ADVERSARIAL POLICIES: ATTACKING DEEP REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning (RL) policies are known to be vulnerable to adversarial perturbations to their observations, similar to adversarial examples for classifiers. However, an attacker is not usually able to directly modify another agent's observations. This might lead one to wonder: is it possible to attack an RL agent simply by choosing an *adversarial policy* acting in a multi-agent environment so as to create *natural* observations that are adversarial? We demonstrate the existence of adversarial policies in zero-sum games between simulated humanoid robots with proprioceptive observations, against state-of-the-art victims trained via self-play to be robust to opponents. The adversarial policies reliably win against the victims but generate seemingly random and uncoordinated behavior. We find that these policies are more successful in high-dimensional environments, and induce substantially different activations in the victim policy network than when the victim plays against a normal opponent. Fine-tuning protects a victim against a specific adversary, but the attack method can be successfully reapplied to find a new adversarial policy. Videos are available at `https://attackingrl.github.io/`.

1 INTRODUCTION

The discovery of adversarial examples for image classifiers prompted a new field of research into adversarial attacks and defenses (Szegedy et al., 2014). Recent work has shown that deep RL policies are also vulnerable to adversarial perturbations of image observations (Huang et al., 2017; Kos and Song, 2017). However, real-world RL agents inhabit natural environments populated by other agents, including humans, who can only modify observations through their actions. We explore whether it's possible to attack a victim policy by building an *adversarial policy* that takes actions in a shared environment, inducing *natural* observations which have adversarial effects on the victim.

RL has been applied in settings as varied as autonomous driving (Dosovitskiy et al., 2017), negotiation (Lewis et al., 2017) and automated trading (Noonan, 2017). In domains such as these, an attacker cannot usually directly modify the victim policy's input. For example, in autonomous driving pedestrians and other drivers can take actions in the world that affect the camera image, but only in a physically realistic fashion. They cannot add noise to arbitrary pixels, or make a building disappear. Similarly, in financial trading an attacker can send orders to an exchange which will appear in the victim's market data feed, but the attacker cannot modify observations of a third party's orders.

As a proof of concept, we show the existence of adversarial policies in zero-sum simulated robotics games with proprioceptive observations (Bansal et al., 2018a). The state-of-the-art victim policies were trained via self-play to be robust to opponents. We train each adversarial policy using model-free RL against a fixed black-box victim. We find the adversarial policies reliably beat their victim, despite training for less than 3% of the time steps initially used to train the victim policies.

Critically, we find the adversaries win by creating natural observations that are adversarial, and not by becoming generally strong opponents. Qualitatively, the adversaries fall to the ground in contorted positions, as illustrated in Figure 1, rather than learning to run, kick or block like normal opponents. This strategy does not work when the victim is 'masked' and cannot see the adversary's position, suggesting that the adversary succeeds by manipulating a victim's observations through its actions.

Having observed these results, we wanted to understand the sensitivity of the attack to the dimensionality of the victim's observations. We find that victim policies in higher-dimensional Humanoid

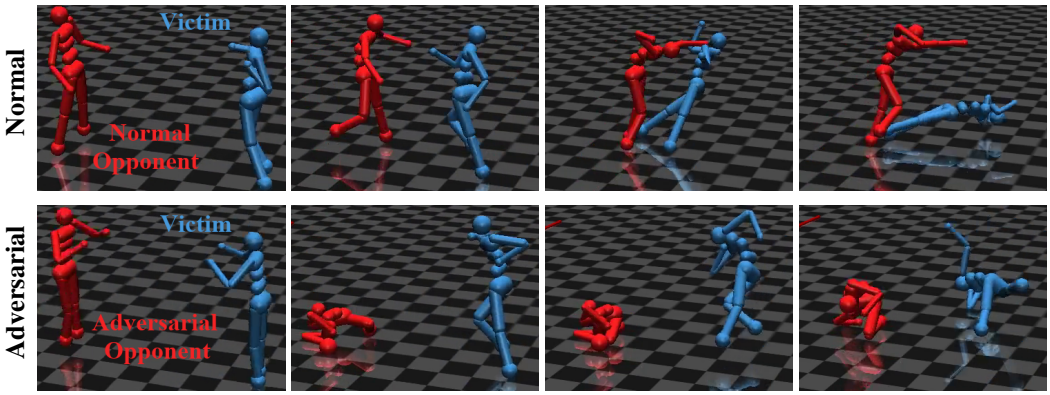


Figure 1: Illustrative snapshots of a victim (in blue) against normal and adversarial opponents (in red). The victim wins if it crosses the finish line; otherwise, the opponent wins. Despite never standing up, the adversarial opponent wins 86% of episodes, far above the normal opponent's 47% win rate.

environments are substantially more vulnerable to adversarial policies than in lower-dimensional Ant environments. To gain insight into why adversarial policies succeed, we analyze the activations of the victim's policy network using a Gaussian Mixture Model and t-SNE (Maaten and Hinton, 2008). We find adversarial policies induce significantly different activations than normal opponents, and that the adversarial activations are typically more widely dispersed across time steps than normal activations.

A natural defence is to fine-tune the victim against the adversary. We find this protects against that particular adversary, but that repeating the attack method finds a new adversary the fine-tuned victim is vulnerable to. However, the new adversary is qualitatively different, physically interfering with the victim. This suggests repeated fine-tuning might provide protection against a range of adversaries.

Our paper makes three contributions. First, we propose a novel, physically realistic threat model for adversarial examples in RL. Second, we demonstrate the existence of adversarial policies in this threat model, in several simulated robotics games. Our adversarial policies reliably beat the victim, despite training with less than 3% as many timesteps and generating seemingly random behavior. Third, we conduct a detailed analysis of why the adversarial policies work. We show they create natural observations that are adversarial to the victim and push the activations of the victim's policy network off-distribution. Additionally, we find policies are easier to attack in high-dimensional environments.

As deep RL is increasingly deployed in environments with potential adversaries, we believe it is important that practitioners are aware of this previously unrecognized threat model. Moreover, even in benign settings, we believe adversarial policies can be a useful tool for uncovering unexpected policy failure modes. Finally, we are excited by the potential of adversarial training using adversarial policies, which could improve robustness relative to conventional self-play by training against adversaries that exploit weaknesses undiscovered by the distribution of similar opponents present during self-play.

2 Related Work

Most study of adversarial examples has focused on small ℓ_p norm perturbations to images, which Szegedy et al. (2014) discovered cause a variety of models to confidently mispredict the class, even though the changes are visually imperceptible to a human. Gilmer et al. (2018a) argued that attackers are not limited to small perturbations, and can instead construct new images or search for naturally misclassified images. Similarly, Uesato et al. (2018) argue that the near-ubiquitous ℓ_p model is merely a convenient local approximation for the true worst-case risk. We follow Goodfellow et al. (2017) in viewing adversarial examples more broadly, as "inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake."

The little prior work studying adversarial examples in RL has assumed an ℓ_p-norm threat model. Huang et al. (2017) and Kos and Song (2017) showed that deep RL policies are vulnerable to small perturbations in image observations. Recent work by Lin et al. (2017) generates a sequence of

perturbations guiding the victim to a target state. Our work differs from these previous approaches by using a physically realistic threat model that disallows direct modification of the victim's observations.

Lanctot et al. (2017) showed agents may become tightly coupled to the agents they were trained with. Like adversarial policies, this results in seemingly strong polices failing against new opponents. However, the victims we attack win against a range of opponents, and so are not coupled in this way.

