victims.

Laser Tag. To explore the impact of population size we train policies with $n = 20$, 40, 60, and 80 opponents in



(a) Averages

(b) Averages with confidence intervals

Figure 4: Training curves for adversaries attacking in Laser Tag. Increasing population size reduces the exploitability by increasing the computational effort required by an attacker. Selfplay is more robust than policies trained using PBRL for adversaries that train longer, however the number of timesteps to find an initial successful attack is shorter with selfplay and increases with population size. Figure (b) shows the confidence interval for each attack and selfplay.

the Laser Tag environment. Figure 4a shows the average return of adversaries attacking these hardened protagonists. We find that using PBRL increases robustness: finding an adversary that achieves higher than 0 reward takes more timesteps on average. While fairly noisy, generally the number of timesteps needed to outperform the victim – when return crosses over the zero line – increases with increasing population size. An adversary attacking a protagonist hardened against a population of size 80 needs to train on almost double the timesteps as the selfplay victim. However, the relative difference is smaller when trained with higher population sizes.

Although the adversarial policy was trained for up to double the number of timesteps as the protagonist agent, note that PBRL used 80 times as much compute for every timestep of the protagonist, since it had to train the opponents for the same number of timesteps. Additionally, we find that adversaries that continue to train eventually do outperform PBRL victims, whereas they perform worse over time against the selfplay victim. The different confidence intervals shown in Figure 4b shows that using PBRL decreases variance compared to selfplay training which produces agents with highly varying robustness.

Simple Push. We focus on making the *aggressor* agent more robust in Simple Push, as Section 5.2 showed that the defender self-play policy is already relatively robust to attack. Consequently the adversary controls the defender agent. We use PBRL to train against n = 2, 4, 8 and 16 opponents. Since the environment is not balanced, we judge performance by comparing to the returns at the end of training as a baseline. The adversary is successful if it achieves higher results against the victim than the PBRL opponents did on average at the end of training against that same victim policy.

Figure 5 shows the returns at 25 million timesteps. Since different PBRL setups could induce different values that agents in this asymmetric setting converge to, we calculate separate baseline thresholds for each of the 4 settings (in addition to the self-play baseline from Figure 3). The PBRL-trained agent is significantly more robust than the self-play policy, in red. In fact, PBRL achieves similar return *under a zero-shot attack* as the self-play policy does *against its self-play opponent* (the dotted threshold on each bar).

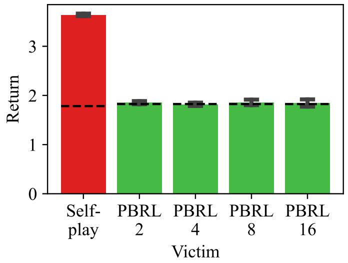


Figure 5: PBRL-trained agents (green) are significantly more robust to adversarial attack than the self-play baseline (red), and attain similar return to the self-play baseline that plays against a *non-adversarial* agent (black dotted line).

Figure 6 shows the training curve, when training adversaries for up to 50 million timesteps. Since there is no discernible difference in the 4 PBRL settings, we average over these for a total of $4 \times 15 = 60$ adversarial policies. We see that even when the adversary trains against a victim which trained for *half the number of timesteps* the attack is not particularly strong, though it does outperform the victim.

Although PBRL is significantly more robust than self-play (effectively PBRL with $n = 1$ opponents), perhaps surprisingly there is little benefit from using more than $n = 2$ opponents. In particular, there is no clear decrease in robustness when using n as low as 2, which is the lowest PBRL settings that does not degenerate to self-play.

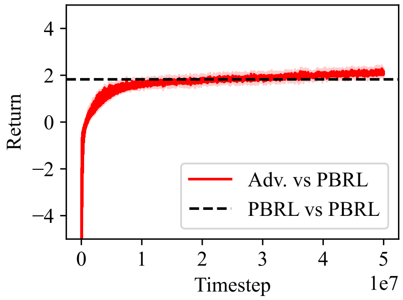


Figure 6: Return of PBRL-trained agents with $n = 2, 4, 8, 16$, averaged over all 60 adversaries, 15 per n. Victims are all similarly robust in Simple Push, adversaries need at least as many timesteps as the victim in order to outperform it.