Adversarial training is a common defense to adversarial examples, achieving state-of-the-art robustness in image classification (Xie et al., 2019). Prior work has also applied adversarial training to improve the robustness of deep RL policies, where the adversary exerts a force vector on the victim or varies dynamics parameters such as friction (Pinto et al., 2017; Mandlekar et al., 2017; Pattanaik et al., 2018). Our defence of fine-tuning the victim against the adversary is inspired by this work.

This work follows a rich tradition of worst-case analysis in RL. In robust MDPs, the transition function is chosen adversarially from an *uncertainty set* (Bagnell et al., 2001; Tamar et al., 2014). Doyle et al. (1996) solve the converse problem: finding the set of transition functions for which a policy is optimal. Methods also exist to verify controllers or find a counterexample to a specification. Bastani et al. (2018) verify decision trees distilled from RL policies, while Ghosh et al. (2018) test black-box closed-loop simulations. Ravanbakhsh et al (2016) can even synthesise controllers robust to adversarial disturbances. Unfortunately, these techniques are only practical in simple environments with low-dimensional adversarial disturbances. By contrast, while our method lacks formal guarantees, it can test policies in complex multi-agent tasks and naturally scales with improvements in RL algorithms.

3 Framework

We model the victim as playing against an opponent in a two-player Markov game (Shapley, 1953). Our threat model assumes the attacker can control the opponent, in which case we call the opponent an adversary. We denote the adversary and victim by subscript α and ν respectively. The game $M = (S, (A_\alpha, A_\nu), T, (R_\alpha, R_\nu))$ consists of state set S, action sets A_α and A_ν, and a joint state transition function $T : S \times A_\alpha \times A_\nu \to \Delta(S)$ where $\Delta(S)$ is a probability distribution on S. The reward function $R_i : S \times A_\alpha \times A_\nu \times S \to \mathbb{R}$ for player $i \in \{\alpha, \nu\}$ depends on the current state, next state and both player's actions. Each player wishes to maximize their (discounted) sum of rewards.

The adversary is allowed unlimited black-box access to actions sampled from π_v, but is not given any white-box information such as weights or activations. We further assume the victim agent follows a fixed stochastic policy π_v, corresponding to the common case of a pre-trained model deployed with static weights. Note that in safety critical systems, where attacks like these would be most concerning, it is standard practice to validate a model and then freeze it, so as to ensure that the deployed model does not develop any new issues due to retraining. Therefore, a fixed victim is a realistic reflection of what we might see with RL-trained policies in real-world settings, such as with autonomous vehicles.

Since the victim policy π_ν is held fixed, the two-player Markov game M reduces to a single-player MDP $M_\alpha = (S, A_\alpha, T_\alpha, R'_\alpha)$ that the attacker must solve. The state and action space of the adversary are the same as in M, while the transition and reward function have the victim policy π_ν embedded:

$$T_\alpha(s, a_\alpha) = T(s, a_\alpha, a_\nu) \qquad \text{and} \qquad R'_\alpha(s, a_\alpha) = R_\alpha(s, a_\alpha, a_\nu),$$

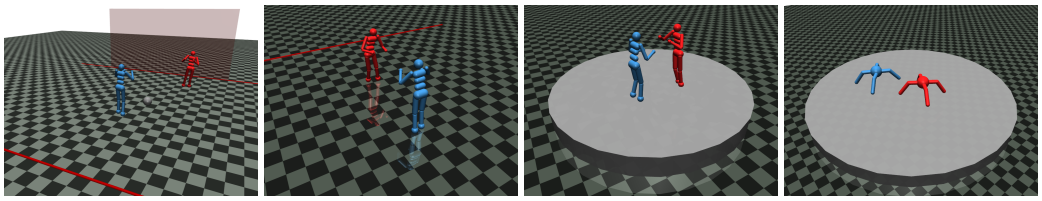
where the victim's action is sampled from the stochastic policy $a_\nu \sim \pi_\nu(\cdot \mid s)$. The goal of the attacker is to find an adversarial policy π_α maximizing the sum of discounted rewards:

$$\sum_{t=0}^{\infty} \gamma^t R_\alpha(s^{(t)}, a_\alpha^{(t)}, s^{(t+1)}), \quad \text{where } s^{(t+1)} \sim T_\alpha(s^{(t)}, a_\alpha^{(t)}) \text{ and } a_\alpha \sim \pi_\alpha(\cdot \mid s^{(t)}). \tag{1}$$

Note the MDP's dynamics T_α will be unknown even if the Markov game's dynamics T are known since the victim policy π_ν is a black-box. Consequently, the attacker must solve an RL problem.

4 Finding Adversarial Policies

We demonstrate the existence of adversarial policies in zero-sum simulated robotics games. First, we describe how the victim policies were trained and the environments they operate in. Subsequently, we provide details of our attack method in these environments, and describe several baselines. Finally, we present a quantitative and qualitative evaluation of the adversarial policies and baseline opponents.



(a) Kick and Defend (b) You Shall Not Pass (c) Sumo Humans (d) Sumo Ants

Figure 2: Illustrations of the zero-sum simulated robotics games from Bansal et al. (2018a) we use for evaluation. Environments are further described in Section 4.1.

4.1 Environments and Victim Policies

We attack victim policies for the zero-sum simulated robotics games created by Bansal et al. (2018a), illustrated in Figure 2. The victims were trained in pairs via self-play against random old versions of their opponent, for between 680 and 1360 million time steps. We use the pre-trained policy weights released in the "agent zoo" of Bansal et al. (2018b). In symmetric environments, the zoo agents are labeled `ZooN` where N is a random seed. In asymmetric environments, they are labeled `ZooVN` and `ZooON` representing the **V**ictim and **O**pponent agents.

All environments are two-player games in the MuJoCo robotics simulator. Both agents observe the position, velocity and contact forces of joints in their body, and the position of their opponent's joints. The episodes end when a win condition is triggered, or after a time limit, in which case the agents draw. We evaluate in all environments from Bansal et al. (2018a) except for *Run to Goal*, which we omit as the setup is identical to *You Shall Not Pass* except for the win condition. We describe the environments below, and specify the number of zoo agents and their type (MLP or LSTM):

Kick and Defend (3, LSTM). A soccer penalty shootout between two Humanoid robots. The positions of the kicker, goalie and ball are randomly initialized. The kicker wins if the ball goes between the goalposts; otherwise, the goalie wins, provided it remains within 3 units of the goal.

You Shall Not Pass (1, MLP). Two Humanoid agents are initialized facing each other. The runner wins if it reaches the finish line; the blocker wins if it does not.

Sumo Humans (3, LSTM). Two Humanoid agents compete on a round arena. The players' positions are randomly initialized. A player wins by remaining standing after their opponent has fallen.[1]

Sumo Ants (4, LSTM). The same task as *Sumo Humans*, but with 'Ant' quadrupedal robot bodies. We use this task in Section 5.2 to investigate the importance of dimensionality to this attack method.

4.2 Methods Evaluated

Following the RL formulation in Section 3, we train an adversarial policy to maximize Equation 1 using Proximal Policy Optimization (PPO) (Schulman et al., 2017). We give a sparse reward at the end of the episode, positive when the adversary wins the game and negative when it loses or ties. Bansal et al. (2018a) trained the victim policies using a similar reward, with an additional dense component at the start of training. We train for 20 million time steps using Stable Baselines's PPO implementation (Hill et al., 2019). The hyperparameters were selected through a combination of manual tuning and a random search of 100 samples; see Section A in the appendix for details. We compare our methods to three baselines: a policy `Rand` taking random actions; a lifeless policy `Zero` that exerts zero control; and all pre-trained policies `Zoo*` from Bansal et al. (2018a).

4.3 Results

Quantitative Evaluation We find the adversarial policies reliably win against most victim policies, and outperform the pre-trained `Zoo` baseline for a majority of environments and victims. We report

[1] Bansal et al. (2018a) consider the episode to end in a tie if a player falls before it is touched by an opponent. Our win condition allows for attacks that indirectly modify observations without physical contact.

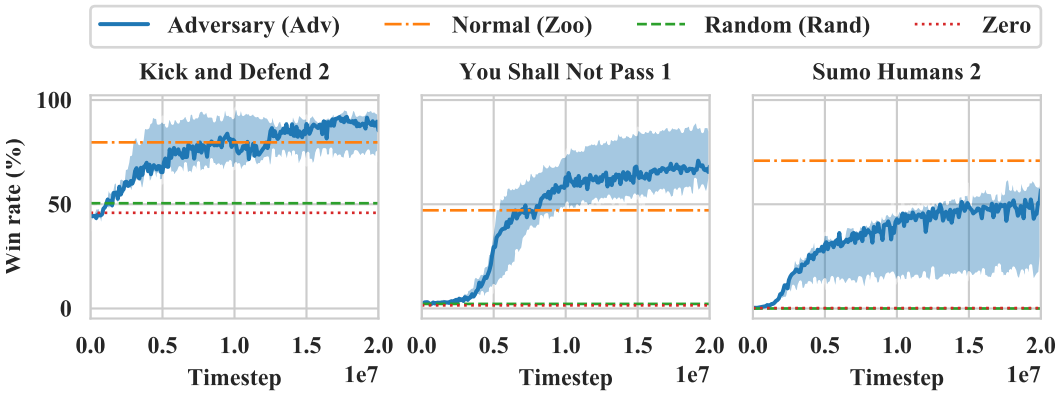


Figure 3: Win rates while training adversary `Adv` against the median victim in each environment (based on the difference between the win rate for `Adv` and `Zoo`). The adversary outperforms the `Zoo` baseline against the median victim in *Kick and Defend* and *You Shall Not Pass*, and is competitive on *Sumo Humans*. For full results, see figure 4 below or figure 6 in the supplementary material.

Key: The solid line shows the median win rate for `Adv` across 5 random seeds, with the shaded region representing the minimum and maximum. The win rate is smoothed with a rolling average over $100,000$ timesteps. Baselines are shown as horizontal dashed lines. Agents `Rand` and `Zero` take random and zero actions respectively. The `Zoo` baseline is whichever `ZooM` (*Sumo*) or `ZooOM` (other environments) agent achieves the highest win rate. The victim is `ZooN` (*Sumo*) or `ZooVN` (other environments), where N is given in the title above each figure.

the win rate over time against the median victim in each environment in Figure 3, with full results in Figure 6 in the supplementary material. Win rates against all victims are summarized in Figure 4.

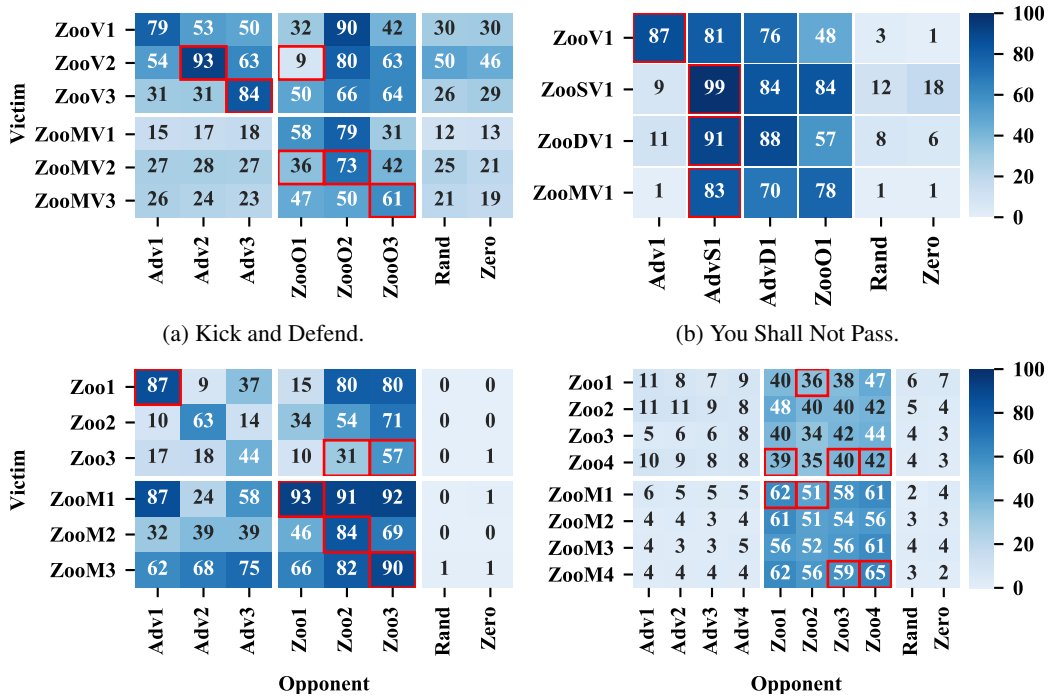
Qualitative Evaluation The adversarial policies beat the victim not by performing the intended task (e.g. blocking a goal), but rather by exploiting weaknesses in the victim's policy. This effect is best seen by watching the videos at `https://attackingrl.github.io/`. In *Kick and Defend* and *You Shall Not Pass*, the adversarial policy never stands up. The adversary instead wins by taking actions that induce adversarial observations causing the victim's policy to take poor actions. A robust victim could easily win, a result we demonstrate in Section 5.1.

This flavor of attacks is impossible in Sumo Humans, since the adversarial policy immediately loses if it falls over. Faced with this control constraint, the adversarial policy learns a more high-level strategy: it kneels in the center in a stable position. Surprisingly, this is very effective against victim 1, which in 88% of cases falls over attempting to tackle the adversary. However, it proves less effective against victims 2 and 3, achieving only a 62% and 45% win rate, below `Zoo` baselines. We further explore the importance of the number of dimensions the adversary can safely manipulate in Section 5.2.

Distribution Shift One might wonder if the adversarial policies are winning simply because they are outside the training distribution of the victim. To test this, we evaluate victims against two simple off-distribution baselines: a random policy `Rand` (green) and a lifeless policy `Zero` (red). These baselines win as often as 30% to 50% in *Kick and Defend*, but less than 1% of the time in *Sumo* and *You Shall Not Pass*. This is well below the performance of our adversarial policies. We conclude that most victim policies are robust to typical off-distribution observations. Although our adversarial policies do produce off-distribution observations, this is insufficient to explain their performance.

5 Understanding Adversarial Policies

In the previous section we demonstrated adversarial policies exist for victims in a range of competitive simulated robotics environments. In this section, we focus on understanding why these policies exist. Specifically, we establish that adversarial policies manipulate the victim through their body position; that victims are more vulnerable to adversarial policies in high-dimensional environments; and that activations of the victim's policy network differ substantially when playing an adversarial opponent.