This is in contrast to Laser Tag, which saw large differences in robustness depending on population size. This is likely due to different environmental dynamics. In Simple Push there are only a handful of high-level strategies that one can pursue. By contrast, the Laser Tag environment allows for more variation in the details of possible strategies, making it harder to achieve full robustness. Additionally, it is possible that the intervention of slightly increasing dimensionality, by adding a “cheap-talk” channel, can be circumvented with minimally higher diversity during training.

6. Discussion and Future Work

In this work we evaluated our hardened PBRL victim agent against adversarial policies trained with similar numbers of timesteps of experience as the victim agent. However, PBRL *also* requires training n opponents for this many timesteps – so the *total* number of timesteps and computational resources is $n + 1$ times greater.

We believe this overhead is often tolerable. First, defenders may be able to limit the number of timesteps an attacker can train against the victim for, such as if access to the policy is behind a rate-limited API. Second, the number of opponents n can sometimes be quite small – just $n = 2$ suffices for Simple Push. Finally, defenders often have significant computational resources: while PBRL is unlikely to prevent a sophisticated attacker like a nation state, it may be enough to defeat many low-resource attacks. Nonetheless, reducing this computational overhead is an important direction for future work. For example, can we obtain similar performance with fewer opponents if we train them to be maximally diverse from one another?

In addition, *if* additional compute resources are available, our approach allows a defender to *make use* of them. Once an agent has converged, using additional compute to continue training in self-play is usually of no use. However, convergence is not sufficient for robustness – as illustrated by the existence of adversarial policies. Our approach enables a purposeful use for additional computing power.

A key open question is how the number of opponents n required for robustness scales with the complexity of the environment. PBRL will scale poorly if the required population size is proportional to the size of the state space: in more complex environments each opponent will take longer to train *and* more opponents will be required. But a priori it seems likely that n may depend more on the number of high-level strategies in the environment. This is only loosely related to the dimensionality of the state space. For example, some simple matrix games have high strategic complexity, while some high-dimensional video games have only a handful of sensible strategies.

Our evaluation uses the original adversarial policy attack of Gleave et al. (2020), which we established was strong enough to exploit unhardened victims in these environments. However, it is possible that alternative attacks would be able to exploit even our hardened victim. We hope to see iterative development of stronger attacks and defences, similar to the trend in adversarial examples more broadly.

7. Conclusion

We introduced the defense by PBRL as a method to reduce exploitability of RL policies. Our results show an increase in zero-shot robustness against new adversaries compared to self-play training. However, this robustness comes with an increased computational cost and we find that some self-play victims are naturally robust. We find the size of the population necessary depends on the environment used, and that larger populations can increase overall robustness. This work serves as another data point in favor of increasing diversity during training – not only for increased capabilities. We hope this work is a helpful contribution towards the goal of making RL agents less exploitable. [2]

Acknowledgements

We thank Philip Sperl and researchers at the Fraunhofer Institute for Applied and Integrated Security and at the Center for Human-Compatible AI for helpful discussions. We also thank our anonymous reviewers for feedback on this paper. Work supported in part by the Berkeley Existential Risk Initative and Centre for Effective Altruism Long Term Future Fund.

[2] https://reducing-exploitability.github.io for supplementary videos, source code available at https://github.com/HumanCompatibleAI/reducing-exploitability.

References

Balduzzi, D., Garnelo, M., Bachrach, Y., Czarnecki, W., Perolat, J., Jaderberg, M., and Graepel, T. Open-ended learning in symmetric zero-sum games. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97, pp. 434–443, 2019.

Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., and Mordatch, I. Emergent complexity via multi-agent competition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., and Russell, S. Adversarial policies: Attacking deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Guo, W., Wu, X., Huang, S., and Xing, X. Adversarial policy learning in two-player competitive games. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139, pp. 3910–3919, 2021.

Heinrich, J., Lanctot, M., and Silver, D. Fictitious self-play in extensive-form games. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 37, pp. 805–813, 2015.