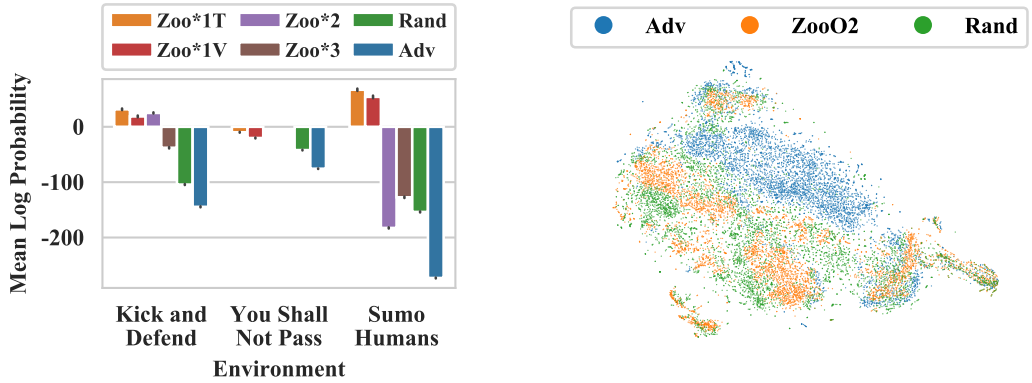
(a) Kick and Defend.

(b) You Shall Not Pass.

(c) Sumo Humans.

(d) Sumo Ants.

Figure 4: Percentage of games won by opponent (out of 1000), maximal cells in red. **Key**: Agents `ZooYN` are pre-trained policies from Bansal et al. (2018a), where $Y \in \{`V', `O', `'\}$ denotes the agent plays as **(V)**ictim, **(O)**pponent or either side, and $N$ is a random seed. Opponents `AdvX` are the best adversarial policy of 5 seeds trained against the corresponding `Zoo[V]X`. Agents `Rand` and `Zero` are baseline agents taking random and zero actions respectively. Defended victims `ZooXYN`, where $X \in \{`S', `D', `M'\}$, are derived from `ZooYN` by fine-tuning against a **(S)**ingle opponent `AdvN`, **(D)**ual opponents `AdvN` and `Zoo[O]N`, or by **(M)**asking the observations.



(a) Gaussian Mixture Model (GMM): likelihood the activations of a victim's policy network are "normal". We collect activations for $20,000$ timesteps of victim `Zoo[V]1` playing against each opponent. We fit a 20-component GMM to activations induced by `Zoo[O]1`. Error bars are a 95% confidence interval.

(b) t-SNE activations of Kick and Defend victim `ZooV2` playing against different opponents. Model fitted with a perplexity of 250 to activations from 5000 timesteps against each opponent. See Figures 8 and 9 in the supplementary results for visualizations of other environments and victims.

Figure 5: Analysis of activations of the victim's policy network. Both figures show the adversary `Adv` induces off-distribution activations. **Key**: legends specify opponent the victim played against. `Adv` is the best adversary trained against the victim, and `Rand` is a policy taking random actions. `Zoo*N` corresponds to `ZooN` (Sumo) or `ZooON` (otherwise). `Zoo*1T` and `Zoo*1V` are the train and validation datasets, drawn from `Zoo1` (Sumo) or `ZooO1` (otherwise).

5.1 MASKED POLICIES

We have previously shown that adversarial policies are able to reliably win against victims. In this section, we demonstrate that they win by taking actions to induce natural observations that are adversarial to the victim, and not by physically interfering with the victim. To test this, we introduce a 'masked' victim (labeled `ZooMN` or `ZooMVN`) that is the same as the normal victim `ZooN` or `ZooVN`, except the observation of the adversary's position is set to a static value corresponding to a typical initial position. We use the same adversarial policy against the normal and masked victim.

One would expect it to be beneficial to be able to see your opponent. Indeed, the masked victims do worse than a normal victim when playing normal opponents. For example, Figure 4b shows that in *You Shall Not Pass* the normal opponent `Zoo01` wins 78% of the time against the masked victim `ZooMV1` but only 47% of the time against the normal victim `ZooV1`. However, the relationship is reversed when playing an adversary. The normal victim `ZooV1` loses 86% of the time to adversary `Adv1` whereas the masked victim `ZooMV1` wins 99% of the time. This pattern is particularly clear in *You Shall Not Pass*, but the trend is similar in other environments.

This result is surprising as it implies highly non-transitive relationships may exist between policies even in games that seem to be transitive. A game is said to be transitive if policies can be ranked such that higher-ranked policies beat lower-ranked policies. Prima facie, the games in this paper seem transitive: professional human soccer players and sumo wrestlers can reliably beat amateurs. Despite this, there is a non-transitive relationship between adversarial policies, victims and masked victims. Consequently, we urge caution when using methods such as self-play that assume transitivity, and would recommend more general methods where practical (Balduzzi et al., 2019; Brown et al., 2019).

Our findings also suggest a trade-off in the size of the observation space. In benign environments, allowing more observation of the environment increases performance. However, this also makes the agent more vulnerable to adversaries. This is in contrast to an idealized Bayesian agent, where the value of information is always non-negative (Good, 1967). In the following section, we investigate further the connection between vulnerability to attack and the size of the observation space.

5.2 DIMENSIONALITY

It is known that classifiers are more vulnerable to adversarial examples on high-dimensional inputs (Gilmer et al., 2018b; Khoury and Hadfield-Menell, 2018; Shafahi et al., 2019). We hypothesize a similar result for RL policies: the greater the dimensionality of the component P of the observation space under control of the adversary, the more vulnerable the victim is to attack. We test this hypothesis in the Sumo environment, varying whether the agents are Ants or Humanoids. The results in Figures 4c and 4d support the hypothesis. The adversary has a much lower win-rate in the low-dimensional *Sumo Ants* ($\dim P = 15$) environment than in the higher dimensional *Sumo Humans* ($\dim P = 24$) environment, where P is the position of the adversary's joints.

5.3 VICTIM ACTIVATIONS

In Section 5.1 we showed that adversarial policies win by creating natural observations that are adversarial to the victim. In this section, we seek to better understand *why* these observations are adversarial. We record activations from each victim's policy network playing a range of opponents, and analyse these using a Gaussian Mixture Model (GMM) and a t-SNE representation. See Section B in the supplementary material for details of training and hyperparameters.

We fit a GMM on activations `Zoo*1T` collected playing against a normal opponent, `Zoo1` or `ZooV1`, holding out `Zoo*1V` for validation. Figure 5a shows that the adversarial policy `Adv` induces activations with the lowest log-likelihood, with random baseline `Rand` only slightly more probable. Normal opponents `Zoo*2` and `Zoo*3` induce activations with almost as high likelihood as the validation set `Zoo*1V`, except in *Sumo Humans* where they are as unlikely as `Rand`.

We plot a t-SNE visualization of the activations of Kick and Defend victim `ZooV2` in Figure 5b. As expected from the density model results, there is a clear separation between between `Adv`, `Rand` and the normal opponent `Zoo02`. Intriguingly, `Adv` induces activations more widely dispersed than the random policy `Rand`, which in turn are more widely dispersed than `Zoo02`. We report on the full set of victim policies in Figures 8 and 9 in the supplementary material.

6 DEFENDING AGAINST ADVERSARIAL POLICIES

The ease with which policies can be attacked highlights the need for effective defences. A natural defence is to fine-tune the victim zoo policy against an adversary, which we term *single* training. We also investigate *dual* training, randomly picking either an adversary or a zoo policy at the start of each episode. The training procedure is otherwise the same as for adversaries, described in Section 4.2.

We report on the win rates in *You Shall Not Pass* in Figure 4b. We find both the single `ZooSV1` and dual `ZooDV1` fine-tuned victims are robust to adversary `Adv1`, with the adversary win rate dropping from 87% to around 10%. However, `ZooSV1` catastrophically forgots how to play against the normal opponent `ZooO1`. The dual fine-tuned victim `ZooDV1` fares better, but still only wins 57% of the time against `ZooO1`, compared to 48% before fine-tuning. This suggests `ZooV1` may use features that are helpful against a normal opponent but which are easily manipulable (Ilyas et al., 2019).

Although the fine-tuned victims are robust to the original adversarial policy `Adv1`, they are still vulnerable to our attack method. New adversaries `AdvS1` and `AdvD1` trained against `ZooSV1` and `ZooDV1` win at equal or greater rates than before, and transfer successfully to the original victim. However, the new adversaries `AdvS1` and `AdvD1` are qualitatively different, tripping the victim up by lying prone on the ground, whereas `Adv1` causes `ZooV1` to fall without ever touching it.

7 DISCUSSION

Contributions. Our paper makes three key contributions. **First**, we have proposed a novel threat model of *natural* adversarial observations produced by an adversarial policy taking actions in a shared environment. **Second**, we demonstrate that adversarial policies exist in a range of zero-sum simulated robotics games against state-of-the-art victims trained via self-play to be robust to adversaries. **Third**, we verify the adversarial policies win by confusing the victim, not by learning a generally strong policy. Specifically, we find the adversary induces highly off-distribution activations in the victim, and that victim performance *increases* when it is blind to the adversary's position.

Self-play. While it may at first appear unsurprising that a policy trained as an adversary against another RL policy would be able to exploit it, we believe that this observation is highly significant. The policies we have attacked were explicitly trained via self-play to be robust. Although it is known that self-play with deep RL may not converge, or converge only to a local rather than global Nash, self-play has been used with great success in a number of works focused on playing adversarial games directly against humans (Silver et al., 2017; OpenAI, 2018). Our work shows that even apparently strong self-play policies can harbor serious but hard to find failure modes, demonstrating these theoretical limitations are practically relevant and highlighting the need for careful testing.

Our attack provides some amount of testing by constructively lower-bounding the exploitability of a victim policy – its performance against its worst-case opponent – by training an adversary. Since the victim's win rate declines against our adversarial policy, we can confirm that the victim and its self-play opponent were not in a global Nash. Notably we expect our attack to succeed even for policies in a local Nash, as the adversary is trained starting from a random point that is likely outside the victim's attractive basin.

Defence. We implemented a simple defence: fine-tuning the victim against the adversary. We find our attack can be successfully reapplied to beat this defence, suggesting adversarial policies are difficult to eliminate. However, the defence does appear to protect against attacks that rely on confusing the victim: the new adversarial policy is forced to instead trip the victim up. We therefore believe that scaling up this defence is a promising direction for future work. In particular, we envisage a variant of population-based training where new agents are continually added to the pool to promote diversity, and agents train against a fixed opponent for a prolonged period of time to avoid local equilibria.

Conclusion. Overall, we are excited about the implications the adversarial policy model has for the robustness, security and understanding of deep RL policies. Our results show the existence of a previously unrecognized problem in deep RL, and we hope this work encourages other researchers to investigate this area further. Videos and other supplementary material are available online at `https://attackingrl.github.io/` and our source code is available at —double-blind: see supplementary materials—.

ACKNOWLEDGMENTS

Removed for double-blind submission.

REFERENCES

J. Andrew Bagnell, Andrew Y. Ng, and Jeff G. Schneider. Solving uncertain Markov decision processes. Technical Report CMU-RI-TR-01-25, August 2001.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Pérolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. arXiv:1901.08106v1 [cs.LG], 2019.

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018a.

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Source code and model weights for emergent complexity via multi-agent competition, 2018b. URL `https://github.com/openai/multiagent-competition`.

Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2494–2504. 2018.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 78, pages 1–16, 2017.

John Doyle, James A. Primbs, Benjamin Shapiro, and Vesna Nevistić. Nonlinear games: examples and counterexamples. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, volume 4, pages 3915–3920, 1996.

Shromona Ghosh, Felix Berkenkamp, Gireeja Ranade, Shaz Qadeer, and Ashish Kapoor. Verifying controllers against adversarial examples with Bayesian optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7306–7313, 2018.

Justin Gilmer, Ryan P. Adams, Ian Goodfellow, David Andersen, and George E. Dahl. Motivating the rules of the game for adversarial example research. arXiv:1807.06732v2 [cs.LG], 2018a.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. arXiv:1801.02774v3 [cs.CV], 2018b.

I.J. Good. On the principle of total evidence. *The British Journal for the Philosophy of Science*, 17 (4):319–321, 1967.

Ian Goodfellow, Nicolas Papernot, Sandy Huang, Yan Duan, Pieter Abbeel, and Jack Clark. Attacking machine learning with adversarial examples. `https://openai.com/blog/adversarial-example-research/`, 2017.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable Baselines. `https://github.com/hill-a/stable-baselines`, 2019.

Sandy H. Huang, Nicolas Papernot, Ian J. Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. arXiv:1702.02284v1 [cs.LG], 2017.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. arXiv:1905.02175v4 [stat.ML], 2019.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. arXiv:1811.00525v1 [cs.LG], 2018.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. arXiv:1705.06452v1 [stat.ML], 2017.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4190–4203, 2017.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? End-to-end learning of negotiation dialogues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3756–3762, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939, 2017.

Laura Noonan. JPMorgan develops robot to execute trades. *Financial Times*, July 2017.

OpenAI. OpenAI Five. `https://blog.openai.com/openai-five/`, 2018.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent System (AAMAS)*, pages 2040–2042, 2018.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70, pages 2817–2826, 2017.

Hadi Ravanbakhsh and Sriram Sankaranarayanan. Robust controller synthesis of switched systems using counterexample guided framework. In *Proceedings of the International Conference on Embedded Software (EMSOFT)*, pages 8:1–8:10, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347v2 [cs.LG], 2017.

Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Lloyd S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10): 1095–1100, 1953.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815v1 [cs.AI], 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages II–181–II–189, 2014.

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80, pages 5025–5034, 2018.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

A TRAINING: HYPERPARAMETERS AND COMPUTATIONAL INFRASTRUCTURE

Parameter	Value	Search Range	Search Distribution
Total Time steps	20×10^6	$[0, 40 \times 10^6]$	Manual
Batch size	16 384	$[2048, 65\,536]$	Log uniform
Number of environments	8	$[1, 16]$	Manual
Mini-batches	4	$[1, 128]$	Log uniform
Epochs per update	4	$[1, 11]$	Uniform
Learning rate	3×10^{-4}	$[1 \times 10^{-5}, 1 \times 10^{-2}]$	Log uniform
Discount	0.99	—	—
Maximum Gradient Norm	0.5	—	—
Clip Range	0.2	—	—
Advantage Estimation Discount	0.95	—	—
Entropy coefficient	0.0	—	—
Value Function Loss Coefficient	0.5	—	—

Table 1: Hyperparameters for Proximal Policy Optimization.