Huang, S. H., Papernot, N., Goodfellow, I. J., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., and Kavukcuoglu, K. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castañeda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., Sonnerat, N., Green, T., Deason, L., Leibo, J. Z., Silver, D., Hassabis, D., Kavukcuoglu, K., and Graepel, T. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Kos, J. and Song, D. Delving into adversarial attacks on deep policies. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Kuutti, S., Fallah, S., and Bowden, R. Arc: Adversarially robust control policies for autonomous vehicles. In *Proceedings of the IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 522–529, 2021.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Perolat, J., Silver, D., and Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J., Jordan, M., and Stoica, I. RLlib: Abstractions for distributed reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80, pp. 3053–3062, 2018.

Littman, M. L. Markov games as a framework for multiagent reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 157–163, 1994.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 6382–6393, 2017.

Mordatch, I. and Abbeel, P. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1495–1502, 2018.

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., and Stoica, I. Ray: A distributed framework for emerging AI applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pp. 561–577. USENIX Association, 2018.

OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., Pinto, H. P. d. O., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Robinson, J. An iterative method of solving a game. *Annals of Mathematics*, 54(2):296–301, 1951.

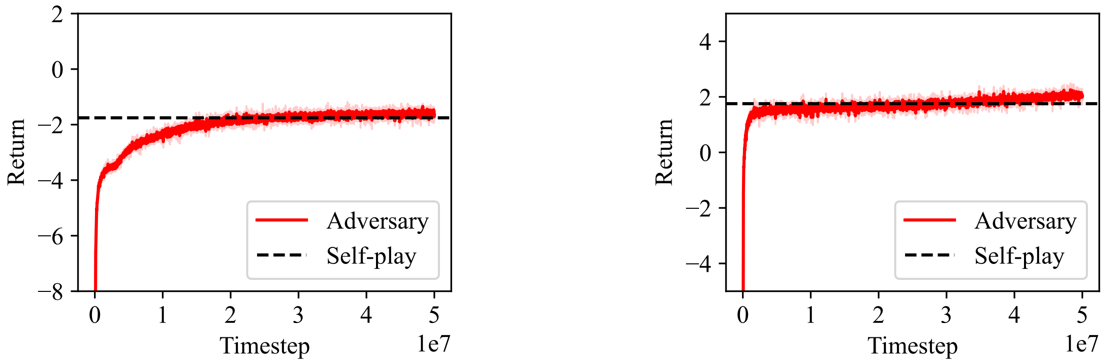
Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv:1707.06347 [cs]*, 2017.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Wu, X., Guo, W., Wei, H., and Xing, X. Adversarial policy training against deep reinforcement learning. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

A. Adding a Communication Channel to Simple Push



(a) Agressor.

(b) Defender.

Figure 7: Average return of agent controlling (a) agressor and (b) defender in Simple Push *without* communication. The adversary, in red, fails to outperform the self-play baseline, in black.

Our initial experiments in the *Simple Push* environment did not lead to adversarial policies that were capable of outperforming their victims (See Figures 7a, 7b).

Inspired by the cooperative environments explained in Lowe et al. (2017), the communication channel allows each agent to observe a one-hot coded action taken by the other agent. This communication channel has no other effect on environment dynamics, and agents reward does not depend on the contents of the communication channel. The size of the communication channel essentially represents the number of tokens either agent can use to communicate with the other. Because this setting is competitive, there is no reason for an agent to provide information in the communication channel which would be beneficial to the opponent. Therefore, an optimal policy should simply ignore this channel. However, this channel increases the dimensionality and offers an adversary the possibility to learn what messages might "confuse" a (sub-optimal) victim.

In a small ablation on communication channels supporting 10, 25, 50, 100 and 1000 tokens, we find that adversarial policies are successful with 50 or more tokens and unsuccessful at less than 25. We also find that the number of timesteps until convergence, as well as general instability during training increases with higher sizes. For further experiments we use a communication channel of 50 tokens to allow for fast training while still providing an environment in which adversarial policies are possible.