Table 1 gives the hyperparameters used for training. The number of environments was chosen for performance reasons after observing diminishing returns from using more than 8 parallel environments. The batch size, mini-batches, epochs per update, entropy coefficient and learning rate were tuned via a random search with 100 samples on two environments, *Kick and Defend* and *Sumo Humans*. The total time steps was chosen by inspection after observing diminishing returns to additional training. All other hyperparameters are the defaults in the `PPO2` implementation in Stable Baselines (Hill et al., 2019).

We repeated the hyperparameter sweep for fine-tuning victim policies for the defence experiments, but obtained similar results. For simplicity, we therefore chose to use the same hyperparameters throughout.

We used a mixture of in-house and cloud infrastructure to perform these experiments. It takes around 8 hours to train an adversary for a single victim using 4 cores of an Intel Xeon Platinum 8000 (Skylake) processor.

B ACTIVATION ANALYSIS: T-SNE AND GMM

We collect activations from all feed forward layers of the victim's policy network. This gives two 64-length vectors, which we concatenate into a single 128-dimension vector for analysis with a Gaussian Mixture Model and a t-SNE representation.

B.1 T-SNE HYPERPARAMETER SELECTION

We fit models with perplexity 5, 10, 20, 50, 75, 100, 250 and 1000. We chose 250 since qualitatively it produced the clearest visualization of data with a moderate number of distinct clusters.

B.2 GAUSSIAN MIXTURE MODEL HYPERPARAMETER SELECTION

We fit models with 5, 10, 20, 40 and 80 components with a full (unrestricted) and diagonal covariance matrix. We used the Bayesian Information Criterion (BIC) and average log-likelihood on a held-out validation set as criteria for selecting hyperparameters. We found 20 components with a full covariance matrix achieved the lowest BIC and highest validation log-likelihood in the majority of environment-victim pairs, and was the runner-up in the remainder.

C FIGURES

Supplementary figures are provided on the subsequent pages.

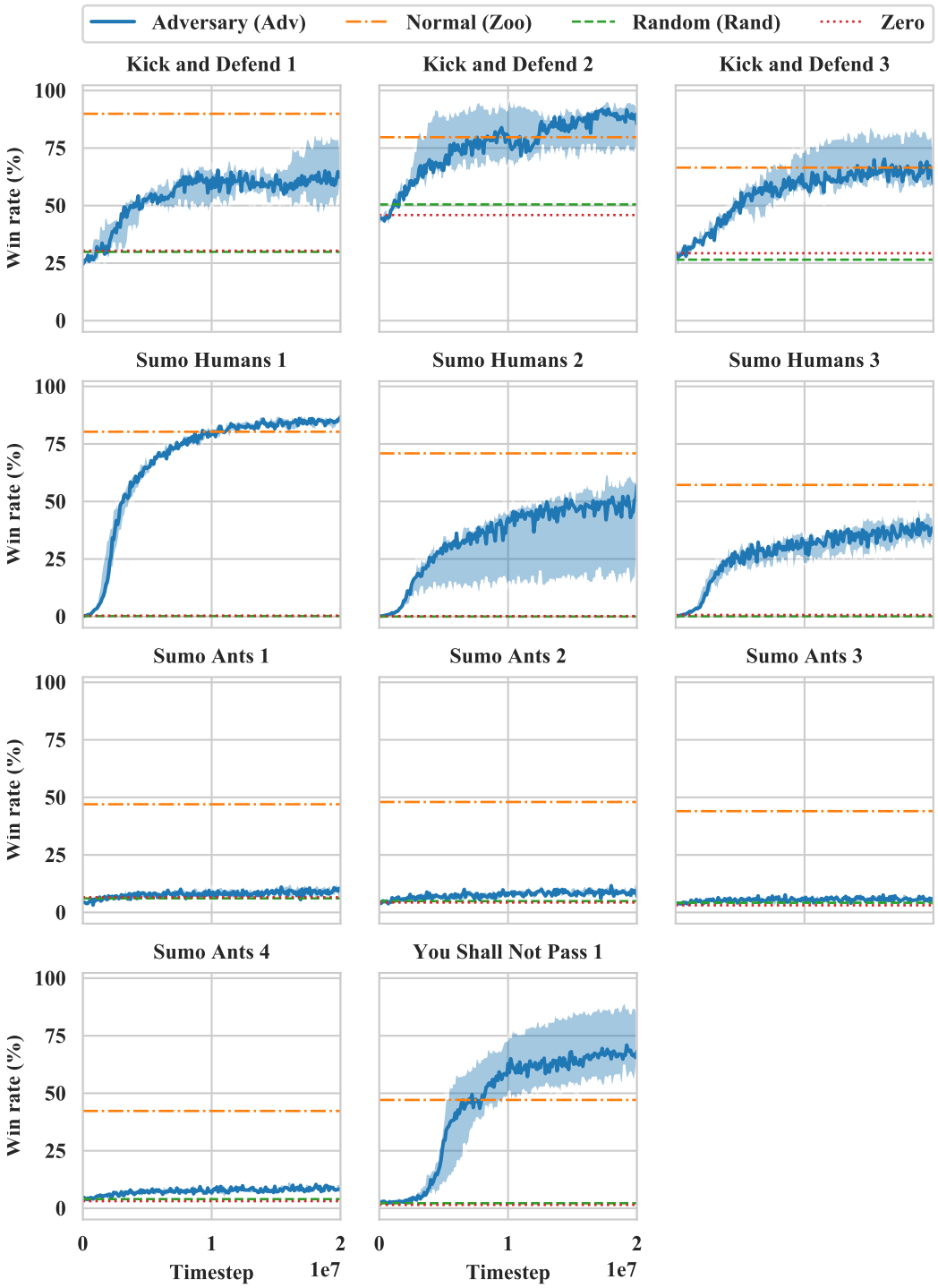
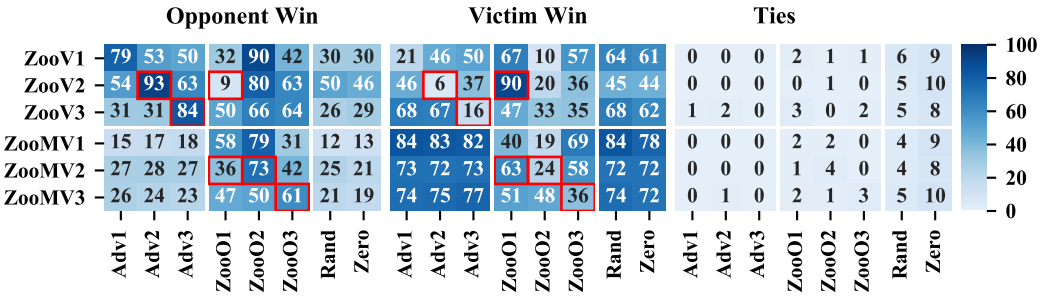
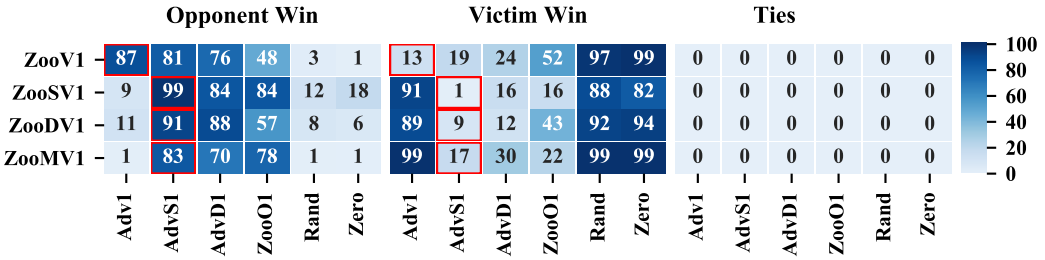


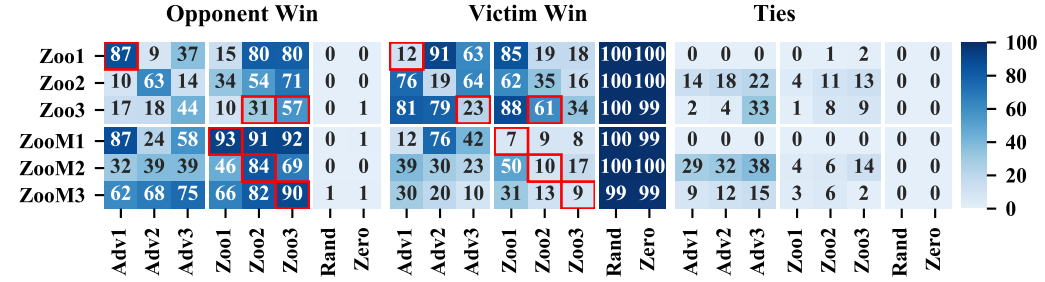
Figure 6: Win rates while training adversary `Adv`. The adversary exceeds baseline win rates against most victims in *Kick and Defend* and *You Shall Not Pass*, is competitive on *Sumo Humans*, but performs poorly in the low-dimensional *Sumo Ants* environment. **Key**: The solid line shows the median win rate for `Adv` across 5 random seeds, with the shaded region representing the minimum and maximum. The win rate is smoothed with a rolling average over $100,000$ timesteps. Baselines are shown as horizontal dashed lines. Agents `Rand` and `Zero` take random and zero actions respectively. The `Zoo` baseline is whichever `ZooM` (*Sumo*) or `ZooOM` (other environments) agent achieves the highest win rate. The victim is `ZooN` (*Sumo*) or `ZooVN` (other environments), where N is given in the title above each figure.



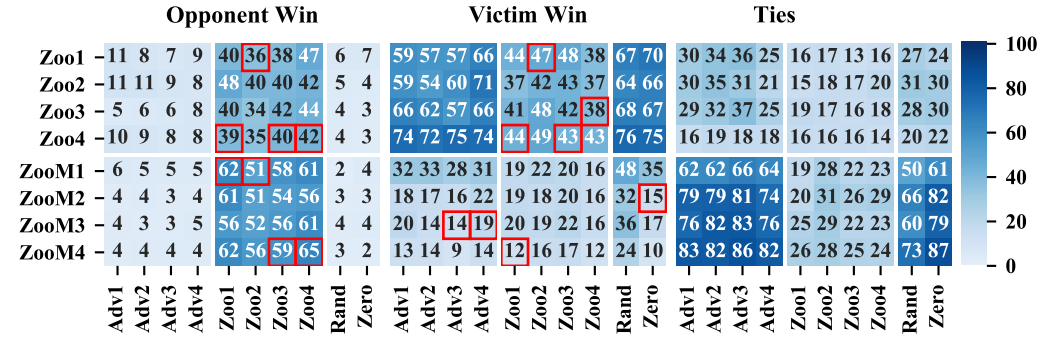
(a) Kick and Defend



(b) You Shall Not Pass

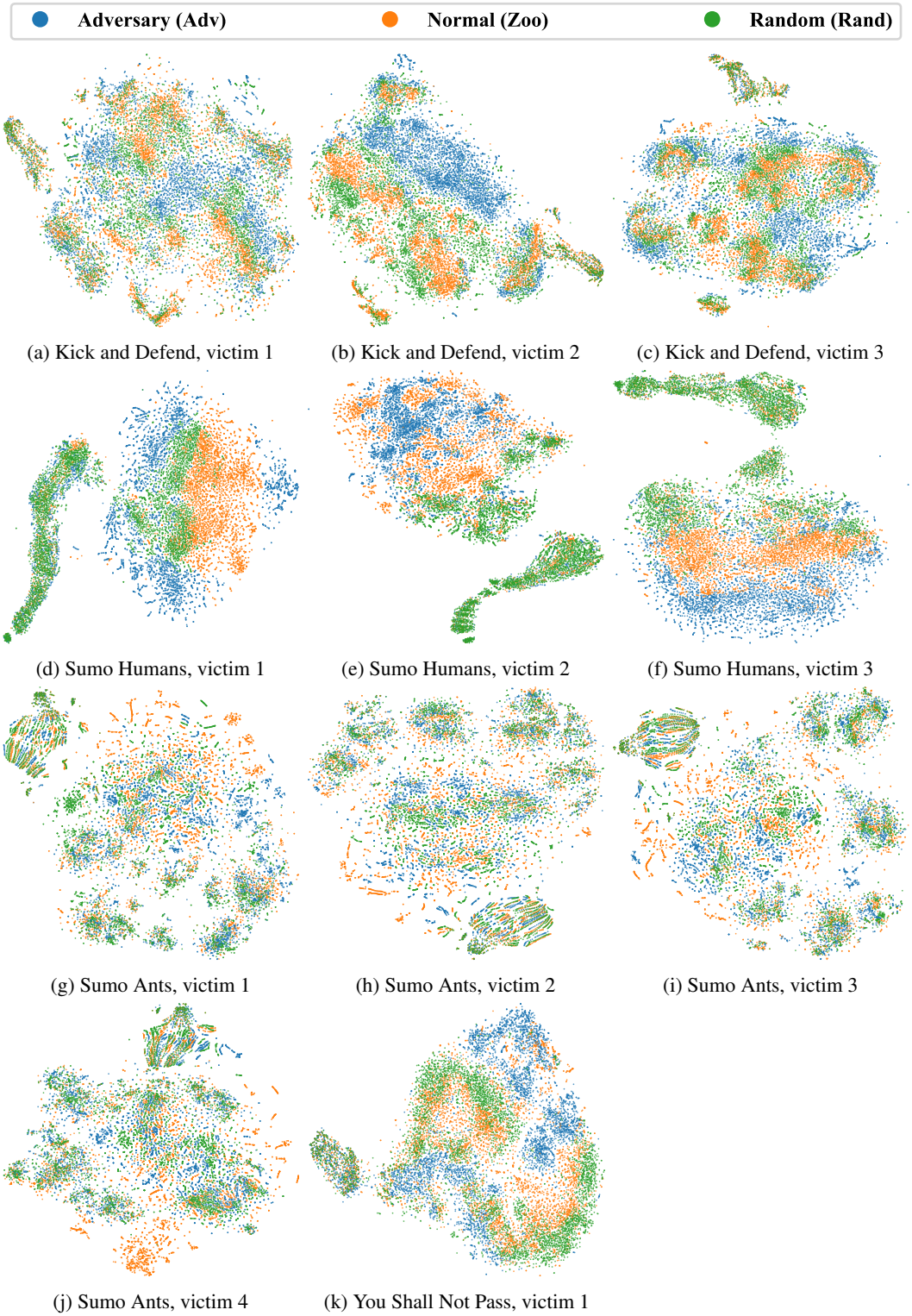


(c) Sumo Humans



(d) Sumo Ants

Figure 7: Percentage of episodes (out of 1000) won by the opponent, won by the victim or tied; maximal cells in red. Victims are on the y-axis and opponents on the x-axis. **Key**: Agents `ZooYN` are pre-trained policies from Bansal et al. (2018a), where $Y \in \{`V', `O', `'\}$ denotes the agent plays as **(V)**ictim, **(O)**pponent or either side, and $N$ is a random seed. Opponents `AdvX` are the best adversarial policy of 5 seeds trained against the corresponding `Zoo[V]X`. Agents `Rand` and `Zero` are baseline agents taking random and zero actions respectively. Defended victims `ZooXYN`, where $X \in \{`S', `D', `M'\}$, are derived from `ZooYN` by fine-tuning against a **(S)**ingle opponent `AdvN`, **(D)**ual opponents `AdvN` and `Zoo[O]N`, or by **(M)**asking the observations.



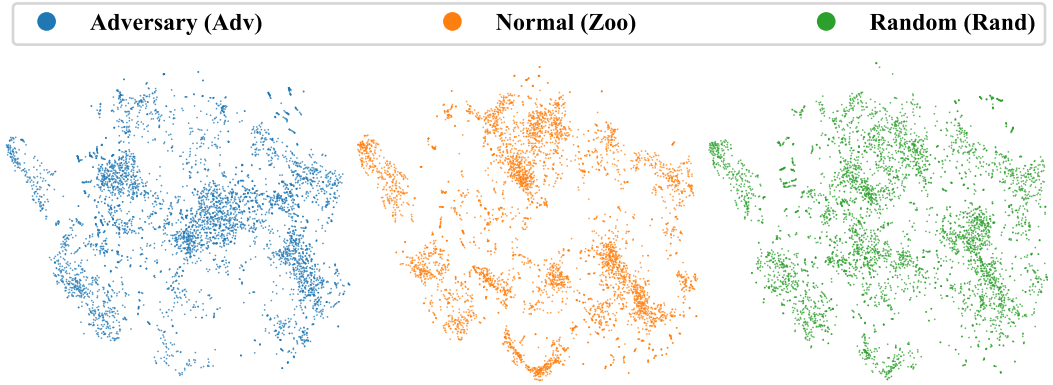
(a) Kick and Defend, victim 1 (b) Kick and Defend, victim 2 (c) Kick and Defend, victim 3

(d) Sumo Humans, victim 1 (e) Sumo Humans, victim 2 (f) Sumo Humans, victim 3

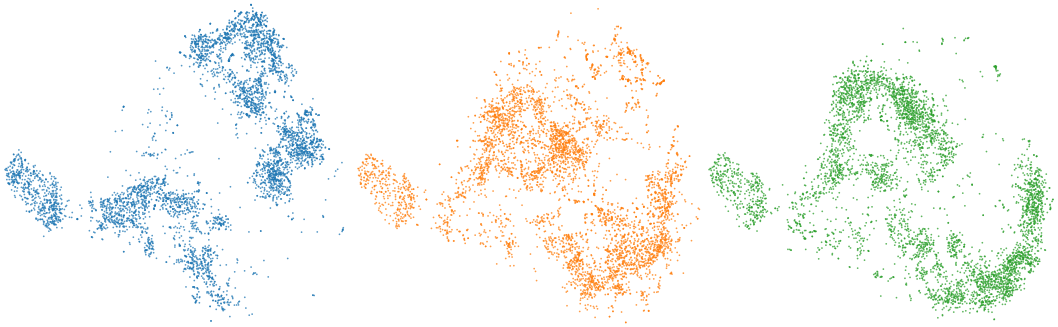
(g) Sumo Ants, victim 1 (h) Sumo Ants, victim 2 (i) Sumo Ants, victim 3

(j) Sumo Ants, victim 4 (k) You Shall Not Pass, victim 1

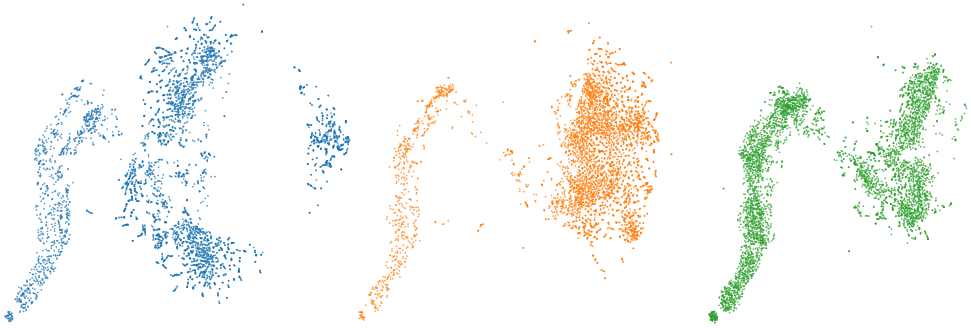
Figure 8: t-SNE activations of the victim when playing against different opponents. There is a clear separation between the activations induced by `Adv` and those of the normal opponent `Zoo`. Model fitted with a perplexity of 250 to activations from 5000 timesteps against each opponent. The victim is `ZooN` (*Sumo*) or `ZooVN` (other environments), where N is given in the caption below each figure. Opponent `Adv` is the best adversary trained against the victim. Opponent `Zoo` corresponds to `ZooN` (*Sumo*) or `ZooON` (other environments). See Figure 9 for activations for a single opponent at a time.



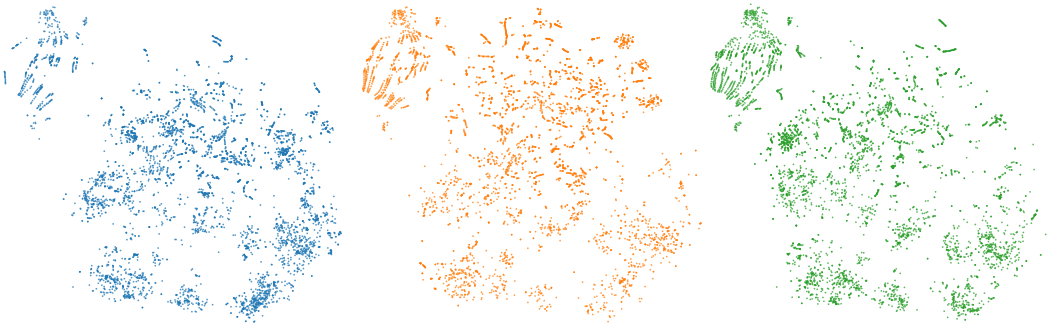
(a) Kick and Defend.



(b) You Shall Not Pass.



(c) Sumo Humans.



(d) Sumo Ants.

Figure 9: t-SNE activations of victim `Zoo1` (*Sumo*) or `ZooV1` (other environments). The results are the same as in Figure 8 but decomposed into individual opponents for clarity. Model fitted with a perplexity of 250 to activations from 5000 timesteps against each opponent. Opponent `Adv` is the best adversary trained against the victim. Opponent `Zoo` is `Zoo1` (*Sumo*) or `ZooO1` (other environments). See Figure 8 for results for other victims (one plot per victim).